УДК 001.89 : 519.82

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ КАК ЗАДАЧА ОПТИМАЛЬНОГО БЫСТРОДЕЙСТВИЯ

В. Н. БУРКОВ

(Москва)

Рассматриваются задачи распределения ограниченных ресурсов по комплексу работ с точки зрения теории оптимального управления. Дается общая постановка задачи, и исследуется частный случай независимых работ. В этом случае удается получить эффективные алгоритмы решения при достаточно общих предположениях относительно множества допустимых управлений U и функций скорости выполнения работ. Даются постановки некоторых нерешенных задач.

Введение

Одной из основных задач теории систем сетевого планирования и управления (СПУ) является задача распределения ограниченных ресурсов по комплексу операций. Пусть имеется n операций A_i. Обозначим через $x_i(t)$ состояние i-й операции в момент t, т. е. объем этой операции, выполненный к моменту t ($x_i(0) = 0$, $x_i(T) = W_i$, где W_i — объем i-й операции, T — момент окончания комплекса). Будем считать, что скорость выполнения i-й операции w_i зависит от времени и количества ресурсов на этой операции, т. е.

$$w_i = f_i(u_i^j(t), t), \tag{1}$$

где $u_i^j(t)$ — количество ресурсов j-го вида, выполняющих i-ю операцию в момент t. Набор векторов $u_i(t) = (u_i^1(t), u_i^2(t), \ldots, u_i^k(t))$ (k — число различных видов ресурсов) принадлежит множеству допустимых распределений ресурсов U. Требуется определить допустимое распределение $u(t) = (u_1(t), \ldots, u_n(t))$, минимизирующее время выполнения комплекса T. В работе рассматривается случай независимых операций, т. е. операций, которые можно выполнить одновременно (сетевой график состоит из параллельных дуг). В разделе 1 задача решается при условии, что скорость выполнения операций не зависит явно от времени. Решение в общем случае сводится к определению точки пересечения прямой с границей выпуклой оболочки некоторого множества. Эту задачу можно приближенно свести к задачам линейного программирования, но довольно большого объема. В разделе 2 рассматривается пример преодоления вычислительных трудностей. Некоторые случаи явной зависимости от времени рассматриваются в разделе 3. В разделе 4 даются постановки двух задач синтеза сетей.

1. Решение задачи при скорости выполнения операций, не зависящей явно от времени

Сделаем несколько допущений относительно вида функций $f_i(u_i)$ (предполагаем, что операция выполняется ресурсами только одного вида) и множества U. Будем считать $f_i(u_i)$ ($u_i \geqslant 0$) неубывающей функцией u_i,

причем $f_i(0) = 0$, т. е. при отсутствии ресурсов операция не может выполняться. Определим покрытие $\hat{U}$ множества U (U — замкнутое множество) следующим образом:

а) для любого $u' \in \hat{U}$ существует $u \in U$ такое, что $u \geqslant u'$;

б) если $u \in U$ и $u \geqslant u' \geqslant 0$, то $u' \in \hat{U}$.

Множество, совпадающее со своим покрытием, назовем правильным.

При сделанных допущениях относительно функций $f_i(u_i)$ можно без ограничения общности рассматривать только правильные множества допустимых распределений. Действительно, если $u_{\text{опт}} \in \hat{U}$ и $u'_{\text{оп}^{\text{т}}} \notin U$, то по свойству «а» найдется $u \in U$, которое «не хуже» $u'_{\text{опт}}$. С другой стороны $U \subset \hat{U}$, поэтому u «не лучше» $u'_{\text{опт}}$. Следовательно, u — также оптимальное распределение.

Определим множество V следующим образом:

$$v \in V \Leftrightarrow u \in U, \text{ где } v_i = f_i(u_i). \tag{2}$$

Лемма 1. Если U — правильное множество, то и V — правильное множество. Доказательство тривиально.

Будем говорить, что пара (U, f), $f = (f_1, f_2, \ldots, f_n)$ удовлетворяет условию достаточной выпуклости, если V — выпуклое множество.

Лемма 2. Если U — выпуклое правильное множество и $f_i(u_i)$ — выпуклые (кверху) функции, то (U, f) удовлетворяет условию достаточной выпуклости.

Доказательство. Имеем

$$u^1 \in U, \qquad u^2 \in U \Rightarrow \lambda u^1 + (1 - \lambda) u^2 \in U \qquad (0 \leqslant \lambda \leqslant 1).$$

Далее, $v^1 = f(u^1) \in V$, $v^2 = f(u^2) \in V$, $v = f(\lambda u^1 + (1 - \lambda) u^2) \in V$, $v^0 = \lambda v^1 + (1 - \lambda) v^2 = \lambda f(u^1) + (1 - \lambda) f(u^2) \leqslant f(\lambda n^1 + (1 - \lambda) u^2) = v \in V$.

Отсюда в силу правильности V, $v^0 \in V$, что и требовалось доказать.

Определим кривую L (линия равномерного выполнения операций) уравнениями $f_i(u_i) = \alpha W_i$ $(0 \leqslant \alpha < \infty, \quad i = 1, 2, \ldots, n)$, α — параметр.

Теорема 1. Если пара (U, f) удовлетворяет условию достаточной выпуклости, то оптимальное распределение определяется точкой пересечения линии равномерного выполнения операций с границей множества допустимых распределений.

Доказательство. Произведем замену переменных $v_i = f_i(u_i)$. Тогда $w_i = v_i$, $v = (v_1, v_2, \ldots, v_n) \in V$ — выпуклое множество. Уравнение линии $L: v_i = \alpha W_i$, $i = 1, \ldots, n$. Пусть $v(t)$ — допустимое распределение и T — время выполнения комплекса, т. е. $\int_0^T v_i(t)\, dt = W_i$. Положим

$$v_i' = \frac{1}{T} \int_0^T v_i(t)\, dt = \frac{W_i}{T}.$$ Распределение v' допустимо, принадлежит линии L и при этом время выполнения комплекса не меняется. Следовательно, для любого $v(t) \in V$ существует постоянное в интервале $(0, T)$ распределение $v' \in L \cap V$, эквивалентное $v(t)$. Отсюда следует утверждение теоремы, так как минимум $T = 1/\alpha$ достигается при максимальном α, т. е. в точке пересечения границы V и L.

Рассмотрим случай, когда скорость выполнения операции прямо пропорциональна количеству ресурсов (изменив объем W_i, всегда можно сделать коэффициент при $u_i(t)$ в (3) равным 1) и U задается системой линейных неравенств

$$w_i(t) = u_i(t) \qquad (i = 1, 2, \ldots, n), \tag{3}$$

$$\sum_{i=1}^{n} b_{ij} u_i(t) \leqslant c_j \qquad (j = 1, 2, \ldots, m, \quad c_j > 0, \quad b_{ij} \geqslant 0). \tag{4}$$

Имеем $u_i = \alpha W_i$ $(i = 1, 2, \ldots, n)$. Подставляя в (4), получаем $\alpha \sum_{i=1}^{n} b_{ij} W_i \leqslant c_j$, откуда

$$\alpha_{\text{макс}} = \min_j \frac{c_j}{\sum_{i=1}^{n} W_i b_{ij}}, \tag{5}$$

минимальное время выполнения комплекса

$$T_{\text{мин}} = \max_j \frac{\sum_{i=1}^{n} W_i b_{ij}}{c_j} \tag{6}$$

и оптимальное распределение ресурсов

$$u_i^0 = W_i \alpha_{\text{макс}} = \frac{W_i}{T_{\text{мин}}}. \tag{7}$$

Замечание. Полученные формулы справедливы не только при U — правильном множестве.

Можно показать, что если U — правильное множество, то все $b_{ij} \geqslant 0$ (при условии, что каждое неравенство в (4) является существенным, т. е. для любого k найдется u, при котором $\sum_{i=1}^{n} b_{ik} u_i = c_k$ и $\sum_{i=1}^{n} b_{ij} u_i < c_j (j \neq k)$). Пусть некоторые $b_{ij} < 0$. Если $\sum_{i=1}^{n} W_i b_{ij} > 0$ для всех $j = 1, 2, \ldots, m$ и $u_i^0 = \alpha_{\text{макс}} W_i$ обращает в равенство только неравенства в (4), не имеющие отрицательных b_{ij}, то (6), (7) также справедливы. В противном случае получаем задачу линейного программирования: максимизировать $\alpha \geqslant 0$ при условиях

$$u_i - W_i \alpha \geqslant 0 \qquad (i = 1, 2, \ldots n), \qquad \sum_{i=1}^{n} b_{ij} u_i \leqslant c_j \qquad (j = 1, 2, \ldots, m). \tag{8}$$

Рассмотрим некоторые обобщения предыдущих результатов.

Задача 1. Пусть состояние операции A_i меняется во времени с постоянной скоростью d_i (работа сил природы и т. п.). В этом случае $w_i = u_i + d_i$, $\sum_{i=1}^{n} b_{ij} u_i \leqslant c_j (c_j > 0;\ b_{ij} \geqslant 0;\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m)$.

Полагая $v_i = u_i + d_i$, получаем $\sum_{i=1}^{n} v_i b_{ij} \leqslant c_j + \sum_{i=1}^{n} d_i b_{ij}$ $(j = 1, \ldots, m)$

(для разрешимости задачи необходимо $c_j + \sum_{i=1}^{n} d_i b_{ij} > 0$). Решая аналогично предыдущему пункту, получаем

$$v_i^0 = W_i \min \frac{c_j + \sum_{i=1}^{n} d_i b_{ij}}{\sum_{i=1} W_i b_{ij}}.$$

Пусть v_i^0 $(i = k+1, \ldots, n)$ меньше d_i (т. е. $u_i^0 = v_i^0 - d_i < 0$). Полагаем $v_i^1 = d_i (u_i^1 = 0)$ для $i = k+1, \ldots, n$ и решаем задачу минимизации T при $w_i = v_i$ $(i = 1, 2, \ldots, k)$ и $\sum_{i=1}^{k} b_{ij} v_i \leqslant c_j + \sum_{i=1}^{k} d_i b_{ij}$ $(j = 1, 2, \ldots, m)$. Минимальное время выполнения комплекса

$$T_{\text{мин}} = \max_j \frac{\sum_{i=1}^{k} W_i b_{ij}}{c_j + \sum_{i=1}^{k} d_i b_{ij}}.$$

Задача 2. Во многих случаях данная операция должна выполняться при определенном соотношении ресурсов нескольких видов. Выбрав для каждой операции один из видов ресурсов в качестве основного и выразив через него все остальные, получим задачу предыдущего пункта.

Задача 3. Скорость выполнения i-й операции зависит от ее состояния, т. е. $\frac{dx_i}{dt} = w_i = \varphi_i(x_i) u_i(t)$. Замена $y_i = \int_0^{x_i} \frac{dx}{\varphi_i(x)}$ сводит эту задачу к рассмотренному случаю $w_i^1 = \frac{dy_i}{dt} = \frac{dy_i}{dx_i} \frac{dx_i}{dt} = u_i(t)$ $(i = 1, 2, \ldots, n)$.

Если пара (U, f) не удовлетворяет условию достаточной выпуклости, то множество V уже не будет выпуклым множеством. Решим задачу, взяв в качестве множества допустимых распределений выпуклую оболочку $\tilde{V}$ множества V. Пусть v^0 — оптимальное распределение. Существует не более n линейно независимых векторов $v^i \in V$, выпуклая линейная комбинация которых дает v^0. Очевидно, эти векторы образуют оптимальное распределение. Задача определения точки пересечения границы выпуклой оболочки множества V и линии равномерного выполнения операций L достаточно сложна в общем случае. Рассмотрим несколько примеров.

1. Множество U состоит из конечного числа точек u^k $(k = 1, \ldots, s)$. Замена $v = f(u)$ $(v_i = f_i(u_i),\ i = 1, \ldots, n)$ переводит каждую u^k в $v^k = f(u^k)$. Получаем задачу линейного программирования: определить $c_k \geqslant 0$, $\alpha \geqslant 0$ так, чтобы

$$\sum_{k=1}^{s} c_k v_i^k - \alpha W_i \geqslant 0 \quad (i = 1, 2, \ldots, n), \qquad \sum_{k=1}^{s} c_k = 1$$

и α приняло максимальное значение (в разделе 2 описан метод решения частного случая этой задачи). Таким образом можно приближенно решать общую задачу, заменив множество U конечным числом точек.

I. Пусть $f_i(u_i) = u_i^{1/\beta_i}$ $(\beta_i > 0)$ и множество U задается неравенством $\sum_{i=1}^{n} u_i(t) \leqslant N$. В случае $\beta_i \geqslant 1$ функции $f_i(u_i)$ являются выпуклыми кверху и согласно результатам предыдущих пунктов $T_{\text{мин}} = 1/\alpha$, где α находится из уравнения

$$\sum_{i=1}^{n} (W_i\alpha)^{\beta_i} = N.$$

Если $\beta_i < 1$, то f_i — выпуклые книзу функции и V — невыпуклое множество. Однако точки v^j вида $v_j^j = f_j(N)$, $v_i^j = 0$ $(i \neq j)$, $(j = 1, 2, \ldots, n)$ являются крайними точками множества V, поэтому оптимальным является поочередное выполнение операций, причем $T_{\text{мин}} = \sum_{i=1}^{n} W_i N^{-1/\beta_i}$.

2. Минимизация времени выполнения комплекса при постоянных интенсивностях потребления ресурсов

Предположим, что операция может выполняться только при наличии на ней вполне определенного количества ресурсов β_i. В этом случае общее время выполнения операции $\tau_i = W_i / f_i(\beta_i)$. Задача заключается в минимизации времени выполнения комплекса при ограниченном количестве ресурсов N.

Обозначим через $y = (y_1, y_2, \ldots, y_n)$ вектор, состоящий из нулей и единиц и такой, что $\sum_{i=1}^{n} y_i\beta_i \leqslant N$ (очевидно, распределение $u = \{y_i\beta_i\}$ является в этом случае допустимым). Пусть $Y = \{y^k\}$ $(k = 1, \ldots, s)$ — множество таких векторов. Обозначим через $t_k \geqslant 0$ длину интервала времени, в котором $u = \{y_i^k\beta_i\}$. Задача заключается в минимизации $T = \sum_{k=1}^{s} t_k$ при ограничениях $\sum_{k=1} t_k y_i^k = \tau_i$ $(i = 1, 2, \ldots, n)$.

Хотя это задача линейного программирования, но большое число допустимых векторов $(s \leqslant 2^n)$ затрудняет применение обычных методов [1]. Предлагается несколько изменить симплекс-метод, определяя очередной вектор для ввода в базис путем решения комбинаторной задачи (так называемая задача о ранце).

Таблица 1

i	1	2	3	4
τ_i	12	15	10	20
β_i	17	12	8	10

Описание алгоритма проведем на примере. Значения β_i и τ_i заданы в табл. 1.

I этап. Получение допустимого решения. Берем произвольный допустимый вектор $y^1 = \{y_i^1\}$, определяем $t_1 = \min_{y_i^1 = 1} \tau_i$ и полагаем $\tau_i^1 = \tau_i - y_i^1 t_1$ (целесообразно выбирать допустимые векторы так, чтобы $\sum_{i=1}^{n} y_i\beta_i$ была как можно ближе к N). Затем

берем новый допустимый вектор $y^2 = \{y_i^2\}$, определяем $t_2 = \min\limits_{y_i^2=1} \tau_i^1$, полагаем $\tau_i^2 = \tau_i^1 - y_i^2 t_2$ и т. д., пока все операции не будут выполнены. Для нашего примера возьмем $y^1 = (1, 1, 0, 0)$, $t_1 = \min(\tau_i, \tau_2) = 12$, $\tau_1^1 = 0$, $\tau_2^1 = 3$, $\tau_3^1 = 10$, $\tau_4^1 = 20$.

Далее берем $y^2 = (0, 1, 1, 1)$, $t_2 = \min(\tau_2^1, \tau_3^1, \tau_4^1) = 3$, $\tau_1^2 = 0$, $\tau_2^2 = 0$, $\tau_3^2 = 7$, $\tau_4^2 = 17$. Продолжая таким образом, получаем решение

$$y^1 = (1, 1, 0, 0),\ t_1 = 12; \qquad y^2 = (0, 1, 1, 1,),\ t_2 = 3;$$
$$y^3 = (0, 0, 1, 1),\ t_3 = 7; \qquad y^4 = (0, 0, 0, 1);\ t_4 = 10$$

с временем выполнения комплекса $T = 32$.

II. этап. Процесс улучшения допустимого решения. Выразим векторы $x^1 = (1, 0, 0, 0)$, $x^2 = (0, 1, 0, 0)$, $x^3 = (0, 0, 1, 0)$, $x^4 = (0, 0, 0, 1)$ через базис y^1, y^2, y^3, y^4. Легко определяем

$$x^4 = y^4,\ x^3 = y^3 - y^4,\ x^2 = y^2 - y^3,\ x^1 = y^1 - y^2 + y^3. \qquad (9)$$

Обозначим $a_i = (x^i E)$, где $E = (1, 1, 1, 1)$ и x^i выражены через y^j $(j = 1, 2, 3, 4)$, и поставим следующую комбинаторную задачу: определить $z_i = \{0, 1\}$, максимизирующие $P = \sum\limits_{i=1}^{n} a_i z_i$ при условии $\sum\limits_{i=1}^{n} \beta_i z_i \leqslant N$ (задача о ранце).

Если максимальное значение P больше 1, то решение можно улучшить, введя в базис вектор, максимизирующий P (в противном случае решение оптимально). В нашем примере $a_1 = 1$, $a_2 = 0$, $a_3 = 0$, $a_4 = 1$. Получаем задачу: максимизировать $z_1 + z_4$ при условии $17z_1 + 10z_4 \leqslant 30$. Решение очевидно: $z_1 = 1$, $z_4 = 1$, $P = 2 > 1$, следовательно, вектор $x^1 + x^4 = y^1 - y^2 + y^3 + y^4$ подлежит вводу в базис. Для определения вектора, который удаляется из базиса, следуя обычной процедуре симплекс-метода, находим минимальное отношение t_i / b_i, где b_i — положительные коэффициенты разложения вектора $x^1 + x^4$, вводимого в базис, по базису (y^1, y^2, y^3, y^4). Имеем $b_1 = 1$, $b_2 = -1$, $b_3 = 1$, $b_4 = 1$; $\min\limits_{b_i>0}\left(\frac{t_i}{b_i}\right) = t_3 = 7$. Следовательно, вектор y^3 подлежит удалению из базиса. Новый базис (y^1, y^2, y^5, y^4), где $y^5 = x^1 + x^4 = (1, 0, 0, 1)$. Соответственно изменяются времена $t_1^1 = t_1 - 7 = 5$, $t_2^1 = t_2 + 7 = 10$, $t_5^1 = 7$, $t_4^1 = t_4 - 7 = 3$. Заменяя в (9) y^3 на $y^5 - y^1 + y^2 - y^4$, получаем $x^1 = y^5 - y^4$, $x^2 = y^1 - y^5 + y^4$, $x^3 = -y^1 + y^2 + y^5 - 2y^4$, $x^4 = y^4$. Далее повторяем процесс: $a_1 = 0$, $a_2 = 1$, $a_3 = -1$, $a_4 = 1$. Решаем задачу: максимизировать $P = z_2 + z_4$ при условии $12z_2 + 10z_4 \leqslant 30$. Решение $z_2 = 1$, $z_4 = 1$, $P = 2 > 1$. Вводим в базис вектор $y^6 = x^2 + x^4 = y^1 - y^5 + 2y^4$, удаляем вектор y^4, так как $\min\left(\frac{t_1^1}{1}, \frac{t_4^1}{2}\right) = \frac{t_4^1}{2} = 1\frac{1}{2}$. Получаем $t_1^2 = t_1^1 - 1\frac{1}{2} = 3\frac{1}{2}$, $t_2^2 = t_2^1 = 10$, $t_5^2 = t_5^1 + 1\frac{1}{2} = 8\frac{1}{2}$, $t_6^2 = 1\frac{1}{2}$, $x^1 = \frac{1}{2}y + \frac{1}{2}y^5 - \frac{1}{2}y^6$, $x^2 = \frac{1}{2}y^1 - \frac{1}{2}y^5 + \frac{1}{2}y^6$, $x^3 = y^2 - y^6$, $x^4 = \frac{1}{2}y^1 + \frac{1}{2}y^5 - \frac{1}{2}y^6$, где $y^6 = y^1 - y^5 + 2y^4$. Имеем $a_1 = \frac{1}{2}$, $a_2 = \frac{1}{2}$, $a_3 = 0$, $a_4 = \frac{1}{2}$. Максимальное значение формы $P = \frac{1}{2}z_1 + \frac{1}{2}z_2 + \frac{1}{2}z_4$ при ограничении $17z_1 + 12z_2 + 10z_4 \leqslant 30$ равно 1, поэтому полученное на последнем шаге решение оптимально. Минимальное время выполнения комплекса $T_{\text{опт}} = t_1^2 + t_2^2 + t_5^2 + t_6^2 = 23\frac{1}{2}$.

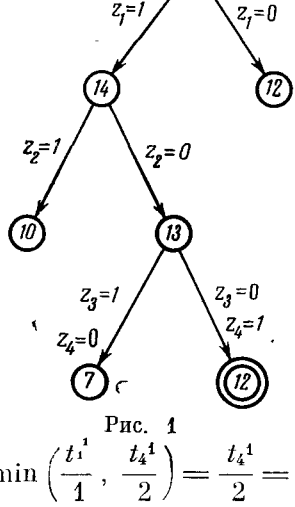

Рис. 1

В рассмотренном примере оптимальное решение задачи о ранце на каждом шаге было очевидным. В более сложных примерах для решения этой задачи можно применить метод ветвлений, взяв в качестве верхней границы подмножества решений оптимальное значение соответствующей непрерывной задачи. Рассмотрим простой пример: максимизировать $P = 3z_1 + 3z_2 + 4z_3 + 9z_4$ при ограничении $z_1 + 2z_2 + 3z_3 + 9z_4 \leqslant 10$ и $z_i = \{0, 1\}$ (z_i расположены так, что отношения коэффициентов $^3/_1$, $^3/_2$, $^4/_3$, $^9/_9$ образуют невозрастающую последовательность). Пусть $z_1 = 1$, тогда оптимальное нецелочисленное решение $z_1 = 1$, $z_2 = 1$, $z_3 = 1$, $z_4 = {}^4/_9$, $P = 14$ (примем значение $P = 14$ за верхнюю границу значений P для подмножества решений, в которых $z_1 = 1$). Аналогично, при $z_1 = 0$ получаем оптимальное нецелочисленное решение $z_1 = 0$, $z_2 = 1$, $z_3 = 1$, $z_4 = {}^5/_9$, $P = 12$. Двигаемся по максимуму верхней границы, т. е. полагаем $z_1 = 1$. Далее, если $z_2 = 1$, то единственное целочисленное решение $z_1 = 1$, $z_2 = 1$, $z_3 = 1$, $z_4 = 0$, $P = 10$. Если $z_2 = 0$, то оценка сверху $P = 13$ (соответствующее решение $z_1 = 1$, $z_2 = 0$, $z_3 = 1$, $z_4 = {}^2/_3$). Полагая $z_2 = 0$ проверяем альтернативы $z_3 = 1$ или $z_3 = 0$. Получаем решение $z_1 = 1$, $z_2 = 0$, $z_3 = 0$, $z_4 = 1$, которое является оптимальным, так как значения верхней границы для остальных ветвей дерева ветвлений (рис. 1) не больше 12.

Замечание. Иногда процесс выполнения комплекса должен удовлетворять дополнительным условиям, запрещающим одновременное выполнение некоторых операций. Этим условиям легко удовлетворить, исключая из числа допустимых векторов запрещенные комбинации (при решении задачи о ранце это соответствует сокращению числа рассматриваемых ветвей дерева). Если в примере данного раздела запретить одновременное выполнение второй и четвертой операций, то максимальное значение $P = z_2 + z_4$ на втором шаге улучшения решения равно 1 и, следовательно, решение (y^1, y^2, y^3, y^4) будет оптимальным.

3. Решение задачи при явной зависимости скорости выполнения операций от времени

Рассмотрим прямо пропорциональную (в некоторых пределах) зависимость скорости выполнения операций от количества ресурсов, т. е. $w_i(t) = a_i(t)u_i(t)$, $u_i(t) \leqslant \beta_i(t)$ и $\sum_{i=1}^{n} u_i(t) \leqslant N(t)$. Целесообразно считать функции $\beta_i(t)$, $a_i(t)$ и $N(t)$ ступенчатыми функциями времени. Пусть $t_1, t_2, \ldots, t_k$ — последовательные интервалы постоянства $a_i(t)$, $\beta_i(t)$, $N(t)$. Обозначим через β_{ij}, a_{ij}, N_j значения $\beta_i(t)$, $a_i(t)$ и $N(t)$ в интервале t_j, φ_{ij} — объем i-й операции, выполненной в интервале t_j. Поставим вопрос, возможно ли выполнение комплекса за время $T = \sum_{i=1}^{k} t_i$?

Очевидны условия

$$\sum_{j=1}^{k} \varphi_{ij} \leqslant W_i \qquad (i = 1, 2, \ldots, n),$$

$$\sum_{i=1}^{n} a_{ij} \leqslant N_j t_j = V_j \qquad (j = 1, 2, \ldots, k), \qquad a_{ij} = 1/a_{ij}, \tag{10}$$

$$\varphi_{ij} \leqslant c_{ij} = \beta_{ij} a_{ij} t_j \qquad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k).$$

Определим транспортную сеть с вершинами x_0, $x_1, \ldots, x_n$, y_1, $y_2, \ldots, y_k$, z. Вершину x_0 соединим с каждой вершиной x_i дугой (x_0, x_i) с пропускной способностью $c(x_0, x_i) = W_i$. Каждую вершину y_j соединим с вершиной z дугой (y_j, z) с пропускной способностью $c(y_j, z) = V_j$. Наконец, каждую вершину x_i соединим с каждой вершиной y_j дугой (x_i, y_j) с пропускной способностью $c(x_i, x_j) = c_{ij}$. Определим поток φ по дугам сети следующим образом: (11)

$$\varphi(x_0, x_i) \leqslant W_i \quad (i = 1, 2, \ldots, n), \qquad \varphi(y_i, z) \leqslant V_j \quad (j = 1, 2, \ldots, k),$$
$$\varphi(x_i, y_j) = \varphi_{ij} \leqslant c_{ij} \qquad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k),$$

$$\sum_{j=1}^{n} \varphi_{ij} = \varphi(x_0, x_i) \quad (i = 1, 2, \ldots, n), \qquad \sum_{i=1}^{n} a_{ij}\varphi_{ij} = \varphi(y_j, z) \quad (j = 1, 2, \ldots, k).$$

Такой поток называется потоком с увеличениями [2] (действительно, поток по дуге (x_i, y_j) увеличивается в α_{ij}). Назовем величиной потока $\varphi_{x_0} = \sum_{i=1} \varphi(x_0, x_i) = \sum \varphi_{ij}$. Поставим задачу: определить поток максимальной величины. Рассмотрим двойственную задачу: минимизировать $\sum_{i=1}^{n} w_i W_i + \sum_{j=1}^{k} v_j W_j + \sum_{i,j} c_{ij} u_{ij}$ при ограничениях

$$w_i + \alpha_{ij} v_j + u_{ij} \geqslant 1 \quad (w_i \geqslant 0,\ v_j \geqslant 0,\ u_{i,j} \geqslant 0,$$
$$i = 1, 2, \ldots, n,\ j = 1, 2, \ldots k).$$

Для решения задачи удобно применять метод, аналогичный венгерскому методу решения транспортной задачи [3].

Идея заключается в том, что определяются начальные значения $w_i^0 = 0$, $v_j^0 = \max a_{ij}$ и в сети оставляются только такие дуги (x_i, y_j), для которых $w_i^0 + \alpha_{ij} v_j^0 = 1$ (для простоты примем $\beta_{ij} = \infty$). В полученной сети определяется максимальный поток. Если он не насыщает входные или выходные дуги, то производится соответствующее изменение чисел w_i и v_j, так что в сети появляются новые дуги, для которых $w_i + \alpha_{ij} v_j = 1$, что позволяет увеличить поток. Процесс заканчивается, когда входные или выходные дуги насыщаются. Оптимальность решения следует из того, что поток $\varphi_{ij} > 0$ только на дугах, для которых выполняется условие $w_i + \alpha_{ij} v_j = 1$. Следовательно, $\sum_{i,j} \varphi_{ij} = \sum_{i,j} (w_i + \alpha_{ij} v_j) \varphi_{ij} = \sum_{i=1}^{n} w_i W_i + \sum_{j=1}^{k} v_j V_j$ (так как в силу соотношений двойственности, если $\sum_{i=1}^{k} \varphi_{ij} < W_i$, то $w_i = 0$, а если $\sum_{i=1}^{n} \alpha_{ij} \varphi_{ij} < V_j$, то $v_j = 0$).

Пример. Пусть $n = 3$, $k = 3$, $W_1 = 10$, $W_2 = 20$, $W_3 = 10$ (см. табл. 2).

Таблица 2

№ интервала	t_j	a_{1j}	a_{2j}	a_{3j}	N_j	σ_{1j}	σ_{2j}	σ_{3j}
1	2	1/2	1	1/4	5	2	1	4
2	2	1/3	1/2	1/2	10	3	2	2
3	6	1	1/3	1/4	5	1	3	4

Положим $w_1^0 = w_2^0 = w_3^0 = 0$, $v_1^0 = 1$, $v_2^0 = 1/2$, $v_3^0 = 1$. Оставим в сети дуги, для которых $\alpha_{ij} v_j^0 = 1$ (рис. 2, *а*). Максимальный поток $\varphi_{13} = 10$, $\varphi_{21} = 10$, $\varphi_{22} = 10$. Для определения максимального потока применяется обычный алгоритм [3] с наибольшими изменениями, учитывающими увеличение потока на дугах (x_i, y_j). Дуга (y_3, z) осталась ненасыщенной. Поэтому уменьшаем v_3 до величины $v_3^1 = 1/3$. При этом для сохранения условия $w_1 + \alpha_{13} v_3 = 1$ необходимо, чтобы $w_1^1 = 1 - \alpha_{13} v_3^1 = 2/3$. Теперь в сети появляется дуга (x_2, y_3), так как $w_2^0 + 3 v_3^1 = 1$. Снова определяя максимальный поток, получаем решение, показанное на рис. 2, *б* в квадратных скобках, которое является оптималь-

ным, так как выходные дуги насыщены. Проверяем

$$\sum_{i,\,j} \varphi_{ij} = 10 + 20 + 6^2/_3 = \sum_{i=1}^{n} w_i W_i + \sum_{j=1}^{k} v_j V_j = {}^2/_3 10 + 0 \cdot 20 +$$

$$+ 0 \cdot 10 + 1 \cdot 10 + \frac{1}{2} \cdot 20 + \frac{1}{3} \cdot 30 = 36^2/_3.$$

Таким образом, за время $T = t_1 + t_2 + t_3 = 10$ комплекс выполнить нельзя. Увеличение V_3 на величину Δ позволяет увеличить поток на ве-

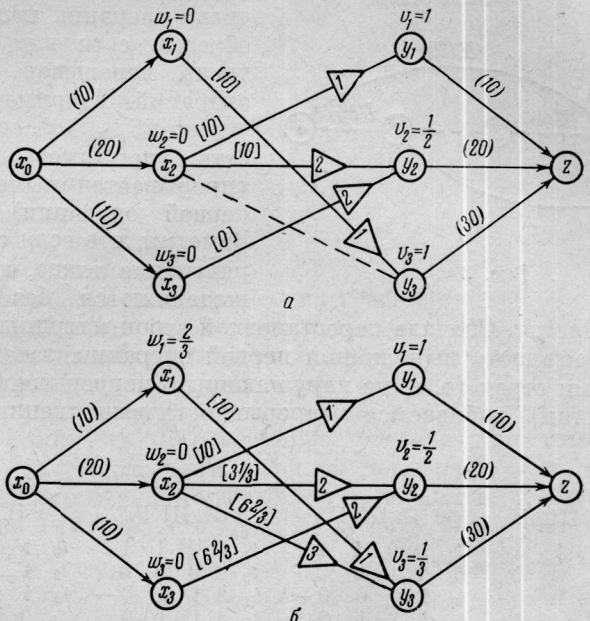

Рис. 2

личину $\Delta / 3$. Следовательно, за время $T_1 = 12$ возможно выполнение всех операций комплекса. При этом в интервале (0,2) выполняется $^1/_2$ объема второй операции, в интервале (2,4) выполняется третья операция, а в интервале (4,12) выполняется первая операция (2 единицы времени) и оставшаяся часть второй операции (6 единиц времени).

4. Задачи синтеза сетей

В этом разделе описываются две задачи теории расписаний, которые можно объединить под общим названием задачи синтеза сетевых графиков.

Задача 1. В разделе 3 была решена задача распределения ресурсов по операциям комплекса, когда i-я операция может выполняться только определенным количеством ресурсов β_i $(i = 1, 2, \ldots, n)$. Решение в общем случае состоит из n интервалов постоянства (интервалов, в которых распределение ресурсов не меняется) длительности t_i $(i = 1, 2, \ldots, n)$. Расположим интервалы в некоторой последовательности, например $t_1, t_2, \ldots, t_n$. Определим сеть из $n + 1$ событий $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$ таким образом, чтобы, если i-я операция (или ее часть) выполняется непрерывно с s-го интервала последовательности по q-й, то от события x_{s-1} к событию x_q шла дуга, соответствующая i-й операции (или ее части). Возьмем для примера, рассмотренного в разделе 3, упорядочение t_1, t_2, t_5, t_6. Первая операция выполняется в первом и третьем интервалах (см. раздел 3). Поэтому проводим в сети дуги (0—1) и (2—3), соот-

ветствующие частям первой операции. Проделав это для каждой операции, получим сеть (рис. 3, *а*). Римские цифры указывают номер операции, числа в скобках равны времени выполнения соответствующей части операции. Число операций в полученной сети $n = n + k$, где k — общее число перерывов в выполнении операций (в данном случае $k = 2$). Естественно поставить задачу определения упорядочения интервалов t_i, минимизирующего число операций полученной таким образом сети или, иначе говоря, число перерывов в выполнении операций. Например, при упорядочении t_1, t_2, t_6, t_5 соответствующая сеть будет иметь пять операций (прерывается выполнение только первой операции) (рис. 3, *б*). Заметим, что если существует упорядочение такое, что все операции выполняются без перерыва, оно легко определяется. Сначала перестановкой строк матрицы решения добиваемся непрерывности выполнения первой операции, затем допустимыми перестановками строк (т. е. не нарушающими непрерывность выполнения первой операции), добиваемся непрерывности выполнения второй операции и т. д.

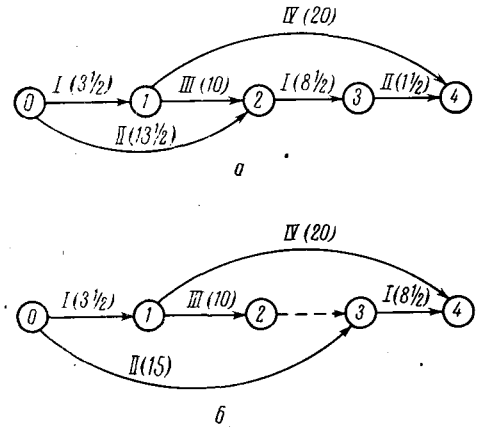

Рис. 3

Пример

	I	0	III	IV		I	II	III	IV		I	II	III	IV		I	II	III	IV
t_1	0	0	1	1	t_1	0	0	1	1	t_1	0	0	1	1	t_3	1	0	0	0
t_2	1	1	0	0	t_2	1	1	0	0	t_3	1	0	0	0	t_2	1	1	0	0
t_3	1	0	0	0 →	t_3	1	0	0	0 →	t_2	1	1	0	0 →	t_5	1	1	1	0
t_4	0	1	1	1	t_5	1	1	1	0	t_5	1	1	1	0	t_4	0	1	1	1
t_5	1	1	1	0	t_4	0	1	1	1	t_4	0	1	1	1	t_1	0	0	1	1

Если такое упорядочение невозможно, то задача сводится к определению гамильтонова пути в графе, имеющего минимальную длину. Определим граф G с $n + 2$ вершинами $y_0, y_1, \ldots, y_{n+1}$. Вершину y_0 соединим с каждой вершиной y_i $(i = 1, 2, \ldots, n)$ дугой, длина которой равна числу операций, выполняемых в интервале t_i. Каждую вершину y_i $(i = 1, 2, \ldots, n)$ соединим с вершиной y_{n+1} дугой, длина которой также равна числу операций, выполняемых в интервале t_i. Любые две вершины y_i, y_j $(i, j = 1, 2, \ldots, n)$ соединим дугами (в обоих направлениях), длина которых равна суммарному числу операций, выполняемых либо в интервале t_i, либо в интервале t_j, но не в обоих вместе. Гамильтонов путь минимальной длины от вершины y_0 к вершине y_{n+1} определит оптимальную последовательность интервалов, причем число операций в соответствующем сетевом графике равно половине длины этого пути. На рис. 4 показан соответствующий граф G для примера раздела 3.

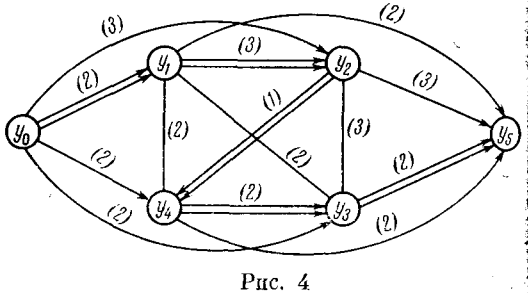

Рис. 4

Один из минимальных путей, которому соответствует сеть рис. 3, *б*, выделен двойными дугами (вершина y_3 соответствует интервалу длительности $t_5 = 8\frac{1}{2}$, вершина y_4 — интервалу длительности $t_6 = 1\frac{1}{2}$).

Задача 2. Задан комплекс из n операций. Время выполнения i-й операции τ_i. Прерывать выполнение начатой операции запрещается. Кроме того, на процесс выполнения комплекса наложены ограничения, запрещающие одновременное выполнение некоторых операций. Такие ограничения удобно задавать в виде графа G, в котором вершины соответствуют операциям и вершины x_i, x_j соединены ребром, если запрещено их одновременное выполнение. Требуется выполнить комплекс за минимальное время.

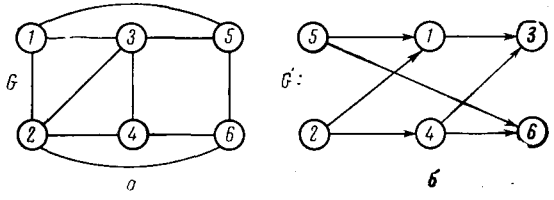

Рис. 5

Эта задача обобщает известную задачу составления расписания уроков. Будем говорить, что симметрический граф допускает сеть, если существует сеть G' такая, что две вершины графа G тогда и только тогда связаны ребром, когда они принадлежат пути в сети G'. На рис. 5, *а* приведен пример графа G, допускающего сеть G' рис. 5, *б*. Если граф допускает сеть, то эта сеть является оптимальной в смысле минимума времени выполнения комплекса. К сожалению, условия допустимости данным графом сети не получены. Если граф не допускает сети, то возникает задача определения ребер, которые следует добавить, чтобы граф допускал сеть и длина критического пути этой сети была минимальной.

Поступила в редакцию
15 апреля 1965 г.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование. Физматгиз, 1964.
2. Jewell William S. Optimal Flow through Network with Gains. Operat. Res. Quart., v. 10, No. 4, 1962.
3. Берж К. Теория графов и ее применения. Изд-во иностр. лит., 1962.

RESOURCES ALLOCATION AS A TIME OPTIMAL PROBLEM

V. N. BURKOV

The problem of limited resources allocation for a complex of jobs from the point of view of optimal control theory is considered. The general statement of the problem is given and a particular case of independent jobs is presented. In this case one is able to obtain efficient solution algorithms for sufficiently general assumptions with respect to the variety of permissible controls V and the functions of speed of carrying out the jobs. The statements of some unsolved problems are given.
