

УДК 519.85+519.863  
ББК 22.19

## **ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ С ВАРИАНТАМИ АЛГОРИТМОВ ВНУТРЕННИХ ТОЧЕК НА НЕЛИНЕЙНЫХ ЗАДАЧАХ ПОТОКОРАСПРЕДЕЛЕНИЯ**

**Зоркальцев В. И.<sup>1</sup>, Медвежонков Д. С.<sup>2</sup>**  
(ФГБУН Институт систем энергетики  
им. Л.А. Мелентьева СО РАН, Иркутск)

*Представлены результаты сравнительных экспериментальных исследований вариантов алгоритмов внутренних точек на нелинейных моделях потокораспределения. В экспериментах выявлено преимущество линейных весовых коэффициентов, деленных на множители Лагранжа, перед квадратичными. Установлено, что при использовании двойственных алгоритмов требуемая точность решения достигается быстрее, чем при использовании прямых алгоритмов внутренних точек.*

Ключевые слова: прямые и двойственные алгоритмы внутренних точек, весовые коэффициенты, нелинейные задачи потокораспределения.

### **1. Введение**

В работах [7, 13, 14] рассматривается класс задач выпуклого программирования с линейными ограничениями и сепарабельной целевой функцией, обладающих свойством симметричной двойственности. Этим термином названа ситуация, когда двойственная к двойственной задаче оптимизации совпадает с

---

<sup>1</sup> Валерий Иванович Зоркальцев, доктор технических наук, профессор, заведующий лабораторией (zork@isem.sei.irk.ru).

<sup>2</sup> Дмитрий Сергеевич Медвежонков, младший научный сотрудник (dmitry@isem.sei.irk.ru).

исходной. Наличие симметричной двойственности даёт ряд преимуществ, в том числе возможность поиска решения исходной оптимизации на основе алгоритмов поиска оптимального решения двойственной задачи.

Основная цель данной статьи – изложить результаты сравнительных экспериментальных исследований вариантов алгоритмов внутренних точек для решения задач нелинейной оптимизации, обладающих свойством симметричной двойственности. Свойство симметричной двойственности позволяет использовать не только так называемые прямые алгоритмы внутренних точек (осуществляющих монотонное улучшение решения исходной задачи оптимизации), но и двойственные алгоритмы внутренних точек (осуществляющие монотонное улучшение решения двойственной задачи оптимизации).

В данной статье рассматривается класс алгоритмов внутренних точек, пионерами в разработке которых в конце 60-х и в 70-х годах прошлого века были отечественные ученые С.М. Анциз, И.И. Дикин, Ю.Г. Евтушенко, В.Г. Жадан, В.И. Зоркальцев [1, 4, 5, 6, 9]. В дальнейшем с 80-х годов прошлого века эти алгоритмы привлекли повышенное внимание во всём мире. Частными случаями рассматриваемых в этой статье алгоритмов являются широко известные алгоритмы под названием *affine scaling method* и *dual affine scaling method*.

К настоящему времени можно считать общепринятым, что в результате проведенных теоретических и многочисленных экспериментальных исследований установлена высокая вычислительная эффективность рассматриваемых алгоритмов внутренних точек. Например, они, как правило, позволяют быстрее получать решения задач линейного программирования разной размерности, чем хорошо отработанные в результате многолетней вычислительной практики алгоритмы симплекс-метода.

В силу конструктивных особенностей обсуждаемых алгоритмов (обуславливающих их вычислительную эффективность) для их теоретического обоснования нельзя воспользоваться стандартной техникой, изложенной, например, Зангвиллом [8]. К настоящему времени полученные результаты теоретического

обоснования алгоритмов внутренних точек рассматриваемого здесь типа относятся в основном к их использованию для решения задач линейного и отчасти квадратичного программирования. Хотя уже с 70-х годов прошлого века эти алгоритмы успешно применяются при реализации многих моделей энергетики, относящихся к другим классам задач математического программирования. В этой связи особый интерес представляют экспериментальные исследования алгоритмов. Как показывает практика предыдущих экспериментальных исследований, их результаты бывают полезны для выработки рекомендаций по использованию вариантов алгоритмов в конкретных типах задач и для последующего теоретического обоснования алгоритмов.

Здесь будет рассматриваться две альтернативы в выборе вариантов алгоритмов внутренних точек. Одна из них – правила задания так называемых «весовых коэффициентов», с помощью которых во вспомогательной задаче определения направления улучшения решения учитываются ограничения-неравенства исходной задачи оптимизации. В [10–12] на основе аксиоматического подхода вводятся и обсуждаются различные возможные правила задания этих коэффициентов, приводящих к алгоритмам, которые обладают разными свойствами. Наиболее известными являются так называемые квадратичные весовые коэффициенты, введенные в первых работах по данному типу алгоритмов внутренних точек И.И. Дикиным [4]. Результаты развития такого алгоритма и практические приложения отражены в монографии [5]. В дальнейшем в работах зарубежных авторов такой алгоритм внутренних точек получил название «affine scaling method».

У данного алгоритма есть одна нехорошая в вычислительном отношении особенность, проявляющаяся уже при решении задачи линейного программирования. При оптимизации в области допустимых решений шаг, с которым осуществляется улучшение решения по выбранному направлению, неограниченно возрастает. Это означает, что на финальных итерациях неизбежные малые погрешности в решении вспомогательной задачи выбора направления корректировки решения сильно проявляют-

ся в вычислительном процессе, в том числе через выход из области допустимых решений.

В целях борьбы с этим негативным явлением было предложено вместо квадратичных весовых коэффициентов на финальных итерациях использовать линейные весовые коэффициенты, деленные на оценки предыдущих итераций множителей Лагранжа соответствующих ограничений-неравенств [11]. Последующие экспериментальные исследования [2] показали, что таким вариантом алгоритмов внутренних точек вполне можно пользоваться не только на финальных, но и на всех итерациях. Теоретические исследования [12] показали, что эти алгоритмы обладают, так же как и алгоритмы с квадратичными весовыми коэффициентами, линейной скоростью сходимости, асимптотически не зависящей от исходных данных задачи, и могут обладать сверхлинейной скоростью сходимости. Практический интерес и интерес для дальнейших теоретических исследований представляет выяснение вопроса на основе экспериментальных расчетов: как соотносятся по времени счета эти два варианта алгоритмов на задачах с нелинейной целевой функцией?

Вторая альтернатива – выбор между «прямыми» и «двойственными» алгоритмами внутренних точек. Прямые алгоритмы осуществляют монотонное улучшение исходной задачи оптимизации в области решений, удовлетворяющих в строгой форме её ограничениям-неравенствам. На этапе ввода в область допустимых решений на каждой итерации происходит сокращение всех компонент вектора абсолютных значений невязок ограничений-равенств. На втором этапе осуществляется процесс оптимизации (монотонного улучшения значения целевой функции) в области допустимых решений исходной задачи.

При этом на каждой итерации при решении вспомогательной задачи выбора направления корректировки решения вырабатывается также приближение к решению двойственной задачи. Как было подмечено еще в 70-х годах И.И. Дикиным, приближения к решению двойственной задачи хотя и не монотонно в каком либо смысле, но значительно лучше (быстрее) сходятся к оптимальному решению двойственной задачи, чем

сходятся монотонно улучшаемые решения исходной задачи к её оптимальному решению. В дальнейшем этот факт получил теоретическое обоснование [10] на задачах ЛП.

Двойственный алгоритм внутренних точек осуществляет монотонное улучшение решения двойственной задачи. При этом на каждой итерации вырабатывается и приближение к решению исходной задачи. В этом случае, по аналогии с предыдущим, следует ожидать более быструю сходимость приближений исходной задачи к её оптимальному решению, чем сходимость приближений двойственной задачи к её оптимальному решению. Важно также отметить, что для рассматриваемых здесь задач легко получить стартовую точку двойственной задачи, удовлетворяющую всем её ограничениям-неравенствам в строгой форме. Оба эти момента позволяют надеяться, что двойственные алгоритмы могут быть более эффективными для определения с заданной точностью оптимального решения исходной задачи оптимизации.

Заметим, что у всех четырех рассматриваемых здесь алгоритмов объем вычислений на каждой итерации (при решении одной и той же задачи) одинаков. Это позволяет использовать число итераций для достижения решения с одной и той же заданной точностью в качестве сравнительного показателя времени счета.

Конкретным объектом экспериментальных исследований данной статьи будут нелинейные модели потокораспределения [3, 7, 13–16]. Этим термином объединяются нелинейные (с затратами на перевозки нелинейно зависящими от объемов перевозок) транспортные задачи, а также электрические цепи и гидравлические цепи (моделирующие транспортировку по трубопроводам воды, нефти или нефтепродуктов).

## **2. Взаимно-двойственные задачи оптимизации**

Заданы матрица  $A$  размера  $m \times n$  и множество номеров  $J = \{1, \dots, n\}$ . Множества  $L, H$  являются некоторыми подмноже-

ствами  $J$ . Также заданы: вектор  $b \in R^m$ , вектор  $c \in R^n$ , величины  $\underline{x}_j, j \in L$ , и  $\bar{x}_j, j \in H$ , при этом  $\underline{x}_j < \bar{x}_j$  для  $j \in L \cap H$ .

Обозначим  $Z$  множество функций из  $R$  в  $R$ , которые равны нулю в нуле и имеют непрерывные возрастающие производные. Причем производные этих функций равны нулю в нуле и изменяются от  $-\infty$  до  $+\infty$ .

Заметим, что любой функции из  $Z$  можно поставить в соответствие сопряженную функцию из  $Z$ , производная которой является обратной функцией к производной исходной функции. Такой переход к новой функции равносителен применению к исходной функции преобразования Лежандра–Фенхеля, известного из выпуклого анализа.

Пусть задан набор функций  $F_j(x_j), j \in J$ , принадлежащих множеству  $Z$ . Введем функции от векторов  $R^n$ :  $F(x) = \sum_{j \in J} F_j(x_j)$ .

Рассмотрим задачу выпуклой оптимизации с выпуклой сепарабельной целевой функцией и линейными ограничениями с вектором переменных  $x \in R^n$ :

- (1)  $F(x) + c^T x \rightarrow \min$
- (2)  $Ax = b$ ,
- (3)  $\underline{x}_j \leq x_j, j \in L$ ,
- (4)  $x_j \leq \bar{x}_j, j \in H$ .

Назовем (1)–(4) исходной задачей оптимизации. Расширенным решением исходной задачи назовем набор, состоящий из вектора исходных переменных  $x$ , вектора  $u \in R^m$  множителей Лагранжа ограничений (2), векторов  $l \in R^n$  и  $h \in R^n$ , содержащих множители Лагранжа ограничений (3) и (4), причем  $l_j = 0, j \in J \setminus L$  и  $h_j = 0, j \in J \setminus H$ , а также вектора  $y \in R^n$  с компонентами  $y_j = f_j(x_j)$ , где  $f_j$  – производная  $F_j, j \in J$ .

Обозначим  $\Phi_j$  функцию из  $Z$ , полученную в результате преобразования Лежандра–Фенхеля из функции  $F_j \in Z, j \in J$ . Введем функцию от вектора  $y \in R^n$ :  $\Phi(y) = \sum_{j \in J} \Phi_j(y_j)$ .

В [7] установлено, что двойственную задачу оптимизации к задаче (1)–(4) можно записать в виде

$$(5) \quad \Phi(y) - b^T u - \sum_{j \in L} \underline{x}_j l_j + \sum_{j \in H} \bar{x}_j h_j \rightarrow \min$$

$$(6) \quad y + c - l + h = A^T u,$$

$$(7) \quad l_j \geq 0, \quad j \in L; \quad h_j \geq 0, \quad j \in H,$$

$$(8) \quad l_j = 0, \quad j \in J, \quad j \notin L; \quad h_j = 0, \quad j \in J, \quad j \notin H.$$

Здесь переменными являются векторы  $y \in R^n$ ,  $u \in R^m$  и компоненты векторов  $l \in R^n$  и  $h \in R^n$  соответственно при  $j \in L$  и  $j \in H$ .

Расширенным решением двойственной задачи назовем набор, состоящий из векторов её переменных  $y$ ,  $u$ ,  $l$ ,  $h$ , а также вектора  $x \in R^n$  множителей Лагранжа ограничений (6).

Рассмотрим систему уравнений и неравенств, включающую условия (2)–(4), (8) и дополнительные ограничения:

$$(9) \quad y_j = f_j(x_j), \quad j \in J,$$

$$(10) \quad y + c - l + h = A^T u,$$

$$(11) \quad l_j = (f_j(x_j) + c_j - [A^T u]_j)_+, \quad j \in L,$$

$$(12) \quad h_j = ([A^T u]_j - f_j(\bar{x}_j) - c_j)_+, \quad j \in H.$$

Здесь  $f_j$  – производная функции  $F_j$ ,  $j \in J$ , а  $(\alpha)_+ = \max\{0, \alpha\}$ .

В [7] обосновывается справедливость утверждений следующей теоремы, которая говорит о возможности восстановить решение исходной задачи по решению двойственной. Это дает право пользоваться двойственными алгоритмами внутренних точек для решения задач потокораспределения.

**Теорема.** Если ограничения исходной задачи (1)–(4) совместны, то решение этой задачи существует и единственно, решение двойственной задачи (5)–(8) существует и единственно по вектору  $y$ , решение системы уравнений и неравенств (2)–(4), (8), (9)–(12) существует, единственно по векторам  $x$ ,  $y$  и совпадает с расширенными решениями исходной и двойственной задач. В противном случае решения исходной и двойственной задач не

существуют, система уравнений и неравенств (2)–(4), (8), (9)–(12) несовместна.

**Замечание.** Ограничения (11), (12) заменяют известные в оптимизации условия дополняющей нежесткости. Эти ограничения лучше в вычислительном отношении, чем билинейные условия дополняющей нежесткости.

### 3. Прямые алгоритмы внутренних точек

Итерационный процесс прямых алгоритмов внутренних точек решения исходной задачи (1)–(4), заключается в последовательном построении нового приближения:

$$(13) \quad x^{k+1} = x^k + \lambda_k s^k,$$

где  $s^k$  – направление корректировки текущего приближения,  $\lambda_k$  – величина шага вдоль этого направления. Процесс начинается из точки  $x^0$ , которая удовлетворяет ограничениям-неравенствам (3)–(4) в строгой форме. Считаем, что матрица  $A$  имеет полный ранг, целевая функция в (1) дважды дифференцируема. Обозначим  $f'_j(x_j)$  – вторую производную функции  $F_j(x_j)$ ,  $j \in J$ .

В этом алгоритме выделяются два этапа вычислений. Сначала осуществляется ввод в область допустимых решений, в процессе которого уменьшаются невязки ограничений-равенств (2). На втором этапе осуществляется оптимизация в области допустимых решений.

Алгоритм итеративно будет повторять перечисленный в пунктах 1–6 набор действий.

**Пункт 1.** Вычисление вектора невязки ограничений (2) для текущего  $k$ -го приближения  $x^k \in R^n$ :

$$(14) \quad r^k = b - Ax^k.$$

Вычислим величину

$$\pi_A^k = \max \{ |r_i^k| : i \in I \}.$$

Зададим  $\delta > 0$  – параметр алгоритма, характеризующий допустимую невязку ограничений-равенств (2). Если выполняется неравенство



$$(15) \pi_A^k < \delta,$$

то  $x^k$  принадлежит с требуемой точностью допустимой относительно ограничений-равенств (2) области. В этом случае алгоритм переходит на этап оптимизации в области допустимых решений. Если условие (15) не выполняется, то алгоритм остается на этапе ввода в область допустимых решений.

**Пункт 2.** Решение вспомогательной задачи поиска направления корректировки  $s^k$  текущего приближения  $x^k$ . В зависимости от этапа, на котором находится алгоритм, эта задача записывается по-разному. Возможны два случая.

1. Если алгоритм находится на этапе ввода в допустимую область, то решаем вспомогательную задачу относительно  $s$ :

$$(16) \frac{1}{2} \sum_{j \in J} \frac{(s_j)^2}{d_j^k} \rightarrow \min,$$

$$(17) As = r^k.$$

Величины  $d_j^k, j \in J$  – положительные весовые коэффициенты, изменяющиеся по итерациям. Используется два варианта их определения.

1) Квадратичные весовые коэффициенты:

$$(18) d_j^k = (x_j^k - \underline{x}_j)^2, \text{ если } j \in L \text{ и } j \notin H,$$

$$(19) d_j^k = (\bar{x}_j - x_j^k)^2, \text{ если } j \in H \text{ и } j \notin L,$$

$$(20) d_j^k = (\min(x_j^k - \underline{x}_j, \bar{x}_j - x_j^k))^2, \text{ если } j \in L \text{ и } j \in H.$$

2) Линейные весовые коэффициенты, деленные на приближения к множителям Лагранжа  $l_j^{k-1}, h_j^{k-1}$  ограничений (3), (4):

$$(21) d_j^k = (x_j^k - \underline{x}_j) / \max(\delta_2, l_j^{k-1}), \text{ если } j \in L \text{ и } j \notin H,$$

$$(22) d_j^k = (\bar{x}_j - x_j^k) / \max(\delta_2, h_j^{k-1}), \text{ если } j \in H \text{ и } j \notin L,$$

$$(23) d_j^k = \frac{\min(x_j^k - \underline{x}_j, \bar{x}_j - x_j^k)}{\max(\delta_2, l_j^{k-1}, h_j^{k-1})}, \text{ если } j \in L \text{ и } j \in H.$$

Здесь  $\delta_2$  – параметр, представленный некоторым малым числом и позволяющий избежать деления на ноль; параметры  $l_j^{k-1}, h_j^{k-1}$  – это двойственные оценки, вычисляемые на предыду-

щей итерации алгоритма (если алгоритм на первой итерации, то положим  $l_j^0 = 0, h_j^0 = 0$ ).

При  $j \notin L \cup H$  значение коэффициентов  $d_j^k$  приравняется числу  $\omega^k = \max\{\omega_1, \omega_2, \omega_3\}$ , где  $\omega_1 = \max\{x_j^k - \underline{x}_j : j \in L \setminus H\}$ ,  $\omega_2 = \max\{\bar{x}_j - x_j^k : j \in H \setminus L\}$ ,  $\omega_3 = \max\{\min\{x_j^k - \underline{x}_j, \bar{x}_j - x_j^k\} : j \in L \cap H\}$ .

Отметим, что на этом этапе согласно (13), (14), (17) выполняется соотношение

$$(24) \quad r^{k+1} = (1 - \lambda_k)r^k,$$

поэтому на каждой итерации происходит сокращение абсолютных значений всех компонент вектора невязок балансовых ограничений  $r^k$ , если  $\lambda_k$  ограничено сверху единицей.

Найдем решение задачи (16), (17), используя правило множителей Лагранжа. Выразим компоненты  $s$  из условий оптимальности этой задачи:

$$(25) \quad s_j = d_j^k [A^T u]_j, \quad j \in J,$$

здесь  $u \in R^m$  – вектор множителей Лагранжа ограничений (17).

Обозначим  $D^k$  – диагональную матрицу с элементами  $d_j^k$  на главной диагонали. Подставим полученные в (25) выражения для  $s$  в систему уравнений (17). Получаем систему линейных уравнений относительно вектора  $u$ :

$$(26) \quad AD^k A^T u = r^k.$$

Матрица  $AD^k A^T$  – симметричная, положительно определенная. Решив систему (26), получим вектор  $u^k$ . Затем вычислим компоненты вектора  $s^k$ , используя правило (25).

II. Если алгоритм находится на этапе оптимизации в области допустимых решений, решаем вспомогательную задачу поиска направления корректировки текущего приближения в виде

$$(27) \quad \sum_{j \in J} (f_j(x_j^k) + c_j) s_j + \frac{1}{2} \sum_{j \in J} f_j'(x_j^k) (s_j)^2 + \frac{1}{2} \sum_{j \in J} \frac{(s_j)^2}{d_j^k} \rightarrow \min,$$

$$(28) \quad As = 0.$$

Здесь величины  $d_j^k, j \in J$  вычисляются так же, как и на этапе ввода в область допустимых решений.

Найдем решение задачи (27), (28), используя правило множителей Лагранжа. Выразим компоненты вектора  $s$ :

$$(29) \quad s_j = \frac{(A^T u)_j - f_j(x_j^k) - c_j}{f_j'(x_j^k) + (d_j^k)^{-1}}, \quad j \in J,$$

Используем обозначения:  $G^k$  – диагональная матрица с  $n$  элементами вида  $(f_j'(x_j^k) + (d_j^k)^{-1})^{-1}$  на главной диагонали,  $f(x^k)$  – вектор с элементами  $f_j(x_j^k)$ ,  $j \in J$ . Подставим полученные в (29) выражения для компонент вектора  $s$  в систему уравнений (28). Получим систему линейных уравнений относительно вектора  $u$ :

$$(30) \quad AG^k(A^T u - f(x^k) - c) = 0.$$

Решив систему (30), получим вектор  $u^k$ . Затем вычислим  $s^k$ , используя (29).

**Пункт 3.** Проверка выполнения критерия останова. Проверим для текущего приближения к решению задачи (1)–(4) выполнение условий (2)–(4), (8), (9)–(12). Рассчитаем величины  $l_j^k$ ,  $h_j^k$  по формулам (11), (12). Далее рассчитаем компоненты вектора  $y^k$  по формулам:

$$(31) \quad y_j^k = [A^T u^k]_j - c_j, \quad j \in J \setminus (L \cup H),$$

$$(32) \quad y_j^k = [A^T u^k]_j - c_j + l_j^k - h_j^k, \quad j \in L \cup H.$$

Рассчитаем величину

$$(33) \quad \pi_Y^k = \max \{ |y_j^k - f_j(x_j^k)| : j \in J \}.$$

Если справедливо условие  $\max \{ \pi_Y^k, \pi_A^k \} < \varepsilon$ , где  $\varepsilon$  – параметр допустимой погрешности решения, то найдено решение задачи (1)–(4) с требуемой точностью. В этом случае завершаем работу алгоритма, иначе переходим к следующему пункту.

**Пункт 4.** Определение шага корректировки решения. Вычислим шаг до ближайшей границы допустимой области, задаваемой ограничениями-неравенствами, по формуле:

$$(34) \quad \lambda_{LH} = \min(\lambda_L, \lambda_H),$$

$$\text{где } \lambda_L = \min \{ (\underline{x}_j - x_j^k)(s_j^k)^{-1} : j \in L, s_j < 0 \},$$

$$\lambda_H = \min \{ (\bar{x}_j - x_j^k)(s_j^k)^{-1} : j \in H, s_j > 0 \}.$$

Выбор итогового шага  $\lambda_k$  корректировки решения зависит от этапа, на котором находится алгоритм.

1) Если алгоритм находится на этапе ввода в область допустимых решений, то итоговый шаг  $\lambda_k$  вычислим по правилу:

$$\lambda_k = \min(\gamma\lambda_{LH}, 1), \text{ где } 0 < \gamma < 1 \text{ (например, } \gamma = 0,7 \text{)}.$$

Здесь  $\gamma$  – параметр метода, который используется, чтобы после корректировки приближение к решению оставалось внутри допустимой по ограничениям-неравенствами области.

2) Если алгоритм находится на этапе оптимизации в области допустимых решений, то сначала найдем величину  $\lambda_p$ , решив задачу одномерной минимизации целевой функции (1) из точки  $x^k$  по направлению  $s^k$ :

$$(35) \lambda_p = \arg \min_{\alpha \in R} \{F(x^k + \alpha s^k) + c^T(x^k + \alpha s^k) : 0 < \alpha \leq \lambda_{LH}\}.$$

А затем вычислим итоговый шаг  $\lambda_k$  корректировки решения:

$$(36) \lambda_k = \min(\gamma\lambda_{LH}, \lambda_p).$$

**Пункт 5.** *Вычисление следующего приближения.* Осуществим итеративный переход, используя найденные  $s^k$  и  $\lambda_k$ :

$$x^{k+1} = x^k + \lambda_k s^k.$$

#### 4. Двойственные алгоритмы внутренних точек

Двойственный алгоритм решает двойственную задачу (5)–(8). Он отличается от прямого алгоритма тем, что для него несложно априори сформировать допустимое по ограничениям-равенствам (6) решение, для которого все ограничения-неравенства (7) выполняются в строгой форме. Поэтому одним из преимуществ рассматриваемых двойственных алгоритмов внутренних точек является то, что в них сразу со стартовой точки начинается процесс оптимизации в области допустимых решений.

Для реализации алгоритма необходимо, чтобы целевая функция в (5) была дважды дифференцируемой. Обозначим  $\varphi_j(y_j)$  – вторую производную функции  $\Phi_j(y_j)$ ,  $j \in J$ .

Зададим начальное приближение искомым векторов  $y^0 \in R^n$ ,  $u^0 \in R^m$ ,  $l^0 \in R^n$  и  $h^0 \in R^n$  так, чтобы векторы начального приближения удовлетворяли равенствам (6), (8) и неравенствам (7) в строгой форме. Таким образом, алгоритм будет работать на одном этапе – этапе оптимизации в области допустимых решений.

**Пункт 1.** Поиск направления корректировки текущего приближения. Найдем векторы  $\Delta y^k$ ,  $\Delta u^k$ ,  $\Delta l^k$ ,  $\Delta h^k$ , являющиеся решением задачи

$$(37) \quad \sum_{j \in J} \varphi_j(y_j^k) \Delta y_j + \frac{1}{2} \sum_{j \in J} \varphi_j'(y_j^k) (\Delta y_j)^2 - \sum_{i \in I} b_i \Delta u_i - \sum_{j \in L} \underline{x}_j \Delta l_j + \\ + \sum_{j \in H} \bar{x}_j \Delta h_j + \frac{1}{2} \sum_{j \in L} \frac{(\Delta l_j)^2}{q_j^k} + \frac{1}{2} \sum_{j \in H} \frac{(\Delta h_j)^2}{p_j^k} \rightarrow \min,$$

$$(38) \quad \Delta y + c - A^T \Delta u - \Delta l + \Delta h = 0,$$

$$(39) \quad \Delta l_j = 0, \quad j \in J \setminus L, \quad \Delta h_j = 0, \quad j \in J \setminus H.$$

Для вычисления знаменателей  $q_j^k$  и  $p_j^k$  функций штрафа в целевой функции (37) использовались два способа.

1) Квадратичные весовые коэффициенты:

$$(40) \quad q_j^k = (l_j^k)^2, \quad j \in L; \quad p_j^k = (h_j^k)^2, \quad j \in H.$$

2) Линейные весовые коэффициенты, деленные на множители Лагранжа:

$$(41) \quad q_j^k = l_j^k / \xi_j^{k-1}, \quad j \in L; \quad p_j^k = h_j^k / \psi_j^{k-1}, \quad j \in H.$$

Здесь величины  $\xi_j^{k-1}$ ,  $\psi_j^{k-1}$  являются приближениями к множителям Лагранжа ограничений-неравенств (7) задачи (5)–(8). Причем,  $\xi_j^k = x_j^{k-1} - \underline{x}_j$ ,  $j \in L$ , а  $\psi_j^k = \bar{x}_j - x_j^{k-1}$ ,  $j \in H$ . Величина  $x_j^{k-1}$  – это компонента вектора множителей Лагранжа  $x^{k-1}$  ограничений-равенств (6). Вектор  $x^{k-1}$  вычисляется на предыдущей итерации алгоритма (для первой итерации алгоритма зададим  $\xi_j^0=1$ ,  $\psi_j^0=1$ ).

Найдем решение задачи (37)–(39) используя правило множителей Лагранжа. Выразим компоненты векторов  $\Delta y$ ,  $\Delta l$ ,  $\Delta h$  из

системы, полученной приравниванием нулю производных функции Лагранжа:

$$(42) \Delta y_j = (x_j - \varphi_j(y_j^k)) / \varphi_j'(y_j^k), \quad j \in J,$$

$$(43) \Delta l_j = (\underline{x}_j - x_j) q_j^k, \quad j \in L,$$

$$(44) \Delta h_j = (x_j - \bar{x}_j) p_j^k, \quad j \in H.$$

Подставим полученные выражения в (38), выразим из получившихся уравнений вектор двойственных оценок  $x$ :

$$(45) x = H^k (A^T \Delta u - \Pi^k).$$

Здесь  $H^k$  – диагональная матрица с  $n$  элементами вида  $((\varphi_j'(y_j^k))^{-1} + q_j^k + p_j^k)^{-1}$  на главной диагонали;  $\Pi^k$  – диагональная матрица с  $n$  элементами вида  $(c_j - \varphi_j'(x_j^k)(\varphi_j'(x_j^k))^{-1} - \underline{x}_j q_j^k + \bar{x}_j p_j^k)$  на главной диагонали. Для элементов матриц  $H^k$  и  $\Pi^k$  справедливы соотношения:

$$(46) q_j^k = 0, \quad j \in J \setminus L; \quad p_j^k = 0, \quad j \in J \setminus H.$$

Подставим выражение (45) для  $x$  в уравнение (2). Получим систему линейных уравнений относительно вектора  $\Delta u$ :

$$(47) AH^k(A^T \Delta u - \Pi^k) = b.$$

Считаем, что матрица  $A$  имеет полный ранг. Тогда матрица  $AH^kA^T$  – симметричная, положительно определенная. Решив систему (47), получим вектор  $\Delta u^k$ . Затем найдем векторы  $\Delta y^k$ ,  $\Delta l^k$ ,  $\Delta h^k$ , используя (39), (42)–(44), и вектор двойственных переменных  $x^k$ , используя (45).

**Пункт 2. Проверка выполнения критерия останова.** Проверим для текущего приближения к решению выполнение условий (2)–(4), (8), (9)–(12). Рассчитаем следующие величины:

$$\pi_{xY}^k = \max \{ |\varphi_j(y_j^k) - x_j^k| : j \in J \},$$

$$\pi_L^k = \max \{ (f_j(\underline{x}_j) + c_j - [A^T u^k]_j)_+ - l_j^k : j \in L \},$$

$$\pi_H^k = \max \{ ([A^T u^k]_j - f_j(\bar{x}_j) - c_j)_+ - h_j^k : j \in H \},$$

$$\pi_{xL}^k = \max \{ (\underline{x}_j - x_j^k)_+ : j \in L \}, \quad \pi_{xH}^k = \max \{ (x_j^k - \bar{x}_j)_+ : j \in H \}.$$

Если справедливо условие  $\max\{\pi_{XY}^k, \pi_L^k, \pi_H^k, \pi_{xL}^k, \pi_{xH}^k\} < \varepsilon$ , где  $\varepsilon$  – параметр допустимой погрешности вычислений, то условия оптимальности решаемой задачи оптимизации выполнены с требуемой точностью. В этом случае завершаем работу алгоритма, иначе переходим к пункту 3.

**Пункт 3.** *Выбор шага корректировки текущего приближения.* Сначала определим шаг корректировки до ближайшей границы допустимой области, задаваемой ограничениями-неравенствами. Шаг до ближайшей границы определяется по формуле  $\lambda_{LH} = \min\{\lambda_L, \lambda_H\}$ , где

$$\lambda_L = \min\left\{(-l_j^k)/\Delta l_j^k : j \in L, \Delta l_j^k < 0\right\},$$

$$\lambda_H = \min\left\{(-h_j^k)/\Delta h_j^k : j \in H, \Delta h_j^k < 0\right\}.$$

Найдем величину  $\lambda_p$ , решив задачу одномерной минимизации целевой функции (5) по направлению, задаваемому векторами  $\Delta y^k, \Delta u^k, \Delta l^k, \Delta h^k$ :

$$(48) \quad \lambda_p = \arg \min_{\alpha \in R} \{\Phi(y^k + \alpha \Delta y^k) - b^T(u^k + \alpha \Delta u^k) - \sum_{j \in L} \bar{x}_j(l_j^k + \alpha \Delta l_j^k) + \sum_{j \in H} \bar{x}_j(h_j^k + \alpha \Delta h_j^k) : 0 < \alpha \leq \lambda_{LH}\}.$$

Вычислим итоговый шаг  $\lambda_k$  корректировки решения:

$$(49) \quad \lambda_k = \min(\gamma \lambda_{LH}, \lambda_p), \text{ где } 0 < \gamma < 1, \text{ (например, } \gamma = 0,7).$$

**Пункт 4.** Вычисление следующего приближения. Осуществим итеративный переход, используя шаг корректировки  $\lambda_k$ :

$$y^{k+1} = y^k + \lambda_k \Delta y^k; \quad u^{k+1} = u^k + \lambda_k \Delta u^k;$$

$$l_j^{k+1} = l_j^k + \lambda_k \Delta l_j^k, \quad j \in L; \quad l_j^k = 0, \quad j \in J \setminus L;$$

$$h_j^{k+1} = h_j^k + \lambda_k \Delta h_j^k, \quad j \in H; \quad h_j^k = 0, \quad j \in J \setminus H.$$

## 5. Результаты экспериментальных исследований

Была выполнена программная реализация прямых и двойственных алгоритмов внутренних точек для решения задач потокораспределения на языке C++.

В экспериментальных расчетах использовались два варианта прямого и два варианта двойственного алгоритмов внутренних точек с двумя видами весовых коэффициентов: квадратичными и линейными коэффициентами, деленными на приближения к множителям Лагранжа. В таблице 1 приводятся характеристики решенных в ходе эксперимента задач и результаты расчетов для них. Критерием остановки алгоритмов было выполнение условий (2)–(4), (8), (9)–(12) с максимальной невязкой не более 0,1.

Таблица 1. Результаты расчетов задач потокораспределения с использованием четырех вариантов алгоритмов

Характеристики задач				Количество итераций для вариантов алгоритмов внутренних точек			
Узлов	Ветвей	Двуст. ограничен.	Активн. огранич.	Прямой	Двойств	Прямой	Двойств
				Квадратичн. весовые коэффиц.		Линейные весовые коэффиц.	
25	39	25	3	30	47	24	16
25	39	35	9	63	42	38	20
25	48	40	4	25	24	15	15
25	48	40	11	112	46	60	20
50	60	25	7	47	35	37	23
50	60	50	8	83	41	74	24
50	136	88	39	118	64	25	31
50	136	100	19	70	34	25	21
100	116	20	10	64	15	42	24
100	116	81	7	70	39	24	22
100	195	79	37	65	250	32	27
100	195	90	15	44	28	32	26
200	300	150	10	56	31	26	18
200	300	150	26	121	39	39	23
338	712	500	11	108	86	27	23
338	712	500	16	96	79	39	31
Среднее геометрическое:				66,7	44,4	32,5	22,3

Расчеты показывают, что прямой и двойственный алгоритмы с линейными весовыми коэффициентами в среднем в два раза быстрее (по числу итераций) своих аналогов с квадратич-



ными коэффициентами. Двойственные алгоритмы в среднем в полтора раза быстрее своих прямых аналогов.

Кроме того, был проведен эксперимент для сравнения скорости достижения требуемой точности решения прямым и двойственным алгоритмом по исходным и по двойственным переменным. Точность проверялась относительно предварительно найденного с большой точностью решения. В таблице 2 приводятся результаты расчетов по выборке из 6 задач, для которых условия (2)–(4), (8), (9)–(12) выполнялись с точностью 0,01 на последней итерации.

Таблица 2. Сравнение скорости достижения требуемой точности решения по исходным и по двойственным переменным

	Расчеты прямым алгоритмом			Расчеты двойственным алгоритмом		
	Кол-во итерационных рац.	Точность решения по исходным перемен.	Точность решения по двойствен. перемен.	Кол-во итерационных рац.	Точность решения по исходным перемен.	Точность решения по двойствен. перемен.
25, 39	29	8,04E-06	2,85E-08	20	2,85E-08	1,20E-03
25, 48	29	1,04E-05	7,91E-07	21	2,90E-08	2,60E-03
50, 60	34	7,95E-05	3,11E-07	24	1,27E-08	1,59E-03
50, 136	32	2,26E-05	1,83E-07	25	4,21E-06	1,07E-03
100, 116	31	1,51E-05	4,70E-07	32	1,26E-09	1,53E-03
100, 195	30	4,05E-06	4,19E-08	24	2,00E-06	1,81E-03

Из таблицы 2 следует, что при использовании прямого алгоритма требуемая точность решения достигается быстрее по двойственным переменным, чем по исходным. Для двойственного алгоритма требуемая точность достигается быстрее по исходным переменным, чем по двойственным.

## 5. Заключение

Проведены экспериментальные исследования прямых и двойственных алгоритмов внутренних точек с различными способами выбора весовых коэффициентов на нелинейных задачах потокораспределения.

Показано, что варианты прямых и двойственных алгоритмов с линейными весовыми коэффициентами, деленными на множители Лагранжа, в среднем в два раза быстрее по числу итераций, чем варианты алгоритмов с квадратичными весовыми коэффициентами. Установлено, что при использовании двойственных алгоритмов требуемая точность решения достигается в среднем в полтора раза быстрее, чем при использовании прямых алгоритмов.

Сделан вывод, что при использовании прямого алгоритма требуемая точность решения достигается быстрее по двойственным переменным, чем по исходным. Для двойственного алгоритма требуемая точность достигается быстрее по исходным переменным, чем по двойственным.

В связи с результатами экспериментов справедлива следующая рекомендация: для более быстрого нахождения решения исходной задачи с заданной точностью целесообразно использовать двойственные алгоритмы внутренних точек с линейными весовыми коэффициентами, деленными на множители Лагранжа.

### **Литература**

1. АНЦЫЗ С.М., ДИКИН И.И. *Об одном численном методе решения задачи линейного программирования и некоторых её обобщений* // В кн.: Управляемые системы. Вып. 3. – Новосибирск, ИМ СО АН СССР, 1969. – С. 54–56.
2. ВОЙТОВ О.Н., ЗОРКАЛЬЦЕВ В.И., ФИЛАТОВ А.Ю. *Определение допустимых режимов электроэнергетических систем алгоритмами внутренних точек* // Сибирский журнал индустриальной математики – 2000. – Том 3, №1(5). – С. 57–65.
3. ДЕННИС Дж.Б. *Математическое программирование и электрические цепи*. – М.: Изд-во иностранной литературы, 1961, – 216 с.
4. ДИКИН И.И. *Итеративное решение задач линейного программирования* // Доклады АН СССР. – 1967. – Т. 174. – С 747–748.

5. ДИКИН И.И., ЗОРКАЛЬЦЕВ В.И. *Итеративное решение задач математич. программирования (алгоритмы метода внутренних точек)*. – Новосибирск: Наука, 1980. – 144 с.
6. ЕВТУШЕНКО Ю.Г., ЖАДАН В.Г. *Релаксационный метод решения задач нелинейного программирования* // Журнал вычислительной математики и математической физики. – 1977. – Том 17, №4. – С. 890–904.
7. ЕПИФАНОВ С.П., ЗОРКАЛЬЦЕВ В.И., МЕДВЕЖОНКОВ Д.С. *Модель гидравлической сети с регуляторами расхода* // Управление большими системами. – 2010. – Специальный выпуск 30.1 «Сетевые модели в управлении». – С. 286–299.
8. ЗАНГВИЛЛ У. *Нелинейное программирование. Единый подход*. – Пер. с англ. под ред. Е.Г. Гольштейна. – М.: «Сов. радио», 1973. – 312 с.
9. ЗОРКАЛЬЦЕВ В.И. *Итеративный алгоритм решения задачи линейного программирования* // В кн.: Алгоритмы и программы решения задач линейной алгебры и мат. программирования. – Иркутск, СЭИ СО АН СССР, 1978. – С. 77–89.
10. ЗОРКАЛЬЦЕВ В.И. *Класс алгоритмов внутренних точек* // Журнал вычислительной математики и математической физики. – 2009. – №12. – С. 3–28.
11. ЗОРКАЛЬЦЕВ В.И. *Методы прогнозирования и анализа эффективности функционирования системы топливоснабжения*. – М.: Наука, 1988. – 144 с.
12. ЗОРКАЛЬЦЕВ В.И. *Проективные алгоритмы оптимизации, использующие множители предыдущей итерации* // Журнал вычисл. Математики и матем. физики. – 1994. – Т. 34, №7. – С. 943–950.
13. ЗОРКАЛЬЦЕВ В.И. *Симметричная двойственность в оптимизации и ее приложения* // Известия высших учебных заведений. Математика, 2006. – №2. – С. 53–59.
14. ЗОРКАЛЬЦЕВ В.И., ХАМИСОВ О.В. *Равновесные модели в экономике и энергетике*. – Новосибирск: Наука, 2006. – 221 с.

15. BERTSEKAS D.P. *Network Optimization: Continuous and Discrete Models*. – Athena Scientific, Belmont, Massachusetts, 1998. – 608 p.
16. ROCKAFELLAR R. *Tyrrell: Network flows and monotropic optimization*. – Pure and Applied Mathematics. – New York: Wiley-Interscience, 1984. – 616 p.

## COMPUTATIONAL EXPERIMENTS WITH VARIANTS OF INTERIOR-POINT ALGORITHMS FOR NONLINEAR FLOW DISTRIBUTION PROBLEMS

**Dmitry Medvezhonkov**, Institute of Energy Systems of SB RAS, Irkutsk, junior research assistant (dmitry@isem.sei.irk.ru).

**Valery Zorkaltsev**, Institute of Energy Systems of SB RAS, Irkutsk, Doctor of Science, prof., head of laboratory (zork@isem.sei.irk.ru).

*Abstract: We present results of computational experiments which compare variants of primal and dual interior-point algorithms for nonlinear problems of flow distribution. The experiments show that linear weight coefficients divided by Lagrange multipliers dominate quadratic weight coefficients. We also show that the required solution accuracy is achieved faster when using dual algorithms, rather than primal ones.*

Keywords: primal and dual interior-point algorithms, weight coefficients, nonlinear flow distribution problems.

*Статья представлена к публикации членом редакционной коллегии М.В. Губко*

*Поступила в редакцию 18.04.2013.*

*Опубликована 30.11.2013.*