ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь—июнь 2002. Серия 2. Том 9, № 1, 78–98

УДК 519.83:330.115

НОВЫЕ АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ СО СПЕЦИАЛЬНОЙ СТРУКТУРОЙ*)

М. В. Пудова

Рассматривается построение эффективных вычислительных схем для решения задач линейного программирования в ситуации, когда матрица ограничений имеет узкоблочную с окаймлением структуру. Описаны модификации четырех итерационных алгоритмов и приведены результаты сравнения трудоемкости предложенных вычислительных схем.

Введение

При нахождении оптимальных стратегий функционирования иерархических систем часто возникают задачи линейного программирования (ЗЛП) специального вида, в которых матрицы ограничений имеют узкоблочную с окаймлением структуру. В таких задачах могут содержаться несколько тысяч или десятков тысяч ограничений и переменных. Кроме того, для исследования одной иерархической системы эти задачи нужно решать неоднократно из-за частой корректировки исходных данных. Поэтому актуальным является вопрос о построении эффективных вычислительных схем для решения таких задач. В зарубежной литературе [4, 8] и в отечественных публикациях [7, 10, 12] (см. также [9, 11]) предложены различные модификации симплекс-метода и метода последовательного улучшения плана для решения ЗЛП с узкоблочной с окаймлением структурой. С тех пор появились новые методы решения ЗЛП с полиномиальной временной сложностью. Представляется интересным построение алгоритмов, учитывающих структуру решаемых задач, на основе новых методов.

Наилучшую теоретическую оценку временной сложности решения ЗЛП имеет алгоритм, предложенный Н. Кармаркаром [16]. Результаты

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 00–01–00804 и 00–15–9884) и Федеральной целевой программы «Интеграция» (проект 274).

апробации, описанные в [13, 19], подтвердили его эффективность. В [1] для решения ЗЛП, матрицы ограничений которых имеют узкоблочную с окаймлением структуру, предложены две модификации метода Кармаркара и метода, изложенного в [2], в котором допустимая область аппроксимируется симплексами. В [15, 18] были рассмотрены модификации метода Кармаркара для решения ЗЛП большой размерности с некоторой специальной структурой.

И. И. Дикин показал, что формулы, реализующие модификацию метода Кармаркара для ЗЛП в каноническом виде [14], практически совпадают с формулами, реализующими метод из [5]. Приведенные в [5] численные эксперименты указывают на то, что эти методы близки по временной сложности. Поэтому в [1] были также построены модификации метода Дикина для ЗЛП со специальной структурой.

Возникает проблема: какую из предложенных вычислительных схем рекомендовать для программной реализации. Очевидно, что лучше пользоваться тем алгоритмом, который наиболее эффективно учитывает специальную структуру ограничений ЗЛП. Теоретический анализ показал, что в этом смысле среди предложенных схем наилучшей является схема, основанная на методе Дикина. Убедимся в этом.

Рассматривается задача математического программирования (ЗМП) в следующей форме:
найти минимум линейной формы

$$\sum_{j=1}^{N} c_j x_j \tag{1}$$

или квадратичного функционала

$$H(x) = \sum_{i=1}^{M} \left(\sum_{j=1}^{N} a_{ij} x_j - b_i \right)^2 \tag{2}$$

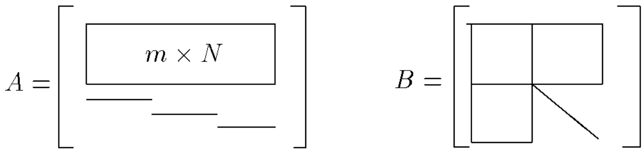
при условиях

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad 1 \leqslant j \leqslant N \tag{3}$$

и

$$Ax = b. \tag{4}$$

Здесь A — матрица размера $M \times N$, а b есть M-мерный вектор, x, c, α и β суть N-мерные векторы. Предполагается, что матрица A имеет узкоблочную с окаймлением структуру: первые m строк есть матрица размера $m \times N$, а строки с $(m+1)$-й по M-ю — последовательность R блоков. Размеры блоков равны $1 \times n_r$, $r = 1, \ldots, R$. При этом предполагается, что матрица A имеет большую размерность. Описанные матрицы изображены на рисунке.



а *б*

Матрицы со специальной структурой:
а — узкоблочная матрица A; *б* — матрица $B = ADA^T$

Теперь рассмотрим новые алгоритмы, построенные на базе трех методов с полиномиальной временной сложностью и метода Дикина, которые предназначены для решения этих задач.

1. Алгоритм, основанный на методе симплексных погружений

Аналогично [2] вычислительную схему будем строить для решения задачи (2), (3). Пусть в R^N задан N-мерный симплекс S_N^k, определяемый матрицей

$$X^k = \begin{pmatrix} x_{11}^k & . & . & . & x_{1N}^k \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{N+1,1}^k & . & . & . & x_{N+1,N}^k \end{pmatrix}$$

и содержащий точку минимума x^* задачи (2), (3). Строки матрицы X^k соответствуют вершинам симплекса S_N^k, где k — номер итерации. Пусть V^k — объем симплекса S_N^k. Опишем шаг итеративного процесса.

1. Находится центр тяжести $\overline{x}^k$ симплекса S_N^k по формуле

$$\overline{x}_j^k = \frac{1}{N+1} \sum_{i=1}^{N+1} x_{ij}^k, \quad j = 1, \ldots, N.$$

2. Определяется вектор нормали $g^k = \nabla H(\overline{x}^k)$ плоскости, отсекающей часть симплекса S_N^k: $g^k(x - \overline{x}^k) \leqslant 0$. Получается усеченный симплекс

$$\widetilde{S^k} = \{x \in S_N^k \mid g^k(x - \overline{x}^k) \leqslant 0\}.$$

Симплекс $\widetilde{S^k}$ погружается в симплекс S_N^{k+1} минимального объема. Для этого

3. Находятся числа $\gamma_i^k = g^k(x_i^k - \overline{x}^k)$, $i = 1, \ldots, N+1$, где x_i^k — i-я строка матрицы X^k. Пусть $I^- = \{i \mid \gamma_i^k < 0\}$, $I^+ = \{i \mid \gamma_i^k > 0\}$.

4. Определяется индекс q из условия $\gamma_q^k = \min\limits_{i \in I^-} \gamma_i^k$. Очевидно, что $\gamma_q^k < 0$.

5. На отрезке $0 \leqslant t \leqslant -\frac{1}{\gamma_q^k}$ находится точка минимума t^* функции

$$\Pi(t) = \prod_{1 \leqslant i \leqslant N+1, i \neq q} (1 + \gamma_i^k t).$$

Пусть $q_k = \frac{1}{\Pi(t^*)}$, следовательно, $q_k < 1$.

6. Вычисляется $v_i^* = 1 + t^* \gamma_i^k$, $\tau_i^* = \frac{1}{v_i^*}$, $i = 1, \ldots, N+1$, $i \neq q$.

7. Вершины симплекса S_N^k пересчитываются по формуле

$$x_i^{k+1} = x_i^k + \tau_i^*(x_i^k - x_q^k), \quad i = 1, \ldots, N+1, \quad i \neq q, \quad x_q^{k+1} = x_q^k.$$

Новая матрица X^{k+1} будет соответствовать новому симплексу S_N^{k+1} минимального объема, содержащему усеченный симплекс $\tilde{S}^k$.

8. Подсчитывается объем симплекса по формуле $V^{k+1} = V^k q_k$.

9. Если $V^k < \varepsilon$, где ε — заданная точность, то процесс заканчивается. В противном случае k заменяется на $k+1$ и осуществляется переход к п. 1.

В [2] доказано, что в этой схеме число итераций полиномиально зависит от N.

Наиболее трудоемкой операцией, позволяющей использовать структуру матрицы A при реализации этой схемы, является нахождение градиента целевой функции, определяемого формулой

$$\nabla_j H(x) = 2 \sum_{i=1}^{M} a_{ij} \sum_{l=1}^{N} (a_{il} x_l^k - b_i), \quad j = 1, \ldots, N. \tag{5}$$

В ситуации, когда матрица A является узкоблочной, полагаем

$$\delta_i = \begin{cases} \sum\limits_{l=1}^{N} (a_{il} x_l^k - b_i), & \text{если } i \leqslant m; \\ \sum\limits_{l=s_r+1}^{s_r+n_r} (a_{il} x_l^k - b_i), & \text{если } i = m + r, \text{ здесь } s_r = \sum\limits_{l=1}^{r-1} n_l. \end{cases} \tag{6}$$

Теперь для нахождения градиента вычисления выполняются по формуле

$$\nabla_j H(x) = 2 \left[\sum_{i=1}^{m} a_{ij} \delta_i + a_{rj} \delta_r \right], \tag{7}$$

где r определяется из условия $\sum\limits_{l=1}^{r-1} n_l + 1 \leqslant j \leqslant \sum\limits_{l=1}^{r} n_l$.

Таким образом, отличие алгоритма решения ЗМП с узкоблочной матрицей ограничений, основанного на методе симплексных погружений, от алгоритма, реализующего метод симплексных погружений для задач с матрицей общего вида, состоит в том, что градиент целевой

функции вычисляется по формулам (6)–(7). Все остальные действия остаются без изменений.

Для того чтобы сравнить эффективность предлагаемых ниже алгоритмов, введем следующий критерий. Величину, равную отношению трудоемкости алгоритма, модифицированного для решения ЗМП со специальной структурой матрицы, к трудоемкости алгоритма, предназначенного для решения ЗМП общего вида, назовем *коэффициентом уменьшения трудоемкости* (КУТ) модифицированного алгоритма. Трудоемкостью алгоритма назовем число элементарных операций, необходимых для нахождения оптимального решения ЗМП. Коэффициент уменьшения трудоемкости будем обозначать через Ω_A, где A идентифицирует метод, на основе которого построен новый алгоритм: $A = \mathrm{C}$ для метода симплексных погружений, K — Кармаркара, D — Дикина и R — Ринальди. Очевидно, что $\Omega_A < 1$. Докажем теперь следующее утверждение.

Лемма 1. *Коэффициент уменьшения трудоемкости* Ω_C *не меньше* $O(m/M)$.

Доказательство. Заметим, что число итераций для реализации метода симплексных погружений при решении задачи (2)–(3) со специальной структурой совпадает с числом итераций, необходимых для решения аналогичной задачи общего вида. Поэтому число операций для реализации метода уменьшается за счет более экономного осуществления итерации. Если матрица A имеет общий вид, то для нахождения градиента по формуле (5) требуется $O(NM)$ операций. Из приведенной выше схемы легко определить, что остальные действия для выполнения одной итерации можно проделать за $O(N^2)$ операций. Из (6) и (7) следует, что число операций для нахождения градиента в узкоблочном случае равно $O(Nm + n^r(M-m))$, где $n^r = \max\limits_{1\leqslant i\leqslant R}\{n_i\}$.

Трудоемкость остальных действий для выполнения итерации совпадает с общим случаем.

Оценим теперь величину Ω_C. Имеем

$$\Omega_C = O\Big(\frac{N^2 + Nm + n^r(M-m)}{N^2 + NM}\Big)$$
$$= O\Big(\frac{N(m+N)}{N(M+N)} + \frac{n^r(M-m)}{N(M+N)}\Big) \geqslant O\Big(\frac{m+N}{M+N}\Big) > O\Big(\frac{m}{M}\Big),$$

что и требовалось доказать.

Пример 1. Для того чтобы проиллюстрировать предлагаемые в настоящей статье алгоритмы, рассмотрим пример с одной и той же матрицей для каждого нового алгоритма.

Пусть матрица A функционала (4) имеет следующий вид:

$$A = \begin{pmatrix} 1 & 2 & \dots & R & R+1 & \dots & & & \dots & R^2 \\ 0 & 1 & \dots & R-1 & R & \dots & & & \dots & R^2-1 \\ 1 & 2 & \dots & R & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & \dots & R & 0 & \dots & 0 \\ & & & & & \vdots & & & & \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & \dots & R \end{pmatrix},$$

где $R = 30$. В этой матрице содержатся две строки окаймления ($m = 2$) и R узких блоков одинаковой длины ($n_r = R$). При $R = 30$ имеем $N = 900$, $M = 32$ и число ненулевых элементов матрицы равно 2999. С помощью алгоритма, основанного на методе симплексных погружений, минимизировался квадратичный функционал (2) при $b_i = 0$, $i = 1, \dots, 32$, на параллелепипеде $0 \leqslant x_j \leqslant 1000$, $j = 1, \dots, 900$.

Программы, реализующие универсальный метод симплексных погружений и модифицированный алгоритм, основанный на этом методе, решали данный пример за $T_c = 937$ с и $T_{cy} = 904$ с соответственно. При этом было проведено 156 итераций. Программы написаны на языке FORTRAN и выполнялись на ПЭВМ с процессором AMD K6-2, частота 400 МГц.

2. Алгоритм, основанный на методе Дикина

Вычислительная схема метода внутренней точки (метода Дикина) для ЗЛП (1), (3), (4) состоит из двух этапов [5]. Первый этап заключается в нахождении допустимой точки, удовлетворяющей условиям (3) и (4), а второй — в нахождении оптимума. Опишем оба этапа.

Перед выполнением этапов определяется точка $x^0 = (x_1^0, \dots, x_N^0)$, удовлетворяющая (3):

$$x_j^0 = (\alpha_j + \beta_j)/2, \quad j = 1, \dots, N. \tag{8}$$

1. Нахождение допустимой точки.

Для нахождения точки, лежащей в области (3), (4), выполняется следующий итерационный процесс.

1.1. Находятся величины

$$\sigma_j^k = \min[(x_j^k - \alpha_j)^2, (\beta_j - x_j^k)^2], \quad j = 1, \dots, N. \tag{9}$$

1.2. Решается система линейных уравнений

$$\sum_{t=1}^{M} b_{st}^k u_t^k = r_s^k, \quad s = 1, \dots, M, \tag{10}$$

где $b_{st}^k = \sum\limits_{j=1}^{N} \sigma_j^k a_{sj} a_{tj}$, $s, t = 1, \ldots, M$, а правые части определяются по формулам $r_i^k = b_i - \sum\limits_{j=1}^{N} a_{ij} x_j^k$, $i = 1, \ldots, M$.

1.3. Находятся двойственные невязки

$$\delta_j^k = \sum_{i=1}^{M} a_{ij} u_i^k, \quad j = 1, \ldots, N.$$

1.4. Вычисляется функционал

$$\Phi^k = \sum_{j=1}^{N} \sigma_j^k (\delta_j^k)^2. \tag{11}$$

1.5. Определяется направление спуска

$$s_j^k = \delta_j^k \sigma_j^k, \quad j = 1, \ldots, N. \tag{12}$$

1.6. Осуществляется итерационный переход по формулам

$$x_j^{k+1} = x_j^k + \lambda_k s_j^k, \quad j = 1, \ldots, N, \tag{13}$$

где $\lambda_k = \min(1, \varphi_k)$,

$$\varphi_k = \max(\rho\mu_k, (\Phi^k)^{-1/2}), \quad 0,5 \leqslant \rho \leqslant 1, \tag{14}$$

$$\mu_k = \min(\mu_k', \mu_k''), \quad \mu_k' = \min_{1 \leqslant j \leqslant N} [(x_j^k - \alpha_j)/s_j^k], \quad \mu_k'' = \min_{1 \leqslant j \leqslant N} [(\beta_j - x_j^k)/s_j^k].$$

Если евклидова норма $\|r^k\| \leqslant \varepsilon$, то точка x^k, удовлетворяющая (3) и (4), получена. В противном случае процесс продолжается при $\sqrt{\Phi^k} > \varepsilon$, а ситуация, когда $\sqrt{\Phi^k} \leqslant \varepsilon$, соответствует случаю отсутствия допустимого решения ЗЛП (1), (3) и (4).

2. Нахождение оптимума.

2.1. Вычисляются величины σ_j^k по формулам (9).

2.2. Решается система линейных уравнений (10), где матрица $\|b_{st}^k\|$ определяется так же, как в п. 1.2, а правые части вычисляются по формуле $r_i^k = \sum\limits_{j=1}^{N} c_j \sigma_j^k a_{ij}$, $i = 1, \ldots, M$.

2.3. Находятся невязки

$$\delta_j^k = \sum_{i=1}^{M} a_{ij} u_i^k - c_j, \quad j = 1, \ldots, N.$$

2.4. Вычисляется функционал Φ^k по формуле (11).

2.5. Определяется направление спуска s_j^k по формуле (12).

2.6. Осуществляется итерационный переход
$$x_j^{k+1} = x_j^k + \lambda_k s_j^k, \quad j = 1, \ldots, N,$$
где $\lambda_k = \max(\rho\mu_k, \varphi_k)$, а величина φ_k вычисляется по формуле (14). Если $\sqrt{\Phi_k} < \varepsilon$, то вычисления прекращаются и вектор x^{k+1} полагается решением ЗЛП (1), (3), (4).

Известно [6], что, начиная с некоторого k, последовательность двойственных переменных сходится к своему пределу не медленнее чем с квадратичной скоростью.

Учитывая, что коэффициенты системы линейных уравнений (10) являются произведениями строк матрицы A с неотрицательными весовыми множителями, докажем следующее утверждение.

Лемма 2. *Если матрица A — узкоблочная с окаймлением, то симметричная матрица B^k является такой диагональной с окаймлением, что ее первые m строк и первые m столбцов являются матрицами размера $M \times m$ и $m \times M$ соответственно, а матрица, составленная из последних $M - m$ строк и $M - m$ столбцов, является диагональной* (см. рисунок, *б*).

Доказательство. Из п. 1.2 следует, что в случае, когда матрица A является узкоблочной, формулы для вычисления b_{st}^k выглядят следующим образом:
$$b_{st}^k = \begin{cases} \sum\limits_{j=1}^{N} \sigma_j^k a_{sj} a_{tj}, & 1 \leqslant s, t \leqslant m \quad \text{(окаймление)}; \\ \sum\limits_{j=l}^{l+n_r-1} \sigma_j^k a_{sj} a_{tj}, & l = 1 + \sum\limits_{p=1}^{r-1} n_p, \quad s = t > m \quad \text{(диагональ)}; \\ 0 & \text{в остальных случаях.} \end{cases} \tag{15}$$
Из (15) следует справедливость леммы 2.

Кроме того, отметим, что матрица B^k является положительно определенной, если матрица A имеет полный ранг [3].

Построим теперь основанную на методе Дикина вычислительную схему решения узкоблочной ЗЛП. Начальная точка x^0 отыскивается по формуле (8), и далее осуществляется итерационный процесс поиска допустимого решения.

3. Нахождение допустимой точки с учетом структуры матрицы A.

3.1. Вычисляются величины σ_j^k по формуле (9).

3.2. Матрица B^k определяется по формуле (15), а правые части системы (10) находятся по формулам
$$r_i^k = b_i - \sum_{j=1}^{N} a_{ij} x_j^k$$

при $i \leqslant m$ и

$$r_i^k = b_i - \sum_{j=l}^{l+n_r-1} a_{ij} x_j^k,$$

где $l = 1 + \sum\limits_{s=1}^{r-1} n_r$, если $i = m + r$.

3.3. Первые m компонент вектора $u^k \in R^M$ находятся из решения системы линейных уравнений

$$\sum_{t=1}^{m} v_{st}^k u_t^k = e_s^k, \quad 1 \leqslant s \leqslant m, \tag{16}$$

где коэффициенты v_{st}^k вычисляются по формуле

$$v_{st}^k = b_{st}^k - \sum_{l=m+1}^{M} b_{sl}^k b_{lt}^k / b_{ll}^k, \quad s, t = 1, \ldots, m, \tag{17}$$

а

$$e_s^k = r_s^k - \sum_{l=m+1}^{M} \frac{b_{s,l}^k}{b_{l,l}^k} r_l^k. \tag{18}$$

3.4. Вычисляются последние $M - m$ компонент вектора u^k:

$$u_t^k = \frac{1}{b_{t,t}^k} \left[r_t^k - \sum_{j=1}^{m} b_{t,j}^k u_j^k \right], \quad t = m+1, \ldots, M. \tag{19}$$

3.5. Вычисляются невязки

$$\delta_j^k = \sum_{i=1}^{m} a_{ij} u_i^k + a_{m+r,j} u_{m+r}^k,$$

где $1 + \sum\limits_{l=1}^{r-1} n_l \leqslant j \leqslant \sum\limits_{l=1}^{r} n_l$.

3.6. Определяется функционал Φ^k по формуле (11).

3.7. Направление спуска s^k находится по формуле (12).

3.8. Итерационный переход осуществляется по формулам (13)–(14).

Критерий завершения поиска допустимого решения аналогичен критерию завершения процесса 1.1–1.6.

Нахождение оптимума проводится с помощью описанного выше итерационного процесса 3.1–3.8, в котором

правые части r_i^k из п. 3.2 равны $\sum\limits_{j=1}^{N} a_{ij} \sigma_j^k c_j$ при $i \leqslant m$ и $\sum\limits_{j=l}^{l+n_r-1} a_{ij} \sigma_j^k c_j$,

где $l = 1 + \sum\limits_{s=1}^{r-1} n_r$ при $i = m + r$;

в п. 3.5 невязки δ_j^k определяются по формуле

$$\delta_j^k = \sum_{i=1}^{m} a_{ij} u_i^k + a_{m+r,j} u_{m+r}^k - c_j$$

для $1 + \sum\limits_{l=1}^{r-1} n_l \leqslant j \leqslant \sum\limits_{l=1}^{r} n_l$, а п. 3.8 аналогичен описанному выше п. 2.6.

Окончание счета происходит в ситуации, когда $\sqrt{\Phi^k} \leqslant \varepsilon$.

Утверждение 1. *Число итераций, используемое при решении* ЗЛП (1), (3), (4) *с узкоблочной матрицей алгоритмом, основанным на методе Дикина, совпадает с числом итераций, используемым при решении этой задачи базовым алгоритмом Дикина.*

Доказательство. Легко проверить, что вектор u^k, полученный в п. 3.2–3.4 вычислительной схемы, основанной на методе Дикина, совпадает с решением СЛУ (10) алгоритма Дикина на этапе нахождения допустимого решения или с решением СЛУ (10) из п. 2.2 алгоритма Дикина на этапе поиска оптимума. Поскольку совпадение остальных пунктов проверяется визуально, последовательность допустимых точек $x^0, x^1, \ldots, x^k, \ldots$, получаемая на двух этапах предлагаемой вычислительной схемы, совпадает с последовательностью точек, получаемых методом Дикина. Отсюда следует справедливость утверждения 1.

Из леммы 2 следует, что матрица B^k является симметричной и положительно определенной. Докажем, что матрица V^k обладает такими же свойствами. Это позволит решать СЛУ из п. 3.3 методом квадратного корня.

Лемма 3. *Матрица* $V^k = \|v_{ij}^k\|$ *симметрична и положительно определена.*

Доказательство. Используя свойство симметричности матрицы B^k, из формулы (17) получаем

$$v_{ji}^k = b_{ji}^k - \sum_{s=m+1}^{M} b_{js}^k b_{si}^k / b_{ss}^k = b_{ij}^k - \sum_{s=m+1}^{M} b_{sj}^k b_{ij}^k / b_{ss}^k = v_{ij}^k, i, j = 1, \ldots, m.$$

Следовательно, V^k симметрична.

Теперь докажем положительную определенность матрицы V^k. Матрицу B^k представим в виде

$$B^k = \begin{bmatrix} V & Y \\ Y^T & D \end{bmatrix},$$

где V, Y и D — матрицы размера $m \times m$, $m \times (M-m)$ и $(M-m)\times(M-m)$ соответственно. Тогда $V^k = V - YD^{-1}Y^T$.

Умножим слева матрицу B^k на P и справа на P^T, где

$$P = \begin{bmatrix} E_m & -YD^{-1} \\ 0 & E_{M-m} \end{bmatrix}.$$

Тогда

$$PB^kP^T = \begin{bmatrix} V - YD^{-1}Y^T & 0 \\ 0 & D \end{bmatrix}.$$

В левом верхнем углу матрицы PB^kP^T находится матрица V^k. Пусть $x^* = (x_1^*, \dots, x_m^*)^T \neq 0$ — произвольный вектор. Положим $x = (x_1^*, \dots, x_m^*, 0, \dots, 0)^T$ — M-мерный вектор. Тогда

$$(V^kx^*, x^*) = (PB^kP^Tx, x) = \Big(B^kP^Tx, \underbrace{P^Tx}_{x'}\Big) = (B^kx', x') > 0$$

в силу положительной определенности матрицы B^k. Следовательно, матрица V^k положительно определена, что и завершает доказательство.

Теперь можем доказать следующее утверждение.

Лемма 4. *Коэффициент уменьшения трудоемкости* $\Omega_{\text{Д}}$ *не превосходит* $O(\frac{m^2}{M^2})$.

Доказательство. В ситуации, когда матрица A имеет общий вид, из формул 1.1–1.6 следует, что вычисление σ_j^k, $1 \leqslant j \leqslant N$, требует $O(N)$ операций; построение матрицы B^k требует $O(NM^2)$ операций; подсчет правых частей СЛУ (10) — $O(NM)$ операций; ее решение — $O(M^3)$ операций; двойственные невязки в п. 1.3 определяются за $O(NM)$ операций; все остальные действия в данном разделе требуют $O(N)$ операций. Одна итерация выполняется за $O(NM^2 + M^3 + NM)$ операций.

В алгоритме для решения ЗЛП с узкоблочной матрицей для нахождения числа операций будем использовать формулы (15)–(19). Здесь определение величин r_j^k, $j = 1, \dots, N$, требует $O(Nm)$ операций; составление матрицы B^k по формулам (15) требует $O(Nm^2)$ операций; правые части системы $e_s^k, s = 1, \dots, m$, вычисляются за $O(Mm - m^2)$ операций; матрица V^k составляется за $O(Mm^2)$ операций; решение системы (16) требует $O(m^3)$ операций; последние компоненты вектора u^k находятся за $O(Mm - m^2)$ операций; двойственные невязки в этом случае вычисляются за $O(Nm)$ операций; трудоемкость остальных операций не отличается от общего случая. На выполнение одной итерации в рассматриваемом алгоритме необходимо использовать $O(Nm^2 + Mm^2 + m^3 + Mm - m^2 + Nm)$ операций. Далее, используя утверждение 1, получаем

$$\begin{aligned} \Omega_D &= O\Big(\frac{Nm^2 + Mm^2 + m^3 + Mm - m^2 + Nm + NM}{NM^2 + M^3 + NM}\Big) \\ &= O\Big(\frac{Nm^2 + Mm^2 + m^3}{NM^2 + M^3}\Big) \leqslant O\Big(\frac{m^2}{M^2}\frac{N + M + m}{N + M}\Big). \quad (20) \end{aligned}$$

Рассмотрим второй множитель в скобках отдельно:

$$\frac{N+M+m}{N+M} = 1 + \frac{m}{N+M}.$$

Учитывая, что число m по сравнению с M и N невелико, можно сделать вывод, что $\frac{N+M+m}{N+M} < 2$. Таким образом,

$$\Omega_D \leqslant O\Big(\frac{m^2}{M^2}\Big).$$

Лемма 4 доказана.

Пример 2. Рассмотрим решение ЗЛП с матрицей A ограничений (4), приведенной в примере 1, и нулевым вектором правых частей. Пусть при $R = 30$ минимизируется функция

$$\sum_{j=1}^{900} x_j$$

при условии, что $0 \leqslant x_j \leqslant 1000$, $j = 1, \ldots, 900$.

При решении этой задачи универсальным методом Дикина на каждой итерации необходимо решать СЛУ размера 32×32, тогда как при решении этой же задачи предложенным в настоящем разделе алгоритмом, основанным на методе Дикина, размерность решаемой СЛУ равна 2×2. При программной реализации оптимальное значение целевой функции было найдено за 7 итераций. При решении этой задачи универсальным методом Дикина время решения $T_{\text{Д}}$ составило 8,18 с, а модифицированным алгоритмом $T_{\text{Ду}}$ — 0,55 с.

3. Алгоритм решения узкоблочной ЗЛП, основанный на методе Кармаркара

В [16] описание метода Кармаркара и доказательство его сходимости приведены для ЗЛП в следующем виде:

$$\begin{gathered} \sum_{j=1}^{N} c_j x_j \to \min, \\ Ax = 0, \quad \sum_{j=1}^{N} x_j = 1, \quad x_j \geqslant 0, \quad j = 1, \ldots, N. \end{gathered} \tag{21}$$

Предполагается, что для ЗЛП (21)

1) $x^0 = e/N$, где $e = (1, \ldots, 1)^T$ является допустимой точкой;

2) если x — допустимое решение, то $cx \geqslant 0$, а если x^* — оптимальное решение, то $cx^* = 0$.

Очевидно, что $x^0 = e/N$ может быть взята в качестве начальной точки процесса, k-я итерация которого выполняется следующим образом.

Сначала на шагах 1.1–1.4 находится очередная точка последовательности $x^{k+1} = \phi(x^k)$. Функция $\phi(x)$ определяется следующей последовательностью операций.

1.1. Пусть

$$B = \begin{pmatrix} AD & & \\ 1 & \dots & 1 \end{pmatrix},$$

где $D = \mathrm{diag}(x_1, \dots, x_N)$. Вычисляется ортогональная проекция $D\overline{c}$ в нуль-пространстве преобразования B:

$$c_p = [E - B^T(BB^T)^{-1}B]Dc. \tag{22}$$

Обращение матрицы BB^T производится по формуле одноранговой модификации за $O(M^2\sqrt{N})$ операций. Это преобразование эквивалентно решению не более $\sqrt{N}$ треугольных СЛУ [16].

1.2. Нормируется вектор c_p

$$\tilde{c} = \frac{c_p}{\|c_p\|}.$$

1.3. Определяется сдвиг

$$\phi'(x) = \frac{1}{N} - \frac{\gamma}{\sqrt{N(N-1)}}\tilde{c},$$

где $\gamma \in (0,1)$ — параметр.

1.4. Применяется проективное преобразование

$$\phi(x) = \frac{D\phi'(x)}{e^T D\phi'(x)},$$

обратное к $\phi'(x)$.

Далее точка x^{k+1} проверяется на недопустимость.

Определяется потенциальная функция как

$$f(x) = \sum_{i=1}^{N} \ln \frac{cx}{x_i}.$$

В [16] доказано, что если $f(x^{k+1}) > f(x^k) - \delta$, где $\delta = \ln(1+\gamma)$, то значение целевой функции ЗЛП (21) строго положительно. В этом случае, если ЗЛП (21) получена из ЗЛП (1), (3), (4), то решение последней задачи не ограничено.

В заключение точка x^{k+1} проверяется на оптимальность, т. е. автоматически переходим на шаг 1.1 только в ситуации, когда $cx^{k+1} > \varepsilon > 0$.

Докажем следующее утверждение.

Лемма 5. *Коэффициент уменьшения трудоемкости Ω_K не меньше 1/6 при $\sqrt{N} \leqslant M$ и больше $O(m/M)$ в противном случае.*

Доказательство. Заметим, что число итераций для решения задач со специальной структурой не превосходит $O(N \ln N)$ и совпадает с их числом в случае матрицы общего вида. В общем случае для построения матрицы B необходимо $O(NM)$ операций; вычисление Dc из (22) требует $O(N)$ операций; вычисление BDc требует $O(2NM)$ операций (в отличие от общепринятой методики оставим у выражения NM коэффициент 2); $(BB^T)^{-1}$ — $O(M^2\sqrt{N})$ операций (алгоритм с данной трудоемкостью предложен в [16]); окончание вычисления $B^T(BB^T)^{-1}BDc$ требует $O(M^2 + 2NM)$ операций. В этой ситуации итерация выполняется за $O(5NM + M^2\sqrt{N})$ операций (величины порядка N и M^2 можно опустить). В узкоблочном случае для построения матрицы B используется $O(Nm)$ операций; определение BDc требует $O(Nm + n^r(M-m))$ операций, где $n^r = \max\limits_{r=1,\dots,R}\{n_r\}$; $(BB^T)^{-1}$ — $O(M^2\sqrt{N})$ операций; вычисление $(BB^T)^{-1}BDc$ — еще $O(M^2)$ операций; вычисление $B^T(BB^T)^{-1}BDc$ требует дополнительно $O(Nm)$ операций. В целом итерация выполняется за $O(Nm + n^r(M-m) + M^2\sqrt{N})$ операций. Отсюда получаем

$$\Omega_K = O\Big(\frac{Nm + n^r(M-m) + M^2\sqrt{N}}{5NM + M^2\sqrt{N}}\Big).$$

Рассмотрим два случая. В первом случае $M \geqslant \sqrt{N}$ и

$$\Omega_K \geqslant O\Big(\frac{n^rM + M^2\sqrt{N}}{5NM + M^2\sqrt{N}}\Big) > O\Big(\frac{M^2\sqrt{N}}{5NM + M^2\sqrt{N}}\Big) > 1/6.$$

Во втором случае $M < \sqrt{N}$ и

$$\Omega_K \geqslant O\Big(\frac{Nm + n^r(M-m) + M^2\sqrt{N}}{5NM}\Big) \geqslant O\Big(\frac{m}{M}\Big).$$

Лемма 5 доказана.

Вернемся к примеру 2. Заметим, что этот пример отвечает всем условиям, необходимым для применения метода Кармаркара. При программной реализации оптимальное значение целевой функции было найдено за 9 итераций. При решении этой задачи универсальным методом Кармаркара время решения $T_{\text{К}}$ составило 542 с, а модифицированным методом Кармаркара $T_{\text{Ку}}$ — 118 с, так что требуемое неравенство также выполняется.

4. Алгоритм решения задач с двусторонними ограничениями, основанный на методе Ринальди

Алгоритм, рассмотренный в предыдущем разделе, сформулирован для ЗЛП частного вида, которые на практике встречаются редко. Поэтому приведем новую вычислительную схему проективного алгоритма

типа Кармаркара для решения ЗЛП с узкоблочной матрицей ограничений и двусторонними ограничениями. Эта схема основана на алгоритме, предложенном Дж. Ринальди [17].

С использованием преобразования

$$\tilde{x} = \frac{1}{N} B^{-1}(x - \alpha), \tag{23}$$

где $B = \operatorname{diag}((\beta_1 - \alpha_1), \ldots, (\beta_N - \alpha_N))$, задача (1), (3), (4) приводится к виду: минимизировать

$$\sum_{j=1}^{n} \tilde{c}_j^T \tilde{x}_j \tag{24}$$

при ограничениях

$$\tilde{A}\tilde{x} = \tilde{b}, \quad 0 \leqslant \tilde{x}_j \leqslant 1/N, \quad j = 1, \ldots, N.$$

Здесь

$$\begin{aligned} \tilde{c}_j &= c_j(\beta_j - \alpha_j), \\ \tilde{a}_{ij} &= N a_{ij}(\beta_j - \alpha_j), \\ \tilde{b}_i &= b_i - \sum_{j=1}^{N} a_{ij}\alpha_j. \end{aligned}$$

Очевидно, что это преобразование сохраняет узкоблочную структуру матрицы A. Поэтому в дальнейшем для удобства обозначений знак волны будем опускать.

Рассмотрим задачу (24) при следующих предположениях:

1) $c^T x^* = 0$, где x^* — оптимальное решение задачи (24);

2) точка $\frac{1}{2N}e$ допустима для задачи (24) и $c^T e > 0$;

3) ранг матрицы A равен M.

С учетом этих предположений вычислительную схему предлагаемого алгоритма следует начать с точки $x^0 = \frac{1}{2N}e$. Далее выполняется следующий итерационный процесс.

Шаг 1. Находится направление спуска b_k. Полагается

$$\begin{aligned} d^k &= \left[I - J_k^T A^T (A J_k H_k^{-1} J_k^T A^T)^{-1} A J_k H_k^{-1}\right] D_k c, \\ b_k &= \left[(d^k)^T H_k^{-1} d\right]^{-1/2} H_k^{-1} d^k, \end{aligned}$$

где

$$D_k = \operatorname{diag}(x_1^k, \ldots, x_N^k),$$

$$G_k = \frac{1}{N} I - D_k,$$
$$F_k = \frac{1}{N} e^T G_k^{-1} e D_k + \frac{1}{N} e e^T (I - G_k^{-1} D_k),$$
$$J_k = \frac{1}{N} e^T G_k^{-1} e D_k + D_k e e^T (I - G_k^1 D_k),$$
$$H_k = I + \frac{1}{(e^T G_k^{-1} e)} F_k^T G_k^{-2} F_k = I + D_k^2 G_k^{-2} + v_k u_k^T + u_k v_k^T.$$

Здесь $u = (I - D_k G_k^{-1})e$, а
$$v = \frac{1}{2} \frac{e^T G_k^{-2} e}{(e^T G_k^{-1} e)^2} D_k G_k^{-2} e.$$

(Заметим, что при таком представлении матрица H_k является суммой двух диагональных матриц и двух матриц $v_k u_k^T$ и $u_k v_k^T$ ранга 1. Поэтому обращение матрицы H_k можно выполнить за $O(N^2)$ операций по формуле
$$\begin{aligned}&(P + uv^T + vu^T)^{-1}\\ &= P^{-1} - \frac{1}{1+b} P^{-1}(uv^T + vu^T)P^{-1} + \frac{1}{(1+a)^2 - cd} P^{-1}(duu^T + cvv^T)P^{-1},\end{aligned}$$
где P — диагональная матрица, $a = v^T P^{-1} u$, $b = a - cd/(1+a)$, $c = u^T P^{-1} u$, $d = v^T P^{-1} v$.) Последнее преобразование эквивалентно двум применениям одноранговой модификации.

Шаг 2. Полагается
$$x^{k+1} = \frac{1}{N}\Big(1 + \alpha \sqrt{\frac{2N}{2N-1}} \frac{e^T (I - G_k^{-1} D_k) b^k}{e^T G_k^{-1} e}\Big)^{-1} \Big(x^k - \alpha \sqrt{\frac{2N}{2N-1}} D_k b^k\Big).$$

Шаг 3. Если $c^T x^{k+1}/c^T x^k \leqslant 2^{-p}$, то процесс завершен, иначе переход на шаг 1.

Основной учет специфики матрицы ограничений в вычислительной схеме, основанной на методе Ринальди, происходит при более экономном вычислении матрицы $A J_k H_k^{-1} J_k^T A^T$, а также умножении матриц A и A^T на N-мерные и M-мерные векторы соответственно.

Оценим КУТ метода Ринальди для узкоблочной матрицы ограничений.

Лемма 6. *Коэффициент уменьшения трудоемкости Ω_R не меньше $O(m/M)$.*

Доказательство. Заметим, что число итераций для реализации метода при решении задачи (24) со специальной структурой совпадает

с числом итераций, необходимых для решения аналогичной задачи общего вида.

Трудоемкость выполнения одной итерации в общем случае такова. Для нахождения матриц G_k, F_k, J_k и H_k необходимо использовать $O(N^2)$ операций; для обращения матрицы H_k, как отмечалось ранее, требуется $O(N^2)$ операций; при вычислении d_k наиболее трудоемкой частью является нахождение $AJ_kH_k^{-1}J_k^TA^T$ и ее обращение; все остальные операции представляют собой умножение матрицы на вектор и могут быть выполнены за $O(N^2)$ операций. Определение матрицы $AJ_kH_k^{-1}J_k^TA^T$ требует в этом случае $O(NM(N+M))$ операций. Легко показать (см. доказательство леммы 2), что эта матрица является симметричной и положительно определенной, если ранг матрицы A равен M. Поэтому ее обращение может быть выполнено за $O(M^3)$ операций. Следовательно, выполнение одной итерации требует $O(NM(N+M)+M^3)$ операций.

В случае узкоблочной матрицы A вычисление матрицы $AJ_kH_k^{-1}J_k^TA^T$ выполняется за $O(N^2m+N^2M+NMm)$ операций, ее обращение — так же, как и в общем случае, за $O(M^3)$ операций. Других существенных отличий от общего случая нет. Поэтому трудоемкость одной итерации в узкоблочном случае равна $O(N^2m+N^2M+NMm+M^3)$. Следовательно,

$$\Omega_R = \frac{O(N^2m+N^2M+NMm+M^3)}{O(N^2M+NM^2+M^3)} = \frac{O(N^2M+NMm+M^3)}{O(N^2M+NM^2+M^3)} + \frac{O(N^2M)}{O(N^2M+NM^2+M^3)} \geqslant \frac{O(N^2M+NMm+M^3)}{O(N^2M+NM^2+M^3)} > O\Big(\frac{m}{M}\Big).$$

Последнее неравенство легко проверяется. Лемма 6 доказана.

Рассмотрим пример 2. При программной реализации оптимальное значение целевой функции было найдено за 15 итераций. При решении этой задачи универсальным методом Ринальди время решения T_R составило 998,12 с, а модифицированным методом Ринальди $T_{R\text{у}}$ — 784,43 с.

В заключение заметим, что если под эффективным учетом структуры ЗЛП понимать коэффициент уменьшения трудоемкости вычислений этой задачи, то справедливо следующее

Утверждение 2. *Наиболее эффективно учитывать структуру узкоблочной матрицы ограничений позволяет вычислительная схема, основанная на методе Дикина.*

Доказательство с очевидностью следует из лемм 1, 4–6.

Во всех предыдущих рассуждениях проводилось сравнение трудоемкости выполнения итерации алгоритма, модифицированного для решения задач с узкоблочной структурой матрицы, и алгоритма,

предназначенного для решения произвольной ЗМП. Было показано, что в методе Дикина происходит самое существенное уменьшение трудоемкости. Однако в общем случае трудоемкость итерации метода Кармаркара больше, чем трудоемкость итерации метода Дикина. Это объясняется экономным способом обращения матрицы в методе Кармаркара по формуле одноранговой модификации, что требует $O(M^2\sqrt{N})$ операций вместо $O(M^3)$ операций. Следующая теорема утверждает, что применение модифицированного алгоритма Дикина дает не только относительное, но и абсолютное уменьшение трудоемкости выполнения итерации алгоритма.

Теорема 1. *Если*

$$\frac{m}{M} < \frac{2M}{5M+N}, \quad \frac{m^2}{M^2} < \frac{\sqrt{N}}{N+1,5M}, \tag{25}$$

то трудоемкость итерации вычислительной схемы, основанной на методе Кармаркара для решения ЗЛП с узкоблочной матрицей ограничений, больше трудоемкости итерации модифицированного метода Дикина.

Доказательство. При доказательстве лемм 4 и 5 было показано, что трудоемкость одной итерации в методе Кармаркара в случае узкоблочной с окаймлением матрицы A равна $O(Nm + n^r(M-m) + M^2\sqrt{N})$ операций, а трудоемкость одной итерации метода Дикина равна $O(Nm^2 + Mm^2 + m^3 + Mm - m^2 + Nm)$ операций. Можно уточнить эти оценки и найти трудоемкость итерации этих алгоритмов в полиномиальном виде.

В узкоблочном случае при выполнении итерации метода Кармаркара вычисление матрицы B требует $Nm + \sum\limits_{r=1}^{R} n_r = Nm + N$ операций; для нахождения Dc из (22) требуется N операций; определение BDc требует $2N(m+1) + 2\sum\limits_{r=1}^{R} n_r = 2Nm + 4N$ операций; вычисление $(BB^T)^{-1}$ по формуле одноранговой модификации требует $M^2\sqrt{N}$ операций; вычисление $(BB^T)^{-1}BDc$ требует $2(M+1)^2$ операций; вычисление $B^T(BB^T)^{-1}BDc$ требует $2N(m+2)$ операций. Нормировка вектора, вычисление $\phi'(x^k)$ и выполнение обратного преобразования, очевидно, выполняются за $9N$ операций. Таким образом, итерация метода Кармаркара выполняется за

$$P_K = 5Nm + M^2\sqrt{N} + (M+1)^2 + 22N \tag{26}$$

операций.

Для нахождения трудоемкости итерации в методе Дикина для решения ЗЛП с узкоблочной матрицей используем формулы вычислительной

схемы 3.1–3.9. Подсчет r_j^k, $j = 1, \dots, N$, требует $2Nm + 2\sum_{r=1}^{R} n_r + M = 2Nm + 2N + M$ операций; составление матрицы B^k по формулам (15) требует $\frac{1}{2}\Big(Nm + \sum_{r=1}^{R} n_r + 2Nm^2 + 3m\sum_{r=1}^{R} n_r + 2\sum_{r=1}^{R} n_r\Big) = Nm^2 + 2Nm + \frac{3}{2}N$ операций; правые части системы e_s^k, $s = 1, \dots, m$, вычисляются за $m(1 + 3(M - m)) = 3m(M - m) + m$ операций; матрица V^k находится за $\frac{m^2(1+3(M-m))}{2} = \frac{3}{2}m^2(M - m) + \frac{m^2}{2}$ операций; решение системы (16) требует $O(\frac{1}{6}m^3)$ операций; последние компоненты вектора u^k из п. 3.5 находятся за $(M-m)(2m+1) = 2m(M-m)+M-m$ операций; двойственные невязки в узкоблочном случае вычисляются за $N(2m+1) = 2Nm+N$ операций. Очевидно, что остальные действия (нахождение величин σ_j, $j = 1, \dots, N$, функции Φ^k, определение направления спуска и выполнение итерационного перехода) требуют $14N$ операций. Таким образом, на выполнение одной итерации метода Дикина в узкоблочном случае используется

$$P_{Д} = Nm^2 + \frac{3}{2}m^2(M-m) + \frac{m^2}{2} + 5m(M-m) + \frac{1}{6}m^3 + 6Nm + 18\frac{1}{2}N + 2M \tag{27}$$

операций.

Исходя из условий (25), получаем

$$2M^2 > 5Mm + Nm, \quad M^2\sqrt{N} > Nm^2 + 1,5Mm^2.$$

Тогда

$$\begin{aligned} P_K - P_D &= 5Nm + M^2\sqrt{N} + (M+1)^2 + 22N \\ &- \Big(Nm^2 + \frac{3}{2}m^2(M-m) + \frac{m^2}{2} + 5m(M-m) + \frac{1}{6}m^3 + 6Nm + 18\frac{1}{2}N + 2M\Big) \\ &= \Big(M^2\sqrt{N} - Nm^2 - \frac{1}{6}m^3\Big) + (5Nm + 2M^2 + 4,5m^2 - 6Nm - 5Nm) \\ &\quad + \Big(22N + 4M + 2 - 18,5N - 3,5M - \frac{3}{2}m\Big). \end{aligned}$$

Очевидно, что при выполнении условий (25) последнее выражение имеет положительное значение. Теорема 1 доказана.

ЛИТЕРАТУРА

1. **Анцыз С. М., Пудова М. В.** Методы внутренней точки для решения задач со специальной структурой. Новосибирск, 1997. 27 с. (Препринт / РАН. Сиб. отд-ние. Ин-т математики; № 44).

2. **Ащепков Л. Т., Белов Б. И., Булатов В. П.** Методы решения задач математического программирования и оптимального управления. Новосибирск: Наука, 1984.

3. **Воеводин В. В.** Численные методы алгебры: Теория и алгорифмы. М.: Наука, 1966.

4. **Данциг Дж.** Линейное программирование, его применение и обобщения. М.: Прогресс, 1966.

5. **Дикин И. И., Зоркальцев В. И.** Итеративное решение задач математического программирования. Новосибирск: Наука, 1980.

6. **Дикин И. И., Попова О. М.** Исследование и ускорение сходимости алгоритмов метода внутренних точек. Новосибирск: Наука, 1997.

7. **Звягина Р. А.** Программа реализации на М-20 модифицированного симплекс-метода с узкоблочной матрицей // Оптимальное планирование: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1966. Вып. 4. С. 63–124.

8. **Корнаи И., Липтак Т.** Планирование на двух уровнях // Применение математики в экономических исследованиях. М.: Мысль, 1965. Т. 3.

9. **Лэсдон Л. С.** Оптимизация больших систем. М.: Наука, 1975.

10. **Рубинштейн Г. Ш.** О решении задач линейного программирования большого объема // Оптимальное планирование: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1964. Вып. 2. С. 3–22.

11. **Цурков В. И.** Декомпозиция в задачах большой размерности. М.: Наука, 1981.

12. **Шмырев В. И.** Контроль исходных данных для программы, реализующей модифицированный симплекс-метод с узкоблочной матрицей // Оптимальное планирование: Сб. науч. тр. Новосибирск: Ин-т математики СО АН СССР, 1966. Вып. 4. С. 125–136.

13. **Adler I., Resende M. G., Veiga G., Karmarkar N.** An implementation of Karmarkar's algorithm for linear programming // Math. Program. 1984. V. 44, N 3. P. 297–337.

14. **Bernes E. K.** A variation on Karmarkar's algorithm for solving programming problems // Math. Program. 1986. V. 36, N 2. P. 174–182.

15. **Choi I. C., Goldfarb D.** Exploiting special structure in a primal-dual path-following algorithm // Math. Program. 1993. V. 58, N 2. P. 33–53.

16. **Karmarkar N.** A new polynomial-time algorithm for linear programming // Combinatorica. 1984. V. 4, N 4. P. 373–395.

17. **Rinaldi G.** A projective method for linear programming with box-type constrains // Algorithmica. 1986. V. 1, N 4. P. 517–527.

18. **Todd M. J.** Exploiting special structure in Karmarkar's linear programming algorithm // Math. Program. 1988. V. 41, N 1. P. 97–113.

19. **Vanderbei R. J., Carpenter I. J.** Symmetric indefinite systems for interior point methods // Math. Program. 1993. V. 58, N 1. P. 1–33.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия

Статья поступила
10 декабря 2001 г.,
переработанный вариант —
8 февраля 2002 г.