Institute of Control Sciences
named by V.A.Trapeznikov
Russian Academy of Sciences
(Moscow, Russian Federation)

Ariel University Center of
Samaria (Ariel, Israel)

D. GOLENKO-GINZBURG

STOCHASTIC NETWORK MODELS IN INNOVATIVE PROJECTING

Воронеж
«Научная книга»
2011

УДК 004.5
ББК 65.050.9
Г 60

Reviewers:

Burkov V.N., Dr.Sc., Professor,
Moscow Institute of Physics and Technology (Moscow, Russia)

Barkalov S.A., Dr.Sc., Professor,
Voronezh State Architecturally-Building University (Voronezh, Russia)

Golenko-Ginzburg D. Stochastic network models in innovative projecting. – Voronezh: Science Book Publishing House, 2011. – 356 p.

Г 60 Голенко-Гинзбург Д. Стохастические сетевые модели в инновационном проектировании: Монография (на англ. языке). – Воронеж: «Научная книга», 2011. – 356 с.

ISBN 978-5-98222-749-2

The book presents a unification of the most essential models to monitor stochastic network projects of innovative nature. The book comprises various on-line control models for different kinds of projects with both fixed and stochastic structures and constrained project scheduling models with various resource delivery schedules. The backbone of the monograph centers on different kinds of alternative stochastic network models. The book is widely illustrated with examples.

The monograph is intended for researchers in innovation-oriented design offices and companies, academic institutions as well as for graduate scholars specializing in "Project Management", "Industrial Engineering" and "Operations Research".

УДК 004.5
ББК 65.050.9
Г 60

ISBN 978-5-98222-749-2

CONTENTS

Editorial

In the middle of the 20th century a group of gifted scientists including D.Malcolm, L. Roseboom, C. Clark and W. Fazar, suggested a new method of project management based on network analyzing. This method which acquired world-wide popularity under the name PERT provided excellent application results in managing complex space- and defense-related projects “Apollo” and “Pollarius”. In the 60-s and 70-s, many scientists from all over the world did their best to further develop theoretical and application aspects of the PERT methodology.

Yet, as soon as in the early 80-s the promising PERT legacy began transmitting distress signals. The main problem boiled down to the growing gap between the latest theoretical PERT achievements, on one hand, and the rather poor level of practical field implementations based on insufficient and improper assumptions, on the other. The latter almost reduced the universal PERT method to merely managing models of deterministic type only. All other models of managing complex R&D projects not only including random stochastic elements and restrictions but also displaying stochastic structure being subject to random influences and disturbances, either disappeared from the scientists’ working table or became subject to fierce and often justified practitioners’ criticism. As a matter of fact, from the very first days of PERT implementations a number of vigilant and emphatic scientists warned [13,101,128,142,152] the broad international PERT community about difficulties and even principal failures originating from infertile attempts to implement complicated probabilistic models on the basis of primitive assumptions reducing their profound stochastic nature to a cheap determinate palliative. When the latter scenario was chosen, any kind of monitoring the project’s model became virtually impossible; as a result, the pregiven due date was most commonly unmet.

Following the regarded misfortunate developments, the PERT method through recent years ceased to meet its primary objective, namely, providing a powerful on-line control method for complex stochastic network projects; instead, it downgraded to a kind of advisory information system with control actions linked to managerial decisions only. The Russian delegation to the 22nd World Symposium of the International Project Management Association (Rome, 2008) mentioned that every complex system being denied its scientific basement ceases to be creational and vibrant. Nowadays, the Russian Federation decided to create within its borders a science-oriented future-city Skolkovo aimed at developing pioneering high-tech projects of immense complexity and importance. That is why we may be nothing but proud to accommodate the Publishing House to issue the new scientific monograph by Golenko-Ginzburg Dimitri on monitoring complex and often unique stochastic network projects. The author made an ambitious and at the same time successful attempt to demonstrate that it is his theory and that of his scientific school developed and cherished through the last

50 years first in the USSR and later on in Israel with fruitful links to colleagues and followers in the nowadays Russian Federation, that suits mostly the novel innovative projects developed at Skolkovo, which require modern probabilistic models of multi-choice and stochastically-driven nature.

Referring to the monograph's contents, it can be well-recognized that practically all top questions related to:

- planning stage modeling;
- on-line control models;
- stochastic network project scheduling under chance constraints;
- hierarchical control models for several stochastic network projects,

- have been explicitly and thoroughly outlined. All types of presented models are applicable to both types of stochastic R&D projects, namely:

a) projects with fixed structure and random activities' durations;

b) projects with alternative structure and stochastic multi-variant outcomes.

Thus, a conclusion can be drawn that the monograph represents a useful research which is published in due time and within the right scientific community which is by all means mature to make the utmost benefits from its legacy.

The monograph can be used as tutorial for graduate scholars specializing at "Project Management", "Industrial Engineering", "Operations Research", as well as in Academic Institutions and Design Offices.

Scientific Editor
Vladimir Voropaev - Professor, SOVNET President,
Academician of the Russian Academy of Natural Sciences

Preface

It is important for me to share with my readership the reasons which brought me to the decision to write this book. Circa five decades ago a scientific group in the former USSR under my supervision started undertaking research in the area of managing stochastic network projects. In the 60's and 70's I was responsible for R&D stochastic network projecting linked to the Ministry of Aviation. After my immigration to Israel in 1985 the group proceeded with the research, being ultimately joined by gifted Israeli scholars. Today, thanks to my ongoing participation in the major world conferences held by the International Project Management Association (IPMA), I am well-informed about the current state of things in my research area. The resulting picture is neither one-sided nor simple.

It can be well-recognized that the initial period of excitement caused by the effectiveness of the primary network models applied to world-renowned R&D projects "Pollarius" and "Apollo" (1960-1970), gave way to a phase of sobering not to say disappointment. At that period it became clear enough to anybody involved that the variety of existing projects can be subdivided into two different types. The first one is characterized by a very low level of indeterminacy, more-or-less simple graph structure with a well-known, standard project's goal. This type of projects comprises, e.g., construction enterprises aimed at providing standardized living houses in populated areas, where all activities entering the network projects, have practically deterministic durations. In order to monitor such projects one has merely to substitute the duration of a certain activity (when necessary) by its mean value. Thus, the network project becomes in fact deterministic, and can be easily managed. There is no need in on-line control, and the project manager is fully satisfied by receiving periodically advisory information.

The second type of projects is characterized by high indeterminacy and is usually aimed at creating new unique high-technology products, which have no prototypes in the past. The project's activities' durations are random values with a large variance range. Projects may comprise branching nodes of random or deterministic types and milestones of deterministic type (decision nodes). Monitoring such projects cannot be facilitated by means other than on-line control models. Representative examples of this type can be found among R&D projects, especially those linked to innovative technologies.

Over the years run, the fate of both projects' types was different. Non-complicated projects nowadays are lucky to benefit from the world-wide support of above 350 software packages available on market, with the annual sales revenue of over 25 billion dollars [158]. On the contrary, complicated R&D projects have very much of a feel of being left behind.

From the beginning, in the early 80's, an attempt was made to manage unique complicated projects by merely the same techniques as those which did so well for the case of simple deterministic projects. This attempt, however, proved very soon to become a major failure. Within more than a decade this shortcoming has been the subject of a prolonged professional debate involving also sharp and sometimes emotional criticism [101,128,143, etc.].

In our opinion, the main reason for the nowadays situation when complicated projects are all usually completed late and remain, in practice, uncontrolled, boils down to the very fact that they are carried out under random disturbances (new estimates of random nature without any prior experience, random activities' durations, periodical revisions of networks over time due to random emergency situations, etc.). However, project managers usually [128] avoid probabilistic terms since they are not sufficiently trained. They are trying to control highly complicated projects with uncertainty by using deterministic techniques. This leads to biased estimates that underestimate the actual time required to accomplish the project. Therefore the targeted project's due date can rarely be met.

Since I am undertaking research mainly for that (second) type of projects, I was often asked about the reasons of such inconsistencies. The question becomes even more challenging in view of the well-known fact that many of our Japanese colleagues demonstrate over years convincing success of numerous realized innovative projects with a high level of indeterminacy [141,148]. That is why I carefully examined the situation to compare results accumulated by our scientific group with those stemming from Japanese conceptions.

As a result of this cross-over examination, it became evident that as far as our scientific group is concerned, our main research philosophy is not only non-contradictive but even close to the basic Japanese conceptions of planning and controlling with uncertainty. Moreover, they supplement each other.

What is the essence of the Japanese philosophy when controlling a system with uncertainty and being at the outset of something which is basically indeterminate? Many examples from high performance practice in Japan show that under such circumstances the control system should not work to a predetermined plan, but should be inherently adaptable, seeking at each decision node to assess the *best route forward*, reconfiguring if appropriate the ultimate goals.

Note that the subproblem of determining the *best route* may be very difficult and complicated, especially for systems with a high level of indeterminacy. Solving this subproblem usually results in solving the general control problem.

Further, what is our philosophy in project planning and control with indeterminacy? We are not predetermining the initial network model; moreover, in certain cases the structure of such a model may be indeterminate. At the initial stage of the project's realization, the network may be restricted to a source node and several alternative sink nodes (goals) together with some milestones (a decision-tree model). Various activities are usually of random duration. Such a stochastic alternative network is renewed permanently over time, including changes in the ultimate goals. At each decision node our techniques enable us to choose the optimal outcome. Decision making is repeatedly introduced for the renewed network at every sequentially reached decision node.

Thus, the modern project manager should not fear indeterminacy but on the contrary, has to treat the latter the way the Japanese do, i.e., as a friend and assistant, and avoid excluding indeterminacy from the international Project Management community like the devil being banished from church.

In 1996 I was appointed key speaker of the NATO workshop "Managing and Modeling Complex Projects" [68]. From the broad spectrum of various planning and control problems for stochastic network projects, within several days of discussion it was decided to choose and recommend for practical usage four milestones, namely

1. Alternative network models.
2. On-line control models.
3. Stochastic network project scheduling.
4. Multilevel control models for several stochastic network projects.

Since that forum 15 years have elapsed but little was done if at all to resolve the above mentioned stochastic project management contradiction. In certain senses, the situation even became more critical [158]. As a matter of fact, former PERT creators [117] have been brilliant scientists, both in mathematics, industrial engineering and management. Nowadays, their majority are not with us any longer. New project managers have certain experience in managing industrial enterprises but nothing more than that! Most of them are not trained either in cybernetics (including the probabilistic area) or in industrial engineering. Some of them prefer undertaking voluntaristic decisions which are not based on any theoretical grounds. This leads to an extremely dangerous situation when science is emasculated *de-facto* from PM.

It can be well-recognized that in the last several years a variety of countries, especially those entering the BRIC group (Brasilia, Russia, India, and China), exercise a great effort to boost and modernize their industries. This, in turn, causes for the necessity to carry out ambitious innovative projects, the majority of them belonging to the regarded class of complicated stochastic network projects. Taking into account that since 1977, when the excellent monograph by S. Elmaghraby [40] has been published, not a single book on managing stochastic network projects has been presented to the readers, it becomes clear why I decide to write this book. The general idea is to summarize all the results developed by our scientific group within five decades in order to help the innovative projects companies to carry out their projects on the basis of scientifically grounded planning, control and scheduling techniques.

This is not a text-book but a monograph. The difference between the two causes me to refrain from rewriting anew classical theoretical grounds developed and presented so well like [40]; instead, I use to quote appropriate references.

This monograph refers not only to R&D projects but to all other complicated projects under random disturbances, which are *innovative in nature*. For example, the venture of constructing the Trans-Siberian pipe-line from the Arctic coast to China cannot be catalogued as an R&D enterprise; yet, it definitely involves a great amount of sophisticated models of alternative type with branching outcomes. Thus, to carry out such a project successfully the manager really has to be experienced in stochastic network control and participate in a great amount of "brain-storming". Another example may be associated with developing a multi-well major oil/gas field with variable well capacities, to minimize the project's total expenses as well as consecutive exploitation costs, etc.

The structure of the book is as follows. We have subdivided the monograph into five main parts. The first part "General Concepts of Stochastic Network Projects" comprises the first three chapters. In Chapter 1 a brief characteristic of the most essential models to monitor stochastic network projects, is presented. In Chapters 2-3 a justification of determining the main parameters of stochastic network projects by means of analytical and simulation methods, as well as their usage in planning, controlling and scheduling, is outlined.

The second part "On-Line Control Models for Stochastic Network Projects" considers the mostly used on-line control models, namely:

- models based on sequential statistical analysis (Chapter 4),
- models based on risk averse decision-making (Chapter 5), and
- models based on the chance constraint principle (Chapter 6).

The third part "Alternative Stochastic Network Models" is the core of the book and comprises four chapters. Chapter 7 presents the general description of an alternative stochastic model, Chapter 8 - the fully divisible controlled alternative activity network (CAAN), Chapter 9 - the non-divisible controlled GAAN model. Chapter 10 outlines a two-parametrical optimization algorithm for the CAAN model.

The fourth part "Resource Constrained Project Scheduling for Stochastic Network Projects" comprises three chapters. Chapter 11 considers various resource supportability models without predetermined resource delivery schedules in advance, while Chapter 12 presents deterministic resource delivery schedules, i.e., before the project actually starts. In Chapter 13 various resource supportability models of mixed type, which can be used both on planning and control stages, are outlined.

The fifth part "Hierarchical Models for Planning and Controlling Several Stochastic Network Projects" comprises four last chapters. Chapters 14-15 present a hierarchical on-line control model for PERT-COST projects. In Chapter 14 the planning stage models are outlined, while Chapter 15 presents local on-line control models together with a unified three-level hierarchical model including planning, on-line control and scheduling stages. In Chapter 16 two hierarchical decision-making models for a CAAN type alternative model are presented. Both cases of cost resources (a hierarchical PERT-COST model) and renewable resources are considered. Thus, Chapters 14-16 cover our basic results in creating hierarchical on-line control models as well as hierarchical support models. In Chapter 17 novel harmonization models to estimate the stochastic network projects' utility, are outlined. Here the concept of utility signifies the quality of the project's functioning.

In conclusion, I would like to thank my gifted pupils N. Archangelski, D.Blokh, A. Gonik, V. Kuzmin, S. Livshitz, A. Malisheva and Sh. Sitniakovski who helped me in preparing the book's material.

I am deeply obliged to Prof. Vladimir Voropaev for his valuable editorial assistance as well as to Dr. Avner Ben-Yair for his excellent secretarial duties.

I am privileged to thank my colleagues Profs. Vladimir Burkov and Sergey Barkalov for their great help and support in the course of compiling the monograph.

PART I

GENERAL CONCEPTS OF STOCHASTIC NETWORK PROJECTS

Chapter 1. A Survey of Planning, Controlling and Scheduling Models in R&D Innovation Projects Under Random Disturbances

§1.1 Alternative stochastic network projects

1.1.1 Basic stochastic network models for innovative projects

From the broad spectrum of various planning, control and scheduling models for stochastic network R&D projects the following ones can be considered [68]:

1. *Alternative network projects* under random disturbances with various alternative outcomes in key nodes. The control model chooses the optimal outcome direction at every decision node that is reached in the course of the project's realization.

2. *On-line control models* for network projects, for which the project's progress can be evaluated only by means of inspection in control points. The project's due date and the chance constraint to meet the deadline are pregiven. An on-line control model determines both the control points and the control actions to be introduced at those points to reorient the progress of the project in the desired direction.

3. *Stochastic network project scheduling* with several non-consumable activity related limited resources. Each activity is operated at a random speed that depends on the resource capacities assigned to that activity. The model determines, for each activity entering the project, both starting time values and corresponding resource capacities. The model's objective is to minimize the expected project duration.

4. *A multilevel control model for several stochastic network projects* which unifies the models outlined above.

1.1.2 Alternative network projects under random disturbances

While the literature on PERT and CPM network techniques is quite vast, the number of publications on alternative networks remains very scanty. Various authors, e.g., Eisner [37], Elmaghraby [38-40], Pritsker [131-133], Whitehouse [161], etc., introduced the concept of a Research and Development (R&D) project as a complex of problems and actions towards achieving a definite goal. Several adequate network models for such projects have been considered. The first significant development in that area was the pioneering work of Eisner [37] in which a "decision box" with both random and deterministic alternative outcomes was introduced. Elmaghraby [38] introduced additional logic and algebra in network techniques, while Pritsker, Happ and Whitehouse [131-133,161] developed the GERT techniques for alternative network models with stochastic outcomes in key nodes. Xespos and Strassman [166] introduced the concept of the stochastic decision tree, while Crowston and Thompson [28-30] and later on Hasting and Mello [99] suggested the concept of multiple choices at such alternative nodes, when decision-making is of deterministic nature (Decision CPM models). Lee, Moeller and Digman [111,123] developed the VERT model that enables the analyst to simulate various decisions with alternative technology choices within the stochastic decision tree network. Golenko-Ginzburg [53-57] has developed a unified controlled alternative activity network (CAAN model) for projects with both random and deterministic alternative outcomes in key nodes. At each routine decision-making node, the developed algorithm singles out all the subnetworks (the so-called joint variants) that correspond to all possible outcomes from that node.

Decision-making results in determining the optimal joint variant and following the optimal direction up to the next decision-making node. However, the techniques thus far developed can only be applied to fully-divisible networks that can be subdivided into non-intersecting fragments. The CAAN model does not include non-fully-divisible networks. Thus, the model is not relevant to most R&D projects, since the latter are usually structured from non-divisible subnetworks. Golenko-Ginzburg and Blokh [67] have developed a more universal alternative network - the Generalized Alternative Activity Network (GAAN model). All types of the previously developed alternative network models, namely, Eisner's model, GERT, Decision-CPM, VERT and CAAN networks, are particular cases of the GAAN model.

1.1.3 <u>The GAAN model</u>

Let's take a brief overview of the GAAN model. A detailed description of the latter will be presented later on, in Chapter 9.

A GAAN model is a finite, oriented, acyclic activity-on-arc network $G(N,A)$ with the following properties:

I. $G(N,A)$ has one source node n_0 and no less than two sink nodes n'.

II. Each activity $(i,j)\in A$ refers to one of the following three different types:

<u>Type 1</u>: activity (i,j) is a PERT activity (PA) with the logical "must follow" emitter in node i and the "and" receiver in node j;

<u>Type 2</u>: activity (i,j) is an alternative stochastic activity (ASA) with the logical "exclusive or" emitter in node i. Each $(i,j)\in A$ of ASA type corresponds to a probability $0\le p_{ij}\le 1$, while node i comprises a set of at least two probabilities p_{ij}, $\sum_j p_{ij}=1$;

<u>Type 3</u>: activity (i,j) is an alternative deterministic activity (ADA) with the logical "exclusive or" emitter in node i. Node i is a decision-

making node, and the sum of the corresponding transfer probabilities (at least two of them) is assumed to be unity.

III. Activities of all types may come out of the same node $i \in B$. Thus, unlike the CAAN model, the GAAN model is not a fully-divisible network.

IV. Activities of all types may enter one and the same node.

A joint variant of the GAAN model $G(N, A)$ is a subgraph (subnetwork) $G^*(N^*, A^*)$ satisfying the following conditions:

1. $G^*(N^*, A^*)$ has one source node coincident with that of graph $G(N, A)$.
2. If $G^*(N^*, A^*)$ comprises a certain node i, i.e., $i \in N^*$, then $G^*(N^*, A^*)$ comprises all activities (i, j) of types PA and ASA leaving node i.
3. If $G^*(N^*, A^*)$ comprises a certain node i having alternative outcomes of ADA type in the GAAN model $G(N, A)$, then $G^*(N^*, A^*)$ comprises only one activity of this type leaving that node.
4. $G^*(N^*, A^*)$ is the maximal subgraph satisfying conditions 1-3.

Call *a full variant of joint variant* $G^*(N^*, A^*)$ a subnetwork of PERT type $G^{**}(N^{**}, A^{**}) \subset G^*(N^*, A^*)$ which can be extracted from the latter by simulating non-contradictory outcomes of ASA type in interconnected nodes and excluding alternative non-simulated outcomes.

Call *the probability of realizing a full variant* G^{**} the product of all values p_{ij} for all activities of ASA type entering the full variant.

1.1.4 <u>Decision making in CAAN and GAAN -type models</u>

To control a project, such as any production process, it is necessary to introduce decision-making in order to reach the goal while optimizing a given objective (the optimized value OV) subject to certain restrictions (the restrictive values RV). When the objective is the project's duration, the primary restriction is usually the project's cost, and *vice versa*. For a project represented by a GAAN type model decision-making boils down to choosing the directions of the project's progress in controlled nodes (decision-making nodes) with alternative outcomes of ADA type, since alternatives of ASA type are uncontrollable. Thus the optimization problem consists of the following steps:

<u>Step 1</u>. At each decision-making node which has been reached at moment t in the course of the project's realization,

- to determine and to single out all the joint variants from the remaining project G_t at moment t;
- to calculate the optimized value OV and all the restrictive values RV for each variant.

<u>Step 2</u>. To determine the optimal joint variant and to follow the optimal direction up to the nearest decision-making node. The problem should be repeatedly solved for the reduced network in every sequentially encountered decision-making node.

1.1.5 Mathematical formulation

The mathematical formulation of the optimization problem is as follows [57,67]: determine the optimal joint variant $G^{*opt} \subset G(N,A)$ that optimizes the objective function

$$E\left[F\left(G^{*opt}\right)\right] = \underset{\{G^*\}}{Min}(Max) \sum_{\{G^{**}\}\subset G^*} \left[F\left(G^{**}\right)\cdot \Pr\left\{G^{**}\right\}\right] \tag{1.1.1}$$

subject to

$$E\left[Q_v\left(G^{*opt}\right)\right] = \sum_{\{G^{**}\}\subset G^*} \left[Q_v\left(G^{**}\right)\cdot \Pr\left\{G^{**}\right\}\right] \le H_v, \quad 1 \le v \le V. \tag{1.1.2}$$

Here, $F\left(G^{**}\right)$ is the objective function of full variant G^{**}, $\Pr\left\{G^{**}\right\}$ is the probability of realizing G^{**}, $Q_v\left(G^{**}\right)$ is the v-th constraint criterion, and H_v is the preset restriction level for that criterion. Note that for certain particular cases, the value of V may be zero, i.e., the optimization problem is unconstrained, or the problem comprises only one constraint (1.1.2) without objective function (1.1.1).

Since problem (*1.1.1-1.1.2*) *is NP-complete* [10,67]*, in order to obtain the optimal solution one has to develop a lookover algorithm to single out all the joint variants.*

The idea to enumerate the joint variants of the CAAN model [57] is based on introducing lexicographical order to the set of maximal paths in the CAAN graph. The corresponding lookover algorithm is very simple in usage [57,68]. In the case of a GAAN network the order on the set of paths has to be substituted for the order on the set of subgraphs [67]. To develop the enumeration algorithm, we use the ideas to enumerate the so-called trajectories for assignment problems, or special matrices for traveling salesman problems [9,67]. Note that singling out the maximal trajectory for an assignment problem is similar to determining the joint variant with the maximal objective value. Since a trajectory can be regarded as a vector and the latter, in turn, can be mapped onto a set of integer numbers, the trajectories can be enumerated. Similar ideas are used in developing a lookover algorithm to enumerate and single out all the joint variants [67].

If the number of joint variants becomes very high, the developed lookover algorithms (both for the CAAN and GAAN models) require much computational time, especially for networks with many alternatives. Golenko-Ginzburg, Blokh and Gutin suggested an approximate method which is based on the ideas of combinatorial optimization with two parameters [75,82]. Unfortunately, the developed method suits only the CAAN type network models.

Note, in conclusion, that alternative stochastic models (ASM) may be costly and complicated in usage. But for modern and complex innovation projects the gain from implementing such models may be tremendous.

§1.2 On-line control models for network projects

After determining the optimal joint variant (in the case of an alternative network project) the latter is realized and controlled in order to meet the project's due date on time. A joint variant may be either a PERT, or a GERT type network, usually with random activity durations. For most R&D projects the progress of the project cannot be inspected and measured continuously, but only at preset inspection points. An on-line control has to determine both inspection points and control actions to be implemented at those points to alter the progress of the project in the desired direction. On-line control is usually carried out to minimize the number of inspection points needed to meet the target, since inspecting the project's output is usually a costly operation. In addition, on-line control for a stochastic network project has to be carried out subject to a chance constraint. Thus, the generalized on-line control model has to be formulated as follows [64,66,68]: determine both optimal control points t_g to inspect the project and optimal control actions $CA(t_g, r_g)$ to be implemented at those control points (r_g being the index of the control action), in order to minimize the number W of inspection points

$$\underset{\{t_g, r_g\}}{Min} W \tag{1.2.1}$$

subject to

$$\Pr\{t_g, r_g\} \geq p^*, \tag{1.2.2}$$

$$t_0 = 0, \tag{1.2.3}$$

$$t_W = D, \tag{1.2.4}$$

$$t_{g+1} - t_g \geq \Delta. \tag{1.2.5}$$

Note that if implementing a control action $CA(t_g, r_g)$ results in determining the *project's speed* v_{t_g} to proceed with until the next control point t_{g+1} and if *several alternative speeds can be chosen*, then the optimal control action enables adopting the *minimal speed* while honoring chance constraint (1.2.2) [66,68].

It can be well-recognized that control model (1.2.1-1.2.5) is in fact a stochastic optimization problem with a non-linear chance constraint and a random number of optimized variables. Such a problem is too difficult to solve in the general case. Thus, heuristic control algorithms have been developed [64, 66, 68, 72] to determine the next inspection point t_{g+1}. Three algorithms are considered:

A. Using sequential statistical analysis to maximize the time span $\Delta t_g = t_{g+1} - t_g$.

B. Using the methodology of a risk-averse decision-maker.

C. Using the methodology of the chance constraint principle.

Algorithm A [66,68] solves the on-line control problem as follows: to maximize the objective $(t_{g+1} - t_g)$ subject to (1.2.3-1.2.5) and

$$\Pr\{V_t \geq V_t^*(t_g)\} \geq p^*, \quad \forall t : t_g \leq t \leq t_{g+1}. \tag{1.2.6}$$

This problem can be solved by determining the maximal value T^* satisfying

$$T^* = \underset{t_g < t \le D}{Max}\left\{ t : \psi\left(q_t\right) \ge p^* \right\}. \qquad (1.2.7)$$

Here

$$\psi(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{u^2}{2}} du, \quad q_t = \frac{\overline{H}_t}{S^2(H_t)}, \quad H_t = V_t - V_t^*(t_g), \qquad (1.2.8)$$

while $\overline{H}_t$ and $S^2(H_t)$ designate the average and variance of random value H_t, correspondingly. In practice, T^* can be calculated by means of simulation with a constant step of length Δ. The procedure of increasing t step-by-step is followed until (1.2.7) ceases to hold. The thus determined value T^* satisfies $t_g + T^* = t_{g+1}$.

Algorithm B is based on the concept of risk-averse decision-making [68,72]. Given a routine inspection point t_g, the project's output observed at that moment V_{t_g} and the control action $CA(t_g, r_g)$ to be implemented at moment t_g up to the next inspection point, the problem is to determine that next point t_{g+1}. As for Algorithm A, the objective is to maximize the time span $(t_{g+1} - t_g)$. Value t_{g+1} is determined so that even *if the project develops most unfavorably in the interval* $\left[t_g, t_{g+1}\right]$*, i.e., with the minimal rate* $v'(t_g, r_g)$*, then introducing the most effective control action* $CA(t_{g+1}, r)$ *at moment* t_{g+1} *enables the project to meet its target on time, subject to the chance constraints*. Here r is the index of the most effective control action, e.g., r is the index of the highest possible speed to be introduced. Value is determined via "risk-averse" heuristics

$$V_{t_g} + v'(t_g, r_g)(t_{g+1} - t_g) + \bar{v}(t_{g+1}, r)(D - t_{g+1}) = V^*. \qquad (1.2.9)$$

Note that the minimal rate $v'(t_g, r_g)$ can be substituted for a p-quantile of the random speed $v(t_g, r_g)$ when the confidence level p is close to zero.

Both on-line control algorithms are implemented in real time. However, on order to check the validity of any of them, the algorithms' functioning can be simulated. The comparative efficiency of Algorithms A and B has been tested on various examples of medium-size PERT projects. A general conclusion can be drawn [68] that applying the second algorithm rather than the first results both in essentially smaller computational time and in cheaper project realization. Both methods honor the chance constraint p^* and can be implemented for various control models for projects of PERT type.

Note that the above outlined on-line control models can also be applied to control projects of GERT type, i.e., to network projects which comprise various random alternative outcomes. For such projects, a certain part of the activities will not be carried out in the course of the project's realization. Golenko-Ginzburg et al [65] recommend splitting the remaining project into two subgraphs at each decision-making node. The first would be a PERT graph that is realized before meeting the nearest random alternative node, while the second subgraph is a GERT network. After the next random alternative node is reached and the random outcome is simulated, the procedure of subdividing the remaining project is carried out anew.

However, both models A and B do not support solving cost-optimization problems. This shortcoming called for the creation of the on-line control model C which is a cost-optimization model and based on the so-called chance constraint principle [73,83-84].

Given the average processing costs per time unit for each activity to be operated under each speed, together with the average cost of performing a single inspection at the chosen control point, the problem at a routine control point t_g is to determine the proper speed $v^{(k)}$ and the next control point t_{g+1}, in order to minimize the total processing costs within the planning horizon, subject to a chance constraint. At each control point, decision-making centers around the assumption that there is no more than one additional control point before the due date. Following that assumption, two speeds $v^{(k_1)}$ and $v^{(k_2)}$ have to be chosen at a routine control point t_g:

1. Speed $v^{(k_1)}$ which has to be actually introduced at point t_g up to the next control point t_{g+1};
2. Speed $v^{(k_2)}$ which is forecast to be implemented at control point t_{g+1} up to the due date D.

The couple $\left(v^{(k_1)}, v^{(k_2)}\right)$ providing the minimal total cost expenses, has to be accepted.

The model is particularly effective when each activity can be measured as a partial accomplishment of the entire planned program.

We suggest applying control model C for small- and medium-size projects. In cases of large projects, we suggest aggregating the initial model order to transfer the latter to an equivalent one, but of medium- or small-size. After observing the project's output at a routine control point and introducing proper control actions, i.e., determining the new processing speed and the next control point, the aggregated network is transformed back to the initial one, and the project's realization proceeds.

§1.3 Stochastic network project scheduling with non-consumable limited resources

Golenko-Ginzburg, Gonik and Sitniakovski have developed a variety of algorithms on resource constrained project scheduling under random disturbances and with limited resources [68-71,74,78-80,90,93-94].

An activity-on-arc network project of PERT or GERT type with random activity durations is considered. Several non-consumable activity related resources, such as machines or manpower, are utilized to carry out the project. Each activity (i,j) in such a project requires resources of various types k with variable capacities r_{ijk}. Each resource capacity r_{ijk} assigned to any activity is limited within pregiven bounds $r_{ijk}^{\min}$ and $r_{ijk}^{\max}$. Each type of resource k is in limited supply with a resource limit R_k that is fixed at the same level throughout the project's duration. It is assumed that each activity is operated at a random speed that depends on the resource capacities assigned to that activity.

The problem is to determine, for each activity (i,j) entering the project, both the starting time values S_{ij}, i.e., the timing of feeding-in resources, and the resource capacities r_{ijk} for each type of resource k assigned to that activity. The problem's goal is to minimize the expected project duration [71].

The problem's mathematical formulation is as follows:

$$\underset{S_{ij}, r_{ijk}}{Min}\, E\left\{T\left(G/S_{ij}, \vec{r}_{ijk}\right)\right\} \tag{1.3.1}$$

subject to

$$r_{ijk}^{\min} \le r_{ijk} \le r_{ijk}^{\max} \quad \forall (i,j) \in G(N,A), \tag{1.3.2}$$

$$R_k^*\left(t/S_{ij}, \vec{r}_{ijk}\right) \le R_k(t) \ \ \forall t \ge 0, \ \ 1 \le k \le n. \tag{1.3.3}$$

Model (1.3.1-1.3.3) is a stochastic optimization problem that cannot be solved analytically in the general case; the problem allows only a heuristic solution. Decision-making, i.e., determining values S_{ij} and r_{ijk}, is facilitated at decision points F_{ij} and T_i, either when one of the activities (i,j) is finished and additional resources become available, or when all activities (i,j) leaving node i are ready to be processed. Thus, both values S_{ij} and r_{ijk} are *not calculated beforehand* and are random variables dependent on our future decisions. If one or more activities $(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)$, $m \ge 1$, are ready to be processed at a routine decision point t and *all of them can be supplied by all types of available resources of maximal capacity*, the needed resources are fed in and activities (i_q, j_q), $1 \le q \le m$, start to be operated at moment t, i.e., $S_{i_q j_q} = t$, $r_{i_q j_q k} = r_{i_q j_q k}^{\max}$, $1 \le k \le n$. Otherwise, a competition has to be arranged to choose the optimal subset of activities that can be supplied by available resources.

An important auxiliary procedure precedes holding the competition, namely, calculating, for all the competitive activities (i_q, j_q), their conditional probabilities $p(i_q, j_q)$ to be on the critical path in the course of the project's realization. Calculating values $p(i_q, j_q)$ is carried out via simulation: at each decision point, all the activities that have not yet started to be operated are simulated using the corresponding probability density functions.

Two cases are considered [68]:

a) all resource capacities r_{ijk} for each k-th type of resources are fixed and remain unchanged;

b) values r_{ijk} may vary within pregiven bounds $r_{ijk}^{\min}$ and $r_{ijk}^{\max}$.

In the first case random values t_{ij} do not depend on values r_{ijk}, and the corresponding density functions remain unchanged in the course of the project's simulation. Later on, the critical path of the remaining graph with simulated activity durations is determined. By repeating this procedure many times, the calculated frequencies for each activity (i_q, j_q) to be on the critical path are taken as $p(i_q, j_q)$. Values $p(i_q, j_q)$ enter the zero-one integer programming model to carry out the competition [70].

For the case of variable resources r_{ijk} resource capacity values $r_{i_q j_q k}$ that will be assigned to the activities under competition are unknown beforehand; the same goes for all other activities in the remaining project. Thus, we are unable to simulate the activities' durations, that depend parametrically on values $r_{i_q j_q k}$. To overcome these difficulties, the authors in [71] use heuristics, e.g., by assuming $r_{ijk} = 0.5 \cdot \left(r_{ijk}^{\min} + r_{ijk}^{\max}\right)$. After calculating conditional probabilities $p(i, j)$ the knapsack reallocation problem among the competitive activities has to be solved at each decision point [71].

For the case of fixed r_{ijk} a classical zero-one programming problem with a precise solution can be formulated as follows: determine integer values $\xi_{i_q j_q}$, $1 \le q \le m$, to maximize the objective

$$\underset{\{\xi_{i_q j_q}\}}{Max}\left\{\sum_{q=1}^{m}\left[\xi_{i_q j_q} \cdot p(i_q, j_q) \cdot \mu_{i_q j_q}\right]\right\} \tag{1.3.4}$$

subject to

$$\sum_{q=1}^{m}\left(\xi_{i_q j_q} \cdot r_{i_q j_q k}\right) \le R_k(t), \quad 1 \le k \le n, \tag{1.3.5}$$

where

$$\xi_{i_q j_q} = \begin{cases} 0 & \textit{if activity } (i_q, j_q) \textit{ will not obtain resources}; \\ 1 & \textit{otherwise}. \end{cases}$$

In the case of variable resource capacities r_{ijk} a heuristic model is suggested [71]. Since the project management has to choose the subset of activities and to reallocate among them the available resources *in order to maximize the total contribution to the expected project duration*, the following resource reallocation problem (to be solved at each decision point t) is suggested:

Determine optimal values $S_{i_q j_q}$ and $r_{i_q j_q k}$, $1 \le k \le n$, $1 \le q \le m$, to maximize the objective

$$J = \underset{S_{i_q j_q}, r_{i_q j_q k}}{Max}\left\{\sum_{q=1}^{m}\xi_{i_q j_q} \cdot p(i_q, j_q) \cdot \sum_{k=1}^{n}\left(r_{i_q j_q k} \cdot \psi_{i_q j_q k}\right)\right\} \tag{1.3.6}$$

subject to

$$r_{i_q j_q k}^{\min} \le r_{i_q j_q k} \le r_{i_q j_q k}^{\max} \quad \forall (i_q, j_q) \in G(N, A), \tag{1.3.7}$$

$$\sum_{q=1}^{m}\left(\xi_{i_q j_q} \cdot r_{i_q j_q k}\right) \le R_k(t) \quad \forall t \ge 0, \quad 1 \le k \le n, \tag{1.3.8}$$

where $\xi_{i_q j_q}$ is as before.

Problem (1.3.6-1.3.8) is NP-complete. Both a precise solution facilitated by means of a lookover algorithm, and a heuristic solution based on an essential diminishing of the set of feasible solutions to be examined, are obtained [71].

The resource constrained project scheduling algorithm comprises the knapsack problem (1.3.4-1.3.5) or (1.3.6-1.3.8), together with the auxiliary problem to determining conditional values $p(i,j)$. The algorithm to solve problem (1.3.1-1.3.3) is implemented in real time: namely, *all activities can be operated only after obtaining necessary resources*. Decision moments F_{ij} and T_i cannot be predetermined. However, should the question of evaluating the efficiency of the resource constrained project scheduling model (1.3.1-1.3.3) arise, the algorithm's functioning can be simulated by random sampling of the actual duration of activities. By simulating the algorithm many times, the average project's duration as well as the probability of accomplishing the project by a given due date (if necessary) can be estimated. Intensive experimentation [68] has been carried out for various medium-size PERT and GERT projects with several (3÷5) non-consumable limited resources. A conclusion can be drawn that the algorithm performs well and is easy to handle.

Further progress in the area of developing resource constrained project scheduling models has been achieved by the scientific school of Golenko-Ginzburg [69-74,77-90]. Several types of models have been developed. The first model considers a simplified case of several stochastic projects in the form of a chain of consecutive operations. Models of the second type consider several simultaneously realized stochastic network projects of PERT type. Resource scheduling models of the third type also cover PERT type projects, but with two different kinds of renewable resources:

a) extremely costly resources (A-resources) which have to be utilized for a short time within the project's time span. Such resources have to be prepared and delivered externally at planned moments;

b) renewable resources (B-resources) which are at the system's disposal.

In all types of models each project's activity utilizes several non-consumable related resources with fixed capacities, e.g., machines or manpower. Each type of resource at the management's disposal is in limited supply, with a resource limit that is fixed at the same level throughout the entire project's duration, i.e., until the last project is actually accomplished. For each operation, its duration is a random variable with given density function. Processing costs per time unit to hire and to utilize all the total available resources are pregiven.

The problem is to determine:

- the earliest starting moment for each project's realization;
- the limited resource levels for each type of resources to be stored during the projects' realization;
- the moment when resources are fed in and projects' activities start, -

in order to minimize the average total expenses of hiring and maintaining resources subject to the chance constraints.

For the third class of developed models the problem boils down:

a) to predetermine *in advance*, i.e., before each projects starts to be realized, the deterministic delivery schedule for A-resources which are not at the projects' disposal;

b) to determine both the starting times and the resource capacities to be utilized for activities which require limited renewable B-resources which are not at the projects' disposal;

c) to determine the starting moment S of each project's realization, -

in order to minimize the average total projects' expenses subject to the chance constraint.

The problem is solved by means of simulation, in combination with a cyclic coordinate descent method and a knapsack resource reallocation model. The simulation model comprises three optimization cycles and can be used for small- and medium-size projects only. Otherwise, aggregation has to be applied.

Our basic concept which has been fully supported by the NATO Forum "managing and Modeling Complex Projects" (Kiev, Ukraine, December 1996) is as follows:

a) Scheduling and control procedures must *not* be incorporated in one model.

b) A control model has to be based on probabilistic approaches and has to implement probabilistic terms. Such a model has to be used only at several control (inspection) points. We suggest applying the control model not to the initial network (which for some projects may comprise a large amount of activities), but to a modified one, with a medium amount of activities at the utmost. For such a modified model, an activity can be a subnetwork (a fragment) of the initial network.

c) Scheduling procedures are applied to the initial network and are carried out *between two adjacent routine control points*. They are usually based on heuristic procedures (sometimes very doubtful) and may result in biased estimates and errors. *But the latter are periodically corrected* by means of introducing proper control actions.

d) Thus, we recommend *developing the on-line control model as an additional tool, as a decision-making support model to assist the project manager carry out the project*. On the basis of such a model, the project manager may implement any action he finds reasonable, e.g., to enhance the progress of the project.

§1.4 Multilevel control model for several stochastic network projects with restricted resources

For many years the scientific school of Golenko-Ginzburg has undertaken extensive research in the area of hierarchical project management [7,53-54,64-65,68,92].

A company realizing several stochastic network projects $G_\ell(N,A)$, $1 \le \ell \le d$, is considered. The total budget C at the company's disposal to carry out the projects is limited. A hierarchical control model [68] as presented in Fig. 1.1 is suggested. At each level the model undertakes optimal control actions as follows:

- at the *company level* the control action boils down to optimal budget reassignment among the projects (Problem I);
- at the *project level*, in case of an alternative stochastic network project, its optimal joint variant is determined (Problem II);
- at the *project level*, if the project is of PERT or GERT type, optimal control actions result either in optimal budget reallocation among the project activities (Problem IIIA for PERT-COST projects) or in determining optimal speed of the project's realization (Problem IIIB); thus, solving Problems IIIA and IIIB results in optimizing the progress of the project towards its goal, in order to re-orient the project in the desired direction;
- at the *inspection level*, on-line control is carried out, i.e., optimal control points to inspect the progress of the project are determined (Problem IV);
- at the lowest level considered, namely, the *scheduling level*, resource constrained project scheduling is implemented by reallocating, if necessary, non-consumable resources among the project's activities (Problem V). Although Problem V is an optimization one, it cannot be regarded as a control action. This is because the problem's solution is not based on the project's output V_{t_g} which is observed at control point t_g.

Optimal budget reallocation (Problem I) can be formulated for two alternative cases:

a) projects are of equal importance;
b) projects have different priorities η_ℓ, $1 \le \ell \le d$.

In case of a), the optimization problem becomes as follows [7,54,92]:

At any moment $t \ge 0$ reassign budget among projects to optimize objective

$$J_1 = \underset{C_{\ell t}}{Max}\left\{\sum_{\ell=1}^{d}\left[\eta_\ell \cdot P_{\ell t}\left(C_{\ell t}\right)\right]\right\} \tag{1.4.1}$$

subject to

$$\sum_{\ell=1}^{d} C_{\ell t} \le C_t, \tag{1.4.2}$$

$$P_{\ell t}\left(C_{\ell t}\right) \ge p^* \quad \forall \ell : 1 \le \ell \le d. \tag{1.4.3}$$

In case of b), the problem is as follows:

$$J_2 = \underset{C_{\ell t}}{Max}\ \underset{\ell}{Min}\ P_{\ell t}\left(C_{\ell t}\right), \quad t \ge 0, \tag{1.4.4}$$

subject to (1.4.2-1.4.3). Both problems are solved at time $t = 0$ or have to be repeatedly resolved at $t > 0$, after an emergency is declared at the project level. Problems (1.4.1-1.4.3) and (1.4.2-1.4.4) are solved by means of simulation, under additional heuristic assumptions [64].

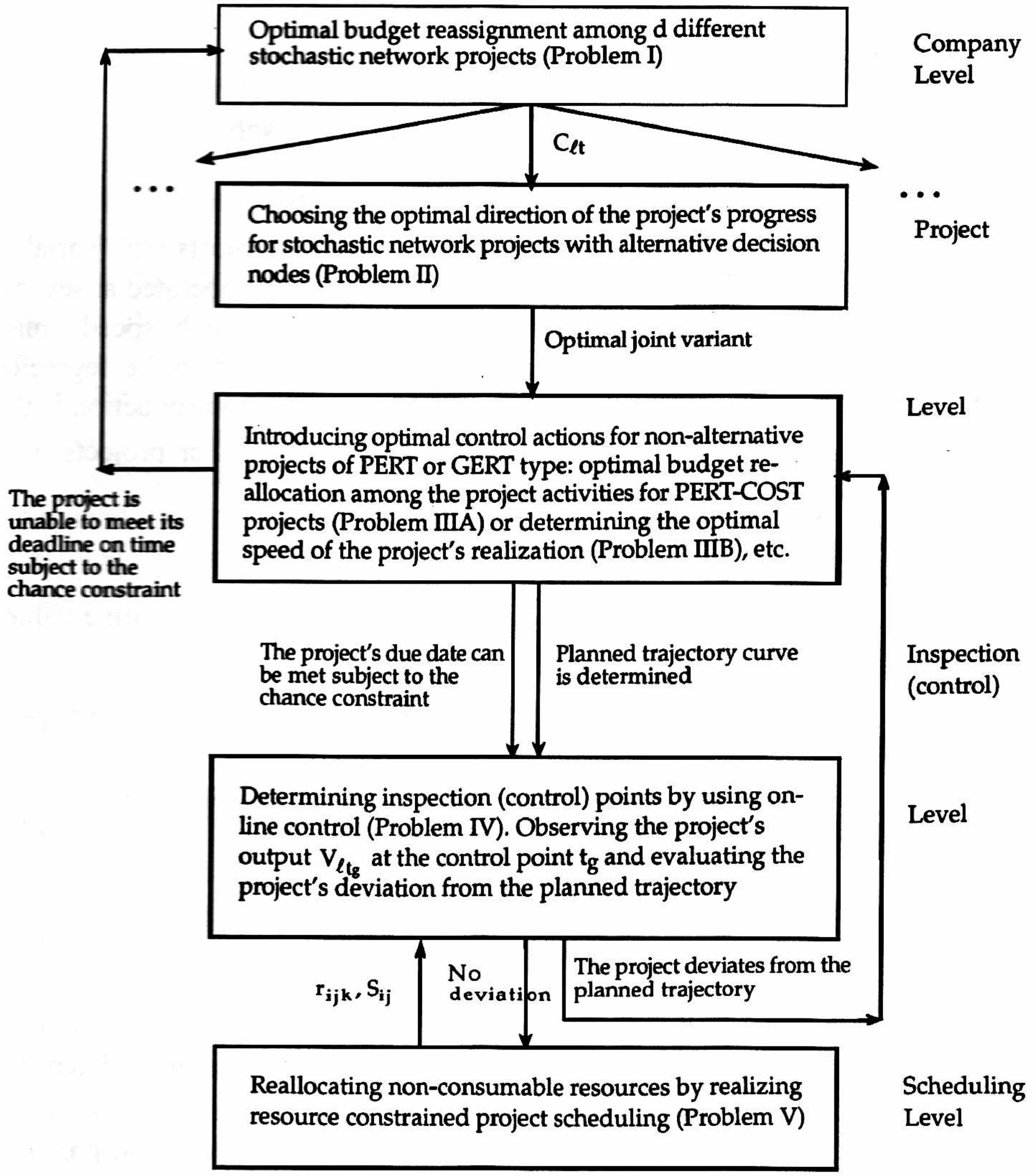

Figure 1.1. Multilevel control model (at moment t_g)

Problem I is outlined in Chapter 14, while Problem II is considered in Chapter 16. As to Problem IIIA, it can be formulated as follows:

Given budget $C_{\ell t}$ assigned to project $G_\ell(N,A)$ at moment t, determine optimal values $c(i,j)$, $(i,j)\in A$, to maximize objective

$$\underset{\{c(i,j)\}}{Max}\{P_{\ell t}(C_{\ell t})\} \tag{1.4.5}$$

subject to

$$c_{ij}^{\min} \le c_{ij} \le c_{ij}^{\max}, \tag{1.4.6}$$

$$\sum_{(i,j)} c_{ij} = C_{\ell t} . \qquad (1.4.7)$$

Problem (1.4.5-1.4.7), together with several related optimization problems, is solved by means of simulation [7,54,62,64,92].

Problem IIIB can be applied to various stochastic network projects with variable speeds, e.g., to construction projects where each activity can be operated at several possible speeds that are subject to random disturbances. Such speeds may correspond to different hours a day per worker and, thus, depend on the degree of intensity of the project's realization. Thus index r_g of the control action is the index of the speed to be introduced at each control point t_g. For projects with variable speeds two optimization objectives may be implemented [66]:

- to minimize the number of inspection points, and
- to minimize the average index of the project's speeds.

The control model is as follows: at any routine control point t_g, determine values t_{g+1} and r_g to minimize two contradicting objectives

$$\underset{\{t_{g+1}, r_g\}}{Min} \left\{ W(t_g) \right\}, \qquad (1.4.8)$$

$$\underset{\{t_{g+1}, r_g\}}{Min} \left\{ \bar{r}_g (t_g) \right\} \qquad (1.4.9)$$

subject to

$$t_{g+1} - t_g \ge \Delta , \qquad (1.4.10)$$

$$r_g = \underset{1 \le r_g \le r}{Min} \left\{ r_g : \Pr(t_g, r_g) \ge p^* \right\} . \qquad (1.4.11)$$

Restriction (1.4.11) means that at each control point t_g, the problem is to determine the *minimal* index of the project's speed that, with the given chance constraint guarantees meeting the project's due date on time. Thus, the restriction prohibits unnecessarily high intense speeds. The solution of optimization problem (1.4.8-1.4.11) is outlined in [66].

The solution of Problem IV, i.e., determining control points t_g [68], is outlined in Chapters 4 and 15-16. Both algorithms A and B may be applied, but the second one is more efficient and requires less computational time [72]. The solution of the scheduling Problem V at the lowest hierarchical level is outlined in [70,72]. The solution of Problem I serves as the initial data for Problems II and III (at the project level). The solution of Problem III serves, in turn, as initial data for Problem IV, which carries out on-line control, i.e., determines the optimal control points to inspect the progress of the project. This is done by determining the planned trajectories that must be repeatedly corrected in the course of the project's realization. If, at any control point, it turns out that a project deviates from the planned advancement trajectory, an error signal is generated, and decision-making takes place based on resolving Problem III to re-orient the progress of the project in the desired direction, i.e., to maximize the probability of meeting the deadline in time. If the problem's solution enables the project's deadline to be met, subject to the chance constraint, a corrected planned trajectory is determined and Problem IV is solved again to determine the next control point. Otherwise, an emergency signal is generated and decision-making takes place at the company level. Problem I is resolved under emergency conditions to reassign the remaining resources (e.g., the remaining budget) among the non-accomplished projects. Thus, in the course of controlling a group of projects, the latter are first optimized on line "from top to bottom". In the case of emergency, the generated "bottom-to-top" signals are converted into control actions to enable the projects' due dates to be met on time.

§1.5 A linkage between deterministic and stochastic approaches in project management

1.5.1 The main stages

It can be well-recognized that controlling *a large-scale stochastic R&D network project*, e.g., of PERT type, cannot be facilitated by using deterministic methods only. Substituting random activity durations by their corresponding average values leads to unavoidable mistakes, mainly in calculating the project's parameters. Thus, the project remains practically uncontrolled. However, it is also practically impossible to apply stochastic techniques to a *large-scale* project, since the existing control procedures usually prove to be unfit for large-scale organization systems [54,147,149]. Thus, the only possible outcome to be suggested is as follows:

1. First, modify the initial large-scale project to an enlarged aggregated network *of medium size* (comprising not more than 40÷50 activities).
2. Second, apply to that aggregated project all the *stochastic control techniques* under consideration, in order to determine the project's proper speeds and inspection points.
3. Third, *reaggregate* the enlarged project to its initial size.
4. Four, *reschedule* the activities between the adjacent inspection (control) points according to their average values, i.e., implement deterministic scheduling techniques for project's fragments between adjacent decision-making points. The latter can be utilized as corrective indications.

Let us examine the main stages of the regarded procedure in greater detail.

1.5.2 Developing enlarged aggregated networks with random activity durations

According to the project's Work Breakdown Structure (WBS) [149], an initial network is presented in the form of a group of lists of initial activities. The name of the activity is taken from the WBS.

We will henceforth call a fragment a list of activities together with all the links both entering and leaving that fragment. The step-by-step procedure [53-

54] of developing an aggregated network is as follows:

Given:

- activities (i, j) entering the PERT initial network $G(N, A)$;
- random activity durations t_{ij} with pregiven density distribution.

Step 1. Simulate random durations t_{ij}, $(i, j) \subset G(N, A)$.

Step 2. On the basis of simulated values t_{ij} calculate for each $i \in N$ the earliest moment of the event's realization, $T^{\xi}(i)$, where ξ denotes the index of the simulation run.

Step 3. Repeat Steps 1→2 M times in order to obtain representative statistics.

Step 4. Calculate

$$T_{ear}(i) = \min_{1 \le \xi \le M} T^{\xi}(i);$$

$$T_{lat}(i) = \max_{1 \le \xi \le M} T^{\xi}(i).$$

Step 5. By using decomposition methods [49,149,153] subdivide the initial set into enlarged fragments. Each fragment comprises a list of detailed activities together with all links connecting activities entering the list ("internal" links) as well as "external" links connecting the fragment with other fragments.

Steps 6→10 have to be implemented for each fragment $F \subset G(N, A)$ separately.

Step 6. Determine two events i_{st}^F and i_{fin}^F which we will call henceforth the start and the finish events of fragment F, respectively:

$i_{st}^F \in F$ delivers the minimum to $\underset{i \in F}{Min}\{T_{ear}(i)\}$, and

$i_{fin}^F \in F$ delivers the maximum to $\underset{i \in F}{Min}\{T_{lat}(i)\}$, where $T_{ear}(i)$ and $T_{lat}(i)$ have been calculated on Step 4.

Step 7. For both events i_{st}^F and i_{fin}^F calculate the earliest and the latest moments (refer also to Step 4): $T_{ear}(i_{st}^F)$, $T_{lat}(i_{st}^F)$, $T_{ear}(i_{fin}^F)$, $T_{lat}(i_{fin}^F)$.

Step 8. Calculate the minimal fragment's duration $\tau_F^{\min} = T_{ear}(i_{fin}^F) - T_{lat}(i_{st}^F)$.

Step 9. Calculate the maximal fragment's duration $\tau_F^{\max} = T_{lat}(i_{fin}^F) - T_{ear}(i_{st}^F)$.

Step 10. Assume that the fragment's duration τ_F is a random variable with a β-distribution density function

$$p_F(x) = \frac{12}{\left(\tau_F^{\max} - \tau_F^{\min}\right)^4}\left(x - \tau_F^{\min}\right)\left(\tau_F^{\max} - x\right)^2$$

within the range $\left[\tau_F^{\min}, \tau_F^{\max}\right]$. The justification of this probability law will be demonstrated in Chapter 2.

Thus, the project is aggregated with random durations of enlarged activities.

1.5.3 On-line control problems for medium-size projects

For most medium-size network projects under random disturbances the progress of the project cannot be inspected and measured continuously, but only at preset inspection points. On-line control determines both inspection points and control actions to be implemented at those points in order to alter the progress of the project in the desired direction. Such control actions may boil down to the following:

1) redistributing the budget among the project activities in order to enhance the project's speed, or
2) introducing additional shifts, etc., to change the speed of the progress of the project without investing additional resources, etc.

Such control actions are usually aimed at minimizing either the number of inspection points, or the average project's speed subject to a chance constraint to meet the project's due date on time. The corresponding control algorithms are outlined in Chapters 4-6 and can be applied to small- and medium-size projects only.

After implementing the control actions the modified aggregated network has to be transformed back to the initial network.

Consider a medium-size PERT type network model with due date D. A desirable probability p^* that in practice enables completion of the project in time, is pregiven. At each control moment t_g the project management may introduce several possible alternative speeds v_{t_g} to proceed with until the next control point. Let V_t be the project's output (project volume) observed at control point $t > 0$ and let the project's target (goal) be V^*. Denote $\Pr(t_g, v_{t_g})$ the confidence probability to accomplish the project in time after introducing speed v_{t_g} at control point t_g.

The main control problem [68] boils down to determining both control (inspection) points t_g ($g = 1,2,...,N$) and speeds v_{t_g} to proceed with from that point on until the next adjacent control point t_{g+1}, in order to minimize number N of inspection points

$$\underset{\{t_g, v_{t_g}\}}{Min} N \tag{1.5.1}$$

subject to

$$\Pr\{t_g, v_{t_g}\} \geq p^*, \tag{1.5.2}$$

$$t_0 = 0, \tag{1.5.3}$$

$$t_N = N, \tag{1.5.4}$$

$$t_{g+1} - t_g \geq \Delta. \tag{1.5.5}$$

Pregiven value Δ is usually introduced to force convergence.

Note that if introducing control actions results in determining the *project's speeds* v_{t_g} to proceed with until the next control point t_{g+1} and if *several alternative speeds can be chosen*, then the optimal control action enables implementing *the minimal speed* to develop the project honoring chance constraint (1.5.2).

Control model (1.5.1-1.5.5) is a stochastic optimization problem with a non-linear chance constraint and a random number of optimized variables. Such a problem proves to be too difficult to solve in the general case, especially for large-size projects.

§1.6 Conclusions

1. After introducing control actions outlined in §1.2, the modified medium-size aggregated network is transformed back to the initial network and the project's realization proceeds.
2. All other procedures at the project's level, e.g., scheduling procedures, are carried out for the initial network *between two adjacent control points* by using traditional deterministic techniques based on the activities' average durations. *Although such calculations usually comprise biased estimates and evaluation errors, they are periodically corrected by introducing proper control actions. That is why those procedures in combination with control actions are more effective than without controlling the project in inspection points.*
3. The suggested approach has to be implemented as an *additional tool* in order to help the project manager to realize the project on time. Applying the corresponding techniques does not result in undertaking any revisions in traditional project management procedures.

In the next two Chapters we are going to undertake a review of all probabilistic aspects and parameters of stochastic network modeling. Later on we will outline in detail the problems of network R&D projecting under random disturbances.

Chapter 2. Random Activity Durations in Stochastic Project Management

§2.1 Justification of probability laws for man-machine network activity durations

In all Network Analysis Methods (NAM) applied to planning and control while creating a new complicated advanced technology project with uncertainty of its activity durations, it is common to assume those durations being in fact random variables. In other words, indeterminate activity durations are assumed to be randomly distributed with a probability law accepted for the regarded NAM and common to all activities engaged. As to parameters of the probability law, they are preset for each activity by their responsible executers on the basis of either standard values, or a-priori considerations, or their personal professional experience. Nearly for all NAM the activity durations' probability density function (p.d.f.) is *a-priori* assumed to possess the following qualities:

a) continuity;

b) unimodality;

c) two non-negative intersection points between the p.d.f. and the x axis.

The most common probability law conforming to the above requirements is the famous beta-distribution which is successfully used in major NAM [7, 22, 39,46,49-51,98,100,104,116-117,125,146,etc.].

The general property of beta-distribution boils down to a variety of insignificant random factors with only minor influence on the p.d.f. shape, aside a few random factors of significant influence. As a result of the latter the resulting p.d.f. shape becomes usually asymmetrical. This circumstance becomes dominant when executing the majority of the network activities. This is also the main reason to a-priori preference of beta-distribution as the typical p.d.f. for man-machine operations.

The relation for beta-distribution p.d.f. may be written down as

$$B(p,q,x)=\begin{cases} \dfrac{1}{B(p,q)}x^{p-1}(1-x)^{q-1} & for \quad 0\le x\le 1, \\ 0 & for \quad x<0,\ x>1, \end{cases} \tag{2.1.1}$$

where $B(p,q)$ stands for the beta-function

$$B(p,q)=\int_0^1 x^{p-1}(1-x)^{q-1}dx=\frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}, \tag{2.1.2}$$

and the gamma-function $\Gamma(z)$ is determined as

$$\Gamma(z)=\int_0^\infty e^{-t}t^{z-1}dt,$$

while for integer z the function $\Gamma(z)=1\cdot 2\cdot\ldots\cdot(z-1)=(z-1)!$. The central moment of order r may be calculated as

$$\frac{1}{B(p,q)}\int_0^1 x^{r+p-1}(1-x)^{q-1}dx=\frac{B(p+r,q)}{B(p,q)}. \tag{2.1.3}$$

For $r=1$ we obtain the relation for average $E(x)$

$$E(x)=\frac{B(p+1,q)}{B(p,q)}=\frac{\Gamma(p+1)\Gamma(q)\Gamma(p+q)}{\Gamma(p+q+1)\Gamma(p)\Gamma(q)}=\frac{p}{p+q}. \tag{2.1.4}$$

Variance $V(x)$ (for $r=2$) may be calculated as follows:

$$V(x)=\frac{B(p+2,q)}{B(p,q)}-\left(\frac{p}{p+q}\right)^2=\frac{p(p+1)}{(p+q)(p+q+1)}-\frac{p^2}{(p+q)^2}=\frac{pq}{(p+q)^2(p+q+1)}. \tag{2.1.5}$$

It can be well-recognized that specific function properties in (2.1.1) depend parametrically on p and q, while for $p>2$ (and, correspondingly, for $q>2$) the p.d.f. turns zero in its left (or right) terminal point together with its first derivative. For $1<p<2$ (and, correspondingly, $1<q<2$) the p.d.f. has a vertical tangent in its left (or right) terminal point. For $0<p<1$ (and, correspondingly, $0<q<1$) the p.d.f. turns infinity, if values of x fall into the left (right) terminal point, while a vertical line intersecting its left terminal point would be the tangent. For $p\le 0$ (and, correspondingly, $q\le 0$) the integral in (2.1.2) turns infinity, which means the p.d.f. ceases to exist.

The justification of using probability laws for man-machine operations in organization systems under random disturbances has been considered in [7,49,54]. It can be well-recognized that the outlined results fully comprise the case of activity networks in stochastic project management. Two cases have been considered [7,49,54]:

- case of one processor to operate a man-machine activity;
- case of several processors.

The first case covers a man-machine operation which is carried out by one processor, i.e., by one resource unit. The processor may be a machine, a proving ground, a department in a design office, etc.

It is assumed that the operation starts to be processed at a pregiven moment T_0. The completion moment F of the operation is a random value with distribution range $[T_1,T_2]$. Moment T_1 is the operation's completion moment on condition that the operation will be processed without breaks and without delays, i.e., value T_1 is a pregiven deterministic value. Assume, further, that the interval $[T_0,T_1]$ is subdivided into n equal elementary periods with length $(T_1-T_0)/n$. If within the first elementary period $[T_0, T_0+(T_1-T_0)/n]$ a break occurs, it causes a delay of length $\Delta=(T_2-T_1)/n$. The operation stops to be processed within the period of delay in order to undertake necessary refinements, and later on proceeds functioning with the finishing time of the first elementary period $T_0+(T_1-T_0)/n+(T_2-T_1)/n = T_0+(T_2-T_0)/n$.

It is assumed that there cannot be more than one break in each elementary period. The probability of a break at the very beginning of the operation is set to be p. However, in the course of carrying out the operation, the latter possesses certain features of self-adaptivity, as follows:

- the occurrence of a break within a certain elementary period results in increasing the probability of a new break at the next period by value η, and

- on the contrary, the absence of a break within a certain period decreases the probability of a new break within the next period, practically by the same value.

The probabilistic self-adaptivity can be formalized as follows:

Denote A_i^k the event of occurrence of a break within the $(i+1)$-th elementary period, on condition, that within the i preceding elementary periods k breaks occurred, $1 \le k \le i \le n$. It is assumed that relation

$$P\left(A_i^k\right) = \frac{p + k \cdot \eta}{1 + i \cdot \eta} \qquad (2.1.6)$$

holds. Note that (2.1.6) is, indeed, a realistic assumption.

Relation (2.1.6) enables obtaining an important assertion. Let $P\left(A_i^0\right)$ be the probability of the occurrence of a break within the $(i+1)$-th period on condition, that there have been no breaks at all as yet. Since

$$P\left(A_i^0\right) = \frac{p}{1 + i \cdot p}, \qquad (2.1.7)$$

it can be well-recognized that relation

$$\frac{P\left(A_i^{k+1}\right) - P\left(A_i^k\right)}{P\left(A_i^0\right)} = \frac{\eta}{p} \qquad (2.1.8)$$

holds. Thus, an assertion can be formulated as follows:

Assertion. Self-adaptivity (2.1.6) results in a probability law for delays with a constant ratio (2.1.8) for a single delay.

Let us calculate the probability $P_{m,n}$ of obtaining m delays within n elementary periods, i.e., the probability of completing the operation at the moment

$$F = T_1 + m \cdot \Delta = T_1 + \frac{m}{n}\left(T_2 - T_1\right).$$

The number of sequences of n elements with m delays within the period $[T_0, F]$ is equal C_n^m, while the probability of each such sequence equals

$$\frac{\left[\prod_{i=0}^{m-1}\left(p + i\eta\right)\right]\left[\prod_{i=0}^{n-m-1}\left(1 - \eta + i\eta\right)\right]}{\prod_{i=0}^{n-1}\left(1 + i\eta\right)}. \qquad (2.1.9)$$

Relation (2.1.9) stems from the fact that if breaks occurred within h periods and did not occur within k periods, the probability of the occurrence of the delay at the next period is equal

$$\frac{p + h\eta}{1 + \left(k+h\right)\eta}, \qquad (2.1.10)$$

while the probability of the delay's non-appearance at the next period satisfies

$$\frac{1 - \eta + k\eta}{1 + \left(k+h\right)\eta}. \qquad (2.1.11)$$

Using (2.1.10-2.1.11), we finally obtain

$$P_{m,n} = C_n^m \frac{\left[\prod_{i=0}^{m-1}(p+i\eta)\right]\left[\prod_{i=0}^{n-m-1}(1-\eta+k\eta)\right]}{\prod_{i=0}^{n-1}(1+i\eta)} . \qquad (2.1.12)$$

Note that $\eta = 0$, i.e., the absence of self-adaptivity, results in a regular binomial distribution.

Let us now obtain the limit value $P_{m,n}$ on condition that $n \to \infty$. It has been shown [49,54] that the p.d.f. of random value $\xi = \lim_{n\to\infty} \frac{m}{n}$ satisfies

$$p_\xi(x) = \frac{1}{B(p,q)} x^{p-1}(1-x)^{q-1} , \qquad (2.1.13)$$

where $B(p,q)$ represents again the beta-function. Thus, relation (2.1.13) practically coincides with (2.1.1).

Thus, changing more or less the implemented assumptions, we may alter to a certain extent the structure of the p.d.f. At the same time, its essential features (e.g., asymmetry, unimodality, etc.) remain unchanged.

The considered in [7,49,54] case of several processors enables conclusion as follows:

1. For a broad spectrum of activities being processed by means of several identical resource units, the corresponding time – activity density functions prove to be asymmetric functions with finite upper and lower distribution limits. Those p.d.f.'s are close to a beta-distribution p.d.f.
2. Various assumptions in activity – time analysis (and in risk analysis as well!) center on determining a numerous "family" of beta-distributions with different versions - parameters p and q - of the general p.d.f. (2.1.1). Those versions may result in changing certain estimates for certain activities. At the same time, *they have practically no influence on the project as a whole*.
3. Thus, a general conclusion can be drawn that a random activity – time duration has a very high potential to be close to one of the beta-distribution probability density functions. The obtained theoretical grounds cover a broad variety of activities including the man-machine activities (with one processor) and semi-automated activities (with several processors).

§2.2 The basic concepts of PERT analysis

In the course of creating the theoretical and methodological basis for NAM, additional information is required. This information should include the probabilistic network model of developing the new complicated advanced technology project, as well as estimates of parameters entering the p.d.f. of activity durations $t(i,j)$ within the network.

This section will be dedicated mostly to the description of probabilistic models in PERT-type systems. The methodological basis of research and development in PERT includes the following assumptions [22-23,25,42,49-51,100,116-117,etc.]:

1. Activity duration $t(i,j)$ is a random variable distributed on the interval $[a,b]$ by the beta-distribution law with p.d.f.
$$\varphi(t) = C(t-a)^{p-1}(b-t)^{q-1} \quad . \tag{2.2.1}$$
2. P.d.f. $\varphi(t)$ central moments - namely, average $E(x)$ and variance $V(x)$ - may be determined from relations
$$E(i,j) = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}, \tag{2.2.2}$$
$$V(i,j) = \frac{(b_{ij} - a_{ij})^2}{36}, \tag{2.2.3}$$
where a_{ij}, b_{ij} and m_{ij} stand, correspondingly, for the optimistic, pessimistic and most probable (modal) duration estimates preset by the responsible executers of activity (i,j).

Additional assumptions refer to the methodology of calculating network parameters in general and would be outlined in the following sections. As demonstrated below, relations (2.2.2-2.2.3) may be partly considered as being of empirical origin.

Consider p.d.f. $\varphi(t)$ with parameters $p-1=\alpha$, $q-1=\gamma$, $a=0$, $b=1$. We obtain
$$\varphi(t) = Ct^{\alpha}(1-t)^{\gamma}, \tag{2.2.4}$$
where
$$C = \frac{\Gamma(\alpha+\gamma+2)}{\Gamma(\alpha+1)\Gamma(\gamma+1)}.$$

Standardized parameters $E(x)$, m_x and $V(x)$ in this case will satisfy
$$E(x) = \frac{\alpha+1}{\alpha+\gamma+2}, \tag{2.2.5}$$
$$m_x = \frac{\alpha}{\alpha+\gamma}, \tag{2.2.6}$$
$$V(x) = \frac{(\alpha+1)(\gamma+1)}{(\alpha+\gamma+2)^2(\alpha+\gamma+3)}. \tag{2.2.7}$$

In can be well-recognized that the standardized modal value m_x is connected with activity duration estimates a_{ij}, b_{ij} and m_{ij} by means of relation
$$m_x = \frac{m_{ij} - a_{ij}}{b_{ij} - a_{ij}}.$$

Modal estimate m_{ij} (together with its standardized value m_x) is preset by responsible executors and may be regarded as a fixed value; thus, relation (2.2.6) enables to express parameter γ by means of α:
$$\gamma = \alpha\frac{1-m_x}{m_x}. \tag{2.2.8}$$

Since for each considered activity (i,j) the ratio $\frac{1-m_x}{m_x}$ is a constant value, it proves beneficial to re-write the p.d.f. relation $\varphi(t)$ in the form of

$$\varphi(t) = Ct^{\alpha}(1-t)^{\alpha\left(\frac{1}{m_x}-1\right)}, \qquad (2.2.9)$$

where the unbound parameter α determines properties of the curve of the corresponding density function. When diminishing α the curve becomes more sloping and turns into the uniform distribution law with variance $V(x)=\frac{1}{12}$ and average $E(x)=\frac{1}{2}$. For increasing values of α the curve becomes increasingly less asymmetrical and approximates the curve shape of normal density distribution.

For $m_x \to 1$ distribution function (2.2.9) approximates the power function Ct^{α}, for $m_x \to 0$ p.d.f. $\varphi(t)$ approximates the δ-function of Dirac. Abrupt changes of m_x cause significant fluctuations of the asymmetry coefficient. Thus, density function (2.2.9) satisfies conditions of unimodality, continuity and possesses two non-negative intersection points with the x axis. It can be well-recognized that, in other words, (2.2.9) conforms to the necessary requirements of network activity durations distribution law outlined in the previous section.

Assume $t=(x-a)/(b-a)$ and switch over to a non-standardized distribution law

$$f(x) = N(x-a)^{\alpha}(b-x)^{\alpha\left(\frac{1}{m_x}-1\right)}, \qquad (2.2.10)$$

where $N = \left[(b-a)^{\alpha+\gamma+1}B(\alpha+1,\gamma+1)\right]^{-1}$. Taking into account (2.2.8), relation $\gamma = \alpha\left(\frac{1}{m_x}-1\right)$ holds. The modal value m can be easily calculated from relation

$$\alpha(b-m) = \gamma(m-a). \qquad (2.2.11)$$

It can be well-recognized [49] that the first and the second central moments (average $E(x)$ and variance $V(x)$, being designated as K_1 and K_2, correspondingly) comply with the following relations

$$(\alpha+\gamma+2)K_1 = (\alpha+\gamma+2)m + (a+b) - 2m,$$

$$(\alpha+\gamma+3)K_2 = (a+b)K_1 - ab - K_1^2.$$

Central moments of higher orders (for $n = 2,3,...$) may be determined by a recurrent relation

$$(\alpha+\gamma+2+n)K_{n+1} = n(a+b)K_n - n\left[K_1K_n + (n-1)K_2K_{n-1} + ... + K_nK_1\right]. \qquad (2.2.12)$$

The relation for K_1 may be re-written as

$$K_1 - m = \frac{2}{\alpha+\gamma+2}\left(\frac{(a+b)}{2} - m\right),$$

and finally we obtain

$$E(x) = K_1 = \frac{(\alpha+\gamma)m + (a+b)}{\alpha+\gamma+2}. \qquad (2.2.13)$$

After carrying out a thorough statistical analysis taking into account both empirical and experimental subjects, the creators of PERT methodology established [117] that $\alpha + \gamma \approx 4$. This enables developing the following modification for the recurrent relation (2.2.12)

$$\begin{cases} 6K_1 = (a+b) + 4m, \\ 7K_2 = (a+b)K_1 - ab - K_1^2, \\ 4K_3 = (a+b)K_2 - 2K_1K_2, \\ 3K_4 = (a+b)K_3 - 2(K_1K_3 + K_2^2), \ etc. \end{cases} \tag{2.2.14}$$

Variance $V(x)$ on condition $\alpha + \gamma = 4$ may be determined from (2.2.14) as follows:

$$V(x) = K_2 = \frac{(b-a)^2}{28} - \frac{4}{63}\left[\frac{a+b}{2} - m\right]^2. \tag{2.2.15}$$

If the modal value appears in the vicinity of the average $(a+b)/2$, variance $V(x)$ may be estimated as $V(x) = (b-a)^2/28$; if, on the contrary, the modal value falls close to the border of a, this would result in $V(x) = 5(b-a)^2/252$. Thus, it can be well-recognized that the variance is slightly affected by the modal value position and may be located in the interval $\left[(b-a)^2/49, (b-a)^2/25\right]$. The latter circumstance enabled creators of PERT to replace the more accurate although somewhat cumbersome relation (2.2.15) by its much easier approximation

$$V(x) \approx (b-a)^2/36. \tag{2.2.16}$$

As far as the average is concerned, the first relation from those listed in (2.2.14) boils down to the well-known PERT estimate

$$E(x) = (a+b+4m)/6. \tag{2.2.17}$$

A number of researchers (e.g., [23,25,36,42,95,116,146,etc.]) suggest a slightly different justification for approximate relations (2.2.16-2.2.17). On the basis of estimates $\alpha = 2+\sqrt{2}$, $\gamma = 2-\sqrt{2}$ or vice versa ($\alpha = 2-\sqrt{2}, \gamma = 2+\sqrt{2}$) suggested by Pearson, one may obtain relations (2.2.16-2.2.17) directly from the previously established formulae (2.2.11-2.2.12). Although, it should be noted that by the latter approach, we are in fact fixing parametrical values m_x quite rigidly (the same goes of course for values m as being connected with m_x by $m = m_x(b-a) + a$). This inflexibility contradicts to the principle of empowering the responsible executor to estimate values m on the basis of his personal professional experience and skills.

Thus, it can be well-recognized that the theoretical grounds of PERT contain certain fundamental contradictions, which would be outlined in greater detail below. These contradictions stem mostly from the fact that it is impossible to derive relations (2.2.16-2.2.17) directly from (2.2.10), since three parameters out of four in (2.2.10) depend on assessing values a, b and m by responsible executors, while imposing any additional binding assumptions on these three will immediately cause discrepancies with the accepted estimation procedure.

Let us consider possible inaccuracies in estimating average $E(x)$ and variance $V(x)$ due to assumptions which had to be made in the theoretical grounds of PERT. These inaccuracies can be separated into three groups:

a) a group of errors due to accepting the beta-distribution law as the standard probabilistic instrument for man-machine activity durations; we will designate those errors as type 1;

b) a group of errors originating from using relations (2.2.16-2.2.17) to estimate average $E(x)$ and variance $V(x)$ on condition that activity durations p.d.f. is indeed represented by (2.2.1) - errors of type 2;

c) a group of errors (call them type 3) caused by inaccuracies of estimating parametrical values a, b and m by responsible executors (experts) on condition that both p.d.f. (2.2.1) and relations (2.2.16-2.2.17) are indeed applicable.

McCrimmon and Ryavec [116] conducted research as to possible errors in estimating average $E(x)$ and variance $V(x)$ due to accepting the beta-distribution law as the only standard probabilistic instrument for man-machine activity durations (errors of type 1), on condition that parametrical values a, b and m are predetermined by responsible executors as anticipated. For the sake of simplicity consider the standardized distribution interval with $a = 0$, $b = 1$, $0 \le m_x = (m - a)/b - a < \frac{1}{2}$. While comparing various distribution laws with similarly shaped central moments (quasi-uniform distribution with the average close to 0.5, and quasi-delta distribution law with the average matching the modal value m), the following marginal total errors have been identified:

a) for the standardized average $E(x)$ error Δ_1 equals $\Delta_1 = \frac{1}{3}(1 - 2m_x)$;

b) for the standardized standard deviation σ_x the maximal error Δ_2 equals $\Delta_2 = \frac{1}{6}$.

It can be well-recognized that error Δ_1 depends parametrically on modal value m_x; when m_x gets close to the interval margin Δ_1 might become as large as 33%, while error Δ_2 does not depend on the modal value.

Estimate for Δ_1 has been subsequently improved by Lukaszewicz [115] who demonstrated that the maximal total error for a single-mode continuous distribution in interval $[0,1]$ should not exceed $\Delta_1 = \frac{1}{3}\left(1 - \frac{m_x}{2}\right)$. Thus, the results by McCrimmon and Ryavec, on one hand, and of Lukaszewicz, on the other hand, provide a perfect match for distributions with modal values close to zero, but they differ substantially for $m_x \approx \frac{1}{2}$, which as a matter of fact proves to be the most common case in practice.

Consider now [116] maximal total errors in estimating average $E(x)$ and standard deviation σ_x for beta-distribution populations caused by assumption $\sigma_x = \frac{1}{6}(b-a)$ and approximation $E(x)=(a+b+4m)/6$, namely, type 2 errors. Using relations (2.2.5-2.2.7) for estimating standardized values of average, mode and variety, correspondingly, and comparing them with appropriate assessments utilized in PERT, we obtain the following relations for errors Δ_1 and Δ_2 (for $a=0$, $b=1$):

$$\begin{cases} \Delta_1 = \left| \dfrac{4m_x+1}{6} - \dfrac{m_x(\alpha+1)}{\alpha+2m} \right|; \\ \Delta_2 = \left| \dfrac{1}{6} - \sqrt{\dfrac{m_x^2(\alpha+1)(\alpha-\alpha m_x+m_x)}{(\alpha+2m_x)^2(\alpha+3m_x)}} \right|. \end{cases} \tag{2.2.18}$$

Further analysis of (2.2.18) reveals that depending on parameters involved, errors Δ_1 and Δ_2 may obtain the following highest values:

$$\begin{cases} \max \Delta_1 = 33\% \\ \max \Delta_2 = 17\% \,. \end{cases} \tag{2.2.19}$$

Finally, consider type 3 errors Δ_1 and Δ_2 on assumption that both the beta-distribution law and relations (2.2.16-2.2.17) are indeed applicable and true, but the experts' estimates of parameters a, b and m may contain inaccuracies. Following [116], assume that a, b and m denote *true values* of the lower and higher bounds as well as of the mode of the distribution, while responsible executors acting as experts determine approximate estimates t_a, t_b and t_m of the same satisfying $0.8a \le t_a \le 1.1a$; $0.9b \le t_b \le 1.2b$; $0.9m \le t_m \le 1.1m$. Apply additional condition $a \le m \le \frac{a+b}{2}$ and obtain the worst total estimate for error Δ_1 (type 3):

$$\Delta_1 = \frac{1}{b-a}\max\left\{ \left| \frac{(0.8a+3.6m+0.9b)-(a+4m+b)}{6} \right|, \left| \frac{(1.1a+4.4m+1.2b)-(a+4m+b)}{6} \right| \right\} = $$
$$= \frac{1}{60}\left(\frac{a+4m+2b}{b-a} \right). \tag{2.2.20}$$

In the same way, the corresponding estimate Δ_2 for σ_x may be calculated as

$$\Delta_2 = \frac{1}{b-a}\max\left\{ \left| \frac{0.9b-1.1a-(b-a)}{6} \right|, \left| \frac{1.2b-0.8a-(b-a)}{6} \right| \right\} = \frac{1}{30}\,\frac{b+a}{b-a}. \tag{2.2.21}$$

As for previous types of errors, modal value m does not influence the estimate for Δ_2.

Internal contradictions within the methodology of determining probabilistic parameters for network activity durations distribution in PERT caused scientific researchers dealing with development and implementation of NAM, to criticize the method [22-23,25,36,49,116,etc.]. Some of the critics include recommendations as to further improvement and modification of the regarded procedure, as outlined in the following sections.

§2.3 Attempts to refine the PERT assumptions

To sum up, no one scientific discussion in the last five decades caused so much agitation and excitement as the ongoing attempts to improve the PERT analysis based on the subjective determination of the “optimistic”, “most likely” and “pessimistic” activity-times (a, $m$and b, respectively). There are, indeed, nothing but a few areas as open until now to such a sharp criticism as in PERT applications.

The creators of PERT [22,44-45,116-117,etc.] worked out the basic concepts of PERT analysis, and suggested estimates of the average and variance values (2.2.2-2.2.3) subject to the assumption that the density distribution of the activity time is a beta-distribution (2.2.1). Take once more a brief overlook of the heated discussion as to PERT weaknesses and challenges.

Grubbs [95] pointed out the lack of theoretical justification and the unavoidable defects of the PERT statements, since estimates (2.2.2) and (2.2.3) are, indeed, “rough” and cannot be obtained from (2.2.1) on the basis of values a, m and b determined by the analyst. Various authors noted [46,49,146,162] that there is a tendency to choose the most likely activity-time m much closer to the optimistic value a than to the pessimistic one, b, since the latter is usually difficult to determine and so is chosen conservatively large. Moreover, it is shown [49] that value m_x, being subsequently determined, has approximately one and the same relative location point in $[a,b]$ for different activities. This provides an opportunity to simplify PERT analysis at the expense of some additional assumptions. McCrimmon and Ryavec [116], Lukaszewicz [115] and Welsh [160] examined various errors introduced by imposing PERT assumptions, and came to the conclusion that these errors may be as great as 33%. Murray [125], Donaldson [36] and Coon [25] suggested certain modifications of the PERT analysis, but the main contradictions remained. Farnum and Stanton [42] presented an interesting improvement of estimates (2.2.2-2.2.3) when the modal value m is close to the upper or lower limits of the distribution. This modification, however, makes the distribution law rather uncertain, and makes it difficult to simulate the activity network. However, it can be shown that there are still theoretical grounds for improving the estimates without complicating the PERT analysis. We will present some modifications of the PERT model under various assumptions which may refine the model’s accuracy.

In our opinion, assumption $\alpha + \gamma \approx 4$ may become inadequate since the actual standard deviation may be smaller than $1/6$, especially in the tails of the distribution [58]. In order to make the assumption more flexible, we assume that the sum $\alpha + \gamma$ in (2.2.6) *is approximately constant but not predetermined*, i.e., relation

$$\alpha + \gamma \approx Z \text{ - const} \qquad (2.3.1)$$

holds. From (2.2.6) we obtain

$$\alpha = Zm_x, \qquad (2.3.2)$$

and values $E(x)$ (2.2.5) and $V(x)$ (2.2.7) are

$$E(x)=\frac{Zm_x+1}{Z+2}, \qquad (2.3.3)$$

$$V(x)=\frac{1+Z+Z^2m_x-Z^2m_x^2}{(Z+2)^2(Z+3)}. \qquad (2.3.4)$$

To satisfy the main PERT assumptions we introduce a reasonable statement [58]: the average value $V(m_x)$ for $0<m_x<1$ has to be equal $1/36$, i.e.,

$$\int_0^1 V(m_x)dm_x=\frac{1}{36}. \qquad (2.3.5)$$

Substituting (2.3.4) by (2.3.5) and solving (2.3.5) for Z, we obtain $Z=4.55$. Approximating Z to 4.5 and getting

$$\begin{cases}\alpha=4.5m_x\\ \gamma=4.5(1-m_x),\end{cases} \qquad (2.3.6)$$

we finally obtain

$$E(x)=\frac{9m_x+2}{13}, \qquad (2.3.7)$$

$$V(x)=\frac{1}{1268}\left(22+81m_x-81m_x^2\right), \qquad (2.3.8)$$

together with the final density distribution function

$$\varphi(t)=\frac{\Gamma(6.5)}{\Gamma(4.5m_x+1)\Gamma(5.5-4.5m_x)}\cdot t^{4.5m_x}(1-t)^{4.5(1-m_x)}. \qquad (2.3.9)$$

Density function (2.3.9) can be simulated, e.g., by the acceptance-rejection method [43], but this requires much computational time.

For a simplified approximation $Z=5$ we, in turn, obtain a simplified beta-function

$$\varphi(t)=C\cdot t^{5m_x}(1-t)^{5(1-m_x)} \qquad (2.3.10)$$

with parameters

$$E(x)=\frac{5m_x+1}{7}, \qquad (2.3.11)$$

$$V(x)=\frac{1}{392}\left(6+25m_x-25m_x^2\right). \qquad (2.3.12)$$

Here coefficient C stands for

$$C=\frac{\Gamma(7)}{\Gamma(5m_x+1)\cdot\Gamma(6-5m_x)}. \qquad (2.3.13)$$

For the general beta-distribution of activity time, estimates (2.3.7-2.3.8) are transformed to

$$E(x)=\frac{2a+9m+2b}{13}, \qquad (2.3.14)$$

$$V(x)=\frac{(b-a)^2}{1268}\left\{22+81\cdot\frac{m-a}{b-a}-81\left(\frac{m-a}{b-a}\right)^2\right\}. \qquad (2.3.15)$$

For the simplified case $Z=5$ estimates (2.3.11-2.3.12) are transformed to

$$E(x)=\frac{a+5m+b}{7}, \qquad (2.3.16)$$

$$V(x)=\frac{(b-a)^2}{392}\left\{6+25\cdot\frac{m-a}{b-a}-25\left(\frac{m-a}{b-a}\right)^2\right\}. \qquad (2.3.17)$$

Note that when the estimated mode m_x is close to the upper or lower limits of the distribution, variance (2.3.8) provides a better approximation to $1/36$, than (2.3.12). However, estimate (2.3.8) is essentially more complicated and more difficult in usage, especially in simulation modeling. Thus, all attempts to amend the PERT analysis models result in raising the complexity of the latter. A conclusion can be drawn that a certain compromise has to be agreed upon in order to close the acute scientific discussion. At the same time, one has to bear in mind that both estimates (2.3.12) and (2.3.8) being more complicated perform better (from the theoretical point of view) than the "classical" estimates (2.3.5-2.3.7).

Let us turn to another important simplification in PERT assumptions. Analyzing over a lengthy period different network projects [49], one may come to the conclusion that the "most likely" activity-time estimate m is practically useless. Its relative location in time interval $[a,b]$ is usually close to the point $(2a+b)/3$. In the course of the analysis a group of analysts was requested to determine subjectively (for a large number of activities selected from different projects) for optimistic a, pessimistic b and most likely m completion times. Afterwards two different samples of values m were compared by means of statistical testing – the one obtained from the analysts' subjective estimations, and the other by calculating $m^*=(2a+b)/3$. This experimentation has been repeated over and over again with the same result: the samples under comparison belonged to one and the same general population.

Table 2.1 presents a sample of 20 activities which have been selected from one of the R&D projects. The difference between the two samples of m versus m^* under comparison is not statistically significant (even with 0.01 level).

After the elimination of the "most likely" estimate, the additional improving suggestion would be assuming $\alpha=1$ and $\gamma=2$. Various statistical experiments [49] lead to the conclusion that these additional assumptions are, indeed, reasonable, since they simplify PERT analysis and do not change essentially the parameters of the network project. Thus, the general density distribution can be modified to

$$\varphi(t)=\frac{12}{(b-a)^4}(t-a)(b-t)^2 \qquad (2.3.18)$$

with the average, variance and mode as follows:

$$E(x)=0.2(3a+2b), \qquad (2.3.19)$$

$$V(x)=0.04(b-a)^2, \qquad (2.3.20)$$

$$m=(2a+b)/3. \qquad (2.3.21)$$

Table 2.1. Numerical values of m versus m^*

Activity	Value a	Value b	Value m	Value m^* (approximated to integer numbers)
	(determined by an analyst)			
1	16	21	18	18
2	20	45	30	28
3	10	25	16	15
4	6	13	8	8
5	20	35	24	25
6	12	20	15	15
7	15	27	18	19
8	3	9	5	5
9	18	28	22	21
10	10	15	13	12
11	25	40	28	30
12	24	40	30	29
13	25	50	30	33
14	30	60	35	40
15	20	35	24	25
16	20	50	25	30
17	15	40	22	23
18	30	55	40	38
19	30	65	38	42
20	7	13	10	9

Thus, the PERT statements can be replaced by a simpler methodology, since the analyst will be from now on asked to determine only two values – the optimistic and pessimistic activity-times. It goes without saying that for an individual activity there may be certain deviations in estimating the average and the variance when applying the three- and the two-value methodologies. But for *a project as a whole* there is in most cases no practical difference; i.e., for the project's main parameters there is no essential deviation. In practice, values a, m and b in PERT analysis are subjectively determined by the person responsible for the completion of the activity. He is usually not a specialist in mathematical statistics, as is stated in some PERT studies [49-50], and determining the most likely activity-time may become a real problem for him.

The PERT modification with two values has been introduced in many practical network-planning systems [49-54] which have functioned successfully for a long time.

The following conclusions can be drawn from the study:

a) The PERT analysis is efficient, and can be used in project management when each activity is carefully estimated by an experienced analyst. Otherwise, the two-value modification with the beta-distribution density function (2.3.18) and simple estimates (2.3.19-2.3.20) for the average and variance is preferable as being simple and not less efficient.

b) The commonly used subjective estimates of the average and variance in PERT analysis can be replaced by improved estimates (2.3.7-2.3.8) or (2.3.11-2.3.12). The latter provide better accuracy if the estimated mode is located in the tails of the distribution.

§2.4 A challenge against beta-density? A new approach to the activity-time distribution in PERT

2.4.1 Introduction

Problems associated with computing the density function of the completion time of PERT stochastic networks have been discussed extensively in scientific literature. Numerous publications refer mostly to three main directions.

The first one is associated with a direct and general solution of the problem of determining the completion-time distribution by sequential reduction of the initial network under various assumptions [45,103,107,119]. The activity network is partitioned into standard subnetworks of two different classes – activities in series and activities in parallel – each subnetwork being later reduced to an equivalent arc. The results obtained can be applied to small networks only. For large-size networks the solution boils down to interchanging the convolution and maximization integral operators, and is too complicated.

Various research has been undertaken to derive the upper and lower bounds of the completion-time distribution for discrete and continuous activity-time distributions [51,53,102-103]. In the same course, attempts have been made to derive bounds with better accuracy for normally distributed activity-times [103,119,135-138], but the results obtained are not far from those of Clark [22]. Although this direction still seems to be a promising one, it needs further, more successful, achievements.

The main shortcoming of both directions is the non-stability of the activity-time distribution with respect to convolution and maximization. We call activity-time distribution unstable with respect to convolution (maximization) if the sum (maximum) of two independent activity-times has another distribution. Unfortunately, the beta-distribution, which is generally superior to other activity-time distributions in project planning, is unstable with respect to both convolution and maximization. As a result, a new direction has appeared: various research has been carried out either to replace the beta-distribution by another one [8,102,162-163], or to explore the errors involved in approximation of both the maximum and sum of two independent random beta variables, each by another beta variable [51,76].

In our opinion, replacing the beta-distribution by the normal one (which is stable with respect to convolution) contradicts the basic principles of PERT analysis. As outlined in Section 2.1, activity-time distribution, unlike the normal one, is asymmetrical for many reasons [49,51,53-54,60]. Since the activity time is always positive, whoever wishes to use normal distribution has to apply it in the positive area only. However, the transformed distribution becomes unstable with respect to convolution.

As to beta-distribution, the error involved in assuming that the maximum of two beta variables is also a beta variable can be indicated by the Kolmogorov-Smirnov one-sample test [75,96]. Various experiments show [75] that the closeness of the approximation depends significantly on the ratio r of the variable ranges. If inequality $0.4 \le r \le 2.5$ does not hold, the approximation should be rejected. Otherwise the error measure D_N, fits the test with confidence probability close to one. However, even $5 \div 10$ sequential maximizations may pile up the approximation error to a substantial value. Thus, beta-distribution can be regarded stable to maximization for small networks only.

In this section we present an asymmetric activity-time distribution which is close to the beta-distribution. That distribution is stable with respect to maximization, and is close to stable with respect to convolution [60].

2.4.2 <u>Stable distributions with respect to convolution</u>

By definition, a cumulative probability distribution function (c.d.f.) $F(x)$ is regarded stable with respect to convolution if for any $a_1, a_2 > 0$, b_1, b_2 there exist values $a_3 > 0$, b_3 such that, for all x, relation

$$F(a_1 x + b_1) * F(a_2 x + b_2) = F(a_3 x + b_3) \qquad (2.4.1)$$

holds, $(*)$ being the convolution operator.

It can be shown [51,60,76] that if and only if the distribution is stable with respect to convolution, relation

$$\ln \phi(t) = i\gamma t - C|t|^{\alpha}\left\{1 + i\beta \frac{t}{|t|}\omega(t,\alpha)\right\} \qquad (2.4.2)$$

holds, $\phi(t)$ being the characteristic function of the probability density function (p.d.f.) $f(x)$, α, β, γ, C constants – namely, $0 < \alpha \le 2$, $-1 \le \beta \le 1$, $C \ge 0$, γ any real value - and

$$\omega(t,\alpha) = \begin{cases} tg\dfrac{\pi}{2}\alpha \ \ if \ \ \alpha \ne 1 \\ \dfrac{2}{\pi}\ln t \ \ otherwise. \end{cases}$$

Value $\alpha = 2$ corresponds to the normal distribution, values $\alpha = 1$, $\beta = 0$ to the Cauchy distribution, values $\alpha = 0.5$, $\beta = 1$, $\gamma = 0$ and $C = 1$ to the p.d.f.

$$f(x) = \frac{1}{\sqrt{2\pi}\,x^{3/2}} \cdot \exp\left\{-\frac{1}{2x}\right\}, \quad x > 0. \qquad (2.4.3)$$

Denote henceforth the set of stable distributions with respect to convolution by D_α, where α is the characteristic index in (2.4.2).

<u>Theorem 1</u>

When $x \to \infty$, the p.d.f. of D_α is asymptotically close to $\frac{B(\alpha,\beta)}{x^{\alpha+1}}$, $-1 \le \beta \le 1$, $0 < \alpha \le 2$, where B is independent of x.

Theorem 1 is proved in [51] by examining different cases of the p.d.f. of D_α in the distribution tails.

2.4.3 Stable distributions with respect to maximization

By definition, a c.d.f. $F(x)$ is regarded stable with respect to maximization if for any $a_1, a_2 > 0$, there exists a value $a_3 > 0$ such that, for all $x \geq 0$, relation

$$F(a_1 x)F(a_2 x) = F(a_3 x) \tag{2.4.4}$$

holds.

Theorem 2

The c.d.f.

$$F_\upsilon(x) = \begin{cases} \exp\left[-\left(\dfrac{\theta}{x}\right)^\upsilon\right], \ \theta, \upsilon > 0 \ \mathit{if} \ x > 0 \\ 0 \qquad \mathit{otherwise} \end{cases} \tag{2.4.5}$$

is stable with respect to maximization on the set of distributions with identical υ.

Proof

The proof is obtained by substituting (2.4.5) in (2.4.4).

Note that mode m satisfies

$$m = \theta\left(\frac{\upsilon}{\upsilon+1}\right)^{1/\upsilon}. \tag{2.4.6}$$

For $\upsilon > 1$ average $E(x)$ satisfies

$$\begin{cases} E(x) = \theta\,\Gamma\left(\dfrac{\upsilon-1}{\upsilon}\right), \\ \Gamma(x) = \displaystyle\int_0^\infty a^{x-1}e^{-a}\,da. \end{cases} \tag{2.4.7}$$

When maximizing n random variables with parameters $\theta_1, \theta_2, \ldots, \theta_n$, respectively, we obtain the same distribution with parameters υ and $\theta_{\max} = \left(\sum_{i=1}^{n}\theta_i^\upsilon\right)^{1/\upsilon}$. Distribution F_υ can be easily restricted from below, namely

$$F_\upsilon(x) = \begin{cases} \exp\left[-\left(\dfrac{\theta}{x-a}\right)^\upsilon\right] \quad \mathit{for} \ x > a > 0 \\ 0 \qquad \mathit{otherwise}\,. \end{cases} \tag{2.4.8}$$

Denote henceforth the set of stable distributions with respect to maximization by D_υ.

Theorem 3

As $x \to \infty$, the p.d.f. of D_υ is asymptotically close to $\frac{A}{x^{\upsilon+1}}$, $A = \upsilon\theta^\upsilon$ being a constant independent of x.

Proof

The proof is obtained from examining the p.d.f.

$$f_\upsilon(x)=\begin{cases}\dfrac{\upsilon\theta^{\upsilon}}{x^{\upsilon+1}}\exp\left[-\left(\dfrac{\theta}{x}\right)^{\upsilon}\right] & for\ x>0\\ 0 & otherwise\,.\end{cases} \tag{2.4.9}$$

Corollary

As $x\to\infty$, the p.d.f. of D_α is asymptotically close to the p.d.f. of D_υ with values $0<\alpha=\upsilon<2$.

The proof is obvious. A conclusion can be drawn that in the distribution tails the p.d.f. of D_α can be regarded as a stable one with respect to maximization, and *vice versa*. This is important, especially as most of the approximation errors appear just in the distribution tails.

2.4.4 Experimentation

In order to examine the closeness of the beta-distribution to the D_υ distribution, various examples were run. Since any beta-distribution can be transformed to a standard p.d.f.

$$f(x)=\frac{\Gamma(m+n+2)}{\Gamma(m+1)\Gamma(n+1)}x^m(1-x)^n,\quad 0<x<1,\quad m,n>-1, \tag{2.4.10}$$

we examined the closeness between two standard distributions (2.4.9) and (2.4.10). This can be achieved [51,53,76] by equating the $(1-p)$-th quantile, $p<<1$, of the D_υ distribution (2.4.9) with the upper level of the standard beta-distribution. Later on, values θ and υ are determined, such that the main parameters (the average, variance, and the most likely values) of both distributions under comparison are close to each other, respectively. For example, the p.d.f. (2.4.10) with $m=1$, $n=2$, which is often used in project planning [49], is close to the D_υ distribution with $\upsilon=2$ and $\theta\cong 0.32$ $(p=0.1)$; moreover, the main parameters practically coincide.

Note that the set of D_υ distributions (2.4.9) is preferable for practical applications since it corresponds to a broad spectrum of beta-distributions and has simple formulae for calculating the average and most likely values.

2.4.5 Conclusions

We consider the problem of computing the density function of the completion time of PERT stochastic networks. An activity-time distribution is suggested which is stable to maximization and close to the beta-distribution. Thus classical network-reducing methods, e.g., Martin's algorithm [119], can be essentially simplified.

Based on the results outlined above, we recommend using the c.d.f. (2.4.5) for activity-time distribution purposes in stochastic PERT networks. Index θ varies from activity to activity, while parameter υ which may be interpreted as the uncertainty level for the project as a whole, is kept constant. From the discussion outlined above, it follows that value $\upsilon=2$ is preferable.

A subnetwork Σ_1 of n independent activities (i, j) in parallel is thus reduced to an equivalent arc with parameters $\upsilon_{\Sigma_1} = \upsilon$ and $\theta_{\Sigma_1} = \left[\sum_{(i,j)\in\Sigma_1} \theta_{ij}^{\upsilon} \right]^{1/\upsilon}$. From (2.4.7) we obtain that a subnetwork Σ_2 of n activities in series can be reduced to one arc with average

$$E(\Sigma_2) = \sum_{(i,j)\in\Sigma_2} E(i, j) = \Gamma\left(\frac{\upsilon - 1}{\upsilon}\right) \sum_{(i,j)\in\Sigma_2} \theta_{ij} . \qquad (2.4.11)$$

It follows from the corollary that the c.d.f. $D_{\upsilon=2}$, being stable with respect to maximization, can be regarded as a stable one with respect to convolution too. An additional reason for this approximation can be obtained by examining relations (2.4.7) and (2.4.11) with similar structure. Distribution $D_{\upsilon=2}$ fits, especially, when the analyst estimates the activity time by *one value*, namely by the most likely time m. In this case the corresponding value θ is immediately determined by (2.4.6), parameter υ being externally pregiven. It goes without saying that when each activity is carefully examined and later on estimated by an experienced analyst, the three-value PERT analysis is efficient and can be implemented in project management. But for entirely new innovative R&D projects including activities with no similar prototypes in the past, the estimates of the pessimistic and optimistic duration times are usually rather poor [49,164], e.g., pessimistic assessments are chosen conservatively large, etc. Under such circumstances PERT analysis estimates may misrepresent the real activity distribution, and the one-value estimate m is more simple but not less efficient. Since $F_\upsilon(x)$ can be easily transformed to distribution range $a < x < \infty$, the three-value activity-time estimates (a, b and m) can also be used when necessary. Note that when using $F_\upsilon(x)$, all the network-reducing methods, e.g., Martin's algorithm [119], are essentially simplified since only one variable parameter, θ, is utilized.

Chapter 3. Estimating Parameters of Stochastic Network Models

§3.1 New concepts in stochastic network models' parameters

In numerous books (see, e.g., [48-54,121-122,130]) a variety of algorithms to calculate the network's critical path length by means of simulation, has been outlined. Simulation methods can be used for calculating other network parameters, which determine the level of intensity for both activities and paths forming a part of a network with random activity durations. In [49] we have introduced the concept of the p*-quantile intensity level of path* L which is calculated as follows

$$k_{p.\text{int}}(L) = W_p\{k_{\text{int}}(L)\}, \tag{3.1.1}$$

where $k_{\text{int}}(L)$ determines the intensity level of path L for a fully deterministic network model calculated by

$$k_{\text{int}}(L) = \frac{t(L) - t^*_{cr}(L)}{T_{cr} - t^*_{cr}(L)}. \tag{3.1.2}$$

Here $t(L)$ denotes the length of path L connecting the network's source and terminal nodes, while $t^*_{cr}(L)$ denotes the summarized duration of activities entering both path L and the critical path L_{cr} with length T_{cr}. To determine value $W_p\{k_{\text{int}}(L)\}$ one should undertake N multiple simulation runs for all activities entering the considered stochastic network model. In the course of each simulation run the deterministic value $k_{\text{int}}(L)$ has to be calculated. The p-quantile intensity value for *any activity* (i,j) entering the stochastic network can be determined by

$$k_{p.\text{int}}(i,j) = W_p\{k_{\text{int}}(i,j)\}, \tag{3.1.3}$$

where intensity value $k_{p.\text{int}}(i,j)$ for activity (i,j) *entering a deterministic network* has to be calculated [48-49,53] by

$$k_{\text{int}}(i,j) = \frac{t[L(i,j)_{\max}] - t^*_{cr}(i,j)_{\max}}{t^{**}_{cr}(i,j)_{\max}}. \tag{3.1.4}$$

Here $t^*_{cr}(i,j)_{\max}$ and $t^{**}_{cr}(i,j)_{\max}$ denote, correspondingly, the summarized durations of all activities coinciding and not coinciding with the *maximal* path connecting the network's source and terminal nodes, and passing through activity (i,j).

In the same way, another level of intensity - the so-called p-quantile reserve level [48-49,53] - can be determined by

$$k_{p.res}(i,j) = W_p\{k_{res}(i,j)\}, \tag{3.1.5}$$

where $k_{res}(i,j)$ has to be calculated for a network *with deterministic activity durations* $t(i,j)$ [49,53]

$$k_{res}(i,j) = \frac{t_{ear}(j) - t_{lat}(i)}{t(i,j)}, \tag{3.1.6}$$

values $t_{ear}(j)$ and $t_{lat}(i)$ being the earliest and the latest moments for realizing events (nodes) j and i, correspondingly.

Implementing the p-quantile concepts enables subdividing the entire set of activities entering a stochastic network, into three parts - the *critical*, the *intermediate* and the *reserve* zones. We suggest allocating to the critical zone all activities (i,j) with values $W_p\{k_{\text{int}}(i,j)\}$ exceeding the preset level $1-\eta$ $(\eta > 0)$. Note that increasing value η results in increasing the volume of the critical zone. This fully corresponds to the concept of confidence level [20] which is widely implemented in mathematical statistics. Thus, practically speaking, using network analysis models with random activity durations leads to introducing new concepts in planning and controlling network projects, namely, defining p-quantile confidence zones as follows:

a) p*-quantile critical zone* comprises activities (i,j) with $W_p\{k_{\text{int}}(i,j)\} \geq p_1$ (in real design offices $p_1 \approx 0.8 \div 0.9$);

b) p*-quantile reserve zone* unifies activities satisfying $W_p\{k_{\text{int}}(i,j)\} \leq p_2$ (in practice $p_2 \approx 0.2 \div 0.3$);

c) p*-quantile intermediate zone* comprising the remaining activities (i,j) satisfying $p_2 < W_p\{k_{\text{int}}(i,j)\} < p_1$.

It can be well-recognized that the easiest means of determining the regarded confidence zones for any network with random activity durations and pregiven values p_1 and p_2 is to undertake multiple simulation runs by the Monte-Carlo method, i.e., by simulating activity durations with p.d.f. (2.2.1) (or any other probability law). In the course of each simulation run deterministic values $k_{\text{int}}(i,j)$ (using (3.1.4)) have to be calculated for each activity (i,j) entering the network. After carrying out N simulation runs statistical empirical frequencies for the N-amount sample for each activity to fall into one of the three zones have to be calculated. Note that usually such a frequency is determined by $\overline{p}_{ij} = \frac{N_{ij}}{N}$, where N_{ij} denotes the number of cases (from N runs) when value $k_{\text{int}}(i,j)$, being calculated by (3.1.4), is either less than p_2 (reserve zone), or exceeds p_1 (critical zone) or belongs to the intermediate zone. For any (i,j), by means of the statistical hypothesis theory [25], we can compare values $\overline{p}_{ij}$, p_1 and p_2 in order to determine as to what zone does activity (i,j) belong. Thus, if N is sufficiently large to provide representative statistics, values $\overline{p}_{ij}$ form the three confidence zones.

Similar to intensity estimates (3.1.1-3.1.4), new conceptions based on calculating the p-quantiles for reserve estimates, have to be implemented in NAM with stochastic activity durations. Besides estimates (3.1.5-3.1.6), we suggest using [49,53] the p-quantile of a *full time reserve* for each activity (i,j) calculated by $W_p\{R_{full}(i,j)\}$, where the full time reserve $R_{full}(i,j)$ for a deterministic network (being simulated each time by means of a routine simulation run) satisfies

$$R_{full}(i,j) = t_{lat}(j) - t_{ear}(i) - t(i,j). \qquad (3.1.7)$$

Such an estimate is used in various books on network planning [49,53,149]. As to introducing the concept of p-quantile of a *free reserve* for activity (i,j), the latter may, correspondingly, be calculated by

$$W_p\{R_{free}(i,j)\} = W_p\{t_{ear}(j) - t_{lat}(i) - t(i,j)\}. \qquad (3.1.8)$$

Finally, the p-quantile of *time reserve for path* L may be determined as

$$W_p\{R(L)\} = W_p\{T_{cr} - t(L)\}. \qquad (3.1.9)$$

Note that increasing confidence probability p results always in decreasing the corresponding p-quantiles of time reserves $R(i,j)$ and $R(L)$. This, in turn, averts unreasonable time resource transmissions between network activities which may cause unstable equilibrium situations.

Another concept of estimating time reserves from a probabilistic point of view is as follows [49,53]: denote $t_{pl}(i,j)$ the planned time to execute activity (i,j). Estimate

$$W_p(i,j) = W_p\{t_{ear}(j)\} - W_p\{t_{ear}(i)\} - t_{pl}(i,j) \qquad (3.1.10)$$

can be used in various NAM models [49,53] as a pregiven due date for starting activity (i,j), i.e., as a milestone. Under certain circumstances relation (3.1.10) may represent a realistic physical meaning.

§3.2 Estimating the accuracy of probability network parameters by means of simulation

Consider the accuracy of estimating one of the network parameters by means of simulation for a fixed and pregiven number N of simulation runs, namely, by estimating the earliest moment of event's k realization. Let $t_i(k)$ be the earliest moment for event k to be realized in the i-th simulation run. The sample under consideration comprises the set $\{t_i(k)\}$, $1 \le i \le N$. A random value

$$\bar{t}(k) = \frac{1}{N}\sum_{i=1}^{N} t_i(k) \qquad (3.2.1)$$

is the estimate of average $E\{t(k)\}$ of the earliest moment when event (node) k is realized. Variance $V\{t(k)\}$ of random value $t(k)$ has an estimate

$$S^2[t(k)] = \frac{1}{N}\sum_{i=1}^{N}\left[t_i(k) - \bar{t}(k)\right]^2. \qquad (3.2.2)$$

Since $t_i(k)$ are independent and $E[t_i(k)] \equiv E[t(k)]$, $V[t_i(k)] \equiv V[t(k)]$ hold, we obtain

$$E\left\{\frac{1}{N}\sum_{i=1}^{N}[t_i(k)]\right\} = E[t(k)],$$

$$V\{\bar{t}(k)\} = \frac{1}{N}V[t(k)] = \frac{\sigma^2[t(k)]}{N}.$$

In accordance with the Lyapunov theorem [27] relations

$$
\begin{cases}
P\left\{\dfrac{\bar{t}(k)-E[t(k)]}{\dfrac{\sigma[t(k)]}{\sqrt{N}}}<z\right\}_{N\to\infty} \to \dfrac{1}{2}+\dfrac{1}{\sqrt{2\pi}}\displaystyle\int_0^z \exp\left\{-\dfrac{t^2}{2}\right\}dt\,, \\
P\left\{\dfrac{\bar{t}(k)-E[t(k)]}{\dfrac{S[t(k)]}{\sqrt{N}}}<z\right\}_{N\to\infty} \to \dfrac{1}{2}+\dfrac{1}{\sqrt{2\pi}}\displaystyle\int_0^z \exp\left\{-\dfrac{t^2}{2}\right\}dt\,.
\end{cases}
\tag{3.2.3}
$$

hold. Thus, for sufficiently large N relation

$$
P\left\{\left|\bar{t}(k)-E[t(k)]\right|\le \frac{z\cdot S[t(k)]}{\sqrt{N}}\right\}\approx 2\Phi(z)
\tag{3.2.4}
$$

holds, where

$$
\Phi(z)=\frac{1}{\sqrt{2\pi}}\int_0^z \exp\left\{-\frac{t^2}{2}\right\}dt,\quad z>0.
$$

For a preset probability value α, from $2\Phi(z_\alpha)=\alpha$, one can determine value z_α satisfying

$$
P\left\{\bar{t}(k)-z_\alpha\cdot\frac{S[t(k)]}{\sqrt{N}}<E[t(k)]<\bar{t}(k)+z_\alpha\cdot\frac{S[t(k)]}{\sqrt{N}}\right\}=\alpha\,.
\tag{3.2.5}
$$

Estimate the accuracy of standard deviation S as an approximate value of $\sigma[t(k)]$. Analyzing

$$
E\{t_i(k)-E[t(k)]\}^2=E\{t(k)-E[t(k)]\}^2=V[t(k)]=\sigma^2[t(k)],
$$

$$
V\{t_i(k)-E[t(k)]\}^2=V\{t(k)-E[t(k)]\}^2=\mu_4-\mu_2^2=\sigma_2^2\,,
$$

and taking into account

$$
S_N^2=\frac{1}{N}\sum_{i=1}^{N}\left\{t_i(k)-E[t(k)]\right\}^2,
$$

we obtain from the Central Limit Theorem [27]

$$
P\left\{z_0<\frac{S_N^2-\sigma^2[t(k)]}{\dfrac{\sigma_2}{\sqrt{N}}}<z_1\right\}_{N\to\infty}\to\frac{1}{\sqrt{2\pi}}\int_{z_0}^{z_1}\exp\left\{-\frac{t^2}{2}\right\}dt=\Phi(z_1)-\Phi(z_0).
\tag{3.2.6}
$$

This means that value $\dfrac{\{S_N^2-\sigma^2[t(k)]\}\sqrt{N}}{\sigma_2}$ for large N is approximately distributed with the normal p.d.f. Taking into account that $S_N^2-S^2=\{\bar{t}(k)-E[t(k)]\}^2$, we obtain

$$
\sqrt{N}\left(S_N^2-S^2\right)=\sqrt{N}\left\{\bar{t}(k)-E[t(k)]\right\}^2.
\tag{3.2.7}
$$

From the other side, any ε as small as possible satisfies [27]

$$
\lim_{N\to\infty}P\left\{\left|\bar{t}(k)-t(k)\right|>\frac{\varepsilon}{\sqrt[4]{N}}\right\}=0\,,
\tag{3.2.8}
$$

since Chebyshev's inequality enables

$$P\{|\xi - E[\xi]| < \varepsilon\} \le \frac{1}{\varepsilon^2} V[\xi].$$

For sufficiently large N we obtain

$$P\left\{\left|\bar{t}(k) - E[t(k)]\right| > \frac{\varepsilon}{\sqrt[4]{N}}\right\} < \frac{\sqrt{N}}{\varepsilon^2} \cdot V\left[\bar{t}(k)\right] = \frac{\sigma^2[t(k)]}{\sqrt{N} \cdot \varepsilon^2},$$

which, in turn, enables relation (3.2.8).

Thus, $\lim\limits_{N\to\infty}{}^{prob} \sqrt[4]{N}\left\{\left|\bar{t}(k) - E[t(k)]\right|\right\} \to 0$ holds, and we obtain probability convergence $\lim\limits_{N\to\infty}{}^{prob} \sqrt{N}\left\{\bar{t}(k) - E[t(k)]\right\}^2 = 0$. From analyzing (3.2.6) and (3.2.7), one may conclude that values $\sqrt{N}S_N^2$ and $\sqrt{N}S^2$ for sufficiently large N are asymptotically normally distributed, i.e.,

$$\lim_{N\to\infty} P\left\{ z_0 < \frac{S^2 - \sigma^2[t(k)]}{\frac{\sigma_2}{\sqrt{N}}} < z_1 \right\} = \frac{1}{\sqrt{2\pi}} \int_{z_0}^{z_1} \exp\left\{-\frac{t^2}{2}\right\} dt \tag{3.2.9}$$

holds. Taking into account that $\sigma_2^2 = \mu_4 - \mu_2^2$, we arrive at the conclusion that S^2 is normally distributed with parameters $\left[\sigma^2[t(k)], \frac{1}{N}\left(\mu_4 - \mu_2^2\right)\right]$. Thus, parameter S^2 can be regarded as an unbiased and sustainable estimate for variance $\sigma^2[t(k)]$. On the basis of probability convergence $\lim\limits_{N\to\infty}{}^{prob} \frac{S}{\sigma[t(k)]} = 1$, we finally obtain

$$\lim_{N\to\infty} P\left\{ z_0 < \frac{S - \sigma[t(k)]}{\frac{\sigma_2}{2\sigma[t(k)]\sqrt{N}}} < z_1 \right\} = \frac{1}{\sqrt{2\pi}} \int_{z_0}^{z_1} \exp\left\{-\frac{t^2}{2}\right\} dt \,, \tag{3.2.10}$$

and value S for a sufficiently large N is asymptotically normally distributed, with parameters $\left[\sigma[t(k)], \frac{\sigma_2}{2\sigma[t(k)]\sqrt{N}}\right]$.

To estimate value σ_2 by substituting central moments μ_4 and μ_2 for their sample estimates m_4 and m_2, we finally obtain the approximate equality

$$\sigma_2 \approx S_2 = \sqrt{\frac{\sum_{i=1}^{N}\left[t_i(k) - \bar{t}(k)\right]^4}{N} - \left[\frac{1}{N}\sum_{i=1}^{N}\left[t_i(k) - \bar{t}(k)\right]^2\right]^2}$$

or, taking into account (3.2.2),

$$\sigma_2 \approx S_2 = \sqrt{\frac{1}{N}\left\{\left[\sum_{i=1}^{N} t_i(k) - \bar{t}(k)\right]^4\right\} - S^4[t(k)]}\,. \tag{3.2.11}$$

It can be well-recognized [27] that if a random value ξ is normally distributed, its central moments satisfy $\sqrt{\mu_4 - \mu_2^2} \approx \sqrt{2}\sigma^2$. This enables the approximate equality

$$\sigma_S = \frac{\sigma_2}{2\sigma[t(k)]\sqrt{N}} = \frac{\sqrt{\mu_4 - \mu_2^2}}{2\sigma[t(k)]\sqrt{N}} \approx \frac{\sigma^2\sqrt{2}}{2\sigma\sqrt{N}} = \frac{\sigma}{\sqrt{2}\sqrt{N}}. \tag{3.2.12}$$

Estimates (3.2.3) or (3.2.5), (3.2.10) or (3.2.12) enable calculation of the proper sample amount N (number of simulation runs) required to ensure estimations $E[t(k)] \approx \bar{t}(k)$ and $\sigma^2[t(k)] \approx S^2[t(k)]$ with predetermined accuracy, correspondingly. To determine N for the case of estimating the average value one has to preset the confidence level α and the accuracy of substituting $E[t(k)] \approx \bar{t}(k)$, i.e., inequality

$$P\left\{ \left| \bar{t}(k) - E[t(k)] \right| < \varepsilon \right\} \geq \alpha \tag{3.2.13}$$

should hold. By using (3.2.3) and inequality $z_\alpha \cdot \frac{S[t(k)]}{\sqrt{N}} \leq \varepsilon$, we finally obtain

$$N \geq \frac{z_\alpha^2 \sigma^2[t(k)]}{\varepsilon^2} = \frac{\Phi^{-1}\left(\frac{\alpha}{2}\right) \cdot \sigma^2[t(k)]}{\varepsilon^2}. \tag{3.2.14}$$

By representing the limit error as a quota of the standard deviation value $q = \frac{\varepsilon}{\sigma[t(k)]}$, (3.2.14) may be transferred to a simplified form

$$N \geq \Phi^{-1}\left(\frac{\alpha}{2}\right) \Big/ q^2. \tag{3.2.15}$$

It can be well-recognized that from similar considerations, (3.2.14) may be substituted also for

$$N \geq \frac{\Phi^{-1}\left(\frac{\alpha}{2}\right) \cdot S^2[t(k)]}{\varepsilon^2} = \Phi^{-1}\left(\frac{\alpha}{2}\right) \Big/ q_1^2, \tag{3.2.16}$$

where $q_1 = \frac{\varepsilon}{S[t(k)]}$.

Similarly, the required number of simulation runs may be assessed by means of (3.2.10), (3.2.12). Indeed, from

$$P\left\{ |S - \sigma| \leq z_\alpha \frac{\sigma}{\sqrt{N}} \right\} = 2\Phi(z_\alpha) \geq \alpha$$

in conjunction with inequality $\frac{z_\alpha \cdot \sigma}{\sqrt{2N}} \leq \varepsilon_\sigma$, we obtain the restricted from below sample value N estimate

$$N \geq \frac{z_\alpha^2 \sigma^2}{\varepsilon_\sigma^2 \cdot 2} = \frac{\Phi^{-1}\left(\frac{\alpha}{2}\right) \cdot \sigma^2[t(k)]}{2\varepsilon_\sigma^2} \tag{3.2.17}$$

or

$$N \geq \Phi^{-1}\left(\frac{\alpha}{2}\right) \Big/ 2q_\sigma^2, \quad q_\sigma = \frac{\varepsilon_\sigma}{\sigma}. \tag{3.2.18}$$

Thus, we have considered the problem of estimating the number N of simulation runs to be able deciding with required confidence as to whatever p-quantile zone each activity (i,j) has to be allocated. As outlined in §3.1, two probabilities p_1 and p_2, $p_2 < p_1$, are pregiven. In case $\overline{p}_{ij} \ge p_1$ activity (i,j) belongs to the critical zone, when $\overline{p}_{ij} < p_2$ activity (i,j) has to be allocated to the reserve zone. For $p_1 > \overline{p}_{ij} \ge p_2$ activity (i,j) refers to the intermediate zone. Although put in easy terms, the appropriate decision-making when grounded on solid statistical theory [27], becomes not that obvious. The main reason for that stems from the fact that in allocating activities (i,j) to the three zones, we substitute the theoretical probability p_{ij} by its simulated statistical frequency $\overline{p}_{ij} = N_{ij}/N$ (see §3.1). To avoid statistical inconsistencies in such a procedure, consider the regarded decision-making in greater detail.

In terms of theoretical probability p_{ij}, assume $p_{ij} > p_1$, i.e., the appropriate activity (i,j) belongs to the critical zone. With the help of the well-known de Moivre-Laplace Theorem [27], we obtain for this case

$$P\left\{\frac{N_{ij}}{N} < p_1 - k\sqrt{\frac{p_1 \cdot (1-p_1)}{N}}\right\} < 1 - \frac{1}{\sqrt{2\pi}}\int_{-k}^{\infty} \exp\left\{-\frac{t^2}{2}\right\} dt = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-k} \exp\left\{-\frac{t^2}{2}\right\} dt . \quad (3.2.19)$$

Let us preset the confidence level α, i.e., the probability of the fact that for the theoretically true event (namely, $p_{ij} > p_1$) we will not take a false decision on the basis of the simulated statistical frequency $\overline{p}_{ij} = N_{ij}/N$ and will not wrongly allocate activity (i,j) to the second or third zones (intermediate or reserve). For the sake of determinacy, assume $\alpha = 0.95$.

From relation $P\left\{\frac{N_{ij}}{N} < p_1 - k\sqrt{\frac{p_1 \cdot (1-p_1)}{N}}\right\} < 1 - \alpha$ value k may be singled out by means of

$$k = F^{-1}(\alpha); \quad F(k) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{k} \exp\left\{-\frac{t^2}{2}\right\} dt = \alpha . \quad (3.2.20)$$

For $\alpha = 0.95$ we obtain $k = 1.65$. Thus, the difference between $\overline{p}_{ij}$ and p_1 with probability α should not exceed

$$F^{-1}(\alpha) \cdot \sqrt{\frac{p_1 \cdot (1-p_1)}{N}} = \varepsilon . \quad (3.2.21)$$

After calculating the deviation value ε, we are able to determine the required number of simulation runs N by

$$N \ge \frac{p_1 \cdot (1-p_1)}{\varepsilon^2}\left[F^{-1}(\alpha)\right]^2 . \quad (3.2.22)$$

Thus, after fixing the confidence level α, we determine the deviation limit ε. The latter means that after carrying out N simulation runs, with N satisfying (3.2.22), empirical frequency $\overline{p}_{ij} = N_{ij}/N$ should belong to interval $[p_1 - \varepsilon; 1]$ (for the case of $p_{ij} > p_1$). In other words, to check inequality $p_{ij} > p_1$, one has to carry out N simulations and later on to compare the calculated value of $\overline{p}_{ij}$ with $p_1 - \varepsilon$. If $\overline{p}_{ij} < p_1 - \varepsilon$ the conclusion should be drawn that relation $p_{ij} > p_1$ is false and activity (i, j) should not be placed in the critical zone.

All other hypothesis referring to other activities and different zones, have to be checked in a way similar to that outlined above.

§3.3 Simulating stochastic network models by means of equivalent transformations

We will outline several algorithms of simulating stochastic networks to determine their critical path length (or, more exact, a statistical analogue of the critical path distribution). This is achieved by constructing a transformed network of lower size but with equivalent probability distribution parameters. We will henceforth call two stochastic networks equivalent ones if their parameters' p.d.f. practically coincide. Creating an equivalent network model of lower size enables simulating the latter by lower computational time.

Note that the problem of transferring the initial network to one of a smaller size, but with similar p.d.f. parameters, has been a common research area for many scientists (e.g., [35]). We have not been an exclusion, although all the results obtained can be used for small- and medium-size networks only. We will present an alternative approach outlined in [49].

To obtain an equivalent network we have to exclude from the initial one all the activities which have no influence on the statistical parameters of the network as a whole. Two methods to develop equivalent stochastic networks will be outlined:

1. Analytical method.
2. Method based on simulation modeling.

The first method is based on singling out the subset of activities which theoretically cannot belong to a critical path in the course of a routine simulation run. The following theorems [49] are at the underlay of the analytical method:

Theorem I. Given a stochastic network model with activities (i, j), each with a random duration beta-distributed in the interval $[a(i, j); b(i, j)]$ (all positive values $a(i, j)$ and $b(i, j)$ pregiven). All m paths connecting the source and the terminal events, are enumerated and denoted $L_1, L_2, ..., L_m$. Let N simulation runs be carried out, and denote:

- $t(L_k^i)$ - the duration of path L_k in the i-th simulation run;
- $t(L_{cr})_a$ - the critical path length of the deterministic network model with set $t(i, j) = a(i, j)$.

The theorem asserts that for each one of N simulation runs there exists at least one path L_ξ satisfying $t(L_\xi^i) > t(L_{cr})_a$; here i stands for the number of the simulation run.

Theorem II. For each simulation run the deterministic critical path length is not less than $t(L_{cr})_a$.

<u>Theorem III</u>. Activity (i,j) belongs to the subsets of possible critical paths, i.e., has a probability exceeding zero to be found on the critical path in the course of a routine simulation, if $t[L_{ij}^{\max}]_b \geq t(L_{cr})_a$ holds. Here $t[L_{ij}^{\max}]_b$ denotes the length of the maximal path comprising (i,j), for a deterministic network with $b(i,j) \Rightarrow t(i,j)$.

Denote the subset of possible critical paths in the stochastic network under consideration, by Q. Thus, it can be well-recognized that a network with structure Q is equivalent to the initial network.

<u>Theorem IV</u>. Activity (i,j) belongs to set Q if its full reserve $R_{full}(i,j)$ calculated as

$$\begin{cases} R_{full}(i,j) = t_{lat}(j) - t_{ear}(i) - t(i,j) \\ t(i,j) = b(i,j) \end{cases} \tag{3.3.1}$$

for a deterministic network, is less than $t(L_{cr})_b - t(L_{cr})_a$.

The outlined above theorems [49] enable developing the following step-wise algorithm to construct an equivalent network model:

<u>Step 1</u>. Set $t(i,j) = a(i,j)$ for all activities (i,j).

<u>Step 2</u>. Calculate the critical path length $t(L_{cr})_a$.

<u>Step 3</u>. Set $t(i,j) = b(i,j)$ for all activities (i,j).

<u>Step 4</u>. Calculate the critical path length $t(L_{cr})_b$.

<u>Step 5</u>. Using (3.3.1), calculate full time reserves $R_{full}(i,j)$ for all activities (i,j) entering the network.

<u>Step 6</u>. If $R_{full}(i,j) \geq t(L_{cr})_b - t(L_{cr})_a$ holds, exclude activity (i,j) from set Q of activities comprising the equivalent network.

<u>Step 7</u>. In case $R_{full}(i,j) < t(L_{cr})_b - t(L_{cr})_a$ include activity (i,j) into set Q.

Note that implementing the above algorithm does not necessarily result in essential reducing the volume of the initial network. To further improve the procedure, a refined algorithm [49] can be suggested. By the amended procedure, activity (i,j) should belong to set Q on condition that in the course of a routine simulation run the maximal path in the deterministic network connecting the source and the terminal nodes, and comprising activity (i,j), is the network's critical path with probability not less than $\varepsilon > 0$.

In other words,

$$P\{t(L_{ij}^{\max}) = t_{cr}\} \geq \varepsilon > 0 \tag{3.3.2}$$

holds.

Denote the average duration of path $L_{ij}^{\max}$ comprising activity (i,j) by $E\left\{t[L_{ij}^{\max}]\right\}$ and its variance - by $\sigma^2\left\{t[L_{ij}^{\max}]\right\}$. Several assumptions are implemented in the modified algorithm:

1. The length of any path L is normally distributed with average $\bar{t} = \sum_{(i,j)\in L} \bar{t}(i,j)$ and variance $\sigma_t^2 = \sum_{(i,j)\in L} \sigma^2[t(i,j)]$;
2. The p.d.f. of the critical path is similar to the p.d.f. of the maximum of all paths entering the network.

Thus, the network can be regarded as a unification of two paths as follows:

a) the critical path L_{cr};

b) the maximal path $L_{ij}^{\max}$ comprising activity (i,j).

It is shown [49] that activity (i,j) can be included in the equivalent network Q if

$$R_{full}(i,j) \le \sigma[R_{full}(i,j)]\sqrt{2}\Phi^{-1}(1-2\varepsilon), \tag{3.3.3}$$

where $\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_0^z \exp\left\{-\frac{t^2}{2}\right\}dt, \quad z > 0.$

The step-wise procedure of the *modified algorithm* may be outlined as follows:

Step 1. Determine all activities belonging to the critical path, for the case $t(i,j) = \bar{t}(i,j)$.

Step 2. Calculate variance $V\{t(L_{cr})\} = \sum_{(i,j)\in L_{cr}} \sigma^2[t(i,j)]$.

Step 3. Determine all activities belonging to the maximal path $L_{ij}^{\max}$, comprising activity (i,j), for Step 1, i.e., for $t(i,j) = \bar{t}(i,j)$.

Step 4. Calculate variance $V\{t(L_{ij}^{\max})\} = \sum_{(i,j)\in L_{ij}^{\max}} \sigma^2[t(i,j)]$.

Step 5. Calculate the variance of the full time reserve by

$$V[R_{full}(i,j)] = V[t(L_{cr})] + V[t(L_{ij}^{\max})] \tag{3.3.4}$$

Step 6. If $R_{full}(i,j) > 3\sqrt{V[R_{full}(i,j)]}$ holds, exclude activity (i,j) from set Q of activities comprising the equivalent network.

Step 7. In case $R_{full}(i,j) \le 3\sqrt{V[R_{full}(i,j)]}$ include activity (i,j) into set Q.

Note that using this refined algorithm decreases the volume of set Q essentially in comparison with its previously outlined initial version. However, both methods may be implemented only for medium-size networks (comprising circa 40-50 activities).

Determining the equivalent set Q may be also facilitated by means of simulation modeling. One has only to carry out a comparatively small number of simulation runs in order to single out activities with values $k_{\text{int}}(i,j)$ exceeding the pregiven levels. In the course of simulation we use the models (practically without any changes) outlined above, in §3.2.

§3.4 Estimating parameters of stochastic networks by significant paths' analysis

Consider an analytical method described in [49] which is of certain interest. This method is applicable for analyzing the distribution of the earliest accomplishment of an arbitrary event k of a network model (including, of course, the terminal event).

The essence of the method boils down to singling out from the entire set of paths entering the regarded event k, the most *significant* paths having a major influence on the distribution law $F_k(t)$ of the earliest accomplishment of that event. It should be noted that the regarded method imposes less "severe" requirements to the knowledge of the distribution laws of the activities' durations than many other methods, including the simulation method. If network simulation requires knowledge of the density distribution function for each network activity, for the significant paths analysis it is sufficient to determine average values $\bar{t}$ and variances $V(t)$ of the activities' durations. Thus, it can be well-recognized that for determining average values and standard deviations both triple-parametrical PERT estimates as well as double-parametrical ones, can be successfully implemented, namely

$$\begin{cases} \bar{t} = \dfrac{a+4m+b}{6} \\ \sigma_t = \dfrac{b-a}{6} \end{cases} \quad \text{as well as} \quad \begin{cases} \bar{t} = \dfrac{3a+2b}{5} \\ \sigma_t = \dfrac{b-a}{5} \end{cases}.$$

Each event k of a network model may be regarded as a terminal one with regards to a certain fragment of this model. That is why all further considerations are applied to terminal events, since it does not impose on the discussed method any real restrictions.

The earliest possible accomplishment of a terminal event k of a network model may be determined from the following relation:

$$T_k = \max\{t(L_1), t(L_2), \ldots, t(L_n)\}, \qquad (3.4.1)$$

where $t(L_1), t(L_2), \ldots, t(L_n)$ represent random values corresponding to all paths' lengths of the network model which connect the initial event with the terminal one.

It can be well-recognized that T_k itself may be regarded as a random value, while the probability of accomplishing the terminal event by a certain moment t may be calculated as

$$F(t) = \mathrm{P}\{T_k \le t\} = \mathrm{P}\{t(L_1) \le t, t(L_2) \le t, \ldots, t(L_n) \le t\}. \qquad (3.4.2)$$

The length $t(L_\mu)$ of path L_μ may be determined as the sum of all activities' durations entering this path, namely

$$t(L_\mu)=\sum_{(i,j)\in L_\mu} t(i,j), \tag{3.4.3}$$

where $t(i,j)$ stands for the duration of activity (i,j).

Since network models comprise usually activities of a similar specification level, different values $t(i,j)$ may be regarded as comparable by their relative influence on random fluctuations of the sum $t(L_\mu)$ in (3.4.3). According to the Central Limit Theorem and taking into account the assumption about independency of the network model activities' durations, the distribution of random value $t(L_\mu)$ for a total of 5-7 addendums may be approximately considered as normal with mean value

$$\overline{L_\mu}=\sum_{L_\mu}\bar{t}(i,j) \tag{3.4.4}$$

and variance

$$V(L_\mu)=\sum_{L_\mu} V\{t(i,j)\}, \tag{3.4.5}$$

where summarizing is carried out for all activities comprising path L_μ.

The joint distribution of random values $L_1, L_2, ..., L_n$ is also a multidimensional normal distribution. Thus, relation (3.4.2) may be represented as

$$F(t)=\int_{-\infty}^{t}...\int_{-\infty}^{t} f_N(\ell_1,\ell_2,...,\ell_n)d\ell_1 d\ell_2 .. d\ell_n, \tag{3.4.6}$$

where $f_N(\ell_1,\ell_2,...,\ell_n)$ denotes the multidimensional normal density distribution with mean $\begin{bmatrix}\overline{L_1}\\ \overline{L_2}\\ ...\\ \overline{L_n}\end{bmatrix}$, variance $\begin{bmatrix}V(L_1)\\ V(L_2)\\ ...\\ V(L_n)\end{bmatrix}$ and correlation coefficients matrix $\begin{bmatrix}1 & & & \\ \rho_{12} & 1 & & \\ ... & ... & ... & \\ \rho_{1n} & \rho_{2n} & ... & 1\end{bmatrix}$. Here ρ_{st} stands for correlation coefficient between L_s and L_t, which, in turn, is determined by relation

$$\rho_{st}=\frac{\sum_{L_s\cap L_t} Vt(i,j)}{\sqrt{V(L_s)V(L_t)}}. \tag{3.4.7}$$

In (3.4.7), condition $L_s \cap L_t$ means that summarizing is carried out for activities common for both paths L_s and L_t. If for a given network no paths have common activities at all, then values $t(L_1), t(L_2), ..., t(L_n)$ become independent (in the probabilistic sense of the word). In such a case

$$F(t)=\frac{1}{2^n}\prod_{\mu=1}^{n}\left[\Phi\left(\frac{t-\overline{L_\mu}}{\sqrt{2V(L_\mu)}}\right)+1\right], \tag{3.4.8}$$

where $\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_0^x e^{-u^2/2}du$ is the Laplace function.

As a matter of fact, network paths usually do have common activities, which is a good reason why (3.4.8) is in a general case unfit for determining $F(t)$. Besides, the main difficulty in calculating $F(t)$ boils down to the fact that even for a medium-size network model the total number of paths becomes significantly high. Thus, implementing computer algorithms enabling a total lookover of the network paths with subsequently determining their dependence (correlation) estimates, as well as directly solving the n-dimensional integral (3.4.6), becomes an enormously complicated and time-consuming problem. At the same time, examining random values similar to (3.4.1) shows [49] that when $n \to \infty$ they start displaying some asymptotic features, which become substantial already for $n = 15 \div 20$. The latter property enables singling out from the entire set of network paths a limited subset of about $15 \div 20$ *significant paths* which have the major influence on distribution function parameters.

As demonstrated by examining real-life network projects [49], model paths L_i characterized by the greatest mean values $\overline{L_i}$ of their durations and possessing at the same time the least correlation with the rest of the paths $L_1, L_2, ..., L_n$, would have the utmost influence on distribution function parameters.

Thus, to determine the distribution function $F(t)$ and its respective parameters, one should establish about $15 \div 20$ significant network paths with subsequently creating the random value

$$T_k^* = \max\{t(L_1), t(L_2), ..., t(L_m)\}, \; m = 15 \div 20, \tag{3.4.9}$$

representing with sufficient accuracy the theoretical $F(t)$.

In [49] several algorithms to single out the required set of significant paths for a given network model, including determining the mean values and variances of their durations by (3.4.4) and (3.4.5), are outlined. The below listed network model parameters are assumed to be known and are part of the input information for the algorithms:

- mean values $\bar{t}(i,j)$ and variances $V\{t(i,j)\}$ for all activities' durations;
- mean value $\overline{L}_{cr}$ and variance $V\{L_{cr}\}$ for the critical path duration. The critical path in the network model is the one with the longest duration when $t(i,j) = \bar{t}(i,j)$;
- full time reserves $R_f(i,j)$ for all activities in the network model, when their durations $t(i,j)$ have been determined by their mean values; it can be well-recognized that in such a case the full time reserve for any given activity (i,j) may be calculated as the residual between the critical mean values path $\overline{L}_{cr}$ and the longest mean values path $\overline{L}_v$ comprising the regarded activity; in other words, $R_f(i,j) = \overline{L}_{cr} - \max\limits_{(i,j)\in L_v}\{\overline{L}_v\}$.

3.4.1 Significant paths search algorithm

The essence of the described algorithm boils down to a lookover procedure comprising not the entire set G of all network model activities but a subset G' comprising activities whose full time reserves are less than a pregiven permissible level. The amount of activities entering subset G' increases along the increase of their full time reserves from the minimal level to the permissible one.

The permissible level R_{per} of full time reserves for a given network model may be calculated by means of the following empiric relation [49]

$$R_{per} = (1.5 \div 2.0)\sqrt{V\{L_{cr}\}}. \qquad (3.4.10)$$

The significant paths search algorithm consists of the following steps:

Step 1. For each activity (i, j) auxiliary value x_{ij} is being determined,

$$x_{ij} = \begin{cases} 1, & if\ (i, j) \in L_{cr} \\ 0, & if\ (i, j) \notin L_{cr}. \end{cases} \qquad (3.4.11)$$

Step 2. For all events $j \in L_{cr}$ parameter ω_j is being calculated, equal to the accumulated sum of duration variances for all activities on the critical path preceding event j:

$$\omega_j = \sum_{\substack{k=0 \\ (k,\ell)\in L_{cr}}}^{\ell=j} V\{t(k, \ell)\}. \qquad (3.4.12)$$

Step 3. For the given activity (i, j) check condition $x_{ij} = 1$. If the condition holds, go to Step 9, otherwise (e.g., when $x_{ij} = 0$) proceed to the next step.

Step 4. For all events i check condition $i \in L_{cr}$. If the condition holds, go to Step 5, otherwise proceed to Step 6.

Step 5. For the considered activity (i, j) check whether it belongs to any already established significant path *excluding* the critical path. If "yes", go to Step 7, otherwise proceed to Step 8.

Step 6. The procedure of the step is identical to that of Step 5, with the only exception that in case of the positive answer to the question go to Step 9, otherwise - to Step 10.

Step 7. For the considered activity (i, j) determine auxiliary value α_{ij}. Set $\alpha_{ij} = \omega_i + V\{t(i, j)\}$.

Step 8. For the considered activity (i, j) determine auxiliary value α_{ij}. Set $\alpha_{ij} = \omega_i$.

Step 9. For the considered activity (i, j) determine auxiliary value α_{ij}. Set $\alpha_{ij} = v_i + V\{t(i, j)\}$. As to parameter v_i, its calculation is outlined in the below Step 12. Note that for the initial event (e.g., when $i = 0$) $v_0 = 0$.

Step 10. For the considered activity (i, j) determine auxiliary value α_{ij}. Set $\alpha_{ij} = v_i$.

Step 11. Check whether auxiliary value α_{ij} has been determined *for all* activities (i,j) entering event j. If "yes", go to the next step, otherwise return to Step 3.

Step 12. For the considered event j calculate parameter v_j as follows:

$$v_j = \min(\alpha_{ij}). \tag{3.4.13}$$

Activity (i,j) for which α_{ij} reaches its minimal value is being marked by assigning $h_{ij} = 1$.

Step 13. Check whether parameter v_j has been calculated for all events j. If "yes", proceed to the significant paths laying out algorithm. Otherwise, return to Step 3.

It can be well-recognized that the above outlined significant paths search algorithm provides for each event j of the network model its assigned parameter value v_j. As a matter of fact, parameter v_j represents the minimal sum of activities durations variances for activities common to a certain path L_v and the set of already laid out significant paths of the network. Minimization is carried out for all paths L_v ($v = 1,2,...,k$) entering event j.

3.4.2 Algorithm for laying out significant paths

Significant path L_v to be determined is laid out through network model events with the least values of parameter v_j, starting from the terminal event. The algorithm comprises the following main steps:

Step 1. For the given event j determine set of events $\{i\}$ directly preceding j. Also determine the corresponding set of activities $\{(i,j)\}$ connecting each i with j.

Step 2. Verify marker h_{ij} values for activities $\{(i,j)\}$ determined at Step 1 (see procedure of Step 12 from the previous significant paths search algorithm outlined above).

Step 3. Determine event i from set $\{i\}$ from which activity (i,j) with $h_{ij} = 1$ starts. Check whether event i is the initial one. If yes, go to the next step. If no, assign j index i and return to Step 1.

Step 4. The sequence of events determined on Steps 1-3, together with the corresponding set of activities, form the required significant path L_v to be laid out. For L_v, calculate its respective duration mean value and variance:

$$\overline{L}_v = \sum_{L_v} \bar{t}(i,j), \tag{3.4.14}$$

$$V\{L_v\} = \sum_{L_v} V\{t(i,j)\}. \tag{3.4.15}$$

3.4.2 Algorithm for determining mutual correlation among significant paths

The algorithm described in [49] is intended to determine correlation coefficients of the newly established significant path L_ν with each path belonging to the set $\{L_\mu\}$ of significant paths previously established. For every couple of significant paths L_ν and L_μ, correlation coefficient $\rho_{\nu\mu}$ is calculated. The procedure of the algorithm boils down to the following steps:

Step 1. Determine the set of activities (i,j) belonging jointly both to L_ν and L_μ.

Step 2. Check condition $\{(i,j)\} = \varnothing$. If the condition holds, assign $\rho_{\nu\mu} = 0$ and go to Step 5, otherwise proceed to the next step.

Step 3. Calculate correlation coefficient $\rho_{\nu\mu}$ by relation

$$\rho_{\nu\mu} = \frac{S_{\nu\mu}}{\sqrt{V\{L_\nu\} \cdot V\{L_\mu\}}}, \tag{3.4.16}$$

where $S_{\nu\mu}$ stands for the sum of duration variances of activities (i,j) belonging jointly both to L_ν and L_μ.

Step 4. Compare calculated value $\rho_{\nu\mu}$ with the pregiven permissible target $\rho_{per} = 0.8 \div 0.9$. If inequality $\rho_{\nu\mu} < \rho_{per}$ holds, return to Step 1. Otherwise, when $\rho_{\nu\mu} \geq \rho_{per}$, it can be well-recognized that the newly laid out significant path L_ν is practically quite similar to the already existing path L_μ and, thus, wouldn't have any further influence of the distribution function $F(t)$ of accomplishing the network terminal event. Go to Step 5.

Step 5. The algorithm for determining mutual correlation estimates among significant paths terminates.

Consecutive implementation of the above outlined algorithms enables singling out from the entire set of paths entering a given event, a subset of significant paths $\{L_\mu\}$ ($\mu = 1,2,\ldots,m$, $m = 15 \div 20$) alongside with their durations mean values $(L_1, L_2, \ldots, L_m)$, variances $(V\{L_1\}, V\{L_2\}, \ldots, V\{L_m\})$ and correlation coefficients matrix ρ_{ij}.

As mentioned above, determining $F(t)$ is subject to multiple (n-dimensional) integrating of a multi-dimensional normal distribution function, while the significant paths method succeeded in reducing the dimensions space of (3.4.6) to a total of $n = 15 \div 20$. Unfortunately, even in the reduced case the existing analytical solution methods for such problems prove to be extremely complicated, which makes their practical implementation virtually impossible.

For determining statistical moments of distribution function $F(t)$ one may apply a method outlined in [49]. Its essence is based on the fact that a random value $T_k^* = \max\{L_1, L_2, \ldots, L_m\}$ may be represented as follows:

$$\begin{cases} y_1 = \max(L_1, L_2), \\ y_2 = \max(y_1, L_3), \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ T_k^* = \max(y_{m-2}, L_m). \end{cases} \tag{3.4.17}$$

Mean value and variance of y_1 may be determined, in their turn, as listed below:

$$\begin{cases} \bar{y}_1 = \bar{L}_1 F(\alpha) + \bar{L}_2 F(-\alpha) + af(\alpha), \\ V\{y_1\} = \left(\bar{L}_1^2 + V\{L_1\}\right)F(\alpha) + \left(\bar{L}_2^2 + V\{L_2\}\right)F(-\alpha) + (L_1 + L_2)af(\alpha) - (\bar{y}_1)^2, \\ \alpha = \dfrac{\bar{L}_1 - \bar{L}_2}{a}, \\ a^2 = V\{L_1\} + V\{L_2\} - 2\rho_{12}\sqrt{V\{L_1\}\cdot V\{L_2\}}, \end{cases} \tag{3.4.18}$$

where ρ_{12} designates the correlation coefficient between L_1 and L_2.

On assumption that random value y_1 is distributed normally, similar relations apply for determining $\bar{y}_2$ and $V\{y_2\}$. Proceeding analogously, determine eventually mean value and variance of T_k^*. Assuming T_k^* also distributed normally, one may obtain p-quantile assessments of the entire time span for accomplishing the project as a whole.

To implement the regarded procedure, there is need to know the correlation coefficient between random values L_k and $\max\{L_1, L_2, ..., L_{k-1}\}$ at each step of the algorithm. This correlation coefficient, in its turn, may be assessed with sufficient accuracy by the multiple correlation coefficient $\rho_{L_k}\{L_1, L_2, ..., L_{k-1}\}$ which determines the dependency measure between L_k and set $\{L_1, L_2, ..., L_{k-1}\}$:

$$\rho_{L_k}\{L_1, L_2, ..., L_{k-1}\} = \sqrt{\frac{\Delta^*}{\Delta}}, \tag{3.4.19}$$

where

$$\Delta^* = (-1)^k \begin{vmatrix} \rho_{k1} & \rho_{k2} & \cdots & \rho_{k(k-1)} & 0 \\ 1 & \rho_{12} & \cdots & \rho_{1(k-1)} & \rho_{1k} \\ \rho_{21} & 1 & \cdots & \rho_{2(k-1)} & \rho_{2k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_{(k-1)1} & \rho_{(k-1)2} & \cdots & 1 & \rho_{(k-1)k} \end{vmatrix}, \tag{3.4.20}$$

$$\Delta = \begin{vmatrix} 1 & \rho_{21} & \cdots & \rho_{(k-1)1} \\ \rho_{12} & 1 & \cdots & \rho_{(k-1)2} \\ \cdots & \cdots & \cdots & \cdots \\ \rho_{1(k-1)} & \rho_{2(k-1)} & \cdots & 1 \end{vmatrix}. \tag{3.4.21}$$

To assess the accuracy of the discussed method of determining p-quantile assessments of the entire time span for accomplishing the project as a whole, the methodology of statistical simulation of the random value $T_k^* = \max\{L_1, L_2, ..., L_m\}$ may be implemented.

Let us illustrate the latter by numerical example represented in Fig. 3.1.

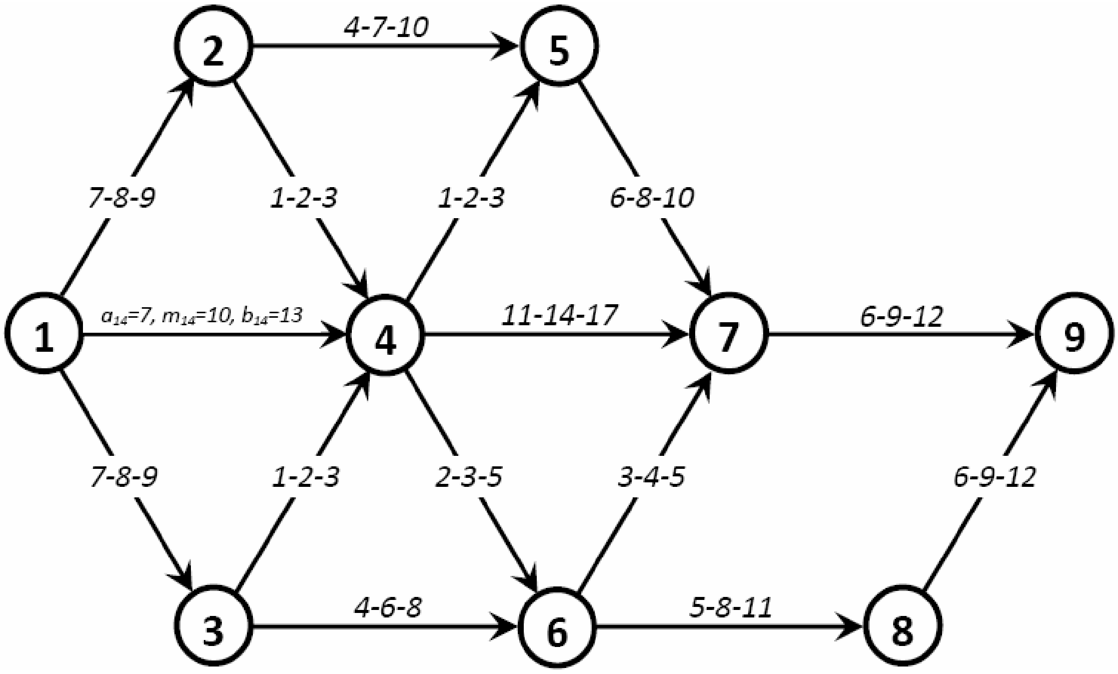

Figure 3.1. Network model example

Consider a network model comprising activities with randomly distributed durations, their mean values and variances being determined by relations

$$\bar{t}(i,j) = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6},$$

$$\sigma_{ij}^2 = \frac{1}{36}\left(b_{ij} - a_{ij}\right)^2,$$

with the results of the calculations presented in Tab. 3.1.

Tab. 3.2 represents network paths L_k ($k = 1,2,...,15$) as well as mean values $\bar{L}_k$ and variances $V\{L_k\}$ of their durations: $\bar{L}_k = \sum\limits_{(i,j)\in L_k} \bar{t}(i,j)$, $V\{L_k\} = \sum\limits_{(i,j)\in L_k} \sigma_{ij}^2$.

For the regarded numerical example, consider the procedure of applying the significant paths search algorithm in conjunction with the algorithm for laying out significant paths.

Table 3.1. Statistical parameters for network model activities

Activity	$\bar{t}(i,j)$	σ_{ij}^2	*Activity*	$\bar{t}(i,j)$	σ_{ij}^2	*Activity*	$\bar{t}(i,j)$	σ_{ij}^2
(1, 2)	8.0	0.11	(3, 4)	2.0	0.11	(5, 7)	8.0	0.44
(1, 3)	8.0	0.11	(3, 6)	6.0	0.44	(6, 7)	4.0	0.11
(1, 4)	10.0	1.00	(4, 5)	2.0	0.11	(6, 8)	8.0	1.00
(2, 4)	2.0	0.11	(4, 6)	3.0	0.25	(7, 9)	9.0	1.00
(2, 5)	7.0	1.00	(4, 7)	14.0	1.00	(8, 9)	9.0	1.00

From the paths represented in Tab. 3.2 choose path $L_8 = (1,4,7,9)$ with $\bar{L}_8 = 33$ and regard it as the critical one L_{cr}. Determine the permissible level R_{per} of full time reserves $R_{per} = 2\sqrt{V\{L_{cr}\}} = 3.5$ and the permissible target value of the correlation coefficient $\rho_{per} = 0.8$.

1. From the entire set of activities entering the network single out the subset Q of activities with the full time reserves parameter not less than the permissible level R_{per}. It can be well-recognized that this subset doesn't comprise activities $(4,5)$ and $(6,7)$ which are excluded from further examination.
2. Determine values x_{ij}: $x_{14} = x_{47} = x_{79} = 1$; for all other activities $x_{ij} = 0$.
3. Calculate auxiliary parameter ω_j for activities on the critical path: $\omega_1 = 0$, $\omega_4 = 1.0$, $\omega_7 = 2.0$, $\omega_9 = 3.0$.

Table 3.2. Network paths and their durations parameters

Network paths	$\bar{L}_k$	$V\{L_k\}$	*Network paths*	$\bar{L}_k$	$V\{L_k\}$
L_1=(1, 2, 5, 7, 9)	32	2.55	L_9 =(1, 4, 6, 7, 9)	26	2.36
L_2=(1, 4, 5, 7, 9)	29	2.89	L_{10}=(1, 3, 4, 7, 9)	33	2.22
L_3=(1, 3, 6, 7, 9)	27	1.56	L_{11}=(1, 3, 4, 6, 7, 9)	26	1.58
L_4=(1, 2, 4, 5, 7, 9)	29	1.67	L_{12}=(1, 3, 6, 8, 9)	31	2.55
L_5=(1, 3, 4, 5, 7, 9)	29	1.67	L_{13}=(1, 2, 4, 6, 8, 9)	30	1.71
L_6=(1, 2, 4, 7, 9)	33	2.22	L_{14}=(1, 4, 6, 8, 9)	30	2.69
L_7=(1, 2, 4, 6, 7, 9)	26	1.60	L_{15}=(1, 3, 4, 6, 8, 9)	30	1.91
L_8=(1, 4, 7, 9)	33	3.00			

The below Tab. 3.3 represents the following procedure of sequentially applying the significant paths search algorithm in conjunction with the algorithm for laying out significant paths.

Accumulated final results of implementing both algorithms are brought together in Tab. 3.5.

At the output of computerized analysis of the regarded network model, the following paths have been determined as significant (these paths are visually indicated in Tab. 3.2 by grey marker): $\{L_1, L_8, L_{10}, L_{12}\}$. Tab. 3.4 displays the calculated correlation coefficients matrix $\rho_{v\mu}$ for the regarded paths (for any pair of paths L_v and L_μ, the correlation coefficient $\rho_{v\mu}$ is calculated by relation $\rho_{v\mu} = \frac{1}{\sqrt{V\{L_v\} \cdot V\{L_\mu\}}} \sum_{L_v \cap L_\mu} \sigma_{ij}^2$).

The results of determining distribution function parameters for the moment of accomplishing the terminal event (9) are exhibited in Tab. 3.6.

On assumption that the terminal event's (9) accomplishment is distributed normally with mean $\bar{\lambda}_9 = 34.16$ and standard deviation $\sigma_9 = 1.53$, we may determine now the probability $P\{t \le t_{early}(9)\}$, where $t_{early}(9)$ stands for the earliest moment of accomplishing event (9), by relation

$$P\{t \le t_{early}(9)\} = \frac{1}{\sigma_9 \sqrt{2\pi}} \int_{-\infty}^{t_{early}(9)} \exp\left\{-\frac{1}{2} \frac{(t - \bar{\lambda}_9)^2}{\sigma_9^2}\right\} dt = 0.222 .$$

Tab. 3.7 showcases comparative results of estimating statistical parameters of the regarded network model, by means of three competing methods (the above outlined analytical procedure versus classical PERT and the Monte-Carlo simulation method).

It can be well-recognized that the outlined above method of significant paths is essentially closer to the exact results ($\bar{\lambda}_9$ = 34.20) obtained by means of simulation than the classical PERT methods.

Table 3.3. Detailed procedure of sequentially applying the search and laying out algorithms

Serial	Step of the algorithm	Description of procedure
		Significant paths search algorithm
1	Step 3	Choose from subset Q activity $(1,2)$. For the chosen activity $x_{12}=0$
2	Step 4	Check whether event (1) belongs to the critical path - "*Yes*"
3	Step 5	Check whether activity $(1,2)$ belongs to any already established significant path - "*No*"
4	Step 8	Calculate auxiliary value α_{12}: $\alpha_{12}=\omega_1=0$
5	Step 11	Check whether auxiliary value α_{ij} has been determined *for all* activities (i,j) entering event (2) - "*Yes*"
6	Step 12	Calculate parameter v_2: $v_2=\alpha_{12}=0$. Activity $(1,2)$ is being marked by assigning $h_{12}=1$ and is excluded from further examination
7	Step 13	Check whether subset Q is empty - "*No*"
8	Step 3	Choose from subset Q activity $(1,3)$. For the chosen activity $x_{13}=0$
9	Step 4	Check whether event (1) belongs to the critical path - "*Yes*"
10	Step 5	Check whether activity $(1,3)$ belongs to any already established significant path - "*No*"
11	Step 8	Calculate auxiliary value α_{13}: $\alpha_{13}=\omega_1=0$
12	Step 11	Check whether auxiliary value α_{ij} has been determined *for all* activities (i,j) entering event (3) - "*Yes*"
13	Step 12	Calculate parameter v_3: $v_3=\alpha_{13}=0$. Activity $(1,3)$ is being marked by assigning $h_{13}=1$ and is excluded from further examination
14	Step 13	Check whether subset Q is empty - "*No*"
15	Step 3	Choose from subset Q activity $(1,4)$. For the chosen activity $x_{14}=1$
16	Step 9	Calculate auxiliary value α_{14}: $\alpha_{14}=v_1+\sigma_{14}^2=1$
17	Step 11	Check whether auxiliary value α_{ij} has been determined *for all* activities (i,j) entering event (4) - "*No*"
18	Step 3	Choose from subset Q activity $(2,4)$. For the chosen activity $x_{24}=0$
19	Step 4	Check whether event (2) belongs to the critical path - "*No*"
20	Step 6	Check whether activity $(2,4)$ belongs to any already established significant path - "*No*"

21	Step 10	Calculate auxiliary value α_{24}: $\alpha_{24} = v_2 = 0$
22	Step 11	Check whether auxiliary value α_{ij} has been determined *for all* activities (i, j) entering event (4) - "*No*"
23	Step 3	Choose from subset Q activity $(3,4)$. For the chosen activity $x_{34} = 0$
24	Step 4	Check whether event (3) belongs to the critical path - "*No*"
25	Step 6	Check whether activity $(3,4)$ belongs to any already established significant path - "*No*"
26	Step 10	Calculate auxiliary value α_{34}: $\alpha_{34} = v_3 = 0$
27	Step 11	Check whether auxiliary value α_{ij} has been determined *for all* activities (i, j) entering event (4) - "*Yes*"
28	Step 12	Calculate parameter v_4: $v_4 = \min(\alpha_{14}, \alpha_{24}, \alpha_{34}) = 0$. Activity $(2,4)$ is being marked by assigning $h_{24} = 1$. Activities $(1,4)$, $(2,4)$, $(3,4)$ are excluded from further examination
29	Step 13	Continue calculations in the same manner until parameter v_j is determined *for all* network model activities. Then switch over to laying out significant paths
		Algorithm for laying out significant paths
1	Step 1	Choose event (9) and determine the set of activities entering this event - $\{(7,9),(8,9)\}$
2	Step 2	Verify marker h_{ij} values for activities $(7,9)$ and $(8,9)$: $h_{79} = 0$, $h_{89} = 1$
3	Step 3	Locate event (8) and check whether it is the initial event - "*No*"
4	Step 1	Determine the set of activities entering event (8) - $\{(6,8)\}$
5	Step 2	Verify marker h_{ij} value for activity $(6,8)$: $h_{68} = 1$
6	Step 3	Locate event (6) and check whether it is the initial event - "*No*"
7	Step 1	Determine the set of activities entering event (6) - $\{(3,6),(4,6)\}$
8	Step 2	Verify marker h_{ij} values for activities $(3,6)$ and $(4,6)$: $h_{36} = 1$, $h_{46} = 0$
9	Step 3	Locate event (3) and check whether it is the initial event - "*No*"
10	Step 1	Determine the set of activities entering event (3) - $\{(1,3)\}$
11	Step 2	Verify marker h_{ij} value for activity $(1,3)$: $h_{13} = 1$
12	Step 3	Locate event (1) and check whether it is the initial event - "*Yes*"
13	Step 4	The sequence of activities $(1,3)$, $(3,6)$, $(6,8)$, $(8,9)$ forms the required significant path to be laid out

Table 3.4. Correlation coefficients matrix $\rho_{\nu\mu}$ for significant paths

L_ν \ L_μ	L_1	L_8	L_{10}	L_{12}
L_1	1	0.36	0.24	0
L_8	0.36	1	0.78	0
L_{10}	0.24	0.78	1	0.05
L_{12}	0	0	0.05	1

§3.5 Upon monitoring stochastic network projects with time parameters

In the section under consideration we will *not* describe the control techniques as a feedback model which allows project managers identifying deviations from the target and initiating corrective actions to reorient the progress of the project in the desired direction. We are committed to outline such control techniques in the following chapters of this monograph. Instead, we are going to describe the use of various stochastic network's parameters entering an information-advisory system, without optimization techniques whatsoever. For projects under random disturbances such an information-advisory system may determine the probability of meeting the target's due date on time and is more effective on the planning stage, where the project's workable plan has to be checked. Note that planning does not end when the project starts to be realized since replanning and updating goes hand by hand with on-line control.

Table 3.5. Accumulated final results of applying the search and laying out algorithms

Activity	¦	(1,2)	(1,3)	(1,4)	(2,4)	(3,4)	(2,5)	(3,6)		(4,6)	(4,7)		(5,7)	(6,8)	(7,9)		(8,9)
Event	1	2	3		4		5		6			7		8		9	
$L^{(1)}=L_8$	+			+							+				+		
x_{ij}	---	0	0	1	0	0	0	0		0	1		0	0	1		0
ω_j	0				1.0							2.0				3.0	
α_{ij}	0	0	0	1.0	0	0	0	0		1.0	1.0		0	0	1.0		0
v_j	0	0	0		0		0		0			0		0		0	
h_{ij}	---		1					1		0				1	0		1
$L^{(2)}$	+		+					+						+			+
α_{ij}	---	0	0.11	1.0	0	0.11	0	0.55		1.0	1.0		0	1.55	1.0		2.55
v_j	0	0	0.11		0		0		0.55			0		1.55		1.0	
h_{ij}	---	1					1				0		1		1		0
$L^{(3)}$	+	+					+						+		+		
α_{ij}	---	0.11	0.11	1.0	0.11	0.11	1.11	0.55		1.0	1.11		1.55	1.55	2.11		2.55
v_j	0	0.11	0.11		0.11		1.11		0.55			1.11		1.55		2.11	
h_{ij}	---	1		0	1	0					1		0		1		0
$L^{(4)}$	+		+			+					+				+		

Table 3.6. Consecutive estimates of random distribution function parameters

Auxiliary parameter λ_i	Mean $\bar{\lambda}_i$	Standard deviation σ_{λ_i}
$\lambda_1 = L_8$	33.00	1.73
$\lambda_2 = \max\{L_8, L_{10}\}$	33.61	1.65
$\lambda_3 = \max\{\lambda_2, L_1\}$	33.97	1.57
$\lambda_4 = \max\{\lambda_3, L_{12}\}$	34.16	1.53

Note that some problems of controlling stochastic network projects have already been outlined above, in Chapter 1. The main difference between those parts of the book is that in Chapter 1 the material is outlined on behalf of the *creators of projects' control models*, whereas in Chapter 3 we describe the problem of monitoring a project to be used by the *project manager* who is often unfamiliar with modern control approaches in project management.

Table 3.7. Comparative estimates of the project's random duration parameters by means of different methods

Methods	$\bar{\lambda}_9$	σ_9	$P\{t \le t_{early}(9)\}$
PERT	33.00	1.73	0.500
Monte-Carlo	34.20	1.55	0.219
Analytical	34.16	1.53	0.222

On the planning stage the project manager has first to preset the confidence level p, i.e., the probability of meeting the project's target on time, while the due date has to be pregiven as well. It goes without saying that value p depends fully on the complexity, novelty and indeterminacy of the project's goal. Several important concepts in determining the project's due date have to be implemented:

a) the project's due date is calculated by adding to the project's starting moment the p-quantile of the project's duration. In some projects under random disturbances the p-quantile is not pregiven, but calculated on the basis of the previously preset due date. As a rule, such direct and inverse calculations can be carried out by simulative modeling, as outlined in §§3.1-3.2.

b) confidence values about the progress of the project have to be obtained regularly and later on analyzed and compared with the previously established p-quantile value. In the simplest case the critical path durations to be determined over time have to be compared with their p-quantile estimates.

Practically, four different situations may emerge at the planning stage [49,53]:

1. The critical path length does not differ essentially from other paths entering the critical zone (see §3.1) while the p-quantile estimates (e.g., for $p = 0.7$ and $p = 0.8$) differ substantially from each other. This may occur when the project comprises a group of activities which have a certain tendency to be on the critical path and an essentially lower probability to belong to other paths of the critical zone. In certain cases they may possess a large variance of their durations. Such a situation may take place either by non-objective underestimating the activities' parameters a and b by their executors, or by impossibility for one reason or another to estimate their optimistic and pessimistic durations. At any case those activities have to be checked in order to narrow the interval $[a,b]$.

2. The p-quantile estimates do not differ essentially one from another, as well as the network's paths entering the critical zone. This results in a certain steadiness of the progress of the project.
3. The p-quantile estimates, as well as the paths entering the critical zone differ from each other essentially. Similar to case (2), we have no reasons to suspect the executors in presenting deliberately incorrect activity parameters.
4. The p-quantile estimates differ from each other non-essentially, while the paths entering the critical zone differ essentially one from another. This is a relatively rare situation and, if observed, might be explained by the following reasons:
   a) the path with the highest probability to become critical in the course of the project's realization comprises activities with larger average durations than other paths entering the critical zone, but the latter possess higher duration variances than the first one. It should be noted that p-quantile values depend both on average and variance values as distinct from the critical paths durations depending only on their averages. Such a case needs to be clarified in order to correct the information obtained from activities executors;
   b) in certain cases the postulated p.d.f. for a group of activities may deviate from the true distribution law. The reason may be established by simulation modeling.

To sum up, only p-quantile estimates are the stochastic network parameters which have to be compared periodically with the project's due date, within the course of the project's implementation. In case of essential deviations the project's structure has to be corrected, until the deviation will become insignificant. The corrective actions may boil down either to changing the project's due date or amending the project's targets. Periodical information regarding p-quantile estimates' changes within the course of the project's realization is usually forwarded to the manager in a form similar to that of Tab. 3.8.

Columns 1-4 are self-explainable. Columns 5-9 represent the information regarding the progress of the project, usually in equidistant time moments $t_1, t_2, ..., t_i$, etc. The information is intended for the project manager and refers to project's milestones $\{\gamma\}$ - the most important events including the terminal event. At any routine moment t_i the project is inspected and the following stochastic network model's parameters are determined. Column 7 presents the confidence probability $p_{con.i}(\gamma)$ to reach milestone γ by means of inspection at moment t_i, while column 6 showcases the analogous value obtained at the previous inspection point t_{i-1}. Column 9 contains the calculated time moment p_{giv}-quantile of reaching node γ on the basis of the updated project at moment t_i, while column 8 displays similar information obtained at the previous moment t_{i-1}. Note that p_{giv} is a pregiven confidence probability to be accepted by the project manager before the project actually starts. By examining the changes in columns 6-7 and 8-9, correspondingly (for all milestones γ), the manager estimates the progress of the project and, if necessary, takes appropriate decisions. The latter may include quite sophisticated corrective *control actions*, e.g., plan's updating, resource reallocation, amendment of project's targets, etc. Possible control actions will be examined and outlined in greater detail in the next chapters of our monograph.

Table 3.8. Information on the progress of the project

<table>
<tr><td colspan="2" rowspan="2">System code</td><td colspan="2">No. of calculation</td><td colspan="4" rowspan="2">Expected moments of milestones' realizations</td><td>Date</td><td>Page</td></tr>
<tr><td>Basic</td><td>Analyzed</td><td></td><td>Total pages</td></tr>
<tr><td rowspan="4">Serial</td><td rowspan="4">Development stage code</td><td colspan="2">Milestones</td><td colspan="5">Realization moments</td><td rowspan="4">Comments</td></tr>
<tr><td rowspan="3">Code $\{\gamma\}$</td><td rowspan="3">Name</td><td colspan="3">Inspected</td><td colspan="2">Pregiven</td></tr>
<tr><td rowspan="2">Date of $\{\gamma\}$</td><td colspan="2">Confidence p_{con}</td><td colspan="2">Due date</td></tr>
<tr><td>t_{i-1}</td><td>t_i</td><td>t_{i-1}</td><td>t_i</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="10">Responsible Executor</td></tr>
</table>

Thus, the outlined techniques refer to the planning stage which is usually deeply linked to on-line control. Note that the term "plan" usually means time scheduling for all activities entering the project. Without scheduling the project cannot actually be realized since resources cannot be delivered in time to ensure proper execution of activities. Thus, bridges have to be build between the planning, control, and scheduling stages. For a network project to be carried out under random disturbances the problem of linking together those three main stages refers to one of the most complicated problems in project management which has not been solved as yet. Note that the complexity of this problem stems from the contradictions still being part of the PERT techniques and discussed in depth in Chapter 2.

In conclusion, let us present the three main concepts of solving this main problem in stochastic project management.

The **first concept** is based on *analytical methods* to determine calendar plans of scheduling activities with random durations. The general idea is to substitute each random activity duration by a deterministic one, with duration equal to the average value of the initial activity. Thus, we deal further on with a deterministic network project which enables calculating scheduling parameters on the basis of determine values $t_{ear}(i)$ and $t_{lat}(i)$. Due dates for reaching the project's milestones can be easily calculated. Thus, control actions at the on-line control stage (in case of essential deviations from the target) boil down to introducing additional external resources in order to compensate the deviations. Such a monitoring has to be aimed, first, at lowering the durations for activities entering the critical zone. Note that a deterministic model is easy to operate, although it usually requires additional resources and, thus, raises the cost of the project. Various project management companies use, in addition to the regarded compensating resources, a flexible policy of changing the intensity of activities' realization, especially for activities entering the critical zone. However, the described concept would not be recommended for complicated innovative R&D projects, since the damage caused by such a "simplified" monitoring may be very high. From the theoretical point of view such a deterministic model cannot be accepted, since substituting random values by their average values results in unbiased errors which in certain cases may become as high as 40-50% [49,152,157].

According to the **second concept**, activities' calendar planning and scheduling is determined *by means of simulation*. Given the project's due date D and all the activities' parameters $a(i,j)$ and $b(i,j)$, it is not difficult to calculate the p-quantile estimates, i.e., confidence probabilities for all the project's milestones to be reached in the course of the project's realization. If for milestone γ the p-confidence value $W_p(\gamma)$ to reach this event is calculated, then all activities (k,γ) entering γ have to take value $W_p(\gamma)$ as a schedule for their latest time to be finished. Certain activities will now possess probabilistic time reserves outlined in §3.1. Thus, it becomes possible to calculate for such a p-quantile scale new calendar plans $t_{ear}(i)$ and $t_{ear}(j)$. Note that p-quantile estimates of reaching a certain event γ as early as possible, may be transformed to the latest possible moment of reaching the same milestone. The methodology involved will not undergo drastic changes.

For the simulation concept the new durations of executing a routine activity (i,j) may be determined by

$$t(i,j)=W_p\{t_{ear}(j)\}-W_p\{t_{ear}(i)\} \qquad (3.5.1)$$

or

$$t(i,j)=W_{1-p}\{t_{ear}(j)\}-W_{1-p}\{t_{ear}(i)\}. \qquad (3.5.2)$$

Relations (3.5.1-3.5.2) enable developing compensative control actions to diminish the deviation from the project's target. Although such an approach requires certain mathematical experience on behalf of the project's managing team, it seems more realistic and provides less unavoidable errors. The main shortage of the approach is that in certain cases (not very often ones) implementing relations (3.5.1-3.5.2) might introduce changes in the very topology of the network's structure.

The **third approach** is, as a matter of fact, a combined one and has been outlined in Chapter 1. It is based on the following concepts. Monitoring (i.e., scheduling) the project is carried out by analytical estimates, while on-line control is realized by means of simulation. To our opinion, such a combined method is the best and the most justified one. Control actions are implemented on the basis of the project's inspection in periodically determined control points, and the deviations from target trajectories are obtained without errors. Thus, local errors caused by deterministic scheduling, are periodically corrected at each routine control point, within the project's functioning.

§3.6 Conclusions

The following conclusions can be drawn from the Chapter:

1. In stochastic network project management the existing classical network parameters developed for deterministic networks, have to be substituted for probabilistic parameters. We recommend using the p-quantile values which are especially beneficial for PM systems of information-advisory type.
2. Simulation modeling remains as yet the easiest in application, especially in comparison with analytical methods. Being interesting from the theoretical point of view, the latter can be applied in practice to calculate network parameters for small-size networks only.
3. Monitoring stochastic network projects has to be carried out on two levels. On the lower level simplified deterministic scheduling has to be implemented whereas at the upper level controlling has to take place by using appropriate on-line models.
4. With only time parameters involved monitoring stochastic network projects can be carried out by project managers who can be easily empowered and qualified. For the case of time-cost parameters monitoring becomes more complicated and requires special experience and scientific skills. The latter case refers fully to modern innovative projects.

PART II

ON-LINE CONTROL MODELS FOR STOCHASTIC NETWORK PROJECTS

Chapter 4. On-Line Control Models Based on Sequential Analysis

§4.1 On-line control model for a PERT-COST project with a fixed speed

4.1.1 *Introduction*

This section presents results outlined in [64,66,68] with the aim of developing an on-line control model for various types of stochastic network projects. A hierarchical on-line control model for several PERT-COST type projects being carried out simultaneously is considered.

On the project level, each project is controlled separately in order to minimize the number of control points subject to a chance constraint, which seeks to prevent deviations from the planned trajectory within the planning horizon with pregiven probability. If at the control point it is anticipated that the project will not be on target subject to the chance constraint, then an emergency is declared and the company level is faced with the problem of reassigning the remaining budget among the projects so that the faster ones may help the slower ones. Thus, the model has in fact two objectives: minimizing the number of control points and maximizing the probability that the slowest project can meet its due date on time.

We will *not* describe the mathematical formulations of all optimization problems that are imbedded in the hierarchical model. Those problems will be outlined in depth in Chapters 14-15 when considering the hierarchical cost-simulation control model. Instead, in the present section we will focus on analyzing only one element of the multilevel control model, namely, the problem of determining the next control point t_{g+1} for a single PERT-COST project.

In order to proceed, we will require several notations.

4.1.2 *Notation*

Let us introduce the following terms:

$G(N,A)$ - network project (graph) of PERT-COST type;

D - the due date of the project (pregiven);

p^* - least permissible probability for the project to be accomplished on time (pregiven);

$C(t)$ - the remaining budget which is not utilized at moment t;

$G(t)$ - the remaining network of the project at point t;

Δ - the minimal time span between two consecutive control points in order to force convergence (pregiven);

$(i,j)\in G(N,A)$ - activity leaving node i and entering node j;

$t(i,j)$ - random duration of activity (i,j);

$c(i,j)$ - budget assigned to activity (i,j);

$A(i,j)$ - pregiven value to satisfy $A(i,j)/c(i,j)=a(i,j)$, which is the lower bound of random value $t(i,j)$;

$B(i,j)$ - pregiven value to satisfy $B(i,j)/c(i,j)=b(i,j)$, which is the upper bound of random value $t(i,j)$;

$N(t)$ - the remaining number of control points to inspect the progress of the project, beginning with moment t; $N(0)=N$ (total number of control points);

t_g - the g-th control point, $g=0,1,...,N$;

$V^{pl}(t)$ - the planned trajectory curve.

$V^f(t)$ - the state variable observed at the project's inspection in control point t.

We will adopt the justification outlined in Chapter 2 and, thus, assume the p.d.f. of activity (i,j) duration as follows:

$$f_{ij}(x)=\frac{12}{[b(i,j)-a(i,j)]^4}[x-a(i,j)][b(i,j)-x]^2. \tag{4.1.1}$$

4.1.3 <u>On-line control problem at the project level</u>

Like any other on-line control it has to be carried out by comparing the state variable of the progress of the project at control points with the corresponding values of the planned project target (trajectory). Thus, to carry out on-line control, we have to determine for each project $G(t)$ its planned trajectory curve $V^{pl}(t)$ together with the state variable $V^f(t)$.

At the project level the following control model has to be implemented [64]:

At any control point t_g, $0\le g\le N$, to minimize the objective

$$\min_{t_{g+1}>t_g} N(t_g) \tag{4.1.2}$$

subject to

$$\begin{cases} t_0=0 \\ t_N=D, \end{cases} \tag{4.1.3}$$

$$t_{g+1}-t_g\ge\Delta, \tag{4.1.4}$$

$$\Pr\left\{V^f(t)\le V^{pl}(t)\right\}\ge p^* \quad \forall t: t_g\le t\le t_{g+1}. \tag{4.1.5}$$

The general idea of the on-line control is as follows:

At each routine control point t_g, $g=0,1,...,N$, inspection is undertaken to observe the remaining budget $C(t_g)$. Value $C(t_g)$ is the state variable $V^f(t)$ at point $t=t_g$. At the beginning of the project realization, at $t=t_0=0$, the budget is still unspent with $C(0)=C$, and since according to its plan project $G(N,A)$ has to be accomplished not later that at $t=D$ together with its full budget utilization, we determine the planned trajectory curve (first iteration) $V^{pl}(t)^{(1)}$ by a straight line connecting the two points with coordinates $[0,C]$ and $[D,0]$. Thus, we obtain

$$V^{pl}(t)^{(1)} = C - t \cdot \frac{C}{D}, \quad (4.1.6)$$

which is used within interval $t \in [0, t_1]$, up to the first control point.

Note that no restrictions are imposed on the project's actual cost-duration except for the fact such a function has to be continuous and decreasing.

If, at a routine control point $t_g > 0$, it is observed that $C(t_g) \le V^{pl}(t_g)^{(q)}$ (q-th iteration) there is no need for any interference in the project's realization, since the project meets in fact a stricter chance constraint than required. Thus, the progress of the project proceeds, trajectory curve $V^{pl}(t)^{(q)}$ remains unchanged, and the next control point has to be determined. If, on the contrary, inequality $C(t_g) > V^{pl}(t_g)^{(q)}$ is observed, an error signal has to be generated, and we have to examine the project in greater detail. Optimization problem at the higher level has to be solved, in order to calculate the maximal probability of the project meeting its deadline on time without any additional help from other projects. If solution p_t satisfies $p_t \ge p^*$, then new budget values $c(i,j)$ obtained by that solution are reallocated among activities (i,j). A corrected planned trajectory curve $V^{pl}(t)^{(q+1)}$ ($(q+1)$-th iteration) has to be determined by a straight line connecting two points with coordinates $\left[t_g, C(t_g)\right]$ and $[D,0]$. The corresponding trajectory curve to be used within the interval $\left[t_g, t_{g+1}\right]$ is as follows:

$$V^{pl}(t)^{(q+1)} = \frac{D \cdot C(t_g)}{D - t_g} - t \cdot \frac{C(t_g)}{D - t_g}. \quad (4.1.7)$$

It can be clearly recognized that in the course of the project's realization its actual cost-duration function (irrespective of any assumption on that function) is approximated closer and closer by repeatedly corrected trajectory curves (4.1.6-4.1.7) between adjacent control points.

Since minimizing the number of future control points results in maximizing the time span between two routine adjacent control points t_{g+1} and t_g, the problem at hand is to maximize the value

$$\delta_g = t_{g+1} - t_g \quad (4.1.8)$$

subject to (4.1.3-4.1.5).

Denoting $V^{pl}(t) - V^f(t) = H(t)$, we substitute optimization problem (4.1.3-4.1.5,4.1.8) for:

$$\max_{C(t_g)} \delta_g \quad (4.1.9)$$

subject to (4.1.3-4.1.4) and

$$\Pr\{H(t) \ge 0\} \ge p^*. \quad (4.1.10)$$

Let us examine random variable $H(t)$, $t > t_g$, in greater detail. Since each activity time duration $t(i,j) \in G(t)$ is a random variable with a density function dependent on budget value $c(i,j)$, random variable $H(t)$ is a result of multiple random disturbances. Thus, it is reasonable to assume that $H(t)$ has a normal distribution with parameters $E[H(t)]$ and $V[H(t)]$. Note that both these values can be easily simulated to calculate their corresponding unbiased and consistent estimates

$$\overline{H}(t)=\frac{1}{M}\sum_{r=1}^{M}H^{(r)}(t), \tag{4.1.11}$$

$$S^2[H(t)]=\frac{1}{M-1}\sum_{r=1}^{M}\left[H^{(r)}(t)-\overline{H}(t)\right]^2, \tag{4.1.12}$$

where M is the number of simulation runs and $H^{(r)}(t)$ is the value $H(t)$ obtained by the r-th simulation.

Note that chance constraint (4.1.10) can be written in another form

$$\phi(q)\geq p^*, \tag{4.1.13}$$

where

$$\begin{cases} q=-\dfrac{\overline{H}(t)}{S[H(t)]}, \\ \phi(x)=\dfrac{1}{\sqrt{2\pi}}\displaystyle\int_x^{\infty}\exp\left\{-\frac{u^2}{2}\right\}du. \end{cases} \tag{4.1.14}$$

According to (4.1.9) and (4.1.10) the maximal value T^* satisfying

$$\begin{cases} T^*=\max\limits_{t_g\leq t\leq D}\left\{t:\left[\phi(q)\geq p^*\right]\right\}, \\ t_{g+1}-t_g\geq\Delta, \end{cases} \tag{4.1.15}$$

should be determined as the next control point t_{g+1}.

In practice, t_{g+1} can be calculated by means of simulation with a constant step of length Δ. The procedure of consecutively increasing value $t>t_g$ step-by-step is followed until restriction (4.1.15) ceases to hold. Thus problem (4.1.3-4.1.4, 4.1.9-4.1.10) can be solved via simulation in order to capture the last moment before the project deviates from its target.

The on-line algorithm to determine the next control point t_{g+1} is outlined below.

4.1.4 The on-line algorithm to determine the next control point

The algorithm determines at each control point t_g for project $G(t_g)$ the next control point t_{g+1}. The step-wise procedure of the algorithm is as follows:

Step 0. Given at time $t=t_g$:

- the remaining budget $C(t_g)$ which is not utilized at moment $t=t_g$;
- the project's due date D;
- the remaining network of the project $G(t_g)$;
- values $A(i,j)$ and $B(i,j)$ for activities $(i,j)\in G(t_g)$;

- budget values $c(i,j)$ for activities $(i,j) \in G(t_g)$, $\sum_{(i,j)\in G(t_g)} c(i,j) = C(t_g)$;
- minimal confidence probability p^*.

Step 1. Determine the project's planned trajectory:

$$V^{pl}(t) = \frac{D \cdot C(t_g)}{D - t_g} - t \cdot \frac{C(t_g)}{D - t_g}, \quad t \ge t_g.$$

Step 2. Determine the minimal value of the next control point t_{g+1}

$$T^* = t_g + \Delta.$$

Step 3. If $T^* \ge D$ go to Step 15. Otherwise apply the next step.

Step 4. Simulate random time duration for each remaining activity $(i,j) \in G(t_g)$. Simulating p.d.f. (4.1.1)

$$f_{ij}(x) = \frac{12}{\left[\frac{B(i,j)}{c(i,j)} - \frac{A(i,j)}{c(i,j)}\right]^4} \left[x - \frac{A(i,j)}{c(i,j)}\right]\left[\frac{B(i,j)}{c(i,j)} - x\right]^2$$

can be easily implemented by transformation

$$\frac{x - \frac{A(i,j)}{c(i,j)}}{\frac{B(i,j)}{c(i,j)} - \frac{A(i,j)}{c(i,j)}} = z$$

and then using the classical Neumann method [43] for simulating a standard beta distribution with density function $f_z(x) = 12x(1-x)^2$. After simulating the standard beta value z, we may calculate the desired random value $t(i,j)$ by

$$t(i,j) = z \cdot \left[\frac{B(i,j)}{c(i,j)} - \frac{A(i,j)}{c(i,j)}\right] + \frac{A(i,j)}{c(i,j)}.$$

Step 5. Single out all the activities entering the remaining network that have been accomplished (according to their simulated time durations) up to the moment $T^* + \Delta$. If, at that moment, an activity is under way but has not yet been finished, calculate the ratio of the time the activity has actually been in progress and the simulated activity duration.

Step 6. Calculate the summarized amount of budgets $c(i,j)$ for activities singled out at Step 5. For the partially operated activities we include the corresponding ratio of their budgets. Denote the utilized budget within the time period $\left[t_g, T^* + \Delta\right]$ by $\Delta C(t_g, T^* + \Delta)$.

Step 7. Calculate value $H(T^* + \Delta) = V^{pl}(T^* + \Delta) + \Delta C(t_g, T^* + \Delta) - C(t_g)$.

Step 8. Repeat the procedure of Steps 4-7, M times independently (M - sufficiently large number of simulation runs). Denote by $H^{(r)}(T^* + \Delta)$ value $H(T^* + \Delta)$ obtained by the r-th simulation run, $1 \le r \le M$.

Step 9. Calculate value $q = -\frac{\overline{H}(T^* + \Delta)}{S[H(T^* + \Delta)]}$, where

$$\overline{H}(T^* + \Delta) = \frac{1}{M}\sum_{r=1}^{M} H^{(r)}(T^* + \Delta),$$

$$S^2[H(T^* + \Delta)] = \frac{1}{M-1}\sum_{r=1}^{M}\left[H^{(r)}(T^* + \Delta) - \overline{H}(T^* + \Delta)\right]^2.$$

Step 10. If relation $\phi(q) \geq p^*$ holds apply the next step. Otherwise go to Step 13.

Step 11. If $T^* + \Delta \geq D$ go to Step 15. Otherwise apply the next step.

Step 12. Increase value T^* by Δ, i.e. $T^* + \Delta = T^*$. Return to Step 4.

Step 13. Value T^* is the next control point t_{g+1}. Inspect the project at moment T^*.

Step 14. Check inequality $C(t_{g+1}) \leq V^{pl}(t_{g+1})^{(q)}$. If inequality holds, return to Step 1 without changing the planned trajectory. Otherwise apply the higher level of the multilevel on-line control model. Solve the appropriate optimization problem and determine new planned trajectory $V^{pl}(t_{g+1})^{(q+1)}$. Update the information available at Step 0. Proceed to Step 1.

Step 15. Inspect the project at moment D. If the target has not been reached apply the higher hierarchical level. Otherwise the project's goal is reached and the algorithm terminates.

Note that when increasing value T^* by Δ and applying Step 4 (from Step 12), we must not simulate time durations $t(i,j)$ anew. All calculations and decision-makings on Steps 5-10 for increasing values T^*, $T^* + \Delta$, $T^* + 2\Delta$,…, are carried out on the basis of the M simulation runs that have already been realized for the minimal time value T^* (see Step 8). It goes without saying that at the new routine control point t_{g+1} we reapply all the simulations for a revised network.

§4.2 On-line control model with variable speeds

4.2.1 Introduction

Consider an activity-on-arc network project of PERT type with random activity durations. The accomplishment of each activity is measured in percentages of the total project. Each activity can be processed at several possible speeds that are subject to random disturbances. The number of possible speeds is common to all activities. For each activity, speeds are sorted in ascending order of their average values - namely, speeds are indexed. It is assumed that at any moment $t > 0$ activities, that are operated at that moment, have to implement speeds of one and the same index. That is to say, for all teams working simultaneously, the shift has to be of equal length. It can be clearly recognized that the index of the speed indicates the speed that actually determines the project's realization.

The progress of the project can be evaluated only by means of inspection at control points that have to be determined. Assume, further, that the project's speed can be changed only at a control point, that is, all the project's activities, being realized between two adjacent control points, have to be operated with speeds of one and the same index. There exists only one exception: if an activity *must continue in operation* through a control point, the activity speed cannot be changed.

The project's due date and the minimum permissible probability of meeting the deadline on time are both pregiven.

Two basic concepts have to be implemented in the on-line control model:

1) the number of control points has to be minimized, since inspecting the progress of the project is a complicated and costly procedure;

2) the project should avoid unnecessary surplus speeds, since repeated and lengthy work at higher speeds can prematurely wear out resources utilized in the project.

In the further outlined control model, a stochastic control problem is formalized and solved at each control point. Two conflicting objectives are imbedded in the model:

1) to minimize the number of control points, and
2) to minimize the average index of the project's speeds within the planning horizon.

At each routine control point, decision-making centers on determining the next control point and the new index of the speeds (for all activities to be operated) to be employed up to that point.

This section is a further development of §4.1 where an on-line control model has been suggested for a PERT-COST project. That model, however, cannot be applied to projects with variable speeds.

We will outline an on-line control model and will describe the mathematical formulation of the optimization problem that is imbedded in the model. The solution of the problem enables control actions to be taken at inspection points to meet the project's due date on time. A heuristic algorithm is outlined, its efficiency is evaluated by means of simulation. In *4.2.5*, the algorithm to determine the speed at the next routine control point is presented. *4.2.6* deals with the subproblem of determining the next control point for on-line control.

4.2.2 Notation

Let us introduce the following terms:

G_t - a PERT type project at moment $t \geq 0$, $G_0 = G$;

$(i,j) \in G$ - the project's activity;

$\upsilon_{ij}^{(k)}$ - the k-th speed to process activity (i,j), $1 \leq k \leq m$;

m - number of possible speeds;

$t_{ij}^{(k)}$ - random duration of activity (i,j) when processed at speed $\upsilon_{ij}^{(k)}$;

$a_{ij}^{(k)}$ - lower bound of random activity duration $t_{ij}^{(k)}$ (pregiven);

$b_{ij}^{(k)}$ - upper bound of random activity duration $t_{ij}^{(k)}$ (pregiven);

ρ_{ij} - percentage of activity (i,j) in the entire project (pregiven);

D - the due date of the project (pregiven);

p - least permissible probability of meeting the project's due date on time (pregiven);

$N(t)$ - the remaining number of control points to inspect the progress of the project, beginning with moment t; $N(0) = N$ (total number of control points);

t_g - the g-th control point, $g = 0,1,..., N$; $t_0 = 0$, $t_N = D$;

s_g - the index of the project's speed to be implemented within interval $\left[t_g, t_{g+1}\right]$, $1 \le s_g \le m$;

$\bar{k}(t_g) = \frac{1}{D - t_g} \sum_{\tau = t_g}^{N(t_g) - 1} s_\tau (t_{\tau+1} - t_\tau)$ - the average index of the project's speeds beginning with control point t_g, $g \ge 0$;

$k^* \subset \{1, m\}$ - the lower bound of surplus speeds. The latter can be used only in an emergency to enable meeting the deadline subject to the chance constraint;

Δ - the minimal time span between two consecutive control points t_g and t_{g+1} in order to force convergence (pregiven);

$\Pr\{G_t, s_g\}$ - probability that the project will reach its due date D on time, on condition that from moment t until D, only speed with index s_g will be used throughout;

$V^f(t)$ - actual project output (in percentages of the entire project) observed at moment $t \ge 0$, $V^f(0) = 0$;

V^{pl} - the planned entire project volume (pregiven).

Similar to relation (4.1.1), assume random activity duration $t_{ij}^{(k)}$ distributed by the beta-law with p.d.f.

$$f_{ij}^{(k)}(x) = \frac{12}{\left[b_{ij}^{(k)} - a_{ij}^{(k)}\right]^4} \left[x - a_{ij}^{(k)}\right] \left[b_{ij}^{(k)} - x\right]^2. \qquad (4.2.1)$$

The initial data of the control model for each activity (i, j) includes:

i; j; ρ_{ij}; $a_{ij}^{(1)}$; $b_{ij}^{(1)}$;...; $a_{ij}^{(m)}$; $b_{ij}^{(m)}$.

Note that for each activity (i, j), the speed indices k are sorted in descending order of the corresponding average values $\bar{t}_{ij}^{(k)}$, namely, $k_1 > k_2$ results in $\bar{t}_{ij}^{(k_1)} < \bar{t}_{ij}^{(k_2)}$.

Besides beta-distribution with p.d.f. (4.2.1), the outlined control model may adopt additional alternative probability distributions. A variety of probability distributions from which to choose for activity durations is outlined in Chapter 2, as well as in [54,105-106,163]. In order to evaluate the performance of the control model, we will outline below comparative results with other practically used distributions, namely, the uniform and the normal ones.

4.2.3 The control model

In §4.1, we presented on-line procedures for a PERT-COST project. But we did not consider the possibility of several operating speeds, and all the model's parameters were based on cost values. This is the focal point of the main differences from the on-line control model outlined below.

Several basic concepts are imbedded in the model:

a) the model comprises a chance constraint to meet the deadline;
b) the total number of inspection point has to be kept to a minimum to make the control less costly and less difficult;
c) since operating at higher speeds (i.e., with higher intensities) is always costlier than at slower ones, the average index of the project's speeds has to be minimized also.

The related optimization problem is therefore as follows:

At any routine control point t_g, determine values t_{g+1} and s_g to minimize both:

- the number of future control points

$$\min_{\{t_{g+1},s_g\}} N(t_g) \tag{4.2.2}$$

- and the average index of future project's speeds

$$\min_{\{t_{g+1},s_g\}} \bar{k}(t_g) \tag{4.2.3}$$

subject to

$$t_{g+1} - t_g \geq \Delta\,, \tag{4.2.4}$$

$$t_{g+1} - t_g = \Delta \quad \forall g : s_g \geq k^*\,, \tag{4.2.5}$$

$$D - t_{g+1} \geq \Delta\,, \tag{4.2.6}$$

$$t_0 = 0\,, \tag{4.2.7}$$

$$t_N = D\,, \tag{4.2.8}$$

$$\Pr\left\{G_{t_g}, s_g - 1\right\} < p \leq \Pr\left\{G_{t_g}, s_g\right\}. \tag{4.2.9}$$

Restriction (4.2.9) means that at each control point t_g, the problem is to determine the minimal index of the project's speed that, with the given chance constraint, guarantees meeting the project's due date on time. Thus, the restrictions prohibits unnecessarily high speeds.

Restriction (4.2.5) means that if decision variable s_g refers to a surplus speed the latter has to be implemented within a restricted time interval of length Δ. Afterwards the project has to be inspected anew. Thus, both restrictions (4.2.5) and (4.2.9) encourage using possible slower speeds but honoring the chance constraint of meeting the project's due date on time.

Problem (4.2.2-4.2.9) is a stochastic optimization problem with two conflicting objectives, a non-linear chance constraint and a random number of optimized variables. It can be well-recognized that such a problem is too difficult to be solved in the general case. A heuristic solution will be outlined below.

The general idea to solve the problem heuristically is as follows: at each routine control point t_g, we observe the remaining network project G_{t_g} and the actual project output $V^f(t_g)$ (on the basis of values ρ_{ij}). Decision variable s_g is determined as the minimal speed satisfying (4.2.9). Its calculation is carried out by means of simulation: graph G_{t_g} has to be repeatedly simulated with different project speeds in order to determine

$$s_g = \min_{1 \le k \le m} \left\{ k : \left[\Pr\left\{ G_{t_g}, k \right\} \ge p \right] \right\}. \tag{4.2.10}$$

The algorithm to determine s_g is outlined in *4.2.5*.

The next control point t_{g+1} is determined as follows. If $s_g \ge k^*$, then $t_{g+1} = t_g + \Delta$. Otherwise, in case $1 \le s_g \le k^* - 1$, the next control point t_{g+1} is determined by solving stochastic optimization problem

$$\max \left\{ t_{g+1} \right\} \tag{4.2.11}$$

subject to (4.2.4) and

$$\Pr\left\{ V^f(t) > V^{pl}(t) \right\} \ge p \quad \forall t : t_g \le t \le t_{g+1}, \tag{4.2.12}$$

where $V^{pl}(t)$ denotes the planned trajectory between two adjacent control points t_g and t_{g+1}.

If point $\left[t_g, V^f(t_g) \right]$ is above the straight line connecting points $[0,0]$ and $\left[D, V^{pl} \right]$, namely, $V^f(t_g) > \frac{V^{pl}}{D} \cdot t_g$ holds, then the planned trajectory is that straight line, satisfying equation

$$V^f(t) = \frac{V^{pl}}{D} \cdot t(dx). \tag{4.2.13}$$

Otherwise, in case $V^f(t_g) \le \frac{V^{pl}}{D} \cdot t_g$, the planned trajectory is determined as the straight line connecting points $\left[t_g, V^f(t_g) \right]$ and $\left[D, V^{pl} \right]$, namely

$$V^{pl}(t) = \frac{V^{pl} - V^f(t_g)}{D - t_g} \cdot t + \frac{V^f(t_g) \cdot D - V^{pl} \cdot t_g}{D - t_g}, \quad t_g \le t \le t_{g+1}. \tag{4.2.14}$$

Case $V^f(t_g) > \frac{V^{pl}}{D} \cdot t_g$ means that the project is carried out under favorable circumstances, and it is therefore reasonable to bring the state variable closer to the planned trajectory (4.2.13) by introducing slower speeds, and then to capture the last moment before the project output drops below the trajectory curve. Case $V^f(t_g) \le \frac{V^{pl}}{D} \cdot t_g$ means that there is danger the project will not meet its deadline. Thus, higher speeds have to be implemented to keep the state variable $V^f(t)$ above the new trajectory (4.2.14).

Objective (4.2.11) is obvious, since minimizing the number of control points results in maximizing the time span between two adjacent control points. Thus, both objectives (4.2.2) and (4.2.3) of the general problem (4.2.2-4.2.9) are implemented in the heuristic solution via (4.2.9) and (4.2.11). The solution of problem (4.2.4, 4.2.11-4.2.12) is outlined in *4.2.6*.

4.2.4 Heuristic algorithm

The algorithm outlined below determines at each control point t_g the minimal speed s_g that, with pregiven probability p, guarantees completion of the project on time. The next control point t_{g+1} is determined by maximizing the time span between two adjacent control points t_{g+1} and t_g, honoring restriction (4.2.12), on condition that s_g does not correspond to a surplus speed. It can be well-recognized that chance constraint (4.2.12) is, in essence, stricter than (4.2.10). The latter only ensures that the project will meet its deadline with probability not less than p, while chance constraint (4.2.12) enables the state variable $V^f(t)$ to exceed the planned trajectory $V^{pl}(t)$ at any moment t within the interval $\left[t_g, t_{g+1}\right]$.

The step-wise procedure of the algorithm is as follows:

Step 1. Start with $g=0$, $t_0=0$, $V^f(t_0)=0$.

Step 2. Determine the project's speed:
find s_g satisfying constraint (4.2.10). The corresponding algorithm, which we will henceforth call Algorithm I, is outlined in *4.2.5*.
If chance constraint (4.2.10) does not hold for all s_g, $1 \le s_g \le m$, set $s_g = m$ and go to Step 6.

Step 3. Determine the next control point t_{g+1}. The corresponding Algorithm II is outlined in *4.2.6*.

Step 4. Monitor closeness to the due date and to the next control point:
If $D - t_{g+1} \le \Delta$, then set $t_{g+1} = D$ and go to Step 6.
If $t_{g+1} - t_g < \Delta$, then set $t_{g+1} = t_g + \Delta$ and apply the next step.

Step 5. Observe $V^f(t_{g+1})$, set $g = g+1$ and return to Step 2.

Step 6. Observe $V^f(D)$. The algorithm terminates.

The algorithm is implemented in real time; namely, each iteration of the algorithm can be performed only after the project's output $V^f(t_g)$ is actually realized. The control points and the corresponding speeds to be introduced cannot be predetermined. However, if we want to evaluate the efficiency of the control model, that is the probability of completing the total project on time, we can simulate the project's realization by randomly sampling the actual duration of each activity. In such a case, Steps 5 and 6 of the algorithm have to be modified as follows:

Step 5*. Simulate durations of activities (i,j) that belong to the interval $\left[t_g, t_{g+1}\right]$. Single out all the activities entering the remaining project G_{t_g} which will be accomplished (according to their simulated durations) up to the moment t_{g+1}. If, at that moment, an activity is underway but has not yet been accomplished, calculate the ratio of the time the activity has already been in progress, and the entire simulated activity duration. Note, that in the course of operation, no activity can change its speed (even at a control point). Thus, being simulated at the beginning of the activity's realization, its duration remains unchanged. Calculate the summarized amount of values ρ_{ij} for the singled-out activities. For the partially operated activities include the corresponding ratio of their ρ_{ij} values. Denote the partially accomplished amount of the project within the time period $\left[t_g, t_{g+1}\right]$ by $\Delta V\left[t_g, t_{g+1}\right]$. Calculate the simulated project's output $V^f\left(t_{g+1}\right)$ as follows:

$$V^f\left(t_{g+1}\right) = V^f\left(t_g\right) + \Delta V\left[t_g, t_{g+1}\right]. \qquad (4.2.15)$$

Note that value $V^f\left(t_g\right)$ has been determined before, at the previous control point t_g. Afterwards set $g = g+1$ and return to Step 2.

Step 6* is implemented in a way similar to Step 5, with the exception that after determining $V^f(D)$ step 2 is *not* applied, and the simulation run terminates.

By simulating the development of the project many times, the probability of meeting its due date on time, the average number of control points and the average index of the project's speeds within the planning horizon may be evaluated.

4.2.5 Algorithm I to determine the minimal speed subject to the chance constraint

The problem is to determine the minimal index of the speed s_g to be implemented at each routine control point t_g (see objective (4.2.10)). The step-wise procedure of the algorithm is as follows:

Step 0. Given at time $t = t_g$:

- the remaining network project G_{t_g};
- the project's due date D;
- minimal confidence probability p;
- values ρ_{ij} for activities $(i,j) \in G_{t_g}$;

- upper and lower bounds $a_{ij}^{(k)}$, $b_{ij}^{(k)}$, $1 \le k \le m$, $(i,j) \in G_{t_g}$.

Step 1. For all activities $(i,j) \in G \setminus G_{t_g}$, which have been already completed, set their duration values t_{ij} obtained by means of simulation at previous control points. These values remain fixed and unchanged within the planning horizon.

Step 2. Set $k = 1$.

Step 3. Simulate random durations $t_{ij}^{(k)}$ for all activities $(i,j) \in G_{t_g}$ (besides activities which are underway and have been already simulated) using beta-distribution (4.2.1) or alternative density distributions.

Step 4. Calculate the critical path length $L_{cr}\left[t_{ij}^{(k)}\right]$ of network G with activity durations determined at Steps 1 and 3.

Step 5. If $L_{cr}\left[t_{ij}^{(k)}\right] \le D$ counter $Q+1=Q$ works and afterwards proceed to Step 6.

If $L_{cr}\left[t_{ij}^{(k)}\right] > D$ apply the next step.

Step 6. Repeat Steps 3-5 R times to obtain representative statistics.

Step 7. Calculate ratio $p^* = Q/R$; if $p^* \ge p$ go to Step 11. Otherwise apply the next step.

Step 8. Counter $k+1=k$ works.

Step 9. If $k \le m$ clear counter Q and return to Step 3. Otherwise apply the next step.

Step 10. Applying this step means that even implementing the maximal speed cannot guarantee for the project to meet its deadline honoring chance constraint (4.2.9). An emergency is declared, and the highest speed υ_m is enforced for all activities up to moment , i.e., without intermediate control points.

Step 11. Set $s_g = k$. The algorithm terminates.

Note that the simulated values $t_{ij}^{(k)}$, $(i,j) \in G_{t_g}$, will be used later on to determine the next control point (see Algorithm II in *4.2.6*).

4.2.6 Algorithm II to determine the next control point

According to the general idea of on-line control outlined in §4.1 [64,68], value t_{g+1} is determined as the maximal value satisfying

$$\begin{cases} t_{g+1} = \max\limits_{t_g \le t \le D} \left\{ t : [\phi(\tau_t) \ge p] \right\}, \\ t_{g+1} - t_g \ge \Delta, \end{cases} \tag{4.2.16}$$

where

$$\phi(\tau_t) = \frac{1}{\sqrt{2\pi}} \int_{\tau_t}^{\infty} \exp\left\{-\frac{u^2}{2}\right\} du, \tag{4.2.17}$$

$$\tau_t = -\frac{\overline{H}(t)}{S[H(t)]}. \qquad (4.2.18)$$

Here

$$\overline{H}(t) = \frac{1}{R}\sum_{r=1}^{R} H^{(r)}(t), \qquad (4.2.19)$$

$$S^2[H(t)] = \frac{1}{R-1}\sum_{r=1}^{R}\left[H^{(r)}(t) - \overline{H}(t)\right]^2, \qquad (4.2.20)$$

$$H(t) = V^f(t) - V^{pl}(t), \qquad (4.2.21)$$

where is the number of simulation runs (equal to that used in Algorithm I outlined in *4.2.5*) and $H^{(r)}(t)$ denotes value $H(t)$ obtained by the r-th simulation. Control point t_{g+1} is calculated by means of simulation with a constant step of length Δ. The procedure of consecutively increasing value $t > t_g$ step-by-step is followed until restriction (4.2.16) ceases to hold. Value $V^f(t)$ is evaluated by using Step 5 of the heuristic algorithm outlined in *4.2.4*; consecutive points $t_g + \Delta, t_g + 2\Delta, \ldots$ are considered, in order to capture the last moment before the project deviates from its trajectory.

The step-wise procedure of Algorithm II is as follows:

Step 0. Given at time $t = t_g$:

- the remaining network project G_{t_g};
- the project output $V^f(t_g)$ observed by inspection at control point t_g;
- the project's planned trajectory $V^{pl}(t)$ determined by (4.2.13) or (4.2.14);
- the project's due date D;
- minimal confidence probability p;
- the index of speed $s_g = k$ to carry out the project up to the next control point;
- values ρ_{ij} for activities $(i, j) \in G_{t_g}$;
- simulated values $t_{ij}^{(k)}$, $(i, j) \in G_{t_g}$, obtained at Steps 3-5 of Algorithm I outlined in *4.2.5*.

Step 1. If $s_g \geq k^*$ apply the next step. Otherwise go to Step 3.

Step 2. Set $t_{g+1} = t_g + \Delta$. Proceed to Step 4 of the heuristic algorithm outlined in *4.2.4*.

Step 3. If $V^f(t_g) < \frac{V^{pl}}{D} \cdot t_g$ go to Step 5. Otherwise apply the next step.

Step 4. Determine the planned trajectory $V^{pl}(t)$ by (4.2.14). Proceed to Step 6.

Step 5. Determine the planned trajectory $V^{pl}(t)$ by (4.2.13). Proceed to Step 6.

Step 6. Determine the minimal value of the next control point t_{g+1}: $t_{g+1}^* = t_g + \Delta$.

Step 7. If $D - t^*_{g+1} < \Delta$ go to Step 16. Otherwise apply the next step.

Step 8. Calculate for the r-th simulation run, $1 \le r \le R$, value $\Delta V^{(r)}\left[t_g, t^*_{g+1} + \Delta\right]$ according to Step 6 of the heuristic algorithm outlined in *4.2.4*.

Step 9. Calculate value

$$H^{(r)}\left(t^*_{g+1} + \Delta\right) = V^f\left(t_g\right) + \Delta V^{(r)}\left[t_g, t^*_{g+1} + \Delta\right] - V^{pl}\left(t^*_{g+1} + \Delta\right).$$

Step 10. Repeat Steps 8-9 R times to obtain representative statistics.

Step 11. Calculate value

$$\tau_{t^*_{g+1}+\Delta} = -\frac{\overline{H}_k\left(t^*_{g+1} + \Delta\right)}{S\left[H_k\left(t^*_{g+1} + \Delta\right)\right]},$$

where

$$\overline{H}_k\left(t^*_{g+1} + \Delta\right) = \frac{1}{R}\sum_{r=1}^{R} H^{(r)}\left(t^*_{g+1} + \Delta\right),$$

$$S^2\left[H_k\left(t^*_{g+1} + \Delta\right)\right] = \frac{1}{R-1}\sum_{r=1}^{R}\left[H^{(r)}\left(t^*_{g+1} + \Delta\right) - \overline{H}\left(t^*_{g+1} + \Delta\right)\right]^2.$$

Step 12. If inequality $\phi\left(\tau_{t^*_{g+1}+\Delta}\right) \ge p$ holds, apply the next step. Otherwise go to Step 15.

Step 13. If $t^*_{g+1} + \Delta > D - \Delta$ go to Step 16. Otherwise apply the next step.

Step 14. Increase value t^*_{g+1} by Δ, namely $t^*_{g+1} + \Delta = t^*_{g+1}$. Return to Step 8.

Step 15. Set $t_{g+1} = t^*_{g+1}$. The algorithm terminates.

Step 16. Set $D = t_{g+1}$. The algorithm terminates.

After determining the next control point we apply either Step 5 (in case $t_{g+1} \ne D$) or Step 6 ($t_{g+1} = D$) of the heuristic algorithm outlined in *4.2.4*.

4.2.7 Experimentation and conclusions

Extensive experimentation has been undertaken to check the fitness of the on-line control model outlined in this Chapter. Several distribution laws (beta, normal, uniform) together with different confidence probability p values have been examined. The following conclusions have been drawn from the Chapter [66]:

1) It can be clearly recognized that for all examples examined, the simulated probability $\overline{p}$ of meeting the due date on time exceeds the pregiven confidence probability p. This is because in order to carry out the on-line control, we implement chance constraint (4.2.12) which is stricter than the initial chance constraint (4.2.10).

2) Introducing beta-distribution results in realizing the project in time with slowest speeds and the least number of control points, that is for the case of beta-distribution the control algorithm is more efficient than in other cases. This is because the mean value of beta-distribution (4.2.1) $E\left[t^{(k)}_{ij}\right] = 0.6a^{(k)}_{ij} + 0.4b^{(k)}_{ij}$ is smaller than the mean value $0.5\left[a^{(k)}_{ij} + b^{(k)}_{ij}\right]$ for the

normal and the uniform distributions. For the case of a uniform distribution the average expenses of carrying out the project are higher than for other probability distributions, namely, the project becomes the most costly one.

3) Removing constraint (4.2.5), namely, assuming $k^* = 4$, results for all distributions both in minimizing the number of control points and maximizing the average index of the speeds. Decreasing number k^* results in decreasing the average speed together with increasing the number of inspection points. Thus, a trade-off between the number of control points and the average index of speeds can be achieved by varying k^*.

4) The developed on-line control algorithm can be implemented for activity-on-arc network projects where each activity can be operated at several possible speeds subject to random disturbances and the activity's accomplishment is measured as a part of the entire project. Such projects include construction projects, various R&D projects, etc.

5) The control algorithm is implemented by means of simulation and can be easily programmed on PC, mainly for projects of medium size.

6) On-line control models subject to a chance constraint can be used for other types of PERT network projects, for example, PERT-COST projects comprising budget reallocation problems. Thus, the idea of on-line control has the potential of becoming a general one.

7) The developed on-line control model is a decision-making model that is employed in inspection points only. The model does not revise any existing techniques in project management, e.g., resource reallocation, project scheduling, etc. It is an additional support tool that helps the project manager to determine inspection points and to choose the proper project speed after evaluating the progress of the project at a routine inspection point. Such control actions enhance the prospects of the project to meet its deadline on time, subject to a chance constraint.

8) As outlined above, the average expenses to carry out the project cannot be used directly in the model. However, the developed algorithm enables calculation of those expenses, by means of simulation, for each combination of p and k^*. One has only to attach the processing costs to all speeds per time unit for each activity, together with the cost of performing a single control of the project, with the penalty cost for non-accomplishing the project at the due date. Thus, a wise choice of parameter values p and k^* can be determined to minimize the expenses of carrying out the project.

Chapter 5. Control Models Based on Risk Averse Decision-Making

§5.1 On-line control model

5.1.1 Introduction

We will consider an activity-on-arc network project of PERT type with random activity durations. It is assumed that the progress of the project can only be inspected and measured at preset inspection points since it is impossible or too costly to measure it continuously. The developed and outlined in the previous Chapter on-line control models determine both control (inspection) points and control actions to be introduced at those points in order to alter the progress of the project in the desired direction. The timing of inspection points is carried out by determining planned trajectories that must be repeatedly corrected in the course of the project's realization. On-line control is carried out by solving an optimization problem to minimize the number of control points needed to meet the planned trajectory, subject to the chance constraint. Stated another way, the problem's objective is to maximize the time span between two routine adjacent control points, subject to the chance constraint. The solution of that problem, i.e., determining the next inspection point, is carried out by means of extensive simulation with a constant time step. To consecutively increase the time span value step-by-step, the procedure is as follows. At each intermediate control point, decision-making based on sequential statistical analysis has to be undertaken, either

a) to proceed further and to examine the next control point; or
b) to determine the control point under consideration as the last moment before the project deviates from its target subject to the chance constraint. Thus, the next routine control point is determined.

The main shortcoming of such step-by-step control algorithms is their long computational time due to the need to make numerous decisions. In order to speed up the model's performance, we present an on-line heuristic control algorithm in which the timing of inspection points does not comprise intermediate control points and is based on the behavior of a risk averse decision-maker. Given a routine inspection point t_i, the next point t_{i+1} is determined so that even in the case that the project develops *most unfavorably* in the interval $[t_i, t_{i+1}]$, introducing proper control action at moment t_{i+1} enables the project to meet its target on time, subject to the chance constraint.

The outlined below control algorithm has been tested on several PERT projects of different types, e.g., on PERT-COST projects, construction projects with random activity durations, etc. In all cases the algorithm's computational time has been essentially shortened (by a factor of some 25-30) in comparison to step-by-step control procedures. The developed algorithm provides better solutions than would be attained by using on-line sequential statistical analysis.

5.1.2 Notation

Let us introduce the following terms:

$G(N, A)$ - activity-on-arc network project (graph) of PERT type;

D - the due date of the project (pregiven);

p^* - least permissible probability of meeting the project's due date on time (pregiven);

V^* - the project's target, i.e., the planned total project volume (pregiven);

G_t - the remaining project $G(N, A)$ at moment $t \geq 0$; $G_0 = G(N, A)$;

V_t - the project's output (project volume) observed at control point $t > 0$;

t_g - the g-th inspection (control) point, $g = 0,1,..., N$, $t_0 = 0$, $t_N = D$ (optimized variable);

υ_t - the project's speed at moment $t \geq 0$ set by the control device (controlled variable);

N - the number of control points within the planning horizon;

$(i, j) \in G_t$ - the project's activity, $t \geq 0$;

t_{ij} - random duration of activity (i, j);

Δ - the minimal time span between two consecutive control points t_g and t_{g+1} in order to force convergence (pregiven);

ρ_{ij} - the weight (contribution) of activity (i, j) in the total project volume (pregiven);

S_{ij} - the actual time activity (i, j) starts;

$F_{ij} = S_{ij} + t_{ij}$ - the actual moment activity (i, j) is finished;

c_{ij} - budget assigned to activity (i, j) (for PERT-COST projects);

$c_{ij\min}$ - minimal possible budget to operate activity (i, j) (pregiven);

$c_{ij\max}$ - maximal possible budget to operate activity (i, j) (for PERT-COST projects, pregiven); in case $c_{ij} > c_{ij\max}$ additional budget is redundant;

C_t - available remaining budget to carry out a PERT-COST project G_t (observed at control point t);

$CA(t, k) = \upsilon_t^{(k)}$ - the k-th control action on project G_t introduced by the control device at moment t to alter the controlled variable υ_t in the desired direction;

$P(t, k)$ - confidence probability to accomplish project G_t on time after introducing control action $CA(t, k)$, $1 \leq k \leq m$;

$m > 1$ - the number of possible control actions;

$W_p(t, k)$ - the p-quantile of the moment project G_t will be finished, on condition that control action $CA(t, k)$ is introduced (time moment to be met with preset probability p);

$p^{**} > p^*$ - probability value being very close to 1 (pregiven).

Note that for some types of network projects, e.g., construction projects, various R&D projects, etc., each activity can be operated at several possible speeds. Introducing a control action results in choosing one of those speeds which corresponds to one and the same resource capacity and depends only on the degree of intensity of the project's realization [66]. The speeds are sorted in ascending order from 1 to m. At each routine control point t_g, a control action boils down to determining the index of the speed k, $k \in \{1, m\}$, for all project activities to be operated with that speed from point t_g, up to the next control point t_{g+1}. An additional constraint is introduced in the control model which enables choosing the *minimal* index k, on condition that the chance constraint to meet the project's deadline on time must be honored [66]. Thus, for that type of project

$$CA(t,k) = k = \min_{1 \le q \le m}\left[q : P(t_g, q) \ge p^* \right]. \tag{5.1.1}$$

For other stochastic network projects, e.g., of PERT-COST type, a control action, if introduced, results in optimal budget reallocation as follows [62,64]: at a routine control point $t_g \ge 0$ determine optimal budget values c_{ij}, $(i,j) \in G_{t_g}$, to maximize the probability of meeting the project's due date on time,

$$P(t_g, k) = \Pr\left\{ t_g + T\left[G_{t_g} \middle| c_{ij}\right] \le D \right\} \ge p^*, \tag{5.1.2}$$

subject to

$$c_{ij\,\min} \le c_{ij} \le c_{ij\,\max}, \tag{5.1.3}$$

$$\sum_{(i,j)\in G_{t_g}} c_{ij} = C_{t_g}, \tag{5.1.4}$$

where $T\left[G_{t_g} \middle| c_{ij}\right]$ denotes the random duration of project G_{t_g} with reallocated budget values c_{ij}. Note that random durations t_{ij} have a density function which depends parametrically on value c_{ij}. For practical cases [92], the project management can adopt any suitable distribution as long as its density function presents a *linkage between time and cost*, e.g., t_{ij} has a beta-distribution with boundary values t_{ij}^* and t_{ij}^{**}, where

$$t_{ij}^* = \frac{A_{ij}}{\left(c_{ij}\right)^\tau} \quad \text{and} \quad t_{ij}^{**} = \frac{B_{ij}}{\left(c_{ij}\right)^\tau}, \tag{5.1.5}$$

$0.5 < \tau \le 1$, and values A_{ij} and B_{ij} are pregiven.

It can be well recognized that a PERT-COST project described above has only two ($m = 2$) possible control actions, namely:

1. The project continues to function without changing budget values c_{ij}, $(i,j) \in G_{t_g}$ if at a routine control point t_g it is anticipated that the project meets its deadline on time under the chance constraint ($k = 1$). A decision

not to change values c_{ij} has to be taken if value $\Pr\left\{t_g + T\left[G_{t_g} \middle| c_{ij}\right] \le D\right\}$ is not less than p^*.

2. Otherwise, i.e., if $P(t_g, 1) < p^*$ holds, the project has to undergo budget reallocation in order to increase the confidence probability of meeting the due date on time ($k = 2$). The reallocation optimization problem (5.1.2-5.1.4) has then to be solved.

5.1.3 The control model

On the basis of particular cases outlined above we will outline a generalized on-line control model [72] as follows: determine optimal control points t_g (to inspect the project) and optimal project's speeds $\upsilon_{t_g}^{(k)}$ to proceed with until the next control point t_{g+1}, in order to minimize the average number of inspection points

$$\min_{\{t_g, \upsilon_{t_g}^{(k)}\}} \overline{N} \tag{5.1.6}$$

subject to (5.1.1) and

$$P(t_g, k) = \Pr\left\{t_g + T\left[G_{t_g} \middle| \upsilon_{t_g}^{(k)}\right] \le D\right\} \ge p^*, \tag{5.1.7}$$

$$t_{g+1} - t_g \ge \Delta, \tag{5.1.8}$$

$$t_0 = 0, \tag{5.1.9}$$

$$t_N = D, \tag{5.1.10}$$

$$D - t_{g+1} \ge \Delta, \quad 1 \le k \le m. \tag{5.1.11}$$

Here, $T\left[G_{t_g} \middle| \upsilon_{t_g}^{(k)}\right]$ is the random duration of the remaining project G_{t_g} after introducing control action $CA(t_g, k)$ at moment t_g.

In order to avoid unnecessarily high speeds, an additional control action (5.1.1) is introduced at each routine inspection point (see §4.2). This means that at each control point t_g the problem is to determine the *minimal index* of the project's speed which will enable meeting the project's due date on time, subject to the chance constraint.

Control model (5.1.6-5.1.11) is suitable for construction and R&D network projects with different speeds. In the case of a PERT-COST project, the project's duration $T\left[G_{t_g} \middle| \upsilon_{t_g}^{(k)}\right]$ is substituted for $T\left[G_{t_g} \middle| c_{ij}\right]$. Budget values c_{ij} are obtained by solving the stochastic optimization problem (5.1.2-5.1.4).

Thus, two basic concepts are imbedded in the on-line control model (5.1.6-5.1.11):

(a) the model comprises a chance constraint to meet the deadline;

(b) the number of control (inspection) points has to be minimized.

Note that the model's solutions are, in essence, on-line control algorithms. If an algorithm enables the project to be completed by the due date with a probability less than p^*, the algorithm is unfeasible and cannot be accepted. An *optimal solution* is a control algorithm which

- is a feasible algorithm, i.e., enables the project to be completed on time with a probability not less than p^*, subject to the chance constraint, and
- offers the minimal average number of control points in comparison with any other feasible control algorithm.

Thus, when comparing two different on-line control algorithms with respect to the model's objective, the first algorithm is considered to be better than the second one if they are both feasible, and if less inspection points are required in the first algorithm than the second.

It has to be pointed out that since the general model incorporates an additional control action (5.1.1), a second objective is, in essence, imbedded in this model, namely, to minimize the average index of the project's speed within the planning horizon. Thus, the developed generalized control model comprises both control models outlined in Chapter 4 [62,64] as specific cases.

The control algorithm comprises two main parts:

Subalgorithm I determines the project's speed $\upsilon_{t_g}^{(k)}$ at moment t_g, i.e., formalizes the control action $CA(t_g,k)$. The subalgorithm is carried out by means of simulation, by determining the minimal speed index k for which restriction (5.1.1) holds. Value $P(t_g,k)$, $1 \le k \le m$, is calculated by simulating $t_g + T\left[G_{t_g}\middle|\upsilon_{t_g}^{(k)}\right]$, on condition that speed $\upsilon_{t_g}^{(k)}$ is used throughout [66].

Subalgorithm II determines the next inspection point t_{g+1} on the basis of the routine control point t_g, the project's output V_{t_g} observed at that point, and the project's speed $\upsilon_{t_g}^{(k)}$ is implemented as a control action. The subalgorithm develops a heuristic solution of the stochastic optimization problem as follows: determine the next inspection point t_{g+1} to maximize the time span between two adjacent control points:

$$\max t_{g+1} \tag{5.1.12}$$

subject to (5.1.8) and

$$\Pr\{V_t \ge V_t^*\} \ge p^* \quad \forall t : t_g \le t \le t_{g+1}, \tag{5.1.13}$$

where V_t^* is the straight line (trajectory curve) connecting two points (t_g, V_{t_g}) and (D, V^*).

Subalgorithm I determines value t_{g+1} by means of simulation with a constant time step Δ. The procedure of consecutively increasing the time span value is carried out by means of sequential statistical analysis at each intermediate control point [66,68,150]. Due to the need to make numerous decisions, carrying out Subalgorithm II takes much computational time, more than is needed to implement Subalgorithm I.

Thus, the problem to speed up the on-line control algorithm centers on developing a faster subalgorithm for determining routine inspection points. The developed heuristic algorithm is outlined below.

5.1.4 Determining next inspection points

Let us consider the general case of an activity-on-arc network project $G(N,A)$ observed at a routine control point t_g. Given:

- remaining network project G_{t_g};
- routine control point t_g to inspect the project;
- the project's output V_{t_g} observed at moment t_g; and
- control action $CA(t_g,k)$ to be introduced at moment t_g up to the next inspection point,

the problem is to determine that next point t_{g+1}. The general approach to solving the problem is presented in Fig. 5.1 and is as follows [72]:

Since control action $CA(t_g,k)$ is determined by realizing Subalgorithm I subject to restriction (5.1.1), the project's speed index k satisfies $P(t_g,k)\geq p^*$. Therefore (see Fig. 5.1) relation $W_{p^*}(t_g,k)\leq D$ holds. Assume that after introducing speed $\upsilon_{t_g}^{(k)}$ *the project will be carried out under the most unfavorable circumstances*. That means that the actual moment the project is accomplished is close to the upper bound of the density distribution of the project's duration (see Fig. 5.1), i.e., the project's target will be reached at $W_{p^{**}}(t_g,k)$ with probability p^{**} being close to one. To illustrate the heuristic Subalgorithm I we will assume a number of such probabilities, e.g., $p^{**}=0.90, 0.93, 0.95, 0.97$, etc. Thus, the straight line HF connecting two points (t_g,V_{t_g}) and $(W_{p^{**}}(t_g,k),V^*)$ can be regarded as the most unfavorable direction of the project's progress (we will henceforth call it the pessimistic line), while the straight line HE connecting two points (t_g,V_{t_g}) and $(W_{p^*}(t_g,k),V^*)$ enables the deadline to be met on time under the chance constraint (we will henceforth call it the optimistic line). Let us draw a line through the target point (D,V^*) parallel to the optimistic line HE, until the intersection with the pessimistic line at point H' (see Fig. 5.1). It can be clearly recognized that, on condition that control action $CA(t_g,k)$ on the project is introduced and, due to random disturbances, the project advances to its target, until the intersection point H', with the minimal speed, then, from that point on, the project can meet its target on time under the chance constraint $\Pr\left\{t_g+T\left[G_{t_g}\middle|\upsilon_{t_g}^{(k)}\right]\leq D\right\}\geq p^*$. Thus, the abscissa of the intersection point is determined as the next inspection point t_{g+1}.

Note that such a heuristic procedure enables the project to meet its the due date on time with a probability which *exceeds* the pregiven lower boundary p^*. If, indeed, the project will be realized between two adjacent control point t_g and t_{g+1} *under most unfavorable circumstances, i.e., with the minimal rate*, then, beginning from the control point t_{g+1}, the project will meet its target on time under the chance constraint. But actual completion of the project with the minimal rate is extremely rare since the probability, $1-p^{**}$, of such an occurrence is close to zero. Thus, with probability p^{**} close to one, the project has a higher chance than it would have had with p^*, of meeting its deadline on time.

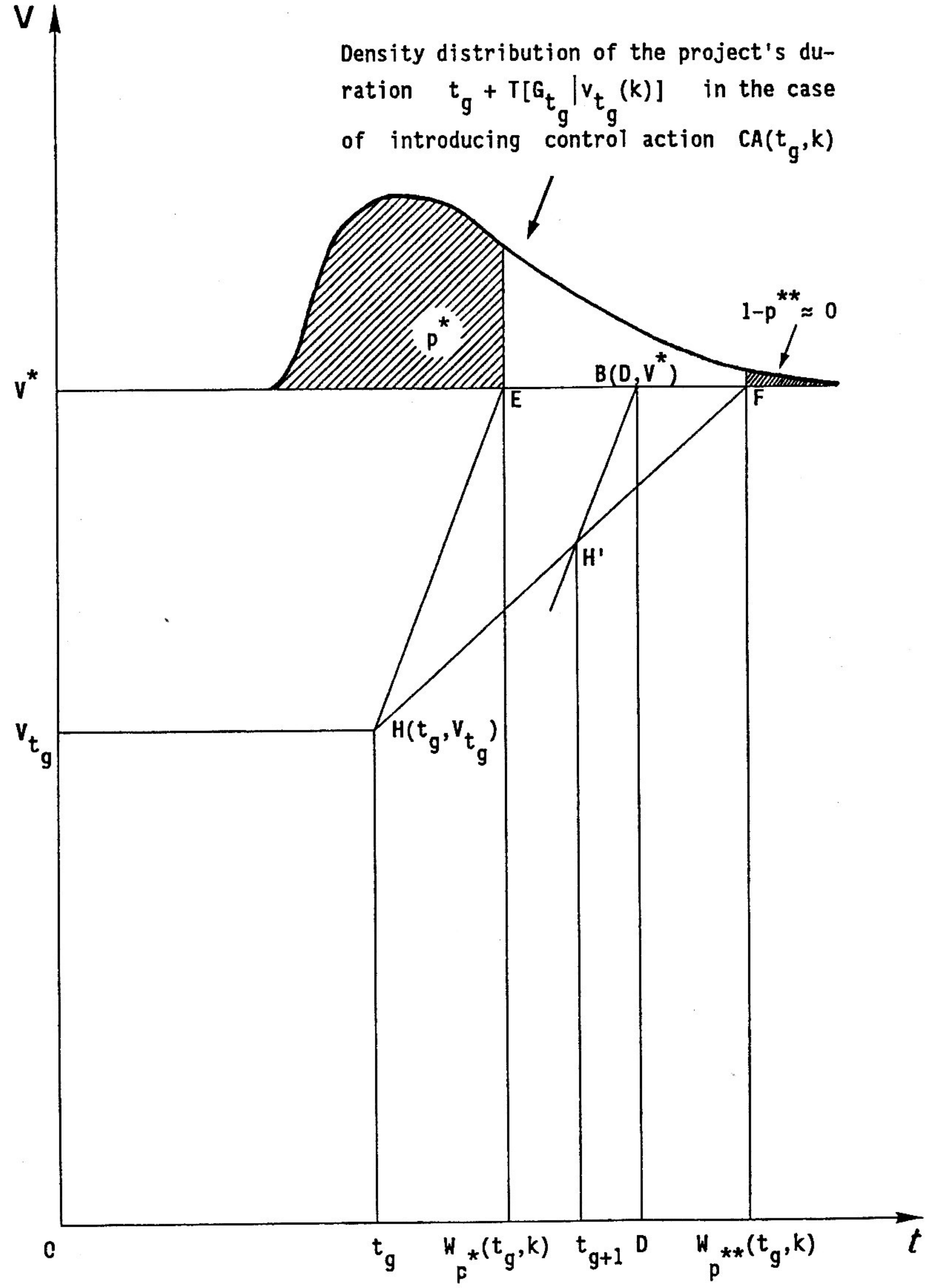

Figure 5.1. *The general idea of determining the next inspection point by risk averse decision-making*

The heuristic algorithm outlined below fits the assumption of a risk averse decision maker. Although it is designed to honor the chance constraint which practically enables completion on time, it does not trust the random circumstances and assumes the worst until the next inspection point.

The step-by-step heuristic algorithm is as follows:

Step 1. Simulate random durations t_{ij}, $(i,j)\in G_{t_g}$, according to the speed $\upsilon_{t_g}^{(k)}$ to be engaged at moment t_g.

Step 2. Simulate (on the basis of calculating the critical path length) the project's duration value $t_g + T\left[G_{t_g}\middle|\upsilon_{t_g}^{(k)}\right]$.

Step 3. Repeat Steps 1-2 M times to obtain representative statistics.

Step 4. Calculate values $W_{p^*}(t_g,k)$ and $W_{p^{**}}(t_g,k)$ on the basis of statistics obtained on Step 3.

Step 5. Determine the optimistic line drawn between two points H and E (see Fig. 5.1)

$$y = \frac{V^* - V_{t_g}}{W_{p^*} - t_g}x + \frac{V_{t_g}W_{p^*} - t_g V^*}{W_{p^*} - t_g}. \tag{5.1.14}$$

Step 6. Determine a straight line through the point B (D,V^*) parallel to the optimistic
line (see Fig. 5.1)

$$y = \frac{V^* - V_{t_g}}{W_{p^*} - t_g}x + V^* - D\frac{V^* - V_{t_g}}{W_{p^*} - t_g}. \tag{5.1.15}$$

Step 7. Determine the pessimistic line connecting points H and F (see Fig. 5.1)

$$y = \frac{V^* - V_{t_g}}{W_{p^{**}} - t_g}x + \frac{V_{t_g}W_{p^{**}} - t_g V^*}{W_{p^{**}} - t_g}. \tag{5.1.16}$$

Step 8. Determine the point of intersection H' of lines (5.1.15) and (5.1.16) and denote it (X,Y). If $X - t_g < \Delta$, set $t_{g+1} = t_g + \Delta$. Otherwise X is accepted as the next inspection point t_{g+1}.

In the case of a PERT-COST project (see §4.1) Step 1 has to be modified by simulating random values t_{ij} on the basis of budget values c_{ij} according to time-cost functions (5.1.5). Values c_{ij} are determined in the course of budget reallocation, by solving optimization problem (5.1.2-5.1.4).

5.1.5 Heuristic algorithm of the on-line control model

The algorithm outlined below is performed in real time, by introducing control actions and inspecting the project's output periodically. We will present the algorithm in a general form that can be "tuned" for practically all cases of controlling network projects.

Assume that $m > 1$ different control actions may be introduced at any inspection point to control the project. The step-by-step heuristic control algorithm is as follows:

Step 1. Start with $g = 0$, $t_0 = 0$, $V_{t_0} = 0$.

Step 2. Set $k = 1$.

Step 3. Consider control action $CA(t_g, k)$. For different types of network projects facilitating Step 3 results in carrying out different procedures. For example, for network projects with different speeds, control action $CA(t_g, k)$ results in introducing the k-th speed for all activities beginning from $t = t_g$ [72]. For PERT-COST projects, $CA(t_g, 1)$ means assigning to all activities $(i, j) \in G_{t_g}$ values c_{ij} which have been previously determined at $t = t_{g-1}$, while $CA(t_g, 2)$ means optimal budget reallocation, which will be outlined later, in Chapters 14-15.

Step 4. Simulate random durations t_{ij}, $(i, j) \in G_{t_g}$, on the basis of control action $CA(t_g, k)$.

Step 5. Calculate the critical path length of network G_{t_g}, $L_{cr}[t_{ij}]$, with activity durations
determined at Step 4.

Step 6. Calculate the project's duration $t_g + L_{cr}[t_{ij}]$.

Step 7. Repeat Steps 4-6 M times to obtain representative statistics.

Step 8. Calculate $P(t_g, k)$ by examining the statistical data obtained at Step 7.

Step 9. If $P(t_g, k) \geq p^*$ go to Step 13. Otherwise apply the next step.

Step 10. Counter $k = 1 \Rightarrow k$ works.

Step 11. If $k \leq m$ return to Step 3. Otherwise apply the next step.

Step 12. Applying this step means that even by introducing control action $CA(t_g, m)$ the project cannot meet its due date on time subject to the chance constraint (5.1.7).
An emergency is declared and the project management is faced with introducing additional control actions, e.g., set $k = m$ and $t_{g+1} = D$. Go to Step 19.

Step 13. Calculate values $W_{p^*}(t_g, k)$ and $W_{p^{**}}(t_g, k)$ on the basis of statistical data obtained at Step 7.

Step 14. Determine straight lines (5.1.15) and (5.1.16).

Step 15. Determine the next inspection point t_{g+1} as the abscissa of the intersection point of lines (5.1.15) and (5.1.16).

Step 16. Monitor closeness to the due date and to the next control point:
If $D - t_{g+1} \leq \Delta$, then set $t_{g+1} = D$ and go to Step 19.
If $t_{g+1} - t_g < \Delta$, then set $t_{g+1} = t_g + \Delta$ and apply the next step.

Step 17. Observe project $G_{t_{g+1}}$ at the next inspection point t_{g+1} and determine the project's output $V_{t_{g+1}}$. Various techniques to calculate the project's output at an inspection point by simulating the on-line control algorithm are outlined in Chapter 4. These techniques, in essence, calculate the partially accomplished amount of the project $\Delta V\left[t_g, t_{g+1}\right]$ within the time period $\left[t_g, t_{g+1}\right]$, as follows:

$$\Delta V\left[t_g, t_{g+1}\right] = \Sigma_1 + \Sigma_2 + \Sigma_3 + \Sigma_4,$$

where

$$\Sigma_1 = \sum_{\substack{S_{ij} < t_g \\ t_g < F_{ij} \le t_{g+1}}} \left[\rho_{ij} \cdot \frac{F_{ij} - t_g}{t_{ij}}\right],$$

$$\Sigma_2 = \sum_{\substack{S_{ij} \ge t_g \\ F_{ij} \le t_{g+1}}} \rho_{ij},$$

$$\Sigma_3 = \sum_{\substack{t_{g+1} > S_{ij} \ge t_g \\ F_{ij} > t_{g+1}}} \left[\rho_{ij} \cdot \frac{t_{g+1} - S_{ij}}{t_{ij}}\right],$$

$$\Sigma_4 = \sum_{\substack{S_{ij} < t_g \\ F_{ij} > t_{g+1}}} \left[\rho_{ij} \cdot \frac{t_{g+1} - t_g}{t_{ij}}\right],$$

where t_{ij} are simulated durations of activities (i,j) that start before inspection point t_{g+1} and are not finished before t_g. Thus, value $V_{t_{g+1}} = V_{t_g} + \Delta V\left[t_g, t_{g+1}\right]$.

Step 18. Set $g = g + 1$ and return to Step 2.

Step 19. Observe the project at the due date D. The algorithm terminates.

§5.2 Experimentation

5.2.1 The experimental design

The comparative efficiency of the developed control algorithm can be illustrated by a numerical example. A construction project where partial accomplishments are usually measured in percentages of the total project, is presented. The project's initial data is given in [72]. Each activity can be operated at three possible speeds that are subject to random disturbances and correspond to different hours a day per worker. Thus, a control action, introduced at the inspection point, results in determining the index of the speed. It is assumed that for all construction teams working simultaneously, the shift has to be of equal length; thus, activities being operated simultaneously have to apply speeds of one and the same index. The index k of the speed to be introduced at moment t actually determines the speed $\upsilon_t^{(k)}$ of the project's realization. Two heuristic algorithms to solve on-line control problem (5.1.6-5.1.11) for the construction network project have been examined:

- the control algorithm based on risk-averse decision making (we will henceforth call it RADM), and
- the former algorithm outlined in Chapter 4, based on decision making via sequential analysis (we will henceforth call it SADM).

To verify the comparative efficiency of the developed algorithm various examples were run. The experimental design is presented in Tab. 5.1. Three parameters were varied: distribution of t_{ij} for all project's activities, the least permissible probability p^* of meeting the project's due date on time, and the versions of the control algorithm:

1. RADM with $p^{**} = 0.90$;
2. RADM with $p^{**} = 0.93$;
3. RADM with $p^{**} = 0.95$;
4. RADM with $p^{**} = 0.97$;
5. SADM.

Table 5.1. The experimental design

Parameters	Values given in the experiment	Number of combinations
Due date D	262	1
Minimal time span	10	1
Number of speeds m	3	1
Distribution of t_{ij}	Uniform, normal, beta	3
Desired probability p^*	0.60; 0.75	2
Versions of the control algorithm	RADM with $p^{**} = 0.90$; RADM with $p^{**} = 0.93$; RADM with $p^{**} = 0.95$; RADM with $p^{**} = 0.97$; SADM.	5

Three alternative distributions are considered:

1. Beta distribution with density function
$$f_{ij}(t) = \frac{12}{\left(b_{ij} - a_{ij}\right)^4}\left(t - a_{ij}\right)\left(b_{ij} - t\right)^2;$$
2. Uniform distribution in the interval $\left[a_{ij}, b_{ij}\right]$;
3. Normal distribution with average $\left(a_{ij} + b_{ij}\right)/2$ and variance $\left[\left(b_{ij} - a_{ij}\right)/6\right]^2$.

Thus, a total of 30 combinations have been considered. For each combination, 100 simulation runs were carried out. The number M of statistical trials in Step 7 of the algorithm is 100. Several measures were considered, as follows:

$\overline{N}$ - the average number of control points;
$\overline{p}$ - the average actual probability of meeting the due date on time;
$\overline{k}$ - the average index of the project's speeds within the planning horizon;
$\overline{T}_c$ - average computational time of a simulation run (in seconds).

Value $\overline{k}$ has been calculated as follows: denote k_{rij} the index of the speed to operate activity (i, j) and t_{rij} its simulated duration, for the r-th simulation run, $(i, j) \in G(N, A)$, $1 \le r \le 100$. Value $\overline{k}$ is evaluated by

$$\overline{k} = 0.01 \cdot \sum_{r=1}^{100} \left[\frac{\sum_{(i,j)\in G(N,A)} k_{rij} t_{rij}}{\sum_{(i,j)\in G(N,A)} t_{rij}} \right].$$

The summary of results is presented in Tab. 5.2.

5.2.2 Conclusions

The following conclusions can be drawn from the summary:

1. Both control algorithms are feasible since their average actual probabilities $\overline{p}$ exceed the least permissible level p^*. This complies with the reasons outlined in *5.1.4* and in §4.2.
2. Introducing beta distribution results in carrying out the project with slowest speeds for both algorithms RADM and SADM. Since using slower speeds results in decreasing processing costs per time unit, introducing beta distribution means realizing the project with the smallest expenses. Introducing uniform distribution leads, on the contrary, to the highest average speeds and, thus, increases the expenses in comparison with other distributions.
3. It can be clearly recognized that for *any* combination of distribution of t_{ij} and value p^* there exists at least one algorithm RADM with value $\overline{N}$ less than the value of $\overline{N}$ obtained by using SADM. Thus, a conclusion can be drawn that the newly developed algorithm RADM *is better* than SADM. In the case of the uniform distribution, we recommend to use RADM with $p^{**} = 0.90$, while for the normal and for beta distributions, we suggest $p^{**} = 0.97$.
4. It can be clearly recognized that, for practically all examples, using algorithm SADM results in a higher value of $\overline{p}$ than when using control algorithm RADM, *but at the expense of introducing higher speeds*. Thus, a conclusion can be drawn that although both control algorithms enable the project's deadline to be met on time, subject to the chance constraint, using RADM *results in a cheaper realization*, than by using SADM.
5. The computational time of realizing algorithm RADM is essentially smaller than for SADM. For a project of small size (36 activities), using RADM results in speeding up the on-line control by a factor of about 25

in comparison with SADM. This is because most computational time in SADM is utilized for carrying out decision making at numerous intermediate check points. In RADM these complicated and lengthy techniques have been removed.

Table 5.2. The summary of results

Distribution	p^*	Versions of the control algorithm	Outcome values			
			$\overline{N}$	$\overline{p}$	$\overline{k}$	$\overline{T}_c$
UNIFORM	$p^* = 0.60$	RADM, $p^{**} = 0.90$	2.88	0.68	2.64	10.2
		RADM, $p^{**} = 0.93$	3.35	0.72	2.57	10.1
		RADM, $p^{**} = 0.95$	3.51	0.73	2.53	10.1
		RADM, $p^{**} = 0.97$	3.87	0.70	2.45	10.0
		SADM	5.01	0.84	2.83	254
	$p^* = 0.75$	RADM, $p^{**} = 0.90$	2.87	0.76	2.71	10.2
		RADM, $p^{**} = 0.93$	3.13	0.79	2.71	10.2
		RADM, $p^{**} = 0.95$	3.09	0.80	2.73	10.2
		RADM, $p^{**} = 0.97$	3.24	0.75	2.78	10.3
		SADM	4.95	0.93	2.95	252
NORMAL	$p^* = 0.60$	RADM, $p^{**} = 0.90$	6.48	0.77	2.22	10.4
		RADM, $p^{**} = 0.93$	5.91	0.78	2.27	10.3
		RADM, $p^{**} = 0.95$	5.55	0.79	2.30	10.3
		RADM, $p^{**} = 0.97$	3.93	0.85	2.55	10.1
		SADM	5.08	0.91	2.75	255
	$p^* = 0.75$	RADM, $p^{**} = 0.90$	8.95	0.80	2.26	10.5
		RADM, $p^{**} = 0.93$	7.89	0.86	2.32	10.4
		RADM, $p^{**} = 0.95$	7.48	0.89	2.33	10.4
		RADM, $p^{**} = 0.97$	4.67	0.91	2.57	10.3
		SADM	5.12	0.96	2.97	253
BETA	$p^* = 0.60$	RADM, $p^{**} = 0.90$	3.39	0.67	2.01	10.2
		RADM, $p^{**} = 0.93$	3.38	0.73	2.01	10.2
		RADM, $p^{**} = 0.95$	3.51	0.68	2.02	10.2
		RADM, $p^{**} = 0.97$	3.57	0.73	2.07	10.2
		SADM	5.10	0.89	2.22	256
	$p^* = 0.75$	RADM, $p^{**} = 0.90$	5.47	0.83	2.07	10.3
		RADM, $p^{**} = 0.93$	5.33	0.85	2.16	10.3
		RADM, $p^{**} = 0.95$	4.82	0.87	2.16	10.3
		RADM, $p^{**} = 0.97$	4.03	0.86	2.26	20.2
		SADM	4.98	0.89	2.21	256

5.2.3 Applications

1. The developed on-line control algorithm is high-speed and can be used for controlling stochastic network projects of practically any size.

2. The algorithm is based on the risk averse decision-making approach. Such an approach has been used successfully in other areas of operations management, e.g., in on-line production control [63]. Simulation results obtained for a practical control problem in project management show that such an approach is also a very effective procedure in controlling stochastic network projects [68,72].
3. Within the last two decades extensive research has been undertaken to develop on-line control models for various organization systems under random disturbances, e.g., [54,68,72,92]. It is obvious that controlling a production unit (activity) with several possible speeds under random disturbances requires, on average, 2.5-3 inspections [72], while controlling a flexible manufacturing cell comprising two production units will require 4.3-4.8 inspections [63]. Since a section comprising two activities is essentially less complicated than a network project with 36 activities, the latter needs, at a minimum, the same number of inspection points as the system outlined in [63]. Taking into account that the best RADM algorithms require only 3-4.6 inspection points, on average, the conclusion which can be drawn is that the RADM algorithm developed here *is close to the best solutions of the general on-line control model*. (Since our control algorithm is a heuristic one, we prefer to avoid the term “optimal solution”).

Chapter 6. Control Models Based on Chance Constraint Principle

§6.1 The chance constraint principle

6.1.1 Introduction

In our publications in the recent decade [54,73,77,83-84] we have developed a new class of on-line control models based on the chance constraint principle and applied to solving cost-optimization problems. The previously developed and outlined in Chapters 4-5 on-line control models are not suitable for solving those problems.

Let us take an overview of the general idea of the chance constraint principle, since otherwise it might be not easy to implement on-line chance constraint concepts for control models in stochastic network projecting.

6.1.2 The system's description

The system under consideration produces a single product or a production program that can be measured by a single value, e.g., in percentages of the planned total volume. Such an approach is often used for R&D projects, in mining, etc. The system is subject to a chance constraint, i.e., the least permissible probability of meeting the due date on time is pre-set. The system utilizes non-consumable resources that remain unchanged throughout the planning horizon. There are several alternative processing speeds to realize the program, corresponding to the same given levels of resources and depending only on the degree of intensity of the production process. However, for different speeds, the average processing costs per time unit vary. The evaluation of advancing to the goal, i.e., observing the product's actual output, can be carried out only via timely inspections at pre-set control points. At every inspection (control) point, the decision-maker observes the amount produced and has to determine both, the proper advancement speed and the next control point. Assume that it is prohibited to use unnecessarily high speeds (especially at the beginning of manufacturing the products), unless there is an *emergency situation*, i.e., a tendency to deviate from the target which may cause delay of the completion time. This is because lengthy work at higher speeds when utilizing restricted resources (e.g., manpower employed in two or three shifts, etc.) can prematurely wear out the regarded system. Assume, further, that the inspection and the speed-reset times equal zero. The costs of all processing speeds per time unit, as well as cost of performing a single inspection at the control point, are pregiven.

6.1.3 Notation

Let us introduce the following terms:

- V - the system's plan (target amount);
- D - the due date (planning horizon);
- $V^f(t)$ - the actual output observed at moment t, $0 < t \le D$; $V^f(0) = 0$;
- $C^f(t)$ - the actual accumulated processing and control costs calculated at

moment t, $0 < t \le D$; $V^f(0) = 0$;

- t_i - the i-th inspection moment (control point), $i = 0,1,...,N$;
- N - the number of control points (a random value);
- v_j - the j-th speed, $1 \le j \le m$ (a random value with pregiven probability density function $f_j(v)$);
- $\bar{v}_j$ - the average speed v_j; it is assumed that speeds v_j are sorted in ascending order of the average values and are independent of t;
- m - the number of possible speeds;
- s_i - index of the speed chosen by the decision-maker at control point t_i;
- c_j - the average processing cost per time unit of speed v_j, $1 \le j \le m$ (pregiven); note that $j_1 < j_2$ results in $c_{j1} < c_{j2}$;
- c_{ins} - the average cost of carrying out a single inspection (pregiven);
- Δ - the minimal value of the closeness of the inspection moment to the due date (pregiven);
- d - the minimal given time span between two consecutive control points (in order to force convergence);
- p - the least permissible probability of meeting the due date on time (pregiven);
- a_j - lower bound of random speed v_j;
- b_j - upper bound of random speed v_j;
- $W_p(t,j)$ - the p-quantile of the moment when production program V will be accomplished on condition that speed v_j is introduced at moment t and will be used throughout, and the actual observed output at that moment is $V^f(t)$ (time moment to be met with pre-set probability p); in other words, $W_p(t,j)$ is the p-quantile of random value $\left[t + \left(V - V^f(t)\right)/v_j\right]$.

Values $V^f(t)$, as well as the parameters of the probability density functions $f_j(v)$, $1 \le j \le m$, are given in percentages of the planned target V. We will, henceforth, implement the beta-distribution with density function

$$p_j(v) = \frac{12}{\left(b_j - a_j\right)^4}\left(v - a_j\right)\left(b_j - v\right)^2. \tag{6.1.1}$$

6.1.4 The problem

Let us consider the cost-optimization control problem. The problem is to determine both, control points $\{t_i\}$ and production speeds $\{s_i\}$ to minimize the system's expenses

$$J = \min_{\{t_i, s_i\}} \sum_{i=0}^{N-1}\left[c_{s_i}\left(t_{i+1} - t_i\right)\right] + N \cdot c_{ins} \tag{6.1.2}$$

such that:

$$\Pr\left\{V^f(D)\geq V\right\}\geq p, \qquad (6.1.3)$$

$$t_0 = 0, \qquad (6.1.4)$$

$$t_N = D, \qquad (6.1.5)$$

$$t_{i+1} - t_i \geq d, \quad 0 \leq i \leq N-1, \qquad (6.1.6)$$

$$D - t_i \geq \Delta, \quad 0 \leq i \leq N-1, \qquad (6.1.7)$$

$$s_i \leq k = \min_{1\leq q\leq m}\left[q : W_p(t_i, q)\leq D\right]. \qquad (6.1.8)$$

Objective (6.1.2) enables minimization of all system's expenses, while objective (6.1.3) reflects the chance constraint. Relation (6.1.4) implies that the first control point to undertake decision-making is zero, namely, the starting moment to process the production program. Relation (6.1.5) implies that the last inspection point is the due date D. Restriction (6.1.6) ensures the time span between each two consecutive control points, while (6.1.7) provides the means of ensuring the closeness of the inspection moment to the due date. Relation (6.1.8) means that the production speed to be chosen at any routine control point must not exceed *the minimal speed which guarantees meeting the deadline on time, subject to the chance constraint*. Thus, as outlined above, *unnecessary surplus speeds are not implemented*.

The problem defined in (6.1.2-6.1.8) is a very complicated stochastic optimization problem which cannot be solved in the general case; it allows only a heuristic solution. The algorithm outlined below, in *6.1.6*, determines at each control point t_i both, the next control point t_{i+1} and the speed v_{s_i} at which to proceed until that control point.

6.1.5 The chance constraint principle

The chance constraint principle is the basic approach for determining the next control point t_{i+1} on the basis of the routine control point t_i and the actual output $V^f(t_i)$ observed at that moment. Note that such an approach has been successfully implemented in [83-84] for controlling stochastic network projects.

Consider a routine control point t_i, together with the actual output observed at that point, $V^f(t_i)$. For each production speed v_j, $1\leq j\leq m$, calculate by means of simulation a representative statistical sample $\left\{T_j^{(s)}\right\}$, where $T_j^{(s)}$ is the simulated value of the completion time of the production program obtained by using speed v_j throughout. It can be well-recognized that the value of $T_j^{(s)}$ can be determined from

$$T_j^{(s)} = \frac{V - V^f(t_i)}{v_j^{(s)}} + t_i, \qquad (6.1.9)$$

where $v_j^{(s)}$ stands for the simulated production speed v_j at control point t_i.

After obtaining samples $\{T_j^{(s)}\}$, $1 \le j \le m$, calculate the corresponding p-quantiles and single out the subset of speeds for which:

$$W_p(t,j) < D \tag{6.1.10}$$

holds. Note that if, for a certain speed j, (6.1.10) holds, then all speeds with higher indices also satisfy (6.1.10). Consider one of the speeds entering the subset, e.g., speed v_q. It can be well-recognized (see Fig. 6.1) that, being introduced from point $A(t_i, V^f(t_i))$ throughout, speed v_q enables the deadline to be met on time, subject to the chance constraint. *Moreover, even if no processing at all takes place within the period of length $\Delta t = D - W_p(t_i, q)$ (see the straight line AF) and afterwards speed v_q is introduced at point F, this speed v_q still enables the deadline to be met on time, under the chance constraint (7.1.3)*. This can be well-recognized by examining two parallel straight lines: line AE, which enables accomplishing the production program with a probability exceeding p (henceforth, call this line $AE^{(q)}$) and line BF which enables the deadline to be met on time with confidence probability equal to p (call this line $BF^{(q)}$). Note that, if the production process proceeds with speed v_q from *any point* on line $BF^{(q)}$, the target will be met on time subject to the chance constraint. This basic principle has been implemented in the heuristic algorithm.

6.1.6 <u>The heuristic algorithm</u>

Referring to [54,73], the heuristic control algorithm at each routine control point t_i, enables minimization of the system's expenses (6.1.2) during the *remaining time* $(D - t_i)$. Thus, the objective function for optimizing decision-making at point t_i includes only *future expenses*, while past expenses, as well as past decision-makings, are considered to be irrelevant for the on-line control procedure. At each control point t_i, decision-making centers around the assumption (see [54,73]) that there is no more than one additional control point before the due date.

It can be well-recognized that the backbone of the heuristic control algorithm is *Subalgorithm I* which, at each routine control point t_i determines both index s_i of the speed to be introduced and the next control point t_{i+1}. Following the assumption outlined above, two speeds have to be chosen at point t_i:

1. Speed v_{j_1}, $j_1 = s_i$, which *has to be actually introduced at point t_i up to the next control point t_{i+1}*;
2. Speed v_{j_2}, $j_2 = s_{i+1}$, which *is forecast to be introduced at control point t_{i+1} within the remaining period $[t_{i+1}, D]$*.

Note that, if speed v_{j_2} is forecast to be the *last processing speed* before the due date D, control point t_{i+1} has to be necessarily on straight line $BF^{(j_2)}$ (see Fig, 6.1), otherwise chance constraint (6.1.3) might not be met. We suggest singling out, at each routine control point t_i, all possible couples (j_1, j_2) satisfying restriction (6.1.8), with subsequent choosing the one delivering the minimum of forecasted production and control expenses, namely

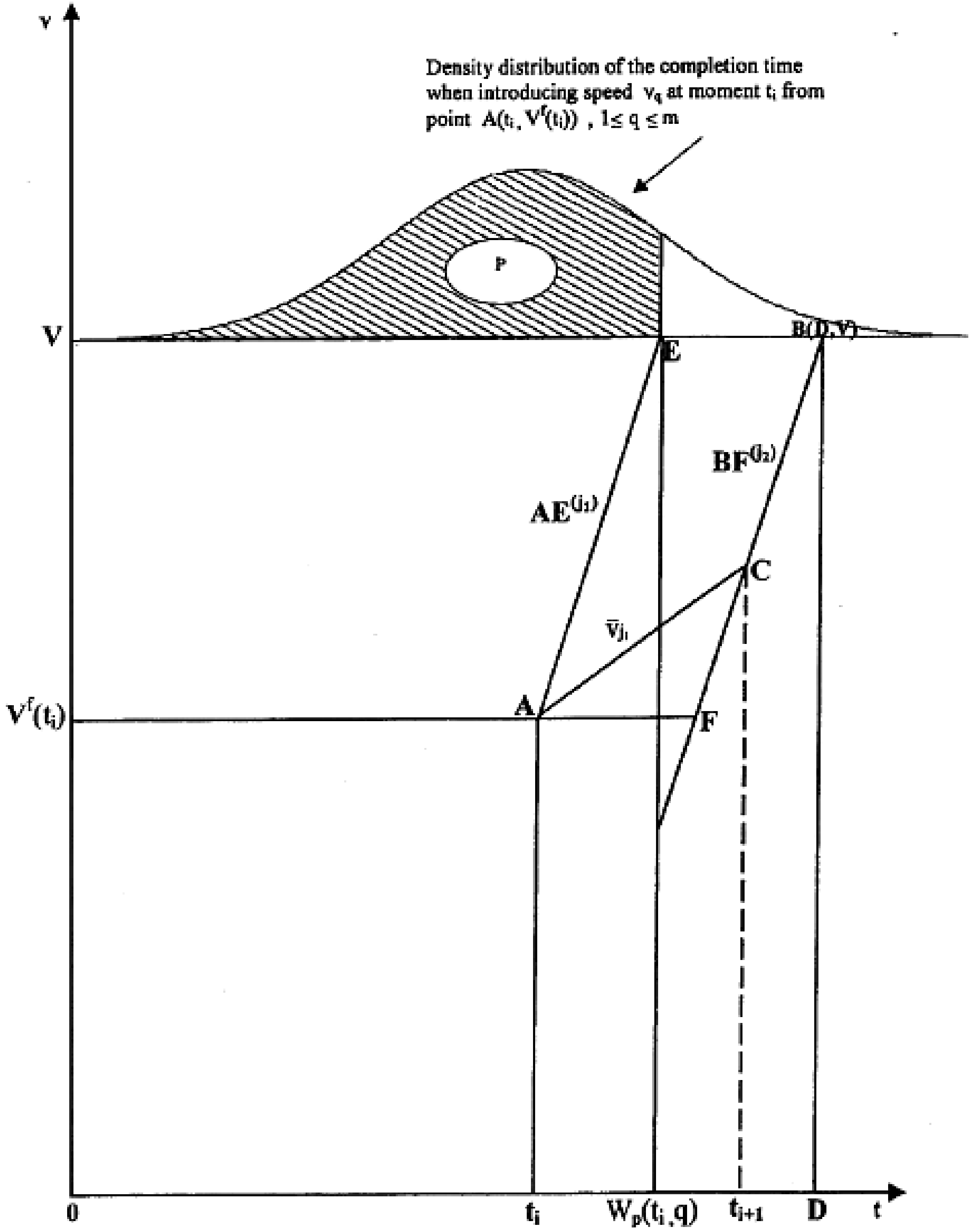

Figure 6.1. The general idea of the chance constraint principle

$$\min_{\{j_1, j_2\}} \left\{ c_{j_1}(t_{i+1} - t_i) + c_{j_2}(D - t_{i+1}) + c_{ins} \right\} \tag{6.1.11}$$

such that:

$$j_1 \le k = \min_{1\le q\le m}\left[q : W_p(t_i, q) \le D \right], \qquad (6.1.12)$$

$$j_2 \ge k \ \ if \ \ j_1 < k, \qquad (6.1.13)$$

$$j_2 \le k \ \ if \ \ j_1 = k. \qquad (6.1.14)$$

Restriction (6.1.12) is embedded in the algorithm to satisfy restriction (6.1.8). Restriction (6.1.13) holds, since case $j_1 < k$, $j_2 < k$ contradicts chance constraint (6.1.3). Case $j_1 = k$, $j_2 > k$ is a pointless one since, for both couples (k,k) and $(k, j_2 > k)$, chance constraint (6.1.3) will be met, but the second possibility proves to be more costly.

As to value t_{i+1}, we suggest calculating the latter on the assumption that, being introduced at t_i, the actual processing speed is $\bar{v}_{j_1}$. Thus, t_{i+1} may be determined as the abscissa of the intersection point C (see Fig. 6.1) of two straight lines:

$$AC: \ v = V^f(t_i) + \bar{v}_{j_1}(t - t_i); \qquad (6.1.15)$$

$$BV^{(j_2)}: \ v = \frac{V - V^f(t_i)}{W_p(t_i, j_2) - t_i} t + V - D \frac{V - V^f(t_i)}{W_p(t_i, j_2) - t_i}. \qquad (6.1.16)$$

Note that case $j_1 = j_2 = k$ is possible if using speed v_{j_1} throughout, until the due date D, results in the cheapest realization. In such a case, value c_{ins} has to be excluded from (6.1.11).

Further on, we will show the possibility of implementing the chance constraint principle in on-line control models for stochastic network projects. Such an implementation has to be made with certain modifications as compared to the control model outlined above, in §6.1.

§6.2 Case of a single project

6.2.1 The system's description

An activity-on-arc network project $G(N, A)$ of PERT type, with random activity durations, is considered. The accomplishment of each activity is measured in percentages of the total project. Since evaluating the project's accomplishment continuously is difficult and costly, periodic inspections are preferred. Non-consumable, i.e., renewable, resources, such as machines and manpower, are utilized to carry out the project.

It is assumed [53,83-84,151] that any activity comprised in the project has to be operated by a standardized set of various resource capacities which we will henceforth refer to as "a Generalized Resource Unit" (GRU). All the non-consumable resources assigned to the project, are subdivided into several *identical* GRU, with the following properties:

- each activity (i, j) entering the project $G(N, A)$ has to be operated, from beginning to the end, by only one GRU;
- different activities cannot be operated simultaneously by one and the same

GRU;

- in the course of processing an activity a GRU may use several possible speeds that are subject to random disturbances. The number of possible speeds is common to all activities.

For each activity, speeds are sorted in ascending order of their average capacities - namely, speeds are indexed. All GRU are indexed in arbitrary order too. Since all GRU are identical the operational speed for each activity does not depend on the index of the GRU which is involved in processing that activity. It is assumed that at any moment $t > 0$ activities that are operated at that moment, have to apply speeds of one and the same index. Assume, further, that any speed can be changed only at a control (inspection) point, that is, all the project's activities being carried out between two adjacent control points, have to be operated with speeds of one and the same index. There exists only one exception: if an activity must continue in operation through a control point, the activity speed cannot be changed.

The project's due date and the minimal permissible probability (chance constraint) of accomplishing the project at the due date are pregiven.

It can be well-recognized that the outlined above stochastic network project covers a broad spectrum of possibilities, including innovative R&D projects and especially construction projects and similar projects with variable speeds. For those projects a GRU is nothing else but a standard building team comprising both machines and personnel while possible speeds correspond to different hours a day per worker. Thus, those speeds depend only on the degree of intensity of the project's realization.

An on-line cost-optimization model is outlined, that at each control point faces a stochastic optimization problem. Given the average processing costs per time unit for each activity to be operated under each speed, together with the average cost of performing a single inspection at the routine control point t_g, the problem is to determine the next control point t_{g+1} and the new index of the speeds (for all activities to be operated) to be employed up to that point.

The problem's solution is based on the combination of the chance constraint principle which has been outlined in [73] for production systems and in [54] for other organization systems, and a resource constrained simulation model for non-consumable limited resources [70]. A heuristic algorithm is outlined; its efficiency is evaluated by means of simulation.

Note that certain refinements which made the chance constraint model more applicable for stochastic network projects (in comparison with the model outlined in §6.1 and aimed at production systems) have been implemented [83-84] in the algorithm (see §6.2).

6.2.2 Notation

Let us introduce the following terms:

$G(N, A)$ - activity-on-arc network project of PERT type;

$(i, j) \in G(N, A)$- activity leaving node i and entering node j;

G_t - the project observed at moment $t \geq 0$, $G_0 = G(N, A)$;

$\upsilon_{ij}^{(k)}$ - the k-th speed to process activity (i, j), $1 \leq k \leq m$;

m - number of possible speeds common to all activities (pregiven);

n - number of identical GRUs (pregiven);

n_t - number of free available GRUs at moment $t \geq 0$, $n_0 = n$;

$t_{ij}^{(k)}$ - random duration of activity (i, j) using speed $\upsilon_{ij}^{(k)}$ throughout;

$a_{ij}^{(k)}$ - lower bound of random activity duration $t_{ij}^{(k)}$ (pregiven);

$b_{ij}^{(k)}$ - upper bound of random activity duration $t_{ij}^{(k)}$ (pregiven);

ρ_{ij} - percentage of activity (i, j) in the entire project (pregiven);

D - the due date of the project (pregiven);

p - least permissible probability (chance constraint) of meeting the project's due date on time (pregiven);

N - number of control points (a random value);

t_g - the g-th control point, $g = 0,1,..., N$; $t_0 = 0$, $t_N = D$;

s_g - the index of the project's speed (identical for all activities) to be implemented within interval $\left[t_g, t_{g+1}\right]$, $1 \leq s_g \leq m$;

Δ_1 - the minimal value of the closeness of the inspection moment to the due date (pregiven);

Δ_2 - the minimal time span between two consecutive control points t_g and t_{g+1} in order to force convergence (pregiven);

$V^f(t)$ - actual project output (in percentages of the entire project) observed at moment $t \geq 0$, $V^f(0) = 0$;

$C^f(t)$ - the actual accumulated processing and control costs calculated at moment $t \geq 0$, $C^f(0) = 0$;

$c_{ij}^{(k)}$ - the average processing cost per time unit for activity (i, j) to be operated with speed $\upsilon_{ij}^{(k)}$ (pregiven);

c_{ins} - the average cost of undertaking the project's inspection (pregiven);

V - the planned entire project volume (pregiven).

$W_p\left[t, k, V^f(t)\right]$ - the p-quantile of the moment the project will be accomplished on condition that the k-th speed for all activities will be implemented at control point t and will be used throughout, and the actual observed output at that moment is $V^f(t)$;

S_{ij} - the actual moment activity (i, j) starts (a random value);

$F_{ij} = S_{ij} + t_{ij}^{(k)}$ - the actual moment activity (i, j) is completed (a random value);

$C(t_1, t_2)$ - processing and control costs calculated within the time interval (t_1, t_2), $0 \leq t_1 < t_2 \leq D$ (a random value).

6.2.3 *The problem*

Let us consider the cost-optimization on-line control problem. The problem is to determine both control points $\{t_g\}$ and activity speeds $\{\upsilon_{ij}^{(k)}\}$ to minimize the average project's expenses

$$J = \min_{\{t_g, \upsilon_{ij}^{(k)}, s_g\}} E\left\{ \sum_{(i,j)\in G} \left(c_{ij}^{(k)} \cdot t_{ij}^{(k)}\right) + N \cdot c_{ins} \right\} \qquad (6.2.1)$$

subject to

$$k = s_g \quad \forall (i,j): t_g \le S_{ij} < t_{g+1}, \quad 0 \le g \le N, \qquad (6.2.2)$$

$$\Pr\left\{ \max_{(i,j)\in G_{t_g}} F_{ij} \le D \right\} \ge p, \qquad (6.2.3)$$

$$t_0 = 0, \qquad (6.2.4)$$

$$t_N = D, \qquad (6.2.5)$$

$$D - t_g \ge \Delta_1, \quad 0 \le g \le N, \qquad (6.2.6)$$

$$t_{g+1} - t_g \ge \Delta_2, \quad 0 \le g \le N, \qquad (6.2.7)$$

$$s_g \le s_g^* = \min_{1\le q\le m}\left\{ q : W_P\left[t, q, V^f(t)\right] \le D \right\}. \qquad (6.2.8)$$

Objective (6.2.1) enables minimization of all operating costs and control expenses. However, referring to [73], the heuristic control algorithm facilitates decision-making at each control point t_g on the basis of *future* expenses only, i.e., during the remaining time $D - t_g$. Past expenses, as well as past decision-makings, are not relevant for the on-line control model. Relation (6.2.2) honors the chance constraint while relations (6.2.3-6.2.7) are obvious.

Let us analyze (6.2.8) in greater detail. Relation (6.2.8) means that the speed to be chosen at any routine control point t_g should not exceed the minimal speed s_g^* that enables meeting the deadline on time, subject to the chance constraint. It can be well-recognized that operating an activity at a higher speed always results in higher costs to accomplish the activity than by using a lower speed. Thus, (6.2.8) prohibits using unnecessary high speeds. Note that after introducing speed s_g at control point t_g all the activities (i,j) starting from t_g, have to be operated at that speed *throughout*, i.e., until the due date D, in order to determine values $W_P\left[t, k, V^f(t)\right]$ and, later on, to choose the minimal index s_g^*. However, since the number of GRUs is limited, at certain moments $t \ge 0$ *the number of activities ready to be processed may exceed the amount* n_t *of free available GRUs*. Thus, if in the course of the project's realization there is a lack of resources, a competition among those activities has to be arranged to single out a subset of activities that will start to be operated at moment t and can be provided with resources. The corresponding decision-making auxiliary algorithm [70-71] which determines values S_{ij} in the course of carrying out the project, is outlined in Chapter 11. The competition among the activities seeking for resources is facilitated by solving a zero-one integer programming problem to maximize the total contribution of the accepted activities to the expected project duration. For each activity (i,j) its contribution equals the product of the average duration of the activity $\bar{t}_{ij}^{(k)}$ and its probability p_{ij} of being on the critical path in the course of the project's realization. Those probability values may be determined by means of simulation. The algorithm outlined in Chapter 11 has to be implemented in order to undertake numerous simulation runs to calculate values $W_p[t,k,V^f(t)]$.

6.2.4 The general idea of the problem's solution

Several main concepts are imbedded in the model:

1. At each control point t_g, decision-making centers around the assumption [73] that there is no more than one additional control point before the due date. Thus, two speeds have to be chosen at each routine control point t_g:
   a) speed $\upsilon^{(k_1)}$ which has to be *actually* introduced at point t_g up to the next control point t_{g+1};
   b) speed $\upsilon^{(k_2)}$ which is *forecast* to be introduced at control point t_{g+1} within the period $\left[t_{g+1}, D\right]$.

   Couple $\left(\upsilon^{(k_1)},\upsilon^{(k_2)}\right)$ provides the minimal processing and control costs among all possible couples subject to (6.2.8). After meeting control point t_{g+1} the on-line problem has to be resolved anew.
2. At any control point t_g the past operational and control expenses are irrelevant to the on-line control problem and are not taken into account whatsoever.
3. If speed $\upsilon^{(k_1)}$ is actually introduced at control point t_g subject to (6.3.8), the project possesses time reserves $D-W_p\left[t_g,k_1,V^f(t_g)\right]$ (see interval AF on Fig. 6.2 where AE and BF are parallel straight lines). Since speed $\upsilon^{(k_2)}$ is forecast to be the last processing speed before the project's due date D, control point t_{g+1} has to be on the straight line BG which is parallel to the straight line connecting points $\left[t_g, V^f(t_g)\right]$ and $\left[W_p\left[t_g,k_2,V^f(t_g)\right],V\right]$. Such a concept which has been outlined in §6.1 is, in fact, implementation of the chance constraint principle [73].
4. As to value t_{g+1}, we suggest calculating the latter on the assumption that the length of interval $\left[t_g, t_{g+1}\right]$ is essentially smaller than the remaining part of the planning horizon $\left[t_{g+1}, D\right]$. In order to determine t_{g+1} via a

short-term forecasting we suggest replacing for all activities (i,j) satisfying $t_g \le S_{ij} < t_{g+1}$, their random durations $t_{ij}^{(k)}$ by the corresponding average values $\bar{t}_{ij}^{(k)}$. Such an assumption enables determining moment t_{g+1} as a deterministic value since the straight line $BG^{(k_2)}$ has a precise model while simulating the project's realization between points A and G (see Fig. 6.2) can be carried out in deterministic terms. *Thus, determining speeds* $\upsilon^{(k_1)}$ *and* $\upsilon^{(k_2)}$ *is carried out via a long-term forecasting on the basis of* p*-quantile estimations, while calculating the next control point* t_{g+1} *is facilitated by using a short-term forecast based on substituting random values for their average ones*. This is actually the main principal contribution of the results outlined in §6.2 when compared to the general chance constraint approach presented in §6.1.

5. Simulating the project's realization in order:
   - to obtain p-quantile parameters $W_p\left[t_g, k_1, V^f(t_g)\right]$ to forecast the optimal speed couple $\left(\upsilon^{(k_1)}, \upsilon^{(k_2)}\right)$, as well as
   - to undertake project's simulation within the interval $\left[t_g, t_{g+1}\right]$,

   is carried out by a combination of a simulation model and a heuristic resource constrained decision-making algorithm [68,70]. Decision-making, i.e., determining values S_{ij} by feeding-in free available GRU, is carried out at decision points t, when at least one activity is ready to be operated. If the number of such activities at a certain moment t exceeds the amount n_t of free available GRU at $t \ge 0$, a zero-one integer programming problem to single out the *optimal* subset of activities to be supplied with resources is solved. For those chosen activities (i,j) their starting moments S_{ij} are equal t. Note that the integer programming problem outlined in [68,70] provides the *exact* solution.

6.2.5 <u>The heuristic on-line control algorithm to determine the next control point and the project's speed</u>

The algorithm outlined below determines at each control point t_g for project G_{t_g} the next control point t_{g+1} and the index of the speed for all activities (i,j) starting from moment t_g. Given at $t = t_g$:

- project G_{t_g};
- the project's due date D;
- the project's chance constraint p;
- the project's planned volume $V = 100\%$;
- lower and upper bounds $a_{ij}^{(k)}$ and $b_{ij}^{(k)}$, $(i,j) \in G_{t_g}$, $1 \le k \le m$, for all random ac-

tivity durations $t_{ij}^{(k)}$;

- number m of possible speeds;

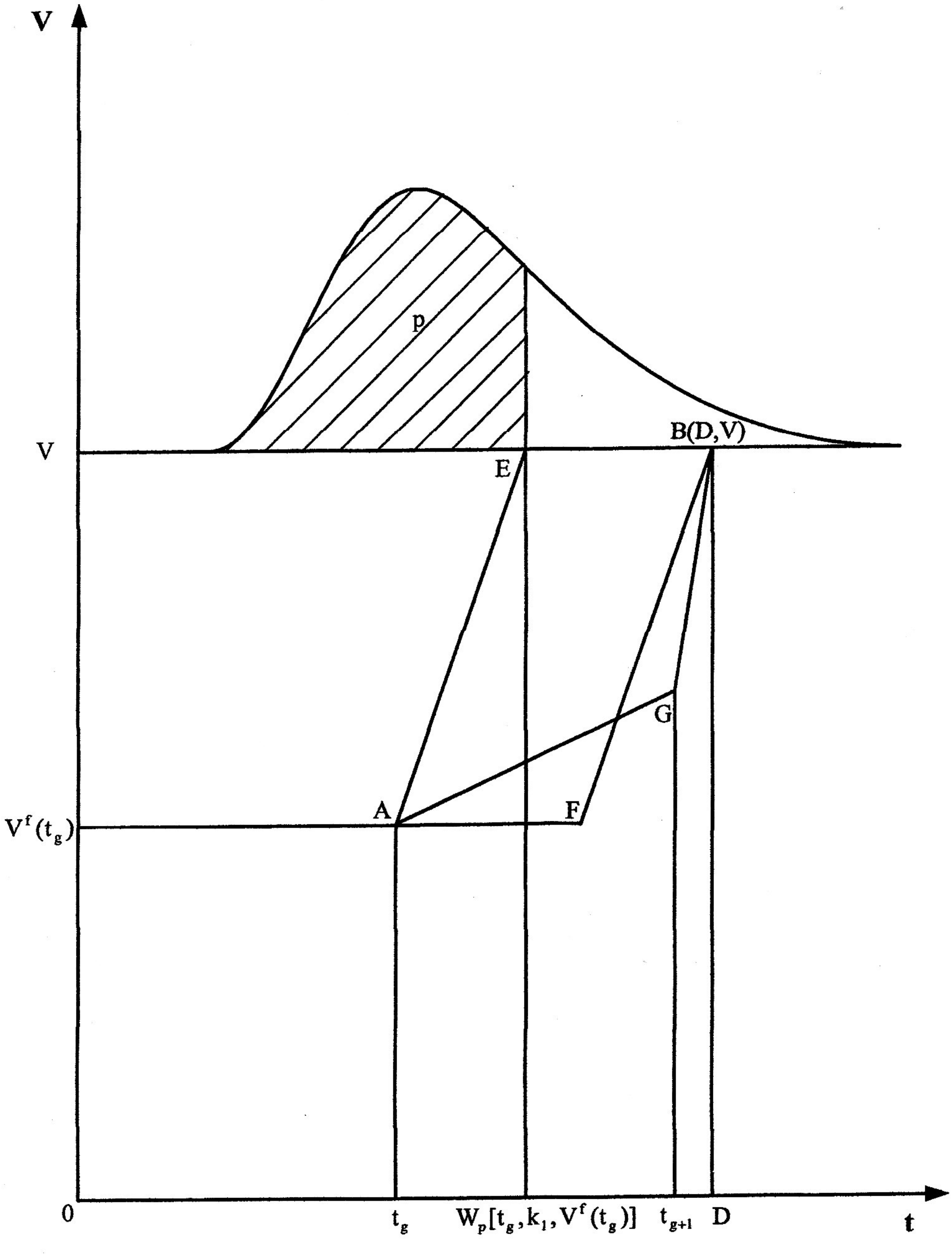

Figure 6.2. Determining the next control point

- number n of identical GRUs;
- number n_{t_g} of free available GRUs at moment $t = t_g$;
- percentages ρ_{ij} of activities $(i, j) \in G_{t_g}$ in the entire project;
- the project's accumulated output $V^f(t_g)$ observed at point $t = t_g$;

- the project's accumulated processing and control costs $C^f(t_g)$ calculated at point $t = t_g$;
- average processing costs $c_{ij}^{(k)}$, $(i,j) \in G_{t_g}$, $1 \le k \le m$, per time unit;
- average cost of carrying out a single inspection c_{ins}.

The step-by-step procedure of the algorithm is as follows:

Step 1. For each speed $\upsilon^{(k)}$, $1 \le k \le m$, determine by means of simulation values $W_p\left[t_g, k, V^f(t_g)\right]$ (a forecasting procedure). Step 1 comprises, in essence, four subalgorithms as follows:

Subalgorithm I actually governs most of the procedures to be undertaken in the course of the project's realization, namely:

1.1 Determines all activities $(i,j) \in G_{t_g}$ being operated at moment $t = t_g$.

1.2 Simulates their finishing times F_{ij}.

1.3 Determines sequentially decision points t when at least one activity is ready to be operated; let a_t be the amount of those activities.

1.4 In case $n_t \ge a_t$ supplies activities with resources and updates the number of free remaining GRUs. Otherwise go to Substep 1.6.

1.5 Introduces speed $\upsilon^{(k)}$ and simulates the corresponding activities' durations.

Subalgorithm II takes over when $n_t < a_t$ and comprises the following substeps:

1.6 Determines all the activities that have not yet started to be operated. Simulate their random durations with speed $\upsilon_{ij}^{(k)}$.

1.7 Calculates the critical path of the remaining graph G_t.

1.8 Repeats Substeps 1.6-1.7 M times to obtain representative statistics.

1.9 Calculates the frequency $p(i,j)_\lambda$ for each activity $(i,j)_\lambda$, $1 \le \lambda \le a_t$, seeking for resources and ready to be operated, to be on the critical path.

Subalgorithm III undertakes the competition in case $n_t < a_t$ to single out the optimal subset from a_t activities seeking for resources by solving the appropriate optimization problem. The zero-one integer programming problem is as follows: determine integer values ξ_λ, $1 \le \lambda \le a_t$, to maximize the objective

$$\max_{\{\xi_\lambda\}} \left\{ \sum_{\lambda=1}^{a_t} \left[\bar{\mu}(i,j)_\lambda \cdot p(i,j)_\lambda \cdot \xi_\lambda \right] \right\} \qquad (6.2.9)$$

subject to

$$\sum_{\lambda=1}^{a_t} \xi_\lambda \le n_t, \tag{6.2.10}$$

where $\xi_\lambda = \begin{cases} 1 \text{ if activity } (i,j)_\lambda \text{ is provided with resources;} \\ 0 \text{ otherwise.} \end{cases}$

After solving problem (6.2.9-6.2.10), i.e., carrying out the competition, Substep 1.5 of Subalgorithm I is applied. Thus, the simulation proceeds until the project's completion at moment $F = \max_{\{i,j\}} F_{ij}$. After determining value F in the course of a simulation run, **Subalgorithm IV** calculates values $W_p\left[t_g, k, V^f(t_g)\right]$ *for each* k, $1 \le k \le m$, *separately*. Determining W_p is carried out by implementing numerous simulation runs to obtain representative statistics.

Note that *in the course of one simulation run* Subalgorithms I-III are applied with the project converging to the target at constant speed $\upsilon^{(k)}$ which has been introduced at control point $t = t_g$ and is used *throughout*, i.e., until D, without any additional control points. Thus, Step 1 results in determining *predictive* values. Simulation of activity durations at that step is carried out *not to simulate actual activity realizations*, but to facilitate forecasting in order to calculate the p-quantiles for each speed $\upsilon^{(k)}$.

Step 2. Determine $s_g^* = \min_{1\le k\le m}\left\{k : W_p\left[t_g, k, V^f(t_g)\right] \le D\right\}$. If s_g^* cannot be determined problem (6.2.1-6.2.8) has no solution. Otherwise apply the next step.

Step 3. Consider the list of possible couples $\left(\upsilon^{(k_1)}, \upsilon^{(k_2)}\right)$ in accordance with restrictions

$$k_1 \le s_g^*, \tag{6.2.11}$$

$$k_2 \ge s_g^* \text{ if } k_1 < s_g^*, \tag{6.2.12}$$

$$k_2 \le s_g^* \text{ if } k_1 = s_g^*. \tag{6.2.13}$$

Restriction (6.2.11) is imbedded in the algorithm to satisfy (6.2.8). Restriction (6.2.12) is true since $k_1 < s_g^*$ and $k_2 < s_g^*$ contradict chance constraint (6.2.3). Case $k_1 = s_g^*$, $k_2 > s_g^*$ is a pointless one since for both cases (k_1, k_1) and $(k_1, k_2 > k_1)$ chance constraint (6.2.3) will be met, but the second case proves to be a costlier one.

Check each possible couple by applying the below Steps 4-13.

Step 4. Determine equation of straight line BG (see Fig. 6.2) as follows

$$V(t) = \frac{V - V^f(t_g)}{W_p\left[t_g, k_2, V^f(t_g)\right] - t_g} \cdot t + V - D \cdot \frac{V - V^f(t_g)}{W_p\left[t_g, k_2, V^f(t_g)\right] - t_g}. \tag{6.2.14}$$

Step 5. This step determines the next control point t_{g+1} by implementing a *predictive* model. All activities (i,j) starting after t_g, obtain a *deterministic* duration $t_{ij}^{(k_1)} = \bar{t}_{ij}^{(k_1)}$, $S_{ij} \ge t_g$. **Subalgorithm V** which duration $t_{ij}^{(k_1)} = \bar{t}_{ij}^{(k_1)}$, $S_{ij} \geq t_g$. **Subalgorithm V** which determines routine decision nodes t when at last one activity $(i,j)_\lambda$, $1 \leq \lambda \leq a_t$, has to be provided with resources, is imbedded in the model. All those activities $(i,j)_\lambda$ have to be determined. Note that calculating moments t is in this case a *deterministic technique*.

Step 6. Quasi-optimal resource reallocation among activities $(i,j)_\lambda$, $1 \leq \lambda \leq a_t$, which at a routine decision node t are ready to be operated and seeking for resources, is carried out. A simplified reallocation model is suggested:

6.1 After determining activities $(i,j)_\lambda$, $1 \leq \lambda \leq a_t$, all the competitive activities are sorted in descending order of their average durations $t(i,j)_\lambda = \bar{t}_{ij}^{(k_1)}$.

6.2 All the sorted activities are examined one after another, in the descending order of their average values, to check the possibility that the activity can be provided with remaining available resources. If for a certain activity $(i,j)_\lambda$ relation $n_t \geq 1$ holds, the GRU is passed to the activity.

6.3 The free, available GRU are updated, $n_t - 1 \Rightarrow n_t$.

6.4 The next activity $(i,j)_{\lambda+1}$ is examined. Subalgorithm V terminates either when all the free available GRU are reallocated, i.e., n_t becomes zero, or all the a_t activities have been examined.

Step 7. Set to each one of the chosen at Step 6 activities $(i,j)_\lambda$ the starting time $S_{(i,j)_\lambda} = t$ and introduce its corresponding time duration $t_{(i,j)_\lambda}^{(k_1)} = \bar{t}_{(i,j)_\lambda}^{(k_1)}$. All other activities $(i,j)_\lambda$ which have not been supplied with resources proceed waiting in the line until the next routine decision node $t^* > t$.

Step 8. At node t^* the activities which proceed seeking for resources in the line obtain additional average time duration amounted as $t^* - t$. Afterwards Step 6 is re-applied. Note that additional average time duration is used at Substep 6.2 for *competition purposes* only. At Step 7 regular average duration values $t_{(i,j)_\lambda}^{(k_1)} = \bar{t}_{(i,j)_\lambda}^{(k_1)}$ are introduced.

Step 9. Steps 5-8 are undertaken until a decision node t coincides with line (6.2.14) with a pregiven accuracy $\varepsilon > 0$. Value t is taken as the next control point t_{g+1}. As outlined above, determining value t_{g+1} is carried out by using deterministic techniques and, thus, no simulation runs are necessary to calculate t_{g+1}. Note that Steps 5-9 are predictive ones and are implemented to determine the next control point t_{g+1}. These steps are not intended to simulate actual activity realizations.

Step 10. For each combination of couples (k_1, k_2) check if $t_{g+1} - t_g \geq \Delta_2$ holds. If not, calculate $t_{g+1} = t_g + \Delta_2$.

Step 11. For each combination of couples (k_1, k_2) check if $t_{g+1} > D$ or $D - t_{g+1} < \Delta_1$. If one of those relations hold, set $t_{g+1} = D$.

Step 12. This step is somewhat similar to Step 1. The purpose of the step is to carry out for each couple $\left(\upsilon^{(k_1)}, \upsilon^{(k_2)}\right)$ under consideration numerous simulation runs within the remaining planning horizon $\left[t_g, D\right]$ with an additional control point t_{g+1} in order to:

- simulate the outcome product $V^f(D)$ and calculate statistical frequencies $\frac{M_D}{M}$, where M is the entire number of simulation runs in order to obtain representative statistics, while M_D stands for the number of simulation runs satisfying $V^f(D) \geq V$;
- simulate the average total operating and control costs within the interval $\left[t_g, D\right]$.

Thus, Step 12 carries out predictive evaluations in order to choose the optimal couple (k_1, k_2). The main difference between Steps 1 and 12 centers on the following:

a. Subalgorithms I, II and III of Step 1 are implemented for a constant speed $\upsilon^{(k_1)}$ within the entire period $\left[t_g, D\right]$ while at Step 12 for all activities (i,j) starting after t_{g+1}, i.e., satisfying $S_{ij} \geq t_{g+1}$, speed $\upsilon^{(k_2)}$ is introduced.

b. Subalgorithm IV of Step 1 calculates values $W_p\left[t_g, k_1, V^f\left(t_g\right)\right]$ for a constant speed used throughout the period $\left[t_g, D\right]$ while Subalgorithm IV of Step 12 calculates, in essence, p-quantiles $W_p\left[t_g, k_1, t_{g+1}, k_2, V^f\left(t_g\right)\right]$.

c. In the course of carrying out M simulation runs Step 12 calculates the average forecasted processing and control costs within the period $\left[t_g, D\right]$, i.e., determines the average value $\overline{C}\left[t_g, D\right]$ where

$$\overline{C}\left[t_g, D\right] = \sum_{(i,j)^*}\left[c_{ij}^{(k_1)} \cdot t_{ij}^{(k_1)}\right] + \sum_{(i,j)^{**}}\left[c_{ij}^{(k_2)} \cdot t_{ij}^{(k_2)}\right] + 2 \cdot c_{ins}, \text{ if } k_1 \neq k_2, \text{ and} \quad (6.2.15)$$

$$\overline{C}\left[t_g, D\right] = \sum_{(i,j)^{**}}\left[c_{ij}^{(k_1)} \cdot t_{ij}^{(k_1)}\right] + c_{ins}, \text{ if } k_1 = k_2. \quad (6.2.16)$$

Here $(i,j)^*$ denotes activities satisfying $t_g \leq S_{ij} < t_{g+1}$, while $(i,j)^{**}$ stands for activities with $S_{ij} \geq t_{g+1}$. Note that in case $k_1 \neq k_2$ there are two remaining inspections (at points t_{g+1} and D), while for $k_1 = k_2$ there is only one inspection left (at point D).

Step 13 singles out from all couples with indices (k_1, k_2) under consideration *feasible couples* (k_1, k_2) satisfying $\frac{M_D}{M} \geq p$ (see Step 12), to honor restriction (6.2.3).

Step 14 determines from all feasible couples which have been singled out at Step 13, the optimal couple (k_1^*, k_2^*) which results in the minimal value $\overline{C}[t_g, D]$.

Step 15. Introduce speed $\upsilon^{(k_1^*)}$ at point t_g and start carrying out the project with that speed. Thus, Step 15 is the *first step* comprised into the algorithm which *actually operates activities* (i, j) *starting from* t_g*, with new speeds* $\upsilon_{ij}^{(k_1^*)}$. Activity durations $t_{ij}^{(k_1^*)}$ are performed either in real time, or they may be based on simulation modeling. Similar to Steps 1 and 12, Step 15 implements Subalgorithms I, II and III in order to reallocate free available resources among activities which at decision nodes are ready to start being processed.

Step 16. When the progress of the project meets control point t_{g+1}, the project has to be inspected at that point, including the output product $V^f(t_{g+1})$ and the costs $C(t_g, t_{g+1})$. Thus, the accumulated value $C^f(t_{g+1})$ is determined by

$$C^f(t_{g+1}) = C^f(t_g) + C(t_g, t_{g+1}). \qquad (6.2.17)$$

Step 17. If $t_{g+1} = D$, the project is inspected at the due date and the control algorithm terminates. In case $t_{g+1} < D$, the input information is updated and Step 1 is re-applied. Thus, at the next control point t_{g+1} the heuristic on-line control algorithm has to be implemented anew.

§6.3 Case of several projects

6.3.1 Introduction

The system under consideration [84] comprises several simultaneously realized activity-on-arc network projects of PERT type with random activity durations. The accomplishment of each project's activity is measured in percentage of the whole project. All the activities are to be operated by one of the identical GRU which may use several possible speeds subject to random disturbances.

Similar to §6.2, it is assumed that the progress of any project can be evaluated only via periodical inspection in control points. At any moment $t > 0$ activities that start to operate at that moment for one and the same project, have to use speeds with similar indices (ordinal numbers). Speeds can be changed only at a control point. Within the projects' realization a GRU can be transferred from one project to another only at an emergency moment common to all projects.

The projects' due dates and their chance constraints, i.e., their minimal permissible probabilities of accomplishing the project on time, are pregiven. All GRU have to be delivered to the company store at the projects' starting time and are released when the last project is accomplished. The cost of hiring and maintaining a GRU, together with the average processing costs per time unit for operating each activity under each speed, the average cost of performing a single inspection at a control point (common to all projects) and the average cost of reallocating GRU among non-finished projects at each emergency moment, are pregiven.

In §6.2 we have outlined a cost-simulation problem for a *single project* as follows: given the fixed number of GRU, at each routine control point t_i determine the next control point t_{i+1} and the new index of the speeds for all the activities to be operated at that point. The objective is to minimize the project's total expenses. This basic problem (we will henceforth call it Problem A1) will be used in order to develop a much more complicated realistic cost-optimization model as follows: determine the optimal number of GRU to minimize the total value of all projects' expenses subject to their chance constraints.

The problem's solution is as follows:

- at the company level a combination of a search procedure to determine the number of GRU together with a resource reallocation model among the projects is considered,
- at the project level a basic cost-optimization on-line control Model A1 is applied for each project independently.

Both resource reallocation model and Model A1 are implemented into a simulation model in order to obtain representative statistics to check the fitness of the problem's solution.

It is assumed that all non-accomplished projects have to be carried out at any moment $t > 0$ with a speed exceeding zero. Thus, at least one GRU unit has to be assigned to each project. At any moment each GRU can operate only one activity.

6.3.2 Notation

Let us introduce the following terms:

$G_e(N,A)$ - the e-th network project of PERT type, $1 \le e \le f$;

f - number of network projects;

f_t - number of network projects which at moment t are not completed, $t \ge 0$;

$(i,j)_e \in G_e(N,A)$ - activity (i,j) entering the e-th project;

G_{et} - project $G_e(N,A)$ observed at moment $t \ge 0$; $G_{e0} = G_e(N,A)$;

$v_{ije}^{(k)}$ - the k-th speed to process activity (i,j), $1 \le k \le m$;

m - number of possible speeds common to all activities (pregiven);

n_{et} - number of identical generalized resource units GRU assigned to project $G_e(N,A)$ at emergency moment $t \ge 0$; $n_e = n_{e0}$;

n - total number of GRU to be hired and maintained throughout the planning horizon by the company (optimized variable, to be determined beforehand);

ρ_{ije} - percentage of activity $(i,j)_e$ in project $G_e(N,A)$ (pregiven), $1 \le e \le f$;

D_e - due date of project $G_e(N,A)$ (pregiven);

p_e - chance constraint to meet the deadline D_e on time (pregiven);

$V_e^f(t)$ - actual project's G_{et} output in percentages of the total project (observed at moment t, $t \ge 0$);

$C_e^f(t)$ - the actual accumulated processing and control costs of project G_{et} calculated at moment t, $t \ge 0$;

$W_p\left[t, k, V_e^f(t)\right]$ - the p-quantile of the moment project $G_e(N,A)$ will be accomplished on condition that the k-th speed for all activities will be introduced at control point t and will be used throughout, and the actual observed output at that moment is $V_e^f(t)$;

t_{ge} - the g-th control point of the e-th project, $g = 0,1,\dots,N_e$, $t_{0e} = 0$, $t_{N_e e} = D_e$;

t_r^* - the system's emergency moment, $t_0^* = 0$, $r = 0,1,\dots,N^*$;

N_e - number of control points of the e-th project (a random value);

N^* - number of emergency moments (a random value);

Δ_{1e} - the minimal value of the closeness of the inspection moment to the due date D_e (pregiven);

Δ_{2e} - the minimal time span between two adjacent control points of the e-th project (pregiven);

$t_{ije}^{(k)}$ - random duration of activity $(i,j)_e$ using speed $v_{ije}^{(k)}$ throughout;

$c_{ije}^{(k)}$ - the average processing cost per time unit for activity $(i,j)_e$ to be operated with speed $v_{ije}^{(k)}$ (pregiven);

c_{ins} - the average cost of undertaking a routine project's inspection (common to all projects, pregiven);

c^* - the average cost of the GRU reallocation among the projects at a routine moment t_r^*;

V_e - the planned volume of project $G_e(N,A)$ (pregiven);

V_{et} - the actual non-accomplished volume of project $G_e(N,A)$ at moment t (a random value);

S_{ije} - the actual moment activity $(i,j)_e$ starts (a random value);

F_{ije} - the actual moment activity $(i,j)_e$ is completed (a random value); $F_{ije} = S_{ije} + t_{ije}^{(k)}$;

c_{GRU} - the average cost of hiring and maintaining a GRU unit per time unit (pregiven);

F_e -the actual moment project $G_e(N,A)$ is completed (a random value);

$$F_e = \underset{\{(i,j)_e \in G_{et}\}}{Max} F_{ije};$$

s_{ge} -the index of the speed to be introduced for all activities $(i,j)_e$ starting in the interval $\left[t_{ge}, t_{g+1,e}\right]$, $1 \le s_{ge} \le m$.

It can be well-recognized that two kinds of control points are imbedded in the model:

1. *Regular* control (inspection) points t_{ge} to introduce proper speeds in order to alter the project's speed in the desired direction.
2. Emergency control points t_r^* to reallocate all GRU at the company level among the non-accomplished network projects, beginning from $t=0$. Emergency moments t_r^* are as follows:
   - $t=0$;
   - t is the moment of one of the project's completion;
   - t is the control moment for one of the projects when it is anticipated that with the previously assigned for that project GRU units the project cannot meet its deadline on time.

6.3.3 The problem's formulation

The cost-optimization on-line control problem for several stochastic network projects is as follows: determine the optimal value $n^{(opt)}$ of GRU units (a deterministic value to be determined beforehand, i.e., before the projects start to be realized) together with values n_{et} assigned to all projects, all control points t_{ge}, the speeds to be introduced at that points for all projects' activities $v_{ije}^{(k_e)}$, $k_e = s_{ge}$, and the actual moments S_{ije} activities $(i,j)_e$ start (random values conditioned on decision-making of the control model), in order to minimize all operational, control, resource reallocation, hiring and maintenance expenses subject to the projects' chance constraints

$$J = \underset{\{n, n_{et}, t_{ge}, S_{ije}, s_{ge}, v_{ije}^{(k)}\}}{Min} E\left\{\sum_{e=1}^{f} \sum_{(i,j)_e \in G_{et}} \left(c_{ije}^{(k_e)} \cdot t_{ije}^{(k_e)}\right) + \sum_{e=1}^{f}\left(N_e \cdot c_{ins}\right) + n \cdot c_{cru} \cdot \underset{e}{Max} F_e + N^* \cdot c^*\right\} \quad (6.3.1)$$

subject to

$$k_e = s_{ge} \ \forall (i,j)_e : S_{ije} = t_{eg},\ 0 \le g < N,\ 1 \le e \le f, \quad (6.3.2)$$

$$Pr\left\{F_e \le D_e\right\} \ge p_e,\ 1 \le e \le f, \quad (6.3.3)$$

$$t_{0e} = 0,\ 1 \le e \le f, \quad (6.3.4)$$

$$t_{N_e} = D_e,\ 1 \le e \le f, \quad (6.3.5)$$

$$D_e - t_{ge} \ge \Delta_{1e},\ 0 \le g \le N_e,\ 1 \le e \le f, \quad (6.3.6)$$

$$t_{g+1,e} - t_{ge} \ge \Delta_{2e},\ 0 \le g \le N_e,\ 1 \le e \le f, \quad (6.3.7)$$

$$s_{ge} \le s_{ge}^* = \underset{1 \le q \le m}{Min}\left\{q : W_p\left[t_{ge}, q, V_e^f\left(t_{ge}\right)\right]\right\}, \quad (6.3.8)$$

$$\sum_{e=1}^{f_t} n_{et} = n \text{ for any emergency moment } t \ge 0,\ n_{et} \ge 1. \quad (6.3.9)$$

Note that the on-line control model undertakes decision-making either at regular routine control point t_{ge} (determining S_{ije}, $v_{ije}^{(k)}$, $k = s_{ge}$), or at emergency points t_r^* (determining n_{et}, $t = t_r^*$), on the basis of future expenses only, i.e., during the remaining time $D_e - t_{ge}$ (for a single project), or by taking into account values D_e and p_e, $1 \le e \le f$. Past expenses and past decision-makings, are not relevant for the on-line control model. Relation (6.3.3) honors the chance constraints. As to (6.3.8), it refers to the on-line cost-optimization algorithm for a single project (see §6.2). Restriction (6.3.8) means that the speed to be chosen at any routine control point t_{ge} must not exceed the minimal speed s_{ge}^* that enables meeting deadline D_e on time, subject to be chance constraint p_e. It can be well- recognized that operating an activity at a higher speed always results in higher costs to accomplish the activity than by using a lower speed. Thus, (6.3.8) prohibits using unnecessary high speeds. Relation (6.3.9) ensures reallocation of n GRU units at the company's disposal among the non-accomplished projects at any emergency moment $t \ge 0$. Relations (6.3.4-6.3.7) are obvious while (6.3.2) ensures assignment of one and the same speed index k_e to all activities which start processing at a routine control point t_{ge}. Note that an activity cannot start at the moment between two adjacent control points t_{ge} and $t_{g+1,e}$.

6.3.4 Subsidiary models

Consider several important subsidiary models which will be used henceforth.

I. Subsidiary Model A1

The *basic subsidiary Model A1* centers on controlling a single project, without taking into account any resource hiring and maintaining costs. The number of GRU is taken as a fixed and pregiven value. Model A1 is an on-line cost-optimization model and is based on the chance constraint principle. The model and its optimization are outlined in §6.2.

II. Subsidiary Model A2

The model differs from Model A1 by implementing the cost of hiring and maintaining GRU resources within the planning horizon. Thus, objective (6.2.1) is substituted by

$$J = \underset{\{t_g, v_{ij}^{(k)}, s_g\}}{Min} E \left\{ \sum_{(i,j)\in G} \left(c_{ij}^{(k)} \cdot t_{ij}^{(k)}\right) + \left(\underset{(i,j)\in G}{Max} F_{ij}\right) \cdot n\, c_{cru} + N \cdot c_{ins} \right\} \qquad (6.3.10)$$

subject to (6.2.2-6.2.8),

while the on-line heuristic algorithm remains unchanged.

III. Subsidiary Model A3

Determine the minimal number of GRU $n^{(opt)}$ for a single project in order to meet the given chance constraint, i.e.,

$$Min\ n \qquad (6.3.11)$$

subject to (6.2.2-6.2.8).

The Solution

Start ascending value n, beginning from 1. For each n solve Problem A1 taking into account for each activity (i,j) its highest speed $v_{ij}^{(m)}$, i.e., t_{ij} refers to one speed only. Value n, for which relation

$$Pr\left\{ \underset{(i,j)}{Max}\ F_{ij} \le D \right\} < p, \tag{6.3.12}$$

ceases to hold, is taken as the solution. Cost parameters are, thus, not taken into account. Denote the optimal number $n^{(opt)}$ by $n(A3)$.

IV. Subsidiary Model A4

Determine the minimal number of GRU units in order to minimize the objective (6.3.10) for the Model A2 subject to the chance constraint. Thus, two objectives are imbedded in the model

$$Min\ n, \tag{6.3.13}$$

$$J = \underset{\{n,t_g,v_{ij}^{(k)},s_g\}}{Min} E\left\{ \sum_{(i,j)\in G} \left(c_{ij}^{(k)} \cdot t_{ij}^{(k)}\right) + \left(\underset{(i,j)\in G}{Max} F_{ij} \right) \cdot n\,c_{cru} + N \cdot c_{ins} \right\} \tag{6.3.14}$$

subject to (6.2.2-6.2.8).

The Solution

Solve Problem A3 in order to determine value $n(A3)$. Afterwards proceed ascending value n, beginning from $n(A3)$, and for each value $n \ge n(A3)$ solve Problem A2. Value $n(A4)$ which delivers the minimum to (6.3.14) is taken as the solution of Problem A4.

6.3.5 The general idea of the problem's solution

Problem (6.3.1-6.3.9) to be considered is a very complicated problem and allows a heuristic solution only. Denote the optimal solution of problem (6.3.1-6.3.9) by $n(A)$. A basic assertion can be formulated as follows:

Assertion. Let $n_e(A4)$ be the solution of problem A4 for each project $G_e(N,A)$, $1 \le e \le f$, independently. Relation

$$n(A) \le \sum_{e=1}^{f} n_e(A4) = n_{max} \tag{6.3.15}$$

holds.

Proof. Any additional GRU unit which results in exceeding value $\sum_{e=1}^{f} n_e(A4)$, has to be assigned to one of the projects $G_e(N,A)$. For that project, as it turns from *Model A4*, the unit becomes redundant. ■

Thus, the general idea of determining $n(A)$ is based on the following concepts:

Concept 1

At the company level the search for an optimal solution is based on examining all feasible solutions $\{n\}$, by decreasing n by one, at each search step, beginning from $n_{\max}$.

Concept 2

Examining a feasible solution centers on simulating the system. Multiple simulation runs have to be undertaken in order to obtain a representative statistics to check the fitness of the model.

Concept 3

A simulation model comprises two-levels. At the higher level – the company level – Subalgorithm I reallocates n GRU units among f_t non-completed projects at all emergency moments t, beginning from $t = 0$. At the lower level (the project level) Subalgorithm II undertakes on-line control for each project independently between two adjacent emergency points t_r^* and t_{r+1}^*, by the use of a single-project algorithm of problem A2.

Concept 4

Each value n is examined via M simulation runs to provide a representative statistics to calculate values $\Pr\{F_e \le D_e\}$, $1 \le e \le f$, and objective (6.3.14).

Concept 5

The search process proceeds by decreasing n by one, i.e., substituting n by $n-1$, if

- all relations $\Pr\{F_e \le D_e\} \ge p_e$, $1 \le e \le f$, hold;
- value (6.3.14) decreases monotonously.

Concept 6

If even for one project $G_e(N, A)$ relation $\Pr\{F_e \le D_e\} \ge p_e$ ceases to hold, or value (6.3.14) ceases to decrease, the last successful feasible solution n has to be taken as an optimal solution $n(A)$.

6.3.6 *The enlarged procedure of resource reallocation (Subalgorithm I)*

At each emergency point $t \ge 0$ (each emergency point is a control point for all projects as well) reassign n GRU unit among f_t non-accomplished projects as follows:

Step 1. At moment t inspect values V_{et}, $1 \le e \le f$. Note that for already accomplished projects their corresponding values $V_{et} = 0$.

Step 2. By any means reassign n GRU units among f_t projects subject to:

- $\sum_e n_{et} = n$;
- n_{et} must be whole numbers;
- n_{et} must be not less than 1;
- relations $n_{et} \ge \left[n \cdot \frac{V_{et}}{\sum_e V_{et}} \right]$, $V_{et} > 0$, $1 \le e \le f$, hold, where $[x]$ denotes the maximum whole number being less than x.

Thus, Step 2 delivers a *non-optimal*, feasible solution.

Step 3. Take value $Z = 10^{17}$, i.e., an extremely large positive value.

Step 4. For all non-accomplished projects G_{et} solve *Problem A2*, independently for each project, with due dates $D_e - t$, chance constraints p_e, resource units n_e and non-accomplished volumes V_{et}. Denote the *actual* probability of meeting the due date on time by $\bar{p}_e$. Values $\bar{p}_e$, $1 \le e \le f$, are obtained via M simulation runs.

Step 5. Calculate values $\gamma_e = \frac{\overline{p}_e - p_e}{p_e}$, $1 \le e \le f$.

Step 6. Calculate values $\gamma_{\xi_1} = \underset{e}{Max}\gamma_e$, $\gamma_{\xi_2} = \underset{e}{Min}\gamma_e$.

Step 7. Calculate $\Delta = \gamma_{\xi_1} - \gamma_{\xi_2}$.

Step 8. If $\Delta < Z$, go to the next step. Otherwise apply Step 12.

Step 9. Set $Z = \Delta$.

Step 10. Transfer one GRU unit from project $G_{\xi_1 t}$ to $G_{\xi_2 t}$, i.e., $n_{\xi_1 t}$ is diminished by one, and $n_{\xi_2 t}$ is increased by one.

Step 11 is similar to Step 4, with the exception of solving Problem A2 for projects $G_{\xi_1 t}$ and $G_{\xi_2 t}$ only. Return to Step 5.

Step 12. Values n_{et} , $1 \le e \le f$, which refer to the last successful iteration, are taken as the optimal solution of Subalgorithm I.

6.3.7 The enlarged two-level heuristic algorithm of simulating the system

The enlarged step-by-step procedure of the problem's algorithm is based on simulating the system. The input of the simulation model is as follows:

- value $n \ge f$ of GRU units (to be examined by simulation);
- pregiven values D_e, p_e, $1 \le e \le f$;
- speeds' parameters $v_{ije}^{(k)}$, $(i,j)_e \in G_e(N,A)$, $1 \le k \le m$;
- cost parameters $c_{ije}^{(k)}$, c_{ins} , c_{GRU} , c^*;
- target parameters V_e, $1 \le e \le f$.

A simulation run comprises the following steps:

Step 1. Set $r = 1$, $t_r^* = 0$.

Step 2. Reallocate at $t = t_r^*$ n GRU units among projects $G_e(N,A)$, $1 \le e \le f$, according to Subalgorithm I.

Step 3. Reassign values n_{et} obtained at Step 2, to projects $G_e(N,A)$.

Step 4. Each project $G_e(N,A)$ is carried out independently according to the *Problem A2* (see *6.3.4*). In the course of realizing each project any routine control point t_{ge} is examined as follows:

- is moment t_{ge} the moment project $G_e(N,A)$ is completed? If yes, go to Step 9. Otherwise proceed examining inspection point t_{ge}.
- is moment t_{ge} the moment when it is anticipated that project $G_e(N,A)$ cannot meet its deadline on time even by introducing the highest speed with index m? If yes, go to the next step. Otherwise proceed realizing the project until the next routine control point $t_{g,e+1}$.

Step 5. Counter $r + 1 \Rightarrow r$ works.

Step 6. Set $t_r^* = t_{ge}$.

Step 7. Inspect all non-finished projects $G_e(N,A)$ at the routine emergency point t_r^*. Calculate values $V_e^f(t)$, $1 \le e \le f$, $t = t_r^*$.

Step 8. Update all remaining targets $V_e - V_e^f(t) \Rightarrow V_e$, $1 \le e \le f$. Return to Step 2 to undertake resource reallocation among non-accomplished projects.

Step 9. Are there at moment $t = t_{ge}$ other, non-accomplished projects? If yes, go to Step 5. Otherwise apply the next step.

Step 10. The simulation run terminates.

In the course of carrying out Steps 2 and 4 the cost-accumulated value J of objective (6.3.1) has to be calculated.

The problem's solution is, thus, based on implementing procedures described in *6.3.5-6.3.7*.

§6.4 Conclusions

The following conclusions can be drawn from the Chapter:

1. The developed cost-optimization simulation algorithms for solving problems (6.1.1-6.1.8), (6.2.1-6.2.8) and (6.3.1-6.3.9) can be applied to a wide range of both production and project management systems. The outlined models enable managing complicated building and construction systems, various R&D systems with different speeds and inspection points, etc.
2. The developed on-line control model is a generalized model: it satisfies a variety of chance constraints and develops cost-minimization for a broad spectrum of expenses in the course of the system's functioning.
3. The structure of the multilevel algorithm for solving project management problems is as follows: at the system's level (the higher level) a search of the optimal number of GRU units is undertaken. At the project's level a basic cost-optimization model for a single project is implemented in the simulation model.
4. The main connection between those two levels is carried out via a developed resource reallocation subalgorithm. The latter is carried out by undertaking probability control to be as close as possible to the projects' chance constraints.
5. Extensive simulation described in [54,73-74,151] for real industrial plants has proved the fitness of the on-line cost-optimization models outlined in the Chapter.
6. The on-line control algorithms perform well and are mostly effective for projects of medium size. In cases of large projects, we suggest aggregating the initial model in order to transfer the latter to an equivalent one, but of medium- or small-size. After observing the project's output at a routine control point and introducing proper control actions, i.e., determining a new processing speed and the next control point, the aggregated network is transformed back to the initial one, and the project's realization proceeds.

PART III

ALTERNATIVE STOCHASTIC NETWORK MODELS

Chapter 7. The Models' Description and Structure

§7.1 Introduction

As outlined above, alternative stochastic network models are characterized by two main features:

a) very high level of indeterminacy;

b) various types of branching nodes in key events.

Examine both properties in greater detail. In case of an innovative "brainstorming" the researchers examine the results which, at the outset, are basically indeterminate, and very often it is impossible to determine the ultimate project's goal. For such kind of an R&D project the control system should be inherently adaptable and flexible, seeking step-by-step the best route to meet the target. In cases of such an R&D it is impossible either to determine the initial network leading to the goal, or even to initiate the structure of such a network. At the initial stage of the project's realization, the network may be restricted to a source node and several alternative terminal (sink) nodes. In certain cases the network may contain several milestones (a decision-tree model) which are usually linked to extensive experimentation with alternative and unpredictable results. Such a stochastic alternative network is renewed permanently over time, including changes of the ultimate goals [54]. At each decision node, in the course of carrying out the project, the project's manager has to choose the optimal outcome. Decision-making is repeatedly introduced at every sequentially reached decision node.

Note that an R&D innovative project as mentioned above usually possesses both features a)-b) altogether. However, in certain cases the project's goals are ultimated beforehand, but carrying out the project meets in the course of its progress a variety of milestones boiling down to undertaking complicated geological surveys, pioneering high-tech experiments with alternative unpredictable outcomes, etc. Those projects, being innovative as well, refer usually to long-term construction projects, e.g., constructing a major Arctic pipeline [26,54], etc. Such projects are characterized with various types of branching nodes in key events.

Thus, it can be well-recognized that nowadays alternative stochastic network models occupy the main seat in R&D innovative projecting where indeterminacy and alternativity often meet. In our opinion, such models should become must items in the methodological portfolio of any modern high-tech company.

Note that while the literature on PERT and CPM network techniques is quite vast, the number of publications on *alternative networks* remains very scanty. The first significant development in that area was the pioneering work of Eisner

[37] in which a "decision box" with both random and alternative outcomes and PERT nodes was introduced. An example of such a network is represented on Fig. 7.1. Numbers above and below the arcs denote the corresponding time and probability, respectively. Eisner used the term *decision box* (DB) to refer to nodes that lead to alternatives (corresponding to the term *probabilistic branching* used in scientific literature). The realization of terminal events A and B depends not only on the outcome of $DB3$ but also on the outcome of $DB2$. This illustrates Eisner's concept of *conjunctive path dependency*, which arises when the planned work on one path depends on the answer to a particular DB on a different path. It led him to the duplication of Node 2 and the introduction of dummy Node 2_1, with the understanding that the two nodes represent, in fact, a single one. Thus, event A will be realized if the outcome at $DB3$ is NO *and* the outcome at $DB2$ is also NO, while event B will be realized if the outcome at $DB3$ is NO *and* the outcome at $DB2$ is YES.

The logical relationships governing the outcomes were given by Eisner [37] as follows:

$$outcomes = \{[(A \cup B) \cup X] \cap (Y \cup E)\} \cup \{(G \cup H) \cup F\},$$

where $(X \cap Y) = (C \cup D)$.

Here, "$\cup$" represents the disjunction operation and "$\cap$" stands for the conjunction operation. Note a special logical relationship between branches X and Y and $DB4$; Node 4 will be realized if *both* activities are realized. This is equivalent to AND relationship in Eisner's terminology.

Given the above structure, it is easy to calculate the possible final outcomes:

A and E	(0.0840)
B and $Y \to B$	(0.1260); since 4 cannot be realized
C	(0.0882)
D	(0.2058)
E and $X \to E$	(0.1960); since 4 cannot be realized
F	(0.1500)
G	(0.0600)
H	(0.0900)

The compound outcomes may be explained as follows. The outcome "A and E" occurs if $DB3$ yields NO and $DB2$ yields NO also; the outcome "B and $Y \to B$" occurs if $DB3$ yields NO and $DB2$ yields YES, in which case activity X will not be realized and, consequently, node 4 cannot be reached; finally, the outcome "E and $X \to E$" occurs if $DB3$ yields YES and $DB2$ yields NO, in which case activity Y will not be realized and, consequently, node 4 cannot be reached.

However, Eisner did not develop algorithms to implement decision-making in R&D projects comprising decision nodes. Elmaghraby [38] introduced additional logic and algebra in network techniques. His representation of the same logic is different, as demonstrated in Fig. 7.2. It differs from Eisner's representation in two major aspects. First, Elmaghraby managed to avoid duplication of *DB* s by adding dummy nodes and arcs, which can always be done to represent any desired relationship among the outcomes of *DB* s. Second, the implied logical relationships are brought into sharper focus.

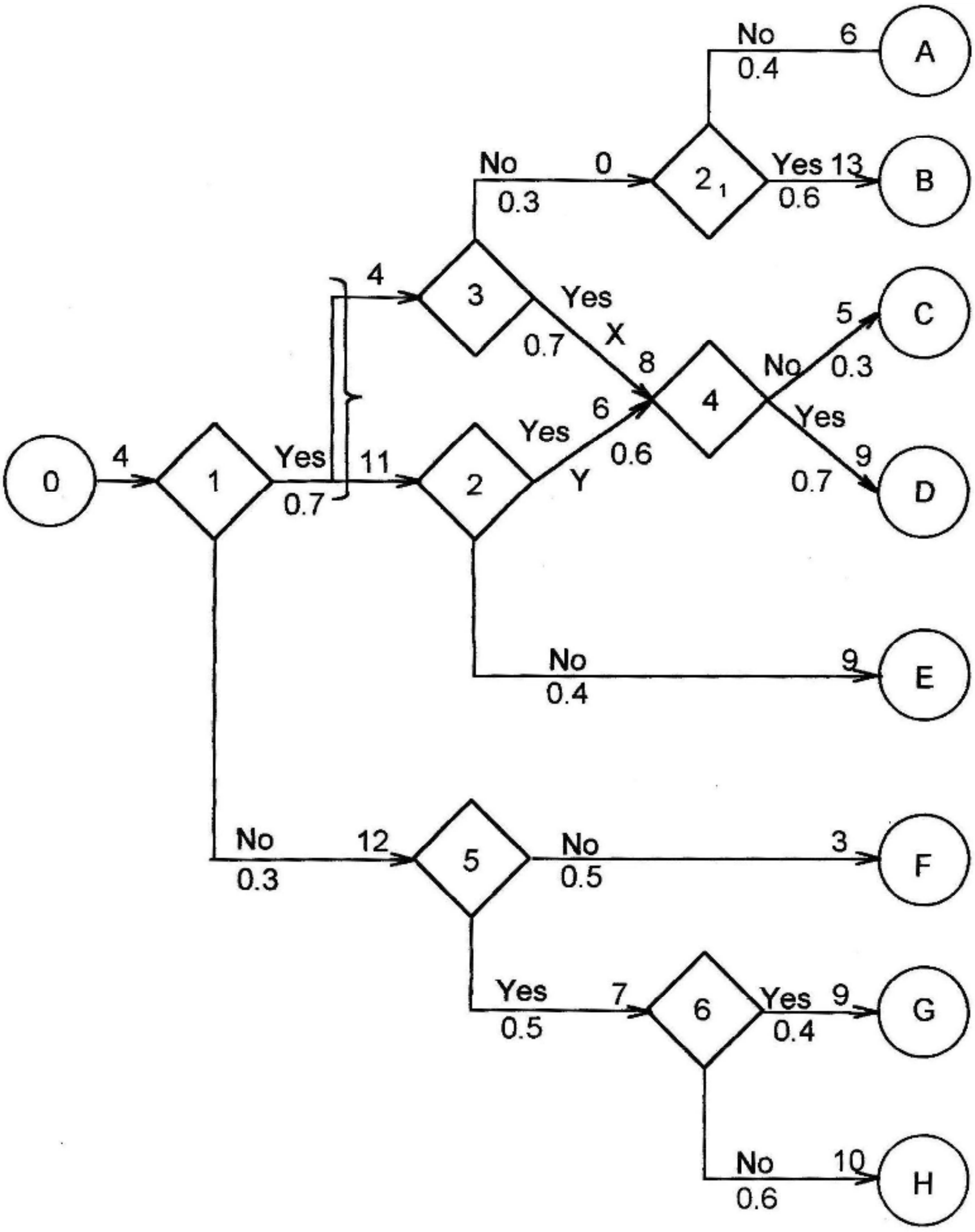

Figure 7.1. An example of Eisner's network with a priori probabilities

Pritsker, Happ and Whitehouse [131-133] developed the GERT techniques for alternative network models with stochastic outcomes in key nodes. When all the nodes of an alternative network are of the "exclusive-or" type on their receiving side, we obtain a GERT (Graphical Evaluation and Review Technique) model. Other nodes can be reduced to "exclusive-or" nodes; thus, the GERT model is quite general.

The GERT network is in fact the representation of a semi-Markov process (SMP). The network itself, after a simple transformation of variables, is a signal flow graph (SFG). Both objects (SMPs and SFGs) have rich mathematical structures.

Xespos and Strassman [166] introduced the concept of a stochastic decision tree, while Crowston and Thompson [28-30] and more recently Hastings and Mello [99] introduced the concept of multiple choices at alternative nodes, when decision-making is of deterministic nature (Decision-CPM models). These networks are characterized by discrete multiple choices at some of their nodes. They may represent either a choice among activities to be undertaken next or a choice among sets of resources to be utilized by the activity itself. In the former case, one or more of the prospective activities must be undertaken. Those activities which were not selected should "disappear" from the network in the sense that all their precedence relations must be eliminated. In the latter case it is evident that allocation planning of resources is intimately related to the scheduling of activities, since the very duration of these activities is dependent on the resources allocated to them.

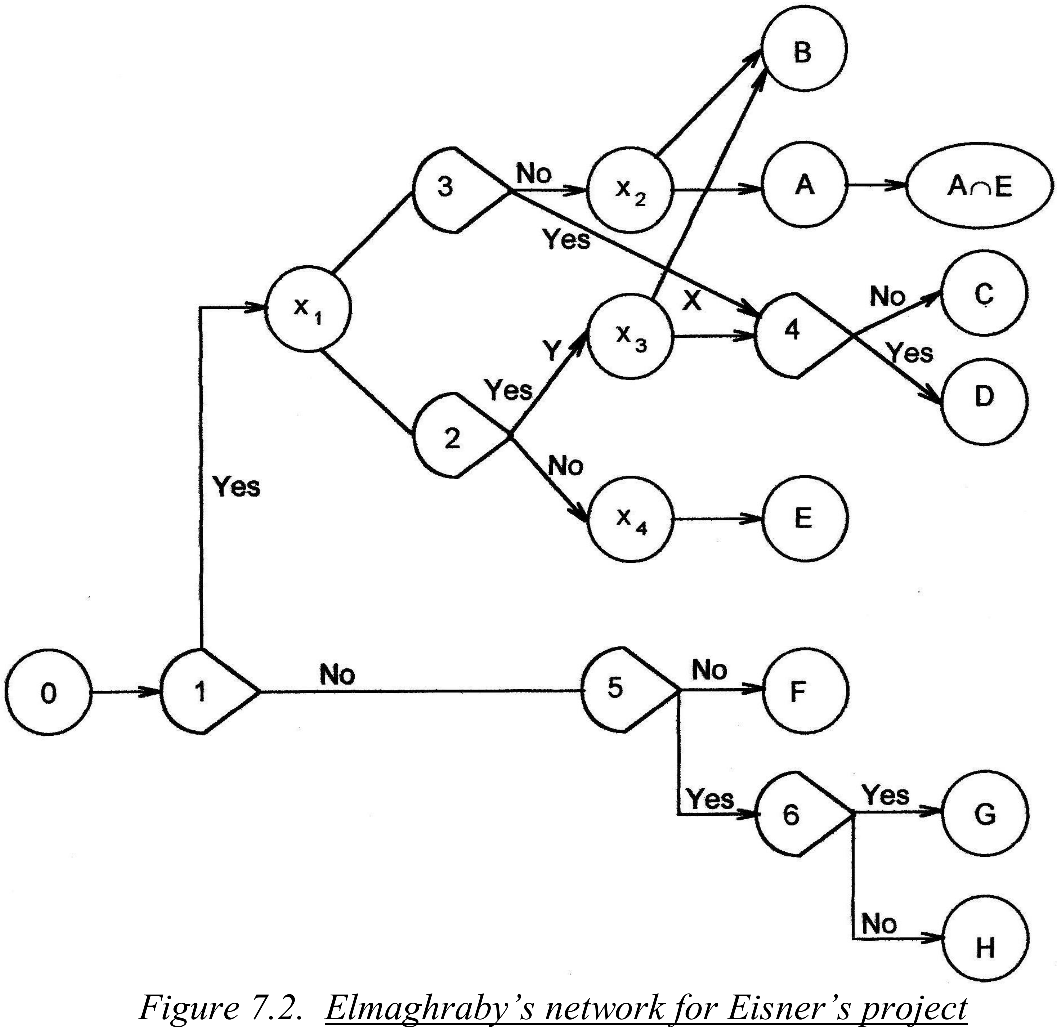

Figure 7.2. Elmaghraby's network for Eisner's project

Lee, Moeller and Digman [111,123] developed the VERT model that enables the analyst to simulate various decisions with alternative technology choices within the stochastic decision tree network. In the VERT model, two basic symbols are used to represent the structure of the network model:

a) nodes represent milestones or direction points, and
b) arcs represent activities which are basically characterized by three parameters, namely:
   1) time,
   2) cost, and
   3) performance factor of executing this activity.

In the VERT model, the network is nothing but a schematic flow device in which the nodes (decision points) channel or gate the flow into arcs (activities) which carry the flow from an input node to an output node. The flow throughout the network represents an actual execution of these activities and milestones which the flow has traversed.

VERT has two types of nodes, which either start, stop or channel the network flow. The most commonly used type is called the split logic node. It has separate input and output logics describing specific types of input and output operations. The second, more specialized and less frequently used type of nodes, has a single-unit logic which covers both input and output operations, simultaneously. There are four basic input logics available for the split-logic nodes:
1) INITIAL;
2) AND;
3) PARTIAL AND; and
4) OR.

They are described as follows [111]:
1. INITIAL input logic serves as a starting point for the network flow. Multiple initial nodes may be used. All initial nodes are assigned with the same time, cost and performance values by the user.
2. AND input logic requires all the input arcs to be successfully completed.
3. PARTIAL AND input logic is nearly the same as AND input logic except that it requires a minimum of one input arc to be successfully completed.
4. OR input logic is similar to the PARTIAL AND logic. It also requires just a minimum of one input arc to be successfully completed. The logic, however, will not wait for all the input arcs to come in.

There are six basic split-node output logics available to distribute the network flow to the appropriate output arc. The cumulative time, cost and performance values computed for the active output arcs consist of the sum of the time, cost and performance factors carried by the input node of the arc.

Moeller and Digman applied the VERT model to an operational planning problem - the evaluation of electric power generation methods [123]. Three alternative methods have been considered: nuclear fusion, nuclear fission and coal gasification. The overall time required and the cost incurred for completing the entire project are the objective function and the constraint, respectively. Also, various confidence probability levels were incorporated into the regarded research. Based on examining optimal values of the objective function with respect to the confidence probabilities for the compared technologies to successfully put through certain tests, the USA Federal Power Commission selected the winning technology (the fusion process) over the alternatives. The VERT model has also been successfully applied to weapon system developments, including tanks, helicopters, fighter planes, artillery, self-propelled howitzers, electronic sensors, air defense systems, and others [123].

Thus, it can be well-recognized that the VERT model has excellent software and a good application area. However, similar to the GERT model, on-line decision-making, as well as introducing control actions in decision-nodes, does not take place.

The next step has been made by Golenko-Ginzburg [49-57] who developed the novel controlled alternative activity network (CAAN model) for projects with both random and deterministic alternative outcomes at key nodes. At each routine decision-making node, the developed algorithm, based on lexicographic scanning, singles out all the sub-networks (the so-called joint variants) that correspond to all possible outcomes from that node. The joint variants of the CAAN model are enumerated by introducing a lexicographic order to the set of maximal paths in the CAAN graph. The corresponding lookover algorithm is very simple in usage. Decision-making results in determining the optimal joint variant and following the optimal direction up to the next decision-making node.

§7.2 Alternative stochastic model's description

The alternative CAAN network model [49,51,54] is a finite connected, oriented acyclic graph $G(U,Y)$ with the following properties:

(1) Graph G has one initial event, y_0 (the network entry), for which $\Gamma^{-1}y_0 = \emptyset$ and $\Gamma y_0 \neq \emptyset$.

(2) Graph G contains a set Y ‘ of events y ‘ (called terminal events, or network exits), where $\Gamma y' = \emptyset$, $\Gamma^{-1}y' \neq \emptyset$ and $|Y'| \geq 2$.

(3) The set of events Y of graph G is not uniform and consists of events of type $\tilde{\chi} \in \tilde{X}$ (classical PERT model) and of more complex logical types, $\tilde{\alpha} \in \tilde{A}$, $\tilde{\beta} \in \tilde{B}$, and $\tilde{\gamma} \in \tilde{\Gamma}$, being represented in the below Tab. 7.1:

Table 7.1. Logical possibilities of alternative network model events

Designation of an event in the model	Logical relations at the event's receiver	Logical relations at the event's emitter
$\tilde{\chi}$	and	and
$\tilde{\alpha}$	and	exclusive "or"
$\tilde{\beta}$	exclusive "or"	and
$\tilde{\gamma}$	exclusive "or"	exclusive "or"

(4) The set of arcs U of graph G is split into a subset U' of arcs corresponding to the actual functioning of the alternative network, and subset U'' of arcs representing the logical interconnections between actual and imaginary

functions.

(5) Vector W_{kl} of values characterizing actual work is constructed preliminary for every arc, $U_{kl} \in U'$, representing an actual activity. Among such values are the time of the activity duration t_{kl}; the required cost C_{kl}; and other components of this vector. The vector's components $\omega_{kl}^{(\rho)}$ ($\rho = 1-k$, k being the vector' dimension) can be represented, depending on the degree of indeterminacy, either by determined estimations or by random values with a given distribution function, $f\left(\omega_{kl}^{(\rho)}\right)$, on the interval $\left[\alpha\left(\omega_{kl}^{(\rho)}\right),\ \beta\left(\omega_{kl}^{(\rho)}\right)\right]$, where $\alpha\left(\omega_{kl}^{(\rho)}\right)$ and $\beta\left(\omega_{kl}^{(\rho)}\right)$ are boundary estimations of the ρ-th component of vector W_{kl}.

(6) For the stochastic alternative model of a combined type, the set of alternative events, $\widetilde{A} \cup \widetilde{\Gamma}$, is split into subsets $\overline{A}$ - alternative events that show the branching of determined variants, and $\overline{\overline{A}}$ - alternative events that represent the situations of branching stochastic variants, where $\widetilde{A} \cup \widetilde{\Gamma} = \overline{A} \cup \overline{\overline{A}}$.

(7) When the network event is of alternative nature, it is assigned a set of estimations of corresponding local variant probabilities. In other words, a non-negative number, $\overline{\overline{p}}_{ij} \le 1$, such that $\sum_{j=1}^{n_i} \overline{\overline{p}}_{ij} = 1$ (where $\overline{\overline{p}}_{ij}$ is the *a priori* probability of transferring from i to j and n_i stands for the number of local variants appearing in event i), is related to each alternative path starting from event i of type $\widetilde{\alpha} \in \overline{\overline{A}}$ or $\widetilde{\gamma} \in \overline{\overline{A}}$ and leading to outcome j.

(8) If event i is related to an alternative event of class $\widetilde{A}$, the corresponding conditional transfer probability, $\overline{\overline{p}}_{ij}$, is usually assumed to be equal one. This means that the process of choosing the direction in which the system has to move towards its target is of a determined character; it is the prerogative of the system's controlling device.

Problems of alternative network model analysis and synthesis are solved by applying the principle of network enlarging and obtaining a special graph - the outcome tree [49-57], which is usually designated as $D(A,V)$ and represents a graph that can be constructed by modifying the original model, $G(Y,U)$, as follows:

(a) The set, which consists of the initial event, finite events, and events that are branching points of alternative paths of graph G, is taken as the set of events of graph D. The initial event, $\alpha_0 = y_0$, is called a hanging event.

(b) The set of arcs $V = \{v_{ij}\}$ of graph D is obtained through an equivalent transformation of a set of sub-graphs, $\{G_{ij}\}$, extracted from network G according to the following procedure:

- any event α_i, except for the finite ones, α', can be the initial event of sub-graph $G_{ij} = (L_{ij}, U_{ij})$, where $\alpha' \in y_{ij}$ and $\Gamma^{-1}\alpha_i \cap Y_{ij} = \emptyset$;

- $Y_{ij} \subset \tilde{\Gamma}\alpha_i$, where $\tilde{\Gamma}\alpha_i$ stands for the transitive closure of mapping α_i;
- only an α-event of graph G, except for the initial event, $\alpha_0 = y_0$, can be a finite event of sub-graph G_{ij}, and
- no $(\alpha_i, ..., \alpha_j)$-type paths that connect the initial event, α_i, with sub-graph finite event α_j in G_{ij}, contain other α-events of graph G.

(c) every arc, v_{ij}, of outcome tree D is obtained by reducing fragment G_{ij} of network $G(Y,U)$ to one arc beginning at α_i and ending at α_j. In addition, realization probability p_{ij}, fulfilment time t_{ij}, and other parameters equivalent to the corresponding characteristic values for initial fragment G_{ij} are brought into correspondence with the enlarged arc v_{ij}.

If different fragments, G_{ij}, of the model do not intersect, the alternative network is called entirely divisible; all events of the corresponding outcome tree prove to be γ-type events.

We will require a supplementary definition. A *partial variant* is a variant of the network model's realization; it corresponds to a definite direction of its development at an individual stage, characterizes one of the possible ways of reaching the intermediate target, and does not contain alternative situations. The variant of realization of the whole project, which does not contain alternative branchings and is formed by a sequence of partial variants, is called a *full variant*. On the outcome tree, $D(A,V)$, a certain arc, v_{ij}, corresponds to the partial variant, while some path connecting root event α_0 with one of the hanging events, corresponds to the full variant.

The combined outcome tree, $D(A,V)$, can be regarded as a union of purely stochastic outcome trees that reflects some homogenous alternative stochastic network models. The latter are obtained by choosing different directions in the controlled devices. Such stochastic outcome trees, which are all part of the combined outcome tree, $D(A,V)$, are called *joint variants* of realizing the stochastic network model.

The joint variant can be extracted from the original graph, $D(A,V)$, by "fixing" certain directions in interconnected events of type $\bar{\alpha}$ and excluding unfixed directions. In other words, every joint variant can be regarded as a realization variant of the network model. Such a variant has a determined topology, but it contains probability situations and has certain possible stochastic finite states.

§7.3 Logical operations in alternative networks

Let us single out three main logical operations which can be realized at both receiver and emitter in different nodes of an alternative network. The logical operations are as follows [157]:

1. Operation “And” has a “must follow” emitter for all activities leaving a certain node and the “And” receiver for all operations entering a node; thus, all activities entering the node or leaving the node are realized.
2. Operation “Exclusive Or” enables only one activity to be realized from a set of activities entering a node or leaving a node. Operation “Exclusive Or” is, in turn, subdivided into two classes:
   a) Operation “Stochastic Exclusive Or” which we will henceforth denote by “Or”. Each alternative activity entering a set corresponds to a certain probability value while a set of activities is a full group of events. The choice of an alternative activity at the node’s receiver or emitter is carried out by a random trial in accordance with the activities’ probability values. Each set comprises not less than two alternative activities.
   b) Operation “Deterministic Exclusive Or” which we will henceforth denote by “Or**”. The choice of an alternative activity from a set of activities at the receiver or at the emitter is carried out by the project manager. Each set, like in case a), comprises not less than two activities.

Besides the outlined above three logical operations, alternative networks may comprise nodes with additional logical operations, namely, various combinations of those operations:

- Operation “And + Or*”. Two different sets of activities are either entering a certain node or leaving a node. All activities entering the first set have to be realized while only one activity has to be chosen from the second set on the basis of a random trial.
- Operation “And + Or**”. The difference between this operation and the previous one is that the choice of an activity from the second set is carried out by the project manager.
- Operation “Or*+ Or**”. Two alternative sets of activities are either entering a node or leaving a node. The choice of an alternative activity from the first set is of random nature and is uncontrolled, while for the second set, choosing an alternative activity is a control action.
- Operation “And + Or*+ Or**”. Three sets of activities are entering or leaving a certain node. All activities entering the first set have to be realized while the choice of an alternative activity from the second and the third sets are carried out by means of random trials and control actions, correspondingly.

Thus, our classification of different alternative networks is based on the combinations of different logical operations at the nodes’ receivers and emitters, as it is shown on Tab. 7.2. Note that in operations 4-7 “together with” may be substituted for “Or” either at the receiver, or at the emitter (but not for both outcomes simultaneously).

In conclusion, we will outline the logical operations of several alternative networks.

1. From the practical point of view, PERT and CPM networks comprise activities (i, j) with the logical "must follow" emitter at node i and the "And" receiver at node j. This means that:
   a) an event may occur only at the moment the last activity entering the event is finished;
   b) all activities leaving any event of the network have to be operated.
2. In Decision-CPM network models [28-30] all events have an "And" receiver, while certain events have controlled deterministic alternative outcomes. Thus, the choice of an alternative network is supervised by the project management.
3. Network models GERT (Graphical Evaluation and Review Techniques) [131-133] besides the logical "And" receiver and "must follow" emitter, comprise certain events with "Stochastic Exclusive Or" either at the emitter or at the receiver, or both at the emitter and the receiver together. The choice of an alternative activity is realized by a random trial of a full group of events with fixed probabilities.
4. Model CAAN (Controlled Alternative Activity Network) [49-57] comprises, besides events with the logical "must follow" emitter and the logical "And" receiver, certain events with "Exclusive Or*" of stochastic nature at the receiver or at the emitter. Certain other events entering the model have an "Exclusive Or**" receiver or emitter. But there are no events which comprise simultaneously two types of alternative sets of activities of "Exclusive Or*" and "Exclusive Or**" entering or leaving one and the same node.
5. The GAAN (General Alternative Activity Network) model [9,53-54,67] has been already mentioned above, in §1.1, and will be outlined in depth below, in Chapter 9.
   The class of GAAN models is the most general one from the point of its alternative structure.
   The GAAN and CAAN models have been successfully used for planning and controlling highly complicated R&D projects [67] where decision-making has to be introduced with incomplete or inadequate information about the alternatives. Such models are especially effective for R&D projects with multiple alternative technology choices, e.g., in opto-electronics, aerospace, defense related industries, in developing an artificial heart [67], in projecting new software (Information Technology Projects), etc.
6. Model SATM (Stochastic Alternative Time-Oriented Network) [53,157] is a further extension of the generalized network models GNM and GAAN. SATM differs from GNM by:
   - implementing various types of alternative relations (stochastic or deterministic alternatives);
   - implementing a broad spectrum of stochastic values.

Note, in conclusion, that stochastic alternative network models CAAN, GAAN and SATM comprise alternative deterministic branching nodes and, thus, refer to the class of controllable network models. However, there is an essential difference between CAAN and GAAN, on one side, and SATM, on the other one. Both CAAN and GAAN models, independently from their structure, enable obtaining feasible solutions and, thus, can be optimized, as outlined below, in Chapters 8 and 9. In the case of SATM models certain combinations of parameters do not provide feasible solutions, i.e., *the project cannot be carried out*. We have to implement a new concept - the *project's availability*, i.e., the probability to ever meet the target. Thus, we determine a new definition of the project's p-availability which reflects the probability for the project to be realized. In our opinion, value p can be determined by means of extensive simulation. If p is close to unity, e.g., $p \cong 0.99$, the regarded SATM model may be accepted for future analysis. In case $p \cong 1$, we suggest optimizing SATM *on condition* that the project will be accomplished. Such a logical analysis is necessary since SATM incorporates the GNM model with its complicated logical relations and links. If p deviates from unity essentially, we have to overlook the structure of SATM, i.e., to amend the network's structure. This by itself is an extremely complicated problem which remains unsolved as yet.

Table 7.2. Logical operations in alternative networks

Event's receiver	Logical operation	Event's emitter
"And"	1: "And"	"Must follow"
"And" or "Stochastic Exclusive Or"	2: "Or*"	"Must follow" or "Stochastic Exclusive Or"
"And" or "Deterministic Exclusive Or"	3: "Or**"	"Must follow" or "Deterministic Exclusive Or"
"And" together with "Stochastic Exclusive Or"	4: "And + Or*"	"Must follow" together with "Stochastic Exclusive Or"
"And" together with "Deterministic Exclusive Or"	5: "And + Or**"	"Must follow" together with "Deterministic Exclusive Or"
"Stochastic Exclusive Or" together with "Deterministic Exclusive Or"	6: "Or* + Or**"	"Stochastic Exclusive Or" together with "Deterministic Exclusive Or"
"And" together with "Stochastic Exclusive Or", together with "Deterministic Exclusive Or"	7: "And + Or* + Or**"	"Must follow" together with "Stochastic Exclusive Or", together with "Deterministic Exclusive Or"
Types 1-7 together with "Generalized Time-Oriented Network Model" (GNM)	8: "And + Or*+Or** + GNM"	Types 1-7 together with GNM

In the next chapter we will outline the most frequently used nowadays controlled alternative activity network - the CAAN model [49-57].

Chapter 8. Controlled Alternative Activity Network (CAAN)

§8.1 The model's description

8.1.1 Structure of a CAAN model

A CAAN model is a finite, connected, oriented, activity-on-arc network $G(N, A)$ with the following properties:

I. Network $G(N, A)$ has one source node n_0 and not less than two sink nodes n^*.

II. The set of nodes of network $G(N, A)$ includes four types of nodes:

Type 1: with the logical "And" receiver and the "'must follow" emitter;
Type 2: with the logical "And" receiver and the "exclusive or" emitter;
Type 3: with the "exclusive or" receiver and the "must follow" emitter;
Type 4: with the "exclusive or" receiver and the "exclusive or" emitter.

III. The set of alternative nodes (types 2 and 4) for a CAAN model is subdivided into subsets:

(a) $\overline{\overline{N}} \subset N$ - alternative nodes which show the branching of stochastic variants;

(b) $\overline{N} \subset N$ - alternative nodes which show the branching of deterministic variants.

When a network node i refers to class $\overline{\overline{N}}$, it is assigned a set of corresponding outcome probabilities $p_{ij} < 1$, $\sum_j p_{ij} = 1$, $i \in B(j)$, where $B(j)$ denotes the set of nodes that connects i to j. When a network node i refers to class $\overline{N}$, the corresponding transfer probabilities p_{ij} are assumed to equal unity. This means that the process of choosing the alternative direction is of deterministic nature; it is the sole prerogative of the project's decision-maker.

According to [49-57], we will henceforth designate the branching nodes, being included in classes $\overline{N}$ and $\overline{\overline{N}}$, as $\overline{\alpha}$ and $\overline{\overline{\alpha}}$, respectively. Note that $\overline{\alpha}$ and $\overline{\overline{\alpha}}$ differ from i only by a special mark which points out their belonging to different sets $\overline{N}$ and $\overline{\overline{N}}$.

8.1.2 Outcome graph

To analyze the CAAN type model we use a special network which we will call the outcome graph. The latter is designated as $G^*(N^*, A^*)$ and can be obtained by reducing the initial network $G(N, A)$. Relation $N^* = n_0 \cup \{n^*\} \cup \overline{N} \cup \overline{\overline{N}}$ holds, i.e., the set of nodes of the outcome graph includes the source node, the sink nodes and all the branching nodes. Every arc $(i, j) \in A^*$ of the outcome graph is equivalent to a certain fragment G_{ij} of the initial network $G(N, A)$. If different fragments $G_{ij} \subset G(N, A)$ do not intersect, both $G(N, A)$ and $G^*(N^*, A^*)$ are called entirely divisible. An example of a CAAN type entirely divisible outcome graph with both connecting and diverging paths is shown in Fig. 8.1.

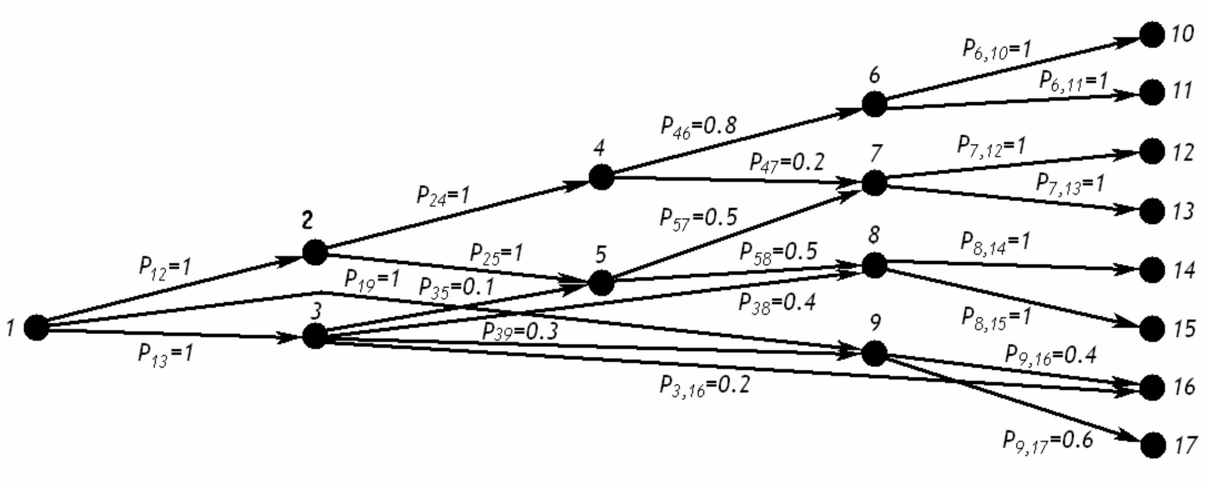

Figure 8.1. The outcome graph

Definition. We will introduce the concept of the *direction of the arc* leaving an α-type branching node. All arcs (i,j) leaving node $\overline{\alpha_i}$ or $\overline{\overline{\alpha_i}}$ are indexed clockwise as $h_{ij} = 1,2,...,n_i$, where n_i is the number of outcomes in node α_i. Thus, the direction of arc (i,j) is equal to the corresponding index h_{ij}.

For example, in Fig. 8.1, arcs leaving node 1 have directions as follows: $h_{12} = 1$, $h_{19} = 2$, $h_{13} = 3$; arcs leaving node 3: $h_{35} = 1$, $h_{38} = 2$, $h_{39} = 3$, $h_{3,16} = 4$, etc.

An algorithm described in [49] transforms an initial activity network to its outcome graph. The algorithm's output information about every arc $(i,j) \in A^*$ (Array I) consists of records: α_i, α_j, h_{ij}, p_{ij}, t_{ij}, c_{ij}.

Here:

- α_i and α_j are the source and terminal nodes of arc (i,j);
- h_{ij} is the direction of arc (i,j);
- p_{ij} is the arc's probability; $p_{ij} = 1$ if $\alpha_i \in \overline{N}$.
- t_{ij} and c_{ij} are the time duration and cost values of the activity represented by arc (i,j).

If necessary, other parameters can be added to the record data (resource or reliability values, etc.).

Definitions. A *partial* (*local*) *variant* is a variant of the node's realization. It corresponds to a definite direction of the project's development at a particular stage. The variant of realization of the whole project, which does not contain alternative branchings and is formed by a sequence of local variants, is called a *full (overall) variant*. On the outcome graph $G^*(N^*, A^*)$, a certain arc (i,j) corresponds to the local variant, and a path connecting source node n_0 with one of the sink nodes corresponds to the full variant. An outcome graph can be regarded as an ensemble of purely stochastic networks with branching nodes of $\overline{\overline{\alpha}}$-type only. These networks are obtained by choosing different directions in the $\overline{\alpha}$-type controlled nodes. Such stochastic networks which are part of the outcome graph, are called *joint variants* of the CAAN model.

Thus, the joint variant can be extracted from the graph $G^*(N^*, A^*)$ by fixing certain non-contradictory directions in interconnected nodes of type $\bar{\alpha}$ and excluding unfixed directions. For example, in Fig. 8.1, one of the joint variants is determined by fixing non-contradictory directions $1 \to 2$, $2 \to 4$, $6 \to 10$, and $7 \to 12$ in $\bar{\alpha}$-nodes 1, 2, 6 and 7. This is presented in Fig. 8.2.

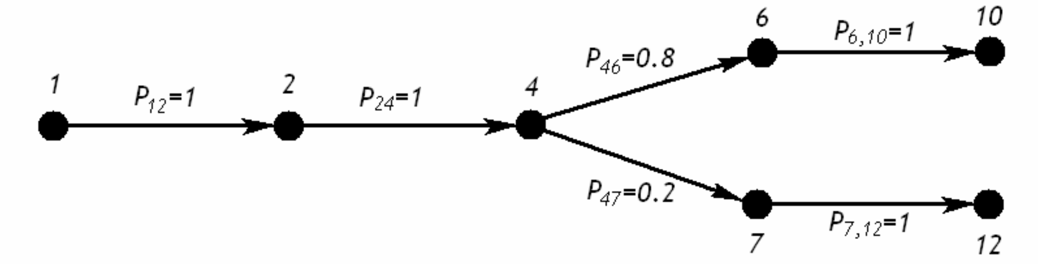

Figure 8.2. The joint variant

§8.2 Decision-making in CAAN type models

8.2.1 Optimization problem

To control a project, such as any production process, it is necessary to introduce decision-making in order to reach the goal while optimizing a given objective function subject to certain restrictions, e.g., minimizing the cost of the project or the time duration. For a project represented by a CAAN type model, decision-making means choosing the directions of the project's development in controlled nodes of $\bar{\alpha}$-type, since $\bar{\bar{\alpha}}$-type nodes are uncontrollable. This means that the optimization problem resolves on choosing a joint variant optimizing the value of the objective function, subject to introduced restrictions.

Let $\bar{N}^* = [\bar{\alpha}_1, \bar{\alpha}_2, ..., \bar{\alpha}_m]$ be the set of $\bar{\alpha}$-nodes of outcome graph $G^*(N^*, A^*)$. Every joint variant is defined by a choice of certain directions in some of these nodes $\alpha_{i_1}, ..., \alpha_{i_r}$ (non-contradictive ones), i.e., by set

$$V = [\bar{\alpha}_{i_1}, h_{i_1 q_1}, ..., \bar{\alpha}_{i_r}, h_{i_r q_r}]. \tag{8.2.1}$$

Definition. A set which indicates the set of m-nodes and the directions in them, and uniquely defines the joint variant, is called an *admissible plan*. The set of joint variants is in one-to-one correspondence with the set of admissible plans. Thus, the optimization problem consists of three steps:

Step 1. To determine and to single out from the outcome graph all the joint variants, together with the corresponding admissible plans.

Step 2. To calculate the values of the objective function and the restrictions (usually in the form of average values) for each variant.

Step 3. To determine the optimal joint variant and to follow the optimal direction up to the nearest deterministic branching node. The problem must be repeatedly solved for the reduced network in every sequentially encountered controlled event of type $\bar{\alpha}$.

Let us consider, for example, that the joint variant in Fig. 8.2 is determined as an optimal one for the initial outcome graph in Fig. 8.1. Beginning the project from node 1, we have to follow direction $1 \to 2$ and reject all other alternative directions, namely $1 \to 3$ and $1 \to 9$, with all the following arcs (3, 5), (3, 9), (3, 8), (3, 16), (9, 16), (9, 17).

However, when the project reaches node 2, we have to resolve the optimization for the remaining part of the network (see Fig. 8.3). In the course of developing the project there may be changes in the parameters of some activities (time duration, cost, outcome probability, etc.) since an activity network is usually revised over time. Perhaps, later on, we may choose $2 \to 5$ as an optimal direction at node 2 instead of direction $2 \to 4$, which was determined at node 1.

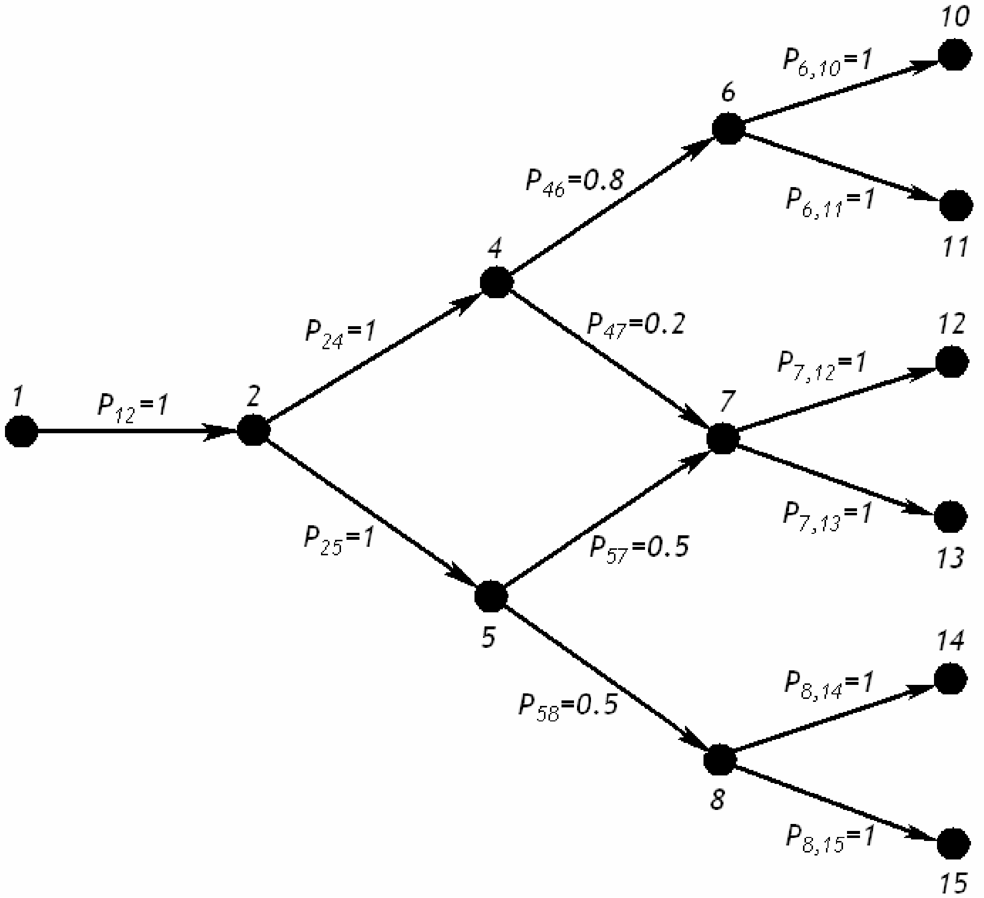

Figure 8.3. <u>The reduced network</u>

8.2.2 <u>Mathematical formulation</u>

The mathematical formulation of the problem is as follows: Determine the joint variant s^* optimizing the mean value of the objective function

$$E\left[F\left(s^*\right)\right] = \underset{s \in \nabla \subset G^*\left(N^*, A^*\right)}{Min\ (Max)} \sum_{\pi_{is} \in \Omega_s} p_{is} F\left(\pi_{is}\right) \tag{8.2.2}$$

subject to

$$E\left[H\left(s^*\right)\right] = \sum_{\pi_{is^*} \in \Omega_{s^*}} p_{is^*} H\left(\pi_{is^*}\right) < H\,. \tag{8.2.3}$$

Here:

Ω_s - set of full variants entering the s-th joint variant;

∇ - set of joint variants entering the $CAAN$ type model;

p_{is} - probability of realizing the i-th full variant π_{is} in the s-th joint variant;

$F(\pi_{is})$ - value of the objective function for the i-th full variant in the s-th joint one;

$H(\pi_{is^*})$ - value of the restriction for the i-th full variant $\pi_{is^*} \in \Omega_{s^*}$;

H - pre-set restriction level. When F is the time duration, restriction H is usually the project's cost, and vice-versa. If necessary, several restrictions can be used.

When all the joint variants are determined (Step 1), one has to examine each of them to look through all the full variants entering the joint variant under examination (Step 2). But since any joint variant contains only alternative nodes of type $\overline{\overline{\alpha}}$, the problem reduces to an analysis of a pure stochastic network, which can be performed with the help of various approaches. Burt, Garman, Gaver and Perlas [18-19] as well as Golenko (Ginzburg) [49-50, etc.] presented various algorithms to solve this problem. Thus, carrying out Step 2 causes no principal difficulties. The main problem is to single out all the joint variants, especially in the case of large initial networks with many branching nodes of both types $\overline{\alpha}$ and $\overline{\overline{\alpha}}$.

8.2.3 Determining joint variants

The procedure of performing Step 1 (see *8.2.1*) boils down to sequential use of the following three algorithms:

Algorithm I. Constructing an $\overline{\alpha}$-frame for the outcome graph $G^*(N^*, A^*)$.

Algorithm II. Determining maximal paths in the $\overline{\alpha}$-frame.

Algorithm III. Determining admissible plans and joint variants.

The algorithms are organized so that the outcome information of each algorithm serves as the initial data for the next one. The initial data for Algorithm I is the information about the outcome graph $G^*(N^*, A^*)$ (Array 1). Note that the procedure can be applied only to a fully divisible network.

§8.3 Algorithm I for constructing an $\overline{\alpha}$-frame

8.3.1 Definitions

Definition 1. We will call a certain path $(\alpha_i, \alpha_k, ..., \alpha_r, \alpha_s)$ in the outcome graph an $\overline{\alpha}$-*simple path* if source and terminal nodes α_i, $\alpha_s \in n_0 \cup \{n^*\} \cup \overline{N}$ and all other nodes $\alpha_k, ..., \alpha_r \in \overline{\overline{N}}$. For example, in Fig. 8.1, path (1, 3, 5, 7) is an $\overline{\alpha}$-simple path since $\alpha_1, \alpha_7 \in \overline{N}$ and $\alpha_3, \alpha_5 \in \overline{\overline{N}}$. Direction h_{ik} of the first arc of the $\overline{\alpha}$-simple path is called the *direction of the path*.

Definition 2. Node α_j will be called $\overline{\alpha}$-achievable from node α_i if there is an $\overline{\alpha}$-simple path leading from α_i to α_j.

Definition 3. Call an $\overline{\alpha}$-frame of outcome graph $G^*(N^*, A^*)$ a directed graph $G^{**}(N^{**}, A^{**})$ for which the following conditions hold:

1) $N^{**} = \overline{N} \cup n_0 \cup \{n^*\}$, i.e., the set of nodes of the $\overline{\alpha}$-frame does not include random branching nodes.
2) Each arc of the $\overline{\alpha}$-frame is determined by a quadruple $(\alpha_i, h_{ik}, \alpha_j, p_{ijk})$ and is included in G^{**} if and only if there exists an $\overline{\alpha}$-simple path leading from α_i to α_j in direction h_{ik}.

Value p_{ijk} is the probability for an $\overline{\alpha}$-simple path $(\alpha_i, ..., \alpha_j)$ with direction h_{ik} to be realized. Note that since there can be more than one $\overline{\alpha}$-simple path between two terminal nodes, G^{**} can be a multigraph. For example, the $\overline{\alpha}$-frame of the outcome graph in Fig. 8.1 is a multigraph since there are two $\overline{\alpha}$-simple paths between nodes 2 and 7, namely $(2, h_{24} = 1, 7, 0.2)$ and $(2, h_{25} = 2, 7, 0.5)$, two $\overline{\alpha}$-simple paths between nodes 1 and 16, etc.

From the definition of an $\overline{\alpha}$-frame, it can be well-recognized that there exists one for any outcome graph and is unique.

8.3.2 The algorithm

The algorithm described in [51,57] determines the $\overline{\alpha}$-frame on the basis of the outcome graph. Let outcome graph G^* have a set of arcs (i, j) with data information from Array I. In the algorithm process, all the $\overline{\alpha}$-nodes together with sink nodes are sorted and each node α_j of type $\overline{\alpha}$ is followed in the opposite direction of all the paths entering it, until the appearance of $\overline{\alpha}$-node α_i, or the source node n_0. Obviously, any path will sooner or later lead to one of the nodes of such a type. This path is made to correspond with set $(\alpha_i, h_{ik}, \alpha_j)$, where α_i and α_j are, respectively, the source and the terminal nodes of the $\overline{\alpha}$-simple path obtained, and h_{ik} is the direction of the path in node α_i. Thus, some set $(\alpha_i, h_{ik}, \alpha_j)$ will correspond to any node α_j which is $\overline{\alpha}$-achievable from an $\overline{\alpha}$-node α_i or from the source node n_0.

By this process, select all sets $(\alpha_i, h_{ik}, \alpha_j)$ corresponding to $\overline{\alpha}$-simple paths leading from α_i to α_j. In the process of finding each path of this kind, calculate its probability.

We then obtain the set of quadruples $(\alpha_i, h_{ik}, \alpha_j, p_{ijk})$, some of them possibly being quadruples with similar $\alpha_i, h_{ik}, \alpha_j$, corresponding to different independent $\overline{\alpha}$-simple paths from α_i to α_j in direction h_{ik}. For example, in Fig. 8.1, there are two $\overline{\alpha}$-simple paths leading from 1 to 8 in one and the same direction $h_{13} = 3$, namely, $(1, h_{13} = 3, 8, 0.05)$ via nodes 3 and 5, and $(1, h_{13} = 3, 8, 0.40)$ via node 3. Each set of quadruples with similar $\alpha_i, h_{ik}, \alpha_j$ is replaced by a single quadruple $(\alpha_i, h_{ik}, \alpha_j, \tilde{p}_{ijk})$, where $\alpha_i, h_{ik}, \alpha_j$ are the values common to all quadruples in the set, and $\tilde{p}_{ijk}$ is the probability of achieving node α_j from α_i in direction h_{ik}. When calculating $\tilde{p}_{ijk}$, it should be taken into account that the outcome graph examined is fully divisible. Various paths in the outcome graph are therefore independent, and the probability $\tilde{p}_{ijk}$ of reaching α_j from α_i in direction h_{ik} equals the sum of probabilities p_{ijk} of all paths $(\alpha_i, h_{ik}, \alpha_j, p_{ijk})$.

As a result of such a transformation, we obtain the $\bar{\alpha}$-frame desired, in the form of a set of different quadruples $(\alpha_i, h_{ik}, \alpha_j, \tilde{p}_{ijk})$.

The $\bar{\alpha}$-frame for the CAAN type outcome graph in Fig. 8.1 (Array II) is given in Tab. 8.1.

Table 8.1. The $\bar{\alpha}$-frame (Array II)

i	h_{ik}	j	$\tilde{p}_{ijk}$
1	1	2	1
1	2	16	0.4
1	2	17	0.6
1	3	7	0.05
1	3	8	0.45
1	3	16	0.32
1	3	17	0.18
2	1	6	0.8
2	1	7	0.2
2	2	7	0.5
2	2	8	0.5
6	1	10	1
6	1	11	1
7	1	12	1
7	2	13	1
8	1	14	1
8	2	15	1

§8.4 Algorithm II for determining maximal paths

8.4.1 Maximal path

Each arc of the $\bar{\alpha}$-frame is identified by a triad $(\alpha_i, h_{ik}, \alpha_j)$. Triads corresponding to different arcs must differ in at least one element. Triads with only the second element differing correspond to arcs connecting one and the same pair of nodes in different directions; triads with only the third element differing correspond to arcs connecting one and the same node in the same direction with differing nodes.

Definition 1. A sequence of arcs

$$(\alpha_1, h_{1k_1}, \alpha_2),\ (\alpha_2, h_{2k_2}, \alpha_3),\ \ldots\ ,\ (\alpha_r, h_{rk_r}, \alpha_{r+1}), \qquad (8.4.1)$$

in which the terminal node of one arc, excluding the last, is the start node of the next one, is called *a path* in the $\bar{\alpha}$-frame and is written:

$$(\alpha_1, h_{1k_1}, \alpha_2, h_{2k_2}, \alpha_3, \ldots, \alpha_r, h_{rk_r}, \alpha_{r+1}). \qquad (8.4.2)$$

Definition 2. A path in the $\bar{\alpha}$-frame will be called *maximal* if it does not belong to any other path. Obviously, any maximal path leads from the source node

n_0 to some sink node n^*.

8.4.2 Lexicographical order

Let all nodes in the $\bar{\alpha}$-frame be enumerated with different numbers, for example, $\alpha_1, \alpha_2, \ldots$ are the node numbers required. Also determine all the different directions leading from every node α_i. Consider two different maximal paths

$X_1 = (\alpha_a, h_{ab}, \alpha_c, \ldots, \alpha_e)$ and $X_1 = (\alpha_m, h_{mn}, \alpha_p, \ldots, \alpha_z)$.

Compare by pair the elements of these paths: α_a and α_m, h_{ab} and h_{mn}, α_c and α_p, etc. Since the paths are different and maximal, a pair of differing elements α_f and α_r (or h_{fg} and h_{rs}) must be found while all the other previous pairs coincide. If in this case $\alpha_f < \alpha_r$ (or $h_{fg} < h_{rs}$), the first lexicographically ordered path precedes the second. Thus, the first path precedes the second if its sequence lexicographically precedes the sequence of the second path. Similarly, a lexicographical order is introduced in the set of all $(\alpha_i, h_{ik}, \alpha_j)$-type arcs.

8.4.3 The algorithm

The algorithm described in [57] determines all the maximal paths in lexicographical order. It consists of two main parts: the procedure for choosing the first maximal path and the procedure for transferring from one arbitrary maximal path to the next one in lexicographical order. By using the first procedure and then, repeatedly, the second one, we obtain all the maximal paths in the $\bar{\alpha}$-frame.

The set of maximal paths (Array III) for the outcome graph in Fig. 8.1 is presented in Tab. 8.2 (the number in braces denotes the arc's direction).

Table 8.2. Maximal paths (Array III)

$X_1 = (1, \{1\}, 2, \{1\}, 6, \{1\}, 10)$	$X_9 = (1, \{2\}, 16)$
$X_2 = (1, \{1\}, 2, \{1\}, 6, \{2\}, 11)$	$X_{10} = (1, \{2\}, 17)$
$X_3 = (1, \{1\}, 2, \{1\}, 7, \{1\}, 12)$	$X_{11} = (1, \{3\}, 7, \{1\}, 12)$
$X_4 = (1, \{1\}, 2, \{1\}, 7, \{2\}, 13)$	$X_{12} = (1, \{3\}, 7, \{2\}, 13)$
$X_5 = (1, \{1\}, 2, \{2\}, 7, \{1\}, 12)$	$X_{13} = (1, \{3\}, 8, \{1\}, 14)$
$X_6 = (1, \{1\}, 2, \{2\}, 7, \{2\}, 13)$	$X_{14} = (1, \{3\}, 8, \{2\}, 15)$
$X_7 = (1, \{1\}, 2, \{2\}, 8, \{1\}, 14)$	$X_{15} = (1, \{3\}, 16)$
$X_8 = (1, \{1\}, 2, \{2\}, 8, \{2\}, 15)$	$X_{16} = (1, \{3\}, 17)$

§8.5 Algorithm III for determining admissible plans and joint variants

The algorithm includes, in turn, three sequentially used subalgorithms. Subalgorithm IIIA transforms the information obtained from Algorithm II since the latter may provide redundant data. Subalgorithm IIIB determines consecutively all the admissible plans while Subalgorithm IIIC singles out the corresponding joint variants.

8.5.1 *Subalgorithm IIIA to obtain auxiliary Array IV*

Subalgorithm IIIA transforms Array III to an auxiliary Array IV by the following procedure:

Step 1. Remove the corresponding sink nodes from all the maximal paths and thus form the set of "truncated" paths.

Step 2. Remove all the truncated paths which are parts of other truncated paths (for example, paths X_{15} or X_{16} in Tab. 8.2 after removing the sink nodes are parts of path X_{14}).

Step 3. If some of the truncated paths coincide after Step 1 leave only one of them (paths X_9-X_{10} in Tab. 8.2).

Array IV obtained after transforming Array III in Tab. 8.2 is presented in Tab. 8.3.

Table 8.3. Array IV for determining admissible plans

1, {1}, 2, {1}, 6, {1}
1, {1}, 2, {1}, 6, {2}
1, {1}, 2, {1}, 7, {1}
1, {1}, 2, {1}, 7, {2}
1, {1}, 2, {2}, 7, {1}
1, {1}, 2, {2}, 7, {2}
1, {1}, 2, {2}, 8, {1}
1, {1}, 2, {2}, 8, {2}
1, {2}
1, {3}, 7, {1}
1, {3}, 7, {2}
1, {3}, 8, {1}
1, {3}, 8, {2}

8.5.2 *Subalgorithm IIIB to determine admissible plans*

Definition. Two different truncated maximal paths

$\bar{\alpha}_{i_1}, h_{i_1p_1}, \bar{\alpha}_{i_2}, h_{i_2p_2}, ..., \bar{\alpha}_{i_r}, h_{i_rp_r}$,

$\bar{\alpha}_{j_1}, h_{j_1q_1}, \bar{\alpha}_{j_2}, h_{i_2q_2}, ..., \bar{\alpha}_{j_s}, h_{j_sq_s}$

are called *contradictory* ones if they each possess at least one common alternative node α_f of $\bar{\alpha}$-type with mutually exclusive alternative directions h_{fk} and h_{fq}.

The work of Subalgorithm IIIB is based on lexicographical scanning and resembles Algorithm II (see §8.4). It consists, like the latter one, of two parts: the procedure for determining the first admissible plan, and the procedure for transferring from a routine admissible plan to the next one.

Determine the first admissible plan as follows: choose the first truncated maximal path from Array IV (maximal paths being initially ordered lexicographically). This is the basis for determining the first admissible plan. Then examine the next maximal path and determine, if it is contradictory to the basic admissible plan or not. If so, examine the next routine maximal path. If not, add to the admissible plan all the links (either of type $[\alpha_r, h_{rk}, \alpha_{r+1}]$ or of type $[\alpha_r, h_{rk}]$) from the maximal path under examination which are absent in the basic admissible plan. For example, in Tab. 8.3, the first truncated maximal plan is:

1, $h_{12} = 1$, 2, $h_{24} = 1$, 6, $h_{6,10} = 1$.

The second one is contradictory to the first, but from the third non-contradictory maximal path we add the link $[7, h_{7,12} = 1]$, thus enlarging the admissible plan to

1, $h_{12} = 1$, 2, $h_{24} = 1$, 6, $h_{6,10} = 1$, 7, $h_{7,12} = 1$.

The procedure thus boils down to scanning the maximal paths in Array IV, each time comparing the next maximal path with the "growing" admissible plan already obtained.

Now let $W = [\bar{\alpha}_{i_1}, h_{i_1 q_1}, \bar{\alpha}_{i_2}, h_{i_2 q_2}, ..., \bar{\alpha}_{i_r}, h_{i_r q_r}]$ be an arbitrary admissible plan. The procedure for determining the next one is as follows: exclude the last link $[\bar{\alpha}_{i_r}, h_{i_r q_r}]$ and find out whether it is possible to determine a new admissible plan (which does not coincide with those obtained before) while applying the first of the algorithms. If there is no such admissible plan, exclude the link $[\bar{\alpha}_{i_{r-1}}, h_{i_{r-1} q_{r-1}}]$, and again apply the procedure of determining an admissible plan, and so on. Stop working the algorithm, when the consequently truncated admissible plan becomes empty.

Admissible plans for the outcome graph in Fig. 8.1 (Array V) are presented in Tab. 8.4.

Table 8.4. The set of admissible plans (Array V)

$W_1 = 1,\{1\},2,\{1\},6,\{1\},7,\{1\}$	
$W_2 = 1,\{1\},2,\{1\},6,\{1\},7,\{2\}$	$W_8 = 1,\{1\},2,\{2\},7,\{2\},8,\{2\}$
$W_3 = 1,\{1\},2,\{1\},6,\{2\},7,\{1\}$	$W_9 = 1,\{2\}$
$W_4 = 1,\{1\},2,\{1\},6,\{2\},7,\{2\}$	$W_{10} = 1,\{3\},7,\{1\},8,\{1\}$
$W_5 = 1,\{1\},2,\{2\},7,\{1\},8,\{1\}$	$W_{11} = 1,\{3\},7,\{1\},8,\{2\}$
$W_6 = 1,\{1\},2,\{2\},7,\{1\},8,\{2\}$	$W_{12} = 1,\{3\},7,\{2\},8,\{1\}$
$W_7 = 1,\{1\},2,\{2\},7,\{2\},8,\{1\}$	$W_{13} = 1,\{3\},7,\{2\},8,\{2\}$

8.5.3 Subalgorithm IIIC to determine joint variants

The initial information for the subalgorithm is the set of admissible plans (Array V, see Tab. 8.4) and the outcome graph (see Fig. 8.1). Note that a joint variant is a subgraph of the outcome graph containing all the full variants obtained through a definite selection of directions in the $\bar{\alpha}$-nodes. Therefore, the joint variant, corresponding to a routine admissible plan W, should be a stochastic network s, satisfying the following conditions:

1) s is a subgraph of the outcome graph;

2) s contains all the nodes of set W, but no $\bar{\alpha}$-nodes besides them;

3) Only sink nodes of the outcome graph can serve as sink nodes of s;

4) Any subgraph of the outcome graph satisfying properties 1)-3) is a subgraph of joint variant s (i.e., s is the maximal subgraph satisfying those properties).

Using these properties, Subalgorithm IIIC to determine the joint variant for the routine admissible plan, W, was presented in [53,57]. The idea of the subalgorithm is as follows: for every $\bar{\alpha}_i \in W$, a subgraph of the outcome graph is determined containing all the maximal $\bar{\alpha}$-simple paths leaving α_i in direction $h_{ik} \in W$. Obviously, a combination of all such subgraphs contains s within itself. To determine s, it is now sufficient to cast off part of the arcs in such a way that any maximal path in the remaining subgraph will finish with a node from W or with a sink node n^*.

§8.6 Numerical example

The management is faced with development an R&D project represented by a CAAN type network, its outcome graph given in Fig. 8.1. The time duration and cost values (measured in months and dollars, respectively) of each activity are presented in Tab. 8.5. The expected duration of the project has to be minimized subject to the cost restriction: the mean cost of the project must not exceed \$23,000. The management has to determine an optimal decision policy, i.e., to choose optimal outcome directions from every decision-making node which is reached in the course of the project's development. This is based on singling out all the joint variants of the CAAN network under examination, calculating the mean time and cost values of each joint variant and finally determining the optimal one. Decision-making at node 1 has to be carried out before starting the project's implementation.

Table 8.5. The initial data

Activity (i,j)	Activity duration (in months) t_{ij}	Activity cost (in 1,000 \$) c_{ij}	Activity (i,j)	Activity duration (in months) t_{ij}	Activity cost (in 1,000 \$) c_{ij}
(1,2)	1	5	(5,7)	1	4
(1,3)	1	6	(5,8)	2	10
(1,9)	4	16	(6,10)	2	9
(2,4)	2	7	(6,11)	2	7
(2,5)	1	6	(7,12)	1	5
(3,5)	1	7	(7,13)	2	8
(3,8)	2	8	(8,14)	2	7
(3,9)	2	10	(8,15)	3	12
(3,16)	4	15	(9,16)	3	11
(4,6)	1	6	(9,17)	2	5
(4,7)	1	8			

The results of sequential use of Algorithms I, II and Subalgorithms IIIA and IIIB are presented in Tab. 8.1-8.4, respectively. Applying Subalgorithm IIIC to the data in Tab. 8.4, the management determines all the joint variants s (the joint variant for the first admissible plan is demonstrated in Fig. 8.2). Afterwards their average time durations $\overline{T}_s$ and cost values $\overline{C}_s$ are calculated, the results given in Tab. 8.6. According to the cost restriction the optimal joint variant (see Fig. 8.4) has the expected time duration of 5.12 months and the average cost of $22,220.

Table 8.6. The parameters of the joint variants

The joint variant s	The expected time duration (in months) $\overline{T}_s$	The expected cost (in 1,000 $) $\overline{C}_s$	Feasibility
1	5.8	26.6	No
2	6	27.2	No
3	5.8	25	No
4	6	25.6	No
5	5	24	No
6	5.5	26.5	No
7	5.5	25.5	No
8	6	28	No
9	6.4	23.4	No
10	5.12	22.22	Yes*)
11	5.57	24.47	No
12	5.17	22.37	Yes
13	5.62	24.62	No

*) - optimal joint variant

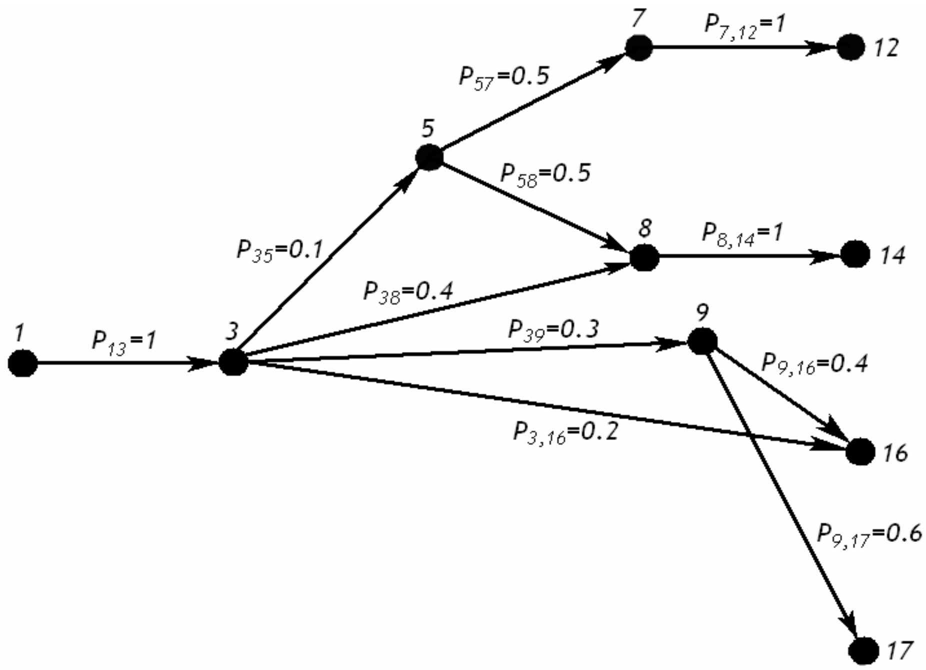

Figure 8.4. The optimal joint variant

Thus the management has to introduce optimal decision-making as follows:

a) At the beginning of the project (at node 1) activity (1,3) has to be chosen.

b) If in the course of the project's development node 7 will be reached and the network will not be revised, activity (7,12) has to be chosen.

c) If node 8 will be reached, activity (8,14) has to be chosen. Nodes 3, 5 and 9 of the joint variant are uncontrollable.

Note that calculating average values ($\overline{T}_s$ and $\overline{C}_s$) in the optimization problem can be easily replaced by calculating the p-quantiles at a given confidence level $1-p$ in order to raise the project's reliability. These estimates can be obtained both for deterministic and probabilistic time durations t_{ij} and cost values c_{ij}. In the latter case Monte Carlo simulation technique has to be implemented [43].

§8.7 Conclusions

In our opinion, CAAN type models are typical innovative models which can be used in several areas, namely:

1. In large and highly complex R&D projects with long-term goals, especially when an entirely new device is designed with no similar prototypes in the past. Such projects occur often in aerospace and other defense related industries. They are usually faced with a great deal of uncertainty in their progress as well as with alternative outcome directions in key events.
2. Long-term projects in construction industry, when creating and building unique installations (various defense systems, undersea tunnels, major pipelines, etc.).
3. Long-term projects when designing or developing new industrial or populated areas. In the activity network the branching nodes of $\overline{\overline{\alpha}}$-type may reflect the alternative results of future geological survey or the influence of climatic factors, while the $\overline{\alpha}$-type nodes may reflect alternative decision-making as follows: to build or not to build a new plant in a certain place, to build a railroad or a motor road between two settlements, etc. The objective function for this type of projects reflects usually capital investments to be minimized.
4. For the case of deterministic alternative networks we recommend using the results obtained by E. Dinic [34] in combination with simulation approaches.
5. As outlined in the previous Chapter, CAAN models cannot be used for optimizing non-divisible alternative networks which cover a variety of exceptionally complicated R&D projects. The latter have to be controlled by a more general model - namely, the GAAN model, which will be outlined in Chapter 9.

Chapter 9. Generalized Alternative Activity Network (GAAN Model)

§9.1 Formal description of GAAN model

As mentioned above, in §1.1, the GAAN model is a finite, oriented, acyclic activity-on-arc network $G(N,A)$ with the following properties [67]:

I. $G(N,A)$ has one source node n_0 and no less than two sink nodes n'.

II. Each activity $(i,j)\in A$ refers to one of the following three different types:

Type 1: activity (i,j) is a PERT activity (PA) with the logical "must follow" emitter in node i and the "and" receiver in node j;

Type 2: activity (i,j) is an alternative stochastic activity (ASA) with the logical "exclusive or" emitter in node i. Each $(i,j)\in A$ of ASA type corresponds to a probability $0<p_{ij}<1$, while node i comprises a set of at least two probabilities p_{ij}, $\sum_j p_{ij}=1$;

Type 3: activity (i,j) is an alternative deterministic activity (ADA) with the logical "exclusive or" emitter in node i. Node i is a decision-making node, and the sum of the corresponding transfer probabilities (at least two of them) is assumed to be unity.

III. Activities of all types may come out of the same node $i\in N$. Thus, unlike the CAAN model, the GAAN model is not a fully-divisible network.

IV. Activities of all types may enter the same node.

An example of GAAN type graph is shown in Fig. 9.1. Here, activities $(1,2)$ and $(3,4)$, $(2,7)$ and $(2,8)$, $(4,9)$ and $(4,10)$ are of ADA type. Activities $(1,4)$ and $(1,5)$, $(3,7)$ and $(3,8)$ are of ASA type, while activities $(1,9)$, $(2,6)$, $(3,9)$, $(5,10)$, $(5,11)$ are of PA type. Note that such a network is a more universal model than the Eisner model, which comprises only activities of Types 1 and 2.

Definitions

Following [67], introduce the concept of a joint variant for a GAAN model. Call *a joint variant of the GAAN model* $G(N,A)$ a subgraph (subnetwork) $G^*(N^*,A^*)$ satisfying the following conditions:

1. $G^*(N^*,A^*)$ has one source node coincident with that of graph $G(N,A)$.
2. If $G^*(N^*,A^*)$ comprises a certain node i, i.e., $i\in N^*$, then $G^*(N^*,A^*)$ comprises all activities (i,j) of types PA and ASA leaving node i.
3. If $G^*(N^*,A^*)$ comprises a certain node i having alternative outcomes of ADA type in the GAAN model $G(N,A)$, then $G^*(N^*,A^*)$ comprises only one activity of this type leaving that node.

Call *a full variant of joint variant* $G^*(N^*,A^*)$ a subnetwork of PERT type $G^{**}(N^{**},A^{**})\subset G^*(N^*,A^*)$ which can be extracted from the latter by simulating non-contradictory outcomes of ASA type in interconnected nodes and excluding alternative non-simulated outcomes.

Call *a full variant* G^{**} *realization probability* the product of all values p_{ij} for all activities of ASA type entering the full variant.

We shall show that for any joint variant $G^* \subset G$ the sum of full variant realization probabilities over all full variants entering G^* is equal to unity.

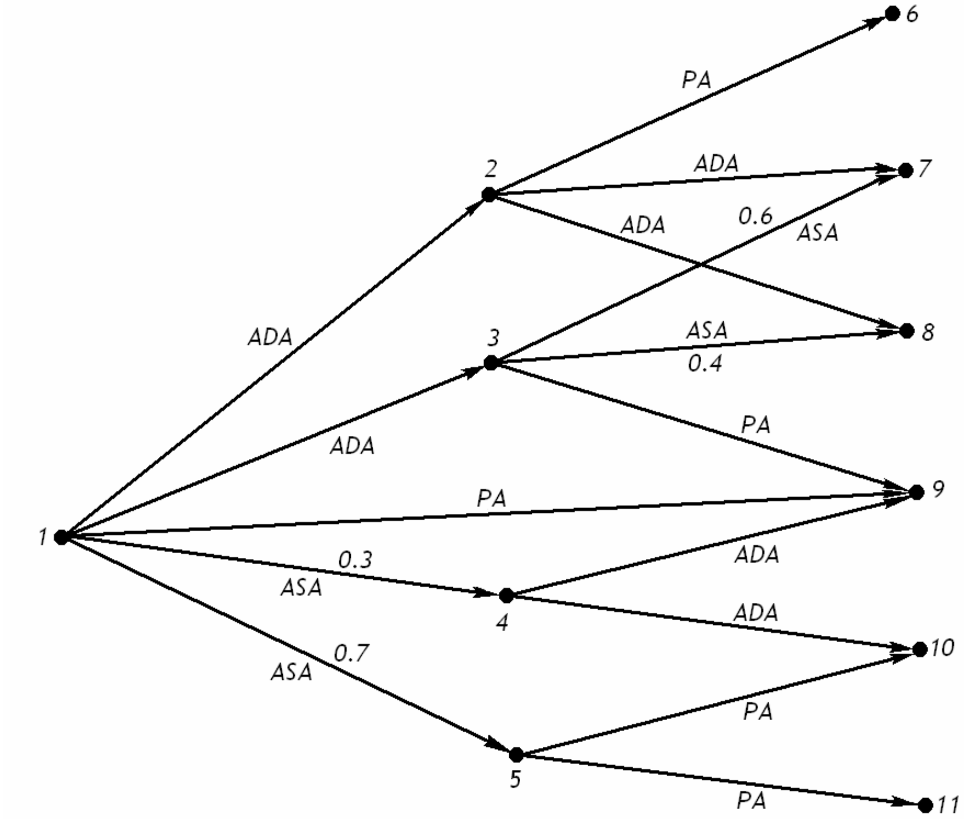

Figure 9.1. The GAAN type graph

Lemma 9.1. The sum of probabilities of realization of all full variants entering a joint variant equals unity.

Proof. First, demonstrate that in the GAAN network, there does exist a node which is connected by arcs leaving it only with sinks.

Consider a set of network paths from the source to all sink nodes. The number of arcs entering a path will be called the path length. Since the number of paths is finite, a path of maximum length does exist. Consider the last but one node of this path, i.e., the node connected to the sink by the arc leaving this node. All the arcs incident to this node enter sink nodes. Actually, if an arc entering an internal node exists, then the path under discussion is not a path of maximal length as agreed.

We shall prove the lemma by induction over the number of network nodes, assuming, without loss of generality, that only one arc enters any sink node.

Also without loss of proof generality, assume that arcs of only one type leave each node.

The lemma obviously holds for $k = 3$. Suppose that it is true for all $k \le n$, and consider a network with $(n+1)$ nodes. The network contains an internal node x

such that for any arc (x,y), y is a sink.

Consider the following three cases.

1. Node x is of PA type (PA-node).

From the initial network, remove the arcs leaving node x, and the sinks connected with these arcs. We obtain a network with the number of nodes less or equal to n. For this network, according to the assumption, the lemma holds.

All the joint variants of the initial network are obtained from the variants of the "smaller" network by completing those containing the node x with the arcs leaving this node, and the corresponding sinks. However, arcs leaving the PA-nodes and entering the sinks cannot change the corresponding full variant realization probabilities. Therefore, in this case, the lemma asserts.

2. Node x is of ADA type (ADA-node).

This case is similar to the previous one.

3. Node x is of ASA type (ASA-node).

Arcs $(x,y_1),(x,y_2),\ldots,(x,y_r)$ leaving node x have probabilities $\alpha_1,\alpha_2,\ldots,\alpha_r$, $\sum_{i=1}^{r}\alpha_i = 1$.

Consider any joint variant comprising node x. From this joint variant, remove the arcs leaving node x, and the sinks connected with these arcs. We obtain a joint variant with the number of nodes less or equal to n. Let the full variants of this "smaller" joint variant have probabilities $\beta_1,\beta_2,\ldots,\beta_s$, and the first ℓ full variants comprise node x. By assumption, $\sum_{s=1}^{r}\beta_s = 1$. The full variants of the "larger" joint variant have probabilities $\beta_1\alpha_1,\beta_1\alpha_2,\ldots,\beta_1\alpha_r,\beta_2\alpha_1,\beta_2\alpha_2,\ldots,\beta_2\alpha_r,\ldots,\beta_\ell\alpha_1,\beta_\ell\alpha_2,\ldots,\beta_\ell\alpha_r,\beta_{\ell+1},\beta_{\ell+2},\ldots,\beta_s$. Further, we obtain $\sum_{j=1}^{\ell}\sum_{i=1}^{r}\beta_j\alpha_i + \sum_{j=\ell+1}^{s}\beta_j = \sum_{j=1}^{\ell}\beta_j\sum_{i=1}^{r}\alpha_i + \sum_{j=\ell+1}^{s}\beta_j = \sum_{j=1}^{\ell}\beta_j + \sum_{j=\ell+1}^{s}\beta_j = 1$, and, hence, the lemma is true.

For joint variants not containing x the lemma is also true. Actually, these joint variants may be considered as those of the network with the number of nodes less or equal to n.

The lemma is proven. ■

§9.2 Optimization problem on GAAN

9.2.1 Mathematical formulation

The mathematical formulation of the problem is as follows: determine the optimal joint variant $G^{*opt} \subset G(N,A)$ that optimizes the objective function

$$E\left[F\left(G^{*opt}\right)\right] = \underset{\{G^*\}}{Min}(Max) \sum_{\{G^{**}\}\subset G^*}\left[F\left(G^{**}\right)\cdot \Pr\left\{G^{**}\right\}\right] \qquad (9.2.1)$$

subject to

$$E\left[Q_v\left(G^{*opt}\right)\right] = \sum_{\{G^{**}\}\subset G^*}\left[Q_v\left(G^{**}\right)\cdot \Pr\left\{G^{**}\right\}\right] \le H_v,\ \ 1 \le v \le w. \qquad (9.2.2)$$

Here, $F(G^{**})$ is the objective function of full variant G^{**}, $\Pr\{G^{**}\}$ is the G^{**} realization probability, $Q_v(G^{**})$ is the v-th constraint criterion, and H_v is the preset constraint level for that criterion. Note that for certain particular cases, the value of w may be zero, i.e., the optimization problem is unconstrained, or the problem comprises only one constraint (9.2.2) without objective function (9.2.1). When F refers to the project's duration, the first constraint H_1 is usually the project's cost, and *vice versa*.

9.2.2 NP-completeness

Show that problem (9.2.1-9.2.2) is NP-complete. Consider a particular case of the GAAN model, in which each activity $(i, j) \subset G(N, A)$ is either a PERT activity (PA) or an alternative deterministic activity (ADA). Only one parameter, namely, the cost c_{ij}, is assigned to each activity (i, j). The values of c_{ij} are fixed and pregiven. Each joint variant G^* is characterized by its cost value $C\{G^*\}$ which is equal to the sum of the cost values of all activities entering this joint variant.

The problem called "minimum weight AND/OR graph solution" [47] boils down to determining the joint variant with the minimal cost value. Sahni [140] proved that this problem is NP-complete. Since the problem is unconstrained and is applied to a particular case of the GAAN model, it can be regarded as a particular case of the general problem (9.2.1-9.2.2). If a particular problem is NP-complete, then the general problem is also NP-complete. *Thus, to obtain a precise solution, one has to develop a lookover algorithm to single out all the joint variants*. Note the techniques for the fully divisible CAAN model [57] cannot be applied straightforwardly to the GAAN network.

§9.3 The general approach to the optimization problem's solution

The idea to enumerate the joint variants of the CAAN model [57] is based on introducing lexicographical order to the set of maximal paths in the CAAN graph. In the case of GAAN network the order on the set of paths has to be substituted for the order on the set of subgraphs. To develop the enumeration algorithm, one may use the ideas of enumerating the so-called trajectories for assignment problems, or special matrices for traveling salesman problems [10-11]. Note that singling out the maximal trajectory for an assignment problem is similar to determining the joint variant with the maximal objective value. Since a trajectory can be regarded as a vector and the latter, in turn, can be mapped onto a set of integer numbers, the trajectories can be enumerated. Similar ideas may be implemented in analyzing a GAAN network in order to enumerate and single out all the joint variants.

To implement algorithms for the GAAN model analysis, one has to carry out consecutively the following three procedures:

Procedure 1

Modify the GAAN model so that each node (except the sink nodes) would be the source of only one type of activity; i.e., only activities of either PA type, or ASA type, or ADA type have to leave the node.

Procedure 2

Modify the GAAN model to satisfy the following conditions:

1. For each activity (i,j), the indices satisfy $i < j$.
2. Any sink node number is greater than that of any internal node, i.e., the one that is not a sink node.

After implementing Procedures 1 and 2 for the graph represented in Fig. 9.1, we obtain the graph shown in Fig. 9.2. The modified graph comprises $n = 12$ internal nodes.

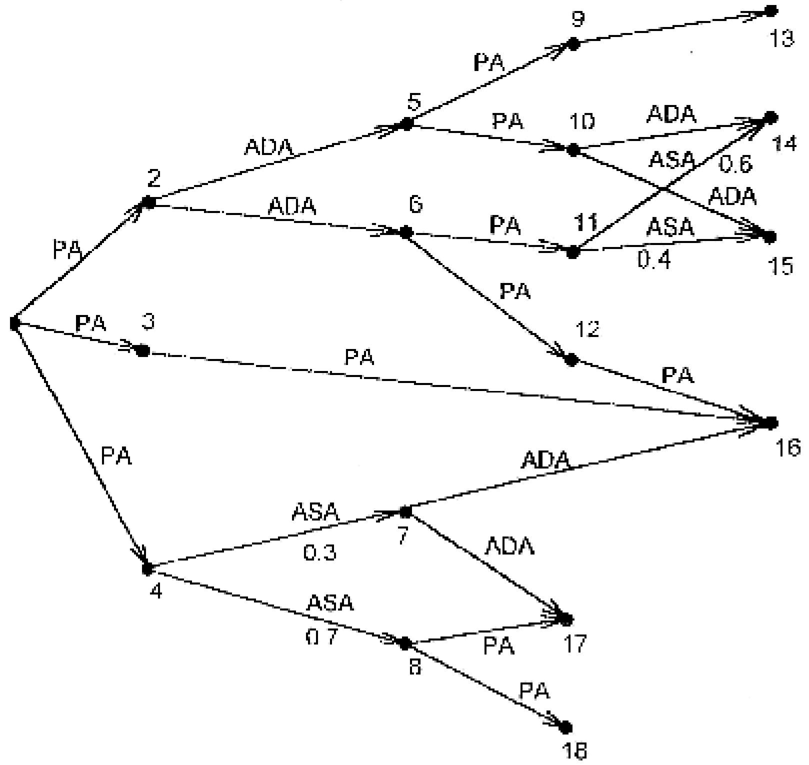

Figure 9.2. The GAAN graph after implementing Procedures 1 and 2

Procedure 3

Examine all the nodes with alternative outcomes of ADA and ASA types and enumerate separately, for each of these nodes, all the activities leaving the node. If m_i alternative outcomes $(i, j_1),(i, j_2),..., (i, j_{m_i})$ leave alternative node i, each activity receives a different ordinal number from 1 to m_i. The assignment idea is as follows: in case $j_1 > j_2$ activity (i,j_1) receives a number greater than (i,j_2). Thus, we have to set values $j_1, j_2,..., j_{m_i}$ in ascending order and assign values $1,2,...,m_i$ to the corresponding activities. Let k_{ij} be the number of alternative activities (i,j), $1 \le k_{ij} \le m_i$.

The input information on each activity $(i,j) \subset A$ entering the GAAN model $G(N,A)$ (after implementing Procedures 1 to 3) comprises the following records:

i	j	p_{ij}	ADA	PA	F_{ij}	$Q_v(i,j)$, $1 \le v \le w$	k_{ij}
Starting node	Terminal node	Outcome probability (ASA type)	Set 1 if (i,j) is of ADA type	Set 1 if (i,j) is of PERT type	(i,j) objective function value	(i,j) constraints	No. of (i,j) activities for ADA or ASA types

Consider a GAAN type graph $G(N,A)$ after the implementation of Procedures 1 and 2. Let M be the number exceeding by 1 the maximal number of *alternative* outcomes (ASA or ADA types) leaving a node entering the graph. Let n be the number of internal nodes (see Procedure 2). Consider the mapping of the set of joint variants $G^*(N^*,A^*) \subset G(N,A)$, onto the set of n-dimensional vectors with coordinates $(f_1, f_2, ..., f_n)$, where f_i, $1 \le i \le n$, $0 \le f_i \le M$, are integers. Each coordinate f_i corresponds to node i. The correspondence rules are as follows:

1. $f_i = 0$, if and only if node i does not enter the joint variant.
2. $f_i = M$, if node i enters the joint variant and is the source of either ASA or PA outcome activities.
3. $f_i = k_{ij}$, if node i enters the joint variant that comprises an alternative activity (i,j) of ADA type leaving that node. Note that only one activity of ADA type may leave a node in a joint variant (stems from the definition of a joint variant).

In order for all the joint variants to be singled out, they must be enumerated. It can be well-recognized that two different joint variants correspond to different vectors. The number of the joint variant is, essentially, the number of the corresponding vector. Thus, to enumerate the joint variants, their corresponding vectors have to be arranged in a certain order. We shall use the lexicographical order as follows:

If two joint variants $G_1^*(N^*,A^*)$, $G_2^*(N^*,A^*)$ are mapped onto vectors $\vec{f}_1 \equiv (f_1^{(1)}, f_2^{(1)}, ..., f_n^{(1)})$, $\vec{f}_2 \equiv (f_1^{(2)}, f_2^{(2)}, ..., f_n^{(2)})$, correspondingly, then G_1^* precedes G_2^* if the first s coordinates, $0 \le s \le n-1$, in both vectors coincide, but the $(s+1)$-th coordinate satisfies $f_{s+1}^{(1)} < f_{s+1}^{(2)}$. Such an order enables the enumeration of all the joint variants to be done.

§9.4 Algorithms for enumerating and determining the joint variants

The procedure for enumerating and determining the joint variants consists of the following algorithms [9,67]:

Algorithm I. *Determination of the Minimal Number Vector from the GAAN Model $G(N,A)$.*

Specify the algorithm for constructing the vector as follows:

<u>Step 1</u>. Consider the node with the minimal number 1 entering the graph

$G(N,A)$. If this node has outcomes of PA or ASA types, set $f_1 = M$. In the case of ADA outcomes, set $f_1 = 1$. Set $j = 1$.

Step 2. Set $j = j + 1$.

Step 3. Examine all the activities entering node j. Let them be $(i_1, j),(i_2, j),...,(i_k, j)$. Two cases are considered:

3.1 Among nodes $i_1, i_2,.., i_k$ there is at least one node i_s, $1 \le s \le k$, with $f_{i_s} \ne 0$. If node j has outcomes of PA or ASA types, set $f_j = M$. If outcomes of ADA type leave node j, set $f_j = 1$; go to Step 4.

3.2 All the coordinates f_{i_s}, $1 \le s \le k$, are equal 0. Set $f_j = 0$.

Step 4. If $j < n$ return to Step 2.

Step 5. The algorithm terminates.

It can be proven that there is no vector that would correspond to a joint variant and lexicographically precede the vector constructed according to Algorithm I.

Lemma 9.2. The number obtained by implementing Algorithm I corresponds to the minimal joint variant, i.e., there is no joint variant with a number less than that obtained using Algorithm I.

Proof. Obviously, Algorithm I develops a joint variant. Let $f = (f_1, f_2,..., f_n)$ be the joint variant number obtained according to Algorithm I, and $h = (h_1, h_2,..., h_n)$ is an arbitrary joint variant of the network. Show that $f \le h$. To do this, demonstrate that if the first k coordinates of the vectors f and h are equal, then, for the $(k+1)$-th coordinate, $f_{k+1} \le h_{k+1}$.

Consider the first coordinate. If node 1 is of ASA- or PA-type, then $f_1 = M$, $h_1 = M$, and $f_1 \le h_1$. If node 1 is of ADA-type, then $f_1 = 1$, $h_1 \ge 1$. If $h_1 > 1$, Lemma 9.2 holds. Otherwise, i.e., if $h_1 = 1$, assume that the first k coordinates of vectors f and h are equal, and consider the $(k+1)$-th coordinate.

The following situations are possible.

1. Node $k+1$ does not belong to the joint variant f, i.e., $f_{k+1} = 0$. But in this case node $k+1$ does not belong to the joint variant h as well, since in these joint variants, the first k coordinates are equal; therefore, $h_{k+1} = 0$ and $f_{k+1} \le h_{k+1}$.
2. Node $k+1$ belongs to the joint variant f and is of PA- or ASA-type. Then, $f_{k+1} = M$. But node $k+1$ belongs to the joint variant h as well, and $h_{k+1} = M$, i.e., $f_{k+1} \le h_{k+1}$.
3. Node $k+1$ belongs to the joint variant f and is alternative. Then, according to Algorithm I, $f_{k+1} = 1$. But node $k+1$ belongs to the joint variant h as well, and $h_{k+1} \ge 1$. Hence, $f_{k+1} \le h_{k+1}$.

Thus, in all possible situations $f \le h$, i.e., the Lemma holds. ■

Algorithm II. *Determination of the Minimal Number Vector with $q < n$ Coordinates Given.*

Assume that there is a subset of vectors with known $q < n$ coordinates $f_1, f_2,..., f_q$. Let $q+1 = j$. Go to Step 3 of Algorithm I and proceed with that algorithm from node j until the end. Join the known coordinates $f_1, f_2,..., f_q$ together with $f_{j=q+1}, f_{j+1},..., f_n$ obtained using Algorithm I. Thus, implementation of Algorithm II results in creating vector $\vec{f} \equiv (f_1, f_2,..., f_q, f_{q+1},..., f_n)$. Note that the algorithm cannot be applied to the case of any arbitrary given coordinates $f_1, f_2,..., f_q$ but only when the latter entity actually belongs to a subset of corresponding vectors.

Lemma 9.3. Given q coordinates, the joint variant obtained using Algorithm II is the minimal one.

Proof. Since the first $j-1$ coordinates correspond to a joint variant, and Algorithm I provides coordinates that correspond to the joint variant, then the number obtained using Algorithm I is a joint variant.

Let $f = (f_1, f_2,..., f_{j-1}, f_j, f_{j+1},..., f_n)$ be the joint variant number obtained according to Algorithm II and $h = (h_1, h_2,..., h_{j-1}, h_j, h_{j+1},..., h_n)$ be the number (joint variant) with $h_1 = f_1,\ h_2 = f_2, \ldots,\ h_{j-1} = f_{j-1}$. Show that $f \le h$.

The first $j-1$ coordinates of vectors f and h are equal. Consider the j-th coordinate.

The following situations are possible.

1. Node j does not belong to the joint variant f, i.e., $f_j = 0$. But in this case node j does not belong to the joint variant h as well, since in these joint variants, the first $j-1$ coordinates are equal; therefore, $h_j = 0$ and $f_j \le h_j$.
2. Node j belongs to the joint variant f and is of PA- or ASA-type. Then, $f_j = M$. But node j belongs to the joint variant h as well, and $h_j = M$, i.e., $f_j \le h_j$.
3. Node j belongs to the joint variant f and is alternative. Then, $f_j = 1$. But node j belongs to the joint variant h as well, and $h_j \ge 1$. Hence, $f_j \le h_j$.

Thus, $f_j \le h_j$. If $f_j < h_j$, then $f < h$ and the Lemma is proved. If $f_j = h_j$, then $f_1 = h_1,\ f_2 = h_2, \ldots,\ f_{j-1} = h_{j-1},\ f_j = h_j$, and, according to the algorithm, $f_{j+1} \le h_{j+1}$. But this means that $f \le h$, and the Lemma is proved. ■

Algorithm III. *Determination of the Next Lexicographically Ordered Vector.*

Given the f-th lexicographically ordered vector $\vec{f} = (f_1, f_2,..., f_n)$, define the steps of the algorithm as follows:

Step 1.Set $j = n$.

Step 2.In cases $f_j = 0$, $f_j = M$ or $f_j = m_j$ (see Procedure 3) apply Step 4. Otherwise, proceed to the next step.

Step 3. Set $g_i = f_i$, $1 \le i \le j-1$, $g_j = m_j + 1$ and apply Algorithm II to obtain the minimal lexicographically ordered vector with given coordinates $g_1, g_2, .., g_j$. Go to Step 7.

Step 4. Set $j = j+1$.

Step 5. If $j \ge 1$, return to Step 2. Otherwise, proceed to the next step.

Step 6. Applying the step means that vector $\vec{f}$ is the maximal one. The algorithm terminates.

Step 7. Vector $g_1, g_2, ..., g_j, g_{j+1}, ..., g_n$ determined by implementing Algorithm II is the next lexicographically ordered vector. The algorithm terminates.

Given the f-th lexicographically ordered vector $\vec{f} = (f_1, f_2, ..., f_n)$, the algorithm determines a new vector $\vec{g} \ne \vec{f}$ adjacent to $\vec{f}$. It can be proven that there is no other vector $\vec{h} \ne \vec{f}$ that lexicographically exceeds vector $\vec{f}$ but precedes $\vec{g}$.

Lemma 9.4. Let f be the number of a joint variant and g be the number determined according to Algorithm III. Then, a joint variant corresponds to number g, and no joint variant does exist with a number h such that $f < h < g$.

Proof. Let $f = (f_1, f_2, ..., f_n)$ be a joint variant of the network and $g = (g_1, g_2, ..., g_n)$ be a joint variant determined according to Algorithm III.

If $f_n = m$, where $0 < m < M$ and m is not equal the maximal number of arc leaving n, then $g = (f_1, f_2, ..., f_n + 1)$, and, hence, the Lemma is true.

Let the last k coordinates of vector $\vec{f}$ be equal either 0, or M, or the maximal number of arc, while the $(n-k)$-th coordinate equals m, where $0 < m < M$ and m is not equal the maximal number of arc leaving node $n-k$.

In compliance with Algorithm III, the first $n-k-1$ coordinates of vectors $\vec{f}$ and $\vec{g}$ coincide, while $g_{n-k} = m+1$.

Consider an arbitrary variant $h = (h_1, h_2, ..., h_n)$ and show that either $h < f$, or $g < h$.

If any of the first $n-k-1$ coordinates of $\vec{h}$ does not coincide with the corresponding coordinates of vectors $\vec{g}$ and $\vec{f}$, then either $h < f$, or $h > g$.

Suppose that the first $n-k-1$ coordinates of vectors $\vec{h}$, $\vec{f}$, and $\vec{g}$ coincide. Then, $h_{n-k} = m$ or $h_{n-k} = m+1$. Otherwise, we would have either $h < f$, or $h > g$.

If $h_{n-k} = m$, i.e., $h_{n-k} = f_{n-k}$, then $h < f$, since vector $\vec{f}$ is the maximum over vectors whose first $n-k$ coordinates are equal $f_1, f_2, ..., f_{n-k}$, respectively. In fact, the last k non-vanishing coordinates take the maximal values.

If $h_{n-k} = m+1$, then $h_1 = g_1$, $h_2 = g_2$, …, $h_{n-k} = g_{n-k}$. But g is the minimal number within the first $n-k$ coordinates $g_1, g_2, .., g_{n-k}$ and, hence, $g < h$. ■

Therefore, either $h < f$, or $h > g$.

Algorithm IV. *Determination of a Joint Variant Corresponding to Vector* $\vec{f}$.

The initial information of this algorithm is the input information on graph $G(N, A)$ and vector $\vec{f}$. The step-by-step procedure of the algorithm is as follows:

Step 1. Set $i = 1$.

Step 2. Examine f_i. If $f_i = 0$, node i does not enter the joint variant; proceed to Step 5. Otherwise, apply the next step.

Step 3. If f_i satisfies $1 \le f_i = m < M$, node i enters the joint variant; select for the joint variant activity (i, j) satisfying $k_{ij} = m$; proceed to Step 5. Otherwise, apply the next step.

Step 4. If $f_i = 0$, node i enters the joint variant; select for the joint variant all the activities (i, j) leaving node i.

Step 5. Set $i = i + 1$.

Step 6. If $i < n$, return to Step 2. Otherwise, proceed to the next step.

Step 7. Activities being selected at Steps 3 and 4 form the corresponding joint variant G^*. The algorithm terminates.

Thus, all the joint variants entering the GAAN model $G(N, A)$ can be determined by repeatedly implementing Algorithms I, III (until the maximal vector $\vec{f}$ is obtained) and IV. Algorithm II is auxiliary.

Algorithm V. *Determination of Full Variants* $G^{**}(N^{**}, A^{**})$ *Entering a Routine Joint Variant* $G^*(N^*, A^*) \subset G(N, A)$.

After obtaining a routine joint variant $G^*(N^*, A^*)$ (see Algorithm IV) define the step-by-step procedure to determine its full variants:

Step 1. For certain activities $(i, j) \in G^*(N^*, A^*)$ change their type as follows:

1.1 Stochastic outcomes of ASA type - change their type to ADA: their probability outcomes p_{ij} are temporarily suspended and substituted by 1;

1.2 Alternative outcomes of ADA type (note that in a joint variant, no more than one activity of ADA type leaves a node) are amended to a new type that from this time on we will refer to as a “deterministic activity” with the ADA type mark and outcome probabilities 1;

1.3 Activities of PA type remain unchanged.

Step 2. Introduce the following changes in Algorithm I:

2.1 When considering the node with number 1 (Step 1 of Algorithm I) set $f_1 = M$ in the case of PA-type outcomes. If the outcome is of ASA type set $f_1 = k_{ij}$, where k_{ij} is the number of activity (i, j) in the joint variant $G^*(N^*, A^*)$. In case of ADA outcome activities, set $f_1 = 1$. Set $j = 1$.

2.2 Introduce the same amendments in Step 3.1 of Algorithm I.

Step 3. Apply repeatedly Algorithms I (in its amended version - see Step 2), II, III and IV (to single out all full variants G^{**} entering the joint variant G^*). Note that each full variant is a subnetwork of PERT type without any alternative nodes.

Step 4. For each full variant $G^{**} \in G^*(N^*, A^*)$:

4.1 Calculate objective $F(G^{**})$;

4.2 Calculate the product of all probabilities p_{ij} for all activities $(i,j) \in G^{**}$ (previously removed at Step 1) to determine $\Pr\{G^{**}\}$.

4.3 Calculate constraint value $Q_v(G^{**})$, $1 \le v \le w$.

Step 5. Calculate objective (9.2.1) together with constraint values (9.2.2) for the routine joint variant $G^*(N^*, A^*)$.

Step 6. Implement repeatedly Steps 1 to 5 for each joint variant $G^* \in G(N, A)$ to choose the optimal joint variant.

Algorithm VI. *Determination of Initial Joint and Full Variants.*

Note that the implementation of inverse transformation from modified graphs $G^*(N^*, A^*)$ to initial ones does not result in changing values $F\{G^{**}\}$, $\Pr\{G^{**}\}$ and $H_v\{G^{**}\}$, $1 \le v \le w$, for full variants as well as values (9.2.1-9.2.2) for joint variants. But in order to introduce proper control actions, it is preferable to deal with initial graphs.

Algorithm VI can be easily developed on the basis of examining Procedures 1 and 2 and introducing opposite actions.

Since implementation of Algorithms I and III results in determining the minimal number joint variant, and the joint variant with the next number, then the use of Algorithms I, III, IV and V enables the enumeration lookover of all joint variants, as well as full variants for each joint variant.

One of the basic advantages of enumeration algorithms boils down to the possibility of presenting the algorithm in the form of parallel computations [9,15,127]. It is readily seen that the exact algorithm of solving the problem of optimizing GAAN model can be presented in this form. For example, let i be the ADA-type node with the minimal number. Then the set of joint variants with $f_i = 1$ and that with $f_i = 2$ can be treated independently.

§9.5 Numerical example

The management is faced with the development of an R&D project represented by the GAAN type network in Fig. 9.1. The objective to be optimized is the project's duration with two upper boundary constraints, the project's cost and the project's entropy. The initial data of each activity is presented in Tab. 9.1 (values t_{ij} and c_{ij} are constant). The mean cost of the project should not exceed \$55,000, while the entropy of the project should not exceed 1. The management has to determine the optimal decision policy, i.e., to find the optimal joint variant together with determination of optimal alternative outcomes of ADA type from every decision-making node reached during the realization of the project. Note that the entropy value of a joint variant $G^* \in G(N, A)$ may be calculated as

$$Ent\{G^*\} = -\sum_{\{G^{**}\}\subset G^*}\left[\Pr\{G^{**}\}\ln(\Pr\{G^{**}\})\right]. \tag{9.5.1}$$

Table 9.1. Initial data for the GAAN type network

No.	(i, j)	ASA (p_{ij})	ADA	PA	t_{ij} (months)	c_{ij} ($1,000)
1	$(1,2)$	–	1	–	4	6
2	$(1,3)$	–	1	–	2	5
3	$(1,4)$	0.3	–	–	8	12
4	$(1,5)$	0.7	–	–	6	10
5	$(1,9)$	–	–	1	4	15
6	$(2,6)$	–	–	1	11	9
7	$(2,7)$	–	1	–	15	6
8	$(2,8)$	–	1	–	4	7
9	$(3,7)$	0.6	–	–	3	14
10	$(3,8)$	0.4	–	–	6	10
11	$(3,9)$	–	–	1	8	12
12	$(3,10)$	–	1	–	16	9
13	$(4,9)$	–	1	–	18	7
14	$(4,10)$	–	–	1	1	4
15	$(5,11)$	–	–	1	7	3

The results of implementing Procedures 1 and 2 are given in Tab. 9.2. It can be well-recognized that this is in fact the input information for Algorithm I. Note that $n = 12$ and $M = 3$.

The results of sequential application of Algorithms I to V are presented in Tab. 9.3. Applying Algorithm I provides the minimal vector (3, 1, 3, 3, 3, 0, 1, 3, 3, 1, 0, 0) that corresponds to the first joint variant comprising two full variants. Their corresponding vectors are shown in Tab. 9.3. Implementing algorithms III to V results in calculating the parameters $\overline{T} = 20.5$ and $\overline{C} = 54.2$ of the joint variant G_1^*. The parameters of other joint variants are shown in Table 9.3. Three of the joint variants, namely, G_1^*, G_3^* and G_4^* satisfy both the cost constraint $C\{G^*\} \le \$55{,}000$ and the entropy constraint $Ent\{G^*\} < 1$, while the other three variants G_2^*, G_5^* and G_6^* exceed these constraint levels. Choosing the joint variant with the minimal expected time and giving due consideration to both constraints results in choosing joint variant G_4^* with parameters $\overline{T} = 18.3\,months$, $\overline{C} = \$54{,}600$ and $Ent = 0.61$. The optimal joint variant (for the initial graph) is demonstrated in Fig. 9.3.

Thus, the decision-making process of controlling the regarded project boils down to the following:

a) at the beginning of the project (node 1), we choose activity $(1,2)$ from two alternative outcomes, $(1,2)$ and $(1,3)$, of ADA type;
b) if, in the course of the project realization, node 4 is reached, direction $(4,10)$ has to be chosen;

c) when node 2 is reached, we choose direction $(2,8)$.

Table 9.2. Input information for Algorithm I

No.	(i,j)	ASA	ADA	PA	t_{ij}	c_{ij}	k_{ij}
1	(1,2)	–	–	1	0	0	–
2	(1,3)	–	–	1	0	0	–
3	(1,4)	–	–	1	0	0	–
4	(2,5)	–	1	–	4	6	1
5	(2,6)	–	1	–	2	5	2
6	(3,16)	–	–	1	4	15	–
7	(4,7)	0.3	–	–	8	12	1
8	(4,8)	0.7	–	–	6	10	2
9	(5,9)	–	–	1	0	0	–
10	(5,10)	–	–	1	0	0	–
11	(6,11)	–	–	1	0	0	–
12	(6,12)	–	–	1	0	0	–
13	(7,16)	–	1	–	16	9	1
14	(7,17)	–	1	–	18	7	2
15	(8,17)	–	–	1	1	4	–
16	(8,18)	–	–	1	7	3	–
17	(9,13)	–	–	1	11	9	–
18	(10,14)	–	1	–	5	6	1
19	(10,15)	–	1	–	4	7	2
20	(11,14)	0.6	–	–	3	14	1
21	(11,15)	0.4	–	–	6	10	2
22	(12,16)	–	–	1	8	12	–

It can be well-recognized that if time durations t_{ij} have random values, the determination of the optimal joint variant can be carried out by simulation [49-57].

§9.6 Conclusions

The following conclusions can be drawn from the Chapter:

1. It can be well-recognized that the GAAN model covers a very broad spectrum of R&D projects. Note that the Eisner's R&D projects [37] are merely a particular case of the GAAN model: to obtain the Eisner's network, one has to remove from the GAAN model alternative outcomes of ADA type. Decision-CPM models can be obtained by removing stochastic alternative outcomes of ASA type from the GAAN model. Note that the GERT model is in fact also a particular case of the GAAN network.
2. The GAAN model fully comprises the CAAN network. In order to obtain the latter one has only to withdraw Property III from specifications outlined in §9.1.
3. The GAAN model is essentially more complicated than the CAAN model. But GAAN models cover unique innovative projects which cannot be monitored by CAAN techniques (see, e.g., [67]).

Table 9.3. Determining parameters of joint and full variants

No.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Joint var. #		1			2			3			4			5					6				
Full var. #		-	1	2	-	1	2	-	1	2	-	1	2	-	1	2	3	4	-	1	2	3	4
f_1	PA	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
f_2	ADA	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
f_3	PA	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
f_4	ASA	3	1	2	3	1	2	3	1	2	3	1	2	3	1	1	2	2	3	1	1	2	2
f_5	PA	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
f_6	PA	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3
f_7	ADA	1	1	0	1	1	0	2	2	0	2	2	0	1	1	1	0	0	2	2	2	0	0
f_8	PA	3	0	3	3	0	3	3	0	3	3	0	3	3	0	0	3	3	3	0	0	3	3
f_9	PA	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
f_{10}	ADA	1	1	1	2	2	2	1	1	1	2	2	2	0	0	0	0	0	0	0	0	0	0
f_{11}	ASA	0	0	0	0	0	0	0	0	0	0	0	0	3	1	2	1	2	3	1	2	1	2
f_{12}	PA	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3
C{G*}		54.2			55.2			53.6			54.6			62.7					62.0				
T{G*}		20.5			17.7			21.1			18.3			16.3					16.9				
Ent{G*}		0.61			0.61			0.61			0.61			1.28					1.28				
P{G**}		0	0.3	0.7	0	0.3	0.7	0	0.3	0.7	0	0.3	0.7	0	0.18	0.12	0.42	0.28	0	0.18	0.12	0.42	0.28
C{G**}			57	53		58	54		55	53		56	54		67	64	63	59		65	61	63	59
T{G**}			24	19		24	15		26	19		26	15		24	24	13	13		26	26	13	13

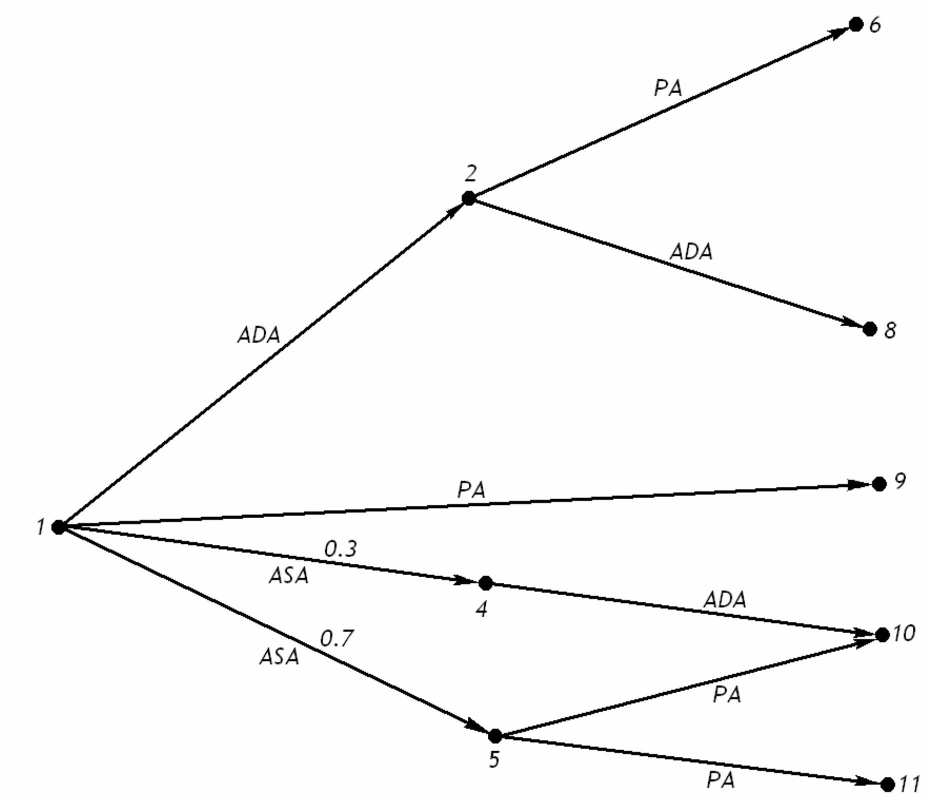

Figure 9.3. The optimal joint variant G_4^* for the initial graph

Chapter 10. Optimization of a Large-Size Alternative CAAN Model by Approximate Methods

§10.1 The CAAN model optimization problem and its complexity

Recall some notions associated with the CAAN model.

A CAAN model is finite, single-source oriented activity-on-arc network $G(N, A)$ with the following properties:

1. Network $G(N, A)$ has one source node and at least two sink nodes.
2. The set of nodes N (excluding the sink nodes) is split into two non-intersecting subsets $\overline{N}$ and $\overline{\overline{N}}$:
   a) Type $\overline{N}$ comprises nodes with the "exclusive or" emitter with stochastic alternatives. Thus, each node $i \in \overline{N}$ corresponds to several (more than one) probability values p_{ij} such that $0 < p_{ij} < 1$ and $\sum_j p_{ij} = 1$. These values are assigned to alternative stochastic activities (i, j) leaving node i and entering node j. Thus, choosing activity (i, j) results in the realization of a random choice from a full group of events.
   b) Type $\overline{\overline{N}}$ comprises nodes also with the "exclusive or" emitter, but with deterministic alternative outcomes. Choosing an alternative activity is the sole prerogative of the project management.

In what follows, we shall call nodes $n \in \overline{N}$ *stochastic nodes* and $n \in \overline{\overline{N}}$ *decision nodes*.

A *joint variant* of graph G is subgraph G_1 satisfying the following conditions:

1) G_1 has a single source coincident with that of graph G;

2) if $i \in G_1$ and $i \in \overline{N}$, then G_1 contains all the arcs leaving node i;

3) if $i \in G_1$ and $i \in \overline{\overline{N}}$, then, in G_1 only one arc leaves node i.

The *realization* (full variant) of a joint variant is a subgraph with a single source coincident with that of the joint variant and with a single arc leaving each node. In other words, the realization of the joint variant is a path connecting the source with one of the sink nodes.

The probability of realization is the product of probabilities p_{ij} of stochastic alternative arcs (i, j) belonging to this realization.

To each arc $(i, j) \in G$, two non-negative deterministic values are assigned: c_{ij} (cost) and t_{ij} (time).

Let $c(s)$ be the sum of costs c_{ij} of all arcs of realization s, $t(s)$ the sum of time values of all arcs of realization s, and $p(s)$ the probability of realization s.

The average sum $c(G_1) = \sum_{s \in G_1} c(s) p(s)$ over all realizations of a joint variant G_1 is called *the cost of the joint variant*.

The average sum $t(G_1) = \sum_{s \in G_1} t(s) p(s)$ over all realizations of a joint variant G_1 is called *the time of the joint variant.*

The problem of choosing an optimal joint variant results in choosing the project's optimal direction to the target. The problem is as follows:

Determine the optimal joint variant $F^* \in \{G_1\}$ such that

$$c(F^*) = \min_{F \subset \{G_1\}} [c(F)] \tag{10.1.1}$$

subject to

$$t(F^*) \le h. \tag{10.1.2}$$

Theorem 10.1. Problem (10.1.1-10.1.2) is NP-complete [75].

Proof. Transform the Knapsack problem [47] which is NP-complete, into problem (10.1.1-10.1.2). Consider a finite enumerated set of pairs (c_i, t_i), $1 \le i \le r$, with given non-negative real values c_i and t_i, together with two pregiven positive values C and T. The Knapsack problem [47] boils down to determining a subset $W \subset \{1,2,...,r\}$ such that

$$\begin{cases} \sum_{i \in W} c_i \le C \\ \sum_{i \in W} t_i \ge T. \end{cases} \tag{10.1.3}$$

Let $\overline{W} = \{1,2,..., r\} \setminus W$ and $D = \sum_{i=1}^{r} t_i$. It can be well-recognized that the Knapsack problem is equivalent to that of determining a set $W \subset \{1,2,...,r\}$ such that

$$\begin{cases} \sum_{i \in W} c_i \le C \\ \sum_{i \in W} t_i \le D - T. \end{cases} \tag{10.1.4}$$

On the other hand, problem (10.1.4) is equivalent to the following particular case of problem (10.1.1-10.1.2). Let $G(N, A)$ be CAAN model with $N = \{n_1, n_2, ..., n_{r+1}\}$, $A = \bigcup_{i=1}^{r} A_i$, where $A_i = \{\alpha_{i,i+1}, \alpha^*_{i,i+1}\}$, $\alpha_{i,i+1}$ and $\alpha^*_{i,i+1}$ being parallel arcs with source node n_i and terminal node n_{i+1}. Let $N = \overline{\overline{N}}$, i.e., the subset of stochastic nodes is empty. For all $i = 1,2,..., r$, set the cost of arc $\alpha_{i,i+1}$ (respectively, $\alpha^*_{i,i+1}$) equal to c_i (respectively, 0), while the time values of both parallel arcs are 0 and t_i. Set also $h = D - T$. Thus, the NP-complete Knapsack problem is clearly equivalent to a particular case of problem (10.1.1-10.1.2), namely: in each node n_i, $1 \le i \le r$, choose an outcome direction, i.e., determine subset W, to satisfy $\sum_{i \in W} c_i \le C$, $\sum_{i \in W} t_i \le h$. Thus, the above particular case of problem (10.1.1-10.1.2) is an NP-complete problem. That means, in turn, that the general problem (10.1.1-10.1.2) is also NP-complete. ■

Since problem (10.1.1-10.1.2) is NP-complete, the optimal solution can be obtained only by means of a lookover algorithm. Such an algorithm was developed in [57]. But for large-size alternative networks a scanning lookover, together with calculating for each joint variant its cost and time values, might require enormous computational time. To overcome this obstacle, we will outline an approximate method based on general ideas of two-parameter combinatorial optimization [9-12,75,110].

It can be well-recognized that problem (10.1.1-10.1.2) relates to a broad variety of two-parameter combinatorial optimization problems. The latter include various assignment problems, minimum spanning tree problems [12], etc.

To be consistent with the general theory outlined in [9-12], we will call a *two-parameter combinatorial optimization* (*TPCO*) *problem* any problem that has a finite set P, a finite family S of subsets of P, a non-negative threshold h, and two non-negative real-valued functions $y: P \to R_+$ (e.g., cost) and $x: P \to R_+$ (e.g., time). One seeks a solution $F^* \in S$ with $y(F^*) = \min\{y(F): F \in S,\, x(F) \le h\}$, where, for any $F \in S$, relations $x(F) = \sum_{f \in F} x(f)$, $y(F) = \sum_{f \in F} y(f)$ hold, f being an element entering F.

An important assumption is implied in the problem, namely: there exists a precise algorithm (it is called Algorithm A) which delivers the optimal solution to the following *one-parameter optimization problem*:

Determine separately $F' \in S$ with $y(F') = \min\{y(F): F \in S\}$, and $F'' \in S$ with $x(F'') = \min\{x(F): F \in S\}$.

It goes without saying that solving a one-parameter optimization problem requires less computational time and is essentially easier than solving the TPCO problem.

It can be well-recognized that problem (10.1.1-10.1.2) is nothing but a particular case of the general problem outlined in [9-12]: the set of activities A of the CAAN model $G(N, A)$ is set P, and real-valued functions $x(F)$ and $y(F)$ are the time and cost values assigned to activity f entering a routine joint variant $F \in S$, where S is the set of joint variants. One has to determine the possibility of solving the one-parameter optimization problem, i.e., to develop Algorithm A applicable to CAAN models. The corresponding algorithm will be outlined below.

§10.2 Approximate algorithm for the CAAN model optimization problem

If only one parameter w_{ij} is assigned to each activity $(i, j) \in G(N, A)$, the optimal joint variant $F^* \in G(N, A)$, $w(F^*) = \min\{w(F): F \in \{G_1\}\}$ can be obtained using

Theorem 10.2. The optimal joint variant F^* can be determined by recurrent relations

$$v_i = \min_j \left(w_{ij} + v_j\right) \qquad (10.2.1)$$

for decision nodes and

$$v_i = \sum_j p_{ij}\left(w_{ij} + v_j\right) \qquad (10.2.2)$$

for stochastic alternative nodes, with initial conditions $v = 0$ for sink nodes.

Proof. In what follows, call values $w(s)$ and $w(G_1)$ the weights of realization s and joint variant G_1, respectively. First, show that the theorem holds for a joint variant, i.e., that the sum $\sum_s p(s)w(s)$ is equal to v_1 for a source node obtained through the use of recurrent relations. Note that for any variant, the above formulae take the form $v_1 = w_{ij} + v_j$ for decision nodes and $v_1 = \sum_j p_{ij}(w_{ij} + v_j)$ for stochastic nodes. This is because in a joint variant, only one deterministic alternative leaves a decision node.

Carry out the proof by induction on the number n of internal nodes of a joint variant.

For $n = 1$, the assertion is obviously true. Assume that it holds for any $n \le k$. Let a variant G_1 have $k+1$ internal nodes. Consider two cases.

1. The source is a decision node. Examine the subnetwork G_2 of variant G_1 including all the arcs and nodes of that joint variant, except the source node and the arc leaving that node. For G_2 the theorem is true, since the number of internal nodes entering G_2 equals k. This implies that $v_2 = \sum_{s'} p(s')w(s')$, where s' is a realization of subnetwork G_2, $p(s')$ and $w(s')$ are the probability and the weight of that realization, respectively, and v_2 is determined by means of the recurrent relation and equals the weighted sum. Note that since $(1,2)$ is a deterministic activity and $s = (1,2) \cup s'$, relations $p(s) = p(s')$ and $\sum_{s'} p(s') = 1$ hold. Then
$$v_1 = w_{12} + v_2 = w_{12} + \sum_{s'} p(s')w(s') = \sum_{s'} p(s')w_{12} + \sum_{s'} p(s')w(s') =$$
$$= \sum_{s'} p(s')[w_{12} + w(s')] = \sum_s p(s)w(s).$$
2. The source is a stochastic node. Assume that two arcs $(1,2)$ and $(1,3)$ are leaving that node. Consider subnetworks G_2 and G_3, where G_2 is the maximal subnetwork with node 2, and G_3 is the maximal subnetwork with a source in node 3. Then, $v_2 = \sum_{s'} p(s')w(s')$ and $v_3 = \sum_{s''} p(s'')w(s'')$. Note that since G_2 and G_3 are *maximal subnetworks*, they include *all possible realizations* which form a full group of events for both subnetworks. Thus, relations $\sum_{s'} p(s') = 1$ and $\sum_{s''} p(s'') = 1$ hold. Further, we obtain
$$v_1 = p_{12}(w_{12} + v_2) + p_{13}(w_{13} + v_3) = p_{12}\left(w_{12} + \sum_{s'} p(s')w(s')\right) + p_{13}\left(w_{13} + \sum_{s''} p(s'')w(s'')\right) =$$
$$= \sum_{s'} p_{12} p(s')[w_{12} + w(s')] + \sum_{s''} p_{13} p(s'')[w_{13} + w(s'')].$$

It can be well-recognized that since $\{s\} \equiv [(1,2) \cup \{s'\}] \cup [(1,3) \cup \{s''\}]$ and two arbitrary realizations $s^* \in [(1,2) \cup \{s'\}]$ and $s^{**} \in [(1,3) \cup \{s''\}]$ *are always different*, relation $\sum_{s'} p_{12} p(s')[w_{12} + w(s')] + \sum_{s''} p_{13} p(s'')[w_{13} + w(s'')] = \sum_s p(s)w(s)$ holds.

The case of more than two arcs can be easily examined by induction.

Let us now prove that the weight of the joint variant obtained by means of recurrent relations (10.2.1-10.2.2) is the minimal one. For the proof, we shall use induction on the number n of internal nodes of the initial network $G(N,A)$. For $n=1$ the assertion holds. Assume that it holds for any $n \le k$. Suppose that two arcs $(1,2)$ and $(1,3)$ are leaving the source node, and consider two cases.

1. The source is a decision node. Examine two maximal subnetworks with sources in nodes 2 and 3. Let v_2 and v_3 be the weights of optimal joint variants of those subnetworks. Show that $v_1 = \min\{w_{12}+v_2; w_{13}+v_3\}$ is the weight of the optimal joint variant. It has been demonstrated earlier that v_1 is a weighted path function. Suppose that the weight v of the optimal joint variant is less than v_1. Note that relation $v = \sum_s p(s)w(s)$ holds over all realizations of the joint variant. Let arc $(1,2)$ belong to the optimal joint variant. Then
$v = \sum_s p(s)w(s) = \sum_s p(s)(w_{12}+w(s')) = \sum_s p(s)w_{12} + \sum_s p(s)w(s') = w_{12} + \sum_s p(s)w(s')$.
Note that since $(1,2)$ is a deterministic activity and $\{s\} \equiv [(1,2) \cup \{s'\}]$, relation $\sum_s p(s)w(s') = \sum_{s'} p(s')w(s')$ holds. Since, by induction assumption, $\sum_{s'} p(s')w(s') = v_2$, $v = w_{12}+v_2$. If $v < v_1$, then $w_{12}+v_2 < v_1 = \min\{w_{12}+v_2; w_{13}+v_3\}$, and we obtain an obvious contradiction. Therefore, $v = v_1$.
2. The source node is an alternative stochastic node. Examine two maximal subnetworks with sources in nodes 2 and 3. Let v_2 and v_3 be the weights of optimal joint variants of these subnetworks. Show that $v_1 = p_{12}(w_{12}+v_2) + p_{13}(w_{13}+v_3)$ is the weight of the optimal joint variant. It has been demonstrated earlier that v_1 is the path-weighted average. Suppose that the weight v of the optimal joint variant is less than v_1. As was shown above,
$v = \sum_s p(s)w(s) = p_{12}\left(w_{12} + \sum_{s'} p(s')w(s')\right) + p_{13}\left(w_{13} + \sum_{s''} p(s'')w(s'')\right)$. Since $\sum_{s'} p(s')w(s') \ge v_2$, and $\sum_{s''} p(s'')w(s'') \ge v_3$, relation $v \ge p_{12}(w_{12}+v_2) + p_{13}(w_{13}+v_3) = v_1$ holds. The evident contradiction proves that $v = v_1$ holds. The case of more than two subnetworks can be easily examined by induction. ■

Thus, to obtain the joint variant with the minimal $w(F)$, the corresponding Algorithm A singles out all the optimal outcome sat each decision node.

Several basic assumptions are implied by the algorithm, which carries out the transformation of the initial graph $G(N,A)$ into the optimal joint variant:

1. The optimal joint variant comprises the source node.
2. If a decision node belongs to the optimal joint variant, then the node for

which recurrent relation (10.2.2) delivers the minimum, also belongs to the optimal joint variant.

3. If a stochastic alternative node belongs to the optimal joint variant, then all the nodes participating in recurrent relation (10.2.2) also belong to the optimal joint variant.

Obviously, the subnetwork obtained by using recurrent relations (10.2.1-10.2.2) is a joint variant, and the applications of these relations to the initial network $G(N, A)$ results in determining just this joint variant. We have proven that the weight of this joint variant is the minimal.

Theorem 10.2 enables developing a one-parameter algorithm which will be further considered on the basis of a numerical example. Developing the algorithm enables, in turn, establishing a two-parameter algorithm for solving the optimization problem of the CAAN model. The algorithm outlined below is, essentially, a transformation of TPCO algorithm for the case of the CAAN model. The step-wise procedure of the algorithm is as follows:

Step 1. Assign two parameters, cost c_{ij} and time t_{ij}, to each activity $(i, j) \in G(N, A)$.

Step 2. Determine the minimal cost joint variant F, i.e., $c(F) = \min\{c(G_1) : G_1 \in G\}$.

Step 3. If the number of such joint variants is more than one, select the variant with the minimal time.

Step 4. If the time of the chosen joint variant does not exceed h, i.e., $t(F) \le h$, then this joint variant is the optimal one. Go to Step 15.

Step 5. Determine the minimal time joint variant H, i.e., $t(H) = \min\{t(G_1) : G_1 \in G\}$.

Step 6. If $t(F) > h$, then the problem has no solution. The algorithm implements an emergency stop.

Step 7. Set $a = c(H) - c(F)$, $b = t(F) - t(H)$, $d = t(F)c(H) - t(H)c(F)$.

Step 8. For each arc $(i, j) \in G$ calculate $w_{ij} = a \cdot t_{ij} + b \cdot c_{ij}$

Step 9. For the network G with parameters w_{ij} assigned to each activity $(i, j) \in G(N, A)$ determine the minimal weight joint variant S, i.e., $w(S) = \min\{w(G_1) : G_1 \in G\}$.

Step 10. If $d = a \cdot t(S) + b \cdot c(S)$ and $t(S) \le h$, then S is an appropriate solution. Set $H = S$. Go to Step 14.

Step 11. If $d = a \cdot t(S) + b \cdot c(S)$ and $t(S) > h$, then H is an appropriate solution. Set $F = S$. Go to Step 14.

Step 12. If $d > a \cdot t(S) + b \cdot c(S)$ and $t(S) \le h$, then set $H = S$. Return to Step 7.

Step 13. If $d > a \cdot t(S) + b \cdot c(S)$ and $t(S) > h$, then set $F = S$. Return to Step 7.

Step 14. Calculate the relative error $\Delta = \dfrac{(c(H) - c(F))(h - t(H))}{c(H)(t(F) - h) + c(F)(h - t(H))}$.

Step 15. The algorithm terminates.

It can be well-recognized that relative error Δ represents, in fact, the performance ratio of the algorithm when applied to a particular CAAN optimization problem. Note that it can be proved [12] that d can never become less than $a \cdot t(S) + b \cdot c(S)$.

§10.3 Numerical example

Tab. 10.1 [82] presents the initial data for a CAAN model $G(N, A)$. In the Table, column p_{ij} designates probabilities of arcs leaving alternative stochastic nodes; column q_{ij} indicates arcs leaving decision nodes, by marking them with "1". The optimization problem boils down to determining the minimal cost joint variant with the time constrained by the upper boundary 26. In other words, one has to determine the optimal joint variant F^* satisfying

$$c(F^*) = \min\{c(G_1) : G_1 \subset G(N, A)\}$$

and $t(F^*) \le 26$.

Table 10.1. Initial data for the CAAN type network

No.	(i,j)	p_{ij}	q_{ij}	t_{ij} (months)	c_{ij} ($1,000)
1	(1,2)	–	1	2	8
2	(1,3)	–	1	4	6
3	(1,4)	–	1	3	9
4	(2,5)	–	1	5	7
5	(2,6)	–	1	8	2
6	(3,6)	0.4	–	7	3
7	(3,7)	0.6	–	6	5
8	(4,7)	–	1	9	1
9	(4,8)	–	1	4	3
10	(5,9)	0.5	–	10	4
11	(5,10)	0.5	–	12	2
12	(6,10)	–	1	14	8
13	(6,11)	–	1	15	5
14	(7,11)	–	1	13	6
15	(7,12)	–	1	16	1
16	(8,12)	0.8	–	14	2
17	(8,13)	0.2	–	11	7

To solve the optimization problem, we shall use the two-parameter approximate algorithm outlined in the previous section. For the sake of simplicity, in the following calculations certain obvious steps from the algorithm procedure are being omitted.

Step 2. Determine the minimum cost joint variant. The initial conditions for the recurrence formulae are $v_9 = v_{10} = v_{11} = v_{12} = v_{13} = 0$. For the rest, we obtain

$$v_8 = p_{8,13}(c_{8,13} + v_{13}) + p_{8,12}(c_{8,12} + v_{12}) = 0.2 \cdot (7+0) + 0.8 \cdot (2+0) = 3;$$

$$v_7 = \min\left\{\underline{c_{7,12} + v_{11}}; c_{7,11} + v_{11}\right\} = \min\{\underline{1+0}; 6+0\} = 1;$$

$$v_6 = \min\left\{\underline{c_{6,12} + v_{11}}; c_{6,10} + v_{10}\right\} = \min\{\underline{5+0}; 8+0\} = 5;$$

$$v_5 = p_{5,10}(c_{5,10} + v_{10}) + p_{5,9}(c_{5,9} + v_9) = 0.5 \cdot (2+0) + 0.5 \cdot (4+0) = 3;$$

$$v_4 = \min\left\{c_{4,8} + v_8; \underline{c_{4,7} + v_7}\right\} = \min\{3+3; \underline{1+1}\} = 2;$$

$$v_3 = p_{3,7}(c_{3,7} + v_7) + p_{3,6}(c_{3,6} + v_6) = 0.6 \cdot (5+1) + 0.4 \cdot (3+5) = 7.8;$$

$$v_2 = \min\left\{\underline{c_{2,6} + v_6}; c_{2,5} + v_5\right\} = \min\{\underline{2+5}; 7+3\} = 7;$$

$$v_1 = \min\left\{\underline{c_{1,4} + v_4}; c_{1,3} + v_3; c_{1,2} + v_2\right\} = \min\{\underline{9+2}; 6+7; 8+7\} = 11.$$

The minimal cost joint variant is therefore $F = \{(1,4);(4,7);(7,12)\}$ with parameters $c(F) = 11$, $t(F) = 28$.

Step 5. Determine the minimal time joint variant. The initial conditions are the same: $v_9 = v_{10} = v_{11} = v_{12} = v_{13} = 0$. For the rest, we obtain

$$v_8 = p_{8,13}(t_{8,13} + v_{13}) + p_{8,12}(t_{8,12} + v_{12}) = 0.2 \cdot (11+0) + 0.8 \cdot (14+0) = 13.2;$$

$$v_7 = \min\left\{t_{7,12} + v_{11}; \underline{t_{7,11} + v_{11}}\right\} = \min\{16+0; \underline{13+0}\} = 13;$$

$$v_6 = \min\left\{t_{6,11} + v_{11}; \underline{t_{6,10} + v_{10}}\right\} = \min\{15+0; \underline{14+0}\} = 14;$$

$$v_5 = p_{5,10}(t_{5,10} + v_{10}) + p_{5,9}(t_{5,9} + v_9) = 0.5 \cdot (12+0) + 0.5 \cdot (10+0) = 11;$$

$$v_4 = \min\left\{\underline{t_{4,8} + v_8}; t_{4,7} + v_7\right\} = \min\{\underline{4+13.2}; 9+13\} = 17.2;$$

$$v_3 = p_{3,7}(t_{3,7} + v_7) + p_{3,6}(t_{3,6} + v_6) = 0.6 \cdot (6+13) + 0.4 \cdot (7+14) = 19.8;$$

$$v_2 = \min\left\{t_{2,6} + v_6; \underline{t_{2,5} + v_5}\right\} = \min\{8+14; \underline{5+11}\} = 16;$$

$$v_1 = \min\left\{t_{1,4} + v_4; t_{1,3} + v_3; \underline{t_{1,2} + v_2}\right\} = \min\{3+17; 4+19.8; \underline{2+16}\} = 18.$$

The minimal time joint variant is therefore $H = \{(1,2);(2,5);(5,9);(5,10)\}$ with parameters $c(H) = 18$, $t(H) = 10$.

Step 7. Calculate values

$a = c(H) - c(F) = 18 - 11 = 7$,

$b = t(F) - t(H) = 28 - 18 = 10$,

$d = t(F)c(H) - t(H)c(F) = 28 \cdot 18 - 18 \cdot 11 = 306.$

Step 8. For each arc (i, j) calculate $w_{ij} = a \cdot t_{ij} + b \cdot c_{ij} = 7 \cdot t_{ij} + 10 \cdot c_{ij}$:

$w_{12} = 7 \cdot 2 + 10 \cdot 8 = 94$;

$w_{13} = 7 \cdot 4 + 10 \cdot 6 = 88$;

$w_{14} = 7 \cdot 3 + 10 \cdot 9 = 111$;

$w_{59} = 7 \cdot 10 + 10 \cdot 4 = 110$;

$w_{5,10} = 7 \cdot 12 + 10 \cdot 2 = 104$;

$w_{25} = 7 \cdot 5 + 10 \cdot 7 = 105$;

$w_{26} = 7 \cdot 8 + 10 \cdot 2 = 76$;

$w_{36} = 7 \cdot 7 + 10 \cdot 3 = 79$;

$w_{37} = 7 \cdot 6 + 10 \cdot 5 = 92$;

$w_{47} = 7 \cdot 9 + 10 \cdot 1 = 73$;

$w_{48} = 7 \cdot 4 + 10 \cdot 3 = 58$;

$w_{6,10} = 7 \cdot 14 + 10 \cdot 8 = 178$;

$w_{6,11} = 7 \cdot 15 + 10 \cdot 5 = 155$;

$w_{7,11} = 7 \cdot 13 + 10 \cdot 6 = 151$;

$w_{7,12} = 7 \cdot 16 + 10 \cdot 1 = 122$;

$w_{8,12} = 7 \cdot 14 + 10 \cdot 2 = 118$;

$w_{8,13} = 7 \cdot 11 + 10 \cdot 7 = 147$.

Step 9. Determine the minimal weight joint variant with the initial conditions for the recurrence formulae $v_9 = v_{10} = v_{11} = v_{12} = v_{13} = 0$:

$$v_8 = p_{8,13}(w_{8,13} + v_{13}) + p_{8,12}(w_{8,12} + v_{12}) = 0.2 \cdot (147 + 0) + 0.8 \cdot (118 + 0) = 123.8;$$

$$v_7 = \min\left\{\underline{w_{7,12} + v_{11}}; w_{7,11} + v_{11}\right\} = \min\{\underline{122 + 0};151 + 0\} = 122;$$

$$v_6 = \min\left\{\underline{w_{6,12} + v_{11}}; w_{6,10} + v_{10}\right\} = \min\{\underline{155 + 0};178 + 0\} = 155;$$

$$v_5 = p_{5,10}(w_{5,10} + v_{10}) + p_{5,9}(w_{5,9} + v_9) = 0.5 \cdot (104 + 0) + 0.5 \cdot (110 + 0) = 107;$$

$$v_4 = \min\left\{\underline{w_{4,8} + v_8}; w_{4,7} + v_7\right\} = \min\{\underline{58 + 123.8};73 + 122\} = 181.8;$$

$$v_3 = p_{3,7}(w_{3,7} + v_7) + p_{3,6}(w_{3,6} + v_6) = 0.6 \cdot (92 + 122) + 0.4 \cdot (79 + 155) = 222;$$

$$v_2 = \min\left\{w_{2,6} + v_6; \underline{w_{2,5} + v_5}\right\} = \min\{76 + 155;\underline{105 + 107}\} = 212;$$

$$v_1 = \min\left\{\underline{w_{1,4} + v_4}; w_{1,3} + v_3; w_{1,2} + v_2\right\} = \min\{\underline{111 + 181.8};88 + 222;94 + 212\} = 292.8.$$

The minimum weight joint variant is therefore $S = \{(1,4);(4,8);(8,12);(8,13)\}$ with parameters $c(S) = 15$, $t(S) = 20.4$.

Step 12. We have $d = 306$, $a \cdot t(S) + b \cdot c(S) = 7 \cdot 20.4 + 10 \cdot 15 = 292.8$, $t(S) = 20.4$, $h = 26$. Since in our case $d > a \cdot t(S) + b \cdot c(S)$ and $t(S) \le h$, set $H = S$ and as a result of this $c(H) = 15$, $t(H) = 20.4$. Return to Step 7.

Step 7. We have $c(F) = 11$, $t(F) = 28$, $c(H) = 15$, $t(H) = 20.4$. Calculate new values

$a = c(H) - c(F) = 15 - 11 = 4$,

$b = t(F) - t(H) = 28 - 20.4 = 7.6$,

$d = t(F)c(H) - t(H)c(F) = 28 \cdot 15 - 20.4 \cdot 11 = 195.6$.

Step 8. For each arc (i, j) calculate $w_{ij} = a \cdot t_{ij} + b \cdot c_{ij} = 4 \cdot t_{ij} + 7.6 \cdot c_{ij}$:

$w_{12} = 4 \cdot 2 + 7.6 \cdot 8 = 68.8$;

$w_{13} = 4 \cdot 4 + 7.6 \cdot 6 = 61.6$;

$w_{14} = 4 \cdot 3 + 7.6 \cdot 9 = 80.4$;

$w_{25} = 4 \cdot 5 + 7.6 \cdot 7 = 73.2$;

$w_{26} = 4 \cdot 8 + 7.6 \cdot 2 = 47.2$;

$w_{36} = 4 \cdot 7 + 7.6 \cdot 3 = 50.8$;

$w_{37} = 4 \cdot 6 + 7.6 \cdot 5 = 62.0$;

$w_{59} = 4 \cdot 10 + 7.6 \cdot 4 = 70.4$;

$w_{5,10} = 4 \cdot 12 + 7.6 \cdot 2 = 63.2$;

$w_{6,10} = 4 \cdot 14 + 7.6 \cdot 8 = 116.8$;

$w_{6,11} = 4 \cdot 15 + 7.6 \cdot 5 = 98.0$;

$w_{7,11} = 4 \cdot 13 + 7.6 \cdot 6 = 97.6$;

$w_{7,12} = 4 \cdot 16 + 7.6 \cdot 1 = 71.6$;

$w_{8,12} = 4 \cdot 14 + 7.6 \cdot 2 = 71.2$;

$w_{47} = 4 \cdot 9 + 7.6 \cdot 1 = 43.6$; $w_{8,13} = 4 \cdot 11 + 7.6 \cdot 7 = 97.2$.

$w_{48} = 4 \cdot 4 + 7.6 \cdot 3 = 38.8$;

Step 9. Determine the minimal weight joint variant with the initial conditions for the recurrence formulae $v_9 = v_{10} = v_{11} = v_{12} = v_{13} = 0$:

$v_8 = p_{8,13}(w_{8,13} + v_{13}) + p_{8,12}(w_{8,12} + v_{12}) = 0.2 \cdot (97.2 + 0) + 0.8 \cdot (71.2 + 0) = 76.4$;

$v_7 = \min\left\{\underline{w_{7,12} + v_{11}}; w_{7,11} + v_{11}\right\} = \min\{\underline{71.6 + 0}; 97.6 + 0\} = 71.6$;

$v_6 = \min\left\{\underline{w_{6,12} + v_{11}}; w_{6,10} + v_{10}\right\} = \min\{\underline{98 + 0}; 116.8 + 0\} = 98$;

$v_5 = p_{5,10}(w_{5,10} + v_{10}) + p_{5,9}(w_{5,9} + v_9) = 0.5 \cdot (63.2 + 0) + 0.5 \cdot (70.4 + 0) = 66.8$;

$v_4 = \min\left\{w_{4,8} + v_8; \underline{w_{4,7} + v_7}\right\} = \min\{38.8 + 76.4; \underline{43.6 + 71.6}\} = 115.2$;

$v_3 = p_{3,7}(w_{3,7} + v_7) + p_{3,6}(w_{3,6} + v_6) = 0.6 \cdot (62 + 71.6) + 0.4 \cdot (50.8 + 98) = 139.68$;

$v_2 = \min\left\{w_{2,6} + v_6; \underline{w_{2,5} + v_5}\right\} = \min\{47.2 + 98; \underline{73.2 + 66.8}\} = 140$;

$v_1 = \min\left\{\underline{w_{1,4} + v_4}; w_{1,3} + v_3; w_{1,2} + v_2\right\} = \min\{\underline{80.4 + 115.2}; 61.6 + 139.68; 68.8 + 140\} =$

$= 195.6$.

The minimum weight joint variant is therefore $S = \{(1,4);(4,7);(7,12)\}$ with parameters $c(S) = 11$, $t(S) = 28$.

Step 11. We have $d = 195.6$, $a \cdot t(S) + b \cdot c(S) = 4 \cdot 28 + 7.6 \cdot 11 = 195.6$, $t(S) = 28$, $h = 26$. Since this time $d = a \cdot t(S) + b \cdot c(S)$ and $t(S) > h$, then H is the required approximate solution, with $c(H) = 15$, $t(H) = 20.4$. Further, set $F = S$, and $c(F) = 11$, $t(F) = 28$.

Step 14. We have $c(F) = 11$, $t(F) = 28$, $c(H) = 15$, $t(H) = 20.4$. Calculate the relative error $\Delta = \dfrac{(c(H) - c(F))(h - t(H))}{c(H)(t(F) - h) + c(F)(h - t(H))} = \dfrac{(15 - 11)(26 - 20.4)}{15 \cdot (28 - 26) + 11 \cdot (26 - 20.4)} = 0.2445$.

Step 15. The algorithm terminates.

We have thus determined joint variant $H = \{(1,4);(4,8);(8,12);(8,13)\}$ representing the required approximate solution for the given CAAN model optimization problem with a relative error of 0.2445.

§10.4 Experimentation

In order to verify the performance of the algorithm for large-size alternative networks, an extensive experimentation was carried out [9]. Alternative networks with 100, 200, …, $1{,}000$ nodes have been considered. For each dimension, 50 initial graphs $G(N, A)$ of CAAN type were simulated using a special program. Cost and time parameters were set by random numbers uniformly distributed between 0 and 99. Value h was simulated by $h = \max\{20n; 0.3n\mu\}$, where μ stands for a random integer uniformly distributed in interval $[1,99]$. For each simulated

CAAN model, the two-parameter algorithm (outlined in §10.2) determined the optimal joint variant together with its relative error Δ. The computational results are represented in Tab. 10.2. In this Table, column $\Re$ displays the maximal number of iterations in each simulated sample.

Table 10.2. Computational results illustrating the efficiency of the algorithm

N \ Δ	$\Delta \leq 0.01$	$0.01 < \Delta \leq 0.05$	$0.05 < \Delta \leq 0.1$	$0.1 < \Delta \leq 0.2$	$0.2 < \Delta \leq 0.3$	$0.3 < \Delta < 1$	Average value $\bar{\Delta}$	$\Re$
100	7	15	8	9	4	7	0.162	5
200	8	12	9	10	5	6	0.125	6
300	6	16	7	11	3	7	0.145	6
400	16	11	9	8	2	4	0.121	6
500	15	13	4	10	5	3	0.105	6
600	9	15	6	11	4	5	0.122	6
700	13	14	10	7	3	3	0.104	7
800	11	21	7	9	1	1	0.110	7
900	10	16	9	8	3	4	0.113	7
1,000	19	12	4	6	7	2	0.096	8

The following conclusions can be drawn from Tab. 10.2:

1. The developed two-parameter approximate algorithm can be applied to large-size CAAN models without any restrictions on the model size.
2. Increasing the model size results in decreasing the relative error Δ, i.e., in increasing the accuracy of the algorithm.
3. Increasing the model size results in a very slow increase of the number of iterations, i.e., the number of iterations realizing the one-parameter algorithm giving the approximate solution.

§10.5 Conclusions

The following conclusions can be drawn from the Chapter:

1. We have developed an iterative approximate algorithm which determines a quasi-optimal solution of the CAAN model optimization problem with two parameters: OV (cost) and RV (time). Also, a relative error determining the accuracy of the solution obtained, is calculated.
2. Extensive experimentation has shown that the algorithm performs well, requires little computational time and provides a very high accuracy. Moreover, increasing the network size results in decreasing the relative error of OV determined by using the approximate algorithms, from the optimal OV. Establishing a quasi-optimal joint variant requires only a small number of iterations.

PART IV

RESOURCE CONSTRAINED PROJECT SCHEDULING FOR STOCHASTIC NETWORK PROJECTS

Chapter 11. Random Resource Delivery Schedules

§11.1 Case of fixed resource capacities

11.1.1 <u>*Introduction*</u>

It can be clearly recognized that there is no shortage of literature on resource constrained project scheduling (see, e.g., [5-6,33,97,109,113,124,155-156,165,167,etc.]). So far all published resource constrained project scheduling algorithms assume fixed activity durations and do not consider stochastic projects of random duration. This is because those algorithms are usually very sensitive and cannot be applied to scheduling procedures based on substituting random activity durations for their average values. Such project schedules with biased estimates usually underestimate the project's duration and, when used in resource constrained project scheduling, provide resource profiles with essential errors. However, a very broad spectrum of innovative R&D projects, including PERT, GERT or VERT type network projects with random activity durations [105], are carried out with limited resources. The need for high quality resource constrained scheduling models for such complicated projects becomes more and more important. Thus, undertaking research in this area is useful for innovative projecting.

We will henceforth consider an activity-on-arc network project of PERT type where each activity requires non-consumable resources of various types with fixed capacities. Each type of resource is in limited supply with a resource limit that is fixed at the same level throughout the project duration. For each activity, its duration is a random variable with given density function. Several alternative density distributions - normal, uniform and beta distribution - will be considered. The problem is to determine starting time values s_{ij} for each activity (i,j) entering the project, i.e., the timing of feeding-in of resources for that activity. Values s_{ij} are not calculated beforehand and are random variables conditional on our future decisions. The model's objective is to minimize the expected project duration. Determination of values s_{ij} is carried out at decision points when at least one activity is ready to be operated and there are free available resources. If, at a certain point of time, a set of more than one activity is ready to be operated but the available amount of resources is insufficient, a competition among the activities is carried out in order to choose a subset of those activities which has to be operated first and can be supplied by the available resources. We decide the competition by solving a zero-one integer programming problem to maximize the total contribution of the accepted activities to the expected project duration. For each activity its contribution is the product of the average duration of the activity and its probability of being on the critical path in the course of the project's realization. Those probability values are calculated via simulation. Solving a zero-one integer programming problem at each decision point results in the following policy: the project management takes all measures to first operate those activities that, being realized, have the greatest effect of decreasing the expected project duration. Only afterwards, does the management take care of other activities. The model is a stochastic optimization problem which cannot be solved in the general case and allows only a heuristic solution.

11.1.2 Notation

Let us introduce the following terms:

$G(N,A)$ - stochastic network project (graph) of PERT type;

$(i,j)\in A$ - the project's activity;

t_{ij} - random duration of activity (i,j);

a_{ij} - lower bound of value t_{ij} (pregiven);

b_{ij} - upper bound of value t_{ij} (pregiven);

μ_{ij} - average value of t_{ij};

r_{ijk} - capacity of the k-th type resource(s) allocated to activity (i,j), $1\le k\le n$ (fixed and pregiven);

n - number of different resources;

R_k - total available resources of type k at the project's management disposal (pregiven and fixed throughout the planning horizon);

$R_k(t)\le R_k$ - free available resources of type k at moment $t\ge 0$;

S_{ij} - the time that resources r_{ijk} are fed in and activity (i,j) starts (a random value conditional on our decisions);

$T(G/S_{ij})$ - random project's duration, on condition that feeding-in of resources r_{ijk} is carried out at moments S_{ij};

$R_k^*(t/S_{ij})$ - maximal value of the k-th resource profile at moment t on condition that activities $(i,j)\in G(N,A)$ start at moments S_{ij};

F_{ij} - the actual moment activity (i,j) is finished ($F_{ij}=S_{ij}+t_{ij}$);

$T(i)$ - earliest possible time of realization of node i;

$p(i,j)$ - conditional probability of activity (i,j) to be on the critical path in the course of the project's realization (dependent on the decisions already taken).

Similar to (4.1.1) and (4.2.1), assume random activity duration t_{ij} distributed by the beta-law with p.d.f.

$$f_t(x)=\frac{12}{\left[b_{ij}-a_{ij}\right]^4}\left[x-a_{ij}\right]\left[b_{ij}-x\right]^2. \tag{11.1.1}$$

Besides beta distribution (11.1.l), the regarded model may adopt other distributions. Three alternative distributions will be considered:

1. t_{ij} has a beta distribution with density function (11.1.1) in the interval $[a_{ij}, b_{ij}]$;
2. t_{ij} has a uniform distribution in the same interval;
3. t_{ij} has a normal distribution with average $\mu_{ij} = 0.5 \cdot (a_{ij} + b_{ij})$ and variance $V_{ij} = [(b_{ij} - a_{ij})/6]^2$.

The initial data of the model for each activity (i, j) includes:
i; j; a_{ij}; b_{ij}; $r_{ij1}, \ldots, r_{ijn}$.

It goes without saying that relations

$$\max_{i,j} r_{ijk} \le R_k, \quad 1 \le k \le n, \tag{11.1.2}$$

hold, otherwise the project cannot be operated.

11.1.3 <u>The model</u>

The problem is to determine values S_{ij} to minimize the expected project duration

$$\min_{S_{ij}} E\left\{T\left(G/S_{ij}\right)\right\}, \tag{11.1.3}$$

subject to

$$R_k^*\left(t/S_{ij}\right) \le R_k \quad \forall t \ge 0, \quad 1 \le k \le n. \tag{11.1.4}$$

We have chosen this objective because various authors, (e.g. [143,156,165]) consider the problem of decreasing the project duration as one of the most urgent ones, especially for stochastic projects of PERT type. The latter usually do not meet their due dates on time [101,143] (see also Chapters 2-3).

Model (11.1.3-11.1.4) is a stochastic optimization problem which cannot be solved in the general case; the problem allows a heuristic solution only.

The basic idea of the heuristic solution is as follows. Decision-making, i.e., determining values S_{ij}, is carried out at essential moments F_{ij} and $T(i)$ (decision points), either when one of the activities (i, j) is finished and additional resources r_{ijk}, $1 \le k \le n$, become available, or when all activities (i, j) leaving node i are ready to be processed. If one or more activities $(i_1, j_1), \ldots, (i_m, j_m)$, $m \ge 1$, are ready to be processed at moment t and all of them can be provided with available resources, the required resources are fed in and activities (i_q, j_q), $1 \le q \le m$, begin to be operated at moment t, i.e., $S_{i_q j_q} = t$, $1 \le q \le m$. If at least for one type k of resources, relation $\sum_{q=1}^{m} r_{i_q j_q k} > R_k(t)$ holds, i.e., there is a lack of available resources at moment t, a competition among the activities has to be arranged to choose a subset of activities that will start to be operated at moment t and can be supplied by resources.

Let us analyze in greater detail the problem of determining values s_{ij}, i.e., the problem of choosing activities to be operated. Problem (11.1.3-11.1.4) refers to a decision-making optimization model to minimize the expected project duration. Thus, supplying the chosen activities with available resources at each decision point centers on reducing the remaining project's duration as much as possible. This means, in turn, that to carry out the competition the project management has to choose and to operate first the subset of activities that provides the maximal total contribution to the expected project duration.

We will assume that in a stochastic network project with random activity durations each activity (i,j) contributes to the expected project duration value $\partial_{ij} = \mu_{ij} \cdot p(i,j)$. Here μ_{ij} is the given average value of the activity duration while $p(i,j)$ is the conditional probability for the activity to be on the critical path. Note that at each decision point t values $p(i,j)$ for all remaining activities (i,j) cannot be calculated beforehand: they are not only dependent on the decisions already taken but are random variables conditional on our future decisions. We suggest a heuristic procedure (see *11.1.4* further on) to determine those values by means of simulation. At each decision point t, all the activities that have not yet started to be operated are simulated using one of the alternative density functions, e.g., (11.1.1). Later on, the critical path of the remaining graph (with simulated activity durations) is determined. By repeating this procedure many times, we obtain frequencies for each activity (i,j) to be on the critical path. Such frequencies are taken as $p(i,j)$. Note that such a simulation approach has been used successfully in other areas of project management, e.g., in budget reallocation models for stochastic network projects [64,66,68].

After obtaining values $p(i_q, j_q)$, $1 \le q \le m$, for all competitive activities at moment t, decision-making boils down to choosing the optimal subset of activities that can be supplied by available resources. The objective is to maximize the sum of values ∂_{ij} for all chosen activities. We suggest solving this problem by using the zero-one programming approach which has been successfully used in similar resource scheduling problems, e.g., in [154-156].

The zero-one programming problem can be formulated as follows: determine integer values $\xi_{i_q j_q}$, $1 \le q \le m$, to maximize the objective

$$\underset{\{\xi_{i_q j_q}\}}{Max} \left\{ \sum_{q=1}^{m} \left[\xi_{i_q j_q} \cdot p(i_q, j_q) \cdot \mu_{i_q j_q} \right] \right\} \qquad (11.1.5)$$

subject to

$$\sum_{q=1}^{m} \left(\xi_{i_q j_q} \cdot r_{i_q j_q k} \right) \le R_k(t), \quad 1 \le k \le n, \qquad (11.1.6)$$

where

$$\xi_{i_q j_q} = \begin{cases} 0 & \text{if activity } (i_q, j_q) \text{ will not obtain resources;} \\ 1 & \text{otherwise.} \end{cases}$$

Problem (11.1.5-11.1.6) is a classical zero-one integer programming problem. Its solution is outlined in many books on operations research, e.g., in [153]. Note that maximizing objective (11.1.5) results in the policy as follows: the project management takes all measures to operate first activities which being realized, decrease more essentially the expected project duration. Only afterwards, does the management take care of other activities.

After feeding-in of resources for the chosen activities, the next earliest "essential" moment is determined and the project's realization proceeds until the sink node cannot be reached. The corresponding heuristic algorithm to schedule the project is outlined below.

11.1.4 <u>The heuristic algorithm</u>

The algorithm [70] to solve problem (11.1.3-11.1.4) is performed in real time; namely, all activities can be operated only after obtaining necessary resources. Essential moments F_{ij} and $T(i)$ cannot be predetermined. However, if we want to evaluate the efficiency of the resource allocation model, we can simulate the algorithm's work by random sampling of the actual duration of activities. By simulating the algorithm's work many times, the average project's duration as well as the probability of accomplishing the project by a given due date (if necessary) can be evaluated.

The heuristic algorithm comprises three subalgorithms as follows:

Subalgorithm I actually governs most of the procedures to be undertaken in the course of the project's realization, namely:

- determines decision points F_{ij} and $T(i)$;
- singles out (at a routine decision point) all the activities that are ready to be operated;
- checks the possibility of supplying these activities with available resources (without undertaking a competition);
- supplies the chosen activities with resources and later on simulates the corresponding activities' durations;
- returns the utilized non-consumable resources to the project management store (at the moment an activity is finished);
- updates the remaining project at each routine decision point.

Subalgorithm II calculates values $p(i,j)$ for all activities entering the remaining project, at a routine decision point. Note that the subalgorithm works *only* in the case when, due to restricted available resources, a competition among the activities waiting to be operated, has to be undertaken. The subalgorithm is implemented by means of simulation as follows:

1. At any routine decision point t, determine all the activities that have not yet started to be operated. Simulate their random durations using one of the alternative density functions.
2. For activities (i,j) entering the remaining project and *being under opera-*

tion at moment t, calculate their remaining durations $F_{ij} - t$.

3. Calculate the critical path length of the remaining graph where activity durations are determined at Steps 1 and 2. Determine all activities that belong to the critical path.
4. Repeat Steps l-3 M times in order to obtain representative statistics.
5. Calculate the frequency for each activity (i, j) to be on the critical path. For a large M, such frequencies are taken as $p(i, j)$.

Note that simulation of activity durations at Step 1 of Subalgorithm II is carried out to determine values $p(i, j)$, i.e., to solve an auxiliary problem, but not to simulate actual activity realizations. The latter are carried out by Subalgorithm I. As outlined above, values $p(i, j)$ are random variables conditional on our future decisions. When we use Subalgorithm II, we do not take future decisions into account. Moreover, the convergence of the frequency values obtained at Step 5, to optimal values $p(i, j)$ is not evident. We see very little chance that these drawbacks can be avoided. However, for practical applications such an approach is effective [62,64,66,68].

Subalgorithm III solves, at a routine decision point t, the multi-dimensional knapsack problem (11.1.5-11.1.6), to choose the subset of activities to be operated and supplied with available resources. Since the initial data for that problem (values $\mu_{i_q j_q}$ and $p(i_q, j_q)$, $1 \le q \le m$) have already been obtained by using Subalgorithms I and II, solving the problem is not difficult. Similar integer programming models have been successfully used for solving various resource-constrained project scheduling problems (see, e.g., [154]). However, several other heuristics might also turn out to be applicable. We have undertaken a comparison between two procedures:

Procedure A is based on solving a zero-one programming problem (11.1.5-11.1.6).

Procedure B is simpler in usage and boils down to the following:

1. After determining values $p(i_q, j_q)$, $1 \le q \le m$, all the competitive activities are sorted in descending order of values $\partial_{i_q j_q} = \mu_{i_q j_q} \cdot p(i_q, j_q)$. In case $p(i_q, j_q) = 0$, the corresponding activities are sorted in descending order of values $\mu_{i_q j_q}$. Activities with higher values $\partial_{i_q j_q}$ are assumed to be of higher priority.
2. All the sorted activities are examined one after another, in the descending order of their priorities, to check, for each activity, the possibility that it can be provided with remaining available resources. If, for a certain activity (i_q, j_q), $1 \le q \le m$, relations $r_{i_q j_q k} \le R_k(t)$, $1 \le k \le n$, hold, the required resources $r_{i_q j_q k}$ are passed to the activity while the remaining resources $R_k(t)$ are updated, $R_k(t) - r_{i_q j_q k} \Rightarrow R_k(t)$, $1 \le k \le n$. Then, the next activity (i_{q+1}, j_{q+1}) is examined. The procedure terminates either when all the available re-

sources are reallocated among activities or all the m activities have been examined.

We have compared both procedures for the numerical example outlined in *11.1.5*. It turns out that the first procedure provides better results. This can be easily explained: both procedures use one and the same objective and are based on the same initial data. However, Procedure A provides an exact solution while Procedure B is a heuristic.

11.1.5 Numerical example

The company is faced with realizing a stochastic network project with non-consumable limited resources. The initial data of the project are given in [70]. The project requires resources of one type, i.e., $n=1$, with resource limit value $R=50$. In order to check the algorithm, 100 simulation runs were undertaken. Three alternative distributions were considered - normal, uniform and beta distributions. For each distribution, on the basis of 100 simulation runs, the p-deciles $W(p)$ for $p=$ 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, together with the project's average duration $\overline{T}\{G(N,A)\}$ were calculated. We have implemented the zero-one programming model in the heuristic algorithm. The summary of results is presented in Tab. 11.1.

Table 11.1. The summary of results

Probability terms	Distributions		
	Normal	Uniform	Beta
$W(0.9)$	461	465	448
$W(0.8)$	457	458	443
$W(0.7)$	454	453	440
$W(0.6)$	451	451	437
$W(0.5)$	449	448	434
$W(0.4)$	447	445	431
$W(0.3)$	445	442	428
$W(0.2)$	440	439	424
$W(0.1)$	436	434	419
$\overline{T}\{G(N,A)\}$	448.85	448.49	433.88

Using Procedure B (see *11.1.4*) results in the following values $\overline{T}\{G(N,A)\}$:

a) Normal distribution: $\overline{T}=461.58$.

b) Uniform distribution: $\overline{T}=461.35$.

c) Beta-distribution: $\overline{T}=447.98$.

Thus, using a simplified heuristic solution versus a more complicated exact solution of problem (11.1.5-11.1.6) results in increasing the expected project duration by 3% only. This seems to be worth paying the price and unavoidable.

Other conclusions can be drawn from the summary:

1. Introducing beta distribution results in carrying out projects with smaller

durations in comparison to normal and uniform distributions.

2. For both normal and uniform distributions, the average project duration is practically the same. But using normal distribution enables random project duration to be obtained with smaller variance than with uniform distribution.
3. It can be clearly recognized that the heuristic algorithm outlined above enables the solution of several important problems in resource constrained project scheduling, namely:

Problem 1. Given resource limit value R and confidence probability p, determine the due date D which can be met with probability not less than p.

The solution of the problem is obtained by means of linear interpolation:

- ❖ Determine integer numbers q and $q+1$, $0 \le q \le 9$, satisfying $q/10 \le p \le (q+1)/10$.
- ❖ Calculate the due date $D = W(q/10) + \{W((q+1)/10) - W(q/10)\}(10p - q)$.

Problem 2. Given value R and due date D, determine confidence probability p.

The solution is similar to that of Problem 1 and is based on applying interpolation methods to statistical data presented in Tab. 11.1.

Problem 3. Given due date D and confidence probability p, determine the minimal value R which enables meeting the deadline on time.

Problem 3 can be solved by varying value R, undertaking numerous simulation runs for each value and applying interpolation methods to the corresponding statistical data. Note that if the number n of different resources is more than one, Problems 1 and 2 remain as easy as before and can be solved by using Tab. 11.1. Problem 3, however, becomes a multi-objective problem with a more difficult solution (see, e.g., [153]).

11.1.6 Conclusions

1. The heuristic algorithm presented here has some advantages. First, it is very simple to use and intuitive. The general idea of the algorithm is to reallocate resources among the project's activities on the basis of priority levels assigned to these activities. Those priority levels are, in essence, the activities' contributions to the project's average duration; they depend both on the activity's average duration and the probability of the activity to be on the critical path in the course of the project's realization. Those probability values can be easily obtained by means of simulation. They have been successfully used for other optimization problems in network planning and control, e.g., in optimal budget redistribution problems for PERT type projects [62,64].
2. The algorithm can be used for practically all activity-on-arc network projects with independent activities of random duration. To be realized each activity requires non-consumable resources of several different types. The resource capacities are fixed and pregiven. The algorithm can be easily implemented on a PC, especially for projects with a medium number of

activities.

3. The algorithm can be used for *any* probability distribution of activity durations. Moreover, each activity may have its individual density function. With minor modifications, the algorithm can be applied to projects comprising activities that may change its probability distribution within the project's realization. For certain activities, as a result of appropriate actions, such changes may be adopted several times. Since the project is usually revised over time, the management's sole requirement is to introduce any alterations in the initial data of the remaining project. This includes implementing additional activities, changing the number of non-consumable resources together with their total available capacities R_k, and capacities r_{ijk}, etc. If, for example, a project becomes late and the activities' durations depend on the assignment of manpower of varying qualifications the management may hire additional workers or may reallocate the most qualified personnel to the most critical activities, etc. The corresponding alterations result in changing the project's initial data; they can be undertaken *at any decision point* t within the project's realization. The heuristic algorithm can adopt these alterations when being performed in real time as well as when being simulated.
4. For certain sets of activities the corresponding durations may be dependent. That means, e.g., that increasing the duration of a certain activity may result in decreasing the durations of other activities. In such cases, multi-dimensional probability distributions have to be introduced. The heuristic algorithm outlined in *11.1.4* can be easily modified to simulate these correlated activities.
5. The main shortcoming of the outlined above model is its applicability for the case of *resources with fixed capacities only*, which obviously restricts the model's flexibility.

§11.2 Case of variable resource capacities

11.2.1 Introduction

We will consider a network project of PERT type with random activity durations and several non-consumable limited resources. For each type of resource k, its limit is fixed throughout the project duration. Each project's activity (i,j) requires resources of various types with variable capacities and is operated at a random speed which depends linearly on the resource amounts r_{ijk} assigned to that activity. The problem is to determine for each activity (i,j) the starting time S_{ij}, i.e., the timing of feeding-in resources, and the assigned resource capacities r_{ijk}. The objective is to minimize the expected project duration [71].

The outlined below research is a further development of the previous §11.1, in which a particular resource constrained scheduling model with fixed resource capacities r_{ijk} was considered. Thus, only starting times S_{ij} are determined. It can be well recognized that such a model is not generalized and fits only certain project management scenarios. It does not cover most cases, when various resource capacities assigned to project activities may vary, e.g., may be utilized within pregiven upper and lower bounds. An activity cannot be operated with even one resource capacity beneath its lower bound, while resources above their upper bounds are redundant. *Moreover, since all the resource capacities in the model developed in §11.1 are fixed and remain constant throughout the planning horizon, they have no influence on the activities' random durations, and the corresponding probability density functions do not incorporate them as parameters*. In practice, this scenario is often unrealistic, since changing the resource capacity assigned to any activity results in changing the density function of the activity's duration. Where several resources are involved, such an influence becomes more complicated and has to be taken into account. Thus, the model presented in §11.1 requires further development and generalization.

In order to solve the resource constrained project scheduling problem with variable capacities we have formulated the general stochastic optimization problem with decision variables S_{ij} and r_{ijk} (call it henceforth Problem A). Values S_{ij} and r_{ijk} are not calculated beforehand and are random variables conditioned on our future decisions. The problem is too complicated to be solved in the general case. To simplify the problem, we replace it by another one, namely, by the knapsack nonlinear resource reallocation problem (call it Problem B). Such a replacement is based on various heuristic assumptions, e.g., that minimizing the average project duration results in reallocating available resources at a routine decision point among those activities (ready to be operated) which deliver the maximal total contribution to the expected project duration. Thus, a stochastic optimization problem is substituted for a deterministic one. Decision variables of problem B are the chosen activities to be supplied by resources and the resource capacities assigned to those activities.

However, even such a simplified model is essentially more complicated than the zero-one integer programming model which was presented in §11.1 for network project scheduling with *fixed* resource capacities (call it Problem C). The classical zero-one integer programming algorithm, which delivers an optimal solution to that problem, cannot be applied to Problem B. Since Problem B is NP-complete, its optimal solution can be obtained only by realizing a lookover algorithm to single out all the feasible solutions. We have developed such an algorithm and we suggest using the latter for cases of small and medium size projects.

For the case where the number of possible feasible solutions becomes very high and much computational time is needed to carry out a lookover, we have developed a heuristic algorithm to solve Problem B.

Thus, the knapsack resource reallocation Problem B, together with both optimization and heuristic algorithms, are the main contributions of §11.2.

Problem B has to be solved at each decision point, when at least more than one activity is ready to be operated but the available amount of resources is limited.

11.2.2 Notation

Let us introduce the following terms:

$G(N,A)$ - stochastic network project (graph) of PERT type;

$(i,j)\in A$ - the project's activity;

Q_{ij} - the amount of activity (i,j) to be operated (pregiven); note that value Q_{ij} can be set in percentages of the total project as well as in other measures;

r_{ijk} - an integer amount of the k-th type resource(s) allocated to activity (i,j), $1\le k\le n$, (a decision variable);

r_{ij} - vector of resource capacities ($r_{ij}\equiv\{r_{ijk}\}$);

$r_{ijk}^{\min}$ - lower bound of value r_{ijk} (pregiven);

$r_{ijk}^{\max}$ - upper bound of value r_{ijk} (pregiven);

$\bar{r}_{ijk}$ - the average value of r_{ijk} ($\bar{r}_{ijk}=0.5\cdot\left(r_{ijk}^{\min}+r_{ijk}^{\max}\right)$);

n - number of different resources;

R_k - total available resources of type k at the project's management disposal (pregiven and fixed throughout the planning horizon);

$R_k(t)$ - free available resources of type k at moment $t\ge 0$;

$\upsilon_{ij}(r_{ij})$ - the speed of operating activity (i,j) in terms of Q_{ij}. Speeds $\upsilon_{ij}(r_{ij})$ are subject
to disturbances and are random values. It is assumed that they depend on resource capacities r_{ijk} linearly, e.g., $\upsilon_{ij}(r_{ij})=\sum_{k=1}^{n}\left(a_{ijk}\cdot r_{ijk}\right)$ hold. Coefficients a_{ijk}, $1\le k\le n$, are pregiven random values;

$\bar{a}_{ijk}$ - average values of a_{ijk};

$t_{ij}(r_{ij})$ - random duration of activity (i,j), on condition that resource capacities r_{ijk}, $1\le k\le n$, are assigned to that activity ($t_{ij}(r_{ij})=Q_{ij}/\upsilon_{ij}(r_{ij})$);

$\mu_{ij}(r_{ij})$ - the average value of $t_{ij}(r_{ij})$;

ψ_{ijk} - additional value by which $\mu_{ij}(r_{ij})$ can be diminished by adding $\Delta r_{ijk}=1$, on condition that all other resource capacities r_{ijv}, $1\le v\le n$, $v\ne k$, are fixed and equal to $\bar{r}_{ijv}$. Thus, values ψ_{ijk} satisfy

$$\psi_{ijk} = \frac{Q_{ij}}{\sum_{r=1}^{n}\left(\bar{a}_{ijr} \cdot \bar{r}_{ijr}\right)} - \frac{Q_{ij}}{\sum_{r=1}^{n}\left(\bar{a}_{ijr} \cdot \bar{r}_{ijr}\right) + \bar{a}_{ijk}} = \frac{Q_{ij} \cdot \bar{a}_{ijk}}{\left\{\sum_{r=1}^{n}\left(\bar{a}_{ijr} \cdot \bar{r}_{ijr}\right)\right\}^2 + \bar{a}_{ijk} \cdot \sum_{r=1}^{n}\left(\bar{a}_{ijr} \cdot \bar{r}_{ijr}\right)}. \quad (11.2.1)$$

S_{ij} - the time that resources r_{ijk} are fed in and activity (i,j) starts (a decision variable);

$T(G/S_{ij}, r_{ij})$ - random project's duration, on condition that feeding-in of resources r_{ijk}, $1 \le k \le n$, is carried out at moments S_{ij};

$R_k^*(t/S_{ij}, r_{ij})$ - maximal value of the k-th resource profile at moment t on condition that activities $(i,j) \in G(N,A)$ start at moments S_{ij} and feeding-in of resources r_{ijk}, $1 \le k \le n$, is carried out also at moments S_{ij};

F_{ij} - the actual moment activity (i,j) is finished ($F_{ij} = S_{ij} + t_{ij}(r_{ij})$);

$T(i)$ - earliest possible time of realization of node i;

$p(i,j)$ - conditional probability of activity (i,j) to be on the critical path in the course of the project's realization.

Let us examine the nature of production speed $\upsilon_{ij}(r_{ij})$ in greater detail. In practice, it is usually not clear exactly how this tempo behaves over time and what is the nature of its random disturbances. What actually happens falls somewhere between two extreme cases:

1. Disturbance occurs only once while adjusting the speed at time point S_{ij}. Then, in the course of processing the activity, the speed remains constant [150];
2. There are continuous stochastic changes in the speed between time points S_{ij} and F_{ij} [150].

In practice, the second case is more realistic, since there are usually various disturbances while processing a project activity. However, from a mathematical viewpoint, it is easier to deal with and to simulate a processing speed that undergoes a random "jump" only once, at moment S_{ij}. *Thus, in the course of simulating the project, we simulate the random speed $\upsilon_{ij}(r_{ij})$ to process each activity $(i,j) \in G(N,A)$ only once, at moment S_{ij}. The activity's random duration is obtained by dividing Q_{ij} by $\upsilon_{ij}(r_{ij})$.*

As to coefficients a_{ijk}, required to determine $\upsilon_{ij}(r_{ij})$, they are random values with a density function in the interval $[a_{ijk}^*, b_{ijk}^*]$, with pregiven values a_{ijk}^* and b_{ijk}^*. As in §11.1, we shall examine three different cases:

1. Values a_{ijk} have a beta-distribution in the interval $\left[a_{ijk}^*, b_{ijk}^*\right]$ with a density function

$$f_t(x) = \frac{12}{\left[b_{ijk}^* - a_{ijk}^*\right]^4}\left[x - a_{ijk}^*\right]\left[b_{ijk}^* - x\right]^2. \tag{11.2.2}$$

2. Values a_{ijk} have a uniform distribution in the same interval.
3. Values a_{ijk} have a normal distribution with average $0.5 \cdot \left(a_{ijk}^* + b_{ijk}^*\right)$ and variance $\left[\left(b_{ijk}^* - a_{ijk}^*\right)/6\right]^2$.

To simulate the processing speed $\upsilon_{ij}\left(r_{ij}\right)$ with assigned resource capacities r_{ijk}, we need to simulate random values a_{ijk} and to apply the linear relation

$$\upsilon_{ij}\left(r_{ij}\right) = \sum_{k=1}^{n}\left(a_{ijk} \cdot r_{ijk}\right).$$

The initial data for each activity (i, j) includes:

i; j; Q_{ij}; a_{ij1}^*; b_{ij1}^*; …, a_{ijn}^*; b_{ijn}^*; $r_{ij1}^{\min}$; $r_{ij1}^{\max}$; …, $r_{ijn}^{\min}$; $r_{ijn}^{\max}$.

Values ψ_{ijk}, $1 \le k \le n$, are calculated on the basis of the initial data by using (11.2.1) with $\bar{a}_{ijk} = 0.5 \cdot \left(a_{ijk}^* + b_{ijk}^*\right)$ for the uniform and the normal distributions and $\bar{a}_{ijk} = 0.6 \cdot a_{ijk}^* + 0.4 \cdot b_{ijk}^*$ for the beta-distribution.

Note, in conclusion, that obvious relations $\max\limits_{i,j} r_{ijk}^{\min} \le R_k$, $1 \le k \le n$, hold, otherwise the project cannot be carried out.

11.2.3 The general model (Problem A)

The problem is to determine values S_{ij} and r_{ijk}, $1 \le k \le n$, to minimize the average project's duration

$$\min_{S_{ij}, r_{ijk}} E\left\{T\left(G/S_{ij}, r_{ij}\right)\right\} \tag{11.2.3}$$

subject to

$$r_{ijk}^{\min} \le r_{ijk} \le r_{ijk}^{\max} \quad \forall (i, j) \in G(N, A), \tag{11.2.4}$$

$$R_k^*\left(t/S_{ij}\right) \le R_k(t) \quad \forall t \ge 0, \; 1 \le k \le n. \tag{11.2.5}$$

Model (11.2.3-11.2.5) refers to a stochastic optimization problem that cannot be solved in the general case; the problem allows a heuristic solution only. Decision-making, i.e., determining values S_{ij} and r_{ijk}, is carried out at essential moments F_{ij} and $T(i)$, either when one of the activities (i, j) is finished and additional resources become available, or when all activities (i, j) leaving node i are ready to be processed. If one or more activities (i_1, j_1), ..., (i_m, j_m), $m \ge 1$, are ready to be processed at moment t and *all of them can be supplied by all types of available resources of maximal capacity*, the required resources are fed in and activities $\left(i_q, j_q\right)$, $1 \le q \le m$, begin to be operated at moment t, i.e., $S_{i_q j_q} = t$, $r_{i_q j_q k} = r_{i_q j_q k}^{\max}$, $1 \le k \le n$. Otherwise a competition has to be arranged to choose the optimal subset of activities that can be supplied by available resources.

As shown above, in §11.1, an important auxiliary procedure precedes holding the competition, namely, calculating, for all the competitive activities (i_q, j_q), their conditional probabilities $p(i_q, j_q)$ to be on the critical path in the course of the project's realization. For the case of fixed resource capacities r_{ijk}, calculating values $p(i_q, j_q)$ is carried out by means of simulation: at each decision point, all the activities that have not yet started to be operated are simulated using the corresponding probability density functions. These functions do not depend on values r_{ijk} and, thus, remain unchanged in the course of the project's realization. Later on, the critical path of the remaining graph with simulated activity durations is determined. By repeating this procedure many times, frequencies for each activity (i_q, j_q) to be on the critical path are calculated and taken as $p(i_q, j_q)$. Values $p(i_q, j_q)$ enter the zero-one integer programming model to carry out the competition.

We will use the same approach, i.e., calculate values $p(i_q, j_q)$ by means of simulation, for the case of variable resource capacities r_{ijk}. However, we do not know beforehand the resource capacity values $r_{i_q j_q k}$ that will be assigned to the activities under competition, as well as to all other activities in the remaining project. Thus, we are unable to simulate the activities' durations, that depend parametrically on values $r_{i_q j_q k}$. In order to overcome these difficulties, several alternative heuristics may be suggested to simulate activities in the remaining project:

1. Take an integer value $r_{ijk} = 0.5 \cdot r_{ijk}^{\min} + 0.5 \cdot r_{ijk}^{\max}$ for all activities that have not yet started and simulate values $t_{ij}(r_{ij})$ to calculate the conditional probabilities which we shall henceforth denote $p_1(i, j)$.
2. Take $r_{ijk} = r_{ijk}^{\min}$ to simulate the probability values which we shall denote $p(i, j)^{\min}$; take $r_{ijk} = r_{ijk}^{\max}$ to simulate the probability values which we shall denote $p(i, j)^{\max}$; calculate final values $p_2(i, j) = 0.5 \cdot p(i, j)^{\min} + 0.5 \cdot p(i, j)^{\max}$.
3. Calculate final probability values
$p_3(i, j) = 0.25 \cdot p(i, j)^{\min} + 0.25 \cdot p(i, j)^{\max} + 0.5 \cdot p_1(i, j) = 0.5 \cdot [p_1(i, j) + p_2(i, j)]$.
4. Each value r_{ijk} is a simulated integer value uniformly distributed in the interval $[r_{ijk}^{\min}, r_{ijk}^{\max}]$. These values are used later on to simulate random speeds $\upsilon_{ij}(\bar{r}_{ij})$ and random activity durations $t_{ij}(\bar{r}_{ij})$. Denote the conditional probability values $p_4(i, j)$.

The four alternative heuristics outlined above will be compared below to obtain the most effective one. It goes without saying that others may be suggested as well.

After calculating conditional probabilities $p(i, j)$ the problem of optimal resource reallocation among the competitive activities has to be solved.

11.2.4 Knapsack resource reallocation problem (Problem B)

We have assumed in §11.1 that in a stochastic network project with random activity durations and fixed resource capacities, each activity (i,j) contributes to the expected project duration value

$$\sigma_{ij} = p(i,j)\cdot \mu_{ij}\,. \tag{11.2.6}$$

Using the same assumption for the case of variable resource capacities, we have to calculate value σ_{ij} depending on vector r_{ij}. Assume (see *11.2.2*) that the average duration $\mu_{ij}(r_{ij})$ is calculated as follows:

$$\mu_{ij}(r_{ij}) = \mu_{ij}(r_{ij}^{\min}) + \sum_{k=1}^{n}\left[(r_{ijk} - r_{ijk}^{\min})\cdot \psi_{ijk}\right] = \left\{\mu_{ij}(r_{ij}^{\min}) - \sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk}^{\min})\right\} + \sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk}).$$

Assume, further, that the constant value in braces is essentially smaller than the value of the second term and can be neglected. Thus, value $\mu_{ij}(r_{ij})$ satisfies approximately

$$\mu_{ij}(r_{ij}) = \sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk}).$$

Using (11.2.6) we finally obtain

$$\sigma_{ij}(r_{ij}) = \sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk}). \tag{11.2.7}$$

Thus $\sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk})$ is the value by which the average duration of activity (i,j) can be diminished by supplying the activity with resources r_{ijk}, $1 \le k \le n$, and $p(i,j)\cdot\sum_{k=1}^{n}(\psi_{ijk}\cdot r_{ijk})$ is the value by which the average project duration can be diminished. Taking into account the fact that to carry out the competition among activities (i_q, j_q), $1 \le q \le m$, the project management has to choose the subset of activities and to reallocate among them the available resources *in order to maximize the total contribution to the expected project duration*, we suggest the following knapsack resource reallocation problem (Problem B) to be solved at each decision point t:

Determine optimal values $S_{i_q j_q k}$ and $r_{i_q j_q k}$, $1 \le k \le n$, $1 \le q \le m$, to maximize objective

$$J = \underset{S_{i_q j_q k},\, r_{i_q j_q k}}{Max}\left\{\sum_{q=1}^{m}\left[\xi_{i_q j_q}\cdot p(i_q, j_q)\right]\cdot\sum_{k=1}^{n}\left(r_{i_q j_q k}\psi_{i_q j_q k}\right)\right\} \tag{11.2.8}$$

subject to

$$r_{i_q j_q k}^{\min} \le r_{i_q j_q k} \le r_{i_q j_q k}^{\max} \quad \forall (i_q, j_q)\in G(N,A), \tag{11.2.9}$$

$$\sum_{q=1}^{m}\left(\xi_{i_q j_q}\cdot r_{i_q j_q k}\right) \le R_k(t) \quad \forall t \ge 0,\ \ 1 \le k \le n, \tag{11.2.10}$$

where

$$\xi_{i_q j_q} = \begin{cases} 0 & \textit{if activity } (i_q, j_q) \textit{ will not obtain resources;} \\ 1 & \textit{otherwise.} \end{cases}$$

Although Problem B is obtained by simplifying the general Problem A by means of various heuristic assumptions, it is nevertheless a very complicated nonlinear problem.

Theorem. *Problem* (*11.2.8-11.2.10*) *is NP-complete.*

Proof. Consider a particular case of the knapsack problem (11.2.8-11.2.10):

1. Set $n = 1$, i.e., the project utilizes resources of one type only.
2. Assume $r_{i_q j_q}^{\min} = r_{i_q j_q}^{\max} = r_{i_q j_q}$, $1 \le q \le m$, i.e., the resource capacities are fixed and remain constant.
3. Set $P_{i_q j_q} \psi_{i_q j_q} = 1$ for all q.

Thus, problem (11.2.8-11.2.10) is transformed to a zero-one knapsack problem (Problem C)

$$\underset{\xi(i_q, j_q)}{Max} \left\{ \sum_{q=1}^{m} \left[r_{i_q j_q} \cdot \xi(i_q, j_q) \right] \right\} \tag{11.2.11}$$

subject to

$$\sum_{q=1}^{m} \left[r_{i_q j_q} \cdot \xi(i_q, j_q) \right] \le R(t), \tag{11.2.12}$$

$$\xi(i_q, j_q) = \{0,1\}. \tag{11.2.13}$$

We will show that problem (11.2.11-11.2.13) delivers a solution to the classical *Partition Problem* [24] as follows:

Given a set of positive integer numbers $(w_1, w_2, ..., w_m)$, determine the optimal subset $(w_{q_1}, w_{q_2}, ..., w_{q_r})$, $r \le m$, $q_s \in \{1, m\}$, $1 \le s \le r$, satisfying $\sum_{s=1}^{r} w_{q_s} = M$, where M is a pregiven integer value.

Set $R(t) = M$ and $r_{i_q j_q} = w_q$, $1 \le q \le m$. If the optimized value of objective (11.2.11) is equal to M, zero-one values $\{\xi_{i_q}, \xi_{i_q j_q}\}$ deliver the optimal solution to the partition problem. Otherwise, i.e., if the objective is less than M, the partition problem has no solution.

Coffman [24] has proved that the partition problem is NP-complete. Thus, problem (11.2.11-11.2.13) is NP-complete too. But if a particular problem is NP-complete then the principal problem (11.2.8-11.2.10) is also NP-complete. ■

Let us examine the results of the Theorem in greater detail. Both the knapsack reallocation Problem B and the zero-one integer programming Problem C are NP-complete. A classical zero-one integer programming algorithm [153] which delivers a precise solution to Problem C, cannot be applied to Problem B. It can be well recognized that the set of feasible solutions of Problem B comprises that of Problem C. Thus, especially for wide ranges $[r_{ijk}^{\min}, r_{ijk}^{\max}]$, $(i, j) \in G(N, A)$, $1 \le k \le n$, the number of feasible solutions of Problem B may become very large.

To solve Problem B, two algorithms have been developed:

- a lookover algorithm which singles out all feasible solutions and, due to the NP-completeness of problem (11.2.8-11.2.10), is the *only algorithm* that provides a precise solution of that problem,
- a heuristic algorithm which delivers an *approximate solution* and offers less computational time than a lookover algorithm.

Both algorithms are outlined in *11.2.5*.

11.2.5 Optimal and heuristic solutions of the knapsack reallocation problem

The developed heuristic algorithm to solve problem (11.2.8-11.2.10) singles out the subset of activities to be operated (among the competitive ones) and determines resource capacities r_{ijk} for each chosen activity and each type of resource. The algorithm comprises several subalgorithms as follows:

Subalgorithm I sorts m competitive activities (i_q, j_q), $1 \le q \le m$, *in descending order of the product*

$$I(i_q, j_q) = p(i_q, j_q) \cdot \sum_{k=1}^{n} \left[\psi(i_q, j_q, k) \cdot 0.5\left(r_{i_q j_q}^{\min} + r_{i_q j_q}^{\max}\right)\right], \qquad (11.2.14)$$

on condition that $p(i_q, j_q) > 0$. Thus, activities with greater contribution to the project's average duration are considered to be more significant and possess smaller ordinal numbers. Note that values $p(i_q, j_q)$ are calculated preliminarily by means of simulation, while other values in (11.2.14) are either pregiven or calculated beforehand by (11.2.1). Activities with $p(i_q, j_q) = 0$ have to be rescheduled at the end of the schedule in decreasing order of values

$$\sum_{k=1}^{n} \left[\psi(i_q, j_q, k) \cdot 0.5\left(r_{i_q j_q}^{\min} + r_{i_q j_q}^{\max}\right)\right].$$

Assume for simplicity that after rescheduling m competitive activities by using Subalgorithm I, the new schedule will start from (i_1, j_1) and finish with (i_m, j_m). Thus, each activity obtains its ordinal number from 1 to m.

Subalgorithm II determines the *basic set* of the schedule. We will henceforth call the basic set the *chain of consecutive activities of maximal length, beginning from activity* (i_1, j_1) *which can be actually supplied by available resources of maximal capacities for all types of resources*. Thus, if the basic set comprises f activities $(i_1, j_1), \ldots, (i_f, j_f)$, relations

$$\sum_{q=1}^{f} r_{i_q j_q k}^{\max} \le R_k(t), \quad 1 \le k \le n, \qquad (11.2.15)$$

hold.

It goes without saying that the basic set may be empty, i.e., $f = 0$, if relation $r_{i_1 j_1 k}^{\max} \ge R_k(t)$ holds for at least one type of resource.

Subalgorithm III carries out the lexicographical scanning in the space of feasible solutions. We will henceforth call a feasible solution of the resource reallocation problem (11.2.8-11.2.10) a set of $b \le m$ activities satisfying the following conditions:

1. The set comprises the basic set of the schedule: other activities may not be consecutive.
2. The set can actually be supplied by available resources, i.e., each activity (i_q, j_q) entering the set has to obtain not less than $r^{\min}_{i_q j_q k}$ resources of type k, $1 \le k \le n$. Thus, the former demand for the basic set to obtain only $r^{\max}_{i_q j_q k}$ resources is now withdrawn.

It can be well recognized that each feasible solution is defined by the sequence of ordinal numbers of the activities that enter the set of the solution. Since all the activities are enumerated with different numbers, a lexicographical order in the space of feasible solutions can be introduced. To compare two different solutions, one has to take the corresponding sequences and compare by pair the elements of those sequences: a pair of differing elements must be found while all the other previous pairs coincide. If, for that differing pair, the element of the first sequence is less than that of the second, the first sequence (together with the feasible solution) lexicographically precedes the second.

Subalgorithm Ill consists of two main parts: the procedure for choosing the first maximal feasible solution and the procedure for transferring from one arbitrary maximal feasible solution to the next one in lexicographical order. *A feasible solution is called the maximal one if it cannot be enlarged by adding any other activity entering the schedule*. It can be well recognized that any part of the maximal feasible solution which comprises the basic set, is also a feasible solution.

Determine the first maximal feasible solution as follows: add the next activity (i_{f+1}, j_{f+1}) to the basic set and examine the possibility of supplying the set (i_q, j_q), $1 \le q \le f+1$, by resources of *minimal capacities*. If this is possible, then the enlarged set is a new feasible solution that has to be stored in a special array. Later on, we proceed to examine the next activity (i_{f+2}, j_{f+2}), etc. If adding a certain activity (i_c, j_c), $f < c \le m$, does not result in obtaining a feasible solution, omit that activity and turn to examining the next one, (i_{c+1}, j_{c+1}). The procedure terminates after examining all the activities entering the schedule.

Now let $(i_1, j_1), \ldots, (i_f, j_f), (i_g, j_g), \ldots, (i_h, j_h), (i_d, j_d)$ be an arbitrary maximal feasible solution. The procedure for determining the next one is as follows: exclude the last link (i_d, j_d) and find out whether it is possible to determine a *new feasible schedule* (*which does not coincide with those obtained before*) while applying the first procedure. If there is no such a possibility, exclude the link (i_h, j_h) and again apply the first procedure of determining a new feasible schedule, and so on. This procedure terminates when the consequently truncated feasible solution comprises only the basic set.

Subalgorithm IV determines, for each routine feasible solution, its *optimal* resource reallocation to maximize objective (11.2.8). Let $(i_1, j_1),...,(i_g, j_g)$ be the feasible solution comprising $g \le m$ activities. The work of Subalgorithm IV is as follows:

1. Supply all the activities (i_q, j_q), $1 \le q \le g$, with all types of resources of minimal capacity, i.e., assign $r_{i_q j_q k} = r^{\min}_{i_q j_q k}$, $1 \le k \le n$.
2. Update the available resources $R_k(t) - \sum_{q=1}^{g} r^{\min}_{i_q j_q k} \Rightarrow R_k(t)$, $1 \le k \le n$.
3. Set counter $w = 1$.
4. Reschedule activities (i_q, j_q), $1 \le q \le g$, in descending order of the product $I_w(i_q, j_q) = p(i_q, j_q) \cdot \psi(i_q, j_q, w)$. Activities with $p(i_q, j_q) = 0$ have to be sorted at the end of the schedule, in descending order of values $\psi(i_q, j_q, w)$. Let the newly rescheduled sequence be $\left(i_q^{(w)}, j_q^{(w)}\right)$, $1 \le q \le g$.
5. The newly sorted activities are examined one after another, in descending order of their priorities, in order to check the possibility, for each activity, that it can be supplied with additional available resources
$$\Delta r\left(i_q^{(w)}, j_q^{(w)}, w\right) = \min\left\{ r^{\max}_{i_q^{(w)} j_q^{(w)} w} - r^{\min}_{i_q^{(w)} j_q^{(w)} w}, R_w(t) \right\}. \tag{11.2.16}$$
Later on the remaining resources are updated
$$R_w(t) - \Delta r\left(i_q^{(w)}, j_q^{(w)}, w\right) \Rightarrow R_w(t), \tag{11.2.17}$$
and the next activity $\left(i_{q+1}^{(w)}, j_{q+1}^{(w)}\right)$ is examined. The procedure terminates either when all available resources of type w are reallocated among activities, or all the g activities have been examined.
6. Counter w works, $w + 1 \Rightarrow w$.
7. If $w \ge n$, apply the next step. Otherwise return to 4.
8. Subalgorithm IV terminates.

Thus, the idea of Subalgorithm IV is to reallocate all types of available resources among the activities separately, one type after another. If, for a particular activity, the shortage of a certain type of resource adds more to the average project duration than for another activity, the remaining available resources of that type must obviously be assigned to the first activity *rather than* to the second.

Note that Subalgorithm IV is an *optimal* procedure that is implemented in the Algorithm.

Subalgorithm V calculates, for each feasible solution with optimal resource reallocation, the objective function (11.2.8) and determines the solution that delivers the maximal value to that objective. This solution is taken as optimal.

Note, in conclusion, that the input information for the heuristic algorithm outlined above is not precise: it cannot be calculated beforehand and is determined by means of simulation and on the basis of various assumptions and heuristics. As to the algorithm itself, a conclusion can be drawn as follows: reducing the basic set results in increasing the number of feasible solutions. Thus, removing Subalgorithm II results in obtaining an optimal solution by means of a full lexicographical lookover. If necessary, the basic set may be reduced in order to organize a compromise between the amount of computations and the proximity of the heuristic solution to the optimal one.

11.2.6 *Decision-making algorithm*

The decision-making algorithm determines at each essential moment t, when at least one activity (i,j) is ready to be processed, both the starting moment S_{ij} and the vector of resource capacities r_{ij} allocated to each of those activities. This is carried out by solving resource reallocation problem (11.2.8-11.2.10). The algorithm is carried out in real time; namely, each iteration of the algorithm can be performed only after either one of the activities (i,j) is finished and additional resources become available (moment F_{ij}), or at the earliest possible time of realization of node i (moment $T(i)$). At that moment all activities (i,j) leaving node i are ready to be processed. The actual duration of each activity is obtained in the course of the project's realization, on the basis of allocated values r_{ij}. Note that before solving problem (11.2.8-11.2.10) an auxiliary problem to calculate conditional probabilities $p(i,j)$ for all remaining activities (i,j) has to be solved. We suggest solving that problem by means of simulation (see §11.1) for all kinds of resource constrained projects, including real-time projects.

However, if we want to evaluate the efficiency of the decision-making model, e.g., to calculate the probability of meeting the project's due date on time, we can simulate the project's realization by randomly sampling the actual duration of each activity. In this case, after determining values r_{ijk}, $1 \le k \le n$, the random value $t_{ij}(r_{ij})$ can be simulated as follows:

$$t_{ij}(r_{ij}) = \frac{Q_{ij}}{\sum_{k=1}^{n}\left(a_{ijk}^{s} \cdot r_{ijk}\right)}, \tag{11.2.18}$$

where a_{ijk}^{s} is the simulated value of random variable a_{ijk} with a given density function in the interval $\left[a_{ijk}^{*}, b_{ijk}^{*}\right]$ and pregiven lower and upper bounds a_{ijk}^{*} and b_{ijk}^{*}.

By simulating the development of the project many times, the probability of meeting the due date on time, as well as other parameters, can be evaluated. The following *11.2.7* presents some experimentation based on evaluating the decision-making model's performance with some widely used probability distributions for simulating production speeds υ_{ij}.

11.2.7 Experimentation

The efficiency of the decision-making algorithm can be illustrated by a numerical example. The company is faced with carrying out a stochastic network project where each activity utilizes five non-consumable resources with variable capacities. The initial data of the project, as well as the calculated values ψ_{ijk}, are outlined in [71]. The resource limit values R_k, $1 \le k \le 5$, are $R_1 = 50$, $R_2 = 45$, $R_3 = 100$, $R_4 = 155$ and $R_5 = 270$.

In order to validate the decision-making algorithm various examples were run. The experimental design is presented in Tab. 11.2.

Table 11.2. The experimental design

Models	Levels of variation	Number of levels
Distribution of random values a_{ijk}	Normal, uniform, beta	3
Heuristic to calculate $p(i,j)$	$p_1(i,j)$, $p_2(i,j)$, $p_3(i,j)$, $p_4(i,j)$	4
Solution for the knapsack problem (11.2.8-11.2.10)	Optimal, quasi-optimal, heuristic	3

Three models were varied: distribution of a_{ijk}, heuristic to simulate confidence probabilities $p(i,j)$, and the level of proximity to the optimal solution for the knapsack problem (11.2.8-11.2.10).

Three alternative distributions of random values a_{ijk} are considered:

1. a_{ijk} has a normal distribution with average $0.5 \cdot \left(a^*_{ijk} + b^*_{ijk}\right)$ and variance $\left[\left(b^*_{ijk} - a^*_{ijk}\right)/6\right]^2$.
2. a_{ijk} has a uniform distribution in the interval $\left[a^*_{ijk}, b^*_{ijk}\right]$.
3. a_{ijk} has a beta distribution with density function (11.2.2) in the same interval.

As to simulating confidence probabilities $p(i,j)$, four different heuristics outlined in *11.2.3* have been considered:

1. $p_1(i,j)$ with $r_{ijk} = 0.5 \cdot \left(r^{\min}_{ijk} + r^{\max}_{ijk}\right)$.
2. $p_2(i,j) = 0.5 \cdot \left[p(i,j)^{\min} + p(i,j)^{\max}\right]$.
3. $p_3(i,j) = 0.5 \cdot \left[p_1(i,j) + p_2(i,j)\right]$.
4. $p_4(i,j)$ with $r_{ijk} = \alpha \cdot r^{\min}_{ijk} + (1-\alpha) \cdot r^{\max}_{ijk}$, where $\alpha \equiv U[0,1]$, and r_{ijk} is a simulated integer value uniformly distributed in the interval $\left[r^{\min}_{ijk}, r^{\max}_{ijk}\right]$.

In order to obtain a representative statistics for calculating confidence probabilities $p(i,j)$, 150 simulations for each heuristic have been performed. The number of simulations was determined by applying the classical estimation theory outlined in Chapter 3. Given the error in estimating the probability $p(i,j)$, the confidence coefficient and the sample standard deviation, we can determine the sample size of simulation runs.

Three different approaches to solve the knapsack problem (11.2.8-11.2.10) are considered:

1. The basic set (see *11.2.5*) is removed and the *optimal solution* for the knapsack problem (11.2.8-11.2.10) is obtained.
2. A *heuristic solution* to problem (11.2.8-11.2.10) is obtained by using the basic set (see *11.2.5*).
3. A *quasi-optimal solution* to problem (11.2.8-11.2.10) is obtained by reducing the basic set. The idea is as follows: the basic set is consecutively, step-by-step, reduced by one activity (see *11.2.5*). This, in turn, results both in increasing the number of feasible solutions and in increasing the value of the objective J of problem (11.2.8-11.2.10). At each step the chosen activity to be removed is that one which, in comparison with other activities from the remaining basic set, contributes the minimal weight to the objective J. After removing the chosen activity the problem is re-solved for the reduced basic set. The step-by-step procedure is followed until objective J ceases to increase. It can be well recognized that such an approach does not always lead to the optimal solution.

Thus, a total of 36 combinations (3 x 4 x 3) were considered. For each combination 100 runs were performed. That number of statistical trials enables estimating via simulation all the project's parameters, as outlined in Chapter 3.

Two outcome values are considered, as follows:

- $\overline{T}_{pr}$ is the average duration of carrying out the project;
- $\overline{T}_c$ is the average computational time of one simulation run.

The summary of the results obtained is presented in Tab. 11.3.

The following conclusions can be drawn from the summary:

1. It can be well recognized that introducing beta distribution for random values a_{ijk} results in realizing projects with larger durations in comparison with normal and uniform distributions. This is because average values $\overline{a}_{ijk} = 0.6 \cdot a^*_{ijk} + 0.4 \cdot b^*_{ijk}$ for a beta distribution (11.2.2) are always smaller than values $\overline{a}_{ijk} = 0.5 \cdot \left(a^*_{ijk} + b^*_{ijk}\right)$ for normal and uniform distributions. This results both in smaller production speeds $\upsilon_{ij}(r_{ij})$ and in higher activity durations $t_{ij}(r_{ij})$ for each activity entering the project. As to normal and uniform distributions, introducing normally distributed a_{ijk}, results in production speeds $\upsilon_{ij}(r_{ij})$ with smaller variances than with uniform distribution. This, in turn, results in smaller project duration in comparison with the uniformly distributed values a_{ijk}. Thus, using normal distribution enables meeting the due date with the highest confidence probability.
2. Substituting any distribution for another one does not result in any considerable increase of the average computational time $\overline{T}_c$.
3. For all methods of calculating conditional probabilities $p(i, j)$ the average project's duration is practically the same. Conclusions can be drawn that:

- method $p_2(i,j)$ doubles the average computational time $\overline{T}_c$ in comparison with $p_1(i,j)$ and is much more complicated than the latter approach;
- method $p_3(i,j)$ practically trebles values $\overline{T}_c$ and is more complicated than method $p_2(i,j)$.

Table 11.3. The summary of results

Distribution of values a_{ijk}	Methods of calculating conditional probabilities $p(i,j)$	Methods of solving the knapsack problem (11.2.8-11.2.10)					
		Optimal		Quasi-optimal		Heuristic	
		Outcome values					
		$\overline{T}_{pr}$	$\overline{T}_c$	$\overline{T}_{pr}$	$\overline{T}_c$	$\overline{T}_{pr}$	$\overline{T}_c$
NORMAL	$p_1(i,j)$	558.9	24.9	564.9	24.9	565.2	24.8
	$p_2(i,j)$	557.7	48.2	562.9	48.1	564.5	48.1
	$p_3(i,j)$	560.1	71.8	562.0	71.8	564.4	71.7
	$p_4(i,j)$	558.3	25.1	564.6	25.1	573.9	25.0
UNIFORM	$p_1(i,j)$	561.8	22.1	574.9	22.1	575.7	22.0
	$p_2(i,j)$	563.3	42.6	572.4	42.6	573.7	42.6
	$p_3(i,j)$	562.2	63.5	572.2	63.5	576.0	63.4
	$p_4(i,j)$	562.4	22.4	574.7	22.4	577.1	22.3
BETA	$p_1(i,j)$	578.1	24.5	582.2	24.4	584.7	24.3
	$p_2(i,j)$	578.6	47.3	582.6	47.2	587.6	47.2
	$p_3(i,j)$	577.4	70.5	586.3	70.4	586.8	70.4
	$p_4(i,j)$	582.1	24.7	584.0	24.7	586.0	24.6

Example. For a normal distribution of values a_{ijk}, the optimal solution of the knapsack problem (11.2.8-11.2.10), and for uniformly simulated resource capacities r_{ijk} to calculate conditional probabilities $p_4(i,j)$, we obtain the following simulated outcome values:

- the average project's duration $\overline{T}_{pr} = 558.3$;
- the average computational time of one simulation run $\overline{T}_c = 25.1$ sec;
- for both methods $p_1(i,j)$ and $p_4(i,j)$ the average project duration, as well as the average computational time, are practically the same. But $p_1(i,j)$ is simpler since, unlike $p_4(i,j)$, values r_{ijk} are deterministic ones and do not require simulation.
  Thus, using $p_1(i,j)$ is simpler, offers less computational time and is not less efficient than using methods $p_2(i,j) \div p_4(i,j)$. We recommend implementing $p_1(i,j)$ by calculating conditional probabilities $p(i,j)$ for project scheduling problems.
- It can be well recognized that for our example the optimal method of solving the multidimensional knapsack problem (11.2.8-11.2.10) compares favorably with other heuristic algorithms and results in smaller project dura-

tions. As to the computational time values $\overline{T}_c$, they do not depend, in practice, on the method of solving the resource reallocation problem (11.2.8-11.2.10). This can be easily explained: the computational time $\overline{T}_c$ depends mainly on calculating confidence probabilities $p(i,j)$ while solving problem (11.2.8-11.2.10) results in a very small contribution to value $\overline{T}_c$.

11.2.8 *Conclusions*

The following conclusions can be drawn from §11.2:

1. The resource constrained project scheduling model can be applied to all kinds of PERT network projects under random disturbances that utilize several non-consumable resources with variable capacities. Those projects include various R&D projects, construction projects, etc.
2. The model determines, at each decision point, the subset of activities from those ready to be operated and reallocates available resources among the chosen activities. The optimal knapsack resource reallocation problem is a NP-complete one. Several solutions to the problem - an optimal solution based on a lexicographical lookover, and various approximate solutions obtained by using heuristic procedures - are considered.
3. The presented resource reallocation model has been used for several PERT network projects where activities require non-consumable resources of variable capacities. A conclusion can be drawn that in cases of relatively small projects (number of activities and number of different resources not exceeding 30÷40 and 3÷5, correspondingly) the newly developed optimal lookover algorithm is the most reasonable option in comparison with other heuristic algorithms. For large size projects, heuristic solutions may be preferable.
4. The presented resource constrained scheduling model is easy to handle; it can be implemented on a PC. Simulating a project of medium size with five different resources takes little computational time.
5. The results obtained are a further development of results outlined in §11.1, in which a resource constrained project scheduling model with fixed capacities has been presented.

§11.3 Stochastic network project scheduling under chance constraints

11.3.1 *Introduction*

It can be clearly recognized that both resource supportability models outlined in §§11.1-11.2 fit only certain project management scenarios. Those models do not include cost objectives, i.e., the costs of hiring and maintaining resources throughout the project's realization. The models do not deal with projects' due dates as well as with chance constraints of meeting the projects' deadlines on time. Those models can be used for one project only.

The research outlined below refers to a more generalized resource supportability model in project management.

Several simultaneously realized stochastic network projects of PERT type are considered. The durations of all projects' activities are random and the corresponding probability density functions are pregiven. Each activity requires various types of renewable resources with fixed capacities. Resources are stored and maintained at one central warehouse; each type of resources is in limited supply and is fixed at the same level throughout the projects' realization. Resources are to be hired and delivered to the central store before the moment the first project starts to be carried out. They are released at the moment when the last project is completed. Each activity starts at the moment when it is ready to be processed and when free available resources can support it. The cost of hiring and monitoring a resource unit per time unit (for each type of resources) is pregiven. Each project has its due date and the least permissible probability of accomplishing the project on time, i.e., its chance constraint. The problem is to determine:

- the earliest starting moment for each project's realization;
- the limited resource levels for each type of resources to be stored during the projects' realization;
- the moments that resources are fed in and projects' activities start, -

in order to minimize the average total expenses of hiring and maintaining resources subject to the chance constraints.

Thus, the developed resource supportability model covers a flexible project management system. The model minimizes the average operational expenses subject to the chance constraints, for each project separately [77-79].

The problem is solved by means of simulation. Two optimization cycles are imbedded in the model. The external cycle deals with optimizing both the projects' earliest starting moments together with the resource levels. Those parameters solve as the input values for the internal cycle. The latter uses heuristic decision-making rules to reallocate free available resources among the projects in order to meet the projects' chance constraints.

Note that models outlined in §§11.1-11.2 are based on solving knapsack resource reallocation problems which are applied at decision points when at least one activity is ready to be operated and there are free available resources. If, at a certain point of time, a set of more than one activity is ready to be operated but the available amount of resources is insufficient, a competition among the activities takes place in order to choose a subset of those activities which has to be operated first and can be supplied by the available resources. Determining such an optimal subset of activities is carried out by means of solving a knapsack problem. However, for several stochastic network projects the corresponding knapsack problem becomes too complicated. We have substituted it by a heuristic decision-making procedure. Note that the developed resource supportability model is a very complicated stochastic optimization problem which cannot be solved in the general case and allows a heuristic solution only.

11.3.2 Notation

Let us introduce the following terms:

$G_l(N,A)$ - the l-th stochastic network project (graph) of PERT type, $1 \le l \le m$;

m - the number of network projects;

$(i,j)_l \in G_l(N,A)$-the project's $G_l(N,A)$ activity;

t_{ijl} - random duration of activity $(i,j)_l$;

a_{ijl} - lower bound of value t_{ijl} (pregiven);

b_{ijl} - upper bound of value t_{ijl} (pregiven);

μ_{ijl} - the average value of t_{ijl};

V_{ijl} - variance of t_{ijl};

r_{ijlk} - capacity of the k-th type resource(s) allocated to activity $(i,j)_l$, $1 \le k \le n$ (pregiven);

n - number of different resources;

R_k - total available resources of type k to be hired and maintained throughout the planning horizon (to be determined);

D_l - the due date for project $G_l(N,A)$, $1 \le l \le m$ (pregiven);

p_l - the minimal admissible probability of meeting the due date of project $G_l(N,A)$ on time, $1 \le l \le m$ (pregiven);

S_l - the earliest starting moment for the project's $G_l(N,A)$ realization, $1 \le l \le m$, i.e., the earliest moment activities $(i,j)_l$ can start to be operated (to be determined);

S_0 - the moment for resources $\{R_k\}$ to be hired and delivered;

S_{ijl} - the moment that resources r_{ijlk} are fed in and activity $(i,j)_l$ starts (a random value conditioned on our decisions);

F_l - the actual moment project $G_l(N,A)$ is accomplished (a random value);

F_{ijl} - the actual moment activity $(i,j)_l$ is finished (a random value);

$R_k(t)$ - available resources of the k-th type at moment t; note that relation $R_k(0) = R_k$ holds, where 0 is the earliest moment when the first project starts to be carried out;

$G_{lt}(N,A)$ - the remaining unfinished project $G_l(N,A)$ at moment $t \ge 0$, $G_{l0}(N,A) = G_l(N,A)$, $1 \le l \le m$;

$T(G_l/S_{ijl})$ - random duration of project $G_l(N,A)$ on condition that feeding-in of resources r_{ijlk} carried out at moments S_{ijl};

$p(i,j)_l$ - conditional probability of activity $(i,j)_l$ to be on the critical path in the course of the project's $G_l(N,A)$ realization (dependent on the decisions already taken);

$R_k(t/S_{ijl})$ - maximal value of the k-th resource profile at moment t on condition that activities $(i,j)_l$ start at moments S_{ijl};

C_k - the cost of hiring and monitoring the k-th resource unit per time unit, $1 \le k \le n$ (pregiven);

$\overline{C}$ - the expected total resource expenses (to be minimized).

Each activity duration t_{ijl} follows either a normal probability density distribution with parameters (μ_{ijl}, V_{ijl}) or a uniform distribution in the interval (a_{ijl}, b_{ijl}), or a beta probability density function

$$f_{ijl}(x) = \frac{12}{(b_{ijl} - a_{ijl})^4}(x - a_{ijl})(b_{ijl} - x)^2 . \tag{11.3.1}$$

Note that obvious relations

$$\max_{i,j,l,k} r_{ijlk} \le R_k , \; 1 \le k \le n , \tag{11.3.2}$$

hold, otherwise the projects cannot be operated.

11.3.3 The problem

The problem is to determine values S_l, $1 \le l \le m$, and S_{ijl} to minimize the expected total resource expenses

$$C = \min_{\{S_l, R_k, S_{ijl}\}} E\left\{\left[\sum_{k=1}^{n}(R_k \cdot C_k)\right]\left(\max_l F_l - \min_l S_l\right)\right\} \tag{11.3.3}$$

subject to

$$\Pr\{F_l \le D_l\} \ge p_l , \; 1 \le l \le m , \tag{11.3.4}$$

$$R_k(t/S_{ijl}) \le R_k(t) \le R_k , \; \forall t \ge 0 , \; 1 \le k \le n . \tag{11.3.5}$$

Model (11.3.3-11.3.5) refers to a very complicated stochastic optimization problem which cannot be solved in the general case; the problem allows a heuristic solution only.

The basic idea of the heuristic solution is as follows. Two levels are incorporated in the model - the upper (external) level and the lower level. At the upper level an approximate search algorithm is implemented to determine the optimal values S_l and R_k, $1 \le k \le n$, $1 \le l \le m$. We will apply the cyclic coordinate descent method which is simple in usage and has been implemented for solving various production control and project management problems (see, e.g., [7,54,92,151,153]. Parameters $\{S_l, R_k\}$ serve as the input values for the lower level where values S_{ijl} are determined by means of simulation. Decision-making is carried out at essential moments F_{ijl} either:

- when one of the activities $(i,j)_l$ is finished and additional resources r_{ijlk}, $1 \le k \le n$, become available, or
- when all activities $(i,j)_l$ belonging to one and the same project $G_l(N,A)$ and leaving node i are ready to be processed, or
- when several subsets of activities ready to be processed belong to different projects.

If one or more activities are ready to be processed at moment t and all of them can be supplied with available resources, the required resources are fed in and the activities begin to be operated at moment t, i.e., $S_{ijl} = t$. If at least for one type k of resources there is a lack of available resources at moment t, a competition among the activities has to be arranged to choose a subset of activities that will start to be operated at moment t and can be supplied by resources. The general idea of decision-making, i.e., the sub-problem of choosing activities to be operated, will be outlined below.

11.3.4 Heuristic decision-making

Assume that at a certain moment t a set of activities $\{(i,j)_l\}$ is ready to be operated. Two cases will be considered:

A. Activities $\{(i,j)_l\}$ enter one and the same network graph (project) $G_l(N,A)$, i.e., for the set $\{(i,j)_l\}$ value l remains constant.

B. Activities $\{(i,j)_l\}$ refer to more than one network graph $G_l(N,A)$.

Let us examine both cases in greater detail.

Case A. To simplify the problem, cancel parameter l since the latter remains unchanged in the course of decision-making. Assume, with respect to §11.1, that at moment t q activities $(i_1,j_1),\ldots,(i_q,j_q)$, $q \geq 1$, are ready to be processed, and at least for one type k of resources there is a lack of available resources, i.e., relation

$$\sum_{\xi=1}^{q} r_{i_\xi j_\xi k} > R_k(t) \tag{11.3.6}$$

holds. Here r_{ijk} a simplified modification of r_{ijlk} for a fixed l (see *11.3.2*). A competition among the activities is arranged following the heuristic outlined in §11.1. According to that heuristic, the subset, which provides the maximal total contribution to the expected project duration subject to (11.3.6), has to be chosen. Each activity (i,j) contributes to the expected project duration value $\vartheta_{ij} = \mu_{ij} \cdot p(i,j)$, where $p(i,j)$, being a simplified version of $p(i,j)_l$, is the conditional probability for activity (i,j) to be on the critical path. At any decision point t values $p(i,j)$ are calculated by means of simulation (see §11.1). After determining values $p(i_\xi, j_\xi)$, $1 \leq \xi \leq q$, for all competitive activities at moment t, the optimal subset is chosen by solving a zero-one integer programming problem as follows: determine integer values $\eta_{i_\xi j_\xi}$, $1 \leq \xi \leq q$, to maximize the objective

$$\max_{\{\eta_{i_\xi j_\xi}\}} \left\{ \sum_{\xi=1}^{q} \left[\eta_{i_\xi j_\xi} \cdot \mu_{i_\xi j_\xi} \cdot p(i_\xi, j_\xi) \right] \right\} \tag{11.3.7}$$

subject to

$$\sum_{\xi=1}^{q} \left(\eta_{i_\xi j_\xi} \cdot r_{i_\xi j_\xi k} \right) \leq R_k(t), \quad 1 \leq k \leq n, \tag{11.3.8}$$

where

$$\eta_{i_\xi j_\xi} = \begin{cases} 0 & \textit{if activity } (i_\xi, j_\xi) \textit{ will not obtain resources;} \\ 1 & \textit{otherwise.} \end{cases} \tag{11.3.9}$$

Problem (11.3.7-11.3.9) is a classical zero-one integer programming problem, which provides a precise solution. However, the problem's parameters, e.g., $\vartheta_{i_\xi j_\xi}$, are obtained through heuristic assumptions.

Case B. This case makes unable decision-making (11.3.7-11.3.9) since the latter does not take into account at moment t different projects $G_{lt}(N,A)$ with different due dates D_l and different chance constraints p_l. Assume that at moment t a set of activities which are ready to be processed and which belong to υ different projects $G_{l_\rho t}(N,A)$, $1 \le \rho \le \upsilon$, is given. This set of activities can be subdivided into υ subsets $\left\{\left(i_{\xi\rho}, j_{\xi\rho}\right)_{l_\rho}\right\}$, $1 \le \xi \le q_\rho$, each subset of volume q_ρ entering the project $G_{l_\rho t}(N,A)$. Assume, that there is a lack of available resources, i.e., at least for one type k of resources relation

$$\sum_{\rho=1}^{\upsilon}\sum_{\xi=1}^{q_\rho} r_{i_{\xi\rho} j_{\xi\rho} l_\rho k} > R_k(t) \tag{11.3.10}$$

holds.

In order to undertake a reasonable decision-making, i.e., to choose a quasi-optimal subset of activities, we suggest a heuristic step-by-step procedure. The procedure is carried out as follows:

Step 1. For each project $G_{l_\rho t}(N,A)$ separately, reorder the activities entering the subset $\left\{\left(i_{\xi\rho}, j_{\xi\rho}\right)_{l_\rho}\right\}$ in the descending order of their corresponding values $\vartheta_{i_{\xi\rho} j_{\xi\rho}} = \mu_{i_{\xi\rho} j_{\xi\rho}} \cdot p\left(i_{\xi\rho}, j_{\xi\rho}\right)_{l_\rho}$.

Step 2. An assumption is introduced that:

- project $G_{l_1 t}$ will *not* obtain at moment t the required resources for any of the q_1 activities $\left\{\left(i_{\xi 1}, j_{\xi 1}\right)_{l_1}\right\}$ ready to be processed;
- the required resources will be fed in for *all* activities $\left\{\left(i_{\xi 1}, j_{\xi 1}\right)_{l_1}\right\}$ at the next decision moment t^*. Value t^* can be calculated as the *minimal* value of the average finishing times of all activities which at moment t undergo processing;
- in future, i.e., at all decision-points $t' > t$, all the remaining activities $(i,j)_{l_1}$ belonging to that projects will not wait for resources in lines until the end of the project's realization.

By means of simulation calculate the project's random duration $T\left(G_{l_1}/S_{ijl}\right)$ honoring the outlined above assumptions.

Step 3. Repeat Step 2 M times in order to obtain representative statistics. Call the simulated random finishing times for project $G_{l_1}(N,A)$: $F_{l_1}^{(1)}$, $F_{l_1}^{(2)}, \ldots, F_{l_1}^{(M)}$.

Step 4. Calculate the statistical frequency Q_{l_1} of completing project $G_{l_1}(N,A)$ on time:

$$Q_{l_1} = \frac{\sum_{\alpha=1}^{M} W_{l_1}^{(\alpha)}}{M}, \qquad (11.3.11)$$

where

$$W_{l_1}^{(\alpha)} = \begin{cases} 0 & if\ F_{l_1}^{(\alpha)} \le D_{l_1}; \\ 1 & otherwise. \end{cases} \qquad (11.3.12)$$

Step 5. Calculate the relative deviation

$$Z_{l_1} = \left(Q_{l_1} - p_{l_1}\right) \cdot \frac{1}{p_{l_1}}. \qquad (11.3.13)$$

Step 6. Repeat Steps 2 → 5 for all projects $G_{l_\rho}(N,A)$, $1 \le \rho \le \upsilon$, participating in the competition.

Step 7. Choose the project with the lowest value Z_ρ. Let it be $G_{l_\omega}(N,A)$, $1 \le \omega \le \upsilon$.

Step 8. For project $G_{l_\omega}(N,A)$, all the sorted activities $\left(i_{\xi\omega}, j_{\xi\omega}\right)_{l_\omega}$ (see Step 1) are examined one after another, in the descending order of their priorities, from top to bottom, to determine the first activity, which can be supplied with available resources. If, for such an activity $\left(i_{\xi\omega}, j_{\xi\omega}\right)_{l_\omega}$, $1 \le \xi \le q_\omega$, relations $r_{i_{\xi\omega} j_{\xi\omega} l_\omega k} \le R_k(t)$, $1 \le k \le n$, hold, the required resources are passed to the activity while the available resources $R_k(t)$ are updated,

$R_k(t) - r_{i_{\xi\omega} j_{\xi\omega} l_\omega k} \Rightarrow R_k(t)$, $1 \le k \le n$.

If such an activity can be determined, go to Step 10. Otherwise apply the next step.

Step 9. If no activity $\left(i_{\xi\omega}, j_{\xi\omega}\right)_{l_\omega}$ can be chosen on Step 8, examine the next project with the lowest value Z_ρ (besides Z_ω) in order to examine that project as well, etc., until a certain activity $\left(i_{\xi\rho}, j_{\xi\rho}\right)_{l_\rho}$ will be determined. If no activity can be found by examining all the projects, go to Step 11. Otherwise apply the next step.

Step 10. Exclude the determined activity from the set of competitive activities; update the available resources. Return to Step 1, i.e., carry out decision-making anew. It can be well-recognized that the procedure terminates either when all the available resources are reallocated among activities or all the competitive projects are examined in the order of their emergency parameters Z_ρ.

Step 11. Calculate the next decision point $t' > t$. Determine the set of activities ready to be operated. Return to Step 1.

A conclusion can be drawn that in Case B decision-making centers on choosing and operating first the activities which enter the "weakest" projects, i.e., the projects being late with meeting their corresponding due dates on time subject to their chance constraints. As to Case A, the project management operates first the optimal subset of activities that provides minimization to the expected project's duration.

11.3.5 The structure of the resource supportability model

The initial data of the model is as follows:

- *at the company level*: resource cost parameters C_k, $1 \le k \le n$;
- *at the project level*: due dates D_l and chance constraints p_l, $1 \le l \le m$;
- *at the activity level*: upper and lower bounds b_{ijl} and a_{ijl}, average values μ_{ijl}, resource capacities r_{ijlk}.

Decision variables R_k, $1 \le k \le n$, and S_l, $1 \le l \le m$, have to be determined *beforehand*, i.e., before the projects will actually start to be carried out. Note that moment S_0 resources R_k have to be hired, delivered and stored at the company's central warehouse satisfies $S_0 = \min_l S_l$ and coincides with the beginning of the projects' realization. However, certain projects may start to be carried out later that at moment S_0.

Thus, the resource supportability model is implemented at two stages:

- at the *planning stage*, i.e., before the projects' realization, when determining optimal planning parameters S_l and R_k, $1 \le l \le m$, $1 \le k \le n$. Those parameters are input values for the stage of monitoring which is performed in the course of the projects' realization;
- at the *stage of monitoring* the resource feeding-in moments S_{ijl} are determined. Those parameters *cannot be predetermined* since they are random values conditioned on our future decisions. At the stage of monitoring the resource supportability model can be implemented in real time; namely, all activities can be operated only after obtaining necessary resources. However, if we want to evaluate the efficiency of the resource supportability model, we can simulate the algorithm's work by random sampling of the actual duration of activities. By simulating the algorithm's work many times, all the projects' cost and probability parameters can be evaluated.

The structure of the resource supportability model and its algorithm is based on the assertion, that the cost objective $\overline{C}$ is a complicated non-linear function of decision variables S_l and R_k, $1 \le l \le m$, $1 \le k \le n$, and, by introducing the outlined above decision-making rules for Cases A and B, is *fully determined* by those decision variables. Thus, it is reasonable to arrange two optimization cycles for the model:

- the external cycle to carry out an optimal search for values $\{S_l\}$ and $\{R_k\}$ by applying the cyclic coordinate descent method, and

- the internal cycle to carry out mutual simulation runs of the projects' realization with input values $\{S_l\}$ and $\{R_k\}$ determined from the external cycles. It goes without saying that decision-making rules for both Cases A and B are incorporated in the simulation model at the internal cycle. At each simulation run objective $\overline{C}$ is calculated.

The combination $\{R_k, S_l\}$ which provides the minimal average objective $\overline{C}$ calculated by (11.3.3), subject to all chance constraints (11.3.4), is taken as the optimal combination which has to be predetermined before the projects' realization. Required resources $\{R_k\}$ are hired at the moment $S_0 = \min_l S_l$, after which the projects' realization actually starts. Feeding-in resource moments S_{ijl} are determined either for real-time projects, or by simulating the projects' realization.

11.3.6 <u>The heuristic algorithm</u>

The enlarged step-by-step procedure of the algorithm is as follows:

<u>Step 1</u>. Set the initial (minimal) values of $\{S_l\}$ and $\{R_k\}$. Note that $\{R_k\}$ are restricted from below:

$$R_k \geq \max_{i,j,l} r_{ijlk}, \quad 1 \leq k \leq n, \tag{11.3.14}$$

otherwise the problem has no solution. For most practical cases values S_l, $1 \leq l \leq m$, can be set equal zero. Thus, the optimal search method has to be arranged in the $(n+m)$-dimensional area. Denote the initial $(n+m)$-dimensional search point by $X^{(0)}$.

<u>Step 2</u>. Implement a cyclic coordinate search method with a positive search step increment Δt (or ΔR_k), beginning from the initial search point $X^{(0)}$. Undertaking a search means shifting one of the coordinates, beginning from S_l (the first group of m coordinates $\{S_l\}$ has to precede the second group $\{R_k\}$) to the right with step Δt or ΔR. If, e.g., from the search point $X^{(q)}$ the search $X^{(q)} \to X^{(q+1)}$ results in changing the η-th coordinate, $1 \leq \eta \leq n+m$, then all other coordinates remain unchanged. If in the course of a search step objective $\overline{C}$ becomes less than it has been before, at point $X^{(q)}$, the search proceeds in the same direction, i.e., an additional increment Δt (or ΔR_k) is implemented. If the objective does not decrease, then we examine the next, $(q+1)$-th coordinate, while all q preceding coordinates remain unchanged with the values they have already received. The routine iteration of the search terminates when all $(n+m)$ coordinates $\{S_l\}$ and $\{R_k\}$ are examined. Thus, each decision variable is *optimized separately*, while all the previous coordinates have already been optimized.

<u>Step 3</u>. At each routine search point $X^{(q)}$ with decision variables $\{S_l^{(q)}, R_k^{(q)}\}$, numerous simulation runs using the simulation model at the internal

cycle have to be undertaken to obtain representative statistics for value $\overline{C}$. The simulation model comprises three submodels as follows:

Submodel I simulates most of the procedures to be undertaken in the course of the projects' realization, namely:

- determines decision points (essential moments) F_{ijl};
- singles out (at a routine decision point) all activities that are ready to be operated;
- if possible, supplies all those activities with available resources and later on simulates the corresponding activities' durations;
- returns the utilized non-consumable resources to the company's central warehouse (at the moment an activity is finished);
- updates the remaining projects (if necessary) at each routine decision point.

Submodel II calculates by means of simulation values $p(i,j)$ to facilitate decision-making for the case of one project (Case A), as well as values $p(i,j)_l$ for the case of several projects (Case B). Submodel II also calculates the forecasted value t^* of the next adjacent decision point (see Step 2 of the decision-making model outlined in *11.3.4*). For each activity $p(i,j)_l$ which at moment t is being processed but has not been completed as yet, the average finishing time $\overline{F}_{ijl}$ is calculated. Given the starting time S_{ijl}, the probability density function of random value t_{ijl} and decision point t under consideration, a *precise* determination of value $\overline{F}_{ijl}$ can be obtained.

Note that simulation of activity durations by using Submodel II is carried out to solve auxiliary forecasting problems, but not to simulate actual activity realizations. The latter is carried out by Submodel I only.

Submodel III solves, at a routine decision point t, the zero-one integer programming problem (11.3.7-11.3.9) to undertake decision-making in the case of one project. Submodel III also simulates Steps 3-9 of the decision-making model in Case B of several projects (see *11.3.4*).

The outcome value of the simulation model at Step 3 is calculated as follows:

$$\overline{C} = \frac{1}{M}\sum_{\delta=1}^{M}\left\{\sum_{k-1}^{n}\left(R_k \cdot C_k\right)\left[\max_l F_l^{(\delta)} - S_0\right]\right\} + \sum_{l=1}^{m}\left(A \cdot X_l\right). \qquad (11.3.15)$$

Here A is an essentially high value (for numerical examples we usually set A equal 10^{17}), while X_l satisfies

$$X_l = \begin{cases} 1 & if \;\; Q_l \langle p_l \\ 0 & otherwise, \end{cases} \qquad (11.3.16)$$

where Q_l is calculated by (11.3.11) and $F_l^{(\delta)}$ is the simulated moment project $G_l(N,A)$ is finished in the δ-th simulation run, $1 \le \delta \le M$.

Thus, relations (11.3.15-11.3.16) enable undertaking search for routine $(m+n)$-dimensional points $X^{(q)}$ honoring chance constraints (11.3.4). If at least one value $X_l = 1$, the corresponding combination $X^{(q)} \equiv \{S_l, R_k\}$ is withdrawn from the cyclic coordinate search process.

Step 4. After optimizing all $m+n$ coordinates $\{S_l\}$ and $\{R_k\}$, i.e., carrying out a routine search iteration, the search process is initiated anew, beginning from the first coordinate S_l. The search process terminates when, for two adjacent iterations f and $f+1$, the relative difference between $\overline{C}^{(f)}$ and $\overline{C}^{(f+1)}$ is less than the pregiven accuracy $\varepsilon > 0$.

Extensive experimentation for medium size network projects has illustrated the efficiency of the developed two-level algorithm. Two iterations are usually enough to finalize the optimization process [151].

11.3.7 Conclusions

The following conclusions can be drawn from §11.3:

1. The developed resource supportability model can be used in project management as a decision support model for planning and monitoring several stochastic network projects. The model has been successfully used for small and medium size projects of PERT type.
2. The developed optimal planning parameters $\{S_l, R_k\}$ result in minimizing the resource average expenses for hiring and maintaining non-consumable resources. For a medium size network project with random activity durations, two cycle iterations resulted in a decrease of more than 50% in the initiated average expenses and were sufficient to finalize the optimization process.
3. The developed resource supportability model is suitable for resource scheduling in stochastic network projects, when the processing of certain activities is based on delivering resources, e.g., in high technology projects, defense related industries, opto-electronics, aerospace, etc.

§11.4 Resource constrained project scheduling model for alternative stochastic network projects

11.4.1 Introduction

In our previous §§11.1-11.3 we have outlined various algorithms in the area of resource constrained project scheduling. However, the regarded research deals with non-alternative network projects *only*, namely, of PERT type.

At the same time, it can be well-recognized that for a certain project its topology may implement various alternative outcomes (deterministic and stochastic), when there are several possible alternative ways for reaching the project's target. Such network projects usually occur when an entirely new device is designed with no similar prototypes in the past (e.g., in chemical industries, aerospace and in other defense related industries). They are faced with a great deal of uncertainty in their progress as well as with alternative outcomes in key events. Since the importance of such projects is significant, practically all industrial developed countries have to consider and to perform the so-called goal programs or goal projects as the basic trend of technological progress. The need for high quality resource constrained scheduling models for such complicated projects becomes more and more important. Thus, undertaking research in this area becomes imperative both from the theoretical and applied points of view.

The following resource constrained scheduling model for projects under random disturbances and with alternative structure is a methodological extension of our research results outlined in §§11.1-11.2, in which activity related resources with fixed and variable capacities have been imbedded in a PERT type network model without alternative branchings.

We will henceforth consider an activity-on-arc network project $G(N,A)$ of CAAN type outlined in Chapter 8, where the set of alternative nodes is subdivided into subsets:

- $\overline{\overline{N}} \subset N$: alternative nodes with stochastic branchings;
- $\overline{N} \subset N$: alternative deterministic nodes (decision nodes).

We have chosen the CAAN model since within the two recent decades it has been used in various main types of alternative network projects [51-57,92,151].

Each activity $(i,j) \in A \subset G(N,A)$ requires renewable resources of various types with fixed or variable capacities. In order to simplify the problem we will consider the case of fixed capacities, although introducing variable capacities results only in additional technical difficulties. Each type of resources is in limited supply with a resource limit that is fixed at the same level throughout the project duration. The duration of each activity is a random variable with given density function.

The problem is to determine starting time values S_{ij} for each activity (i,j) which will be actually processed in the course of the project's development. Note that due to the project's alternative structure, not all the activities entering the project will be carried out. Values S_{ij} are not calculated in advance and are random variables conditioned on the model's future decision. The model's objective is to minimize the expected project's duration. Such an objective is mostly used in project management (see, e.g., [143,156,165,etc.]), and the problem of decreasing the project duration is considered as one of the most important targets, especially for projects under random disturbances [109,143,156]. The suggested heuristic algorithm is implemented in real time by means of simulation. Decision-making in the course of monitoring the project is carried out:

- *at alternative deterministic decision nodes* to single out all alternative subnetworks (the so-called joint variants) in order to choose the one with the minimal average duration;
- *at the project's essential moments* when at least one activity is ready to be operated but the available amount of resources is limited. A competition among those activities is carried out to determine the subset of activities,

which have to be operated first and can be supplied by available resources. Such a competition is realized by a combination of a knapsack resource re-allocation model and a subsidiary simulation algorithm.

Note that those essential moments are as follows:

- when one of activities (i,j) is finished and additional resources become available, or
- when a certain event (node) i is realized and all activities leaving that node are ready to be processed.

Since a joint variant of a CAAN model is a GERT type sub-network with probabilistic outcomes in key events, the problem's solution is based on developing a resource constrained scheduling model for GERT projects. The corresponding algorithm [85] is, in essence, the backbone of the general resource constrained model, and a further development of the models outlined in Chapters 7-10. Thus, presenting the resource-constrained project scheduling model for networks with purely stochastic alternatives is the main contribution of §11.4.

There is no need to recall a description of the CAAN model since the latter has been outlined in depth in Chapter 8, together with the definitions of the joint variant and admissible plan. We will call henceforth AJV the CAAN algorithm for determining joint variants.

11.4.2 Notation

Let us introduce the following terms:

$G(N,A)$ - stochastic network project of CAAN type;

G_t - the remaining network project at moment $t \geq 0$; $G_0 = G(N,A)$;

$i(\overline{\alpha})$ - decision node with deterministic alternative outcomes;

$i(\overline{\overline{\alpha}})$ - alternative node with stochastic outcomes;

(i,j) - activity leaving node i and entering node j, $(i,j) \in A \subset G(N,A)$;

t_{ij} - random duration of activity (i,j), with density function $f_t(i,j)$;

a_{ij} - lower bound of value t_{ij} (pregiven);

b_{ij} - upper bound of value t_{ij} (pregiven);

μ_{ij} - average value of t_{ij} (pregiven);

$(i,j)^*$ - activity (i,j) which will be actually realized in the course of the project's development (conditioned on the model's decision). Note that since $G(N,A)$ is an alternative network, the set of actually realized activities $(i,j)^*$ is a subset of all activities $\{(i,j)\}$ entering $G(N,A)$. Thus, $(i,j)^* \subset \{(i,j)\}$;

S^*_{ij} - the moment resources are fed in and activity $(i,j)^*$ starts (a random value);

F^*_{ij} - the actual moment activity $(i,j)^*$ is finished, $F^*_{ij} = S^*_{ij} + t_{ij}$;

J_{rt} - the r-th joint variant of project G_t (a subnetwork of PERT or GERT type), $1 \le r \le m_t$;

m_t - number of joint variants in project G_t;

n - number of different resources;

$p(i,j)^*$ - conditional probability of activity $(i,j)^*$ to be on the critical path in the course
of the project's realization;

r_{ijk} - capacity of the k-th type resources allocated to activity (i,j), $1 \le k \le n$ (pregiven and fixed);

R_k - total available resources of type k at the project management disposal (pregiven and fixed throughout the planning horizon);

$R_k(t)$ - free available resources at moment $t \ge 0$;

$R_k^{\max}\left(t \middle| S_{ij}^*, J_{rt}^{opt}\right)$ - maximal value of the k-th resource profile at moment t on condition that activities $(i,j)^*$ start at moments S_{ij}^* and at moment t the optimal joint variant J_{rt}^{opt} is chosen;

$T\left(G \middle| S_{ij}^*, J_{rt}^{opt}\right)$ - random project's duration, on condition that according to the resource constrained scheduling model the optimal joint variant J_{rt}^{opt} will be chosen and all activities $(i,j)^*$ start at moments S_{ij}^*;

AJV - the algorithm for determining joint variants in the CAAN model.

11.4.3 The problem

The general resource constrained scheduling problem for a CAAN type model $G(N,A)$ is to minimize the expected project's duration

$$\min E\left\{T\left(G \middle| S_{ij}^*, J_{rt}^{opt}\right)\right\} \tag{11.4.1}$$

subject to

$$R_k^{\max}\left(t \middle| S_{ij}^*, J_{rt}^{opt}\right) \le R_k(t) \quad \forall t \ge 0, \quad 1 \le k \le n. \tag{11.4.2}$$

Problem (11.4.1-11.4.2) is a complicated stochastic optimization model for projects with an alternative structure and topology. The problem cannot be solved in the general case and allows a heuristic solution only.

The general idea of the heuristic algorithm is as follows. Decision-making is carried out in real time, at any routine essential moment t (decision point), either when one of the activities $(i,j)^*$ is finished and additional resources r_{ijk}, $1 \le k \le n$, become available, or when a certain non-alternative node i is realized and all activities leaving that node are ready to be processed, or when a decision node (a node with deterministic alternative outcomes of type α and γ) is reached. In the latter case, by using algorithm AJV (see Chapter 8), all joint variants J_{rt}, $1 \le r \le m_t$, are singled out and later on examined, to determine the optimal joint variant with the minimal expected duration. The procedure of determining the average duration of a joint variant (which is, in essence, a PERT or a GERT type network with purely stochastic alternative outcomes at certain nodes) is carried out by the resource constrained GERT project scheduling algorithm (RCGPS), which will be outlined below.

After examining all the joint variants J_{rt}, $1 \le r \le m_t$, the optimal joint variant J_t^* is chosen and future monitoring centers on carrying out resource constrained scheduling for pregiven total available resources $R_k(t)$, $1 \le k \le n$.

If a routine essential moment is a node with *stochastic alternative outcomes*, the latter are simulated according to their outcome probabilities (in real time projects as well); the simulated activity thus obtains its corresponding duration by means of simulation.

If an essential moment centers on determining a subset of activities (from a set of activities ready to be processed and waiting to be supplied by resources), a competition among the activities has to be arranged. For the case of a PERT network, the corresponding algorithm (call it henceforth RCPPS) is outlined above in §§11.1-11.3. The general idea of the RCPPS algorithm is to reallocate resources among the project's competitive activities on the basis of priority levels assigned to those activities. Those priority levels are the activities' contributions to the project's average duration. They depend both on the activity's average duration and on the probability to be on the critical path in the course of the project's realization. Those probability values are also determined by means of simulation.

The outlined below RCGPS algorithm is a modification of the RCPPS algorithm since stochastic alternative outcomes have to be taken into account.

After singling out the subset and supplying the later by available resources, activities begin to be processed. A new routine essential moment is determined, etc. until the project is accomplished.

Note, in conclusion, that in the course of developing a real project there may be changes in the parameters of some activities, e.g., probability density functions of the activities' durations, outcome probabilities, etc., since activity networks are revised over time. In such a case the problem of determining all the joint variants J_{rt} has to be resolved at each sequentially encountered decision node at moment t, since revising a project may result in changing its optimal joint variant J_t^{opt}. If the network does not undergo revision the problem has to be solved only once, at $t = 0$.

11.4.4 The general resource constrained scheduling heuristic algorithm for a CAAN type model

The outlined below algorithm incorporates two currently developed algorithms, namely:

- the algorithm of determining all joint variants from the initial CAAN model (algorithm AJV);
- the resource-constrained project scheduling algorithm RCPPS for non-alternative PERT networks for cases of fixed and variable resource capacities (see §§11.1-11.2).

The enlarged step-by-step procedure of the heuristic algorithm is as follows:

Step 1. The routine essential moment t of the project's progress is determined at the beginning of the project's realization. An essential moment may occur:

A. At a decision node ($\bar{\alpha}$) with alternative deterministic outcomes (of α and γ types).

B. At an alternative node ($\bar{\bar{\alpha}}$) with stochastic outcomes (of α and γ types).

C. At a non-alternative node (j) (of x and β types).

D. At the moment a certain activity (i,j) is finished, but event j is not realized as yet.

In Case A apply the next step; in Case B go to Step 8; in both Cases C and D apply Step 11.

Step 2. Determine the remaining network project G_t, $t \geq 0$. Note that

$$G_t \equiv G(N,A) \setminus \{(i,j)_t^*\} \setminus \{(i,j)_t^{**}\}, \tag{11.4.3}$$

where $\{(i,j)_t^*\}$ denotes the set of activities which have been already processed till moment t, and $\{(i,j)_t^{**}\}$ denotes the set of activities which have not been operated and, due to the alternative structure of $G(N,A)$ and prior decision-making, will not be carried out in the future.

Step 3. Apply algorithm AJV to single out all the joint variants J_{rt} from the subnetwork G_t. To apply the algorithm one has to implement sequentially four subalgorithms as follows:

- determining the $\bar{\alpha}$-frame for the outcome graph;
- determining the maximal path in the outcome graph;
- determining the admissible plans;
- determining the joint variants which correspond to admissible plans.

Let the determined joint variants be J_{rt}, $1 \leq r \leq m_t$ (see *11.4.2*).

Step 4. For each joint variant J_{rt} determine its average duration T_{rt} by using the resource constrained scheduling model either for GERT or PERT projects. The total pregiven available resource capacities are R_k, $1 \leq k \leq n$. For GERT projects, the regarded resource constrained project scheduling algorithm RCGPS to determine the project's average duration is outlined in *11.4.5*. For PERT projects, the corresponding algorithm (see §§11.1-11.2) enters the RCGPS as a basic part.

Step 5. Choose the joint variant $J_{\xi t}$ with the minimal average duration, i.e.,

$$T_{\xi t} = \min_{1 \leq r \leq m_t} T_{rt}. \tag{11.4.4}$$

Thus, joint variant $J_{\xi t}$ is considered as an optimal one, J_t^{opt}.

Step 6. Choose the outcome direction (activity) leaving node $\bar{\alpha}$ which corresponds to the chosen optimal joint variant J_t^{opt}. Let it be $(\bar{\alpha}, j)$.

Step 7. Cancel all other alternative outcome activities leaving node $\bar{\alpha}$.

Step 8. Determine all the nodes $i \in G_t$ with no activities *entering* those nodes. If such nodes exist, cancel them together with all activities *leaving* those nodes. Proceed carrying out Step 8, until only nodes with a receiver (except the source node) will remain. Go to Step 11.

Step 9. Applying this step means that we have reached an alternative node $\bar{\bar{\alpha}}$ with stochastic outcomes and corresponding probabilities. Simulate the set of full events in order to determine the outcome activity. Let it be $(\bar{\bar{\alpha}}, j)$.

Step 10. Cancel all other non-simulated outcome activities leaving node $\bar{\bar{\alpha}}$. Return to Step 8.

Step 11. Applying this step means that there may be activities ready to be processed at moment t, e.g.,

- activity $(\bar{\alpha}, j)$ (Step 6);
- activity $(\bar{\bar{\alpha}}, j)$ (Step 9);
- activity leaving node j (Case C, Step 1), etc.

At Step 11 in Case D (see Step 1), return the utilized resources r_{ijk}, $1 \le k \le n$, to the project management store.

Step 12. Determine the set of activities $(i_1, j_1), \ldots, (i_q, j_q)$, $q \ge 1$, which are ready to be processed at moment t, together with all available resources $R_k(t)$, $1 \le k \le n$.

Step 13. If all activities (i_υ, j_υ), $1 \le \upsilon \le q$, can be supplied by available resources, the required resources are fed-in and activities $\{(i_\upsilon, j_\upsilon)\}$ begin to be operated at moment t, i.e., $S_{i_\upsilon j_\upsilon} = t$, $1 \le \upsilon \le q$. If there is a lack of available resources, go to Step 15.

Step 14. Simulate (according to the density function) the durations of all activities which have been supplied with resources and started to be realized at moment t. Return to Step 1 to determine the next routine essential moment.

Step 15. Applying this step means that, due to limited amount of resources, a competition among activities (i_υ, j_υ), $1 \le \upsilon \le q$, has to be arranged in order to single out the subset of activities which can be supplied with resources and can start to be operated at moment t. The competition is facilitated by solving a knapsack resource reallocation problem to maximize the total contribution of the chosen activities to the average project's duration. For each activity under competition, its contribution is the product of the average duration of the activity and its probability of being on the critical path. Those probability values are calculated by means of simulation.

Since monitoring a CAAN type project results in monitoring a joint variant, i.e., a GERT type project, the algorithm outlined in §§11.1-11.2 requires modification. The amended resource constrained project scheduling algorithm for a GERT network model will be outlined in *11.4.5* (algorithm RCGPS).

After applying algorithm RCGPS and determining the subset of chosen activities proceed to Step 14.

The general algorithm terminates when the project will reach its target, i.e., when the remaining graph G_t becomes an empty set.

11.4.5 Resource constrained project scheduling algorithm for GERT models (RCGPS)

As outlined above, the RCGPS algorithm is a further development of the resource constrained project scheduling algorithm for PERT projects presented in §§11.1-11.3.

It is assumed that the project's network is properly enumerated, i.e., for all activities (i,j) entering the graph $G(N,A)$ relation $i < j$ holds. The enlarged step-by-step procedure of the algorithm is as follows:

Step 1. Similar to the general algorithm in *11.4.4*, the routine essential moment t is determined (for the monitored optimal joint variant J_t^{opt}). An essential moment occurs:

- at any alternative node ($\overline{\overline{\alpha}}$) with stochastic outcomes;
- at any non-alternative node (j);
- at the moment a certain activity (i,j) is finished, but event j is not realized as yet.

Step 2. The remaining monitored network project G_t for the previously chosen joint variant J_t^{opt} is determined. In Case A (see Step 1) apply the next step. In Cases B or C proceed to Step 6.

Step 3. Similar to Step 9 of the general algorithm, simulate the corresponding probabilistic outcome activity $(\overline{\overline{\alpha}},j)$.

Step 4. Is similar to Step 10 of the general algorithm (see *11.4.4*), and results in canceling all non-simulated alternative stochastic outcomes leaving node $\overline{\overline{\alpha}}$.

Step 5. Is similar to Step 8 of the general algorithm.

Step 6. Is similar to Step 11 of the general algorithm and results in returning the utilized resources r_{ijk}, $1 \le k \le n$, in Case C (see Step 1) to the project management store.

Steps 7-8. Steps 7 and 8 are similar to Steps 13 and 14 of the general algorithm, with one exception: in the case of lack of resources Step 10 is applied.

Step 9. In order to arrange the competition among the activities (i_υ, j_υ), $1 \le \upsilon \le q$, subnetwork G_t has to be simulated in order to be transformed to a PERT network. The simulation algorithm at Step 9 comprises the following operations:

9.1 From the set of stochastic alternative $\overline{\overline{\alpha}}$-nodes entering the remaining network project G_t before carrying out Step 9, determine the node with the minimal number (call it henceforth $\overline{\overline{\alpha}}_{\min}$). If the set of those nodes is empty, go to Step 10. Otherwise apply Substep 9.2.

9.2 Simulate the probability outcomes leaving node $\overline{\overline{\alpha}}_{\min}$ (similar to Step 3).

9.3 Cancel all non-chosen outcome activities leaving node $\overline{\overline{\alpha}}_{\min}$.

9.4 Determine all the nodes (alternative and non-alternative) with no activities entering those nodes. If such nodes exist, cancel them together with all activities leaving those nodes. Proceed with this procedure until only nodes with receivers will remain. Return to Substep 9.1.

Step 10. Simulate the durations of all remaining activities according to their density functions. Implementing that step means that we have simulated all non-contradictory alternative stochastic nodes (this is provided by introducing proper enumeration) and only nodes of x-type remain. Thus, simulating a GERT network at Step 9 results in obtaining a PERT network.

Step 11. Determine the critical path of the simulated network.

Step 12. Repeat the procedure of Steps 9-11 M times in order to obtain representative statistics.

Step 13. Calculate the frequency of each activity (i_υ, j_υ), $1 \le \upsilon \le q$, to be on the critical path. Denote them henceforth $p(i_\upsilon, j_\upsilon)$.

Step 14. In accordance with [70], determine the subset of chosen activities by solving a zero-one programming problem: determine integer values $\xi_{i_\upsilon j_\upsilon}$, $1 \le \upsilon \le q$, to maximize the objective

$$\max_{\{\xi_{i_\upsilon j_\upsilon}\}} \left\{ \sum_{\upsilon=1}^{q} \left[\xi_{i_\upsilon j_\upsilon} \cdot p(i_\upsilon, j_\upsilon) \cdot \mu_{i_\upsilon j_\upsilon} \right] \right\} \tag{11.4.5}$$

subject to

$$\sum_{\upsilon=1}^{q} \left(\xi_{i_\upsilon j_\upsilon} \cdot r_{i_\upsilon j_\upsilon k} \right) \le R_k(t), \quad 1 \le k \le n, \tag{11.4.6}$$

where

$$\xi_{i_\upsilon j_\upsilon} = \begin{cases} 0 & \textit{if activity } (i_\upsilon, j_\upsilon) \textit{ will not obtain resources}; \\ 1 & \textit{otherwise}. \end{cases}$$

Note that solving problem (11.4.5-11.4.6) results in implementing a heuristic approach to decrease the project's duration as much as possible [70]. Model (11.4.5-11.4.6) is, in essence, the backbone of the RCPPS algorithm, which has been successfully applied to many medium-size PERT projects [71].

From Step 14, return to Step 8 in order to simulate the durations of the chosen activities and, later on, to determine the next routine essential moment t.

Note that simulating activity durations at Step 10 is an auxiliary procedure (in order to determine probabilities $p(i_\upsilon, j_\upsilon)$ for problem (11.4.5-11.4.6)) while simulating activity durations at Step 8 is an *actual* activity realization.

The outlined above algorithm RCGPS is performed in real time: namely, all the activities can be operated only after obtaining necessary resources. However, if we want to evaluate the average project's duration T_{rt} for the set of joint variants J_{rt} (see Step 4 of the general algorithm), we can obtain a representative statistics by simulating each joint variant J_{rt} many times to determine its average duration. The minimal number of simulation runs can be estimated from the classical sampling theory [27], outlined in Chapter 3.

11.4.6 Experimentation

In order to verify the efficiency of the developed algorithm, extensive experimentation has been undertaken. A GERT project with constrained renewable resources is presented in [85]. The project requires resources of one type. The initial given data for each activity (i,j) entering the GERT model is as follows: i; j; a_{ij}; b_{ij}; r_{ijl}; p_{ij}, where p_{ij} denotes the probability of realizing activity (i,j). Thus, $p_{ij}=1$ means that node i is of x-type, while $0 < p_{ij} < 1$ corresponds to a stochastic alternative outcome, i.e., $i \equiv \overline{\overline{\alpha}}$.

Three alternative distributions are considered:

1. t_{ij} has a normal distribution in the interval $[a_{ij}, b_{ij}]$ with average $\mu_{ij} = 0.5(a_{ij} + b_{ij})$ and variance $V_{ij} = \frac{1}{36}(b_{ij} - a_{ij})^2$;
2. t_{ij} has a uniform distribution in the interval $[a_{ij}, b_{ij}]$;
3. t_{ij} has a beta distribution with the density function

$$p_{ij}(x) = \frac{12}{(b_{ij} - a_{ij})^4}(x - a_{ij})(b_{ij} - x)^2 . \qquad (11.4.7)$$

In order to check the developed RCJPS algorithm, 100 simulation runs were undertaken. The histograms for the three considered density functions are presented in [85].

The following conclusions can be drawn from §11.4:

1. Introducing the beta distribution results in projects with shorter durations in comparison to the normal and uniform distributions.
2. Introducing the normal distribution results in projects with shorter durations in comparison to the uniform distribution. Thus, the latter can be regarded as the least efficient distribution.
3. The heuristic algorithm presented here is, probably, the first one developed in the area of resource constrained project scheduling for alternative

stochastic network projects. It can be successfully used for monitoring complicated medium-size projects with alternative structure and topology, and with limited activity related renewable resources. The algorithm can be used for CAAN models which cover a broad spectrum of alternative stochastic networks.

4. Since a CAAN model is structured from subnetworks of GERT type, the developed resource constrained project scheduling algorithm is based on multiple implementation of a standardized resource constrained algorithm for GERT models. Such a basic algorithm is easy to apply and can be implemented on a PC. The algorithm can be used for any probability distribution of activity durations.

§11.5 Conclusions

The following conclusions can be drawn from the Chapter:

1. The outlined above resource supportability models do not comprise predetermined resource delivery schedules.
2. The models are implemented:
   - at the *planning stage*, when determining optimal planning parameters (the moments projects actually start and optimal resource capacities for each type of resources), and
   - at the *stage of monitoring*, i.e., at the stage of scheduling and feeding-in resources.
3. At the scheduling stage all calculated parameters are random values conditioned on our future decisions.
4. The backbone of all outlined in the Chapter resource supportability models is the classical zero-one integer programming model which for the case of restricted resources enables (at each decision-making moment) the optimal choice of activities to be supplied by resources.
5. Thus, the models presented in the Chapter can be regarded as mixed type models since they are utilized at several stages of the project's life cycle. Further models referring to the planning stage will be outlined in Chapter 13.

Chapter 12. Resource Constrained Project Scheduling with Deterministic Resource Delivery Model

§12.1 Case of aggregated projects with consecutive operations

12.1.1 Introduction

The method outlined below is a further development of our prior results related to job-shop manufacturing [69], and is probably the first successful attempt to implement a deterministic resource delivery schedule in stochastic network projecting. Various other attempts [2,21,31-32,97,142,147,etc.] do not deal with problems of scheduling operations of random durations.

Unlike the previous Chapter, several additional developments are imbedded in the model:

1. The outlined below reallocation model incorporates cost parameters rather than the time-related models presented in Chapter 11.
2. The model enables a group of projects with random operations and restricted resources to be controlled.
3. Several heuristic preference rules which enable redistribution of free available resources among the operations which are ready to start, are suggested.

The description of the system is as follows: several simultaneously realized projects under random disturbances are considered. Each project comprises numerous operations to be processed in a definite technological sequence. Each operation utilizes several non-consumable related resources with fixed capacities, e.g., machines or manpower. Each type of resource at the management's disposal is in limited supply, with a resource limit that is fixed at the same level throughout the projects' duration, i.e., until the last project is actually completed. For each operation, its duration is a random variable with given density function. Processing costs per time unit to hire and to utilize all the total available resources are pregiven. For any projects' operation, its planned start moment has to be determined. That means that an operation cannot start before the planned moment. If an operation starts processing *after* its planned moment, a pregiven cost penalty per time unit of the delay has to be paid by the management. A special service discipline which, if necessary, reallocates free available resources among operations ready to be carried out, is imbedded in the model.

The problem is to determine optimal planned start moments, in order to minimize total management expenses. In order to simplify the problem, we will assume that each project consists of a chain of consecutive operations. Each operation is characterized by a vector of resource capacities to carry out the operation and by the density function of the operation's random duration.

12.1.2 Notation

Let us introduce the following terms:

$O_{i\ell}$ - the ℓ-th operation of the i-th project, $1 \le i \le n$, $1 \le \ell \le m_i$;

n - the number of projects;

m_i - the number of operations entering the i-th project;

R_j - the total limit of the j-th type resources, $1 \le j \le k$ (at the disposal of the management, pregiven);

k - the number of resources;

$r_{i\ell j}$ - the j-th type resource capacity, $1 \le i \le n$, $1 \le \ell \le m_i$, $1 \le j \le k$, to carry out operation $O_{i\ell}$ (pregiven);

$a_{i\ell}$ - lower bound of random duration of $O_{i\ell}$ (pregiven);

$b_{i\ell}$ - upper bound of random duration of $O_{i\ell}$ (pregiven);

$T_{i\ell}$ - planned moment to start operation $O_{i\ell}$ (a deterministic value to be predetermined);

$S_{i\ell}$ - the moment $O_{i\ell}$ actually starts (a random value conditioned on our decisions);

$t_{i\ell}$ - time duration of operation $O_{i\ell}$ (a random value);

$\bar{t}_{i\ell}$ - average value of $t_{i\ell}$ (pregiven);

$C_{i\ell}$ - cost penalty per time unit for the delay in starting operation $O_{i\ell}$, in case $S_{i\ell} > T_{i\ell}$ (pregiven);

C - processing cost per time unit of hiring and utilizing total resources $\{R_j\}$, $1 \le j \le k$ (pregiven);

T - random time duration of accomplishing all the projects;

$F_{i\ell} = S_{i\ell} + t_{i\ell}$ - actual finishing time of operation $O_{i\ell}$ (random value);

$R_j(t)$ - free (non-utilized) resources at moment t;

Δt_i - positive search step for the operations entering the i-th project, $1 \le i \le n$, (pregiven);

$\varepsilon > 0$ - pregiven search accuracy for the cyclic coordinate method.

It can be well recognized that relation

$$T = \max_i \{F_{im_i}\} - \min_i \{S_{i1}\} \tag{12.1.1}$$

holds.

Note, in conclusion, that evident relations

$$R_j \ge \max_{1 \le i \le n, 1 \le \ell \le m_i} r_{i\ell j}, \quad 1 \le j \le k, \tag{12.1.2}$$

hold, otherwise not all the projects can be carried out.

12.1.3 The problem's formulation

The general problem is to determine both optimal *deterministic* planned values $T_{i\ell}$ (beforehand) and *random* values $S_{i\ell}$ (in the course of the projects' realization and conditioned on our decisions), $1 \le i \le n$, $1 \le \ell \le m_i$, to minimize the average of the total expenses

$$\min_{\{T_{i\ell}, S_{i\ell}\}} E\left[\sum_{i=1}^{n}\left\{\sum_{\ell=1}^{m_i}\left[(S_{i\ell} - T_{i\ell})\cdot C_{i\ell}\right] + C\cdot\left[\max_i(F_{im_i}) - \min_i(S_{i1})\right]\right\}\right] \tag{12.1.3}$$

subject to

$$S_{i\ell} \geq T_{i\ell}, \tag{12.1.4}$$

$$F_{i\ell} = S_{i\ell} + t_{i\ell}, \tag{12.1.5}$$

$$S_{i\ell} \geq S_{i,\ell-1} + t_{i,\ell-1}, \tag{12.1.6}$$

$$\sum_{i=1}^{n}\sum_{\ell=1}^{m_i}\left[r_{i\ell j} \cdot \delta_{i\ell}(t)\right] = R_j - R_j(t), \quad 1 \leq j \leq k, \quad 0 \leq t \leq T, \tag{12.1.7}$$

where $$\delta_{i\ell}(t) = \begin{cases} 1 & \text{if } O_{i\ell} \text{ is processed at moment } t; \\ 0 & \text{otherwise}. \end{cases} \tag{12.1.8}$$

Restriction (12.1.4) ensures that $O_{i\ell}$ cannot start before its planned moment $T_{i\ell}$. Restriction (12.1.5) enables processing $O_{i\ell}$ without interruptions, while (12.1.6) formalizes the consecutive chain order of processing operations in a project. Restriction (12.1.7) means that at any moment t, $0 \leq t \leq T$, for each j-th type of resource, $1 \leq j \leq k$, the summarized amount of utilized resources is less than R_j by the value of free resources $R_j(t)$.

Problem (12.1.3-12.1.8) is a stochastic optimization problem with a large number of optimized variables. The problem is too difficult to solve in the general case. A heuristic solution will be outlined below.

The general approach to solving the problem is as follows: two levels - upper and lower - are considered. At the upper level, a cyclic search by means of a coordinate descent method [74,114] is organized to determine planned start moments $T_{i\ell}$. At each search point, values $\{T_{i\ell}\}$ are passed to the lower level in order to manage the projects by determining actual start moments $S_{i\ell}$ in the course of the projects' realization. This results in developing a simulation model which comprises proper decision-making to carry out a simulation run. Decision-making is based on implementing heuristic decision rules that can be regarded as the service discipline [126]. By repeating, for a fixed vector $\{T_{i\ell}\}$, the simulation procedure many times, we obtain representative statistics to evaluate the average with pregiven accuracy. The set $\{T_{i\ell}\}$, which delivers the minimum to objective (12.1.3), is taken as a quasi-optimal solution. Note that since heuristic decision-making is introduced we shall avoid the term "optimal" from now on.

Note, in conclusion, that the suggested decision-making can be applied to real-time projects as well.

12.1.4 Decision-making in the simulation model

The basic idea of imbedding decision-making in the simulation model is as follows: decision-making is carried out at *essential* moments t when either an operation $O_{i\ell}$ is accomplished and the utilized resources $\{r_{i\ell j}\}$ become free and available for other operations, or when a certain operation $O_{i\ell}$ at moment $t = T_{i\ell}$ is ready to be processed. At each essential moment t, the simulation model:

- returns the utilized non-consumable resources to the management store and evaluates the free available resources $R_j(t)$, $1 \le j \le k$ (in the case when, at moment t, an operation is finished);

- singles out all the operations that are ready to be processed;
- checks the possibility of supplying all those chosen operations with available resources. If this is possible, the required resources are delivered to the operations which start processing at moment t. Later on, the corresponding operations' durations are simulated;
- determines the next routine decision point (essential moment) t.

If it is impossible to provide all the operations with free resources, a competition has to be arranged to single out a subset of operations to be processed and supplied with available resources. Note, that carrying out that competition is, in essence, the decision-making which is imbedded in the model.

Assume that at moment t, q different operations $O_{i_1\ell_1}, O_{i_2\ell_2}, ..., O_{i_q\ell_q}$ are ready to be processed and at least one relation

$$\sum_{r=1}^{q} r_{i_r\ell_r j} > R_j(t), \quad j \in \{1,k\}, \tag{12.1.9}$$

holds.

Decision-making A is a random version of the priority rule SRT ("shortest remaining time") outlined in [50,53,126]. The competitive operations are sorted in the ascending order of their average processing durations $\bar{t}_{i_s\ell_s}$. All the sorted operations are examined one after another, in the ascending order of values $\bar{t}_{i_s\ell_s}$, to check, for each operation, the possibility that it can be supplied with remaining available resources. If, for a certain operation $O_{i_s\ell_s}$, $1 \le s \le q$, relations $r_{i_s\ell_s j} \le R_j(t)$, $1 \le j \le k$, hold, the required resources are assigned to the operation while the remaining resources are updated,

$$R_j(t) - r_{i_r\ell_r j} \Rightarrow R_j(t), \quad 1 \le j \le k, \tag{12.1.10}$$

and the next operation is examined. If a routine operation $O_{i_s\ell_s}$, cannot be provided with available resources, we switch over to the next operation. The procedure terminates either when all the available resources are allocated to operations or all the q chosen operations have been examined.

Decision-making B is a random version of the equally famous priority rule LRT ("longest remaining time" [50,53,126]). All competitive operations are sorted in the descending order of their average remaining processing times

$$\bar{T}_{i_s\ell_s} = \sum_{r=\ell_s}^{m_{i_s}} \bar{t}_{i_s r}, \quad 1 \le s \le q. \tag{12.1.11}$$

All the sorted operations are examined in the descending order of values $\bar{T}_{i_s\ell_s}$, one after another, similar to the procedure outlined above.

Note that additional and not less effective priority rules can be recommended as well, e.g., the pairwise comparison rule [50,53,118,151], the FIFO rule [126], etc.

Introducing proper decision-making enables the projects' realization to be simulated from beginning to the end, i.e., enables average E in (12.1.3) to be calculated on the basis of numerous simulation runs. Note that a simulation run can be carried out only with preset planned values $\{T_{i\ell}\}$.

12.1.5 The heuristic algorithm

The heuristic search algorithm to determine quasi-optimal values $\{T_{i\ell}\}$ is performed in real time, i.e., all operations can be processed only after obtaining necessary resources. Essential moments cannot be predetermined. However, if we want to evaluate the efficiency of the algorithm, we can simulate its work many times, including the cyclic coordinate descent subalgorithm at the upper level and the simulation model with set values $\{T_{i\ell}\}$ at the down level. Thus, the heuristic algorithm comprises two subalgorithms as follows:

Subalgorithm I actually implements the cyclic search procedure, similar to that outlined in §11.3. At the beginning of the search, values $T_{i\ell}$ are as follows:

$$t_{i1} = 0\,,\;\; T_{i\ell} = \sum_{r=1}^{\ell-1} a_{ir}\,,\;\; 1 \le r \le m_i\,,\;\; 1 \le i \le n\,. \qquad (12.1.12)$$

Later on, each coordinate $T_{i\ell}$ has to undertake search steps of length Δt_i.

Several *concepts* are embedded in the subalgorithm:

1. If a routine coordinate $T_{i\ell}$ changes its value in the course of a search procedure, all the preceding values T_{sq}, $1 \le s \le i-1$, $1 \le q \le m_s$, together with values T_{iq}, $q < \ell$, which have been determined before, are fixed and remain unchanged.
2. If a routine coordinate $T_{i\ell}$, $1 \le \ell < m_i$, increases its value by Δt_i, all the next values T_{is}, $\ell < s \le m_i$, entering the same i-th project, are automatically increased by Δt_i. If coordinate $T_{i\ell}$ decreases its value by Δt_i, all consecutive coordinates T_{is} are decreased as well. Values T_{sq}, $i < s \le n$, $1 \le q \le m_s$, remain unchanged.
3. A routine coordinate $T_{i\ell}$ increases in the course of the search procedure, if realizing the previous step brings about a decrease of the objective value (12.1.3), i.e., the average total expenses. Otherwise, an opposite search step with values $(-\Delta t_i)$ has to be carried out honoring Concept 2.
4. After a routine coordinate $T_{i\ell}$ ceases to change its value in the course of the search, the value is fixed and remains unchanged until all the coordinates undergo the search procedure. The next coordinate $T_{i,\ell+1}$ (in case $\ell < m_i$) or $T_{i+1,\ell}$ (in case $\ell = m_i$ and $i < n$) is processed by the search algorithm. Thus, changes in a single routine coordinate $T_{i\ell}$ by means of the search procedure implementing the cyclic coordinate descent method which operates cyclically with respect to all coordinates, enable a quasi-optimal solution to be obtained.

5. After carrying out the search procedure through all the coordinates, the process is then repeated, again starting with coordinate T_{11} (the next iteration). However, the search increments Δt_i, $1 \le i \le n$, have to be diminished (usually by dividing by two). Another difference from the first iteration is that a search procedure for any coordinate $T_{i\ell}$ has to be carried out in two opposite directions $T_{i\ell} \pm \Delta_i$. The search is undertaken along that direction which delivers a decrease in the problem's objective in (12.1.3). Note, that in the course of implementing the q-th iteration, $q > 1$, Concept 2 becomes unnecessary. Only the evident relation $T_{i\ell} \ge T_{i,\ell-1} + a_{i,\ell-1}$ has to be honored in all cases.
6. The search terminates if, in the course of carrying out two adjacent iterations, the relative difference between the two corresponding objective values (12.1.3) becomes less than the pregiven accuracy ε.

Subalgorithm II realizes the simulation model for each search point, i.e., for each fixed vector $\{T_{i\ell}\}$. At each search point, a representative sample of simulation runs has to be carried out. On the basis of the sample, the average value in objective (12.1.3) is calculated. An illustration of a simulation run with fixed $\{T_{i\ell}\}$ will be outlined below.

12.1.6 Numerical example

The system comprises two projects with given planned start moments $T_{i\ell}$. The projects include two and three consecutive operations, correspondingly, with pregiven random time durations. Both projects utilize one type of resource. Thus, $n = 2$, $m_1 = 2$, $m_2 = 3$ and $k = 1$.

The projects' parameters are as follows:

$a_{11} = 31;$ $b_{11} = 40;$ $a_{12} = 48;$ $b_{12} = 55;$
$a_{21} = 30;$ $b_{21} = 38;$ $a_{22} = 18;$ $b_{22} = 30;$
$a_{23} = 28;$ $b_{23} = 39;$
$r_{111} = 15;$ $r_{121} = 17;$ $r_{211} = 15;$ $r_{221} = 20;$
$r_{231} = 27.$

All $C_{i\ell} = 40$ while value $C = 100$. The total limit of resources $R_1 = 30$. Pregiven values $T_{i\ell}$ are as follows:

$T_{11} = 0;$ $T_{12} = 33;$ $T_{21} = 33;$ $T_{22} = 34;$ $T_{23} = 58.$

Assume that all random time durations $t_{i\ell}$ are normally distributed with average $\mu_{i\ell} = 0.5(a_{i\ell} + b_{i\ell})$ and variance $V_{i\ell} = \frac{1}{36}(b_{i\ell} - a_{i\ell})^2$.

In the course of carrying out a simulation run, we will use decision-making B. Assume that in the course of project realization, the simulated random values $t^*_{i\ell}$ are as follows:

$t^*_{11} = 34.7;$ $t^*_{12} = 52.8;$ $t^*_{21} = 36.2;$ $t^*_{22} = 24.3;$ $t^*_{23} = 35.6.$

At the first essential moment $t_1 = 0$, resources have to be redistributed among two competitive operations O_{11} and O_{21}. Since their corresponding planned moments $T_{11} = T_{21} = 0$, both those operations are ready to be processed. Note that decision-making based on the longest average remaining time (Rule B) gives preference to Project 2 at moment $t = 0$ since relation

$$0.5 \cdot (30+38) + 0.5 \cdot (18+30) + 0.5 \cdot (28+39) = 91.5 > 0.5 \cdot (31+40) + 0.5 \cdot (48+55) = 87$$

holds. But since the free available resources are enough to supply both operations ($R_1(0) = 30 = r_{111} + r_{211}$), the latter begin to be processed at moment $t = 0$; thus, $S_{11} = S_{21} = 0$, and $F_{11} = 34.7$, $F_{21} = 36.2$.

The second essential moment $t_2 = \min\{F_{11}, F_{21}\} = 34.7$. At moment $t_2 = 34.7$, operation O_{11} terminates and released resources $r_{111} = 15$ are returned to the management. However, since $r_{121} = 17 > 15$, operation O_{21}, being ready to be processed ($t_2 = 34.7 > T_{21} = 33$), has to wait for additional resources. The next essential moment would be $t_3 = 36.2$.

At moment $t_3 = 36.2$, operation O_{21} terminates and the total value of free available resources $R_1(36.2) = 30$. However, two competitive operations, namely O_{12} and O_{22}, are ready to be processed ($T_{12} = 33 < 36.2$, $T_{22} = 34 < 36.2$). Those two operations cannot be provided simultaneously with available resources since $R_1(36.2) = 30 < r_{121} + r_{221} = 37$ holds. Thus, decision-making based on Rule B has to be introduced. Due to relation

$$0.5 \cdot (48+55) = 51.5 < 0.5 \cdot (18+30) + 0.5 \cdot (28+39) = 57.5,$$

operation O_{22} has to be preferred. Thus, $S_{22} = 36.2$ and $F_{22} = 36.2 + 24.3 = 60.5$, while operation O_{12} has to wait for resources. The next essential moment would be $t_4 = 60.5$.

At moment $t_4 = 60.5$ operation O_{22} terminates, and $r_{221} = 20$ resources are released. It can be clearly recognized that operations O_{12} and O_{23} are both ready to be processed ($T_{12} < 60.5, T_{23} < 60.5$), but cannot be simultaneously supplied with free available resources: $R_1(60.5) = 30 < r_{121} + r_{231} = 44$. Thus, a competition based on decision-making by implementing Rule B has to be introduced. Due to

$$0.5 \cdot (48+55) = 51.5 > 0.5 \cdot (28+39) = 33.5,$$

the preference is given to O_{12}. Thus, $S_{12} = 60.5$ and $F_{12} = 60.5 + 52.8 = 113.3$. The next essential moment is $t_5 = 113.3$.

At $t_5 = 113.3$, only one operation, namely, O_{23}, is ready to be processed ($t = 113.3 > T_{23} = 58$) and is waiting for resources. Since $R_1(113.3) = 30 > r_{231} = 27$, operation O_{23} starts processing, and $S_{23} = 113.3$, while $F_{23} = 113.3 + 34.7 = 148$. At moment $t = 148$, both projects are accomplished.

Let us calculate the projects' expenses within the simulation run. The cost penalties are $C_{12} \cdot (36.2 - 33) + C_{22} \cdot (60.5 - 34) + C_{23} \cdot (113.3 - 58) = 40 \cdot (3.2 + 26.5 + 55.3) = 3{,}400$. The processing costs are $C \cdot (148 - 0) = 14{,}800$. Thus, the total expenses for one simulation run equal $3{,}400 + 14{,}800 = 18{,}200$.

12.1.7 *Conclusions*

The comparative efficiency of decision-making Rules A and B, together with three alternative distributions (uniform, normal and beta) of values $t_{i\ell}$, can be illustrated by extensive experimentation outlined in [53,74,151]. The following conclusions can be drawn from examining the results of solving the general problem (12.1.3-12.1.8):

1. For the case of uniform and normal distributions, using decision Rule B results in obtaining lower average expenses of realizing the projects than by using Rule A. In the case of beta-distribution, using Rule A is preferable, i.e., it results in cheaper project realization.
2. Using Rule A usually results in an essential decrease of the average value $\overline{T}$ in the course of implementing the search procedure, as distinct from decision Rule B. Introducing the latter does not lead to diminishing value $\overline{T}$. Moreover, in several cases, final values $\overline{T}$ became even higher than at the initial search point $T_{i\ell}^{0}$.
3. It can be well-recognized that *for all examples*, solving problem (12.1.3-12.1.8) results in an essential decrease of the average *penalty* expenses (in most cases by the factor of 50÷100). This, in turn, has an influence on the average *total* expenses (12.1.3) to be minimized. In the course of the search procedure's realization, the objective value has been diminished by a half.

§12.2 Resource constrained model for a variety of non-consumable resources

12.2.1 *Introduction*

This Section considers a certain elaboration of the research outlined in §12.1, namely:

a) the simplified aggregated projects are substituted by PERT type projects;

b) various classes of resources are incorporated in the model.

The main goal of this Section is to outline a generalized resource constrained model for a network project under random disturbances. All particular cases of utilizing renewable resources will be imbedded in the model. The problem's solution results in:

- *determining in advance*, i.e., before the project starts to be realized, a *deterministic delivery schedule* for extremely costly and rare external resources which are not at the project's disposal. Note that due to random disturbances, it is unknown beforehand when a certain activity will actually be ready to begin. However, the resources are to be delivered at a pregiven date that must be determined in advance. It goes without saying that an activity cannot start before its corresponding planned moment, i.e., when activity resources are ready and delivered. In practice such a resource delivery schedule is required for a relatively small group of activities;

- determining both the starting times and the resource capacities to be utilized for activities which require limited renewable resources which are at the project's disposal. The corresponding feeding-in resource schedule is random and is conditioned on control actions to reallocate available resources among the activities which at a certain moment are ready to be processed.

The problem's objective is a cost value which comprises two components:

- the average cost penalties paid for the idle costly and rare resource which was delivered at the planned moment, but not utilized since it had to wait for the moment the corresponding activity was ready to be operated;
- the average cost expenses for hiring and maintaining non-consumable resources which are at the project's disposal. Those expenses depend on the project's duration.

Thus, we suggest using a cost objective to minimize the sum of the penalty expenses for all delays of resource utilization and the cost of using constrained resources within the period of the project's realization. Note that decreasing the first component results in increasing the second one, and vice versa. Therefore, a trade-off between the two contradictive cost components is to be resolved.

12.2.2 Classification of non-consumable resources in project management

Non-consumable (renewable) resources used in projects can be classified in several ways (see, e.g., [149]). Referring to [69,149], we will describe the approach based on resource availability.

1. The so-called non-constrained resources (C-category or C-resources) are available in unlimited quantities for a cost throughout the project realization (e.g., unskilled labor or general purpose equipment). C-resources do not require monitoring although they might be expensive and might contribute to the cost-effectiveness of the project. However, using those resources does not result in changing the efficiency of any control policy in project management. That is why C-resources will not be taken henceforth into account to outline the generalized cost-optimization problem.
2. Resources of the second class (B-category or B-resources) are usually in limited supply for each type of resource. A resource limit may be either independent on time, i.e., is fixed at the same level throughout the project's duration, or the limit is a function of time. Various B-resources, e.g., skilled workers, special equipment, etc., for projects under random disturbances require flexible, but not close, monitoring. Since each activity entering the project is of random duration, the corresponding feeding-in resource moment to be determined is a random value too. The delivery schedule for constrained B-resources is not determined in advance, since

the delivery moments coincide with the feeding-in moments and are conditioned on decision-making in the course of the project's realization.

3. Extremely expensive and rare resources (A-resources) are usually external and available for short periods within the time span of the project (e.g., technical experts, test-benches, special and unique facilities, heavy duty equipment and cranes, etc.) A-resources should be strictly monitored because shortages might significantly affect the project schedule. Although it is unknown in advance when a certain activity which utilizes A-resources will actually be ready to begin, A-resources have to be delivered at a pregiven date that has to be determined in advance. Thus, for activities which utilize A-resources, a deterministic schedule of delivering resources is to be predetermined before the project starts to be realized. Both A-and B-resources will be imbedded in the developed resource constrained project scheduling model under random disturbances.

12.2.3 Notation

Let us introduce the following terms:

$G(N,A)$ - PERT type project (a network graph with random activity durations);

(i,j) - activity entering $G(N,A)$;

t_{ij} - duration of (i,j) (a random value);

μ_{ij} - average value of t_{ij};

a_{ij} - lower bound of t_{ij} (pregiven);

b_{ij} - upper bound of t_{ij} (pregiven);

$f_{ij}(t)$ - density function of t_{ij} (pregiven);

$T(i)$ - time moment event (node) i is realized, i.e., the earliest moment when all activities entering i are completed (a random value);

n - the number of activities entering the project;

(i_{ξ_A}, j_{ξ_A}) - activity which utilizes A-resources, $1 \le \xi \le n_A < n$;

n_A - number of activities which have to be supplied with A-resources;

(i_{η_B}, j_{η_B}) - activity which utilizes B-resources, $1 \le \eta \le n_B < n$;

n_B - number of activities which have to be supplied with B-resources;

m - number of different B-resources;

R_k - total capacity of the k-th type B-resources at the disposal of the project management, $1 \le k \le m$;

$r_{i_{\eta_B} j_{\eta_B} k}$ - capacity of the k-th type B-resources to be utilized by activity (i_{η_B}, j_{η_B}) (pregiven for the case of fixed capacities and an optimized variable for the case of variable capacities);

$r^{\max}_{i_{\eta_B} j_{\eta_B} k}$ - the maximal capacity of the k-th type B-resources to process activity (i_{η_B}, j_{η_B}) (pregiven for the case of variable capacities);

$r^{\min}_{i_{\eta_B} j_{\eta_B} k}$ - the minimal capacity of the k-th type B-resources to process activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ (pregiven for the case of variable capacities);

S_{ij} - time moment activity (i,j) actually starts (a random value conditioned on our decisions);

$F_{ij} = S_{ij} + t_{ij}$ - time moment activity (i,j) is completed (a random value);

$T\left(i_{\xi_A}, j_{\xi_A}\right)$ - time moment A-resources have to be delivered to process activity $\left(i_{\xi_A}, j_{\xi_A}\right)$ (a deterministic schedule to be determined in advance);

$c\left(i_{\xi_A}, j_{\xi_A}\right)$ - cost penalty to be paid by the management per time unit of the A-resources idling, i.e., $S\left(i_{\xi_A}, j_{\xi_A}\right) - T\left(i_{\xi_A}, j_{\xi_A}\right)$, which is the difference between delivering and feeding-in A-resources (pregiven);

c_B - cost of hiring and maintaining total $\{R_k\}$ B-resources per time unit throughout the project's realization (pregiven);

C - non-operational project's expenses which comprise cost penalties for idle A-resources and the cost of hiring and maintaining B-resources during the project's realization (a random value);

$R_k(t) \le R_k$ - free (available for utilization) k-th type B-resources, at moment $t > 0$;

h_ξ - the value of the search step of the ξ-th coordinate (pregiven);

$\varepsilon > 0$ - pregiven search accuracy for the cyclic coordinate method;

$D(i)$ - the subset of nodes, which *directly precede* node i, i.e., $i^* \in D(i)$ results in $(i^*, i) \in A \subset G(N, A)$;

p - a probability value close to zero which practically enables determination of the distribution's lower bound by calculating the sample's p-quantile W_p (p pregiven);

$W_p\{T(i)\}$ - the p-quantile of the random value $T(i)$, $i \in N \subset G(N, A)$, with p close to zero, i.e., a value of $T(i)$ which practically cannot be diminished;

$W_p\left\{T(i) \big/ T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}$, $1 \le \xi \le q \le n_A$ - the conditional p-quantile of the random value $T(i)$, on condition that for certain q A-resource activities $\left(i_{\xi_A}, j_{\xi_A}\right)$ their corresponding resource delivery moments $T\left(i_{\xi_A}, j_{\xi_A}\right)$ are fixed and are deterministic values.

12.2.4 <u>The problem</u>

The general resource constrained scheduling problem for a stochastic network project is to determine:

- a deterministic resource delivery schedule $T\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \le \xi \le n_A$, for supplying A-resources (to be determined in advance), and
- actual starting times S_{ij} for all activities $(i,j) \in G(N, A)$

- to minimize the average value of the non-operational expenses

$$J=\min_{\{S_{ij}\},\{T(i_{\xi_A},j_{\xi_A})\}} E\{C\}=\min E\left\{\sum_{\xi=1}^{n_A}\left[\left(S_{i_{\xi_A}j_{\xi_A}}-T(i_{\xi_A},j_{\xi_A})\right)\cdot c(i_{\xi_A},j_{\xi_A})\right]+c_B\cdot\left[\max_{(i,j)}F_{ij}-\min_{(i,j)}S_{ij}\right]\right\}, \quad (12.2.1)$$

subject to

$$S_{i_{\xi_A}j_{\xi_A}}\ge T(i_{\xi_A},j_{\xi_A}),\ \ 1\le\xi\le n_A, \quad (12.2.2)$$

$$S_{ij}\ge T(i)\ \ \forall(i,j)\in G(N,A), \quad (12.2.3)$$

$$t=S_{i_{\eta B}j_{\eta B}}\Rightarrow\sum_{\eta=1}^{q} r_{i_{\eta B}j_{\eta B}k}\le R_k(t),\ \ 1\le\eta\le q\le n_B,\ \ 1\le k\le m. \quad (12.2.4)$$

Note that if some project activities require B-resources *with variable capacities*, value $r_{i_{\eta B}j_{\eta B}k}$ becomes an *additional optimized variable*. An additional restriction

$$r^{\min}_{i_{\eta B}j_{\eta B}k}\le r_{i_{\eta B}j_{\eta B}k}\le r^{\max}_{i_{\eta B}j_{\eta B}k},\ \ 1\le\eta\le q\le n_B,\ \ 1\le k\le m, \quad (12.2.5)$$

is to be imbedded in the resource constrained project scheduling model.

Neither the costs of utilizing C-resources, nor the operational costs of processing project activities are implemented in cost objective (12.2.1). This is done deliberately since all those expenses remain unchanged and do not depend on the control model.

Restriction (12.2.2) means that an activity which utilizes A-resources, cannot start before its corresponding delivery moment. Restriction (12.2.3) means that any activity (i,j) entering $G(N,A)$, cannot start before the moment $T(i)$, i.e., that

$$S_{ij}\ge\max_{i^*\in D(i)}\left\{S_{i^*i}+t_{i^*i}\right\} \quad (12.2.6)$$

holds. Restriction (12.2.4) means that if at a certain decision point t B-resources are reallocated among $q<n_B$ activities, the summarized value of supplied resources (for each k-th type of B-resources) should not exceed the corresponding value $R_k(t)$, i.e., the total capacity of free available k-th type resources at moment t, $1\le k\le m$.

Problem (12.2.1-12.2.5) is a complicated stochastic optimization problem, which cannot be solved by applying non-heuristic algorithms.

The problem's solution is, in essence, a unification of a deterministic resource delivery schedule and a random schedule of activities' starting times obtained by using decision-making during the project's realization. We suggest solving the problem by means of simulation, in combination with a cyclic coordinate search algorithm (see §§11.3, 12.1) and a heuristic resource reallocation algorithm based on numerous applications of the knapsack resource reallocation problem outlined in Chapter 11.

It can be well recognized that decreasing values $T(i_{\xi_A},j_{\xi_A})$ results both:

- in increasing the first additive in objective (12.2.1), i.e., the cost penalties $\sum_{\xi=1}^{n_A}\left[\left(S_{i_{\xi_A}j_{\xi_A}}-T(i_{\xi_A},j_{\xi_A})\right)\cdot c(i_{\xi_A},j_{\xi_A})\right]$ for idle A-resources within the delay interval $\left[T_{ij},S_{ij}\right]$, and

- in decreasing the second additive, namely, the cost of hiring and utilizing B-resources subject to decreasing the project's duration, and vice versa. Thus, a trade-off between both components is to be resolved to minimize the cost objective (12.2.1).

Note, in addition, that due to (12.2.2) and (12.2.3), an evident relation

$$T\left(i_{\xi_A}, j_{\xi_A}\right) \geq T\left(i_{\xi_A}\right), \quad 1 \leq \xi \leq n_A, \qquad (12.2.7)$$

holds, otherwise the A-resource idleness becomes pointless. Thus, (12.2.7) can be regarded as the lower bound for the deterministic resource delivery schedule to be predetermined.

12.2.5 <u>The problem's solution</u>

It can be clearly recognized that determining a feasible schedule of delivering A-resources, i.e., setting values $T\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \leq \xi \leq n_A$, enables reducing problem (12.2.1-12.2.4) to another problem, namely, to a modified version of a resource constrained scheduling problem for stochastic network projects which has been considered in [70-71]. The problem is to reallocate constrained B-resources among the project's activities which utilize those resources, to minimize the average project's duration. Resource reallocation, i.e., feeding-in B-resources, is carried out in every decision moment t when not less than one activity $\left(i_{\eta_B}, j_{\eta_B}\right)$, $1 \leq \eta \leq q \leq n_B$, is ready to be processed but the available resources are limited. Thus, decision-making centers on singling out optimal subsets of $q^* < q$ activities which are supplied with resources. It is suggested to solve problems (12.2.1-12.2.4) or (12.2.1-12.2.5) by implementing a heuristic algorithm which comprises, in turn, two subalgorithms.

Subalgorithm I implements a cyclic coordinate descent search method in an n_A- dimensional space of optimized variables $T\left(i_{\xi_A}, j_{\xi_A}\right)$. In order to carry out the search by avoiding pointless steps, a subsidiary simulation model is introduced to calculate non-conditional and conditional $W_p\{T(i)\}$ and $W_p\left\{T(i) / T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}$ in order to refine lower bounds (12.2.7) at each search point.

Subalgorithm II calculates for each feasible resource delivery schedule $\left\{T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}$ obtained from Subalgorithm I, the average objective value (12.2.l) by simulating the project's realization with limited B-resources. The number of simulation runs should enable obtaining representative statistics. Decision-making in order to reallocate limited B-resources among the corresponding project's activities is imbedded in this Subalgorithm as well.

Schedule $T^*\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \leq \xi \leq n_A$, which delivers the minimum to the average objective value (12.2.1), is taken as the optimal solution. Note, that values $T^*\left(i_{\xi_A}, j_{\xi_A}\right)$ are determined in advance, i.e., before the project starts. In the course of the project's realization the optimized resource delivery schedule $T^*\left(i_{\xi_A}, j_{\xi_A}\right)$

remains fixed and unchanged, while all starting values S_{ij} are determined by means of Subalgorithm II. The structure of the heuristic algorithm is presented in Fig. 12.l.

12.2.6 <u>Subsidiary simulation models</u>

In order to carry out Subalgorithm I, two subsidiary simulation models SMl and SM2 are implemented in the model. Assume that the graph $G(N,A)$ is properly enumerated, i.e., for each activity (i,j) relation $i<j$ holds. Assume further, that a lexicographical order is introduced for all activities (i,j) entering the project. Activity (i_1,j_1) precedes lexicographically activity (i_2,j_2) if either $i_1<i_2$ or both $i_1=i_2$, $j_1<j_2$ hold.

Model SMI undertakes numerous simulation runs to calculate the W_p-quantile for values $T(i)$, $i\in N\subset G(N,A)$. At each simulation run the random value $T(i)$ is calculated by

$$\begin{cases} T(i)=\max\limits_{i^*\in D(i)}\left\{S_{i^*i}+t_{i^*i}\right\}, \quad 1<i\le n, \\ S_{ij}=T(i), \ 1<i\le n-1, \ 1<j\le n, \\ T(1)=0. \end{cases} \tag{12.2.8}$$

Note that since graph $G(N,A)$ is properly enumerated, using recurrent relations (12.2.8) in the course of a simulation run makes utilization of SMI very simple.

For a representative sample obtained by means of SMl value $W_p\{T(i)\}$ is calculated by using the classical sample theory (see, e.g., [27]). Simulation model SM2 calculates conditional values $W_p\left\{T(i)\big/T\left(i_{\xi_A},j_{\xi_A}\right)\right\}$, $1\le\xi\le q\le n_A$ and differs from SMI by using recurrent relations

$$\begin{cases} T(i)=\max\limits_{i^*\in D(i)}\left\{S_{i^*i}+t_{i^*i}\right\}, & 1<i\le n, \\ S_{ij}=\begin{cases} T(i) \ \ if \ \ (i,j)\notin\left\{i_{\xi_A},j_{\xi_A}\right\}, & 1\le\xi\le q, \\ \max\left\{T(i),T\left(i_{\xi_A},j_{\xi_A}\right)\right\} \ if \ (i,j)\in\left\{i_{\xi_A},j_{\xi_A}\right\}. \end{cases} \end{cases} \tag{12.2.9}$$

Note, that models SMl and SM2 differ from each other only by implementing different recurrent relations (12.2.8) and (12.2.9), correspondingly, in the course of a simulation run. Both models will be used henceforth in Subalgorithms I and II.

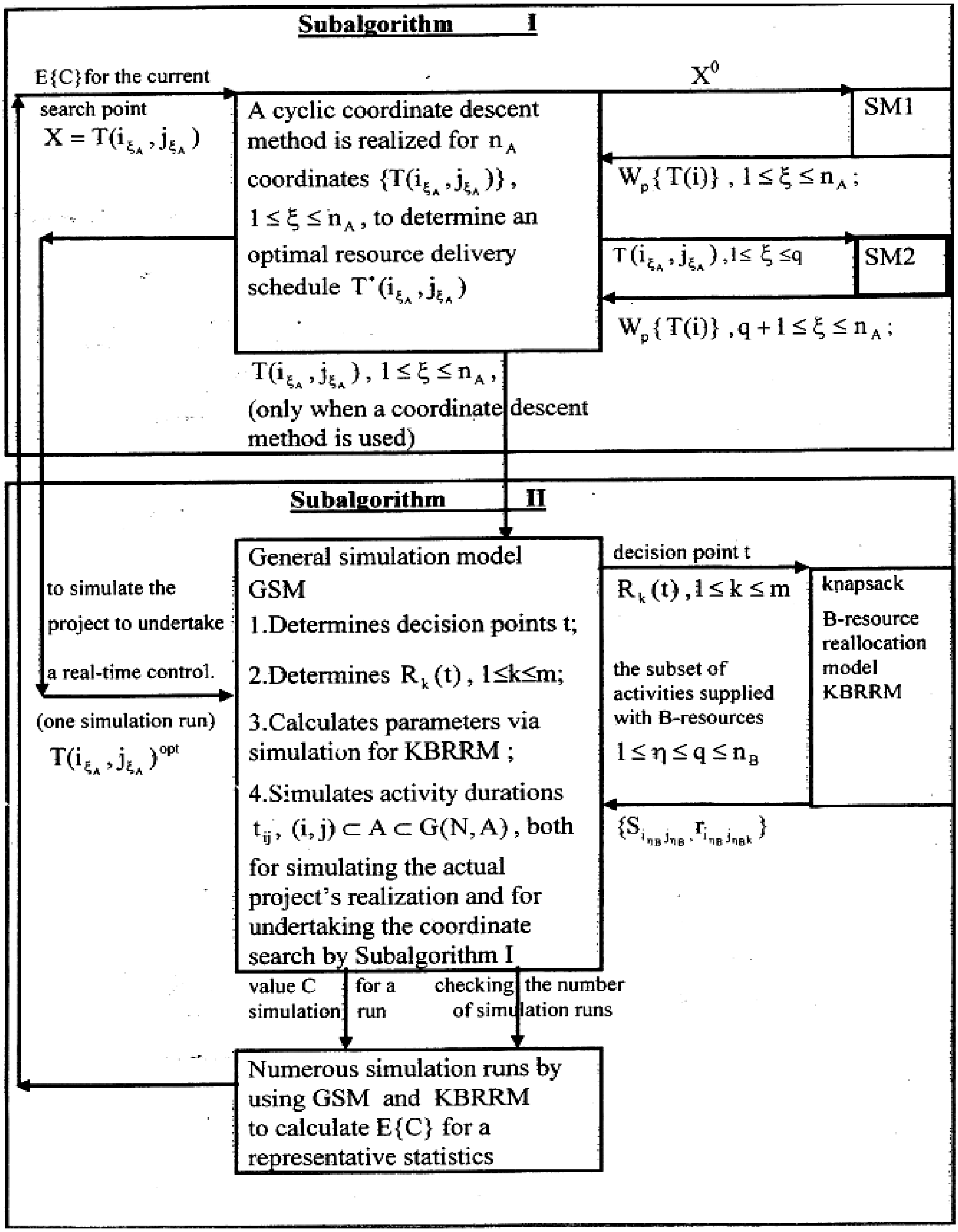

Figure 12.1. <u>The structure of the general resource constrained project scheduling algorithm</u>

12.2.7 *A cyclic coordinate descent subalgorithm to determine A-resource delivery schedules*

Coordinate descent methods are preferred [53-54,114,118,151] because of their easy implementation for cases when the objective is a complicated non-linear function of optimized variables. Since setting n_A deterministic values $T^*\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \le \xi \le n_A$, results in obtaining a non-variable objective value $E\{C\}$, the latter can be regarded as a function $f\left(x_1, x_2, ..., x_{n_A}\right)$ of n_A coordinates $x_\xi \equiv T\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \le \xi \le n_A$. Given a point $\overline{X} = \left\{x_1, x_2, ..., x_{n_A}\right\} \equiv \left\{T\left(i_{1_A}, j_{1_A}\right), ..., T\left(i_{n_A}, j_{n_A}\right)\right\}$, descent with respect to the coordinate x_ξ (ξ fixed) means that one solves $\min_{x_\xi} f\left(x_1, x_2, ..., x_\xi, ..., x_{n_A}\right)$.

Thus changes in the single component x_ξ result in seeking a new and better vector descent in the direction x_ξ or $(-x_\xi)$, where x_ξ is the ξ-th component. By sequential minimizing with regard to different components, a minimum of f might ultimately be determined.

We chose the cyclic coordinate descent algorithm which minimizes f cyclically with respect to coordinate variables. Thus, x_1 is changed first, then x_2 with fixed x_1, and so forth through x_{n_A}. The process is then repeated starting with x_1 again (second iteration), until the relative difference between two adjacent iterations $E\{C\}^{(v)}$ and $E\{C\}^{(v+1)}$ becomes less than the pregiven tolerance $\varepsilon > 0$.

Note that in the course of changing the ξ-th coordinate, $T\left(i_{\xi_A}, j_{\xi_A}\right)$, with the fixed first $(\xi - 1)$ coordinates $T\left(i_{1_A}, j_{1_A}\right), ..., T\left(i_{\xi-1_A}, j_{\xi-1_A}\right)$, all the next coordinates $T\left(i_{\xi+1_A}, j_{\xi+1_A}\right), ..., T\left(i_{n_A}, j_{n_A}\right)$ have to be updated by using simulation models SM1 and SM2. As outlined above, the cyclic coordinate descent method is imbedded in Subalgorithm I.

The extended step-by-step procedure of Subalgorithm I is as follows:

<u>Step 1</u>. Determine the initial search point $X^0 = T\left(i_{\xi_A}, j_{\xi_A}\right)^0$, $1 \le \xi \le n_A$, by means of simulation model SMI by determining lower bounds of value x_ξ satisfying

$$T\left(i_{\xi_A}, j_{\xi_A}\right)^0 = W_p\{T(i)\}, \quad 1 \le \xi \le n_A, \tag{12.2.10}$$

where all $T(i)$, $i \in N \subset G(N, A)$, satisfy (12.2.8).

<u>Step 2</u>. Apply Subalgorithm II to determine the objective $E\{C\}$ for X^0. Denote the obtained value by $\overline{C}^0$.

<u>Step 3</u>. Set counter $v = 1$ for the number of cyclic iterations. Note that $E\{C\}^{(v=0)} = \overline{C}^0$.

<u>Step 4</u>. Set counter $\xi = 1$ for the number of the current coordinate.

<u>Step 5</u>. For each current coordinate $x_\xi = T\left(i_{\xi_A}, j_{\xi_A}\right)$ arrange a local search with a pregiven search increment h_ξ, $x_\xi \pm h_\xi$, where values $x_1, x_2, ..., x_{\xi-1}$ remain fixed and unchanged, while lower bounds of values $x_{\xi+1}, x_{\xi+2}, ..., x_{n_A}$ are determined by using simulation model SM2, namely, by

$$x_q = T\left(i_{q_A}, j_{q_A}\right) = W_p\left\{T\left(i_{q_A}\right) / T\left(i_{s_A}, j_{s_A}\right)\right\}, \quad 1 \le s \le \xi, \quad \xi + 1 \le q \le n_A. \tag{12.2.11}$$

Note that in the course of carrying out the first iteration the search algorithm uses only positive increments, i.e., only search of type

$x_\xi + h_\xi \Rightarrow x_\xi$ is implemented. However, for next iterations, when increments h_ξ, $1 \le \xi \le n_A$, are usually diminished, both directions $x_\xi \pm h_\xi \Rightarrow x_\xi$ may be examined.

Step 6. Apply Subalgorithm II to determine the objective value for the new search point $X = (x_1, x_2, \dots, x_{\xi-1}, x_\xi \pm h_\xi, x_{\xi+1}, \dots, x_{n_A})$. Denote the average value $E\{C\}$ obtained by means of Subalgorithm II, by $\overline{C}(x_\xi \pm h_\xi)$.

Step 7. If $\overline{C}(x_\xi \pm h_\xi) < C(x_\xi)$, undertake a new search step for the coordinate value x_ξ along the direction of the objective's decrease. Return to Step 6. Otherwise, if $\overline{C}(x_\xi \pm h_\xi) \ge C(x_\xi)$, change the search to the opposite direction and apply Step 6 again. If $\overline{C}(x_\xi)$ cannot be decreased by undertaking a search for coordinate x_ξ, apply the next step.

Step 8. Counter ξ works, $\xi + 1 \Rightarrow \xi$.

Step 9. Check inequality $\xi > n_A$. If yes, proceed to the next step. Otherwise return to Step 5.

Step 10. Applying this step means that we have undertaken a local search for all coordinates h_ξ, $1 \le \xi \le n_A$, separately. Denote the final value of objective (12.2.1) by $\overline{C}^{(v)}$ where v is the current number of the cyclic iteration.

Step 11. Calculate the relative closeness $k^{(v)}$ between two adjacent v-th and $(v-1)$-th iterations, $k^{(v)} = \dfrac{\overline{C}^{(v-1)} - \overline{C}^{(v)}}{\overline{C}^{(v)}}$. If $k^{(v)} \ge \varepsilon$, apply the next step. Otherwise go to Step 14.

Step 12. Counter v works, $v + 1 \Rightarrow v$.

Step 13. Diminish values h_ξ, $1 \le \xi \le n_A$, (they are usually subdivided by 2), and return to Step 4.

Step 14. Subalgorithm I terminates and the results obtained at the last, v-th iteration, are taken as the optimal, i.e.,

$$\left\{T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}^{opt} = \left\{T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}^{(v)}. \qquad (12.2.12)$$

Note, in conclusion, that all steps of Subalgorithm I are carried out *in advance*, i.e., before the project starts. *After determining the optimal resource delivery schedule (12.2.12) we apply Subalgorithm II only once, in the course of the actual project's realization* (see Fig. 12.1).

12.2.8 Subalgorithm II to simulate the project's realization by supplying constrained B-resources

The general idea of reallocating renewable constrained resources among the project activities has been outlined in Chapter 11 for stochastic network projects.

Subalgorithm II comprises two important models (see Fig. 12.1):

- the knapsack resource constrained reallocation model to allocate B-resources among the project activities at decision points (see block KBRRN on Fig. 12.1), and

- the general simulation model (GSM) to simulate the project's realization.

The knapsack resource reallocation problem is realized at the so-called decision points t when at least one activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ utilizing B-resources is ready to be operated but the available amount of resources is limited. A competition among the activities has to be carried out in order to choose those activities which can be supplied with resources and which have to be operated first. Assume that at a certain moment t, $q < n_B$, activities $\left(i_{\eta_B}, j_{\eta_B}\right)$, $1 \le \eta \le q$, $q > 1$, are ready to be processed, but at least for one type of resources there is a lack of available B-resources.

In case of fixed B-resource capacities $r_{i_{\eta B} j_{\eta B} k}$, $1 \le k \le m$, we suggest (see §11.1) to solve the zero-one integer programming problem by determining zero-one integer values ρ_η, $1 \le \eta \le q$, to maximize the objective

$$\max_{\{\rho_\eta\}} \left\{ \sum_{\eta=1}^{q} \left[\rho_\eta \cdot p\left(i_{\eta_B}, j_{\eta_B}\right) \cdot \mu_{i_{\eta_B} j_{\eta_B}} \right] \right\} \tag{12.2.13}$$

subject to

$$\sum_{\eta=1}^{q} \left[\rho_\eta \cdot r_{i_{\eta_B} j_{\eta_B} k} \right] \le R_k(t), \quad 1 \le k \le m, \tag{12.2.14}$$

where $p\left(i_{\eta_B}, j_{\eta_B}\right)$ is the probability for activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ to be on the critical path in the course of a simulation run, and

$$\rho_\eta = \begin{cases} 0 & \text{if activity } \left(i_{\eta_B}, j_{\eta_B}\right) \text{ is provided with resources;} \\ 1 & \text{otherwise.} \end{cases} \tag{12.2.15}$$

Thus, product $W_\eta = p\left(i_{\eta_B}, j_{\eta_B}\right) \cdot \mu_{i_{\eta_B} j_{\eta_B}}$ represents the value contributed by activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ to the expected project's duration. The subset of activities which when supplied with resources, results in *minimizing* the project's duration, has to be chosen. All activities entering that subset start operating at moment t.

Problem (12.2.13-12.2.15) is solved by a zero-one integer programming algorithm with a precise solution (see, e.g., [153]). Values $p\left(i_{\eta_B}, j_{\eta_B}\right)$ are determined by means of simulation, by using (12.2.9) and taking into account values $\left\{T\left(i_{\xi_A}, j_{\xi_A}\right)\right\}$ obtained from Subalgorithm I.

In case of *variable* B-resources capacities $r_{i_{\eta B} j_{\eta B} k}$, $1 \le k \le m$, activity duration depends on resource capacities to be allocated to that activity. A more complicated problem has been solved in order to perform the optimal choice at decision points t (see §§11.1-11.2). We solve a knapsack resource reallocation problem as follows:

- to determine ρ_η and $r_{i_{\eta B} j_{\eta B} k}$, $1 \le k \le m$, $1 \le \eta \le q$, to maximize the objective

$$J = \underset{\{\rho_\eta\},\{r_{i_{\eta B} j_{\eta B} k}\}}{Max} \left\{ \sum_{\eta=1}^{q} \left[\rho_\eta \cdot p\left(i_{\eta_B}, j_{\eta_B}\right) \cdot \sum_{k=1}^{m} \left[r_{i_{\eta B} j_{\eta B} k} \cdot \psi\left(i_{\eta_B}, j_{\eta_B}, k\right) \right] \right] \right\} \tag{12.2.16}$$

subject to

$$r^{\min}_{i_{\eta_B} j_{\eta_B} k} \le r_{i_{\eta_B} j_{\eta_B} k} \le r^{\max}_{i_{\eta_B} j_{\eta_B} k} , \tag{12.2.17}$$

$$\sum_{\eta=1}^{q} \left(\rho_\eta \cdot r_{i_{\eta_B} j_{\eta_B} k} \right) \le R_k(t) \quad \forall t \ge 0 , \; 1 \le k \le m , \tag{12.2.18}$$

where ρ_η satisfies (12.2.15). Here value $\psi\left(i_{\eta_B}, j_{\eta_B}, k\right)$ ensures the optimal choice which results in minimizing the project's duration. Problem (12.2.15-12.2.18) is solved by means of heuristics similarly to outline in §11.2, as well as by a look-over algorithm providing a precise solution.

The GSM model:

a) determines decision points t to reallocate B-resources;

b) singles out activities which are ready to be processed;

c) reallocates B-resources among activities on the basis of solving problems (12.2.13-12.2.15) or (12.2.15-12.2.18);

d) supplies activities with resources and determines values $R_k(t)$;

e) simulates the actual time durations for activities which have been supplied with A- or B-resources;

f) returns utilized B-resources to the project's store at the moment an activity was completed;

g) calculates values W_η for the knapsack reallocation problems (12.2.13-12.2.15) or (12.2.15-12.2.18) at decision points t;

h) determines for activities utilizing A-resources their starting moments by using relation (12.2.9).

Thus, Subalgorithm II is used both

- for forecasting purposes to optimize the A-resources delivery schedule, i.e., before the project starts, and
- in the course of the project's realization (see Fig. 12.l), on the basis of the optimized schedule $T^*\left(i_{\xi_A}, j_{\xi_A}\right)$.

12.2.9 *Experimentation*

In order to check the efficiency of the presented resource constrained algorithm, extensive experimentation has been undertaken. Various stochastic network projects comprising 30÷50 activities have been examined [53,93-94], each of them including 3÷5 activities utilizing A-resources, with all other activities requiring B-resources. Numerous combinations of cost parameter values, as well as three alternative probabilistic distribution laws (uniform, normal and beta) have been considered.

The following conclusions can be drawn from the experimentation:

1. The uniform distribution is the most expensive to realize the project, while the normal distribution proves to be the cheapest one.
2. The cyclic coordinate algorithm for determining resource delivery moments $T\left(i_{\xi_A}, j_{\xi_A}\right)$ requires only two iterations to carry out the optimization process. The decrease of expenses between the second iteration and the

initial search point $T(i,j)^0$ showed approximately 45% cost improvement for the uniform distribution, 58% for the beta-distribution and 67% for the normal distribution. Thus, it can be well recognized that the coordinate search algorithm performs well.
3. The comparative efficiency of the normal distribution can be illustrated by analyzing the results of the first iteration. Implementing the latter results in decreasing the average cost expenses $\overline{C}$ by 66% for the normal distribution versus only 23% for the uniform distribution and 29% for the beta-distribution.
4. The outlined resource constrained project scheduling model can be applied to a broad variety of stochastic network projects which utilize various types of renewable resources, e.g., R&D projects.
5. The developed scheduling model covers both rare and expensive resources which require strict monitoring and can be delivered from outside for short periods only together with various types of renewable limited resources at the project management's disposal. Those resources do not require close monitoring. The model provides both a deterministic resource delivery schedule for the rare and expensive resources and undertakes resource reallocation to obtain a feeding-in schedule for the second type of resources.
6. Unlike the model outlined in §12.1 which is not aimed at PERT projects, the model under consideration covers practically all types of stochastic network projects independently of their structure.
7. The main drawback of the model considered in §12.2 is the absence of chance constraint restrictions. The required model's refinement will be outlined below.

§12.3 A generalized resource project scheduling model for several PERT projects under chance constraints

12.3.1 Introduction

In the preceding §12.2 we have outlined a resource supportability model which deals with two different types of renewable resources to be consumed by the project's activities:

- rare and costly resources (A-resources) which have to be delivered from outside for a relatively small group of project activities;
- restricted renewable resources which are feed in at random moments when the resources are available and at least one project activity has to be supported with resources in order to start processing (B-resources). Those resources are in limited supply at the project's disposal throughout the planning horizon.

We have assumed before that the total B-resource capacities for the project management store are fixed and pregiven externally. However, since the cost of hiring and maintaining B-resources is an essential part of the total expenses in the course of the project's realization, the problem of determining the optimal restricted B-resource capacity limits is reasonable for many projects' scenarios.

Minimizing the total project's expenses to meet the target on time, i.e., at a given due date, has not to be the only project management's goal in the course of a long-term cooperation with various customers. To honor the company's good name, an additional requirement has to be implemented in the model: the project has to meet its due date on time with a pregiven confidence probability. Thus, a chance constraint has to be introduced in the resource constrained model.

The cost objective for all models outlined in the previous chapters was to minimize the budget for the resource consumption within the planning horizon. However, it would be reasonable to also take into account additional factors connected with the project's total expenses within the planning horizon, e.g.:

- the starting time of the project's realization, which refers to the optimized variables as well;
- various penalty costs for not meeting the project's target on time and storage costs for the project's completion before the due date.

Thus, developing u generalized resource supportability model under a chance constraint and comprising all the additional parameters outlined above results in raising the model's flexibility. Such a model covers a broader spectrum of project management's systems.

Note that A-resources should be strictly monitored: for operations which utilize A-resources, the corresponding *resource delivery moments have to be predetermined and calculated beforehand, i.e., the resources have to be delivered at a pregiven time*. Although, due to random disturbances affecting the project, it is impossible to forecast with a good accuracy, when a certain activity entering the project will be ready to start, the resource delivery schedule has to be deterministic.

B-resources should also be monitored, but *not closely*: for each activity which consumes those resources its feeding-in resource moment has to be determined. *Those time values are not calculated beforehand and are random values* conditioned on the model's future decisions.

Let us formulate the essence of the modified resource supportability model. *Given*:

- the projects due dates D_l;
- the least permissible probability p of accomplishing each project on time;
- the cost per time unit for hiring and maintaining a B-resource unit (for each type of resources);
- the penalty cost per time unit for the idleness of A-resources (for each activity which utilizes those resources);
- the penalty cost for the projects' delay (a single payment to the customer);
- the penalty cost for each time unit of delay;

- storage changes per time unit for the projects' completion before the due date.

The problem is to determine:

- the starting moment S_l of the project's realization, together with:
- the resource delivery schedule for A-resources, and
- the restricted resource levels for each type of B-resources,

in order to minimize the average total projects' expenses subject to the chance constraint.

12.3.2 <u>*The system's description*</u>

Several stochastic network projects of PERT type are considered. The durations of each activity entering each project are random and the corresponding probability density functions are pregiven. Certain activities entering the projects require extremely costly and rare resources (A-resources) which are usually delivered externally and are available for short periods within the time span of each project (e.g., technical experts, test-benches, special and unique facilities, heavy duty equipment and cranes, etc.). A-resources should be strictly monitored because shortages might significantly affect each project's schedule. Although it is unknown in advance when a certain activity which utilizes A-resources, will actually be ready to begin, A-resources have to be *delivered at a pregiven date* that has to be determined in advance. Thus, for activities, which utilize A-resources, a *deterministic* schedule of delivering resources to all projects is to be predetermined before the projects start to be carried out.

Other activities require constrained renewable B-resources (see §12.2) which are at the disposal of the project management and are in limited supply for each type of resources. Assume that a resource limit is independent on time, i.e., is fixed at the same level throughout each project's duration. Various B-resources, e.g., skilled workers, special equipment, etc., for projects under random disturbances require flexible, but not too close, monitoring. Since each activity entering any project is of random duration, the corresponding feeding-in resource moments to be determined are random values either.

Note that B-resources have to be hired in advance, in order to be delivered to the project's store at the moment the project actually starts. B-resources are released at the moment when the corresponding project is completed. The B-resource limits for each type of resources and for each project are problem's variables to be optimized as well as the moments the projects start to be carried out.

Assume, for simplicity, that an activity may utilize several non-consumable (renewable) B-resources of various types with fixed (pregiven) capacities.

The cost objective of the control model comprises the following expenses:

1. The costs of hiring and maintaining B-resources within the projects' duration, i.e., between the moment the projects start to be realized and the moment of the projects' completion (for each project separately).

2. The cost penalties paid for the A-resource idleness when an A-resource was delivered at the planned moment but not utilized since it had to wait for the moment the corresponding activity was actually ready to be operated.
3. Each project has its due date D_l and the penalty cost C_l^* (paid to the customer) for not accomplishing the project on time. In addition a penalty cost C_l^{**} has to be charged for each time unit of delay after the due date. If the project is accomplished before D_l, it has to be stored until the due date with a C_l^{***} penalty charge for each time unit of storage.

Note that the *operational costs* of processing projects' activities are not implemented in the cost objective. This is done deliberately since all operational expenses remain unchanged and do not depend on the control model.

12.3.3 Notation

Let us introduce the following terms:

$G_l(N,A)$ - the l-th PERT type project, $1 \le l \le n$;
n - the number of projects;
$(i,j)_l \subset G_l(N,A)$ - activity entering project $G_l(N,A)$;
t_{ijl} - duration of $(i,j)_l$ (a random value);
μ_{ijl} - average value of t_{ijl};
a_{ijl} - lower bound of t_{ijl} (pregiven);
b_{ijl} - upper bound of t_{ijl} (pregiven);
$f_{ijl}(t)$ - density function of t_{ijl} (pregiven);
n_l - the number of activities entering $G_l(N,A)$;
$(i_{\xi_A}, j_{\xi_A})_l$ - activity entering $G_l(N,A)$ which utilizes A-resources, $1 \le \xi \le n_{Al} \le n_l$;
n_{Al} - number of activities entering $G_l(N,A)$ which have to be supplied with A-resources;
$(i_{\eta_B}, j_{\eta_B})_l$ - activity entering $G_l(N,A)$ which utilizes B-resources, $1 \le \eta \le n_{Bl} \le n_l$;
n_{Bl} - number of activities entering $G_l(N,A)$ which have to be supplied with B-resources;
m - number of different B-resources;
R_{ql} - resource level of the q-th type B-resources at the disposal of $G_l(N,A)$, $1 \le q \le m$ (an optimized variable to be determined);
S_l - time moment project $G_l(N,A)$ starts to be carried out (an optimized variable to be determined);
D_l - the project's $G_l(N,A)$ due date (pregiven);
p_l^* - least permissible probability of meeting the project's due date on time, i.e., the model's chance constraint;
S_{ijl} - time moment activity $(i,j)_l$ actually starts (a random value condi-

tioned on the
model's decisions);

$F_{ijl} = S_{ijl} + t_{ijl}$ - the moment activity $(i,j)_l$ is completed (a random value);

$T(i_{\varepsilon_A}, j_{\varepsilon_A})_l$ - time moment A-resources have to be delivered to process activity $(i_{\xi_A}, j_{\xi_A})_l$ (an optimal deterministic schedule to be determined in advance);

$r_{i_{\eta_B} j_{\eta_B} ql}$ - capacity of the q-th type B-resources to be utilized by activity $(i_{\eta_B}, j_{\eta_B})_l$ (pregiven);

$F_l = \underset{(i,j)_l}{Max} F_{ijl}$ - the actual moment of project's $G_l(N,A)$ completion (a random value);

C_l^* - a single penalty cost of project's $G_l(N,A)$ delay, i.e., in case $D_l < F_l$ (pregiven);

C_l^{**} - a penalty cost for each time unit of project's $G_l(N,A)$ delay in case $D_l < F_l$ (pregiven);

C_l^{***} - storage charges per time unit for project's $G_l(N,A)$ completion before the due date, in case $F_l < D_l$ (pregiven);

$c(i_{\xi_A}, j_{\xi_A})_l$ - cost penalty to be paid by the management per time unit of the A-resource idling, i.e., in case $T(i_{\xi_A}, j_{\xi_A})_l < S_{i_{\varepsilon_A} j_{\varepsilon_A} l}$ (pregiven);

c_{lq} - cost per time unit for hiring and maintaining the q-th type B-resource unit throughout the project's realization (pregiven);

C_l - non-operational project's expenses comprising all kinds of cost penalties, the cost of hiring and maintaining B-resources throughout the project's $G_l(N,A)$ realization and the cost of storage expenses (a random value to be minimized);

$R_{ql}(t) \le R_{ql}$ - free (available for utilization) q-th type B-resources at moment $t \ge S_l$, for project $G_l(N,A)$ (a random value);

h_ξ - the value of the search step of the ξ-th coordinate in the cyclic coordinate search method (pregiven);

$\varepsilon > 0$ - search accuracy for the cyclic coordinate method (pregiven);

$\overline{C}_l^{opt} / S_l, \{R_{ql}\}, T(i_{\xi_A}, j_{\xi_A})_l$, $1 \le q \le m$, $1 \le \xi \le n_{Al}$ - the optimal conditional value delivering the minimum to project's $G_l(N,A)$ non-operational average expenses, on condition that values $\{S_l\}$, $\{R_{ql}\}$ and $\{T(i_{\xi_A}, j_{\xi_A})_l\}$ *are fixed and externally pregiven*;

$S_{l\min}$ - lower bound of the moment project $G_l(N,A)$ may actually start (determined by the project management and externally pregiven);

$S_{l\max}$ - upper bound of the moment project $G_l(N,A)$ may actually start (determined by the project management and externally pregiven);

$R_{ql\min}$ - lower bound of the resource level R_{ql}, $1 \le q \le m$, $1 \le l \le n$ (determined by the project management and externally pregiven);

$R_{ql\max}$ - upper bound of the resource level R_{ql}, $1 \le q \le m$, $1 \le l \le n$ (determined by the project management and externally pregiven);

C - non-operational expenses comprising all kinds of cost penalties, the cost of hiring and maintaining B-resources and the cost of storage expenses for the entire system comprising n projects (a random value to be minimized);

12.3.4 The problem

The general problem is as follows [81]:

to determine *in advance* optimized deterministic variables $S_l, \{R_{ql}\}$ and $T\left(i_{\xi_A}, j_{\xi_A}\right)_l$, $1 \le l \le n$, $1 \le q \le m$, $1 \le \xi \le n_{Al}$, and *within the projects' realization*, actual starting times S_{ijl} for all activities $(i,j)_l$, $1 \le l \le n$, in order to optimize the average system's expenses

$$\underset{\{S_l\},\{R_{ql}\},T\left(i_{\xi_A}, j_{\xi_A}\right)_l}{Min} \overline{C} \qquad (12.3.1)$$

subject to

$$\Pr\{F_l \le D_l\} \ge p_l^*, \quad 1 \le l \le n, \qquad (12.3.2)$$

$$S_{i_{\xi_A} j_{\xi_A} l} \ge T\left(i_{\xi_A}, j_{\xi_A}\right)_l, \quad 1 \le \xi \le n_{Al}, \qquad (12.3.3)$$

$$S_{ijl} \ge T(i) \quad \forall (i,j)_l \subset G_l(N,A), \qquad (12.3.4)$$

$$t = S_{i_{\eta B} j_{\eta B} l}, \; 1 \le \eta \le d \le n_{Bl} \Rightarrow \sum_{\eta=1}^{d} r_{i_{\eta B} j_{\eta B} ql} \le R_{ql}(t). \qquad (12.3.5)$$

where random value C satisfies

$$C = \sum_{l=1}^{n} \left[\begin{array}{l} \sum\limits_{\left(i_{\xi_A}, j_{\xi_A}\right)_l} \left\{ C\left(i_{\xi_A}, j_{\xi_A}\right)_l \cdot \left[S_{i_{\xi_A} j_{\xi_A} l} - T\left(i_{\xi_A}, j_{\xi_A}\right)_l \right] \right\} + \sum\limits_{q=1}^{m} \left[C_{ql} \cdot R_{ql} \cdot (F_l - S_l) \right] + \\ + \left[C_l^* + C_l^{**} \cdot (F_l - D_l) \right] \cdot \delta_l + \left[C_l^{***} \cdot (D_l - F_l) \cdot (1 - \delta_l) \right] \end{array} \right] \qquad (12.3.6)$$

and

$$\delta_l = \begin{cases} 1 & if \;\; F_l > D_l \\ 0 & otherwise. \end{cases} \qquad (12.3.7)$$

Here $T(i)$ stands for the time moment when event $i \in G_l(N,A)$ is carried out. Restriction (12.3.3) means that an activity which utilizes A-resources, cannot start before their corresponding delivery moment. Restriction (12.3.4) means that any activity $(i,j)_l$ entering $G_l(N,A)$ cannot start before moment $T(i)$. Restriction (12.3.5) means that if at a certain decision-point t B-resources are reallocated among $d < n_B$ activities, the summarized value of supplied resources (for each q-th type of B-resources) should not exceed the corresponding value $R_q(t)$, i.e., the total capacity of free available q-th type resources at moment t, $1 \le q \le m$.

Problem (12.3.1-12.3.7) is a complicated stochastic optimization problem which cannot be solved unless using heuristic methods.

We assume [81] that all projects $G_l(N,A)$, $1 \le l \le n$, are carried out independently each from the other, moreover, they are realized in different places. This means, that one central resource storehouse for all projects cannot be used. We decided to optimize each project independently and later on to summarize the obtained optimal (minimal) values $\overline{C_l}$ in order to determine the global solution $\overline{C}$ of the target function (12.3.1). Thus, the general optimization problem boils down to a simplified problem for a single project, which will be outlined below. In this course, as we transit to a case with $n=1$, index l will be omitted in further relations.

12.3.5 <u>The simplified problem for a single project</u>

The problem is as follows [80-81,151]:

to determine *in advance* optimized deterministic variables $S, \{R_q\}$ and $T\left(i_{\xi_A}, j_{\xi_A}\right)$, $1 \le q \le m$, $1 \le \xi \le n_{Al}$, and, *within the project's realization*, actual starting times S_{ij} for all activities $(i,j) \in G$ (random values) in order to minimize the average project's expenses

$$\underset{\{S\},\{R_q\},T\left(i_{\xi_A}, j_{\xi_A}\right)}{Min} \overline{C} \tag{12.3.8}$$

subject to

$$\Pr\{F \le D\} \ge p^*, \tag{12.3.9}$$

$$S_{i_{\xi_A} j_{\xi_A}} \ge T\left(i_{\xi_A}, j_{\xi_A}\right), \quad 1 \le \xi \le n_A, \tag{12.3.10}$$

$$S_{ij} \ge T(i) \quad \forall (i,j) \subset G, \tag{12.3.11}$$

$$t = S_{i_{\eta B} j_{\eta B}}, \; 1 \le \eta \le d \le n_B \Rightarrow \sum_{\eta=1}^{d} r_{i_{\eta B} j_{\eta B} q} \le R_q(t), \quad 1 \le q \le m, \tag{12.3.12}$$

where random value C satisfies

$$C = \sum_{l=1}^{n} \left[\begin{array}{l} \sum\limits_{\left(i_{\xi_A}, j_{\xi_A}\right)} \left\{ C\left(i_{\xi_A}, j_{\xi_A}\right) \cdot \left[S_{i_{\xi_A}, j_{\xi_A}} - T\left(i_{\xi_A}, j_{\xi_A}\right) \right] \right\} + \sum\limits_{q=1}^{m} \left[C_q \cdot R_q \cdot (F - S) \right] + \\ + \left[C^* + C^{**} \cdot (F - D) \right] \cdot \delta + \left[C^{***} \cdot (D - F) \cdot (1 - \delta) \right] \end{array} \right] \tag{12.3.13}$$

and

$$\delta = \begin{cases} 1 & if \quad F > D \\ 0 & otherwise. \end{cases} \tag{12.3.14}$$

12.3.6 <u>The problem's solution</u>

We will solve the simplified problem (12.3.8-12.3.14) as follows [80-81,151]. Two hierarchical optimization levels (cycles) are imbedded in the model. At the external upper cycle the problem (call it henceforth **problem PI**) is as follows:

Determine optimal values S, $\{R_q\}$, $1 \le q \le m$, to minimize the average project's non-operational conditional costs subject to the chance constraint

$$\min_{S,\{R_q\}} \left\{ \overline{C^{opt}} / S, \{R_q\} + \gamma \cdot K \cdot \left(\overline{p} - p^* \right) \right\} \tag{12.3.15}$$

and subject to restrictions

$R_{q\min} \le R_q \le R_{q\max}$, $1 \le q \le m$, (12.3.16)

$S_{\min} \le S \le S_{\max}$. (12.3.17)

Here:

- $\overline{C}^{opt} / S, \{R_q\}$ is calculated by means of simulation in order to obtain a representative statistics and by solving the internal optimization problem PII (see below);
- $\overline{p}$ is the simulated statistical frequency to meet the project's due date on time, i.e., to satisfy $F \le D$;
- K is a very large number (in the course of experimentation we took it to be equal 10^{17});
- $\gamma(x)$ is a zero-one function

$$\gamma(x) = \begin{cases} 0 & if \ \overline{p} \ge p^* \\ 1 & otherwise. \end{cases} \qquad (12.3.18)$$

Thus, objective (12.3.15) automatically prohibits cases with $\overline{p} < p^*$, i.e., honors the chance constraint (12.3.9).

To solve problem PI, we use a cyclic coordinate descent algorithm which minimizes (12.3.15) cyclically with respect to coordinate variables S, $\{R_q\}$. Value S is optimized first, then R_1, with fixed new (optimized) S, and so forth through R_m (honoring (12.3.16-12.3.17)). The process is then repeated starting with S again (second iteration) until the relative difference between two adjacent iterations becomes less than the pregiven tolerance $\varepsilon > 0$. Thus, implementing the algorithm results in undertaking a search in a $(m+1)$-dimensional space which is a combination of values S and $\{R_q\}$, $1 \le q \le m$, subject to restrictions (12.3.16-12.3.17). After obtaining a routine search point $(S, R_1, ..., R_m) = X$, the internal optimization problem PII at the lower level has to be applied. Thus, values S, $\{R_q\}$, $1 \le q \le m$, are input values for problem PII [80-81].

Problem PII is, in essence, a non-essential modification of the problem outlined in §12.2. The problem boils down to determine the quasi-optimum resource delivery schedule $T(i_{\xi_A}, j_{\xi_A})$, $1 \le \xi \le n_{Al}$, in order to minimize the average project duration by means of solving the resource constrained project scheduling problem via the knapsack resource reallocation problem. The general idea of the problem is as follows:

Given the due date D, the starting moment S of the project's realization and the resource levels $\{R_q\}$, $1 \le q \le m$, determine resource delivery schedule $T(i_{\xi_A}, j_{\xi_A})$, in order to minimize the project's duration by reallocating B-resources among the project activities. Thus, the problem is as follows:

$$\min_{\{S_{ij}\}, \{T(i_{\varepsilon_A}, j_{\varepsilon_A})\}} \left\{ \overline{C}^{opt} / T(i_{\xi_A}, j_{\xi_A}), S, \{R_q\} + \gamma \cdot K \cdot (\overline{t} - t^*) \right\} \qquad (12.3.19)$$

subject to (12.3.10-12.3.12).

The outlined below algorithm is, in essence, a unification of a coordinate search subalgorithm to develop a deterministic A-resource delivery schedule (outlined in §12.2) and a heuristic B-resource reallocation subalgorithm based on numerous applications of the knapsack resource reallocation problem in order to diminish the average project's duration (see §§11.1-11.2).

The combination of $(1+n_A+m)$ optimized variables S, $\{R_q\}$, $T(i_{\xi_A}, j_{\xi_A})$, which results in the minimal average value of non-operational project's costs $\left\{\overline{C}^{opt} \Big/ T(i_{\xi_A}, j_{\xi_A}), S, \{R_q\}\right\}$, has to be taken as the solution of the simplified problem (12.3.8-12.3.14). After determining beforehand (i.e., before the project starts at moment S) all optimized variables, the project has to be monitored with fixed and hired B-resources $\{R_q\}$, $1 \le q \le m$, and with the A-resource delivery schedule $T(i_{\xi_A}, j_{\xi_A})$. Such a methodological approach can be used both for monitoring real-time projects and by undertaking experimentation by means of simulation in order to assess the efficiency of the problem's solution.

Note that if solving problems PI and PII results in carrying out, in the average, M_1 and M_2 search steps, correspondingly, and obtaining representative statistics to calculate $\overline{C}$ results in undertaking M_3 simulation runs to monitor the project, then determining optimized parameters S, $\{R_q\}$, $\{T(i_{\xi_A}, j_{\xi_A})\}$ requires in the average $(M_1 \cdot M_2 \cdot M_3)$ simulation runs. Thus, we recommend applying the regarded control model for small- and medium-size network projects. In the case of large projects we suggest to reduce the amount of the project's activities by means of aggregation.

12.3.7 Monitoring stochastic network projects via resource reallocation simulation model

Values S, $\{R_q\}$ and $\{T(i_{\xi_A}, j_{\xi_A})\}$, obtained by solving problems PI and PII, serve as the input parameters for the simulation model at the lower level. The general idea of such a simulation model has been outlined above, in §§11.1.-11.2.

The simulation model comprises two submodels:

- the knapsack resource constrained reallocation to allocate B-resources among the project activities at decision points and to simulate the project's realization;
- the submodel to simulate the project's realization.

The knapsack resource reallocation problem is solved at decision points t when at least one activity (i_{η_B}, j_{η_B}) utilizing B-resources is ready to be operated but the available amount of resources is limited. A competition among the activities has to be carried out in order to choose those activities which can be supplied with resources and which have to be operated first. Assume that at a certain moment t, $d < n_B$ activities $\left(i_{\eta_B}, j_{\eta_B}\right)$, $1 \le \eta \le d$, $d > 1$, are ready to be processed, but at least for one type of resource there is a lack of available B-resources.

In case of fixed B-resources capacities $r_{i_{\eta_B} j_{\eta_B} q}$, we solve the outlined in §11.1 zero-one integer programming problem by determining zero-one integer values ρ_η, $1 \le \eta \le d$, to maximize the objective

$$\max_{\{\rho_\eta\}} \left\{ \sum_{\eta=1}^{d} \left[\rho_\eta \cdot p\left(i_{\eta_B}, j_{\eta_B}\right) \cdot \mu\left(i_{\eta_B}, j_{\eta_B}\right) \right] \right\}, \tag{12.3.20}$$

subject to

$$\sum_{\eta=1}^{d} \left[\rho_\eta \cdot p\left(i_{\eta_B}, j_{\eta_B}\right) \right] \le R_q(t), \ 1 \le q \le m, \tag{12.3.21}$$

where $p\left(i_{\eta_B}, j_{\eta_B}\right)$ is the probability for activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ to be on the critical path in the course of a simulation run, and

$$\rho_\eta = \begin{cases} 1 & \text{if activity } \left(i_{\eta_B}, j_{\eta_B}\right) \text{ is provided with resources}; \\ 0 & \text{otherwise}. \end{cases} \tag{12.3.22}$$

Thus, product $W_\eta = p\left(i_{\eta_B}, j_{\eta_B}\right) \cdot \mu\left(i_{\eta_B}, j_{\eta_B}\right)$ is the value activity $\left(i_{\eta_B}, j_{\eta_B}\right)$ contributes to the expected project's duration. The subset of activities which being supplied with resources, results in minimizing the project's duration, has to be chosen. All activities entering that subset start operating at moment t.

Problem (12.3.20-12.3.22) is solved by a zero-one integer programming algorithm with a precise solution. Values $p\left(i_{\eta_B}, j_{\eta_B}\right)$ are determined by means of simulation, by using (12.3.4) and taking into account values $T\left(i_{\xi_A}, j_{\xi_A}\right)$.

The simulation submodel is similar to the GSM model outlined in *12.2.8* and carries out the same operations (see §12.2).

12.3.8 Experimentation

In order to evaluate the performance of the algorithm, a medium-size network project of PERT type has been considered. The project's initial data is presented in Tab. 12.l. Two activities, namely, (4,6) and (7,10), utilize A-resources from outside, while all other activities are operated by using two types of non-consumable B-resources. Thus, $m = 2$, and the externally pregiven lower and upper bounds of values R_1 and R_2 are as follows:

$R_{1\min} = 30$, $R_{1\max} = 80$;

$R_{2\min} = 27$, $R_{2\max} = 80$.

The model's chance constraint $p^* = 0.9$, while values r_{ijq}, $1 \le q \le 2$, $(i, j) \in G \setminus (4,6) \setminus (7,10)$, are presented in Tab. 12.1. Four parameters are varied, namely $c\left(i_{\xi_A}, j_{\xi_A}\right)$, c_1, c_2, and the distribution of t_{ij}. Penalty rates $c\left(i_{\xi_A}, j_{\xi_A}\right)$ are similar for both activities (4,6) and (7,10). Two distributions of t_{ij} are considered:

a) t_{ij} is a random value uniformly distributed in the interval $[a_{ij}, b_{ij}]$;

b) t_{ij} is a random value normally distributed with average $\mu_{ij} = 0.5 \cdot (a_{ij} + b_{ij})$ and variance $V_{ij} = \frac{1}{36} \cdot (b_{ij} - a_{ij})^2$.

The experimental design is given in Tab. 12.2.

Table 12.1. The initial data

Activities	i	j	a_{ij}	b_{ij}	r_1	r_2
1	1	2	24	38	20	10
2	1	3	15	31	17	14
3	1	4	18	30	25	18
4	2	3	38	49	18	20
5	2	7	10	18	23	12
6	3	5	32	49	15	9
7	3	7	18	30	30	22
8	4	6	24	38	0	0
9	4	7	12	26	22	20
10	5	9	10	25	26	17
11	6	7	22	43	30	12
12	6	8	11	34	10	15
13	7	10	27	38	0	0
14	8	10	30	48	29	27
15	8	11	24	38	20	10
16	9	12	15	31	17	14
17	10	11	18	30	25	18
18	10	12	38	49	18	20
19	11	13	10	18	23	12
20	12	13	32	49	15	9

A total of 16 combinations (2×2×2×2) were considered. For each combination 500 simulation runs were carried out at each search step in the course of solving problems PI and PII. The values of the search step for all $(1 + n_A + m)$ coordinates have been set $h_\xi = 2$ for the first iteration and $h_\xi = 1$ for the next iterations. The pregiven search accuracy $\varepsilon > 0$ for the cyclic coordinate method has been set $\varepsilon = 0.001$.

Several output measures have been considered as follows:

$\overline{C}$ - the minimal average cost value of total expenses within one simulation run;

$\overline{p}$ - the average actual probability of meeting the due date on time;

S - the predetermined moment the project actually starts;

R_1 - the total capacity of resources of type 1 to be hired at moment S;

R_2 - the total capacity of resources of type 2 to be hired at moment S;

$T(4,6)$ - the planned moment for the A-resources to be delivered to process activity (4,6);

$T(7{,}10)$ - the planned resource delivery moment for activity (7,10).

Table 12.2. The experimental design

Variables	Values set in the experiment	Number of values
Penalty cost per time unit for A-resource idleness $c(i_{\xi_A}, j_{\xi_A})$	1,000; 1,200	2
Distribution of t_{ij}	Uniform, Normal	2
Cost per time unit for hiring a R_1-type unit c_1	3; 5	2
Cost per time unit for hiring a R_2-type unit c_2	3; 5	2

The summary of the experimentation is presented in Tab. 12.3. Note that for all combinations solving problems PI and PII requires 4 iterations, i.e., the quasi-optimal solution $\left\{S, \{R_q\}, T(i_{\xi_A}, j_{\xi_A})\right\}$ obtained at the fourth iteration results in the output value $\overline{C}$ which practically coincides with that obtained at the third iteration. In the course of implementing the cyclic coordinate descent algorithm the initial value of objective $\overline{C}$ has been reduced for all combinations by 85-90% in the average.

Table 12.3. The summary of the experimentation

Distribution	Input parameters			Output parameters						
	$c(i_{\xi_A}, j_{\xi_A})$	c_1	c_2	S	R_1	R_2	$T(i_{\xi_A}, j_{\xi_A})$ (4,6)	$T(i_{\xi_A}, j_{\xi_A})$ (7,10)	$\overline{C}$	$\overline{p}$
Uniform	1,000	3	3	15	50	42	30	106	199,105	0.99
	1,000	3	5	15	44	38	32	104	188,240	0.98
	1,000	5	3	14	52	44	34	109	210,311	1
	1,000	5	5	11	54	45	30	105	215,500	1
	1,200	3	3	17	42	40	28	100	180,902	0.97
	1,200	3	5	13	48	42	27	98	201,325	0.99
	1,200	5	3	10	52	48	26	98	208,850	1
	1,200	5	5	13	46	40	30	100	192,220	0.98
Normal	1,000	3	3	18	52	44	32	107	150,121	1
	1,000	3	5	16	40	43	31	105	140,953	1
	1,000	5	3	16	48	50	31	104	149,211	1
	1,000	5	5	15	49	46	34	109	166,002	0.98
	1,200	3	3	19	50	49	30	103	149,231	1
	1,200	3	5	17	47	43	29	102	145,652	1
	1,200	5	3	16	45	48	29	102	151,653	0.99
	1,200	5	5	15	43	44	32	105	156,845	0.97

The following conclusions can be drawn from the summary:

1. The average probability $\overline{p}$ of meeting the due date on time exceeds (for all

combinations considered in the experimental design) the pregiven chance constraint p^*. Thus, the algorithm minimizes objective $\overline{C}$ with respect to (12.3.9).

2. It can be clearly recognized that the regarded model (12.3.8-12.3.14) is very flexible. Increasing values c_1 and c_2 results either in shifting value S to the left or (and) in decreasing values R_1 and R_2. In other words, compensating control actions are introduced to prevent increasing objective $\overline{C}$. As to cost penalties $c(i_{\xi_A}, j_{\xi_A})$, increasing the latter results always in shifting values $T(i_{\xi_A}, j_{\xi_A})$ to the left. Thus, objective values $\overline{C}$ are protected from drastic fluctuations.
3. Using the normal distribution yields lower total cost expenses $\overline{C}$ than by using the uniform distribution.

§12.4 Conclusions

The following conclusions can be drawn from Chapter 12:

1. It can be well-recognized that model (12.3.1-12.3.7) covers and comprises all local models outlined above, in §§12.1-12.2. Those local models appear to become nothing but particular cases of the generalized model.
2. Being a truly resource supportability model and comprising several local predetermined resource delivery schedules (for individual projects), model (12.3.1-12.3.7) functions simultaneously at the projects' planning stage when determining optimal projects' starting moments S_l and optimal resource capacities R_{ql} subject to the chance constraints. Moreover, this model unifies resource constrained project scheduling with both deterministic resource delivery schedules (for A-resources) and random delivery schedules (for B-resources).
3. As outlined above, the model's optimization algorithm is based on the assumption that projects are independent and the model can be subdivided into non-intersecting and non-interacting fragments. In real life such an assumption cannot sometimes be justified and has to be withdrawn. In the latter case an additional hierarchical level has to be implemented in the model, namely, the level of optimal resource reallocation among the projects. Being essentially more complicated than the previous model, this refined model does not possess unavoidable drawbacks and can be optimized as well. However, in most practical cases model (12.3.1-12.3.7) as it stands now, provides sufficient accuracy [151].
4. Thus, model (12.3.1-12.3.7) can be regarded as one of the basic, universal models, which can be successfully implemented in innovative projecting.

Chapter 13. Stochastic Network Models for Determining Project's Planning Parameters

§13.1 Case of a group of aggregated projects in the form of consecutive operations

13.1.1 Introduction

As outlined above, in Chapters 11-12, besides control and scheduling models used at essential moments of the project's life cycle, certain models are aimed at determining planning characteristics, e.g., the project's due date, total capacities of various types of resources to be stored, etc. Those planning characteristics can be changed overtime, especially at emergency moments (see Chapters 4-6). The models under consideration can be regarded as models of mixed type, since they are implemented both at the planning stage (see Chapter 3), i.e., before the project's realization, and at the stage of monitoring the project. Note that the projects may be of different structure. At the initial stage of any complicated project with no similar prototype in the past, the model may be restricted to a source and a sink nodes connected by a chain of several intermediate consecutive operations of random duration. Thus, at the initial stage, a detailed network model does not exist.

We are considered with several simultaneously realized preliminarily projects (PP) [88] consisting of a chain of operations to be processed in a definite technological sequence. Each project's operation utilizes qualified manpower of various specialties, i.e., several non-consumable resources, with fixed capacities. Each type of resource at the management's disposal is in limited supply, with a resource limit that remains unchanged at the same level throughout the projects' duration, i.e., until the last project is actually completed. Thus, due to the limited resource levels, projects' operations may have to wait in lines for resource supply, in order to start functioning. Since for each operation its duration is a random variable with given density function, a deterministic schedule of the moments operations actually start cannot be determined.

The general problem is to determine:

- optimal *deterministic* total resource capacities for each type of resource at the management's disposal (beforehand), and
- random values of the moments operations actually start (in the course of the projects' realization and conditioned on our decisions),

- to minimize the average of the total expenses of hiring and utilizing all resources subject to the chance constraints of meeting the projects' due dates on time.

The problem is solved by means of a heuristic algorithm by a combination of the cyclic coordinate descent method (at the upper level) and a simulation model (at the lower level). Resource reallocation between the projects waiting in lines is carried out via decision rule based on a mini-max principle. The latter enables support to "weaker" projects from the "stronger" ones in the course of the projects' realization.

13.1.2 Notation

Let us introduce the following terms:

n - number of projects of preliminarily type PP_i, $1 \le i \le n$, to be realized simultaneously in a project system;

O_{ic} - the c-th operation of the i-th project in the form of a consecutive chain, $1 \le c \le m_i$;

m_i - number of operations in project PP_i;

t_{ic} - random duration of operation O_{ic} (a random value);

$\bar{t}_{ic}$ - average value of t_{ic} (pregiven);

V_{ic} - variance of t_{ic} (pregiven);

R_k - the total capacity of the k-th type of resources, $1 \le k \le d$, at the disposal of the project system (a deterministic value to be optimized);

d - number of resources;

r_{ick} - the k-th resource capacity to be assigned to operation O_{ic} (pregiven);

D_i - the due date for project PP_i (pregiven);

p_i^* - chance constraint to meet the due date D_i on time (pregiven);

S_{ic} - the moment operation O_{ic} actually starts (a random value, to be determined by the simulation model by means of a decision rule in the course of carrying out the projects);

F_{ic} - the moment operation O_{ic} terminates (a random value);

F_i - the moment project PP_i terminates, $F_i = S_{im_i} + t_{im_i}$ (a random value);

F - the moment the last project terminates, $F = \underset{i}{Max} F_i$;

$p_i\{\vec{R}_k, D_i\}$ - actual probability of meeting D_i on time on condition that R_k total resource capacities, $1 \le k \le d$, are hired by the project system;

$W_k\{S_{ic}, t\}$ - the summarized capacity of the k-th resource assigned to operations at moment t, on condition that operations O_{ic} start at moments S_{ic}, $1 \le k \le d$;

$R_k(t) = R_k - W_k(S_{ic}, t)$ - free available resources of k-th type at moment t;

s_k - the cost of hiring, maintaining and utilizing the k-th resource unit at the time unit, $1 \le k \le d$ (pregiven, a constant value);

ΔR_k - the positive search step value to optimize variable R_k, $1 \le k \le d$ (pregiven);

ε - the relative accuracy value to obtain an optimal solution (pregiven);

$R_{k\min}$ - the minimal possible level for the total capacity R_k, $1 \le k \le d$ (pregiven);

$R_{k\max}$ - the maximal possible level for value R_k, $1 \le k \le d$ (pregiven);

Q - the system's total resource expenses.

Note that relations

$$R_{k\min} \geq \max_i \max_c r_{ick}, \tag{13.1.1}$$

$$R_{k\max} \leq \sum_i \left\{ \max_{1\leq c\leq m_i} r_{ick} \right\}, \tag{13.1.2}$$

$$R_{k\min} \leq R_k \leq R_{k\max}, \ 1\leq i\leq n, \ 1\leq c\leq m_i, \ 1\leq k\leq d, \tag{13.1.3}$$

hold.

Restriction (13.1.1) is evident since otherwise some of the projects cannot be realized at all. If (13.1.2) does not hold a certain part of resources will not participate in the projects' realization.

13.1.3 The problem's formulation

The general problem is to determine both optimal *deterministic* values R_k, $1\leq k\leq d$, (before the projects' realization) and *random* values S_{ic} (in the course of the projects' realization and conditioned on our decisions), $1\leq i\leq n$, $1\leq c\leq m_i$, to minimize the average of the total resource expenses

$$\min_{\{\vec{R}_k, S_{ic}\}} E\left\{ \sum_{k=1}^{d} s_k R_k \cdot \left[\max_i F_i - \min_i S_{i1} \right] \right\} \tag{13.1.4}$$

subject to (13.1.3) and

$$W_k\{S_{ic}, t\} \leq R_k \quad \forall t : t \geq \min_i S_{i1}, \tag{13.1.5}$$

$$p_i\{\vec{R}_k, D_i\} \geq p_i^*, \ 1\leq i\leq n, \ 1\leq k\leq d, \ 1\leq c\leq m_i. \tag{13.1.6}$$

Note that problem (13.1.3-13.1.6) is a very complicated stochastic optimization problem which does not provide an analytical solution. We suggest solving the problem by using a two-level heuristic algorithm. The latter comprises a simulation model and a subalgorithm to carry out the coordinate descent optimization method.

Note, in conclusion, that to simplify the problem, we will henceforth assume that $\min_i S_{i1} = 0$ holds.

13.1.4 The simulation model

The input data of the simulation model is the vector of total resource capacities $\vec{R}_k$, $1\leq k\leq d$, which is determined in the course of the coordinate descent algorithm's work. Thus, in the course of a routine simulation run vector $\{R_k\}$ is fixed and remains unchanged. It goes without saying that vector $\vec{R}_k$ satisfies (13.1.1-13.1.3).

The main task of the simulation model is to determine (in the course of a simulation run) random starting moments S_{ic} of all operations O_{ic}, $1\leq i\leq n$, $1\leq c\leq m_i$, entering the projects, with respect to a mini-max objective

$$I = \max_{\{S_{ic}\}} \min_i \left[\frac{p_i\{\vec{R}_k, D_i\} - p_i^*}{p_i^*} \right] \tag{13.1.7}$$

and subject to (13.1.5). Value $p_i\{\vec{R}_k, D_i\}$ can be evaluated by means of undertaking numerous simulation runs in order to obtain representative statistics and, later on, calculating frequencies for the probability value $\Pr\{F_i \le D_i\}$. As to the mini-max objective, it is imbedded in the outlined below decision rule to reallocate restricted resources among projects ready to be operated and waiting in lines.

A routine simulation run starts functioning at $t = 0$ and terminates with the completion of the last project. The simulation model comprises three submodels as follows:

Submodel I actually governs most of the procedures to be undertaken in the course of the projects' realization, namely:

- determines essential moments (decision points) when projects may be supplied with free available resources. A routine essential moment usually coincides either with the moment an operation is finished and additional resources become available, or when a subset of new operations O_{ic} becomes ready to be processed;
- singles out (at a routine decision point) all the operations that are ready to be processed;
- checks the possibility of supplying these operations with available resources without undertaking a competition;
- supplies the chosen operations O_{ic} with resources and later on simulates the corresponding durations t_{ic};
- returns the utilized non-consumable resources to the project system store (at the moment an operation is finished);
- updates the remaining projects at each routine decision point;
- determines the completion moment for each projects,

together with several other, less important, procedures.

Submodel II calculates auxiliary decision rule values in case when there is a lack of available resources and not all the operations ready to be processed and waiting in line for resources at a routine decision point t, can start to be realized. Assume that at moment t q operations $O_{i_1c_1}, O_{i_2c_2}, \dots, O_{i_qc_q}$ are ready to be processed and at least for one type k of resources, relation $\sum_{v=1}^{q} r_{i_vc_vk} > R_k(t)$ holds. For each PP_{i_v} waiting in line, Submodel II calculates value

$$\Pr\left\{F_{i_v} \le D_{i_v}\right\} = \Phi\left\{\frac{D_{i_v} - t - \sum_{r=c_v}^{m_{i_v}} \bar{t}_{i_v r}}{\sqrt{\sum_{r=c_v}^{m_{i_v}} V_{i_v c_v}}}\right\},\ 1 \le v \le q\,, \qquad (13.1.8)$$

where

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{z^2}{2}}dz . \qquad (13.1.9)$$

Thus, value (13.1.8) is an approximate probability estimate for project PP_{i_v} to meet its target on time on condition that the project will obtain needed resources at moment t and will not wait in lines henceforth. Such an assumption has been successfully used for many control problems in project management and manufacturing systems [54,70,92-93,118].

After determining the values of deviation from the target by

$$\gamma_{i_v}=\frac{\Pr\left\{F_{i_v}<D_{i_v}\right\}-p^*_{i_v}}{p^*_{i_v}},\ 1\le v\le q , \qquad (13.1.10)$$

we sort the operations in ascending order. Denote the newly reordered operations

$$O_{j_1 f_1},O_{j_2 f_2},...,O_{j_q f_q},\ \ \gamma_{j_\xi f_\xi}<\gamma_{j_{\xi+1},f_{\xi+1}},\ \ 1\le \xi \le q-1 . \qquad (13.1.11)$$

It can be well-recognized that the less value γ_{i_v} is, the more urgent becomes the problem of supplying PP_{i_v} with resources as soon as possible. Here we make no difference between project PP_{i_v} and operation $O_{i_v c_v}$ in (13.1.11) since only one operation of any project may wait in line for resources at a certain decision moment t. Thus, priority value γ_{i_v} refers both to project PP_{i_v} and to operation $O_{i_v c_v}$.

Submodel III undertakes reallocation of free available resources $R_k(t)$, $1\le k\le d$, among project PP_{i_v}, $1\le v\le q$. All the sorted operations in (13.1.11) are examined, one after another, in the ascending order of values γ, to check, for each operation, the possibility that it can be supplied with remaining available resources. If, for a certain operation $O_{j_\xi f_\xi}$, $1\le\xi\le q$, relations $r_{j_\xi f_\xi k}\le R_k(t)$, $1\le k\le d$, hold, the needed resources $r_{j_\xi f_\xi k}$ are passed to the operation while the remaining resources $R_k(t)$ are updated, $R_k(t)-r_{j_\xi f_\xi k}\Rightarrow R_k(t)$, $1\le k\le d$. Then, the next operation $O_{j_{\xi+1},f_{\xi+1}}$ is examined. If not all relations $r_{j_\xi f_\xi k}\le R_k(t)$ hold, we proceed straightforward examining the next operation. The procedure terminates either when all the available resources are reallocated among the operations or all the q operations have been examined. The procedure is simple in usage and has been used in various scheduling problems [70,88,92-93, etc.].

It can be well-recognized that since decision rule (13.1.10) is imbedded in decision-making for resource reallocation, the outlined above simulation model honors objective (13.1.7).

Note that the general idea of the mini-max approach is as follows. In the course of projects' realization the project system takes an urgent care of "weaker" projects which deviate from their trajectories and their chance constraints more than other projects. Those projects have to be supplied with resources in the first place at the expense of other, "stronger" projects. Thus, the general idea of the mini-max objective (13.1.7) is to raise the weakest project as much as possible in order to balance all the projects under realization.

13.1.5 The cyclic coordinate descent subalgorithm

As mentioned above, the suggested heuristic algorithm to solve the problem (13.1.3-13.1.6) comprises two levels. At the lower level the simulation model undertakes numerous simulation runs in order to manage the projects' realization on the basis of the mini-max principle. At the upper level the heuristic search subalgorithm undertakes cyclic coordinate optimization in order to obtain the optimal vector $\vec{R}_k$. The procedure of the optimization is based on minimizing objective (13.1.4) cyclically with respect to coordinate variables $R_1, R_2, ..., R_d$. Coordinate R_1 is optimized first, then R_2, and so forth through R_d. The coordinate descent method is outlined in [53-54,83,89], has been successfully implemented in §§11.3, 12.1-12.2 and is incorporated in the procedure of the search subalgorithm as follows:

Step 1. Determine the initial search point $X^0 = \{R_k^0\}$ by taking deliberately overstating values, e.g., $R_k^0 = R_{k\max}$, $1 \le k \le d$. It can be well-recognized that setting $X^0 = \{R_{k\max}\}$ results in

$$p_i\{\vec{R}_k^0, D_i\} > p_i^*, \ 1 \le i \le n, \qquad (13.1.12)$$

and, thus, X^0 is a feasible solution. Note that for any initial search point X^0 relation (13.1.12) can be checked by means of simulation, on the basis of numerous simulation runs, by comparing the corresponding statistical frequency rates with pre-given values p_i^*, $1 \le i \le n$. If at least for one index i relation (13.1.12) does not hold, problem (13.1.3-13.1.6) has no solution. Otherwise apply the next step.

Step 2. Fix the initial values $\{R_k\}$, $\vec{R}_k = \vec{X}^0$, and start diminishing value R_1 by ΔR_1 consecutively, i.e., $R_1 - r \cdot \Delta R_1 \Rightarrow R_1$, $r = 1, 2, ...$, while all other coordinates $R_2, R_3, ..., R_d$ are fixed and remain unchanged. Each newly determined search point ($R_1 - r \cdot \Delta R_1, R_2, ..., R_d$) has to be examined by means of simulation in order to verify the following statements:

A. Checking a new search point results in decreasing objective (13.1.4);

B. Restrictions (13.1.6) remain valid.

In order to formalize the procedure of verification via a simulation model, we suggest:

- to undertake M simulation runs in order to obtain representative statistics $(M \div 500 - 1000)$;
- to modify objective (13.1.4) on the basis of M simulation runs as follows:

$$Q^* = \left(\sum_{k=1}^{d} s_k R_k\right) \cdot \left\{\frac{1}{M}\sum_{m=1}^{M} F^{(m)} + \sum_{i=1}^{n} K_i \cdot \beta\left(\left[\frac{1}{M}\sum_{m=1}^{M} \alpha\left(F_i^{(m)} - D_i\right)\right] - p_i^*\right)\right\}, \qquad (13.1.13)$$

where

$$\alpha(x)=\begin{cases}1 & \text{if } x\le 0\\ 0 & \text{otherwise}\end{cases},\quad \beta(x)=\begin{cases}1 & \text{if } x< 0\\ 0 & \text{otherwise}\end{cases},$$

$F^{(m)}$ is the moment the last project terminates in the m-th simulation run, $1\le m\le M$,

$F_i^{(m)}$ is the moment the i-th project terminates in the m-th simulation run,

K_i is a very large positive value (usually taken [53,118] close to 10^{17}) in order to prohibit automatically cases $p_i\left\{\vec{R}_k, D_i\right\}< p_i^*$ for any i, $1\le i\le n$.

Thus, verifying the validity of statements A and B independently from each other is substituted by checking the validity of the monotonous decrease of one objective (13.1.13). Note that for the sake of simplicity we have taken $\underset{i}{\mathrm{Min}}\, S_{i1}=0$ in (13.1.13).

Step 3. We proceed examining the monotonous decrease of estimate Q^* in the course of diminishing consecutively the first coordinate R_1, until either:

1. The diminished value R_1 reaches its lower bound $R_{1\min}$, or
2. The monotonous decrease of objective (13.1.13) ceases to hold for $R_{1\min}\le R_1\le R_{1\max}$.

In any case value R_1 which corresponds to the minimal value of Q^*, is fixed, and we start diminishing the second coordinate, R_2, by step ΔR_2 (with fixed values R_1 (newly obtained), $R_3,\ldots, R_d$). The process proceeds for other coordinates, etc., until the last coordinate, R_d, is examined.

Note that in the course of undertaking a coordinate search each successive search results always in decreasing objective (13.1.13). Otherwise, i.e., if a routine search step does not result in decreasing (13.1.13), the corresponding routine coordinate R_k is fixed and the next, the $(k+1)$-th coordinate R_{k+1}, starts to be examined.

Step 4. Obtaining a new search vector $\left\{\vec{R}_k\right\}$ in the course of optimizing *all the coordinates separately*, results in realizing the first iteration to determine the quasi-optimal values $\{R_k\}$. All search steps ΔR_k have to be diminished (mostly by dividing by two), and we proceed to minimize (13.1.13) cyclically with respect to the new coordinate variables, beginning from R_1.

Step 5. For all next iterations in the course of the coordinate optimization, a search is realized for each routine coordinate R_k, $1\le k\le d$, in two opposite directions, namely $R_k-\Delta R_k$ and $R_k+\Delta R_k$, to determine the direction of objective's (13.1.13) decline. The direction which results in the highest objective's decrease, has to be chosen. The search process proceeds in that direction until the objective's decrease ceases to hold.

Step 6. After undertaking a routine search iteration v, $v=1,2,...$, the objective value (13.1.4), Q^v, referring to that iteration, has to be compared with the results of the previous, $(v-1)$-th iteration, by calculating

$$\Delta^{(v)} = \frac{Q^{(v-1)} - Q^{(v)}}{Q^{(v-1)}}. \qquad (13.1.14)$$

Thus, at least two iterations have to be undertaken.

Step 7. If relation $\Delta^{(v)} < \varepsilon$ holds, i.e., if the relative difference between two adjacent iterations $Q^{(v-1)}$ and $Q^{(v)}$ becomes less than the pregiven tolerance $\varepsilon > 0$, the algorithm terminates. Otherwise, Step 2 has to be applied.

13.1.6 Numerical example

In order to check the fitness of the developed mini-max control model experimentation has been undertaken. Three simultaneously realized preliminarily projects are considered. The first project comprises two consecutive operations, while both the second and the third projects comprise three consecutively realized operations. Two types of non-consumable resources participate in the system. The projects' parameters are as follows:

Project No. 1

$r_{111}=15$;	$r_{121}=17$;	
$r_{112}=60$;	$r_{122}=51$;	
$O_{11}=U(31,40)$;	$O_{12}=U(48,55)$.	

Project No. 2

$r_{211}=15$;	$r_{221}=20$;	$r_{231}=27$;
$r_{212}=73$;	$r_{222}=88$;	$r_{232}=85$;
$O_{21}=U(30,38)$;	$O_{22}=U(18,30)$;	$O_{23}=U(28,39)$.

Project No. 3

$r_{311}=20$;	$r_{321}=26$;	$r_{331}=18$;
$r_{312}=64$;	$r_{322}=78$;	$r_{332}=80$;
$O_{31}=U(30,45)$;	$O_{32}=U(16,28)$;	$O_{33}=U(20,30)$.

Other system's parameters are as follows:

$R_{1\min}=27$;	$R_{1\max}=50$;	$s_1=50\$$;
$R_{2\min}=88$;	$R_{2\max}=160$;	$s_2=30\$$;
$p_1=0.75$;	$p_2=0.80$;	$p_3=0.85$;
$D_1=127$;	$D_2=140$;	$D_3=150$.

The optimization process is presented in Tab. 13.1. A conclusion can be drawn that the cyclic coordinate descent algorithm requires only two iterations with 23 search steps. Thus, the two-level heuristic algorithm performs well [90,118].

Table 13.1. Illustrative performance of the algorithm

Search step number	R_1	R_2	No. of iteration	Total resource expenses	Feasibility
1	50	160	1	1,043,900	feasible
2	49	160	1	1,036,750	feasible
3	48	160	1	1,029,600	feasible
4	47	160	1	1,022,450	feasible
5	46	160	1	1,015,300	feasible
6	45	160	1	1,008,150	feasible
7	44	160	1	1,001,000	feasible
8	43	160	1	1,070,300	non-feasible
9	44	159	1	996,710	feasible
10	44	158	1	992,420	feasible
11	44	157	1	988,130	feasible
12	44	156	1	983,840	feasible
13	44	155	1	979,550	feasible
14	44	154	1	975,260	feasible
15	44	153	1	970,970	feasible
16	44	152	1	966,680	feasible
17	44	151	1	962,390	feasible
18	44	150	1	958,100	feasible
19	44	149	1	953,810	feasible
20	44	148	1	949,520	optimal
21	44	147	2	945,230	non-feasible
22	43	148	2	1,014,860	non-feasible
23	45	148	2	956,670	feasible

13.1.7 Conclusions

1. The developed optimization problem covers a realistic situation in a project system, at the stage of developing preliminary projects.
2. The problem can be solved by using a two-level algorithm. At the upper level a heuristic cyclic optimization procedure is carried out. At the lower level a simulation model is implemented.
3. The developed model undertakes cost-optimization and can be used both in planning and monitoring several preliminary projects.
4. The backbone of the simulation model is the outlined above decision rule which is based on the mini-max principle. The latter enables resource support to the “weakest” projects which deviate essentially from their targets, at the expense of “stronger” projects, which are more successful in the course of their realization.
5. The model can be modified for the case of PERT projects with different priorities (see below). This is a perspective research since the objective is based on analyzing the projects’ different importance.

§13.2 Case of a group of PERT projects with different priorities

13.2.1 Introduction

The model described below resembles the one outlined in §11.3 with the exception of two important characteristics:

- unlike the model outlined in §§11.3 and 13.1, the projects are of different priorities;
- the model is aimed at determining the minimal due date for all projects.

The problem [94] is to determine the generalized due date for all projects as well as the moments that resources are fed in and projects' activities start, in order to maximize the heuristic objective taking into account both the projects' priorities and the corresponding chance constraints. Thus, the model is implemented mainly on the planning stage.

The problem is solved by means of simulation. Two optimization cycles are imbedded in the model. The external cycle deals with determining the minimal due date D for all projects. Thus, the due date serves as the input value for the internal cycle. The latter uses heuristic decision-making rules to reallocate free available resources among the projects in order to meet the projects' chance constraints.

13.2.2 Notation

Let us introduce the following terms:

$G_\ell(N_\ell, A_\ell)$ - the ℓ-th network stochastic project, $1 \le \ell \le n$;

n - number of PERT type stochastic network projects;

$(i,j)_\ell$ - activity (i,j) entering project $G_\ell(N_\ell, A_\ell)$;

$r_{ij\ell k}$ - capacity of the k-th type resources allocated to activity $(i,j)_\ell$, $1 \le k \le m$ (pregiven);

m - number of different resources;

R_k - total available capacity of k-th type non-consumable resources (pregiven);

D - the general due date for accomplishing all network projects (to be determined);

$D_{\min}$ - the minimal possible general due date (pregiven);

η_ℓ - the priority index (level of importance) of project $G_\ell(N_\ell, A_\ell)$ (pregiven);

p_ℓ - the minimal admissible probability for project G_ℓ of meeting the due date D on time (pregiven);

$S_{ij\ell}$ - the moments activities $(i,j)_\ell$ actually start (random variables to be determined within the projects' realization);

F_ℓ - the actual moment project $G_\ell(N_\ell, A_\ell)$ is completed (a random value determined on the model's decision-making rule);

ΔD - the time step in order to determine optimal value D.

13.2.3 The problem's formulation

Determine the total due date D for all projects and the random starting moments $S_{ij\ell}$ for all the activities for two contradictive objectives:

$$\min D \tag{13.2.1}$$

and

$$\max_{\{S_{ij\ell}\}} \left\{ \sum_{\ell=1}^{n} \left[\eta_\ell \cdot \overline{P}_\ell(D) \right] \right\} \tag{13.2.2}$$

subject to

$$D \geq D_{\min}, \tag{13.2.3}$$

$$\overline{P}_\ell(D) \geq P_\ell, \tag{13.2.4}$$

where $\overline{P}_\ell(D) = \Pr\{F_\ell \leq D\}$ is the simulated probability of project G_ℓ to meet the due date D on time.

13.2.4 The problem's solution

The solution is based on three following principles:

1. At the upper level a search for value D is carried out by means of increasing D via consecutive steps by ΔD, i.e., $D + \Delta D \Rightarrow D$ is realized.
2. At the lower level a simulation model SM is determined with input values D, $\{\eta_\ell\}$, $\{R_k\}$, $\{r_{ij\ell k}\}$, $\{P_\ell\}$.
3. A decision rule is imbedded in the SM based on the idea of pairwise comparisons.

Two different cases will be examined at each essential moment $t \geq 0$ when certain activities $(i_v, j_v)_\ell$ are ready to consume available resources $R_k(t)$:

Case A: some activities may refer to the same project.

Case B: all activities refer to different projects.

13.2.5 Decision-making rule in Case A (DRA)

When activities $(i_v, j_v)_\ell$ refer to one and the same project $G_\ell(N_\ell, A_\ell)$ the decision-making rule consists of three steps and boils down to the following:

Step 1. By means of simulation calculate values $\overline{P}(i_v, j_v)_\ell$ (the probability of the activity $(i_v, j_v)_\ell$ to be on the critical path) for all activities seeking for resources.

Step 2. Calculate for all activities under competition values $\bar{t}_{i_v, j_v \ell}$ (average duration) and $\xi(i_v, j_v)_\ell = \overline{P}(i_v, j_v)_\ell \cdot \bar{t}_{i_v, j_v \ell}$.

Step 3. Activity with *maximal* value ξ is chosen as the winner.

Thus, practically speaking, rule DRA is based on the knapsack approach [70]. Thus, after implementing DRA, only one winning activity from each project will remain seeking for resources.

13.2.6 Decision-making rule in Case B (DRB)

DRB is based on the idea of pairwise comparison and is always used *after* carrying out DRA. Thus, before applying DRB, it is assumed that *all* competing activities refer to *different* projects and present the corresponding winner by implementing DRA.

The following steps enter the rule DRB:

Step 1. Sort all activities $(i_v, j_v)_\ell$ in descending order of their priority indices.

Step 2. Consider the first two activities, namely, $(i_1, j_1)_1$ and $(i_2, j_2)_2$. Two options are examined:

- the first activity is supplied with resources while the second one waits in the line for time period $\bar{t}_{i_1, j_1 \ell}$, and
- the second activity is supplied with resources while the first one waits in the line for $\bar{t}_{i_2, j_2 \ell}$.

It is assumed that afterwards both projects do not wait in lines.

Step 3. Calculate (by means of simulation) values $\overline{P}_1(D)$ and $\overline{P}_1\left(D - \bar{t}_{i_2 j_2 2}\right)$ for the first project and values $\overline{P}_2(D)$ and $\overline{P}_2\left(D - \bar{t}_{i_1 j_1 1}\right)$ for the second project.

Step 4. If $\eta_1 \cdot \overline{P}_1(D) + \eta_2 \cdot \overline{P}_2\left(D - \bar{t}_{i_1 j_1 1}\right) \geq \eta_2 \cdot \overline{P}_2(D) + \eta_1 \cdot \overline{P}_1\left(D - \bar{t}_{i_2 j_2 2}\right)$, activity $(i_1, j_1)_1$ is the winner. Otherwise activity $(i_2, j_2)_2$ wins the competition.

13.2.7 The compound decision-making algorithm

At any *essential moment* t where at least one activity $(i_v, j_v)_\ell$ is seeking for resources to start operating, the compound decision-making algorithm has to be implemented. The algorithm comprises the following steps:

Step 1. Arrange at any *essential moment* $t \geq 0$ all activities $(i_v, j_v)_\ell$ waiting for resources, in a descending order of their projects' priorities η_ℓ.

Step 2. For all ready activities referring to the same project undertake competition by means of DRA (only one winner allowed for each project).

Step 3. For all projects with a single ready activity (seeking for resources) carry out a competition by means of DRB. The winner competes with the next competitive activity, until only one winner is left; let it be $(i_w, j_w)_\ell$.

Step 4. If relation $r(i_w, j_w)_{\ell k} \leq R_k(t)$, $1 \leq k \leq m$, holds, activity $(i_w, j_w)_\ell$ is provided with resources. Go to Step 5. Otherwise, activity $(i_w, j_w)_\ell$ is excluded from the competition. Go to Step 1.

Step 5. Update the free available resources $R_k(t) - r(i_w, j_w)_{\ell k} \Rightarrow R_k(t)$. Return to Step 1.

Step 6. The process of free resource reallocation terminates when either all free resources $R_k(t)$ are allocated, or all competitive activities $(i_v, j_v)_\ell$ are supplied with resources.

13.2.8 The enlarged procedure of solving the optimization problem

The solution of problem (13.2.1-13.2.4), thus, can be obtained by using the enlarged stepwise procedure as follows:

Step 1. Start examining the increasing value D, beginning from $D_{\min}$, by means of a search procedure: $D+\Delta D \Rightarrow D$, $D \geq D_{\min}$.

Step 2. For any value D obtained at Step 1 simulate the projects' realization on the basis of decision rules DRA and DRB. Those rules have to be incorporated in the simulation model.

Step 3. Undertake M simulation runs in order to obtain representative statistics. Calculate for all projects values $\overline{P}_{\ell}(D)$, $1 \leq \ell \leq n$.

Step 4. If all values $\overline{P}_{\ell}(D)$ satisfy chance constraint (13.2.4), determine the minimal D satisfying restriction (13.2.3-13.2.4). Thus, the optimal solution of problem (13.2.1-13.2.4) is obtained. Otherwise return to Step 1.

As to objective (13.2.2), it is embedded in the algorithm through decision rule DRB.

13.2.9 Conclusions

The following conclusions can be drawn from §13.2:

1. The presented resource constrained reallocation model can be used in project management as a decision support model for planning and monitoring several stochastic network projects. The model has been successfully used for small and medium size projects of PERT type.
2. The outlined model is suitable for resource scheduling in stochastic network projects, when the processing of certain activities is based on delivering resources, e.g., in high technology projects, defense related industries, opto-electronics, aerospace, etc.

§13.3 Stochastic network model with target amount rescheduling

13.3.1 Introduction

The problem associated with developing multilevel on-line production control models under random disturbances for flexible manufacturing systems has been discussed in literature [50-54,61,63,73,83,87, etc.] and outlined in Chapters 6, 11 and 12. Most of those investigations deal with not fully automated plants of 'man-machine' type where the output cannot be measured continuously on-line, but only at preset control points. The main idea of the interaction problems between different levels in hierarchical control systems is based on the conception of emergency introduced by the scientific school of Golenko-Ginzburg (see, e.g., [63]). By using the idea that hierarchical levels can interact only in special situations, the so-called emergency points, one can decompose a general and complex multi-level problem of optimal production control into a sequence of one-level problems. We will show below that this general idea can be applied to stochastic network projects as well.

Two different optimization cases are usually considered:

1. Case with a conflicting two-criteria objective, namely, to maximize the

probability of completing the production on the due date, and to minimize the number of control points; but the first criterion is dominant.

2. The objective is to maximize the expected net profit.

A two-level system is considered to be composed of several different projects U_i, $1 \le i \le n$, at the lower level and a control device at the upper one. The upper system's level is required to provide a given target amount V by a given due date D subject to a chance constraint, i.e. the least permissible probability p of meeting the target on time is pregiven. Each project U_i has several possible speeds v_{i1}, v_{i2}, ... , v_{im}, which are subject to random disturbances. The project's output can be measured only at preset inspection (control) points. The target amount is gauged by a single measure, e.g. in square meters, and may be rescheduled among the projects. For each project, the average costs per time unit for each speed and the average cost of performing a single inspection at a control point to observe the actual output at that point, are given.

In Chapter 6 we have outlined a cost-optimization on-line control model which for a single project determines both control points and speeds to be introduced at those points, in order to minimize the project's expenses within the planning horizon, subject to the chance constraint. We present a two-level on-line control model under random disturbances, which centers on minimizing the system's expenses subject to the chance constraint. The suggested two-level heuristic algorithm is based on rescheduling the system's target among the projects both at $t = 0$, when the system starts functioning, and at each emergency point, when it is anticipated that a certain project is unable to meet its local target on time subject to a chance constraint. At any emergency point t the remaining system's target V_t is rescheduled among the projects; thus, new local targets V_{it}, $1 \le i \le n$, $\sum_i V_{it} = V_t$, are determined. New local chance constraint values p_{it} are determined too. Those values enable the system to meet its overall target at the due date subject to the pregiven chance constraint p.

After reassigning to each project U_i its new target V_{it} and the chance constraint value p_{it}, the projects first work independently and are controlled separately. At each k-th control point t_{ik} of project U_i, given the actual amount already produced, decision-making centers on determining both the next control point $t_{i,k+1}$ and the index j of the new speed v_{ij} to proceed with up to that point, $1 \le j \le m$. The on-line control for each project proceeds either until the next emergency point, or until the due date D.

Rescheduling the remaining system's target amount V_t among the projects is carried out by using heuristic procedures. Determining chance constraint values p_{it} is carried out by using a cyclic coordinate descent method in combination with a two-level simulation model.

The main principal differences between the problems outlined in §§6.2-6.3

and the model under consideration is that:

- in models outlined in Chapter 6 resources (e.g., GRU units) may be rescheduled among the projects while in the case under consideration each project may only vary the level of its intensity by changing the progress of the project's movement to achieve the target, and
- in the model under consideration the control device may reschedule the target amount among the projects. This results in raising the system's flexibility.

The model can be applied, e.g., to such important construction projects like building several derricks (oil-wells) in a new oil-field to reach the oil-field's total desired output (capacity). In the course of carrying out the project, a certain oil-well being for some reasons less effective may get help by lowering its plan target, at the expense of other and more powerful wells. Similar situations may be encountered in the mining industry, e.g., by ore production, etc.

We refer the outlined below model to a mixed type since it combines control actions at emergency moments (and is, in essence, a control model), and determines over time new planning target amounts in the course of monitoring the projects. In our opinion, such a model may be a powerful facilitator for a variety of large-scale innovation projects.

13.3.2 Notation

Let us introduce the following terms:

S - the two-level system composed of n projects U_i, $1 \le i \le n$;

D - the due date (pregiven);

D_t - the length of the remaining planning horizon at moment t, $D_t = D - t$;

F - the actual moment the target amount is completed (a random value);

p - the chance constraint, i.e. the minimal permissible confidence probability of accomplishing the system's plan on time (pregiven);

p_{it} - the chance constraint value for each project U_i determined at the emergency moment $t \ge 0$, $1 \le i \le n$ (to be determined as an optimized variable);

s_{ik} - the index of the speed chosen by the decision-maker at the control point t_{ik};

t_{ik} - the k-th inspection moment (control point) of project U_i, $k = 0,1,\ldots,N_i$;

t_q^{em} - the q-th emergency moment at the system level, $1 \le q \le N_{em}$ (a random value);

N_i - the number of inspection moments for each project U_i;

N_{em} - the number of emergency moments (a random value);

v_{ij} - the j-th speed of project U_i to reach its target, $1 \le j \le m$ (a random value with pregiven density function $f_{ij}(v)$);

$\bar{v}_{ij}$ - the average of speed v_{ij}. It is assumed that for each project U_i speeds v_{i1}, v_{i2}, ... , v_{im} are sorted in ascending order of their average values and are independent of t. Thus, value $\bar{v}_{im}$ is the maximal average speed for project U_i;

V - the pregiven system target (planned program) gauged by a single measure (target amount);

$V^f(t) = \sum_{i=1}^{n} V_i^f(t)$ - the actual system's output observed at moment t (a random value);

V_{it} - the target amount assigned to project U_i at the emergency point t (to be determined); note that $\sum_i V_{it} = V_t$;

$V_i^f(t)$ - the actual output of project U_i observed at moment t, $0 \le t \le D$; $V_i^f(0) = 0$ (a random value);

V_t - the system's remaining target amount at moment t, $V_0 = V$;

$W_p\left[V_i^f(t), V_{it}, j\right]$ - the p-quantile of the moment target amount V_{it} will be completed on conditions that: (a) speed v_{ij} is introduced for project U_i at moment t and will be used throughout, and (b) the actual observed output of project U_i at moment t is $V_i^f(t)$;

m - the number of possible speeds (common to all projects);

d - the minimal time span between two consecutive control points t_{ik} and $t_{i,k+1}$ (pregiven); equal for all projects;

h_i - the search step for determining optimal values p_{it};

Δ - the minimal value of the closeness of inspection moment t_{ik} to the due date D (pregiven and equal for all projects);

a_{ij} - lower bound of random speed v_{ij} (pregiven);

b_{ij} - upper bound of random speed v_{ij} (pregiven);

C - the total operational costs, penalties and charges accumulated for the system in the course of accomplishing the target amount (a random value);

C_{em} - the average cost of rescheduling the remaining target amount V_t among projects U_i by the system at a routine emergency moment $t \ge 0$;

C_{ij} - the average processing cost per time unit of speed v_{ij}, $1 \le i \le n$, $1 \le j \le m$ (pregiven); note that for a fixed i relation $j_1 \le j_2$ results in

$C_{ij_1} < C_{ij_2}$;

C_{ins} - the average cost of performing a single inspection of a project (pregiven, equal for all projects);

$C_i^f(t)$ - the actual accumulated processing and inspection costs calculated at moment t for project U_i, $0 \le t \le D$, $1 \le i \le n$, $C_i^f(0) = 0$;

C^* - the penalty paid to the customer by the system for not accomplishing the target amount on time, i.e. when $F > D$ (a single payment, pregiven);

C^{**} - the penalty cost for each time unit of delay $F - D$ (pregiven);

C^{***} - storage charges per time unit for the target amount's completion before the due date (pregiven).

13.3.3 The control model

A two-level control model is considered where each level faces a stochastic optimization problem [87].

The Problem at the System Level (Problem A)

At each emergency point $t = t_q^{em}$, $1 \le q \le N_{em}$, $t_1^{em} = 0$, determine local production plans V_{it}, $1 \le i \le n$, together with local chance constraints p_{it}, in order to minimize the expected total expenses

$$\min_{\{V_{it}, p_{it}\}} \overline{C} \tag{13.3.1}$$

subject to the chance constraint

$$\Pr\left\{V^f(D) \ge V\right\} \ge p . \tag{13.3.2}$$

Note that random value C satisfies

$$C = \sum_{i=1}^{n} \sum_{k=0}^{N_i - 1} \left[C_{is_{ik}} \left(t_{i,k+1} - t_{ik} \right) \right] + \sum_{i=1}^{n} (N_i - 1) C_{ins} + N_{em} C_{em} + \\ + \left[C^* + C^{**}(F - D) \right] \delta + C^{***} (D - F)(1 - \delta), \tag{13.3.3}$$

where

$$\delta = \begin{cases} 1 & \text{if } F > D \\ 0 & \text{otherwise}, \end{cases} \tag{13.3.4}$$

and values $\{s_{ik}\}$ and $\{t_{ik}\}$ are obtained by solving Problem B at the project level.

Values $\{V_{it}\}$ at each emergency point t, including $t = 0$, are determined according to a widely used heuristic procedure [54,61-64,84,151], namely

$$V_{it} = V_t \frac{\overline{v}_{im}}{\sum_{i=1}^{n} \overline{v}_{im}}, \tag{13.3.5}$$

where $\overline{v}_{im}$ is the *maximal* speed which can be introduced for project U_i.

As to values $\{p_{it}\}$, they are determined by using a cyclic coordinate descent algorithm. The search procedure is carried out by means of simulation, by undertaking numerous realizations of a simulation model at the lower level in order to obtain representative statistics. The simulation model represents the process of manufacturing for several projects U_i with input values $\{V_{it}\}$ and $\{p_{it}\}$, between two adjacent emergency points t_q^{em} and t_{q+1}^{em}. In the case of a routine emergency call the problem at the section level is resolved, new values $\{V_{it}\}$ and $\{p_{it}\}$ are determined, and the manufacturing process proceeds at the lower level, for each project U_i independently.

The Problem at the Project Level (Problem B)

The cost-optimization control model for a single project has been formulated in Chapter 6. We have modified that problem for the case of several projects with additional cost parameters C_{em}, C^*, C^{**} and C^{***}.

For the case of an independent project U_i, given the input values V_{it}, p_{it}, d, Δ and $\bar{v}_{ij}$, $1 \le j \le m$, the problem is to determine both control points $\{t_{ik}\}$ and speeds $\{v_{is_{ik}}\}$ to minimize the expenses

$$J = \min_{\{t_{ik}, v_{is_{ik}}\}} \left\{ \sum_{k=0}^{N_i-1} \left[C_{is_{ik}} \left(t_{i,k+1} - t_{ik} \right) \right] + N_i C_{ins} \right\} \tag{13.3.6}$$

subject to

$$\Pr\left\{ V_i^f(D) \ge V_{it} \right\} \ge p_{it}, \tag{13.3.7}$$

$$t_{i0} = t, \tag{13.3.8}$$

$$t_{iN_i} = \min_{T_i} \left[T_i : \Pr\left\{ V_i^f(T_i) \ge V_{it} \right\} \right], \tag{13.3.9}$$

$$t_{i,k+1} - t_{ik} \ge d, \tag{13.3.10}$$

$$D - t_{ik} \ge \Delta, \quad 0 \le k \le N_i - 1, \tag{13.3.11}$$

$$s_{ik} = j = \min_{1 \le q \le m} q \quad \forall q : W_p\left[V_i^f(t), V_{it}, q \right] \le D. \tag{13.3.12}$$

Restriction (13.3.8) means that after reallocating target amounts at the routine emergency point t, the starting moment to proceed constructing, i.e., the first control point to undertake decision-making and to determine s_{i0} and t_{i1}, is t. Note that at all emergency points the remaining target amount, as well as the due date, are updated, i.e., the ordinate $t = 0$ is shifted to the right. Restriction (13.3.9) means that the last inspection point is the moment target amount V_{it} is reached. Restrictions (13.3.10) and (13.3.11) ensure the closeness between two consecutive control points, as well as the closeness of the routine inspection point to the due date. Restriction (13.3.12) means that the speed to be chosen at any routine control point t_{ik} should not exceed the *minimal speed* which guarantees meeting the deadline D on time, subject to the chance constraint (13.3.7).

The general idea of solving the problem (13.3.6-13.3.12), which is a very complicated stochastic optimization problem, is as follows. At each control point t_{ik} decision-making centers on the assumption (see §6.1) that there is not more than one additional control point before the due date. Two speeds have to be chosen at point t_{ik}:

1. Speed v_{ij_1}, $j_1 = s_{ik}$, which has to be actually introduced at point t_{ik} up to the

next control point $t_{i,k+1}$.

2. Speed v_{ij_2}, $j_2 = s_{i,k+1}$, which is forecast to be introduced at control point $t_{i,k+1}$ within the period $[t_{i,k+1}, D]$.

Thus, j_1 is determined in accordance to (13.3.12) and j_2 is determined by honoring chance constraint (13.3.7). As outlined in Chapter 6, at each routine control point t_{ik} all possible couples are singled out. The couple, which delivers the minimum of forecasted manufacturing and control expenses, has to be chosen. Since couple (j_1, j_2), together with the inspected value $V_i^f(t_{ik})$ and values D and V_{it}, fully determines the next control point $t_{i,k+1}$, speed v_{ij_1} is introduced within the period $[t_{ik}, t_{i,k+1}]$. At moment $t_{i,k+1}$ decision-making has to be carried out anew.

13.3.4 The general idea of the two-level heuristic algorithm

The general idea of the regarded heuristic algorithm is as follows: at each routine emergency point t_q^{em}, $q = 0,1,\ldots,N_{em}$, decision-making centers on minimizing the future costs from point t_q^{em} until F, including the penalty and the storage costs. The costs representing the past (interval $[0, t_q^{em}]$) are irrelevant for this on-line control problem, and there is no need to remember the past decision [63]. The only relevant information to be stored is t_q^{em} and $V_i^f(t_q^{em})$. Thus, decision-making at the system level is carried out only at emergency points t_q^{em} including the moment $t = 0$ the system starts functioning.

Decision-making at the system level at each routine emergency moment $t = t_q^{em}$ centers on determining both new chance constraint values $\{p_{it}\}$ and new target amounts V_{it} for the remaining planning horizon $[t, D]$. Values $\{p_{it}\}$ are obtained by means of simulation, by a combination of a search algorithm and an on-line one-level control algorithm for several projects. The latter work independently and are controlled separately at inspection points. It is generally assumed that at the beginning of the work all the available resources are previously allocated among the projects. Those resources remain unchanged within the planning horizon, i.e. no resource reallocation is performed. Thus, the corresponding speeds v_{ij} for each project U_i remain unchanged too.

If for a certain project U_i at a routine inspection point t_{ik} it is anticipated that the project cannot meet its target V_{it} on time subject to the previously determined chance constraint p_{it}, an emergency is declared, and decision-making is affected at the system level. The remaining target V_t at $t = t_{ik}$, together with the remaining time $D_{t_{ik}} = D - t_{ik}$, is then updated. New quasi-optimal values $\{p_{it}\}$, $t = t_{ik}$, together with new target amounts $\{V_{it}\}$, are then determined. The newly corrected plan is assigned to all projects, and the process proceeds further, until either the new emergency point or until the moment the target amount is completed. Thus, decision-making at the system level centers on numerous recalculations of the system's plan subject to the chance constraint. This is carried out by using a forecasting simulation model with input values $\{V_{it}, p_{it}\}$, $t = t_{ik}$. The matrix $Z = \{V_{it}, p_{it}\}$ which delivers the minimum of total accumulated costs subject to the chance constraint p, is taken as the optimal corrected plan. Afterwards, that corrected plan is passed to the projects, and on-line decision-making is carried out at the project level.

§13.4 Conclusions

The following conclusions can be drawn from the Chapter:

1. The models outlined in Chapter 13 are, in fact, the continuation of various models presented in Chapter 11. The similarity between these two classes of models results in operating both the planning and the monitoring (scheduling) stages of the project's life cycle. The difference stems from the fact that scheduling models outlined in Chapter 11, unlike models of Chapter 13, are focused on feeding-in resources. In the concluded Chapter models are more concentrated on estimating truly planning parameters.
2. Model (13.3.1-13.3.5), unlike other models outlined in the concluded Chapter, cannot be regarded as a *scheduling model* since it operates simultaneously as a *control model* and a *planning model*. In our opinion, this model may benefit in future from a variety of fruitful applications, especially for innovative projects.
3. Model (13.3.1-13.3.5) is a particular case of the general cost-optimization model based on the chance constrained principle (see Chapter 6). The fitness of the outlined algorithm has been checked by means of simulation [87,94,118,151].
4. Although cost-optimization models presented in Chapters 6 and 13 refer to one and the same class of control models, they can be used in different situations. Implementing target amount rescheduling is an attempt to build a bridge between planning and control models. In our opinion, such an attempt is a positive one.

PART V

HIERARCHICAL MODELS FOR PLANNING AND CONTROLLING SEVERAL STOCHASTIC NETWORK PROJECTS

Chapter 14. Hierarchical Model for PERT-COST Projects (Planning Stage)

§14.1 The model's structure

14.1.1 *Introduction*

We will outline a hierarchical on-line control model for several PERT type projects being carried out simultaneously. On the project level, each project is controlled separately in order to minimize the number of control points subject to a chance constraint, which seeks to prevent deviation from the planned trajectory within the planning horizon with pregiven probability. If at a certain control point it is anticipated that the project will not be on target subject to the chance constraint, then an emergency is called and the company level is faced with the problem of reassigning the remaining budget among the projects so that the faster ones may help the slower ones. Thus the model has two objectives: to minimize the number of control points and to maximize the probability that the slowest project can meet its due date on time.

The following realistic assumptions are introduced:

1. Time duration of each activity entering the project is approximately inversely proportional to the budget assigned to that activity [7,53-54,64,68,92].
2. The time-cost curve for the activity with random duration and preset budget assigned to that activity may be determined on the basis of beta or alternative distributions (see Chapter 2).

Two basic concepts are implemented in the outlined model:

A. Decision-making at each control point is based on calculating and examining the probability of meeting the project's due date on time;

B. The on-line control model determines the next control point by solving a stochastic optimization problem: to minimize the number of control points under a chance constraint not to deviate from the planned trajectory. Such a constraint is, in essence, stricter than using confidence probabilities to meet the due date on time.

We will consider a hierarchical control model and will describe the mathematical formulations of all optimization problems that are imbedded in the model. The solutions to the problems enable control actions to be taken on different levels to meet the projects' due dates on time.

14.1.2 Notation

Let us introduce the following terms:

The Company Level

$G_k(N,A)$ - the k-th stochastic network project (graph) of PERT-COST type, $1 \le k \le n$;

n - the number of projects;

G_{kt} - the remaining k-th network project at moment $t \ge 0$; $G_{k0} = G_k(N,A)$;

D_k - the due date for the k-th project (pregiven);

p_k^{**} - probability which practically guarantees completion of the k-th project on time (pregiven);

$p_k^* < p_k^{**}$ - the least permissible probability for the k-th project to be completed on time (pregiven); both values p_k^* and p_k^{**} have to be set by practitioners using expert methods; it may be considered, if not otherwise stated by the company management, that for two different projects with equal priority indices their corresponding confidence probabilities p_k^* and p_k^{**} will be equal too;

C_{kt} - the budget assigned by the company to the k-th project at moment $t \ge 0$;

$C \ge \sum_{k=1}^{n} C_{k0}$ - the total budget for n projects at the company's disposal;

$C_k(t)$ - available remaining budget to carry out project G_{kt} which is observed at control point $t > 0$; $C_k(0) = C_{k0}$;

$p_k[C_{kt}]$ - probability to accomplish the remaining project G_{kt} on time corresponding to the allocated budget value C_{kt}, $t \ge 0$;

C_{kt}^* - budget value satisfying $p_k[C_{kt}^*] = p_k^*$, $t \ge 0$;

C_{kt}^{**} - budget value satisfying $p_k[C_{kt}^{**}] = p_k^{**}$;

$T_k[C_{kt}]$ - random duration of project G_{kt} corresponding to the budget value C_{kt}. Note that obvious relation $p_k[C_{kt}] = \Pr\{t + T_k[C_{kt}] < D_k\}$ holds;

$T_k[C_k(t)]$ - random duration of project G_{kt} with the remaining budget $C_k(t)$;

$p_{kt} = \Pr\{t + T_k[C_k(t)] \le D_k\}$ - probability to accomplish project G_{kt} on time with budget $C_k(t)$;

$C_k^*(t)$ - budget value satisfying $p_{kt} = p_k^*$;

$C_k^{**}(t)$ - budget value satisfying $p_{kt} = p_k^{**}$;

δC - minimal budget unit value by which budget C_{kt} may be changed;

Δp - minimal probability unit value by which confidence probability may be changed;

η_k - priority value of the k-th project (pregiven); note that if $G_{k_1}(N,A)$ is of higher importance than $G_{k_2}(N,A)$, relation $\eta_{k_1} > \eta_{k_2}$ holds.

The Project Level

$(i,j)_k$ - activity leaving node i and entering node j, $(i,j)_k \in G_{kt}$, $t \ge 0$;

$c(i,j)_k$ - budget assigned to activity $(i,j)_k$;

$c(i,j)_{k\min}$ - minimal possible budget to carry out activity $(i,j)_k$ (pregiven);

$c(i,j)_{k\max}$ - maximal budget required to carry out activity $(i,j)_k$ (pregiven); in case $c(i,j)_k > c(i,j)_{k\max}$ additional budget is redundant;

$t(i,j)_k$ - random duration of activity $(i,j)_k$; it is assumed that $(i,j)_k$ has a beta-distribution with density function:

$$p_{ij}(x)_k = \frac{12}{\left[b(i,j)_k - a(i,j)_k\right]^4}\left[x - a(i,j)_k\right]\left[b(i,j)_k - x\right]^2; \qquad (14.1.1)$$

$A(i,j)_k$ - pregiven value to satisfy

$a(i,j)_k = \dfrac{A(i,j)_k}{c(i,j)_k}$ which is the lower bound for random value $t(i,j)_k$;

$B(i,j)_k$ - pregiven value to satisfy

$b(i,j)_k = \dfrac{B(i,j)_k}{c(i,j)_k}$ which is the upper bound for random value $t(i,j)_k$;

On-line Project Control Level

N_k - the number of control points for project $G_k(N,A)$ (on-line control);

$N_k(t)$ - the number of future control points for project $G_k(N,A)$ beginning at moment t;

t_{kg} - the g-th control point for the k-th project, $g = 0,1,\dots,N_k$, $1 \le k \le n$, $t_{k0} = 0$;

Δ_k - the minimal pregiven time span between two adjacent control points t_{kg} and $t_{k,g+1}$ (for practical reasons and in order to force convergence);

$V_k^f(t)$ - state variable of project G_{kt} observed at control point t;

$V_k^{pl}(t)^{(q)}$ - planned trajectory curve between two adjacent control points (the q-th iteration) .

Assume that various projects $G_k(N,A)$ are of different importance. Thus, a priority index (value) η_k has to be set for each project by the management. The management may use for this purpose various expert methods such as the Delphi method [149], and take into consideration qualitative and quantitative properties, e.g., profit expectations, cash flow advantages, international trends, innovation, strategic issues, etc. After considering all the above mentioned issues, the company level has to define η_k. The level of significance of each project can also practically be specified by the project delivery performance. For projects with random activity durations delivery performance is nothing else but the probability of the project to meet its due date on time.

§14.2 Budget allocation among several projects with different priorities

Consider that the company management is faced with controlling n PERT-COST type network projects $G_k(N,A)$, $1 \le k \le n$, which have to be carried out simultaneously. Projects are of different importance and significance; for each project the corresponding priority index η_k is externally pregiven. The total budget C at the company's disposal to carry out all the projects is limited. Thus, the company is faced with the problem of optimal budget allocation among n network projects under consideration. This problem is to be solved [7,53-54,64,92]:

(a) once at the planning stage, at $t=0$, i.e., before the projects' realization, and
(b) repeatedly in the course of the projects' realization, at $t>0$, when an emergency is called by one of the projects due to its deviation from the planned trajectory. In the latter case the remaining budget of all the unaccomplished projects is to be reallocated.

Following is the solution of the general problem for the case $t \geq 0$. Given for each project G_{kt}, $1 \leq k \leq n$,

(a) desirable and least permissible confidence probabilities p_k^{**} and p_k^{*},
(b) priority values η_k,

- the problem is to determine optimal values C_{kt}, $1 \leq k \leq n$, to maximize the objective

$$J_1 = \max_{C_{kt}} \sum_{k=1}^{n} p_k[C_{kt}] \cdot \eta_k \tag{14.2.1}$$

subject to

$$\sum_{k=1}^{n} C_{kt} = \sum_{k=1}^{n} C_k(t), \tag{14.2.2}$$

$$p_k^{*} \leq p_k[C_{kt}] = \Pr\{t + T_k[C_{kt}] \leq D_k\} \leq p_k^{**}. \tag{14.2.3}$$

Note that maximizing objective (14.2.1) means that the management first takes all measures to accomplish on time projects with higher priorities and afterwards handles less important projects.

Problem (14.2.1-14.2.3) is a stochastic optimization problem with very complicated non-linear convolutions $p_k[C_{kt}]$. In order to simplify the problem we assume that probability value $p_k[C_{kt}]$ depends on budget value C_{kt} linearly, i.e., for each k-th project, relation

$$\frac{p_k[C_{kt}^{'''}] - p_k[C_{kt}^{'}]}{C_{kt}^{'''} - C_{kt}^{'}} = \frac{p_k[C_{kt}^{''}] - p_k[C_{kt}^{'}]}{C_{kt}^{''} - C_{kt}^{'}} = \rho_{kt} \tag{14.2.4}$$

holds for any $C_{kt}^{'''} > C_{kt}^{''} > C_{kt}^{'}$, ρ_{kt} being a constant value at a fixed moment t. It goes without saying that values ρ_{kt} may change from project to project; but within the project at a fixed moment $t \geq 0$ they remain unchanged.

To solve problem (14.2.1-14.2.3) we have to solve an auxiliary problem as follows:

For each project G_{kt}, $1 \leq k \leq n$, separately, determine two budget values C_{kt}^{**} and C_{kt}^{*}, to satisfy

$$\Pr\{t + T_k[C_{kt}^{*}] \leq D_k\} = p_k^{*}, \tag{14.2.5}$$

$$\Pr\{t + T_k[C_{kt}^{**}] \le D_k\} = p_k^{**}. \qquad (14.2.6)$$

The solution of this problem will be outlined below, at the project level.

After solving problem (14.2.5-14.2.6) and determining values C_{kt}^{**} and C_{kt}^{*}, $1 \le k \le n$, we solve at the company level the budget reallocation problem. The procedure to solve this problem is as follows:

Compare values $\sum_{k=1}^{n} C_k(t)$ with $\sum_{k=1}^{n} C_{kt}^{*}$ and $\sum_{k=1}^{n} C_{kt}^{**}$; if $\sum_{k=1}^{n} C_k(t) < \sum_{k=1}^{n} C_{kt}^{*}$ problem (14.2.1-14.2.3) has no solution. We have either to reduce the desired confidence probabilities p_k^{*}, or to cancel one of the least important projects, or to ask for additional budget $\Delta C = \sum_{k=1}^{n} C_{kt}^{*} - \sum_{k=1}^{n} C_k(t)$. Such a trade-off is the sole prerogative of the company management.

In case $\sum_{k=1}^{n} C_{kt}^{**} < \sum_{k=1}^{n} C_k(t)$ the solution of the problem is $C_{kt} = C_{kt}^{**}$. Values C_{kt}^{**} must be allocated to project G_{kt}, $1 \le k \le n$, while the remaining budget $\sum_{k=1}^{n} C_k(t) - \sum_{k=1}^{n} C_{kt}^{**}$ may be used for other company activities.

Case $\sum_{k=1}^{n} C_{kt}^{*} < \sum_{k=1}^{n} C_k(t) < \sum_{k=1}^{n} C_{kt}^{**}$ means that, in addition to the minimal budget values C_{kt}^{*}, the remaining budget $\sum_{k=1}^{n} C_k(t) - \sum_{k=1}^{n} C_{kt}^{*}$ has to be reallocated among the projects according to objective (14.2.1). The thus determined optimal solution is as follows:

Since value $p_k[C_{kt}]$ depends on C_{kt} linearly, values $p_k[C_{kt}]$, p_k^{*} and p_k^{**} satisfy

$$\frac{p_k[C_{kt}] - p_k^{*}}{p_k^{**} - p_k^{*}} = \frac{C_{kt} - C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}}, \qquad (14.2.7)$$

and

$$p_k[C_{kt}] = p_k^{*} + \frac{C_{kt} - C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \left(p_k^{**} - p_k^{*}\right). \qquad (14.2.8)$$

Substituting $p_k[C_{kt}]$ in (14.2.1) for (14.2.8) we obtain

$$J_1 = \max_{C_{kt}} \sum_{k=1}^{n} \left\{ C_{kt} \cdot \left[\frac{p_k^{**} - p_k^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k \right] + \left[\frac{p_k^{*} C_{kt}^{**} - p_k^{**} C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k \right] \right\}. \qquad (14.2.9)$$

Taking into account that t is fixed and denoting

$$\frac{p_k^{**} - p_k^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k = a_k, \quad \frac{p_k^{*} C_{kt}^{**} - p_k^{**} C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k = b_k,$$

we substitute objective (14.2.1) for

$$J_1 = \max_{C_{kt}} \sum_{k=1}^{n} \left(a_k C_{kt} + b_k\right) \qquad (14.2.10)$$

subject to

$$\sum_{k=1}^{n} C_{kt} = \sum_{k=1}^{n} C_k(t), \tag{14.2.11}$$

$$C_{kt}^* \le C_{kt} \le C_{kt}^{**}, \; 1 \le k \le n . \tag{14.2.12}$$

Since b_k does not depend on C_{kt}, objective (14.2.10) can be simplified, namely:

$$J_1 = \max_{C_{kt}} \sum_{k=1}^{n} a_k C_{kt} \tag{14.2.13}$$

subject to (14.2.11-14.2.12) .

Taking into account $a_k > 0$, $1 \le k \le n$, the optimal analytical solution of the regarded problem can be obtained by means of a step-wise algorithm as follows:

Step 1. Assign to all projects G_{kt}, $1 \le k \le n$, their minimal budget values C_{kt}^*; denote the remaining budget $\sum_{k=1}^{n} C_k(t) - \sum_{k=1}^{n} C_{kt}^* = \Delta C$.

Step 2. Reorder sequence $\{a_k\}$ in descending order; let their new ordinal numbers be $f_1, f_2, ..., f_n$.

Step 3. Set $j = 1$.

Step 4. Calculate $\gamma_j = \min\left\{\left(C_{f_j t}^{**} - C_{f_j t}^*\right), \Delta C\right\}$.

Step 5. Determine for project $G_{f_j t}$ its final budget $C_{f_j t} = C_{f_j t}^* + \gamma_j$.

Step 6. Update the remaining budget $\Delta C - \gamma_j \Rightarrow \Delta C$. If $\Delta C = 0$ go to Step 9. Otherwise apply the next step.

Step 7. Set $j + 1 \Rightarrow j$.

Step 8. If $j \le n$ return to Step 4. Otherwise apply the next step.

Step 9. The algorithm terminates.

It can be well-recognized that since sequence $\left\{a_{f_j}\right\}$ is a descending one determining the optimal solution results in assigning to each routine project G_{f_j} as much additional budget from the remaining company budget ΔC as possible. Thus, the algorithm develops the optimal solution under the assumptions of linearity of $p_k[C_{kt}]$. It can be proven that in the course of optimal budget reallocation all the projects, *besides not more than one*, will obtain either values C_{kt}^* or C_{kt}^{**}.

Assertion. There exists not more than one project G_{ft} for which $C_{ft}^* < C_{ft} < C_{ft}^{**}$ holds. For all other projects G_{qt}, $q \in \{k\} \setminus f$, C_{qt} is equal either to C_{qt}^* or to C_{qt}^{**}.

Proof. Assume that after *optimal* budget reallocation there are two *different* projects G_{rt} and G_{st} with intermediate values $C_{rt}^* < C_{rt} < C_{rt}^{**}$ and $C_{st}^* < C_{st} < C_{st}^{**}$. Assume, further, that $a_r > a_s$. Calculate budget value $\varepsilon_C = \min\left[\left(C_{rt}^{**} - C_{rt}\right), \left(C_{rt} - C_{rt}^*\right)\right]$ and transfer value ε_C from project G_{st} to project G_{rt}. It can be well-recognized that in the course of such a reallocation:

(a) objective (14.2.13) will increase;

(b) among the two projects under consideration no more than one will remain with the intermediate budget value; the other one will obtain either value C_{kt}^{**}, or C_{kt}^{*}.

Thus the former budget reallocation was not the optimal one. ■

After undertaking budget reallocation among the projects the latter have to be controlled at the project level.

§14.3 Projects of equal significance

In this paragraph the case of several stochastic PERT-COST projects of equal importance, i.e., with equal priority values, will be considered. The idea of optimal budget reallocation among those projects, unlike the case outlined in §14.2, is based on the conceptions which have been outlined in [63] and are as follows:

If a company operates PERT-COST projects with different importance, the management takes all measures to raise the performance of projects with higher priorities. This results in control policy to supply the maximal possible amount of resources to projects of higher significance and to leave the minimal permissible resources to be utilized for less important projects. But in the case of projects which have equal importance the performance of the *slowest* project will determine the performance of the whole group of projects under consideration. *Thus, the conception is to maximize the ability of the slowest project at the expense of the faster ones.*

Let us introduce for each stochastic network project G_{kt}, $t \geq 0$, the term which we will henceforth call "the project's performance degree". It can be calculated at any routine control point t and is equal to $p_k[C_{kt}]$, i.e., it is the probability $\Pr\{t + T_k[C_{kt}] < D_k\}$ of completing the project on time. According to the conception outlined above the slowest project's performance degree determines the possibility for the company to realize a group of projects within their due dates. Thus, the objective to be maximized is as follows [64]:

$$J_2 = \max_{C_{kt}} \min_{k} p_k[C_{kt}], \; t \geq 0, \tag{14.3.1}$$

subject to

$$\sum_{k=1}^{n} C_{kt} = \sum_{k=1}^{n} C_k(t), \tag{14.3.2}$$

$$p_k^* \leq p_k[C_{kt}] \leq p_k^{**}, \;\; 1 \leq k \leq n. \tag{14.3.3}$$

The problem is to be solved at moment $t = 0$ or to be repeatedly resolved at every emergency moment $t > 0$. If for any project G_{kt} at any control point t it is anticipated that the project will fail to reach its due date on time with probability not less than p_k^*, an emergency is called and at the company level the remaining total budget is to be reassigned among the unaccomplished projects so that the faster one can contribute and speed up the slower one.

To solve optimization problem (14.3.1-14.3.3) the same assumption will be introduced as for the case outlined in §14.2, i.e., that $p_k\left[C_{kt}\right]$ depends on C_{kt} linearly.

Values C_{kt}^{*} and C_{kt}^{**} corresponding to confidence probabilities p_k^{*} and p_k^{**}, have to be calculated for each project G_{kt}, $1 \le k \le n$, $t \ge 0$.

The heuristic procedure to obtain values C_{kt}^{*} and C_{kt}^{**} will be presented in §14.4.

The solution of problem (14.3.1-14.3.3) is outlined below.

Using (14.2.7) and (14.2.8) and substituting $p_k\left[C_{kt}\right]$ in (14.3.1) for (14.2.8) we obtain

$$J_2 = \max_{C_{kt}}\left\{\min_k\left[p_k^{*} + \frac{C_{kt} - C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \left(p_k^{**} - p_k^{*}\right)\right]\right\}. \tag{14.3.4}$$

Denoting $\frac{p_k^{**} - p_k^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k = \alpha_k$, $\frac{p_k^{*}C_{kt}^{**} - p_k^{**}C_{kt}^{*}}{C_{kt}^{**} - C_{kt}^{*}} \cdot \eta_k = \beta_k$, we obtain optimization problem as follows:

Maximize

$$J_2 = \max_{C_{kt}}\left\{\min_k\left[\alpha_k C_{kt} + \beta_k\right]\right\} \tag{14.3.5}$$

subject to (14.3.2) and

$$C_{kt}^{*} \le C_{kt} \le C_{kt}^{**}. \tag{14.3.6}$$

A substitution

$$\min_k\left[\alpha_k C_{kt} + \beta_k\right] = Z \tag{14.3.7}$$

modifies problem (14.3.2, 14.3.5-14.3.6) to the following one:

$$\max_{C_{kt}} Z \tag{14.3.8}$$

subject to (14.3.2), (14.3.6) and

$$Z \le \alpha_k C_{kt} + \beta_k, \; 1 \le k \le n. \tag{14.3.9}$$

Problem (14.3.2, 14.3.6, 14.3.8-14.3.9) can be solved by using linear programming. We rely on a standard software package, LINDO [144] as the computational tool. With the algorithm outlined above and by using standard personal computers budget reallocation can be easily performed by any project management.

In conclusion, it has to be pointed out that in case $\sum_{k=1}^{n} C_k(t) > \sum_{k=1}^{n} C_{kt}^{**}$ the total budget is to be decreased by value $\Delta C = \sum_{k=1}^{n} C_k(t) - \sum_{k=1}^{n} C_{kt}^{**}$ which will be at the disposal of the management for other purposes or projects .

Case $\sum_{k=1}^{n} C_k(t) < \sum_{k=1}^{n} C_{kt}^{*}$ is similar to that outlined in §14.2, i.e., problem (14.3.1-14.3.3) has no solution.

For projects of equal significance and with close due dates it is reasonable to set equal confidence probabilities p_k^* and p_k^{**}. It is also recommended to increase for projects with earlier due dates the least permissible confidence probabilities p_k^* relative to projects with tardy deadlines. Such a policy may prevent failures to execute projects on time.

§14.4 Optimal budget reassignment for a PERT project

After obtaining budgets C_{kt}, $1 \le k \le n$, from the company (see §§14.2-14.3) each project is carried out independently, until either the due date D_k, or an emergency call for reallocating the remaining budget among the unaccomplished projects. Several important standard problems are to be solved at the project level.

The first problem deals with optimal budget reallocation among the project's activities to maximize the probability of meeting the project's due date on time. This problem is solved independently for each project and therefore in order to simplify the problem's terms we shall omit the project's index.

The problem is as follows [7,53-54,64,92]:

Determine optimal values $c(i,j)$ to maximize the objective

$$\max_{\{c(i,j)\}} p(C) \tag{14.4.1}$$

subject to

$$c(i,j)_{\min} \le c(i,j) \le c(i,j)_{\max}, \tag{14.4.2}$$

$$\sum_{(i,j)\in G(N,A)} c(i,j) = C. \tag{14.4.3}$$

Here C is the available budget assigned to project $G(N,A)$ and $p(C)$ is its probability to be accomplished on time. Note that value C may be either the budget which has been allocated at the company level or the remaining budget which has been observed at a control point.

Problem (14.4.1-14.4.3) is a complicated stochastic optimization problem which can be solved only by using heuristic procedures. Various variants of the heuristic to solve the problem are outlined in [7,53-54,62,64,92] and can be applied to PERT type projects only.

The step-by-step procedure is as follows:

<u>Step 1</u>. By any means reassign budget C among the project's activities $(i,j)\in G(N,A)$ subject to $c(i,j)_{\min} \le c(i,j) \le c(i,j)_{\max}$ and $\sum_{(i,j)\in G(N,A)} c(i,j) = C$ to obtain a feasible solution of the problem. It is suggested to realize the step by using the bisection method [153] as follows:

<u>1.1</u> Start with $\alpha = 0$, $\beta = 1$;

<u>1.2</u> Determine two values:

$$\Sigma_1 = \sum_{(i,j)\in G(N,A)} c(i,j)_{\min} = \sum_{(i,j)} \left[(1-\alpha)\cdot c(i,j)_{\min} + \alpha\cdot c(i,j)_{\max}\right],$$

$\Sigma_2 = \sum_{(i,j)\in G(N,A)} c(i,j)_{\max} = \sum_{(i,j)} \left[(1-\beta)\cdot c(i,j)_{\min} + \beta \cdot c(i,j)_{\max}\right].$

1.3 Calculate value

$$\Sigma_3 = \frac{\Sigma_1 + \Sigma_2}{2} = \sum_{(i,j)} \left[\left(1 - \frac{\alpha+\beta}{2}\right)\cdot c(i,j)_{\min} + \frac{\alpha+\beta}{2}\cdot c(i,j)_{\max}\right].$$

1.4 Compare values Σ_1 and Σ_2. If $\Sigma_2 - \Sigma_1 < \delta C$ go to 1.8; otherwise proceed to 1.5. Note that δC is a pregiven budget unit value.

1.5 Examine relation $\Sigma_1 \le C \le \Sigma_3$. If it holds go to 1.6; otherwise apply Substep 1.7.

1.6 Set

$\Sigma_3 = \Sigma_2$, $1 - \frac{\alpha+\beta}{2} = 1-\beta$, $\frac{\alpha+\beta}{2} = \beta$. Return to Substep 1.3.

1.7 Applying 1.7 means that $\Sigma_3 < C < \Sigma_2$ holds. Set

$\Sigma_3 = \Sigma_1$, $1 - \frac{\alpha+\beta}{2} = 1-\alpha$, $\frac{\alpha+\beta}{2} = \alpha$, and return to Substep 1.3.

1.8 Value $\Sigma_3 \cong C$ with

$c(i,j) = \frac{2-\alpha-\beta}{2}\cdot c(i,j)_{\min} + \frac{\alpha+\beta}{2}\cdot c(i,j)_{\max}$ is the feasible solution.

Step 2. Calculate $a(i,j) = \frac{A(i,j)}{c(i,j)}$ and $b(i,j) = \frac{B(i,j)}{c(i,j)}$ for all activities $(i,j)\in G(N,A)$.

Step 3. Simulate values $t(i,j)$ with density function (14.1.1).

Step 4. Calculate the critical path length $L_{cr}[t(i,j)]$ and determine all activities $(i,j)\in G(N,A)$ which belong to the critical path.

Step 5. Compare values D and $L_{cr}[t(i,j)]$. If $D \ge L_{cr}[t(i,j)]$ counter $W+1 \Rightarrow W$ works; then go to Step 6. In case $D < L_{cr}[t(i,j)]$ apply Step 6 directly.

Step 6. If a routine activity (i,j) belongs to the critical path counter $W_{ij}+1 \Rightarrow W_{ij}$ works. The step is implemented for all $(i,j)\in G(N,A)$.

Step 7. Repeat Steps 2-6 M times in order to obtain representative statistics.

Step 8. Calculate the average value

$p(C)^{(q)} = \frac{W}{M}$, where q is the number of the current iteration.

Step 9. Compare two adjacent average values $p(C)^{(q)}$ and $p(C)^{(q-1)}$. If $p(C)^{(q)} > p(C)^{(q-1)}$ holds, proceed to the next step. Otherwise apply Step 16.

Step 10. Calculate the frequency of each activity (i,j) of being on the critical path (on the basis of M simulations carried out on Step 7). Denote those frequencies by $\overline{p}(i,j/L_{cr})$, $\overline{p}(i,j/L_{cr}) = \frac{W_{ij}}{M}$.

Step 11. Reschedule all the activities (i,j) as follows:

- For activities (i,j) with $\overline{p}(i,j/L_{cr}) > 0$ reschedule them in *de-*

scending order of the product

$\overline{p}(i, j/L_{cr}) \cdot \upsilon_{ij}$, (14.4.4)

where

$$\upsilon_{ij} = \frac{3A(i,j) + 2B(i,j)}{5c(i,j)_{\min} \cdot c(i,j)_{\max}}. \quad (14.4.5)$$

- Activities (i,j) with $\overline{p}(i,j/L_{cr}) = 0$ have to be rescheduled at the end of the schedule *in descending order of values* υ_{ij} *only* since the product $\overline{p}(i,j/L_{cr}) \cdot \upsilon_{ij}$ equals zero.

Step 12. Determine activity (i_ξ, j_ξ) with the *highest order* for which relation $Z_1 = c(i_\xi, j_\xi)_{\max} - c(i_\xi, j_\xi) > 0$ holds. It goes without saying that activity (i_ξ, j_ξ) is placed at the beginning of the schedule and refers to the critical zone, $\overline{p}(i_\xi, j_\xi/L_{cr}) > 0$.

Step 13. Determine activity (i_η, j_η) with the *lowest order* for which relation $Z_2 = c(i_\eta, j_\eta) - c(i_\eta, j_\eta)_{\min} > 0$ holds. Activity (i_η, j_η) is at the end of the schedule and is a non-critical activity, which has practically no influence on the project's duration.

Step 14. Reassign cost values $Z = \min(Z_1, Z_2)$ from activity (i_η, j_η) to activity (i_ξ, j_ξ).

Step 15. Clear counter W and return to Step 2.

Step 16. Introduce changes in the heuristic procedure as follows:

(a) *in Step 9*: for the case $p(C)^{(q)} \le p(C)^{(q-1)}$ instead of Step 16, proceed to Step 18;

(b) *in Step 14*: value Z to be transferred from activity (i_η, j_η) to (i_ξ, j_ξ) is to be set equal to 1. Afterwards apply Step 17.

Step 17. Take the rescheduled activities (i,j) arranged at Step 11 for the $(q-1)$-th iteration. Continue to Step 12.

Step 18. End of the heuristic procedure. Further application of the algorithm will not lead to any increase of the confidence probability.

Values $c(i,j)$ obtained in the course of the $(q-1)$-th iteration are considered as the optimal ones. The optimal value of the objective, i.e., the maximal confidence probability, is value $p(C)^{(q-1)}$ calculated on Step 8.

In conclusion, it can be well-recognized that in cases $C < \sum_{(i,j)\in G(N,A)} c(i,j)_{\min}$ and $C > \sum_{(i,j)\in G(N,A)} c(i,j)_{\max}$ the corresponding confidence probabilities are 0 and 1, i.e., the problem obtains trivial solutions.

In case $\sum_{(i,j)\in G(N,A)} c(i,j)_{\min} < C < \sum_{(i,j)\in G(N,A)} c(i,j)_{\max}$ the heuristic procedure outlined above is to be used. It will be henceforth referred to as **Procedure I**.

§14.5 The dual problem: Determining budget value corresponding to preset confidence probability

The dual problem will be formulated and solved for PERT type projects. The problem under consideration is as follows:

Given a PERT project $G(N,A)$ together with confidence probability p to meet the due date D on time, determine optimal values $c(i,j)$, $(i,j)\in G(N,A)$, to minimize the objective

$$\min_{\{c(i,j)\}}\left\{C=\sum_{(i,j)\in G(N,A)} c(i,j)\right\} \tag{14.5.1}$$

subject to

$$p(C)=\Pr\{T(C)\le D\}=p, \tag{14.5.2}$$

$$c(i,j)_{\min}\le c(i,j)\le c(i,j)_{\max}, \tag{14.5.3}$$

where $T(C)$ is the random project's duration with assigned budget C.

It can be well-recognized that problem (14.5.1-14.5.3) is, in essence, a dual problem for the direct one, (14.4.1-14.4.3). The heuristic solution outlined below is based on the heuristic procedure outlined in §14.4. Unfortunately, problem (14.5.1-14.5.3) is a stochastic optimization problem which due to non-linear constraints cannot be solved in the general case. Thus only heuristics can be applied to obtain an approximate solution.

The step-by-step heuristic procedure is as follows:

Step 1. Choose budget value
$$C_1=\sum_{(i,j)\in G(N,A)} c(i,j)_{\min}.$$

Step 2. Choose budget value
$$C_2=\sum_{(i,j)\in G(N,A)} c(i,j)_{\max}.$$

Step 3. Calculate
$$C_3=\frac{C_1+C_2}{2}=0.5\cdot\sum_{(i,j)\in G(N,A)}\left[c(i,j)_{\min}+c(i,j)_{\max}\right].$$

Step 4. Solve optimization problem (14.4.1-14.4.3) for values $C=C_1$, $C=C_2$ and $C=C_3$. Denote the determined probability values by $\overline{p}$, $\overline{\overline{p}}$ and $\overline{\overline{\overline{p}}}$, correspondingly.

Step 5. Compare values $\overline{p}$ and $\overline{\overline{p}}$. If $\overline{\overline{p}}-\overline{p}<\Delta p$, go to Step 9. Otherwise apply Step 6. Here Δp is the pregiven minimal value by which a confidence probability can be increased or decreased.

Step 6. Examine relation $\overline{p}\le p\le\overline{\overline{\overline{p}}}$. If it holds, proceed to Step 7. Otherwise apply Step 8. Note that relation $\overline{p}\le p\le\overline{\overline{p}}$ is an evident one since $\overline{p}$ and $\overline{\overline{p}}$ are the minimal and maximal confidence probabilities for the project to meet its deadline, correspondingly.

Step 7. Set $C_3\Rightarrow C_2$,

$$\frac{c(i,j)_{\min} + c(i,j)_{\max}}{2} \Rightarrow c(i,j)_{\max}$$. Return to Step 3.

Step 8. Set $C_3 \Rightarrow C_1$,

$$\frac{c(i,j)_{\min} + c(i,j)_{\max}}{2} \Rightarrow c(i,j)_{\min}$$. Return to Step 3.

Step 9. Value $C = C_3$ represents the minimal budget value to be determined while values $c(i,j)$ obtained at Step 4 when solving optimization problem (14.4.1-14.4.3) for $C = C_3$ are the optimal ones.

It can be well-recognized that problem (14.5.1-14.5.3) is solved by using the bisection method [153] in combination with the heuristic Procedure I outlined in §14.4.

The outlined heuristic procedure delivering a solution to the inverse problem (14.5.1-14.5.3) will be henceforth referred to as **Procedure III**.

Chapter 15. Hierarchical On-Line Control Model for PERT-COST Projects (Control Stage)

§15.1 The control model

In Chapter 14 we have developed planning and control techniques but we have not considered on-line control procedures. In this chapter we shall develop a hierarchical on-line control model including several PERT type projects of one and the same importance.

The following hierarchical control model is outlined [64]:

At any control point $t \ge 0$ to determine:

- optimal budget values C_{kt} assigned to each project G_{kt} of PERT type, $1 \le k \le n$,
- optimal budget values $c(i,j)_k$ assigned to activities $(i,j)_k \in G_{kt}$,
- optimal control points t_{kg} to inspect project G_{kt},

in order to minimize the total number of future control points $N_k(t)$ for all projects G_{kt}

$$\min_{\{C_{kt}\},\{t_{kg} \ge t\},\{c(i,j)_k\}} \sum_{k=1}^{n} N_k(t), \tag{15.1.1}$$

and to maximize the performance degree of the slowest project

$$\max_{\{C_{kt}\},\{t_{kg} \ge t\},\{c(i,j)_k\}} \left[\min_k \Pr\{t + T_k[C_{kt}] \le D_k\} \right], \tag{15.1.2}$$

subject to

$$p_k^{**} \ge p_k[C_{kt}] = \Pr\{t + T_k[C_{kt}] \le D_k\} \ge p_k^*, \tag{15.1.3}$$

$$\sum_{k=1}^{n} C_{kt} = \sum_{k=1}^{n} C_k(t), \tag{15.1.4}$$

$$\sum_{(i,j)_k \in G_{kt}} c(i,j)_k = C_{kt}, \tag{15.1.5}$$

$$c(i,j)_{k\min} \le c(i,j)_k \le c(i,j)_{k\max}, \tag{15.1.6}$$

$$t_{k,g+1} - t_{kg} \ge \Delta_k, \ 1 \le k \le n, \tag{15.1.7}$$

$$t_{k0} = 0, \tag{15.1.8}$$

$$t_{kN_k} = D_k. \tag{15.1.9}$$

Thus, objective (15.1.2) enables the slower projects to obtain help from the faster ones in the course of the projects' realization. Objective (15.1.1) is evident since project inspection is a costly operation.

Problem (15.1.1-15.1.9) is a stochastic optimization problem with two conflicting objectives and a variable number of constraints. The problem cannot be solved in the general case and allows only heuristic solutions. The general control model can be modified to the hierarchical on-line control model that is presented in Fig. 15.1 and comprises three optimization problems. Problem I, at the company level, enables optimal budget reassignment among the projects. The problem's solution, i.e., the budget assigned to each project, serves as the initial data for Problem II (at the project level), where budget is redistributed among the project's activities to maximize the probability of meeting the project's deadline.

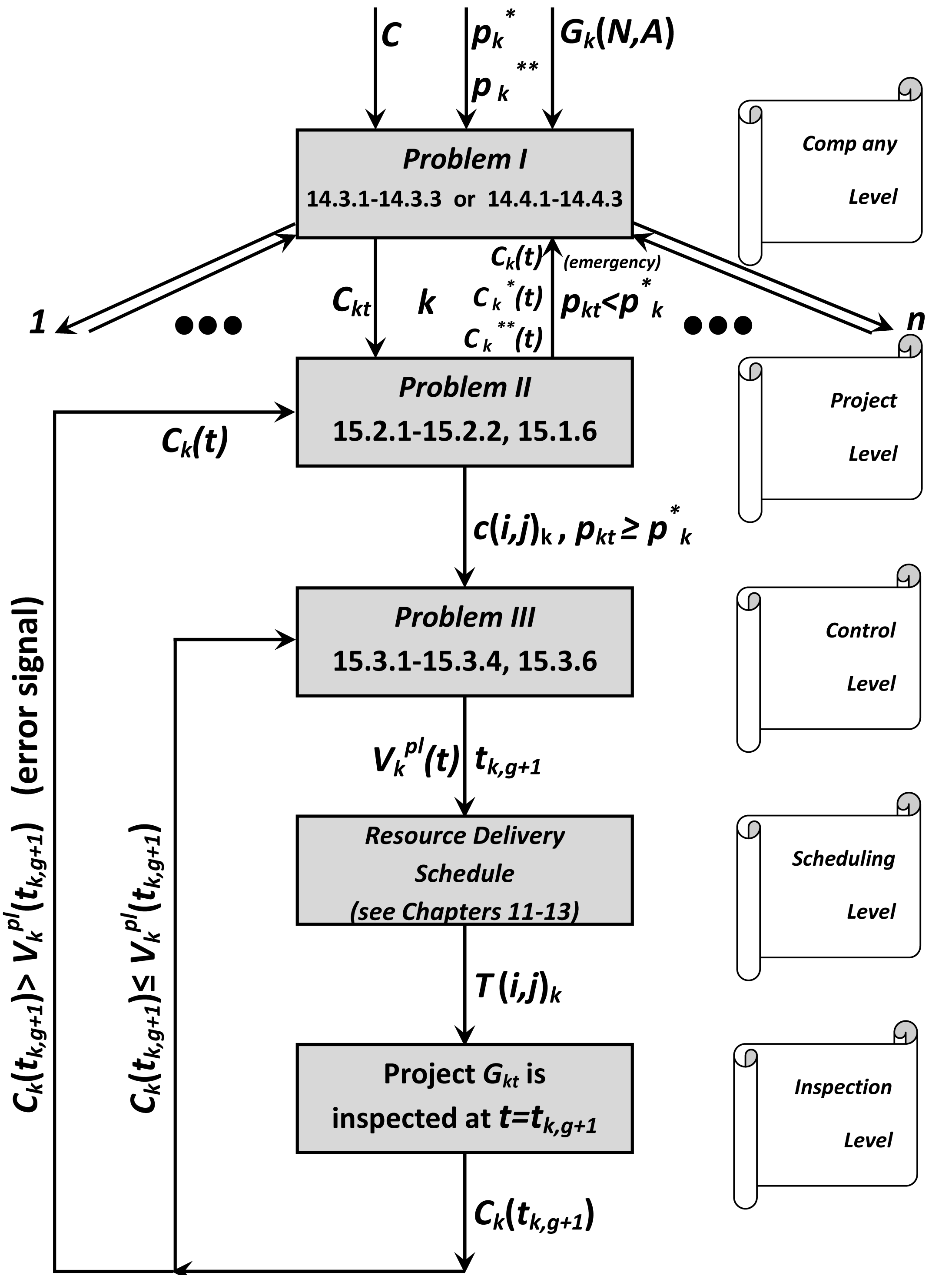

Figure 15.1. *Hierarchical on-line control model* (*emergency at* $t = t_{k,g+1}$)

The solution of Problem II serves, in turn, as the initial data for Problem III, which carries out on-line control, i.e., determines optimal control points t_{kg} to inspect the progress of the project. This is done by determining the planned trajectories that must be repeatedly corrected in the course of the project's realization. The chance constraint to meet the target on time with probability not less than p_k^* is substituted for a stricter one, namely, not to deviate from the planned trajectory at *any* moment $t \in [t_{kg}, D_k]$ with a probability not less than p_k^*. If, at any control point $t = t_{kg}$, $1 \le k \le n$, $1 \le g \le N_k$, it turns out that project G_{kt} deviates from the planned trajectory, an error signal is generated, and decision-making is based on solving Problem II to reassign the remaining budget among the remaining project's activities to maximize value p_{kt}. Here value $p_{kt} = \Pr\{t + T_k[C_{kt}] \le D_k\}$ is the probability of accomplishing on time project G_{kt} with available budget $C_k(t)$. If the problem's solution enables the project's deadline to be met, subject to the chance constraint, i.e., if $p_{kt} \ge p_k^*$ holds, a corrected planned trajectory is determined and Problem III is resolved to determine the next control point $t_{k,g+1}$. Otherwise, i.e., in case $p_{kt} < p_k^*$, an emergency signal is generated and decision-making is carried out at the company level. Problem I is resolved under emergency conditions to reassign the remaining budget among the unaccomplished projects. Thus, in the course of controlling a group of projects, the latter are first, at $t = 0$, optimized on line from top to bottom. In the case of an emergency, the generated "bottom-top" signals are converted into control actions to enable the projects' due dates to be met on time.

Budget reassignment problems (which we call here Problem I) at the company level are outlined in §§14.3-14.5. Note that we have deliberately introduced two additional levels (Scheduling and Inspection Levels). Both levels are auxiliary ones and do not enter the "control circle" comprising levels I-III. However, the presence of both additional levels clarifies the process of monitoring several projects.

§15.2 Optimal budget reassignment among activities (Problem II)

This problem is repeatedly resolved at the project level in two cases:

(1) When budget C_{kt} is assigned to project G_{kt} at moment $t \ge 0$ at the company level, $1 \le k \le n$ (after solving Problem I). Thus, the initial data for solving Problem II is value C_{kt}.

(2) When, in the course of an on-line control, it turns out that project G_{kt} deviates from its planned trajectory. Here, the initial data for Problem II is the actual remaining budget $C_k(t)$ which has been observed at routine control point $t = t_{kg}$.

In both cases the problem is to reallocate the budget among activities $(i,j)_k \in G_{kt}$ to maximize the probability of completing the project by its due date D_k. A stochastic optimization problem is solved for each project G_{kt} separately:

determine optimal values $c(i,j)_k$ to maximize objective

$$\max_{\{c(i,j)_k\}} p_{kt} = \max_{\{c(i,j)_k\}} \left[\Pr\{t + T_k[C_{kt}] \le D_k\} \right] \quad (15.2.1)$$

subject to (15.1.6) and

$$\sum_{(i,j)_k \in G_{kt}} c(i,j)_k = Q_{kt} . \quad (15.2.2)$$

Here $Q_{kt} = \begin{cases} C_{kt} & \textit{if Problem II receives the input from the company level;} \\ C_k(t) & \textit{if Problem II is solved after a project's inspection.} \end{cases}$

Problem II is solved by applying Procedure I outlined in §14.4 to a PERT project G_{kt}.

A dual problem of practical use can be formulated as follows: for each project G_{kt} separately, given probability value p_k, determine budget value C_{kt} which, when solving the optimization problem (15.1.6, 15.2.1-15.2.2), results in delivering the maximum to the objective equal to p_k,

$$p_k[C_{kt}] = p_k, \ 1 \le k \le n . \quad (15.2.3)$$

Problem (15.1.6, 15.2.1-15.2.3) can be solved by using Procedure III outlined in §14.5.

It can be clearly recognized that carrying out optimal budget reassignment among project's activities results in providing first critical activities with budget values $c(i,j)_k$ that are as close as possible to the corresponding values $c(i,j)_{k\max}$. The remaining budget is redistributed among non-critical activities that have practically no influence on the project's duration.

§15.3 On-line control model at the project level (Problem III)

After undertaking budget reassignment among the projects the latter are realized and controlled independently. The problem of on-line control of any project G_{kt}, $1 \le k \le n$, is, in essence, to determine, at each routine control point t_{kg}, the next control point $t_{k,g+1}$ honoring objective (15.1.1). Note, that before carrying out the on-line control, the project management has to reassign the budget value C_{kt}, $1 \le t \le D_k$, among activities $(i,j)_k$, i.e., has to solve optimization problem (15.1.6, 15.2.1-15.2.2). This can be easily realized for practically all PERT type projects, even on personal computers.

Like any other on-line control it has to be carried out by comparing the state variable of the progress of the project at control points with the corresponding values of the planned project target (trajectory). Thus, to implement on-line control, we have to determine for each project G_{kt} its planned trajectory curve $V_k^{pl}(t)$ together with the state variable $V_k^f(t)$.

We suggest that the on-line control problem be solved as follows:

At any control point t_{kg}, $1 \le k \le n$, $0 \le g \le N_k$, determine the next control point $t_{k,g+1}$ to minimize the objective

$$\min_{\{t_{k,g+1} > t_{kg}\}} N_k(t_{kg}), \quad (15.3.1)$$

subject to

$$t_{k0} = 0,\ t_{kN_k} = D_k, \quad (15.3.2)$$

$$t_{k,g+1} - t_{kg} \ge \Delta_k, \quad (15.3.3)$$

$$\Pr\{V_k^f(t) \le V_k^{pl}(t)\} \ge p_k^* \quad \forall t : t_{kg} \le t \le t_{k,g+1}. \quad (15.3.4)$$

Restrictions (15.1.4-15.1.6) are not imbedded in the model since they are honored in Problems I and II, which are solved before carrying out on-line control. Note that chance constraint (15.3.4), unlike (15.1.3), does not include

$$p_k^{**} \ge p_k[C_{kt}]. \quad (15.3.5)$$

This is because restriction (14.3.6) in Problem I includes (15.3.5) and thus prohibits assigning additional redundant budget to any project G_{kt}. If at a control point $t > 0$ it is anticipated that due to most favorable circumstances a part of the project budget is redundant that extra budget will be reassigned at the first emergency call, by solving Problem I at the company level. Note that chance constraint (15.3.4) is, in essence, stricter than (15.1.3). The latter only ensures that the project's due date is met on time with a probability not less than p_k^*, while chance constraint (15.3.4) enables the state variable $V_k^f(t)$ not to exceed the planned trajectory $V_k^{pl}(t)$ *at any moment* t *within the planning horizon* $[0, D_k]$.

In §4.1 we have already outlined the general idea of determining single project's trajectories and calculating next control points. We will overview those concepts in greater detail, especially in connection with other control techniques of the hierarchical control model for several network projects.

The general idea of the on-line control is presented in Fig. 15.2 and is as follows:

At each routine control point t_{kg}, $0 \le g \le N_k$, inspection is undertaken to observe the remaining budget $C_k(t_{kg})$. Value $C_k(t_{kg})$ is the state variable $V_k^f(t)$ at point $t = t_{kg}$. At the beginning of the project's realization, at $t = t_{k0} = 0$, $C_k(0) = C_k$, and since according to its plan project $G_k(N, A)$ has to be accomplished not later than at $t = D_k$ together with its full budget utilization, we determine the planned trajectory curve (first iteration) $V_k^{pl}(t)^{(1)}$ by a straight line connecting two points with coordinates $[0, C_k]$ and $[D_k, 0]$. Thus we obtain

$$V_k^{pl}(t)^{(1)} = C_k - t \cdot \frac{C_k}{D_k}, \quad (15.3.6)$$

which is used within the interval $t \in [0, t_{k1}]$, up to the first control point.

Note that no restrictions are imposed on the project's actual cost-duration function except for the fact that such a function has to be continuous and decreasing.

If, at a routine control point $t_{kg} > 0$, it is observed that $C_k(t_{kg}) \le V_k^{pl}(t_{kg})^{(q)}$ (q-th iteration), there is no need for any interference in the project's realization, since the project meets a stricter chance constraint than (15.1.3). Thus, the progress of the project proceeds, trajectory curve $V_k^{pl}(t)^{(q)}$ remains unchanged, and the adjacent control point $t_{k,g+1}$ has to be determined. In case $V_k^{pl}(t_{kg})^{(q)} < C_k(t_{kg})$, an error signal is generated and we have to examine the project in greater detail. Optimization Problem II, i.e., problem (15.1.6, 15.2.1-15.2.2) has to be solved, with $Q_{kt} = C_k(t_{kg})$, $t = t_{kg}$, in order to obtain the maximal probability of the project meeting its deadline on time without any additional help from other projects. If solution p_{kt} satisfies $p_{kt} \geq p_k^*$, then new budget values $c(i,j)_k$ obtained by that solution are reassigned among activities $(i,j)_k$. A corrected planned trajectory curve $V_k^{pl}(t)^{(q+1)}$ (($q+1$)-th iteration) has to be determined by a straight line connecting two points with coordinates $[t_{kg}, C_k(t_{kg})]$ and $[D_k, 0]$. The corresponding trajectory curve to be used within the interval $[t_{kg}, t_{k,g+1}]$ is as follows:

$$V_k^{pl}(t)^{(q+1)} = \frac{D_k \cdot C_k(t_{kg})}{D_k - t_{kg}} - t \cdot \frac{C_k(t_{kg})}{D_k - t_{kg}}, \tag{15.3.7}$$

If relation $p_{kt} < p_k^*$ holds, an emergency is called, all projects are inspected at moment $t = t_{kg}$, and budget reassignment among the projects is introduced on the basis of solving Problem I. After obtaining new values C_{kt}, $t = t_{kg}$, optimization problem (15.1.6, 15.2.1-15.2.2) is solved again for all projects t_{kg}, $1 \leq k \leq n$, to determine optimal budget values $c(i,j)_k$ assigned to the activities. Afterwards, new planned trajectories $V_k^{pl}(t)^{(q+1)}$ are determined for all projects $G_{kt_{kg}}$, $1 \leq k \leq n$,

$$V_k^{pl}(t)^{(q+1)} = \frac{D_k \cdot C_{kt_{kg}}}{D_k - t_{kg}} - t \cdot \frac{C_{kt_{kg}}}{D_k - t_{kg}}, \tag{15.3.8}$$

to determine next control points $t_{k,g+1}$, and the projects' realization proceeds.

It can be clearly recognized that in the course of the project's realization its actual cost-duration function (irrespective of any assumption of that function) is approximated closer and closer by repeatedly corrected trajectory curves (15.3.6-15.3.8) between adjacent control points. This is illustrated in Fig. 15.3 for the case of a concave cost-duration function. If solving Problem II results in $p_{kt} \geq p_k^*$, an emergency is not called and each new trajectory comes closer to the project's consumption curve.

Since minimizing the number of future control points results in maximizing the time span between two routine adjacent control points $t_{k,g+1}$ and t_{kg}, the problem at hand is to maximize the value

$$\delta_{kg} = t_{k,g+1} - t_{kg}, \tag{15.3.9}$$

subject to (15.3.2-15.3.4).

Denoting $V_k^{pl}(t) - V_k^f(t) = H_k(t)$, we substitute optimization problem (15.3.2-15.3.4, 15.3.9) for:

$$\max_{C_{kt_{kg}}} \delta_{kg} \tag{15.3.10}$$

subject to (15.3.2-15.3.3) and

$$\Pr\{H_k(t) \ge 0\} \ge p_k^*,\ 1 \le k \le n, \tag{15.3.11}$$

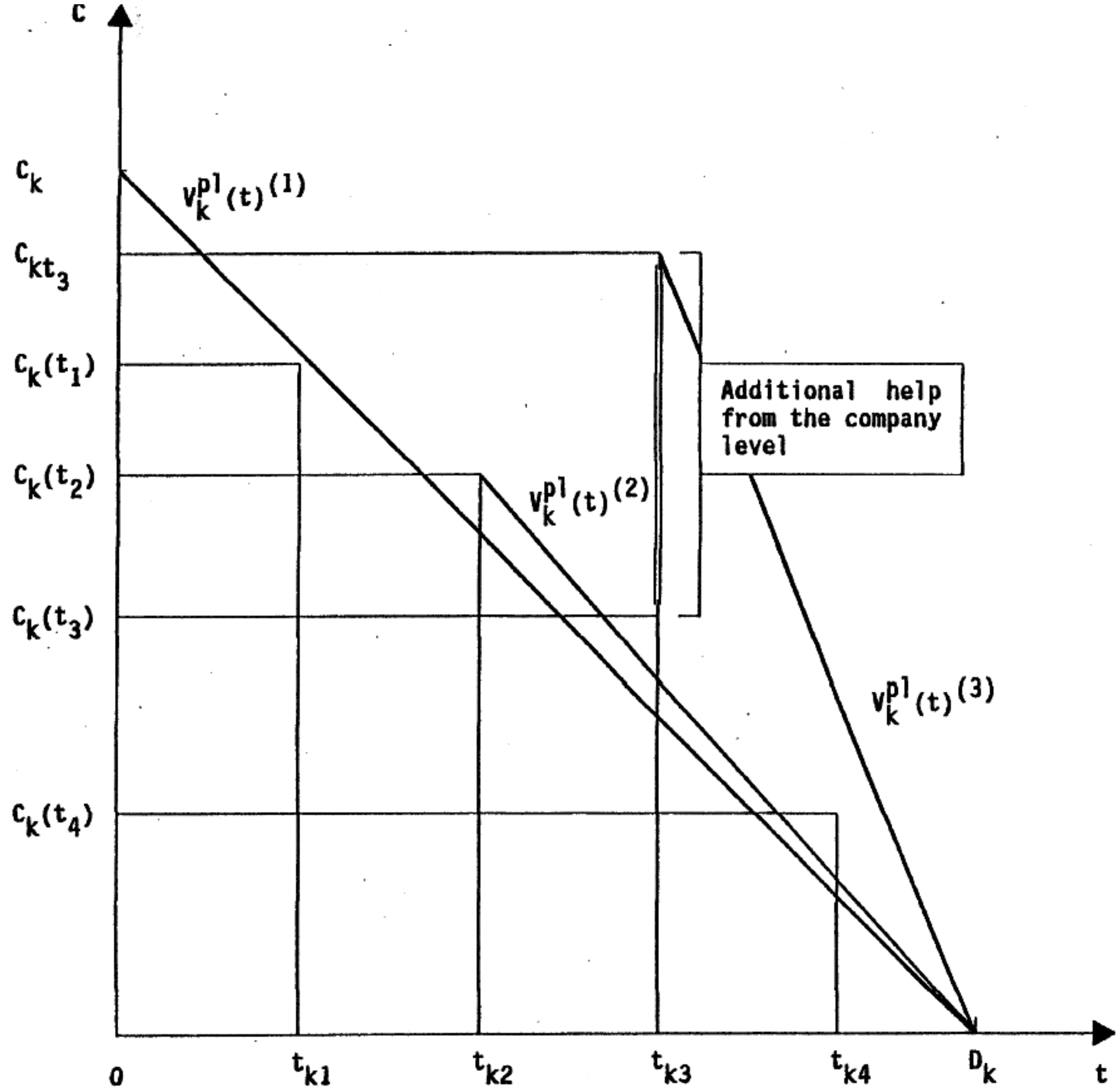

Figure 15.2. *Corrected planned trajectory curves for on-line control* (*emergency at* $t = t_{k3}$)

Let us examine random variable $H_k(t)$, $t > t_{kg}$, in greater detail. Since each activity duration $t(i,j)_k \in G_{kt}$ is a random variable with a density function dependent on budget value $c(i,j)_k$, random variable $H_k(t)$ is a result of multiple random disturbances. Thus, it is reasonable to assume that $H_k(t)$ has a normal distribution with parameters $E[H_k(t)]$ and $V[H_k(t)]$. Note that both these values can be easily simulated to calculate their corresponding unbiased and consistent estimates

$$\overline{H}_k(t) = \frac{1}{M}\sum_{r=1}^{M} H_k^{(r)}(t), \tag{15.3.12}$$

$$S^2[H_k(t)] = \frac{1}{M-1}\sum_{r=1}^{M}\left[H_k^{(r)}(t) - \overline{H}_k(t)\right]^2, \tag{15.3.13}$$

where M is the number of simulation runs and $H_k^{(r)}(t)$ is the value $H_k(t)$ obtained by the r-th simulation.

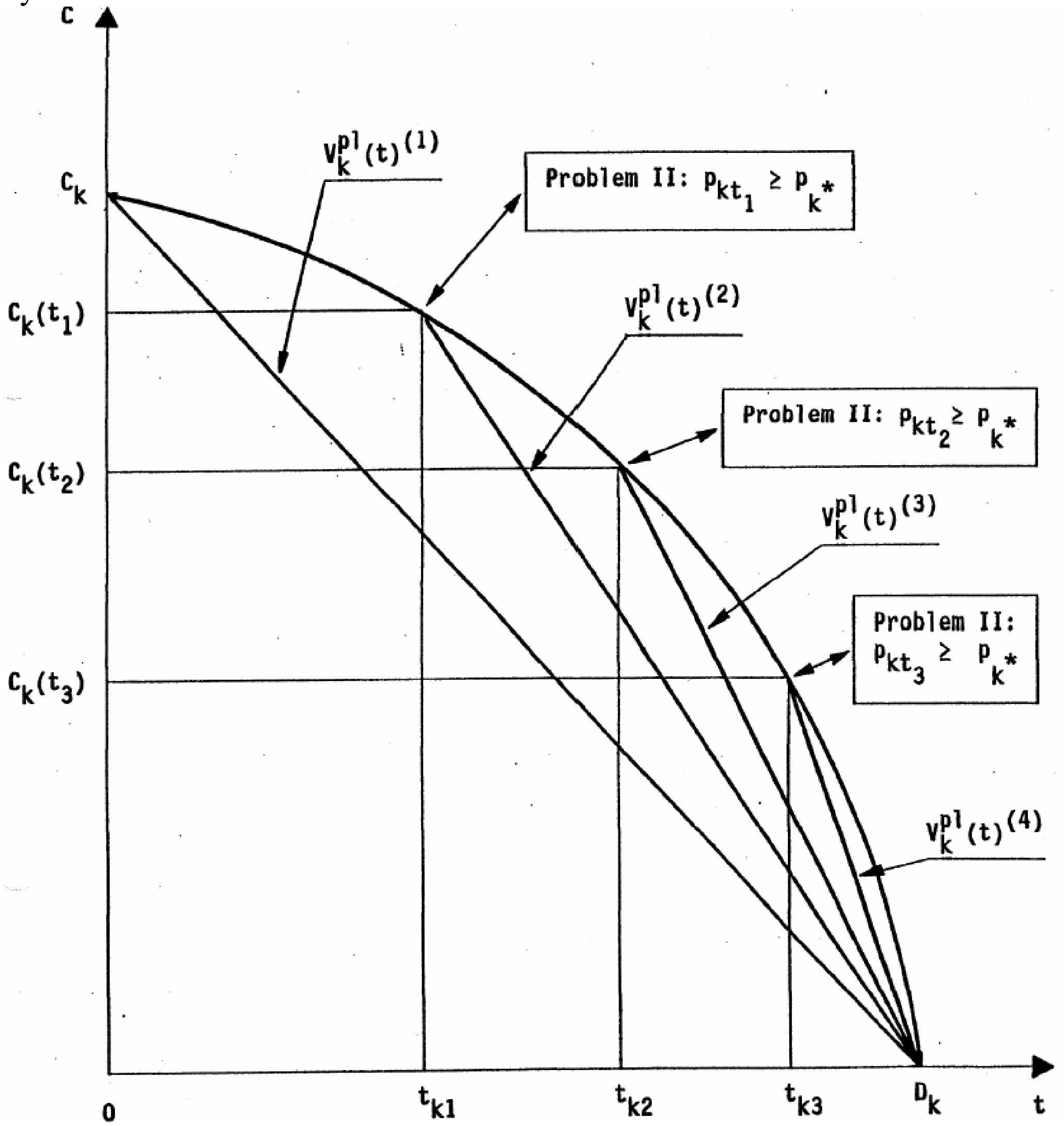

Figure 15.3. *A concave consumption function $C_k(t)$ approximated by trajectory curves (no emergency)*

Note that chance constraint (15.3.11) can be written in another form

$$\phi(q) \geq p_k^*, \tag{15.3.14}$$

where

$$q = -\frac{\overline{H}_k(t)}{\sigma[H_k(t)]}, \quad \phi(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\frac{u^2}{2}}du. \tag{15.3.15}$$

According to (15.3.10) and (15.3.11) the maximal value T^* satisfying

$$T^* = \max_{t_{kg} \le t \le D_k} \{t : [\phi(q) \ge p_k^*]\}, \quad t - t_{kg} \ge \Delta_k , \tag{15.3.16}$$

should be determined as the next control point $t_{k,g+1}$.

In practice, $t_{k,g+1}$ can be calculated by means of simulation with a constant step of length Δ_k. The procedure of consecutively increasing value $t > t_{kg}$ step-by-step is followed until restriction (5.3.16) ceases to hold. Thus, problem (15.3.2-15.3.3, 15.3.10-15.3.11) can be solved by means of simulation in order to capture the last moment before the project deviates from its target.

The on-line algorithm to determine the next control point t_{g+1} for a single project is outlined in §4.1.

§15.4 A hierarchical heuristic algorithm

The goal of this section is to unify all optimization and simulation problems on different levels to create an on-line control algorithm for several PERT-COST projects.

The steps of the algorithm are as follows:

Step 1. At moment $t = 0$ the input data is externally given:

- the initial data for all activities $(i, j)_k$ entering the projects, $1 \le k \le n$;
- probabilities p_k^* and p_k^{**} for all projects;
- company's total budget C to carry out the projects;
- projects' due dates D_k, $1 \le k \le n$.

Step 2. At time $t = 0$ solve dual optimization problem (15.1.6, 15.2.1-15.2.3) for all projects $G_k(N, A)$, for cases $p_k = p_k^*$ and $p_k = p_k^{**}$, respectively.

Step 3. At time $t = 0$ solve budget reassignment Problem I (14.3.2, 14.3.6, 14.3.8-14.3.9) to determine budget values C_{k0}. Note that although Problem I can be solved analytically it is based on a heuristic assumption that $p_k[C_{kt}]$ is a linear function.

Step 4. For all projects separately at time $t = 0$, solve optimization Problem II (15.1.6, 15.2.1-15.2.2) to determine activity budget values $c(i, j)_k$, $1 \le k \le n$.

Step 5. For all projects separately at time $t = 0$, determine planned trajectories $V_k^{pl}(t)^{(1)} = C_k - t \cdot \dfrac{C_k}{D_k}$, $t > 0$, $1 \le k \le n$, $C_k = C_{k0}$.

Note that Step 5 is the end of the planning stage

Step 6. For all projects separately, at time $t = 0$, determine (see §4.1) the next control points t_{kg}, $0 \le g \le N_k$.

Step 7. Determine $t^* = t_{k^*g} = \min_k \{t_{kg}\}$, k^* being the index of the project to be inspected first: if $t^* = D_{k^*}$ proceed to Step 15. Otherwise apply the next step.

Step 8. Inspect project G_{k^*t} at time t^*. If $C_{k^*}(t^*) \le V_{k^*}^{pl}(t^*)$, apply Step 10. Otherwise go to Step 11.

Step 9. Determine for project G_{k^*t} its corrected planned trajectory: the straight line connecting points $[t^*, C_{k^*}(t^*)]$ and $[D_{k^*}, 0]$.

Step 10. Determine for project G_{k^*t} its next control point $t_{k^*,g+1}$ (see §4.1). Return to Step 7.

Step 11. If $C_{k^*}(t^*) > V_{k^*}^{pl}(t^*)$, solve optimization Problem II for $Q_{kt} = C_{k^*}(t^*)$, $t = t^*$, $k = k^*$. If the maximized objective satisfies $p_{k^*t^*} \ge p_{k^*}^*$ apply the next step. Otherwise proceed to Steep 13.

Step 12. Carry out budget reassignment among activities $(i,j)_{k^*}$ on the basis of the unchanged budget $C_{k^*}(t^*)$. Return to Step 9.

Step 13. Call an emergency and observe all projects at moment t^*. Solve budget reassignment Problem I (14.3.2, 14.3.6, 14.3.8-14.3.9) and determine new values C_{kt^*}, $1 \le k \le n$.

Step 14. For all projects G_{kt^*} solve optimization Problem II to obtain new values $c(i,j)_k$, $1 \le k \le n$. Return to Step 9.

Step 15. Inspect project G_{kt^*} at its due date, $t = D_{k^*}$. If the project is completed, apply the next step. Otherwise proceed to Step 17.

Step 16. If there are other projects being uncompleted at moment t^*, go to Step 7. Otherwise apply Step 18.

Step 17. Consider a new due date for project G_{k^*t} (a non-formalized procedure). Return to Step 13.

Step 18. The algorithm terminates.

The algorithm is performed in real time; namely, each iteration of the algorithm can be performed only after the remaining budget for each project is actually inspected. The control points cannot be predetermined. However, if we want to evaluate the efficiency of the hierarchical control model, we can simulate the algorithm's work by random sampling of the actual duration of activities. By simulating the algorithm's work many times, the probability of completion on time and the average number of control points $\sum_{k=1}^{n} N_k$, as well as the average number of budget reassignments, can be evaluated.

Note, in conclusion, that the developed control model does not consider new techniques of *project scheduling*. However, we cannot simulate a network project at any control point without determining start times for each activity. In our algorithm (Steps 8, 10 and 15), project simulation has been carried out by setting start times for each activity $(i,j)_k$ equal to the earliest possible time of realization of node i (on the basis of already simulated activity durations). As to projects to be realized in real time, we suggest that activities $(i,j)_k$ be scheduled between adjacent control points t_{kg} and $t_{k,g+1}$, according to the existing techniques used by the project managers. These techniques are generally poor, since determining a good deterministic schedule of a network project comprising activities of random durations is a thankless task. In combination with proper on-line control, such techniques become more effective since the project schedule with biased estimates is repeatedly corrected over time. This fully complies with our philosophy outlined in Chapters 1 and 3. However, on our opinion, even better results may be achieved by unifying techniques outlined in Chapters 11-13 in conjunction with the results presented in Chapters 14-15.

§15.5 Numerical example

The company is faced with carrying out three network projects of PERT-COST type. The initial projects' data is presented in Tab. 15.1-15.3 [64]. The projects' parameters are as follows:

$D_1 = 80$; $p_1^* = 0.55$; $p_1^{**} = 0.70$.

$D_2 = 130$; $p_2^* = 0.70$; $p_2^{**} = 0.90$.

$D_3 = 150$; $p_3^* = 0.60$; $p_3^{**} = 0.80$.

$\Delta_1 = \Delta_2 = \Delta_3 = 10$.

The total budget at the company's disposal to carry out the projects is $C = 835{,}000\$$.

In order to illustrate the work of the algorithm we will outline a simulation run below.

Solving problem (15.1.6, 15.2.1-15.2.3) for cases $p_k = p_k^*$ and $p_k = p_k^{**}$, $k = 1,2,3$, results in obtaining values C_{k0}^* and C_{k0}^{**} as follows:

$C_{10}^* = 271.55$; $C_{10}^{**} = 276.70$;

$C_{20}^* = 266.18$; $C_{20}^{**} = 276.44$;

$C_{30}^* = 279.47$; $C_{30}^{**} = 285.22$;

Solving the budget reassignment Problem I (14.3.2, 14.3.6, 14.3.8-14.3.9) at $t = 0$ we obtain:

1. $C_{10} = 276.70$. Solving the budget reassigning Problem II (15.1.6, 15.2.1-15.2.2) at the project level results in $p_1[C_{10}] = 0.70 > p_1^* = 0.55$.
2. $C_{20} = 274.52$ with the corresponding $p_2[C_{20}] = 0.85 > p_2^* = 0.70$.
3. $C_{30} = 283.78$. Solving Problem II results in $p_3[C_{30}] = 0.69 > p_3^* = 0.60$.

After the budget has been reassigned among the projects the latter are controlled separately.

Control points t_{11}, t_{21} and t_{31} (see algorithm outlined in §4.1) are as follows: $t_{11} = 70$; $t_{21} = 120$; $t_{31} = 10$. Since $t_{31} = \min(t_{11}, t_{21}, t_{31})$ Project 3 has to be inspected first. Simulating $C_3(10)$ we obtain $C_3(10) = V_3^{pl}(10)^{(1)} = 264.86$.

Thus, we redistribute the remaining budget among the remaining activities in order to calculate the maximal probability of the project being able to meet its deadline on time without any additional help. Solving Problem II results in obtaining $p_3[C_3(10)] = 0.77 > p_3^* = 0.60$. Thus, the on-line control proceeds with new budget values $c(i,j)_3$ and the new planned trajectory.

Table 15.1. Initial data for Project 1

No.	i	j	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	2	1	10	25	81
2	1	3	1	5	22	60
3	1	4	1	6	75	105
4	1	5	4	20	80	132
5	1	6	1	5	30	45
6	1	7	6	40	160	200
7	2	3	8	20	50	100
8	3	13	10	20	110	220
9	3	15	4	123	90	120
10	4	14	6	14	50	100
11	5	9	6	18	150	200
12	5	13	4	8	105	140
13	6	9	1	8	80	150
14	7	8	2	8	42	60
15	8	10	2	8	20	32
16	8	11	4	8	40	80
17	9	11	3	7	90	120
18	9	12	5	7	42	60
19	10	20	3	6	60	89
20	10	21	6	12	105	140
21	11	19	5	20	150	200
22	11	21	6	14	50	100
23	12	18	4	10	90	120
24	13	18	4	12	48	60
25	13	19	4	8	63	110
26	14	16	1	6	58	102
27	14	17	1	4	23	94
28	15	16	3	10	85	120
29	15	17	3	6	60	104
30	16	22	4	10	70	93
31	17	23	7	12	74	140
32	18	23	4	10	80	120
33	19	23	4	10	40	87
34	20	21	1	6	32	72
35	21	23	4	10	63	95
36	22	23	5	12	87	128

The next control point t_{32} is determined by using the algorithm outlined in §4.1: $t_{32} = 20$. Simulating $C_3(20)$ we obtain $C_3(20) = 256.99 > V_3^{pl}(20)^{(2)} = 250.62$.

Solving Problem II results in $p_3[C_3(20)] = 0.79$ which exceeds the minimal value $p_3^* = 0.60$.

Table 15.2. Initial data for Project 2

No.	i	j	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	2	2	8	80	160
2	1	3	1	6	72	110
3	1	4	1	8	30	75
4	1	5	2	15	207	257
5	1	6	1	8	25	45
6	1	7	8	30	160	200
7	2	15	3	8	158	205
8	3	14	10	15	110	220
9	3	15	5	10	90	120
10	4	9	8	12	200	250
11	5	13	5	10	205	240
12	5	14	4	12	210	270
13	6	9	2	5	80	150
14	7	8	3	10	202	265
15	8	11	2	10	160	192
16	8	12	6	10	180	207
17	9	10	1	5	90	120
18	9	11	3	10	42	60
19	10	11	2	5	60	80
20	11	19	7	15	120	170
21	11	20	4	8	63	110
22	11	21	8	12	150	200
23	12	19	5	10	190	230
24	13	17	4	12	160	220
25	13	18	5	10	148	180
26	14	16	1	7	58	102
27	14	18	1	7	23	94
28	15	16	4	9	85	120
29	16	22	4	11	70	93
30	17	22	5	10	182	253
31	17	23	6	10	274	340
32	18	20	2	8	80	120
33	19	23	2	5	90	137
34	20	22	1	4	32	72
35	21	23	3	8	143	195
36	22	23	5	12	87	128

Thus, the on-line control proceeds with new values $c(i,j)_3$ and the new trajectory curve $V_3^{pl}(t)^{(3)}$. The next control point (see §4.1) is $t_{33} = 30$. Simulating the actual remaining budget results in $C_3(30)=245.42>V_3^{pl}(30)^{(3)}=236.29$. Solving again the optimal redistribution Problem II among the remaining activities results in $p_3[C_3(30)]=0.77>p_3^*=0.60$. A new planned trajectory is determined with the next control point $t_{34}=140$. Since $t_{11}=70=\min(70,120,140)$ we now inspect the first project at $t=70$.

Table 15.3. Initial data for Project 3

No.	i	j	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	2	3	6	162	231
2	1	4	1	6	111	203
3	1	5	7	15	207	257
4	1	6	3	5	117	186
5	1	7	2	10	160	200
6	2	3	4	20	102	149
7	2	15	3	6	158	205
8	3	14	10	15	110	220
9	3	15	4	8	90	120
10	4	9	5	10	200	250
11	4	14	7	12	150	200
12	5	13	6	10	205	240
13	6	9	4	8	173	231
14	7	8	2	10	202	265
15	8	11	3	8	160	192
16	8	12	6	10	180	207
17	9	10	2	4	90	161
18	9	11	3	8	142	237
19	10	21	4	10	205	240
20	11	19	4	15	120	170
21	11	20	4	10	63	110
22	11	21	8	14	150	200
23	12	19	4	10	190	230
24	13	17	4	10	160	220
25	13	18	5	10	148	180
26	14	16	2	6	58	102
27	15	16	4	8	85	120
28	15	17	3	7	60	104
29	16	22	4	10	70	93
30	17	22	5	10	182	253
31	17	23	6	9	274	340
32	18	20	3	5	80	120
33	19	23	3	6	90	137
34	20	22	1	10	32	72
35	21	23	2	10	143	195
36	22	23	6	10	87	128

Simulating $C_1(70)$ results in $C_1(70)=13.70<V_1^{pl}(70)^{(1)}=34.59$. Thus, we continue carrying out the project without any budget redistribution. The next control point of the project will be its due date, i.e., $t_{12}=80$. By means of simulation, it was determined that the project meets the due date on time, and we proceed to controlling Projects 2 and 3.

The next control point is $t_{21}=120$ (Project 2). Simulating $C_2(120)$ results in $C_2(120)=2.48<V_2^{pl}(120)^{(1)}=21.12$. We continue carrying out Project 2 and inspect the latter at the due date $D_2=130$. Simulating $C_2(130)$ shows that the project also meets its due date on time.

The last control point is $t_{34}=140$ (Project 3). Simulating value $C_3(140)$ results in $C_3(140)=1.27<V_3^{pl}(140)^{(4)}=20.45$. We proceed to carry out the project without changing its planned trajectory. At the due date $D_3=150$, we inspect the project by means of simulation. Project 3 also meets its due date on time. Thus the simulation run is accomplished. All the projects have met their due dates on time and there was no need for additional budget reassignment among the projects.

Ref. [92] presents some numerical results of the model's performance for a group of ten projects being controlled simultaneously. It can be well recognized that all the projects are expected to be completed before the assigned due dates with not less than the pregiven probability p^* according to the total allocated budget.

In [92] the results of 100 simulation runs for an individual project are outlined. We have chosen Project #7 with initial parameters as follows:

$D_7=140$; $p_7^*=0.85$; $C_7=3{,}868$; $\Delta=10$.

The computational results including both the average number of control points and the actual probability of meeting the due date on time, are as follows:

1. The average number of control points is 4.93.
2. The actual probability $p_7=0.89$ of carrying out Project 7 within the due date is higher than the pregiven confidence probability $p_7^*=0.85$.

A conclusion can be drawn that the outlined above model can successfully control multiple projects.

§15.6 Conclusions

The following conclusions can be drawn from Chapters 14-15:

1. The hierarchical control model can be used for practically all activity-on-arc network projects with independent activities of random durations and cost-duration functions. The model can be used both for controlling single projects and several projects with a restricted company budget.
2. Successful experimentation has been undertaken for the case when the density function of each activity duration $t(i,j)_k$ can be approximated by the beta distribution law, while value $t(i,j)_k$ is approximately inversely

proportional to the budget $c(i,j)_k$. Other alternative distributions, e.g., normal, uniform, triangle, etc., together with other cost-duration functions can be implemented as well, on condition that each activity duration can be simulated on the basis of the budget value assigned to that activity.

3. The outlined control model does not impose any restrictions on the budget's actual cost-duration function. The on-line control is established so that the cost-duration function in the course of the project's realization is approximated closer and closer by repeatedly corrected trajectory curves between adjacent control points.
4. When simulating the project (either to determine a routine control point or to simulate the available remaining budget) we determine start times for each activity. The appropriate techniques of project scheduling outlined in Chapters 11-13, for the case of activities of random durations, become more effective in combination with on-line control.
5. Several optimization problems are embedded in the model. They can all be easily solved on a PC, especially for projects with a medium number of activities.

Chapter 16. Hierarchical Decision-Making Model for Alternative Stochastic Network Projects

§16.1 The problem's description

16.1.1 Introduction

This Chapter is not only a modification but a further development of Chapters 14-15 since an alternative controlled model is usually more complicated than a PERT-COST model.

We will consider a three-level decision-making model for controlling stochastic network projects. The upper level (the company level) is faced with the problem of optimal budget reassignment among several projects. We will consider network projects of CAAN type (see Chapter 8) with random activity durations and alternative outcomes in key nodes. There are two different types of alternative events. The first one reflects stochastic (uncontrolled) branching of the project development. The alternative event of the second type is of a deterministic nature, i.e., the project manager chooses the outcome direction.

At the medium level (project level) the management determines on the basis of the assigned budget the optimal subnetwork (joint variant) which will be realized within the progress of the project. Choosing the optimal joint variant determines the optimal outcome direction at every decision node which will be reached in the course of the project's realization. The objective is either the project's duration or the probability of meeting the due date on time.

At the subnetwork level the project is controlled according to the chosen optimal joint variant. The latter is inspected periodically by calculating the confidence probability of meeting the due date on time. If for any reason the project does not meet the chance constraint, i.e., the probability becomes less than the minimal pregiven value, budget reassignment among the remaining activities has to be undertaken. If despite all control actions the project cannot meet the chance constraint not to deviate from the target, then an emergency is called and the company level is applied to reassign the remaining budget among the unaccomplished projects.

The main goal of this Chapter is to develop a unified three-level decision-making model and to indicate planning and control actions and optimization problems for all levels.

16.1.2 Notation and initial data

It can be well-recognized that the terms at the Company Level fully coincide with those outlined in §14.2 for the hierarchical on-line PERT-COST model - one has only to substitute the PERT-COST type graph by the CAAN type. However, terms at the lower hierarchical levels are subject to significant amendments, as follows:

The Project Level

$S_{rkt} \subset G_{kt}$ -the r -th joint variant of project G_{kt} (a subnetwork of PERT or GERT type obtained by choosing certain directions in decision nodes and excluding unfixed directions), $1 \le r \le m_{kt}$;

$S_{\xi kt} \subset G_{kt}$ -the optimal joint variant of project G_{kt};

m_{kt} - number of joint variants in project G_{kt};

L_{frkt} - the full (overall) variant (realization of a GERT type subnetwork S_{rkt} by simulating random alternative branchings), obtained by the f-th simulation;

p_{rkt} - the probability of joint variant S_{rkt} to meet due date D_k on time ;

$p_{rkt}(Q)$ - confidence probability of the joint variant to be accomplished on time with available budget $Q = C_{kt}$ or $Q = C_k(t)$;

$C_{rkt}(p)$ - budget value which enables S_{rkt} to meet the deadline with probability p.

The Subnetwork (Joint Variant) Level

$(i, j)_k$ - activity leaving node i and entering node j, $(i, j)_k \in G_{kt}$, $t \ge 0$;

$c(i, j)_k$ - budget assigned to activity $(i, j)_k$;

$c(i, j)_{k\min}$ - minimal possible budget to carry out activity $(i, j)_k$ (pregiven);

$c(i, j)_{k\max}$ - maximal budget required to carry out activity $(i, j)_k$ (pregiven); in case $c(i, j)_k > c(i, j)_{k\max}$ additional budget is redundant;

$t(i, j)_k$ - random duration of activity $(i, j)_k$; it is assumed that $(i, j)_k$ has a beta-distribution with density function (14.1.1);

$A(i, j)_k$ - pregiven value to satisfy

$a(i, j)_k = \dfrac{A(i, j)_k}{c(i, j)_k}$ which is the lower bound for random value $t(i, j)_k$;

$B(i, j)_k$ - pregiven value to satisfy

$b(i, j)_k = \dfrac{B(i, j)_k}{c(i, j)_k}$ which is the upper bound for random value $t(i, j)_k$;

For each activity $(i, j)_k \in G_{kt}$ the initial data is given as follows:

i; j; $h(i, j)_k$; $p(i, j)_k$; $c(i, j)_{k\min}$; $c(i, j)_{k\max}$; $A(i, j)_k$; $B(i, j)_k$.

Here $h(i, j)_k$ designates the direction arc $(i, j)_k$, $p(i, j)_k = 0$ denotes that node i in $G_k(N, A)$ or G_{kt} is of PERT type. In case $0 < p(i, j)_k < 1$ node i is a random milestone, i.e., it has an alternative random outcome, while $p(i, j)_k = 1$ means that node i is a decision node of deterministic type.

§16.2 Auxiliary Procedures I-IV

Direct and inverse heuristic auxiliary Procedures I and III have been considered in §§14.4-14.5 for PERT type projects. But in the case of a CAAN type project the joint variant that corresponds to the optimal outcome from a decision node, may be a GERT type subnetwork. Thus, optimal budget reassignment among network activities is actual for GERT type projects as well. But if the optimal joint variant $S_{rk} \subset G_k(N,A)$ is a GERT type network it is impossible to reallocate the budget among *all* the activities $(i,j)_k \in S_{rk}$ since in the course of the project's realization *a part of the activities will not actually be carried out*. We will therefore formulate a modified version of problem (14.4.1-14.4.3) for a GERT type project as follows [65,92]:

Given a GERT network $G(N,A)$ together with a pregiven budget value C and due date D, to maximize the objective

$$\max p(C) \tag{16.2.1}$$

subject to

$$c(i,j)_{\min} \le c(i,j)^f \le c(i,j)_{\max}, \tag{16.2.2}$$

$$\sum_{(i,j)^f \in G(N,A)} c(i,j)^f \le C. \tag{16.2.3}$$

Here $(i,j)^f$ denotes activity $(i,j) \in G(N,A)$ which will actually be carried out in the course of the project's realization. A heuristic procedure of the problem's solution which will be called henceforth **Procedure II**, is outlined below. The step-by-step procedure is carried out by means of simulation as follows:

Step 1. Simulate the non-contradictive outcomes in $G(N,A)$ to obtain a full variant $L \subset G(N,A)$. Denote by L^f a full variant obtained in the course of the f-th simulation. Note that the simulated L^f is a PERT subnetwork.

Step 2. Apply Procedure I to L^f to determine the maximal confidence probability

$$\max_{\{c(i,j)\}} p\left(C \middle| L^f\right) \tag{16.2.4}$$

subject to

$$c(i,j)_{\min} \le c(i,j)^f \le c(i,j)_{\max}, \tag{16.2.5}$$

$$\sum_{(i,j) \in L^f} c(i,j) = C. \tag{16.2.6}$$

Step 3. Repeat Steps 1-2 M times to obtain representative statistics.

Step 4. Calculate

$$\overline{p}(C) = \frac{1}{M} \sum_{f=1}^{M} p\left(C \middle| L^f\right), \tag{16.2.7}$$

where $\overline{p}(C)$ is the maximal value of objective (16.2.1) to be determined.

A modified version of the dual problem (14.5.1-14.5.3) is formulated and later on solved by means of simulation for a GERT network. The problem is as follows:

Given a GERT network $G(N,A)$ with a pregiven confidence probability p and due date D, to minimize objective

$$\min C \tag{16.2.8}$$

subject to (16.2.2-16.2.3) and

$p(C) = p$. (16.2.9)

The corresponding step-by-step heuristic solution, which will be called **Procedure IV**, is as follows:

Step 1. Is similar to Step 1 in Procedure II.

Step 2. Apply Procedure III to subnetwork L^f to obtain the minimal budget value

$$\min_{c(i,j)} \left[C / L^f \right] \qquad (16.2.10)$$

subject to (16.2.2-16.2.3) and

$$p(C) = \Pr\left\{ T\left[C / L^f \right] \le D \right\} = p \,. \qquad (16.2.11)$$

Step 3. Is similar to Step 3 in Procedure II.

Step 4. Calculate the average

$$\overline{C} = \frac{1}{M} \sum_{f=1}^{M} \left[C / L^f \right], \qquad (16.2.12)$$

where $\overline{C}$ is the minimal budget value of objective (16.2.8).

Procedures I-IV develop a solution to all the optimization problems at the project level for CAAN type projects.

§16.3 The hierarchical decision-making model

Let us turn to the description of the hierarchical decision-making model which is presented in Fig. 16.1. The model comprises three levels: company level → project level → subnetwork level. Several concepts are implemented in the model:

1. *Planning* is undertaken at the company and the project levels and results in:
   a) optimal budget assignment among the projects;
   b) determining, for each project separately on the basis of the available budget, the optimal joint variant to be controlled in the course of the project's realization.
2. *Controlling* is carried out at the subnetwork level and is based on:
   a) observing the joint variant periodically or at each key node;
   b) calculating confidence probabilities of the joint variant to meet the project's deadline;
   c) reassigning, if necessary, the available budget among the remaining activities of the joint variant to raise the confidence probability.
3. If it is anticipated that despite all control actions the joint variant cannot meet the chance constraint not to deviate from the target, an emergency is called and the model applies the company level to undergo budget reassignment. Thus, in the model to be considered *planning procedures are realized from top to bottom while control actions are introduced in the opposite direction, until at one of the levels a revised plan will be determined which enables all the projects to meet their due dates on time.*

We have used this general idea for planning and controlling network projects with arbitrary number of levels in the previous chapters.

4. The presented model is a *dynamic* model and includes t as one of the parameters. Therefore we have to use for each joint variant at moment $t \geq 0$ a three-indexed notation S_{rkt} where r is the index of the joint variant of the k-th project, $1 \leq r \leq m_{kt}$, $1 \leq k \leq n$, m_{kt} being the number of joint variants of the k-th project at moment t.
5. Two values are repeatedly calculated when controlling joint variants:
   a) $p_{rkt}(C)$ - confidence probability of the joint variant S_{rkt} to be accomplished on time with available budget $C = C_{kt}$ or $C = C_k(t)$, and
   b) $C_{rkt}(p)$ - budget value which enables S_{rkt} to meet the deadline with probability $p_{rkt} = p$. Note that $p_{rkt}(C)$ and $C_{rkt}(p)$ are direct and inverse functions which can be calculated by applying to S_{rkt} Procedures I (II) or inverse Procedures III (IV), correspondingly.
6. *All planning and control procedures at each hierarchical level are governed by the main problem of that level.* Thus, main Problems I-III form the framework of the hierarchical model (see Fig. 16.1).

Problem I, at the company level, is outlined in §§14.2-14.3 and enables optimal budget assignment among the projects. The problem is solved on the planning stage, at $t = 0$, and later on repeatedly resolved under emergency conditions, in the course of the projects' realization.

By means of certain assumptions imposed on the project's "cost-confidence probability" function, Problem I can be solved on the basis of calculating for each project budget values C^*_{kt} and C^{**}_{kt} which enable meeting the projects' due dates on time with probabilities p^*_k and p^{**}_k, correspondingly. Those budget values are determined at the project level and together with the initial pregiven data of the CAAN model they serve as the input for Problem I.

Problem II, at the project level, is solved separately for each project. The problem's solution enables determining and singling out from the initial CAAN model all the joint variants. For each joint variant separately the confidence probability of meeting the project's due date on time is calculated. This is carried out by using Procedure I if the joint variant is a PERT network, or Procedure II, in the case of a GERT network. Later on the optimal joint variant is determined and the progress of the project follows the optimal direction up to the nearest decision node.

In the course of carrying out the optimal joint variant the latter is controlled at the subnetwork level. If the joint variant is a PERT model budget values $c(i,j)_k$ have to be assigned to the corresponding activities $(i,j)_k$ by using Procedure I at the project level. If the optimal joint variant is a GERT model we cannot redistribute budget C_{kt} among all the activities. Controlling a GERT subnetwork is then based on Problem III which is solved at moment $t = 0$ and is repeatedly resolved in each sequentially encountered random alternative node of the remaining joint variant.

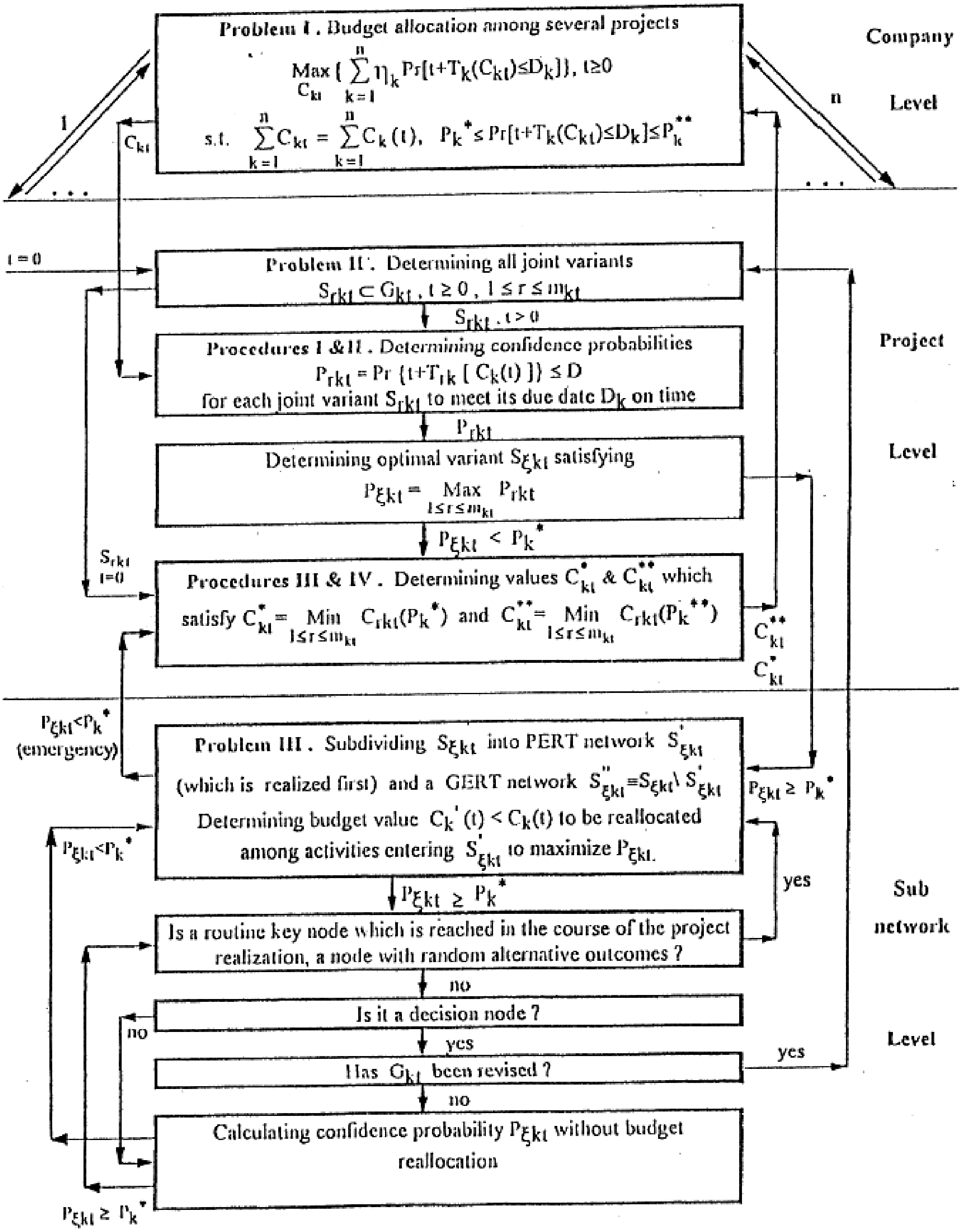

Figure 16.1. *The hierarchical decision-making model*

Problem III is as follows:

(a) divide the remaining joint variant into two subgraphs. The first one is a PERT graph which is realized before meeting the nearest random alternative node, while the second subgraph is a GERT model;

(b) divide the remaining budget into two parts. The first part is to be optimally reassigned among the activities of the first subgraph while the second part is left for the future, i.e., for the second subgraph. Budget reassignment is carried out by using Procedure I.

In the course of controlling the optimal joint variant N we suggest to calculate either periodically or in each sequentially encountered key node the probability $p_{\xi kt}$ of meeting the due date on time. That probability is calculated via simulation. If value $p_{\xi kt}$ becomes less than p_k^* we apply Problem III to undertake optimal budget reassignment among the optimal joint variant's activities. If that control action is ineffective, i.e., $p_{\xi kt}$ remains less than p_k^*, an emergency is called. Values C_{kt}^* and C_{kt}^{**} are then calculated for all the unaccomplished projects to carry out optimal budget reassignment at the company level.

§16.4 Decision-making at the project level

The main problem at that level is to determine and to single out from the initial CAAN graph all the joint variants (Problem II). The solution of that problem is outlined in Chapter 8 and is based on lexicographical scanning of the initial CAAN model. Thus, for each CAAN model G_{kt}, subnetworks $S_{rkt} \subset G_{kt}$, $1 \le r \le m_{kt}$, are determined. Every joint variant S_{rkt} is either a PERT or a GERT subnetwork.

Note that in the course of developing a project there may be changes in the parameters of some activities, e.g., durations, costs, outcome probabilities, etc., since activity networks are usually revised over time. In such a case Problem II has to be resolved in every sequentially encountered decision node since revising a project may result in changing its optimal joint variant. If the network remains unchanged Problem II has to be solved only once, at $t = 0$.

The next problem at the project level is to determine confidence probabilities of meeting the due date on time

$$p_{rkt} = \Pr\{t + T_{rk}[C_k(t)] \le D_k\}, \; 1 \le r \le m_{kt}, \tag{16.4.1}$$

for all joint variants $S_{rkt} \subset G_{kt}$. Here $T_{rk}[C_k(t)]$ is the random duration of joint variant S_{rkt} on the basis of the available budget $C_k(t)$.

Note that value $C_k(t)$ may be either budget C_{kt} assigned by the company level at moment $t \ge 0$, or the project's reduced budget for new joint variants $S_{rkt} \subset G_{kt}$, after resolving Problem II (see Fig. 16.1).

Values p_{rkt} on the basis of budget $C_k(t)$ can be calculated by using either Procedure I (if S_{rkt} is a PERT model) or Procedure II (in the case of a GERT model). Note that in the first case determining p_{rkt} is carried out together with optimal budget reassignment among activities $(i,j)_k \in S_{rkt}$. Thus, in this case using Procedure I delivers a heuristic solution to the optimization problem: determine optimal values $c(i,j)_k$ to maximize

$$\max_{c(i,j)_k} \Pr\{t + T_{rk}[C_k(t)] \le D_k\} \tag{16.4.2}$$

subject to

$$\sum_{(i,j)_k \in S_{rkt}} c(i,j) = C_k(t), \tag{16.4.3}$$

$$c(i,j)_{k\min} \le c(i,j)_k \le c(i,j)_{k\max}\,. \tag{16.4.4}$$

If S_{rkt} is a GERT model value p_{rkt} is calculated but the budget is not reassigned among the activities.

After determining values p_{rkt} for each S_{rkt} the optimal joint variant $S_{\xi kt}$ is chosen which satisfies

$$p_{\xi kt} = \max_{1 \le r \le m_{kt}} p_{rkt}\,. \tag{16.4.5}$$

If $p_{\xi kt} \ge p_k^*$ holds, the optimal joint variant $S_{\xi kt}$ is controlled at the subnetwork level. Otherwise project k fails to meet its chance constraint (14.2.3) and the dual problem has to be applied for all S_{rkt} to determine two budget values:

$$C_{kt}^* = \min_{1 \le r \le m_{kt}} C_{rkt}(p_k^*) \tag{16.4.6}$$

and

$$C_{kt}^{**} = \min_{1 \le r \le m_{kt}} C_{rkt}(p_k^{**}),\ 1 \le k \le n\,. \tag{16.4.7}$$

If S_{rkt} is a PERT model the dual problem is as follows: determine optimal values $c(i,j)_k$ to minimize value $C_{rkt}(p_k^*)$

$$\min_{\{c(i,j)_k\}} [C_{rkt}(p_k^*)] \tag{16.4.8}$$

subject to (16.4.4) and

$$\Pr\{t + T_{rk}[C_{rkt}(p_k^*)] \le D_k\} = p_k^*\,. \tag{16.4.9}$$

This problem can be solved by applying Procedure III to each joint variant $S_{rkt} \subset G_{kt}$. If S_{rkt} is a GERT model, value $C_{rkt}(p_k^*)$ can be calculated by applying Procedure IV, but without optimal budget reassignment. Afterwards the company level is applied to solve Problem I to reassign the budget among the projects. Note that values C_{kt}^* and C_{kt}^{**} correspond to joint variants $S_{\xi_1 kt}$ and $S_{\xi_2 kt}$. In practice case $\xi_1 \ne \xi_2$ occurs very seldom but, generally speaking, values ξ_1 and ξ_2 may differ. It has been outlined in §14.2 that solving Problem I results in assigning to all projects except not more than one, values C_{kt}^* and C_{kt}^{**}. Since for those projects their optimal joint variant $S_{\xi kt}$ and values $p_{\xi kt} \ge p_k^*$ have already been determined before resolving Problem I, we can apply straightforward the subnetwork level. For only one project with values $C_{kt}^* < C_{kt} < C_{kt}^{**}$, value $p_{\xi kt}$ and optimal joint variant $S_{\xi kt}$ are to be determined.

§16.5 Controlling joint variants at the subnetwork level

16.5.1 Controlling joint variants of GERT type

To control the optimal joint variant $S_{\xi kt}$ at the subnetwork level, control actions outlined in §16.3 have to be undertaken. If $S_{\xi kt}$ is a GERT model choosing that joint variant at the project level does not mean that budget $C_k(t)$ has already been reassigned among activities $(i,j)_k \in S_{\xi kt}$. Moreover, since $S_{\xi kt}$ comprises random alternative outcomes, a certain part of the activities will not be carried out in the course of the project's realization. Thus, before the beginning of carrying the project out on the basis of joint variant $S_{\xi kt}$, the management has to reassign a part of budget $C_k(t)$ among those activities which will be realized first, and which precede the first random alternative outcome in $S_{\xi kt}$. It can be well-recognized that such activities will be carried out in any case. The governing problem at the subnetwork level is Problem III which is repeatedly resolved at each random alternative node. The problem is as follows:

a) divide the remaining joint variant $S_{\xi kt}$ at any moment t when a random outcome is met, into two subnetworks $S^{\oplus}_{\xi kt}$ and $S^{\oplus\oplus}_{\xi kt} = S_{\xi kt} \setminus S^{\oplus}_{\xi kt}$. Here $S^{\oplus}_{\xi kt}$ is a PERT network which will be realized before meeting the adjacent random outcome, while $S^{\oplus\oplus}_{\xi kt}$ comprises random outcomes and is a GERT network.

b) divide the remaining budget $C_k(t)$ into two parts: $C^{\oplus}_k(t)$ and $C^{\oplus\oplus}_k(t) = C_k(t) - C^{\oplus}_k(t)$. The first has to be utilized by realizing subnetwork $S^{\oplus}_{\xi kt}$, while the second part is to be left for the future subnetwork $S^{\oplus\oplus}_{\xi kt}$.

c) reassign budget $C^{\oplus}_k(t)$ optimally among activities $(i,j)_k \in S^{\oplus}_{\xi kt}$ in order to maximize the confidence probability of meeting the deadline, i.e.,

$$p_{\xi kt} = \max_{\{c(i,j)_k\},(i,j)_k \in S^{\oplus}_{\xi kt}} \Pr\left\{t + T_{\xi^{\oplus}k}\left[C^{\oplus}_k(t)\right] + T_{\xi^{\oplus\oplus}k}\left[C^{\oplus\oplus}_k(t)\right] \le D_k\right\} \tag{16.5.1}$$

subject to

$$C^{\oplus}_k(t) + C^{\oplus\oplus}_k(t) = C_k(t), \tag{16.5.2}$$

$$S^{\oplus}_{\xi kt} \cup S^{\oplus\oplus}_{\xi kt} = S_{\xi kt}, \tag{16.5.3}$$

$$\sum_{(i,j)_k \in S^{\oplus}_{\xi kt}} c(i,j)_k = C^{\oplus}_k(t), \tag{16.5.4}$$

$$c(i,j)_{k\min} \le c(i,j)_k \le c(i,j)_{k\max}. \tag{16.5.5}$$

The suggested step-wise heuristic to solve Problem III is as follows:

Step 1. Single out subnetwork $S^{\oplus}_{\xi kt} \subset S_{\xi kt}$ of PERT type which precedes the first random alternative outcome.

Step 2. Simulate the non-contradictory outcomes in $S^{\oplus\oplus}_{\xi kt}$ to obtain a PERT subnetwork $L^{\oplus\oplus}_{f\xi kt}$. Here f denotes the number of the simulation run.

Step 3. Unify subnetworks $S^{\oplus}_{\xi kt}$ and $L^{\oplus\oplus}_{f\xi kt}$ to obtain a simulated full joint variant

$L_{f\xi kt}$. Note that $L_{f\xi kt} \supset S_{\xi kt}^{\oplus}$.

Step 4. Reassign budget C_{kt} among activities $(i,j) \in L_{f\xi kt}$ to maximize the probability of meeting the due date on time. This can be carried out by applying Procedure I to subnetwork $L_{f\xi kt}$ (Procedure I is outlined in §14.4). Denote budget values $c(i,j)_k$ obtained in the course of carrying out Step 4, by $c(i,j)_k^f$.

Step 5. Calculate

$$C_k^{\oplus}(t)^f = \sum_{(i,j)_k^f \in S_{\xi kt}^{\oplus}} c(i,j)_k^f ,$$

$C_k^{\oplus}(t)^f$ being the part of budget $C_k(t)$ assigned to PERT subnetwork $S_{\xi kt}^{\oplus}$ in the course of the f-th simulation run.

Step 6. Repeat Steps 2-5 M times to obtain representative statistics.

Step 7. Calculate

$$C_k^{\oplus}(t) = \overline{C_k^{\oplus}(t)^f} = \frac{1}{M}\sum_{f=1}^{M} C_k^{\oplus}(t)^f .$$

Step 8. Solve budget reassignment optimization problem: reassign budget value $C_k^{\oplus}(t)$ among activities $(i,j)_k \in S_{\xi kt}^{\oplus}$ to minimize the duration of subnetwork $S_{\xi kt}^{\oplus}$. The step is carried out by implementing Procedure I.

Thus the project management obtains values $c(i,j)_k$ for $(i,j)_k \in S_{\xi kt}^{\oplus}$ which enable to proceed with the project's realization, until the next adjacent random alternative milestone.

Steps 1-8 are repeatedly carried out for the reduced joint variant after reaching each random alternative outcome during the progress of the project.

16.5.2 Controlling joint variants by confidence probabilities

If a routine key node reached in the course of the project's realization is a decision node, Problem II is applied to determine possible new joint variants, on condition that G_{kt} has been revised. Otherwise the last problem at the subnetwork level is applied, to determine the confidence probability $p_{\xi kt}$ (budget reassignment is not required) by means of simulation (see Fig. 16.1).

The problem of calculating confidence probabilities can be solved in every sequentially reached key node by using intensive simulation. The corresponding step-wise algorithm to calculate $p_{\xi kt}$ without budget reassignment is as follows:

Step 1. If $S_{\xi kt}$ at moment t is a PERT subnetwork apply the next step. Otherwise apply Step 7.

Step 2. Simulate duration values $t(i,j)_k$ for each activity $(i,j)_k \in S_{\xi kt}$. This can be easily undertaken since all budget values $c(i,j)_k$, $(i,j)_k \in S_{\xi kt}$, are fixed and remain unchanged between two adjacent budget reallocations. The techniques to be used are similar to those outlined in

§§14.4-14.5.

Step 3. Determine for the PERT network with simulated deterministic values $t(i,j)_k$ its critical path length L^f_{cr}.

Step 4. If $t + L^f_{cr} \le D_k$ holds counter $1 + M^* \Rightarrow M^*$ works. Otherwise apply the next step with unchanged M^*.

Step 5. Repeat Steps 2-4 M times to obtain representative statistics.

Step 6. Calculate

$p_{\xi kt} = \frac{M^*}{M}$ and proceed to Step 12.

Step 7. Simulate GERT network $S_{\xi kt}$ to obtain a full variant $L_{f\xi kt}$.

Step 8-10. Steps 8-10 are similar to Steps 2-4, besides that $S_{\xi kt}$ is replaced by $L_{f\xi kt}$.

Step 11. Repeat Steps 7-10 M times. Thus each simulation run comprises simulations of random outcome directions in a GERT network $S_{\xi kt}$, together with simulations of random durations $t(i,j)_k$ for a simulated PERT subnetwork $L_{f\xi kt}$. Return to Step 6 .

Step 12. The algorithm terminates.

If $p_{\xi kt} \ge p^*_k$ the progress of the project continues. Otherwise, i.e., $p_{\xi kt} < p^*_k$ (see Fig. 16.1), we apply Problem III to undertake optimal budget reassignment among activities $(i,j)_k \in S_{\xi kt}$. If this control action fails too, an emergency is declared and the hierarchical decision-making model has to undergo budget reassignment at the company level (Problem I), as it is presented in Fig. 16.1.

Note that for PERT or PERT-COST projects it is sometimes reasonable to implement on-line control, similar to that outlined in Chapter 15:

- at the project level (when either the project or its optimal joint variant $S_{\xi kt}$ are PERT projects), or
- at the lower level when subgraph $S^{\oplus}_{\xi kt}$ is a PERT project.

On-line control is based on determining inspection (control) points to evaluate the *actual* progress of the project. At those control points the project is inspected and decision-making centers on:

- determining the next control point;
- reassigning if necessary the remaining budget $C_k(t)$ among the remaining project's activities (t is the routine control point);
- applying to the company level to obtain support from more successful projects.

Note that between two adjacent control points we do not introduce any control actions *besides the project's scheduling*. It has to be pointed out (see Chapters 1, 3) that control actions do not abolish various scheduling rules, which have been outlined in Chapters 11-13.

§16.6 Optimization problems for joint variants comprising activities of random durations and consuming renewable resources

16.6.1 Introduction

It can be well-recognized that the outlined above, in §§16.1-16.5, techniques are based on the following assumptions:

- the outlined hierarchical model enables monitoring alternative stochastic network projects of PERT-COST type only;
- all activities entering the model are of deterministic duration;
- all projects are of equal importance.

However, an essential part of R&D projects (especially innovative ones) possess activities of random duration (see Chapter 2) and consume renewable resources of different types (see §§6.2-6.3, 12.1-12.4). Being related to different areas of technology, industry and life sciences, certain innovative projects are of different importance and have to be monitored with different priorities. All those additional assumptions have to be taken into account for the outlined below optimization models.

In the present §16.6 we will outline the case of consuming renewable generalized resource units (GRU) which have been introduced in §§6.2-6.3 to solve optimization problems under chance constraints. We will assume that each activity entering the model is carried out by means of a certain amount of GRU (externally pregiven).

We will consider the general optimization problem of reallocating a certain restricted amount of GRU at the company's disposal among several simultaneously carried out alternative stochastic network projects of CAAN type. Thus, the optimization problem is of mixed type since it refers both to the planning (at $t=0$) and control stages (at $t>0$).

The outlined below optimization model is actually a combination of models outlined in §§6.2-6.3, 8.1-8.5, 14.2, 16.1-16.3. At moment $t=0$ the model has to reallocate optimally among the projects the total amount of GRU; further on, all projects work independently. At every encountered in the course of each project's realization decision-making node $\bar{\alpha}$ the model has to choose the optimal joint variant for that project.

16.6.2 Notation

Let us introduce the following terms:

$G_k(N,A)$ - the k-th alternative stochastic project of CAAN type, $1 \le k \le m$;

m - the number of projects;

G_{kt} - the remaining "truncated" k-th network project at moment $t \ge 0$; $G_{k0} = G_k(N,A)$;

D_k - the due date for the k-th project (pregiven);

$(i,j)_k$ - activity (i,j) of the k-th project;

N_t - the total number of GRU at the company's disposal at moment $t \ge 0$, $N_0 = N$ (N pregiven);

$t(i,j)_{kn}$ - random duration of activity $(i,j)_k$ on condition that the activity is carried out by means of $n(i,j)_k$ GRU, $n(i,j)_{k\min} \le n(i,j)_k \le n(i,j)_{k\max}$ (probability law externally pregiven);

$n(i,j)_{k\min} \ge 1$ - lower bound of GRU to operate activity $(i,j)_k$ (pregiven);

$n(i,j)_{k\max}$ - upper bound of GRU to operate activity $(i,j)_k$ (pregiven);

$\bar{t}(i,j)_{kn}$ - average value of $t(i,j)_{kn}$;

p_k^* - the least permissible probability for the k-th project to meet its due date on time (pregiven);

$p_k^* > p_k^*$ - the probability that practically guaranties for the k-th project to be completed on time (pregiven);

n_{kt} - the number of GRU assigned to the k-th project G_{kt} at moment $t \ge 0$ (optimal value to be determined); at moment $t=0$, $n_{kt} = n_k$;

$T_{kt}(n_{kt})$ - random duration of project G_{kt} corresponding to the amount n_{kt} of GRU assigned from the company;

$S_{rkt} \subset G_{kt}$ - the r-th joint variant of project G_{kt} at moment $t \ge 0$ (a subnetwork of PERT or GERT type obtained by implementing the algorithm outlined in §§8.1-8.5);

$S_{\xi kt} \subset G_{kt}$ - the optimal joint variant of project G_{kt};

L_{frkt} - full (overall) variant (realization of a GERT network S_{rkt} by simulating random alternative branching of the f-th simulation), $L_{frkt} \subset S_{rkt}$;

$n(L_{frkt})$ - the number of GRU assigned to the full variant L_{frkt};

m_{kt} - the number of joint variants in project G_{kt};

$n(S_{rkt})$ - the number of GRU assigned to S_{rkt} at moment $t \ge 0$;

η_k - priority value of the k-th project (pregiven). Similarly to §16.1, $\eta_{k_1} > \eta_{k_2}$ means higher importance of $G_{k_1}(N,A)$ relatively to $G_{k_2}(N,A)$.

Note that at $t=0$: $n_{kt} = n_k$, $T_{kt}(n_{kt}) = T_k(n_k)$, $S_{rkt} = S_{rk}$, $L_{frkt} = L_{frk}$, $m_{kt} = m_k$, $S_{\xi kt} = S_{\xi k}$.

The following rules, similar to those outlined in §§6.2-6.3, have to be imbedded in the problem's solution:

- each activity has to be operated by at least one GRU;
- one GRU cannot operate simultaneously more than one activity;
- GRU can be transferred from one project to another at emergency moments *only*.

16.6.3 The model

Similarly to the model outlined in §§16.1-16.5, the model under consideration:

a) At moment $t = 0$ reassigns the entire amount of GRU at the company's disposal among the projects $G_k(N, A)$, $1 \le k \le m$, in order to:

- determine optimal values n_k, $1 \le k \le m$, to maximize

$$\max_{n_k}\left\{\sum_{k=1}^{m}\left[\eta_k \cdot \Pr\{T_k(n_k) \le D_k\}\right]\right\} \tag{16.6.1}$$

subject to

$$\sum_{k=1}^{m} n_k \le N, \tag{16.6.2}$$

$$n_k = n(S_{\xi k}),\ 1 \le r \le m_k, \tag{16.6.3}$$

$$n(S_{\xi k}) \ge \max_{(i,j)_k \in S_{\xi k}} n(i,j)_{k\min}, \tag{16.6.4}$$

where $S_{\xi k}$ is a PERT subnetwork,

$$n(L_{f\xi k}) \ge \max_{f}\left[\max_{(i,j)_k \in L_{f\xi k}} n(i,j)_{k\min}\right], \tag{16.6.5}$$

where $L_{f\xi k}$ is a full variant of a GERT subnetwork $S_{\xi k}$,

$$\Pr\{T_k \le D_k\} \ge p_k^*,\ 1 \le k \le m. \tag{16.6.6}$$

Objective (16.6.1) is evident since the management takes all measures to accomplish most important projects as soon as possible. Restriction (16.6.3) means that each k-th project, after obtaining n_k GRU items, has to consume them in order to carry out the optimal joint variant of that project. Restriction (16.6.4) means that for the case of the optimal joint variant of PERT type value n_k has to exceed all the lower bounds of $n(i,j)_k$ for activities entering that joint variant. In the case, when a joint variant (or the project itself) is a GERT subnetwork, restriction (16.6.5) enables that project to be carried out. Restriction (16.6.6) honors the chance constraints.

b) After determining values n_k each project $G_k(N, A)$ proceeds working independently. At each decision point $t \ge 0$ with deterministic alternative outcomes the model:

- singles out all the joint variants and determines the optimal one;
- checks the possibility of reaching the due date D_k on time subject to (16.6.6).

The first stage is outlined and described in depth in §§8.1-8.5. The second stage has to be formalized as follows:

- for each project G_{kt}, $1 \le k \le m$, $t > 0$, solve the problem: determine the *minimal* capacity of GRU n_{kt}^{opt} satisfying

$$n_{kt}^{opt} = n(S_{\xi kt}), \tag{16.6.7}$$

$$n(S_{\xi kt}) \ge \max_{(i,j)_k \in S_{\xi kt}} n(i,j)_{k\min}, \tag{16.6.8}$$

(if $S_{\xi kt}$ is a PERT subnetwork),

$$n\left(L_{f\xi kt}\right)\geq \max_{f}\left[\max_{(i,j)_k\in L_{f\xi kt}} n(i,j)_{k\min}\right], \qquad (16.6.9)$$

(if $S_{\xi kt}$ is a GERT subnetwork and L_{frkt} is its f-th full variant),

$$\Pr\left\{\left[t+T_{kt}\left(n_{kt}^{opt}\right)\right]\leq D_k\right\}\geq p_k^* . \qquad (16.6.10)$$

Call henceforth optimization problem (16.6.7-16.6.10) Problem A.

If it turns out that for one of the projects Problem A cannot be solved, an emergency is declared, and the total amount N_t at the company's disposal (which may be changed overtime in comparison with N) has to be reallocated among the projects.

16.6.4 *Subsidiary optimization Problem A for a single project*

In order to simplify the problem we will omit the project's index k. The problem's initial data is as follows:

Given:

$G(N,A)$ - stochastic alternative network project of CAAN type;

$(i,j)\in G(N,A)$ - activity entering the project;

$n(i,j)$ - number of GRU to operate activity (i,j), $n(i,j)_{\min}\leq n(i,j)\leq n(i,j)_{\max}$;

$n(i,j)_{\min}\geq 1$ - lower bound of value $n(i,j)$ (pregiven);

$n(i,j)_{\max}$ - upper bound of value $n(i,j)$ (pregiven);

n - restricted number of GRU to carry out the project;

$t[(i,j)/n(i,j)]$ - random value of activity (i,j) duration, on condition that $n(i,j)$ GRU participate in executing (i,j) (all p.d.f. of $t[(i,j)/n(i,j)]$ are externally pre-given);

D - the project's due date;

p^* - chance constraint for the project to meet its due date on time;

S_r - the r-th joint variant of project $G(N,A)$;

The problem is to determine the minimal total n^{opt} of GRU in order to carry out the project on time with probability not less than p^*.

The step-wise solution of Problem A is as follows:

Step 1. According to the techniques outlined in §§8.1-8.5, single out all the joint variants entering $G(N,A)$. Let them be m, S_1, $S_2,\ldots,$ $S_r,\ldots,$ S_m.

Step 2. For each routine joint variant S_r check, to which type of subnetwork (PERT or GERT) it does belong. If S_r is a PERT type network, apply the next step. Otherwise proceed to Step 4.

Step 3. Solve problem (16.6.7-16.6.10) to minimize n^{opt} for a network with deterministic structure and random activities durations. The solution is outlined in §6.2. If the determined value satisfies $n^{opt}\leq n$, go to Step 7. Otherwise apply Step 6.

Step 4. In case of a GERT type joint variant proceed as follows:

4.1 Determine the minimal value

$$n_{\min} = \max_f \left[\max_{(i,j)\in S_r} n(i,j)_{\min} \right], \qquad (16.6.11)$$

where f denotes the number of a full variant $L_{fr} \subset S_r$ obtained by simulating random alternative outcomes in a GERT subnetwork.

4.2 Undertake numerous simulation runs (M runs) in order to obtain representative statistics). Number M can be determined by techniques outlined in Chapter 3 [27]. In the course of a single simulation run we obtain a full variant, i.e., a chain of several consecutive activities.

4.3 Simulate random durations of all activities entering the full variant according to their p.d.f. $t\left[(i,j)_{n_{\min}}\right]$. Calculate the summarized duration of the full variant. Assume it to be equal the simulated project's duration. Compare the latter with due date D.

4.4 On the basis of M simulation runs calculate average value $\overline{p} = \frac{M^*}{M}$, where M^* stands for the number of simulation runs with the project accomplished in time not exceeding D. If $\overline{p} \geq p^*$, go to Step 7. Otherwise apply Step 5.

Step 5. Increase $n_{\min}$ by one, $n_{\min} + 1 \Rightarrow n_{\min}$. If $n_{\min}$ exceeds n, proceed to Step 6. Otherwise return to Substep 4.2.

Step 6. Withdraw the joint variant under consideration and proceed examining the next one. Go to Step 7.

Step 7. Check if all the joint variants entering the project, have been examined. If not, return to Step 2. Otherwise proceed to the next step.

Step 8. Examine all joint variants which have not been withdrawn yet, and choose the optimal one with the minimal value n^{opt}. The obtained value is the problem's solution.

If all the joint variants have been withdrawn, Problem A cannot be solved and either n has to be increased, or the due date has to be changed, or probability p^* must be decreased.

16.6.5 The general problem's solution

The enlarged step-wise algorithm of solving problem (16.6.1-16.6.6) resembles the algorithm outlined in §14.2 for the case of several PERT-COST projects with different priorities and is outlined below.

Note, in advance, that the presented algorithm for solving problem (16.6.1-16.6.6) does not depend on the time moment t. If $t > 0$, i.e., in the case emergency was called, we have to change all terms in (16.6.1-16.6.6) as follows:

$n(i,j)_k \Rightarrow n(i,j)_{kt}$,

$S_{rk} \Rightarrow S_{rkt}$,

$n_k^{opt} \Rightarrow n_{kt}^{opt}$,

$N \Rightarrow N_t$, etc.

Thus, we will not rewrite model (16.6.1-16.6.6) with additional term t. The step-wise algorithm of solving the general optimization problem is as follows:

Step 1. For all m projects G_{kt}, $1 \le k \le m$, independently, solve at moment $t \ge 0$ subsidiary optimization Problem A outlined in *16.6.4*, both for probability values p_k^* and p_k^{**}. Denote solutions of the problem by $n_k^{(opt)*}$ and $n_k^{(opt)**}$, $1 \le k \le m$, correspondingly.

Step 2. Assign to all projects G_{kt} their minimal resource capacities $n_k^{(opt)*}$.

Step 3. Calculate the remaining resource reserve $N_t - \sum_{k=1}^{n} n_k^{(opt)*} = \Delta N_t$. If $\Delta N_t < 0$, the problem has no solution, and the system's structure, i.e., either N_t, or $\{p_k^*\}$, or $\{D_k\}$, has to be changed. Otherwise apply the next step.

Step 4. Reorder projects G_{kt} in ascending order of their priority indices ρ_k. Let the new projects' ordinal numbers be $f_1, f_2, ..., f_m$.

Step 5. Set $j = 1$.

Step 6. Calculate

$$\gamma_j = \min\left[\left(n_j^{(opt)**} - n_j^{(opt)*}\right), \Delta N_t\right].$$

Step 7. Determine for project $G_{f_j t}$ its new optimal resource capacity

$$n_{f_j}^* + \gamma_j = n_{f_j}^{opt}.$$

Step 8. Update the remaining resource reserve $\Delta N_t - \gamma_j \Rightarrow \Delta N_t$. If $\Delta N_t = 0$ go to Step 11. Otherwise apply the next step.

Step 9. Set $j + 1 \Rightarrow j$.

Step 10. If $j < m$ return to Step 6. Otherwise apply the next step.

Step 11. The algorithm terminates.

The proof of the solution's optimality fully resembles the proof outlined in §14.2. As to techniques of GRU reallocation at emergency moments t_{em}, when for at least one of the projects Problem A cannot be solved, they are outlined in §6.3 [84].

Note, in conclusion, that the case of alternative projects requiring GRU and with equal priority indices η_k can be examined by implementing a combination of model (16.6.1-16.6.6) and the algorithm for PERT-COST projects outlined in §14.3.

§16.7 Numerical example

In conclusion, the performance of the three-level decision-making model outlined above, in §§16.1-16.5, is illustrated by a numerical example.

The company is faced with realizing three stochastic network projects of CAAN type. The projects' initial data is presented in Tab. 16.1-16.3 [65]. The projects' parameters are as follows:

$D_1 = 50$; $p_1^* = 0.60$; $p_1^{**} = 0.85$; $\eta = 2$;

$D_2 = 80$; $p_2^* = 0.70$; $p_2^{**} = 0.90$; $\eta = 3$;

$D_3 = 100$; $p_3^* = 0.70$; $p_3^{**} = 0.95$; $\eta = 5$;

Table 16.1. Initial data for Project 1

No.	i	h	j	p	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	1	2	1	14	44	220	841
2	1	2	3	1	23	38	182	560
3	1	3	4	1	39	65	430	685
4	2	1	5	0	35	57	390	882
5	2	2	6	0	21	58	320	795
6	3	1	8	1	38	55	160	600
7	3	2	11	1	38	57	200	400
8	4	1	14	0	36	77	683	920
9	4	2	15	0	50	75	610	820
10	5	1	7	0	34	54	220	780
11	6	1	7	0	35	69	290	740
12	7	1	17	1	17	55	192	560
13	7	2	20	1	14	56	220	800
14	8	1	9	0	18	42	100	550
15	8	2	10	0	17	57	100	600
16	9	1	10	0	15	40	300	460
17	10	1	17	1	35	68	200	380
18	10	2	20	1	19	55	200	620
19	10	3	21	1	25	50	342	500
20	11	1	12	0	15	40	405	740
21	11	2	13	0	17	55	380	710
22	12	1	18	0	16	35	360	580
23	13	1	18	0	15	20	205	440
24	14	1	16	0	18	57	500	700
25	14	2	15	0	18	42	400	610
26	15	1	15	0.5	27	59	490	820
27	15	2	22	0.5	39	75	508	820
28	16	1	21	0	37	75	520	804
29	17	1	23	0.6	18	42	320	520
30	17	2	19	0.4	38	58	510	810
31	18	1	21	0.8	15	37	98	202
32	18	2	22	0.2	15	37	183	294

The total initial budget at the company's disposal to carry out the projects equals $C = 1{,}000$. In order to illustrate the performance of the model, we will outline below:

- optimal budget reallocation among the projects obtained by solving Problem I (at the company level);
- optimal joint variants for each project obtained by solving Problem II and applying Procedures I-IV (at the project level);

Table 16.2. Initial data for Project 2

No.	i	h	j	p	$c(i,j)_{min}$	$c(i,j)_{max}$	$A(i,j)$	$B(i,j)$
1	1	1	2	0	32	48	533	887
2	1	2	3	0	31	46	494	864
3	1	3	4	0	31	48	573	922
4	1	4	5	0	21	45	510	898
5	1	5	6	0	31	48	523	820
6	1	6	7	0	38	60	560	900
7	2	1	3	1	38	65	489	667
8	2	2	15	1	23	32	731	919
9	3	1	13	0.6	32	55	510	720
10	3	2	14	0.2	34	52	440	640
11	3	3	15	0.2	35	50	490	720
12	4	1	9	1	30	52	500	950
13	4	2	14	1	37	57	555	900
14	5	1	13	0	30	50	355	840
15	6	1	9	0	22	56	480	850
16	7	1	8	0	33	55	487	855
17	8	1	10	1	32	50	420	932
18	8	2	11	1	36	51	412	888
19	9	1	12	0.7	31	45	490	854
20	9	2	11	0.3	33	50	502	888
21	10	1	20	1	22	45	499	873
22	10	2	21	1	35	50	505	840
23	11	1	16	1	37	55	550	870
24	11	2	21	1	30	52	500	810
25	12	1	18	0	30	50	490	820
26	13	1	17	0.6	30	52	660	820
27	13	2	18	0.1	30	50	518	878
28	13	3	19	0.3	24	48	663	810
29	14	1	16	1	31	47	658	802
30	14	2	17	1	31	47	623	894
31	15	1	17	0	23	45	660	804
32	15	2	19	0	24	49	485	820
33	16	1	22	0	34	51	470	893
34	17	1	22	1	35	60	282	553
35	17	2	23	1	56	75	874	940
36	18	1	23	0	32	48	480	820
37	19	1	23	0	32	45	440	847
38	20	1	21	0	31	44	432	872
39	21	1	23	0	33	44	463	895
40	22	1	23	0	32	75	222	528

- control actions at the subnetwork level for Project 3 with the highest priority.

Table 16.3. Initial data for Project 3

No.	i	h	j	p	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	1	2	1	13	26	252	460
2	1	2	3	1	13	28	166	580
3	1	3	4	1	13	20	200	585
4	1	4	5	1	17	25	190	642
5	1	5	6	1	12	18	160	405
6	1	6	7	1	18	30	160	600
7	2	1	5	0.6	13	25	150	500
8	2	2	15	0.4	13	28	183	620
9	3	1	13	1	15	25	210	720
10	3	2	14	1	14	24	120	640
11	3	3	15	1	15	20	190	620
12	4	1	9	0.3	15	24	400	850
13	4	2	12	0.4	15	20	100	520
14	4	3	14	0.3	17	37	300	800
15	5	1	13	0	15	20	205	640
16	6	1	10	0	12	15	480	850
17	7	1	8	0	13	25	162	490
18	8	1	9	1	12	25	230	462
19	8	2	11	1	16	20	280	580
20	9	1	11	0.9	12	20	190	620
21	9	2	12	0.1	13	20	282	560
22	10	1	20	0	12	30	120	580
23	10	1	21	1	15	28	205	480
24	11	2	16	1	17	30	300	800
25	11	2	21	0	18	22	400	810
26	12	1	18	0	15	20	190	620
27	13	1	17	1	14	24	320	620
28	13	2	19	1	14	28	263	610
29	14	1	16	0.7	12	24	100	600
30	14	2	17	0.3	12	24	141	600
31	15	1	17	0	13	15	360	740
32	15	2	19	0	14	19	165	620
33	16	1	22	0	14	30	240	700
34	17	1	18	1	21	32	352	500
35	17	2	22	1	18	25	260	720
36	18	1	22	0	18	26	260	440
37	19	1	22	0	25	30	380	787
38	20	1	21	0.4	12	28	262	542
39	20	2	22	0.6	15	22	287	482
40	21	1	22	0	13	23	323	595

Project 1 comprises 7 joint variants. Using Procedures I-IV results in determining values

$C_{10}^{*} = 208$; $C_{10}^{**} = 221$; $\xi_1 = \xi_2 = 5$;

For Project 2 which comprises 96 joint variants we obtain

$C_{20}^{*} = 709$; $C_{20}^{**} = 736$; $\xi_1 = \xi_2 = 24$;

while for Project 3 with 20 joint variants values

$C_{30}^{*} = 60$; $C_{30}^{**} = 73$; $\xi_1 = \xi_2 = 9$;

have been calculated. Thus, values 5, 24 and 9 are indices of the optimal joint variants of Projects 1-3, correspondingly.

Solving the budget reallocation problem at $t = 0$ (see §16.2) we obtain [65]:

1. $C_{10} = 218$. This value assigned to Project 1 enables maximal confidence probability $p_{\xi kt} = p_{5,1,0} = \Pr\{T_1[C_{10}] < 50\} = 0.75$ for the 5-th joint variant $S_{\xi kt} = S_{5,1,0}$. This optimal joint variant is presented in Tab. 16.4 and is a PERT subnetwork. Note that due to $C_{10}^{*} < C_{10} < C_{10}^{**}$ value $p_{5,1,0}$ satisfies $p_1^{*} = 0.60 < p_{5,1,0} = 0.75 < p_1^{**} = 0.85$.
2. $C_{20} = 709 = C_{20}^{*}$. Since C_{20} is the minimal possible value which can be assigned to Project 2, the corresponding confidence probability is also the minimal one: $p_{\xi kt} = p_{24,2,0} = 0.70$ for the 24-th joint variant. The optimal joint variant $S_{\xi kt} = S_{24,2,0}$ is presented in Tab. 16.5 and is a GERT subnetwork.
3. $C_{30} = 73 = C_{30}^{*}$. Since C_{30} is the maximal possible budget which can be assigned to Project 3, it corresponds to the maximal confidence probability $p_{\xi kt} = p_{9,3,0} = 0.95$ for the 9-th joint variant $S_{9,3,0}$. The latter is presented in Tab. 16.6 and is a GERT subnetwork.

Table 16.4. Project 1: Optimal joint variant #5 after budget reassignment (PERT type subnetwork)

i	h	j	p	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	2	3	1	23	38	182	560
3	1	8	1	38	55	160	600
8	1	9	0	18	42	100	550
8	2	10	0	17	57	100	600
9	1	10	0	15	40	300	460
10	3	21	1	25	50	342	500

In order to illustrate control actions at the subnetwork level we will outline a simulation run for the most important project, i.e., Project 3. Assume for simplicity that the project will not be altered in the course of its realization.

Since the project's optimal joint variant $S_{\xi kt} = S_{9,3,0}$ is a GERT subnetwork we have to solve first, at $t = 0$, Problem II to single out subgraphs $S_{9,3,0}^{\oplus}$ and $S_{9,3,0}^{\oplus\oplus}$, to determine budget values $C_3^{\oplus}(0)$ and $C_3^{\oplus\oplus}(0)$ and to reassign budget $C_3^{\oplus}(0)$ among activities $(i,j)_3 \in S^{\oplus}_{9,3,0}$. The solution is presented in Tab. 16.3 and is as follows:

Table 16.5. Project 2: Optimal joint variant #24 after budget reassignment (GERT type subnetwork)

i	h	j	p	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
1	1	2	0	32	48	533	887
1	2	3	0	31	46	494	864
1	3	4	0	31	48	573	922
1	4	5	0	21	45	510	898
1	5	6	0	31	48	523	820
1	6	7	0	38	60	560	900
2	1	3	1	38	65	489	667
3	1	13	0.6	32	55	510	720
3	2	14	0.2	34	52	440	640
3	3	15	0.2	35	50	490	720
4	1	9	1	30	52	500	950
5	1	13	0	30	50	355	840
6	1	9	0	22	56	480	850
7	1	8	0	33	55	487	855
8	2	11	1	36	51	412	888
9	1	11	0.7	31	45	490	854
9	2	12	0.3	33	50	502	888
11	2	21	1	30	52	50	810
12	1	18	0	30	50	490	820
13	1	17	0.6	30	52	660	820
13	2	18	0.1	30	50	518	878
13	3	19	0.3	24	48	663	810
14	2	17	1	31	47	623	894
15	1	17	0	23	45	660	840
15	2	19	0	24	49	485	820
17	2	23	1	56	75	874	940
18	1	23	0	32	48	480	820
19	1	23	0	32	45	440	847
21	1	23	0	33	44	463	895

1) Subgraph $S^{\oplus}_{9,3,0}$ (a PERT network) comprises activities $(1,3)_3$ and $(3,14)_3$ which at any rate will be carried out before the first random outcome (node 14).
2) Subgraph $S^{\oplus\oplus}_{9,3,0}$ (a GERT network) comprises two connecting paths $(14,16)_3 \rightarrow (16,22)_3$ and $(14,17)_3 \rightarrow (17,22)_3$ with a random alternative outcome in node 14.
3) Values $C^{\oplus}_3(0)$ and $C^{\oplus\oplus}_3(0)$ equal 35 and 38, correspondingly.
4) Reassigning budget $C^{\oplus}_3(0)=35$ among activities $(1,3)_3$ and $(3,14)_3$ results in

$c(1,3)_3 = 17$ and $c(3,14)_3 = 18$.

5) Budget value $C_3^{\oplus\oplus}(0) = 38$ is left for developing subnetwork $S_{9,3,0}^{\oplus\oplus}$ after node 14 is reached.

Table 16.6. Project 3: Optimal joint variant #9 after budget reassignment (GERT type subnetwork)

	i	h	j	p	$c(i,j)_{\min}$	$c(i,j)_{\max}$	$A(i,j)$	$B(i,j)$
$S_{9,3,0}^{\oplus}$	1	2	3	1	13	28	166	580
	3	2	14	1	14	24	120	640
$S_{9,3,0}^{\oplus\oplus}$	14	1	16	0.7	12	24	100	600
	14	2	17	0.3	12	24	141	600
	16	1	22	0	14	30	240	700
	17	2	22	1	18	25	260	720

Simulating activity duration $t(1,3)_3$ with density function (14.1.1) and budget $c(1,3)_3 = 17$ results in $t(1,3)_3 = 26$. Thus, decision node 3 will be reached at $t = 26$ and since $G_3(N,A)$ has not been revised, calculating confidence probability at $t = 26$ has to be undertaken. We obtain $p_{\xi kt} = p_{9,3,26} = 0.95$ and since that value exceeds $p_3^* = 0.70$ we proceed developing the project. Simulating $t(3,14)_3$ with assigned budget $c(3,14)_3 = 18$ results in $t(3,14)_3 = 19$. Thus, node 14 is reached at $t = 26 + 19 = 45$. Simulating the outcome direction in node 14 we obtain a subgraph comprising activities $(14,17)_3$ and $(17,22)_3$. The calculated confidence probability is $p_{\xi kt} = p_{9,3,45} = 0.94 > 0.70$. Reallocating the remaining budget $C_3^{\oplus\oplus}(0) = 38$ among activities $(14,17)_3$ and $(17,22)_3$ results in $c(14,17)_3 = 19$ and $c(17,22)_3 = 19$. Simulating activity duration $t(14,17)_3$ with density function (14.1.1) and budget $c(14,17)_3 = 19$ we obtain $t(14,17)_3 = 21$. Thus, key node 17 will be met at $t = 45 + 21 = 66$. Since node 17 is a decision node we have to recalculate the confidence probability at $t = 66$, i.e., to obtain value $p_{9,3,66}$. Determining $p_{9,3,66} = 0.89 > 0.70$ and proceeding to carry out the project we simulate $t(17,22)_3 = 25$. Thus, we reach the sink node 22 at $t = 66 + 25 = 91 < D_3 = 100$, and Project 3 meets its due date on time.

§16.8 Conclusions

The following conclusions can be drawn from the Chapter:

1. The outlined above hierarchical models refer to the decision support models class. Decision-making can be undertaken by the project manager at any hierarchical level. The models indicate control actions and solve optimization problems at the company $\rightarrow$ project $\rightarrow$ subnetwork levels.
2. Both groups of models outlined in Chapters 14-15 (a hierarchical on-line

control model for PERT-COST projects) and Chapter 16 (hierarchical decision-making models for PERT-COST and GRU type projects) comprise planning and control stages. In both groups of models planning procedures are carried out from top to bottom, while control actions develop in the opposite direction.

3. The main difference between the two groups of hierarchical models under comparison is as follows:
   - model (15.1.1-15.1.9) is an *on-line control model* which periodically determines routine control points and trajectory curves while models outlined in §§16.1-16.3 and 16.6 are usually not based on classic control techniques and resemble theoretical grounds outlined in §3.5. Those models refer to the *advisory* class on behalf of the project manager.

Chapter 17. Estimating the Quality of Stochastic Network Projects

§17.1 The projects' quality problem

17.1.1 Introduction

In recent years we have developed and outlined in [7,54] a novel concept of the organization system's utility. Nowadays, *the existing utility theory centers on analysing the competitive quality of organization systems' outcome products rather than dealing with the quality of the systems' functioning, i.e., with organization systems in their entirety*. This may result in heavy financial losses, e.g., when excellent project objectives are achieved by a badly organized project's realization [7].

Thus, a conclusion can be drawn that the existing utility theory cannot be used as the system's quality techniques. We have undertaken research in the area of *estimating the quality of the system itself*, e.g., the system's public utility. We will consider a complicated organization system which functions under random disturbances. Such a system usually comprises a variety of qualitative and quantitative attributes, characteristics and parameters, which enable the system's functioning. The problem arises to determine a generalized (usually quantitative) value which covers all essential system's parameters and can be regarded to as a system's qualitative estimate. We will henceforth call such a generalized value *the system's utility*.

Thus, the system's utility signifies the quality of the system's functioning.

We will call the system's utility a weighted linear function of the system's parameters with constant coefficients. The parameters are divided into:

- independent parameters, where for each parameter its value may be preset and may vary independently on other parameters' values, and
- dependent parameters whose values may not depend uniquely on the values of independent parameters. However, when optimized (for the same values of independent parameters), they are solely dependent on those values.

Both independent and dependent parameters together with the coefficients of the utility function, are externally pregiven.

If an organization system functions under random disturbances and comprises n_1 independent basic parameters $R_i^{(ind)}$, $1 \le i \le n_1$, and n_2 dependent basic parameters $R_j^{(dep)}$, $1 \le j \le n_2$, the harmonization problem boils down to maximize the system's utility

$$U_S = \left(\sum_{i=1}^{n_1} \alpha_i R_i^{(ind)} + \sum_{j=1}^{n_2} \alpha_j R_j^{(dep)} \right) \qquad (17.1.1)$$

subject to certain restrictions. We suggest determining the optimal vector

$$\vec{R}_* = \left(R_{1*}^{(ind)}, R_{2*}^{(ind)}, \dots, R_{n_1*}^{(ind)}, R_{1*}^{(dep)}, R_{2*}^{(dep)}, \dots, R_{n_2*}^{(dep)} \right) \qquad (17.1.2)$$

which delivers maximization to the system's utility U_S, by means of the following sequential stages:

<u>Stage I</u> - implement a look-over algorithm to examine all feasible combinations of independent basic values $\left\{R_i^{(ind)}\right\}$;

<u>Stage II</u> - determine *optimal* values $\left\{R_{j\ opt}^{(dep)}\right\}$ for all dependent parameters by means of values $\left\{R_i^{(ind)}\right\}$ obtained at the previous stage; for each j-th dependent parameter an individual optimization model (called henceforth the partial harmonization model PHM_j), is used. The latter enables the optimality of each value $R_{j\ opt}^{(dep)}$ which *solely* depends on the combination $\left\{R_i^{(ind)}\right\}$;

<u>Stage III</u> - calculate the system's utility U_S via (17.1.1) for the combination

$$\left\{R_i^{(ind)}\right\},\ \left\{R_{j\ opt}^{(dep)}\right\} \tag{17.1.3}$$

obtained at *Stages I* and *II*;

<u>Stage IV</u> - calculate the optimal system's utility by determining the optimal combination (17.1.2) for all independent and dependent parameters which delivers the maximum to U_s.

If, due to the high number of possible combinations $\left\{R_i^{(ind)}\right\}$, implementing Stage I requires a lot of computational time, we suggest using a simplified heuristic search procedure, e.g., a cyclic coordinate search algorithm (CCSA), which has been outlined in greater detail in Chapters 11-13.

Thus, we consider an approximate harmonization's problem solution as follows. At the first stage a relatively simple search algorithm in the area of independent parameters, e.g., the cyclic coordinate descent method [114], is implemented. At the second stage, in order to evaluate the optimal value of each dependent parameter, an optimization problem PHM_j, $1 \le j \le n_2$, has to be solved. Thus, the idea is to obtain independent parameters' values at the first stage and to use them as input values of all partial harmonization models at the second stage.

PHM is usually a stochastic optimization model which is solved on the basis of simulation modeling. However, in certain cases, e.g., reliability and safety engineering problems, various *PHM* require more complicated formulations. In such cases we suggest to use additional heuristic models in order to implement realistic quantitative links between the system's attributes. For various dependent parameters the *PHM* may be formulated and solved by means of expert information [7,54].

The techniques of estimating the stochastic network project's utility by means of harmonization are outlined below.

We will use a multi-stage solution of harmonization problems [7,54]. At the first stage a look-over algorithm to examine all feasible combinations of independent basic values, is implemented. The independent parameters' values obtained at that stage are used as input values at the second stage where *for each dependent parameter* a local subsidiary optimization problem is solved in order to raise the system's utility as much as possible. Solving such a problem enables the *solely dependence* of the optimized value on any combination of independent input parameters. At the next stage the system's utility value is calculated by means of basic parameters' values obtained at the previous stages, with subsequent search for the extremum in order to determine the optimal combination of all basic parameters' values delivering the maximum to the system's utility.

§17.2 Harmonization model for PERT-COST projects

17.2.1 The problem's description

For stochastic PERT-COST network projects three parameters are implemented in the model:

- the budget C assigned to the project which has to be redistributed among the project's activities;
- the due date D for the project to be accomplished;
- the project's reliability R, i.e., the probability of meeting its due date on time subject to the pre-assigned budget C.

The harmonization model's solution is achieved by means of implementing a two-level heuristic algorithm. At the upper level a cyclic coordinate search algorithm (CCSA) to determine the quasi-optimal couple (budget – due date) is suggested. At the bottom level a high-speed heuristic procedure serving as a partial harmonization sub-model (PHM), is implemented: on the basis of input values (the assigned budget and the set due date) to maximize the probability of meeting the deadline on time by undertaking optimal budget reallocation among the project's activities.

We will calculate the project's utility by

$$U = \alpha_C \cdot [C_0 - C] + \alpha_D \cdot [D_0 - D] + \alpha_R \cdot [R - R_0], \qquad (17.2.1)$$

where C_0, D_0 and R_0 are the least permissible budget, due date and reliability values which can be implemented in a PERT-COST project, while values C, D and R are the corresponding current values for a project under consideration. Linear coefficients α_C, α_D and α_R define additional partial utilities which the project obtains by refining its corresponding parameter by a unit's value. Note that parameters C and D are in-dependent parameters since they can be preset beforehand independently on each other, while parameter R is practically defined by values D and C and, thus, is a dependent parameter. For the case $C=C_0$, $D=D_0$ and $R=R_0$, the project's utility is called the *basic utility* and is usually pregiven beforehand. Note that quantitative relations between parameters C, D and R are complicated, since setting a couple of values C and D results in a variety of possible values R depending on the budget reassignment among the project's activities. Thus, an additional optimization problem to maximize the reliability value on the basis of preset values C and D has to be imbedded in the project's harmonization model.

Besides those worst permissible pregiven values C_0, D_0 and R_0, one can define the best pregiven possible correspondent values - the minimal budget C_{00} to be assigned to the project, the earliest due date D_{00} (there is no need in accomplishing the project before D_{00}), and the maximal reliability value R_{00} (usually $R_{00}=1$). It can be well-recognized that any project values C, D and R satisfy

$$\begin{cases} C_{00} \le C \le C_0 \\ D_{00} \le D \le D_0 \\ R_0 \le R \le R_{00} \,. \end{cases} \tag{17.2.2}$$

According to Chapters 2 and 14, all random activity durations are assumed to have a beta-distribution, with the p.d.f. (14.1.1):

$$p_{ij}(t) = \frac{12}{\left(b_{ij}-a_{ij}\right)^4}\left(t-a_{ij}\right)\left(b_{ij}-t\right)^2,$$

where $b_{ij}=\dfrac{B_{ij}}{c_{ij}}$ and $a_{ij}=\dfrac{A_{ij}}{c_{ij}}$, A_{ij} and B_{ij} being pregiven constants for each activity (i,j) entering the PERT-COST network model.

17.2.2 Harmonization model

The harmonization model is as follows: determine optimal non-contradictive project parameters $C^{(opt)}$, $D^{(opt)}$ and $R^{(opt)}$ resulting in the maximal project's utility

$$\underset{\{C,D,R\}}{Max}\; U(G) = \underset{\{C,D,R\}}{Max}\left\{U_0 + \alpha_C\left(C_0 - C\right) + \alpha_D\left(D_0 - D\right) + \alpha_R\left(R - R_0\right)\right\} \tag{17.2.3}$$

subject to

$$C_{00} \le C^{(opt)} \le C_0, \tag{17.2.4}$$

$$D_{00} \le D^{(opt)} \le D_0, \tag{17.2.5}$$

$$R_{00} \ge R^{(opt)} \ge R_0 \,. \tag{17.2.6}$$

Note that since the basic utility U_0 is a constant value which remains unchanged, it may be canceled and, thus, the model satisfies

$$\underset{\{C,D,R\}}{Max}\; U(G) = \underset{\{C,D,R\}}{Max}\left\{\alpha_C\left(C_0 - C\right) + \alpha_D\left(D_0 - D\right) + \alpha_R\left(R - R_0\right)\right\} \tag{17.2.7}$$

subject to (17.2.4-17.2.6). Values C, D and R are called non-contradictive if budget C can be reassigned among the project activities to satisfy

$$Pr\left\{T(G)_{c_{ij}} \le D\right\} = R \tag{17.2.8}$$

subject to

$$\sum_{(i,j)} c_{ij} = C \,. \tag{17.2.9}$$

Solving problem (17.2.3-17.2.7) can be carried out by solving two sequential problems: to determine an optimal budget value C and an optimal due date D (Problem 1) and to carry out the PHM (Problem 2) [7,54].

Problem 1 centers on determining an optimal couple $\left(C^{(opt)}, D^{(opt)}\right)$ by means of a look-over algorithm that checks the feasibility of each possible combination (C,D). If the number of combinations is high enough and taking into account that:

- each combination requires a PHM solution, and
- Problem 1 is a NP-complete one,

- solving both problems on a look-over basis requires a lot of computational time [54]. To avoid this obstacle, we use a two-level high-speed approximate heuristic algorithm. At the upper level a heuristic simplified search procedure, e.g., a cyclic coordinate sub-algorithm [114], has to be carried out in the two-dimensional space in order to determine an optimal combination (C,D). At the bottom level, a heuristic high-speed procedure to optimize the partial harmonization model $PHM\big/_{C,D}$ with independent input values C and D, has to be implemented. Thus, we substitute objective (17.2.7) by

$$\underset{C,D}{Max}\left\{CCSA\left\{C,D\right\} \cup PHM\big/_{C,D} \Rightarrow U\left(C,D,R\right)\right\}, \qquad (17.2.10)$$

where $\cup$ stands for a unification sign.

17.2.3 *Partial harmonization model PHM(C,D)=R*

As outlined above, parameters C and D are input values of the model as well as values $c_{ij\min}$, $c_{ij\max}$, A_{ij} and B_{ij}, $(i,j) \subset G(N,A)$. The problem is as follows: determine optimal reassigned budget values c_{ij} for each activity $(i,j) \subset G(N,A)$, to maximize the project's conditional reliability, i.e.,

$$\underset{\{c_{ij}\}, \sum\limits_{(i,j)} c_{ij}=C}{Max}\left[\ Pr\left\{T\left(G\right)_{c_{ij}} \le D\right\}\right] \qquad (17.2.11)$$

subject to

$$c_{ij\min} \le c_{ij} \le c_{ij\max}, \qquad (17.2.12)$$

$$\sum_{(i,j)\subset G(N,A)} c_{ij} = C. \qquad (17.2.13)$$

The procedure of optimizing problem (17.2.11-17.2.13) has been outlined in §§14.4-14.5. Various variants of the algorithm are presented in [7,54,61,92].

§17.3 Estimating the utility of a portfolio of PERT-COST projects

17.3.1 *Introduction*

This presentation is actually a continuation of §17.2 and considers a complicated hierarchical system comprising a variety of projects of different significance. Such projects usually emerge in constructing new industrial and populated areas, where the significance of certain local projects entering the system may undergo changes within the projects' realization. The latter often happens in the course of changing management policy as well as the economic situation.

Another harmonization model covers a simplified although important case when all projects happen to be of equal significance and do not undergo drastic changes in the course of their implementation. The harmonization model becomes simpler in usage, and is based on determining optimal utility values via minimax principles [7,54].

Each project entering the portfolio comprises three essential, basic parameters which define the project's utility:

- budget C_k assigned to each project $G_k(N,A)$, $1 \le k \le n$;
- the appropriate due date D_k;
- reliability parameter $R_k(C_k, D_k)$,

$$R_k(C_k, D_k) = \underset{\{c(i,j)_k\}}{Max} \, Pr\left\{ T_k \le D_k \,\middle|\, \sum_{(i,j)_k} c(i,j)_k = C_k \right\}, \tag{17.3.1}$$

where T_k signifies the moment project $G_k(N,A)$ is completed (a random value), on condition that budget C_k is assigned to $G_k(N,A)$ and optimally reallocated between activities $(i,j)_k$. It goes without saying that relation $C_k > \sum_{(i,j)_k} c(i,j)_k$ holds, otherwise project $G_k(N,A)$ cannot be carried out.

For each k-th project its utility U_k is calculated as follows [7,54]:

$$U_k = \alpha_{C_k}(C_{0k} - C_k) + \alpha_{D_k}(D_{0k} - D_k) + \alpha_{R_k}[R_k(C_k, D_k) - R_{0k}] , \tag{17.3.2}$$

where C_{0k}, D_{0k} and R_{0k} are the least permissible basic values that can be accepted in the course of the project's realization, $1 \le k \le n$, while α_{C_k}, α_{D_k} and α_{R_k} stand for local (partial) utilities per each parametrical unit.

17.3.2 <u>Assumptions</u>

For the models under consideration we will henceforth accept the same reasonable assumption [7,54,64,92] which has been accepted in §14.2, namely (see *14.2.4*):

reliability parameter $R_k(C_k, D_k)$ depends on budget value C_k linearly, i.e., within each project $G_k(N,A)$ with fixed due date D_k relation

$$\frac{R_k(C'_k, D_k) - R_k(C_{00k}, D_k)}{C'_k - C_{00k}} = \frac{R_k(C''_k, D_k) - R_k(C_{00k}, D_k)}{C''_k - C_{00k}} = \rho_{kD_k} \tag{17.3.3}$$

holds, where $\sum_{(i,j)_k} c(i,j)_{k\,\max} \ge C_{0k} > C'_k > C''_k > C_{00k} \ge \sum_{(i,j)_k} c(i,j)_{k\,\min}$ and ρ_{kD_k} depends only on the project's index k and the due date D_k. Here C_{00k} presents the minimal possible budget value assigned to the k-th project. Note that ρ_{kD_k} may alter only when budget reassignment is undertaken *at a fixed moment*, otherwise the structure of the project $G_k(N,A)$ may undergo drastic changes. It goes without saying that values ρ_{kD_k} may differ from project to project.

17.3.3 Harmonization model's application

Budget reassignment among the projects has to be carried out:

- at the beginning of the planning horizon, i.e., at $t=0$;
- at a certain moment t when at least for one k-th routine project values D_k and η_k undergo changes;
- at a certain moment t when one of the projects is accomplished.

17.3.4 Case of different and equal projects' importance

For a project management system with projects of different importance we have solved [7,54,64] the harmonization problem with objective

$$J_1 = \sum_{k=1}^{n} (\eta_k \cdot U_k). \tag{17.3.4}$$

Maximizing objective (17.3.4) means that the project management takes all possible measures first to support projects with higher priorities. Only afterwards it takes care of other, less important, projects.

In case of projects with equal priorities we will implement another objective satisfying [7,54,92]

$$J_2 = \underset{\{C_k, D_k\}}{Max} \; \underset{k}{Min} \; U_k. \tag{17.3.5}$$

Objective (17.3.5) means that for projects with equal significance the project management takes all measures to support the "weakest" projects on the account of the "stronger" and the "faster" ones. That means, in turn, implementing a policy resulting in control actions aimed on projects' leveling, in order to smooth the differing projects' utilities.

§17.4 Harmonization model for projects with different priorities

The problem is as follows: for each k-th project determine optimal due date D_k and budget value C_k, $1 \le k \le n$, as well as optimal reassignment values for each activity $c(i,j)_k$, $(i,j)_k \subset G_k(N,A)$, to maximize the objective:

$$\begin{aligned} Max\, J_1 &= \underset{\{C_k, D_k\}\{c(i,j)_k\}}{Max} \sum_{k=1}^{n} (\eta_k \cdot U_k) = \\ &= \underset{\{C_k, D_k\}\{c(i,j)_k\}}{Max} \left\{ \sum_{k=1}^{n} \left[\eta_k \cdot \left[\alpha_{C_k} (C_{0k} - C_k) + \alpha_{D_k} (D_{0k} - D_k) + \alpha_{R_k} (R_k - R_{0k}) \right] \right] \right\} \end{aligned} \tag{17.4.1}$$

subject to

$$\sum_{k=1}^{n} C_k = C, \tag{17.4.2}$$

$$\sum_{\{(i,j)_k\}} c(i,j)_k = C_k, \tag{17.4.3}$$

$$c(i,j)_{k\,min} \le c(i,j)_k \le c(i,j)_{k\,max}, \tag{17.4.4}$$

$$C_{00k} \le C_k \le C_{0k}, \tag{17.4.5}$$

$$D_{00k} \le D_k \le D_{0k}, \tag{17.4.6}$$

$$R_{0k} \le R_k \le R_{00k}, \tag{17.4.7}$$

$$(C_k - C_{00k})\rho_{kD_k} = R_k(C_k, D_k) - R_k(C_{00k}, D_k), \tag{17.4.8}$$

$$C_{00k} \ge \sum_{\{(i,j)_k\}} c(i,j)_{k\ min}, \tag{17.4.9}$$

$$C_{0k} \le \sum_{\{(i,j)_k\}} c(i,j)_{k\ max}, \quad 1 \le k \le n. \tag{17.4.10}$$

Problem (17.4.1-17.4.10) has been solved in [7,54,64,92]. Extensive experimentation has verified the model's effectiveness [7,64].

§17.5 Harmonization model for projects of equal significance

The problem of maximizing the system's utility can be formalized as follows: for each k-th project determine budget value C_k and due date D_k, $1 \le k \le n$, *to maximize utility of the project with the least utility value*, namely,

$$J_2 = \underset{\{C_k, D_k\}}{Max}\ \underset{k}{Min}\ U_k = \underset{\{C_k, D_k\}}{Max}\ \underset{k}{Min} \left[\alpha_{C_k}(C_{0k} - C_k) + \alpha_{D_k}(D_{0k} - D_k) + \alpha_{R_k}(R_k - R_{0k}) \right] \tag{17.5.1}$$

subject to (17.4.2-17.4.10).

Since the maximin approach has presented itself in a very good light both in production planning and control [54,63] and in project management [54,92,118,151], objective (17.5.1) has been suggested as a priority technique for improving utility values in complicated project management systems operating under random disturbances.

Problem (17.4.2-17.4.10, 17.5.1) has been solved in [7,54,64,92].

Note that the general idea of solving both problems outlined in §§17.4-17.5 has been outlined in §§14.2-14.3 when reassigning the company's budget among the projects. Thus, a certain linkage between the concepts of harmonization and optimal resource reallocation can be drawn.

All the outlined above concepts can be used for the case of several alternative stochastic network projects of CAAN type, outlined in §§16.1-16.6. For the case of PERT-COST projects the combination of models outlined in §§14.2-14.3 and §§17.1-17.5 provides good results (see [7]), while for the case of renewable resources, e.g., resource capacities in the form of GRU, we recommend the combination of models outlined in §§14.2-14.3, §16.6 and §§17.3-17.4. The harmonization models for a portfolio of alternative stochastic projects may vary, but all the principal concepts remain the same.

§17.6 Numerical example: Estimating the quality of a single PERT-COST project

Consider a PERT-COST type project with random activity durations and p.d.f. satisfying (14.1.1).

The basic project's parameters are as follows: project's budget C, due date D and reliability R. Partial utility coefficients are $\alpha_C = 1.0$, $\alpha_D = 0.5$ and $\alpha_R = 1.0$, while the initial search steps (first iteration) for $CCSA$ are $\Delta C = 4$ and $\Delta D = 2$. The number M of simulation runs for the PHM is taken $M = 2{,}000$. Other project's parameters are as follows: $R_0 = 0.7$, $R_{00} = 0.95$, $C_0 = 250$, $C_{00} = 230$, $D_0 = 95$, $D_{00} = 85$, $\delta C = 10$, $\delta D = 2.0$, $\delta R = 0.1$ and $\varepsilon = 0.001$.

The second iteration for the $CCSA$ is carried out with $\Delta C = 2.0$ and $\Delta D = 1.0$, while all further iterations, $v \geq 2$, are carried out with $\Delta C = 1.0$ and $\Delta D = 1.0$.

The performance of the harmonization model's algorithm is illustrated on Tab. 17.1.

Table 17.1. Performance illustration of the harmonization algorithm (for a beta-distribution p.d.f.)

№ of search steps	C	D	R	*№ v of iteration*	*Feasibility*	*Utility $U(C,D,R)$*	*Value $U^{(v)}$ after the v-th iteration*
0	250	95	1.000	1	Feasible	2.50	2.50
1	246	95	0.996	1	Feasible	2.90	2.90
2	242	95	0.922	1	Feasible	3.02	3.02
3	238	95	0.793	1	Feasible	2.13	3.02
4	242	93	0.861	1	Feasible	3.41	3.41
5	242	91	0.723	1	Feasible	3.03	3.41
6	244	93	0.895	2	Feasible	3.55	3.55
7	246	93	0.912	2	Feasible	3.52	3.55
8	240	93	0.814	2	Feasible	3.14	3.55
9	244	94	0.936	2	Feasible	3.46	3.55
10	244	92	0.835	2	Feasible	3.45	3.55
11	**245**	**93**	**0.914**	**3**	**Optimal**	**3.64**	**3.64**
12	243	93	0.875	3	Feasible	3.45	3.64
13	245	94	0.951	4	Feasible	3.51	3.64
14	245	92	0.855	4	Feasible	3.55	3.64
Since values $U^{(3)}$ and $U^{(4)}$ coincide, the algorithm terminates after the fourth iteration							

Thus, the optimal utility value equals 3.64.

The following conclusions can be drawn from the Table:

The cyclic coordinate search algorithm for determining the optimal utility of a medium-size project requires only four iterations to carry out the optimization process. The increase of the project's utility parameter after completing the fourth iteration (14 search points), as compared with the initial search point, shows utility improvement of approximately 45%. Thus, it can be well-recognized that the two-level heuristic algorithm to optimize the project's harmonization model performs well.

Numerical examples for several stochastic network projects are outlined in [7,54].

§17.7 Application areas

Harmonization models can be applied directly to all kinds of PERT-COST network projects with uncertainties associated with activities' durations but without either technological risks or uncertainties on the stage of marketing the project's products. Such projects usually refer to the public service area, like constructing new hospitals, schools, stadiums, theatres, bridges and tunnels, new urban areas, factories, etc. In our opinion, those projects represent an overwhelming majority of existing projects and, thus, require good quality monitoring. For such projects we suggest to use the newly developed harmonization techniques both for estimating the project's utility and for introducing regulating control actions at inspection points to enhance the progress of the project in the desired direction. Thus, harmonization modeling enables certain on-line control procedures for projects under random disturbances.

Besides the example outlined above, the developed harmonization principle covers a broad spectrum of other hierarchical organization systems, especially of man-machine type. Several important examples of potential areas of implementation are presented here [54].

I. Consider a complicated multilevel technical system to be designed, e.g., a new commercial aircraft. Here the number of basic parameters which actually define the aircraft's utility, exceeds three by far; the basic parameters are as follows:
   - the budget assigned for constructing the new aircraft (an independent parameter);
   - the number of passengers to be taken on board (an independent parameter);
   - the flight distance (a partially dependent parameter);
   - the average cruise speed (a dependent parameter);
   - the reliability value, i.e., the probability of the aircraft within a specified exploitation period not to develop any critical failure which may result in air fleet accidents, sometimes of catastrophic nature (a dependent parameter);
   - an environmental failure parameter, e.g., the level of noise (a dependent parameter);
   - various technical design parameters, e.g. the aircraft's size, weight or even certain aesthetic features which nowadays may influence the aircraft's priority level (usually dependent parameters), etc.

   It goes without saying that increasing the number of basic parameters results in a dramatic increase of the level of complexity of the regarded harmonization model.

II. In agriculture, e.g. in cotton harvesting, a multilevel decision-making control system is especially useful for cotton-growing areas with restricted resources. Since all cotton harvesters are equipped with trailers, one of the independent basic parameters of the model should be the amount of trailers available to each harvester. Other basic parameters may be singled out as follows:
   - the volume of the trailer (an independent parameter);
   - the number of trailers used to form the so-called "cotton trains" delivering raw cotton to the cleaning factory (an independent parameter);
   - the number of harvesters (an independent parameter);
   - the weather forecast (a random disturbance parameter);
   - the type and agricultural quality of soil (an independent parameter);
   - the harvesting period for cotton (an independent parameter);
   - the budget to be assigned for cotton harvesting in a cotton-growing district (a dependent parameter);
   - harvesting expenses per square unit of plantation (a dependent parameter);
   - the weight of cleaned cotton obtained from the above (a dependent parameter), etc.

   The cotton harvesting organization system is, thus, an extremely complicated one. However, using harmonization models as suggested in this paper may result in significantly increasing the system's utility.

III. A promising application area of the discussed theory and methodology lies in developing new approaches for designing hospitals (or providing capital investments for expanding existing medical health facilities) in rural areas. The basic parameters to determine hospital's utility may be listed as follows:
   - the main costs of designing and building a new hospital (an independent parameter);
   - the population to be serviced (an independent parameter);
   - accessibility and the geographical distance from the hospital to most remote settlements (an independent parameter);
   - the number of beds (a dependent parameter);
   - various quality and quantity parameters of medical care (partially dependent parameters);
   - the average number of days for a patient to stay in the hospital, i.e., the patient's "turnover" value (a dependent parameter), etc.

   Thus, a hospital is a good field for implementing harmonization trade-off problems. Note that within the last three decades numerous decision-making models on health care and health service have been described in various publications. However, attempts to define the hospital's utility in its entirety have not been undertaken as yet.

Thus, the problem under consideration can be regarded as a fruitful one in stochastic Project Management.

§17.8 Conclusions

The following conclusions can be drawn from the Chapter:

1. We will implement the utility concept as a generalized system's quality estimate which takes into account several essential parameters. The latter usually define the quality of the system as a whole. We have developed a generalized harmonization problem in order to maximize the system's utility. The corresponding model is optimized by means of a two-level heuristic algorithm. At the upper level (the level of independent parameters) a relatively simple search procedure, e.g., the cyclic coordinate algorithm, has to be implemented. At the lower level partial harmonization problems to optimize the dependent parameters, have to be used.
2. For stochastic PERT-COST network projects three parameters are implemented in the model: the budget assigned to the project, the due date and the project's reliability to meet the due date on time. The harmonization model's solution is achieved by means of implementing a two-level heuristic algorithm. At the upper level a cyclic coordinate search algorithm to determine the quasi-optimal couple (budget – due date) is suggested. At the bottom level a high-speed heuristic procedure serving as a partial harmonization sub-model, is implemented: on the basis of input values (the assigned budget and the set due date) to maximize the probability of meeting the deadline on time by undertaking optimal budget reallocation among the project's activities. For the case of several stochastic network projects the developed theory enables determining optimal parametrical values for all projects in order to achieve the maximal utility level for the unification of all projects. The developed algorithm to optimize the harmonization model for a hierarchical project management system in R&D design offices, presents a two-level heuristic procedure. At the upper level a cyclic coordinate search algorithm together with a subsidiary model to verify the feasibility is implemented. At the lower level certain linear programming techniques can be applied to obtain an approximate solution. The harmonization model can be used both for projects with different priorities and for projects of equal significance.
3. For projects with renewable resources the following three parameters have to be implemented in the model: the number of GRU, the due date and the project's reliability. Optimizing the GRU assignment is facilitated by means of techniques outlined in §§17.3-17.4 (for fixed-structure stochastic network projects) and in §16.6 (for alternative projects of CAAN type).
4. It can be well-recognized [7,54] that the outlined above techniques can be applied to estimate the quality of numerous projects in economics, safety engineering, construction, health care and life sciences, industrial engineering, etc.

Conclusions

1. This monograph is a result of all the best which has been developed by myself and my scientific school in Russia and Israel within the past five decades in the area of stochastic network modeling with emphasis on innovative projecting.
2. In the monograph, we have outlined a variety of novel hierarchical models to monitor stochastic network projects with restricted resources. The presented hierarchical models include indicating control actions and optimizations problems for all levels. The models can be implemented for a very broad spectrum of R&D projects, especially for innovative projects. The software to simulate the models can be implemented on a PC, mainly for projects with a medium amount of activities. The outlined hierarchical control models can be used for practically all activity-on-arc network projects with independent activities durations and cost-duration functions. The models can be used both for controlling single projects and several projects with restricted company budget.
3. A broad variety of simulation and optimization models to monitor hierarchical R&D projects can be subdivided into three main classes:
   - planning models;
   - on-line control models;
   - scheduling models.

   As to the structure of the stochastic project's network, the latter may refer to:
   - projects with fixed (deterministic) structure, and
   - projects with stochastic alternative structure.

   All the models presented in the book have been outlined according to this classification. Several important models (mostly scheduling ones) refer simultaneously to more than one class (usually planning and scheduling models).
4. On-line control models enable both determining inspection (control) points t_g to inspect the project and to determine its speed υ_{t_g} to proceed with until the next control point t_{g+1}. Such a model comprises a stochastic optimization problem with a non-linear chance constraint and a random number of optimized variables. The problem is too difficult to be solved in the general case. Thus, heuristic solutions have been developed [54,62,64,66,73,86-87].

   Three types of heuristic algorithms have been presented and described in the monograph:

   A. Using sequential statistical analysis to maximize the time span $\Delta t_g = t_{g+1} - t_g$ and, thus, minimizing the number of inspection points [64,66,68].

B. Using the concept of a risk averse decision-maker [63,68,72]. However, algorithms of both types A and B are unfit to solve cost-optimization problems.

C. Using the chance constrained principle [54,73,83-84], when the problem at a routine control point t_g is to determine the proper speed υ^{τ} and the next control point t_{g+1} in order to minimize total processing costs within the planning horizon, subject to a chance constraint. At each control point the decision-maker centers around the assumption that there is no more than just one additional control point before the due date.

5. Algorithms of type C have been primarily aimed at organization systems [54,73], but later on have been essentially refined for stochastic network projects, both for the case of a single project [83] and for several stochastic projects [84]. The refinement centers on the idea, that determining speed υ^{τ} is carried out by means of a long-term forecasting on the basis of the p-quantile estimation, while determining the next control point t_{g+1} is based on a short-term forecasting by means of substituting random values for their average estimates.
6. The backbone of the monograph is the research which has been undertaken in the area of alternative stochastic network models. In our opinion, this area suits mostly modern innovation projects. The high level of indeterminacy in conjunction with permanent changes in the network's structure makes alternative stochastic models the perfect technique to formalize decision-making "brain-storming". Two controlled models are outlined - the CAAN model [53-57,68,75,82] and the more generalized GAAN model [53,67].
7. The CAAN model is a fully divisible controlled alternative activity network for projects with both random and deterministic alternative outcomes in key nodes. At each routine decision-making node, the algorithm singles out (by means of lexicographical lookover) all the subnetworks (joint variants) corresponding to all possible outcomes from that node. Decision-making results in determining the optimal joint variant and in following the optimal direction up to the next decision-making node. The model can be applied to networks with non-intersecting fragments only.
8. For the case of time-cost optimization of a CAAN model the monograph presents an iterative approximate algorithm enabling a quasi-optimal solution with two parameters. Extensive experimentation has shown [9,75,82] that the approximate algorithm performs well and provides high accuracy solutions.
9. The GAAN model [9,53-54,67-68,157] is a non-divisible network which, being more complicated than its ancestor CAAN, is nevertheless more relevant to highly complicated R&D projects when decision-making has to be facilitated with incomplete or inadequate information on the alternatives. Such a model is particularly important for innovative R&D projects with multiple alternative technology choices, when several alternative ways exist

for reaching intermediate and ultimate goals. The model can be widely used in optoelectronics, aerospace and defense-related industries, in projecting new software (Information Technology Projects), in R&D projects both with multiple technologies and stochastic evolution of technology leading to obsolescence effects [9,67,134], as well as in many other projects of innovative nature.

10. The monograph includes several resource scheduling models for managing stochastic network projects. The first type of models is based on resource reallocation to determine optimal planned start moments in order to minimize total management expenses. This approach has been successfully implemented for the case of consecutive operations entering the project. The regarded model has been applied for a group of projects, and experimentation verifies that the model's algorithm performs well [53,74,151].
    Another resource supportability model has been outlined for monitoring several stochastic projects in the form of network graphs. The model covers a flexible project management system. It minimizes the average operational expenses subject to the chance constraints, for each project separately. The model can be used in project management as a decision support techniques for planning and monitoring several stochastic network projects. It has been successfully used for small and medium size projects of PERT type. The presented model is suitable for resource scheduling in stochastic network projects, when the processing of certain activities is based on delivering resources [79-80].
11. Several presented resource supportability models center on using two resource delivery schedules:
    - for extremely costly and rare resources; the corresponding resource delivery moments have to be predetermined and calculated beforehand, i.e., before the project actually starts;
    - for limited resources which are at the disposal of the project and have to be hired at a predetermined time.

    The objective of the resource supportability model boils down to minimizing the total average expenses of the resource consumption within the planning horizon. Optimization is facilitated by means of simulation, in combination with a cyclic coordinate descent method and a knapsack resource reallocation model. The algorithm performs well and enables the model's flexibility [53,81,151].
12. The monograph includes a resource constrained scheduling model for projects under random disturbances and with alternative structure based on a two-level decision-making [85]:
    - at alternative deterministic decision nodes; the purpose is singling out all alternative subnetworks in order to choose the one with the minimal average duration;

- at the project's essential moments when at least one activity is ready to be operated but the available amount of resources is limited. A competition among those activities is carried out to determine the subset of activities, which have to be operated first and can be supplied by available resources. Such a competition is carried out by a combination of a knapsack resource reallocation model and a subsidiary simulation algorithm.

Since an alternative stochastic network model is usually structured from subnetworks of GERT type, the regarded resource constrained project scheduling algorithm is based on multiple realization of a standardized resource constrained algorithm for GERT models. The algorithm can be "tuned" for any probability distribution of activity durations.

13. Various hierarchical control models for monitoring several stochastic network projects are presented. The included models cover the following cases:
    - several PERT-COST -type projects;
    - several CAAN type projects with budget resources;
    - several CAAN type projects with renewable resources.
14. A multilevel on-line control model for several PERT-COST projects to be realized simultaneously is outlined. On the project level each project is controlled separately to minimize the number of control points, subject to a chance constraint not to deviate from the planned trajectory within the planning horizon with pregiven probabilities. If at the control point it is anticipated that the project will not be on target subject to the chance constraint, then an emergency is called and the remaining projects' budget is reassigned at the second level among the remaining projects so that the faster ones may help the slower. Such a budget reassignment is also performed at the planning stage, i.e., before the beginning of the projects' realization. Two conflicting objectives are embedded in the model: one to minimize the number of control points and the other to maximize the probability for the slowest project to meet its due date on time.
15. A hierarchical decision-making model for controlling CAAN type projects with cost resources is presented. For multilevel control models of such type, the upper level (the company level) is usually faced with the problem of optimal budget reassignment among several network projects. On the medium level (project level) the management determines the optimal outcome direction at every deterministic alternative decision node (milestone) which is reached in the course of the project's realization. In order to determine both the optimal outcome and the corresponding subnetwork, one has to calculate average durations for all subnetworks subject to the limited budget. The subnetwork with the minimum average duration (after solving the budget reassignment optimization problem for all subnetworks to minimize the average duration) determines the optimal outcome direction at the decision node. Later on, the project is carried out on the lower level according to the chosen subnetwork, until the next decision node.

16. For the case of monitoring a multilevel system comprising several CAAN type projects with renewable resources of GRU type we have outlined several optimization models aimed at GRU reallocation among the projects. Both cases of projects of different and equal significance are examined.
17. In case of monitoring a complicated stochastic network project we suggest to honor the following concepts [68]:
    - Scheduling and control procedures must *not* be incorporated in one model.
    - A control model has to be based on probabilistic approaches and has to implement probabilistic terms. Such a model has to be used only at several control (inspection) points. We suggest applying the control model not to the initial network (which for some projects may comprise a large amount of activities), but to a modified one, with a medium amount of activities at the utmost. For such a modified model, an activity can be a subnetwork (a fragment) of the initial network.
    - Scheduling procedures are applied to the initial network and are carried out *between two adjacent routine control points*. They are usually based on heuristic procedures (sometimes very doubtful) and may result in biased estimates and errors. *But the latter are periodically corrected* by means of introducing proper control actions.
    - Thus, we recommend *developing the on-line control model as an additional tool, as a decision-making support model to assist the project manager carry out the project*. On the basis of such a model, the project manager may implement any action he finds reasonable, e.g., to enhance the progress of the project.
18. In the case of a large-scale project [68]:
    - First, modify the initial large-scale project to an enlarged aggregated network *of medium size* (comprising not more than 40÷50 activities).
    - Second, apply to that aggregated project on-line *control techniques* in order to determine the project's proper speeds and inspection points.
    - Third, *reaggregate* the enlarged project to its initial size.
    - Four, *reschedule* the activities between the adjacent inspection (control) points according to their average values, i.e., implement deterministic scheduling techniques for project's fragments between adjacent decision-making points. The latter can be utilized as corrective indications.
19. We have outlined the newly developed [7,54] techniques to estimate the quality of stochastic network projects. Estimating the project's quality is based on application of the harmonization theory [54] to a linear function of basic project's parameters.
20. The monograph presents a standard and very effective two-level heuristic algorithm to solve a variety of outlined optimization problems (see, e.g., Chapters 11-13). At the lower level the simulation model undertakes numerous simulation runs to manage the project's realization. At the upper level a

heuristic search subalgorithm carries out cyclic coordinate optimization to determine the optimal coordinate variables. Extensive experimentation [49-54,92,118,151] has shown the efficiency of the algorithm.

As a matter of fact, we have chosen the cyclic coordinate search method and have used it successfully over a lengthy period because of the deep affection we developed to it from the very beginning, in recognition of its major benefits which combine relative simplicity with outstanding performance. This does not ban of course from integrating in future additional non-linear search methods which might prove to be quite useful as well.

21. This monograph covers not only innovative R&D projects. Other network projects of innovative nature, e.g., long-term projects in the construction industry when creating and building unique installations, designing or developing new transcontinental pipe-lines, etc., can be planned and controlled by using the outlined theory. Note that the latter can be applied to any network project with a high level of uncertainty, namely:
   - projects with random activity durations, but without milestones (decision nodes) and feedback loops;
   - projects with both random activity durations and decision nodes (milestones of deterministic type);
   - projects with random activity durations, milestones of both random and deterministic types and feedback loops. The latter type is often used when designing and creating new machines and unique installations with no similar prototypes in the past. When using our theory we do not see any principal difference between designing a unique missile or projecting a unique software (Information Technology Projects).

   To sum up, there are various industries operating with high technology projects, when the nature of the problems is unknown at the start of the project and where alternative technical solutions can be found in key nodes.

REFERENCES

1. Anklesaria, K.R. and Drezner, Z. (1986), A multivariate approach to estimating the completion time for PERT networks, *J. Oper. Res. Soc.*, 37: 811-815.
2. Ari, E.A. and Axsater, S. (1988), Disaggregation under uncertainty in hierarchical planning, *Eur. Oper. Res.*, 35: 182-186.
3. Arisawa, S. and Elmaghraby, S.E. (1972), Optimal time-cost tradeoffs in GERT networks, *Mgmt. Sci.*, 18(11): 589-599.
4. Arsham, H. (1993), Managing project activity-duration uncertainties, *Omega*, 21(1): 111-122.
5. Artigues, C., Roubellat, F. and Billat, J. (1998), Characterization of a set of schedules in a resource-constrained multi-project scheduling problem with multiple nodes, *Int. J. Ind. Engn.*, 12(4): 48-70.
6. Bell, C.E. and Han, J. (1991), A new heuristic solution method in resource constrained project scheduling, *Naval Res. Logis.*, 38(3): 315-331.
7. Ben-Yair, A. (2004), Harmonization models in strategic management and safety engineering, Ph.D. Thesis, *Ben-Gurion University of the Negev*, Beer-Sheva, Israel.
8. Berny, J. (1989), A new distribution function for risk analysis, *J. Oper. Res. Soc.*, 40(12): 1121-1127.
9. Blokh, D. (1997), Developing alternative network models for controlling stochastic projects, Ph.D. Thesis, *Ben-Gurion University of the Negev*, Beer-Sheva, Israel.
10. Blokh, D. (1989), Assignment problem for three-index Jacobi matrices, *Automation and Remote Control*, 50(2), Part 2: 672-677.
11. Blokh, D. and Gutin, G. (1995), Maximizing traveling salesman problem for special matrices, *Disc. Appl. Math.*, 56: 83-86.
12. Blokh, D. and Gutin, G. (1996), An approximate algorithm for combinatorial optimization problems with two parameters, *Australasian J. of Combinatorics*, 14: 157-164.
13. Bobrowski, P.M. (1989), Project management control problems: An information systems focus, *Project Management Journal*, 20(2): 11-16.
14. Boctor, F. (1996), Resource constrained project scheduling by simulated annealing, *Int. J. Prod. Res.*, 34(8): 2333-2351.
15. Bourland, K.E. and Kaplan, C.E. (1994), Parallel-machine scheduling with fractional operation requirement, *IIE Transactions*, 26(5): 56-65.
16. Britney, B.R. (1976), Bayesian point estimation and the PERT scheduling of stochastic activities, *Mgmt. Sci.*, 22(9): 938-948.
17. Burkov, V.N. and Irikov, V.A. (1994), *Models and Methods for Controlling Organization Systems*, Moscow: Nauka (in Russian).
18. Burt, J.M., Gaver, D.P. and Perlas, M. (1970), Simple stochastic networks: Some problems and procedures, *Naval Res. Logis. Quarterly*, 17(4): 360-439.
19. Burt, J.M. and Garman, M.B. (1971), Monte Carlo techniques for stochastic PERT network analysis, *INFOR*, 9(3): 248-262.
20. Charnes, A., Cooper, W. and Thompson, G. (1964), Critical path analysis via chance constrained and stochastic programming, *Oper. Res.*, 12(3): 29-40.

21. Chen, H. and Mandelbaum, A. (1994), Hierarchical modeling of stochastic networks, in *Stochastic Modeling and Analysis of Manufacturing Systems*, Yao, D.D. (ed.), New-York: Springer-Verlag.
22. Clark, C.E. (1962), The PERT model for the distribution of an activity, *Oper. Res.*, 10: 405-406.
23. Clinger, C.T. (1964), A modification of Fulkerson's PERT algorithm, *Oper. Res.*, 12: 623-629.
24. Coffman, E.G. (1976), Computer and Job-Shop Scheduling Theory, New-York: Wiley.
25. Coon, H. (1965), Note on Donaldson's 'The estimation of the mean and variance of PERT activity time', *Oper. Res.*, 13(3): 386-397.
26. Cooper, D.F. and Chapman, C.B. (1987), *Risk Analysis for Large Projects*, New-York: Wiley.
27. Cramer, H. (1946), *Mathematical Methods of Statistics*, NJ, Princeton: Princeton University Press (reprinted in Princeton Landmarks in Mathematics and Physics series, 1999).
28. Crowston, W.B. (1970), Decision CPM: Network reduction and solution, *Oper. Res. Quarterly*, 21(4): 434-445.
29. Crowston, W.B. (1971), Models for project management, *Sloan Management Review*, 12(3): 25-42.
30. Crowston, W.B. and Thompson, G.L.(1967), Decision CPM: A method for simultaneous planning, scheduling and control of projects, *Oper. Res.*, 15: 407-426.
31. Dempster, M.A.H., Fisher, M.L., Jansen, L., Lageweg, B.J., Laenstra, J.K. and Rinnooy Kan, A.H.G. (1981), Analytical evaluation of hierarchical planning systems, *Oper. Res.*, 29(4): 707-716.
32. Dempster, M.A.H., Fisher, M.L., Jansen, L., Lageweg, B.J., Laenstra, J.K. and Rinnooy Kan, A.H.G. (1981), Analysis of heuristics for stochastic programming: Result for hierarchical scheduling problems, *Mathematics of Operations Research*, 8: 525-537.
33. Digman, I.A. and Green, G.I. (1981), A framework for evaluating network planning and control techniques, *Res. Mngm.*, 24(1): 10-17.
34. Dinic, E. (1990), The fastest algorithm for the PERT problem with AND- and OR-nodes, Procs. of a Conference on Integer Programming and Combinatorial Optimization, *University of Waterloo Press*, 185-187.
35. Dodin, B. (1985), Reducibility of stochastic networks, *Omega*, 13: 223-232.
36. Donaldson, W.A. (1965), The estimation of the mean and variance of PERT activity time, *Oper. Res.*, 13(3): 382-385.
37. Eisner, H. (1962), A generalized network approach to the planning and scheduling of research projects, *Oper. Res.*, 10(1): 115-125.
38. Elmaghraby, S.E. (1964), An algebra for the analysis of generalized activity networks, *Mgmt. Sci.*, 10: 494-514.
39. Elmaghraby, S.E. (1967), The expected duration of PERT type network, *Oper. Res.*, 13: 299-306.
40. Elmaghraby, S.E. (1977), *Activity Networks: Project Planning and Control by Network Models*, New-York: Wiley.

41. Evans, J. and Minieka, E. (1992), *Optimization Algorithms for Networks and Graphs*, New-York: Dekker.
42. Farnum, N.R. and Stanton, L.W. (1987), Some results concerning the estimation of beta distribution parameters in PERT, *J. Oper. Res. Soc.*, 38: 287-290.
43. Fishman, G. (1997), *Monte Carlo Concept, Algorithms and Applications*, Springer Series in Operation Research, New-York.
44. Ford, L.R.Jr. and Fulkerson, D.R. (1962), *Flows in Networks*, Princeton: Princeton University Press.
45. Fulkerson, D.R. (1962), Expected critical path lengths in PERT networks, *Oper. Res.*, 10: 808-817.
46. Gallagher, C. (1987), A note on PERT assumptions, *Mgmt. Sci.*, 33: 1360-1362.
47. Garey, U.R. and Johnson, D.C. (1979), *Computer and Intractability: A Guide to the Theory of NP-Completeness*, San-Francisco: Freeman and Co.
48. Golenko (Ginzburg), D.I., Levin, N.A., Mihelson, V.S. and Naidov-Jelezov, Ch.G. (1965), *Automated Planning and Control for Development Plants*, Riga: Zvaigzne (in Russian).
49. Golenko (Ginzburg), D.I. (1972), *Statistische Methoden der Netzplantecknik*, Leipzig: BSB B.G. Teubner Verlagsgesellschaft (in German, translated from Russian: *Statistical Methods in Network Planning and Control*, Moscow: Nauka, 1968).
50. Golenko (Ginzburg), D.I. (1973), *Statistical Models in Production Control*, Moscow: Statistika (in Russian).
51. Golenko (Ginzburg), D.I. et al (1976), *Statistical Modeling in R&D Projecting*, Leningrad: Leningrad University Press (in Russian).
52. Golenko (Ginzburg), D.I., Lauenroth, H.G., Schultz, H. and Schultze, G. (1977), *Kybernetik zur Steuerung Ökonomischer Prozesse: Grundlage und Anwendungen*, Berlin: Akademie-Verlag (in German).
53. Golenko-Ginzburg, D. (2010), *Stochastic Network Models in R&D Projecting*, Voronezh: Nauchnaya Kniga (in Russian).
54. Golenko-Ginzburg, D. (2011), *Planning and Controlling Multilevel Man-Machine Organization Systems under Random Disturbances*, Voronezh: Nauchnaya Kniga.
55. Golenko (Ginzburg), D.I. (1983), Planning and control by means of alternative stochastic network models (Part I), *Annals of New-York Academy of Sciences*, 410, 249-254.
56. Golenko (Ginzburg), D.I. (1983), Planning and control by means of alternative stochastic network models (Part II), *Annals of New-York Academy of Sciences*, 410, 255-261.
57. Golenko-Ginzburg, D. (1988), Controlled alternative activity networks in project management, *Eur. J. Oper. Res.*, 7: 336-346.
58. Golenko-Ginzburg, D. (1988), On the distribution of activity time in PERT, *J. Oper. Res. Soc.*, 39(8): 767-771.
59. Golenko-Ginzburg, D. (1989), PERT assumptions revisited, *Omega*, 17(4): 393-396.
60. Golenko-Ginzburg, D. (1989), A new approach to the activity-time distribution in PERT, *J. Oper. Res. Soc.*, 40(4): 389-393.

61. Golenko-Ginzburg, D. (1990), A two-level production control model with target amount rescheduling, *J. Oper. Res. Soc.*, 41(11): 1021-1028.
62. Golenko-Ginzburg, D. (1993), A two-level decision-making model for controlling stochastic projects, *Int. J. Prod. Econ.*, 32: 117-127.
63. Golenko-Ginzburg, D. and Sinuany-Stern, Z. (1993), Hierarchical control of semiautomated production systems, *Prod. Plan. Cont.*, 4(4): 361-370.
64. Golenko-Ginzburg, D. and Gonik, A. (1996), On-line control model for cost-simulation projects, *J. Oper. Res. Soc.*, 47: 266-283.
65. Golenko-Ginzburg, D. and Gonik, A. (1996), Hierarchical decision-making model for planning and controlling stochastic projects, *Int. J. Prod. Econ.*, 46-47: 39-54.
66. Golenko-Ginzburg, D. and Gonik, A. (1997), On-line control model for network construction projects, *J. Oper. Res. Soc.*, 48: 175-183.
67. Golenko-Ginzburg, D. and Blokh, D. (1997), A generalized activity network model, *J. Oper. Res. Soc.*, 48: 391-400.
68. Golenko-Ginzburg, D. and Gonik, A. (1997), Project planning and control by stochastic network models, in *Managing and Modeling Complex Projects*, Williams, T.M. (ed.), NATO ASI Series, The Netherlands: Kluwer Academic Publishers.
69. Golenko-Ginzburg, D. and Gonik, A. (1997), Job-shop resource scheduling via simulating random operations, *Math. Comp. Sim.,* 44: 427-440.
70. Golenko-Ginzburg, D. and Gonik, A. (1997), Resource constrained project scheduling for non-consumable limited resources, *Int. J. Prod. Econ.*, 48: 29-37.
71. Golenko-Ginzburg, D. and Gonik, A. (1998), A heuristic for network project scheduling with random activity durations depending on the resource reallocation, *Int. J. Prod. Econ.*, 55: 149-162.
72. Golenko-Ginzburg, D. and Gonik, A. (1998), High performance heuristic algorithm for controlling stochastic network projects, *Int. J. Prod. Econ.*, 54: 235-245.
73. Golenko-Ginzburg, D. et al (1999), Developing cost-optimization production control model via simulation, *Math. Comp. Sim.,* 49: 335-351.
74. Golenko-Ginzburg, D. et al (1999), Managing resource reallocation among several projects under random disturbances, *Communications in DQM*, 2(1): 137-149.
75. Golenko-Ginzburg, D., Blokh, D. and Gutin, G. (2000), A two-parametric approximate method to optimize alternative activity network models; Part I: The general approach and the algorithm, *Communications in DQM*, 3(1): 18-24.
76. Golenko-Ginzburg, D. et al (2000), Analysis of the stable laws of distribution of the duration of operations in the stochastic network projects, *Automation and Remote Control*, 61(12): 2068-2080.
77. Golenko-Ginzburg, D., Gonik, A. and Sitniakovski, Sh. (2000), Two-level COST-optimization production control model under random disturbances, *Math. Comp. Sim.,* 52: 381-398.
78. Golenko-Ginzburg, D. et al (2000), Resource supportability simulation model for

a man-machine production system, *Math. Comp. Sim.,* 53: 105-112.
79. Golenko-Ginzburg, D. et al (2000), Resource constrained project scheduling for several stochastic network projects, *Communications in DQM*, 3(1): 63-73.
80. Golenko-Ginzburg, D. et al (2000), Resource supportability model for stochastic network projects under a chance constraint, *Communications in DQM*, 3(1): 89-102.
81. Golenko-Ginzburg, D. et al (2001), Algorithms of optimal supply of resources to a group of projects (stochastic networks), *Automation and Remote Control*, 62(8): 1366-1375.
82. Golenko-Ginzburg, D. and Blokh, D. (2001), A two-parametric approximate method to optimize alternative activity network models; Part II: Numerical results, *Communications in DQM*, 4(1): 6-15.
83. Golenko-Ginzburg, D. et al (2001), On-line cost-optimization problem in project management under random disturbances, *Communications in DQM*, 4(1): 24-29.
84. Golenko-Ginzburg, D. et al (2002), On-line cost-optimization model for several stochastic network projects with different speeds, *Communications in DQM*, 5(1): 21-32.
85. Golenko-Ginzburg, D. et al (2003), Resource constrained scheduling simulation model for alternative stochastic network projects, *Math. Comp. Sim.,* 63(2): 105-117.
86. Golenko-Ginzburg, D. and Gonik, A. (2004), Hierarchical control model for several stochastic network projects, *Communications in DQM*, 7(1): 19-26.
87. Golenko-Ginzburg, D. and Malisheva, A. (2004), Control model for several construction projects, *Communications in DQM*, 7(1): 52-62.
88. Golenko-Ginzburg, D. et al (2005), Decision-making simulation model for controlling several stochastic projects, *Communications in DQM*, 8(1): 33-43.
89. Golenko-Ginzburg, D. and Gonik, A. (2005), Optimization problems for alternative stochastic network models, *Communications in DQM*, 8(1): 65-73.
90. Golenko-Ginzburg, D., Gonik, A. and Baron, A. (2006), Resource constrained project scheduling models under random disturbances, Ch. 3 in *Perspectives in Modern Project Scheduling*, J. Jozefowska and J. Weglarz (eds.), New-York: Springer Science Publishers, pp. 53-78.
91. Golenko-Ginzburg, D. et al (2008), Managing alternative stochastic network projects, *Proceedings of the 22nd IPMA World Congress*, Rome, November 9-11, II: 804-809.
92. Gonik, A. (1995), Planning and controlling multilevel stochastic projects, Ph.D. Thesis, *Ben-Gurion University of the Negev*, Beer-Sheva, Israel.
93. Gonik, A. (1999), Resource scheduling model with cost objectives for stochastic network projects, *Communications in DQM*, 2(1): 102-108.
94. Gonik, A. et al (2003), Resource constrained scheduling for several stochastic network projects with different priorities, *Communications in DQM*, 6(1): 66-71.
95. Grubbs, F.E. (1962), Attempts to validate certain PERT statistics or "picking on PERT", *Oper. Res.*, 10: 912-915.

96. Gumbel, E. (1962), *Statistics of Extremes*, New-York: Columbia University Press.
97. Harrison, J.M. and Wein, L.M. (1990), Scheduling networks of queues: Heavy traffic analysis of a two-station closed network, *Oper. Res.*, 38: 1051-1064.
98. Hartley, H.O. and Wortham, A.W. (1966), A statistical theory for PERT classical path analysis, *Mgmt. Sci.*, B(12): 469-481.
99. Hasting, M.A. and Mello, J.M. (1979), *Decision Networks*, New-York: Wiley.
100. Healy, T.L. (1961), Activity subdivision and PERT probability statements, *Oper. Res.*, 9(3): 104-108.
101. Hughes, M.W. (1986), Why projects fail: An effect of ignoring the obvious, *Ind. Engn.*, 18(4): 14-18.
102. Kamburovski, J. (1985), Normally distributed activity duration in PERT networks, *J. Oper. Res. Soc.*, 36: 1051-1057.
103. Kamburovski, J. (1986), An upper bound on the expected completion time of PERT networks, *Eur. J. Oper. Res.*, 21: 206-212.
104. Kelley, J.E. Jr. (1961), Critical path planning and scheduling: Mathematical basis, *Oper. Res.*, 9(3): 296-320.
105. Kidd, J. (1987), A comparison between the VERT program and other methods of project duration estimates, *Omega*, 15(2): 129-134.
106. Kidd, J. (1989), Project analysis today - the end users' disquiet? *Omega*, 17(2): 103-119.
107. Kleindorfer, G.B. (1971), Bounding distributions for a stochastic cyclic network, *Oper. Res.*, 19: 1586-1601.
108. Klingel, A.R., Jr. (1966), Bias in PERT project completion time calculations for a real network, *Mgmt. Sci.*, B13(4): 194-201.
109. Kolish, R. (1995), *Project Scheduling Under Resource Constraints*, Phisica-Verlag.
110. Lawler, E.L. (1976), Combinatorial Optimization: Networks and Matroids, New-York: Rinehart and Winston.
111. Lee, S.M., Moeller, G.L. and Digman, L.D. (1981), Network Analysis for Management Decisions, Boston: Kluwer-Nijhoff.
112. Littlefield, T.K. Jr. and Randolph, P.H. (1987), Another note on PERT times, *Mgmt. Sci.*, 33: 1357-1359.
113. Lorterapong, P. (1994), A fuzzy heuristic method for resource constrained project scheduling, *Project Management Journal*, 25(4): 12-18.
114. Luenberger, D.G. (1989), *Linear and Non-Linear Programming*, 2nd ed., Massachusetts: Addison Wesley Publishing Co.
115. Lukaszevicz, J. (1965), On the estimation of errors introduced by standard assumptions concerning the distribution of activity duration PERT calculations, *Oper. Res.*, 13: 326-327.
116. MacCrimmon, K.R. and Ryaveck, C.A. (1964), An analytical study of the PERT assumptions, *Oper. Res.*, 12: 16-37.
117. Malcolm, D., Roseboom, J., Clark, C. and Fazar, W. (1959), Application of a technique for research and development program evaluation, *Oper. Res.*, 7: 646-669.
118. Malisheva, A. (2005), Control and planning models for aggregated projects in a

project office, Ph.D. Thesis, *Ben-Gurion University of the Negev*, Beer-Sheva, Israel.
119. Martin, J.J. (1965), Distribution of the time through a directed acyclic network, *Oper. Res.*, 13: 46-66.
120. Mesarovich, M. et al (1970), *Theory of Hierarchical Multilevel Systems*, New-York: Academic Press.
121. Moder, J.J., Phillips, C.R and Davis, E.W. (1983), *Project Management with CPM and PERT and Precedence Diagramming*, New-York: Van-Nostrand Reinhold Co., Inc.
122. Moder, J.J. and Cecil, R.P. (1970), *Project Management with CPM and PERT*, New-York: Van-Nostrand Reinhold Co., Inc.
123. Moeller, G.L. and Digman, L.D. (1981), Operations planning with VERT, *Oper. Res.*, 29(4): 676-697.
124. Monciardi, R., Paolucci, M. and Puliafito, P.P. (1994), Development of heuristic project scheduler under resource constraints, *Eur. J. Oper. Res.*, 79(2): 176-182.
125. Murray, J.E. (1962), *Consideration of PERT Assumptions*, Conduction Corporation, Ann. Arbor, Michigan.
126. Muth, T. and Thompson, G. (eds., 1963), *Industrial Scheduling*, NJ, Englewood Cliffs: Prentice Hall.
127. Orlin, I.B. (1993), Parallel algorithms for the assignment and minimum-cost problems, *Oper. Res. Letters*, 14: 181-186.
128. Pearson, A.W. (1990), Planning and control in research and development, *Omega*, 18(6): 573-581.
129. Perry, C. and Greig, I.D. (1975), Estimating the mean and variance of subjective distributions in PERT and decision analysis, *Mgmt. Sci.*, 21(12): 1477-1480.
130. Phillips, D. and Garcia-Diaz, A. (1990), *Fundamentals of Network Analysis*, NJ, Englewood Cliffs: Prentice Hall.
131. Pritsker, A.A.B. and Happ, W.W. (1966), GERT: Graphical evaluation and review technique, Part I: Fundamental, *J. Indust. Engn.*, 17(5): 267-274.
132. Pritsker, A.A.B. and Whitehouse, G.E. (1966), GERT: Graphical evaluation and review technique, Part II: Probabilistic and industrial engineering applications, *J. Indust. Engn.*, 17(6): 293-301.
133. Pritsker, A.A.B. (1977), *Modeling and Analysis Using Q-GERT Networks*, New-York: Wiley.
134. Rajagopalan, S. (1994), Capacity expansion with alternative technology choices, *Eur. J. Oper. Res.*, 77: 392-403.
135. Ringer, L.J. (1969), Numerical operators for statistical PERT critical path analysis, *Mgmt. Sci.*, B(16): 136-143.
136. Ringer, L.J. (1971), A statistical theory of PERT in which completion time of activities are independent, *Mgmt. Sci.*, 17: 717-723.
137. Robillard, P. and Trahan, M. (1976), Expected completion time in PERT networks, *Oper. Res.*, 24: 177-182.
138. Robillard, P. and Trahan, M. (1977), The completion time of PERT networks, *Oper. Res.*, 25: 15-29.

139. Saaty, T.L. (1980), *The Analytic Hierarchy Process*, New-York: McGraw-Hill.
140. Sahni, S. (1974), Computationally related problems, *SIAM J. Comp.*, 3(4): 262-279.
141. Satoshi, H. (2002), *Inside the Mind of Toyota*, New-York: Productivity Press.
142. Schneeweiss, Ch. (1995), A conceptual framework for hierarchical planning and bargaining, in: *Design Models for Hierarchical Organizations: Computation, Information and Decentralization*, B. Obel and R. Burton (eds.), 137-160.
143. Schonberger, R.J. (1981), Why projects are "always" late: A rationale based on manual simulation of a PERT/CPM network, *Interfaces*, 11(5): 66-70.
144. Schrage, L. (1991), *LINDO: An Optimization Modeling System*, Chicago, Illinois: The Scientific Press.
145. Scully, D. (1983), The completion time of PERT networks, *J. Oper. Res. Soc.*, 34: 155-158.
146. Scully, D. (1989), A historical note on PERT times, *Omega*, 17(2): 195-196.
147. Sethi, S.P. and Zhang, Q. (1994), *Hierarchical Decision-Making in Stochastic Manufacturing Systems*, Cambridge, MA: Birkhauser Boston.
148. Shigeo, Sh. (1995), *New Japanese Manufacturing Philosophy*, Novi Sad: Prometheus.
149. Shtub, A., Bard, J. and Globerson, S. (1994), *Project Management: Engineering, Technology and Implementation*, New-York: Prentice Hall International, Inc.
150. Sinuany-Stern, Z. et al (1996), Production control in semiautomated production systems, *Prod. Plan. Cont.*, 7(2): 176-185.
151. Sitniakovski, Sh. (2002), Control and scheduling models in stochastic project management, Ph.D. Thesis, *Ben-Gurion University of the Negev*, Beer-Sheva, Israel.
152. Slyke, R. Van (1963), Monte-Carlo methods and the PERT problem, *Oper. Res.*, 11(5): 58-65.
153. Taha, H.A. (1987), *Operations Research: An Introduction*, New-York: MacMillan.
154. Thesen, A. (1976), Heuristic scheduling of activities under resource and precedence restrictions, *Mgmt. Sci.*, 23(4): 412-442.
155. Toker, A., Kondacsi, S. and Erkip, N. (1991), Scheduling under a non-renewable resource constraint, *J. Oper. Res. Soc.*, 42(9): 811-814.
156. Ulisoy, G. and Ozdamar, R. (1995), A heuristic scheduling algorithm for improving the duration and net present value of a project, *Int. J. Oper. Prod. Mgmt.*, 15(1): 89-98.
157. Voropaev, V. et al (2000), Structural classification of network models, *Int. J. of Proj. Mgmt.*, 18: 361-368.
158. Voropaev, V. et al (2008), Network models in project management (Theory and practice: Successes, disappointments, future perspectives), *Proceedings of the 22nd IPMA World Congress*, Rome, November 9-11, I: 645-650.
159. Voropaev, V. et al (2009), Decision making in controlled cyclic alternative network projects with deterministic branching outcomes, *Proceedings of the 23rd IPMA World Congress*, Helsinki, June 15-16.

160. Welsh, D. (1965), Errors introduced by a PERT assumption, *Oper. Res.*, 13: 141-143.
161. Whitehouse, G.E. (1973), *Systems Analysis and Design Using Networks Techniques*, NJ, Englewood Cliffs: Prentice Hall.
162. Williams, T.M. (1992), Criticality in stochastic networks, *J. Oper. Res. Soc.*, 43(4): 353-357.
163. Williams, T.M. (1992), Practical use of distributions in network analysis, *J. Oper. Res. Soc.*, 43(3): 265-270.
164. Williams, T.M. (1995), What are PERT estimates? *J. Oper. Res. Soc.*, 46(12): 1498-1504.
165. Willis, R.J. (1985), Critical path analysis and resource constrained project scheduling: Theory and practice, *Eur. J. Oper. Res.*, 21: 149-155.
166. Xespos, R.F. and Strassman, P.A. (1965), Stochastic decision trees for the analysis of investment decisions, *Mgmt. Sci.*, B(11): 244-259.
167. Zhan, J. (1994), Heuristics for scheduling resource constrained projects in MPM networks, *Eur. J. Oper. Res.*, 76(1): 192-205.

Научное издание

Дмитрий Голенко-Гинзбург

Стохастические сетевые модели в инновационном проектировании

Монография

На английском языке

Публикуется в авторской редакции

Дизайн обложки С.А.Кравец

Dmitry Golenko-Ginzburg

Stochastic network models in innovative projecting

Подписано в печать 04.05.2011. Формат 60x84 1/16
Усл. печ. л. 22,2. Заказ 0000. Тираж 500.

ООО Издательство «Научная книга»
394077, г. Воронеж, ул. 60-й Армии, 25-120
www.sbook.ru

Отпечатано с готового оригинал-макета в ООО «Цифровая полиграфия»
394036, г. Воронеж, ул. Ф. Энгельса, 52.
Тел.: (4732)61-03-61