ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

2003 МАТЕМАТИКА № 6 (493)

УДК 519.853

М.В. ПУДОВА

О КОЭФФИЦИЕНТЕ УМЕНЬШЕНИЯ ТРУДОЕМКОСТИ НЕКОТОРЫХ АЛГОРИТМОВ

1. Введение

В [1] были предложены алгоритмы решения задач линейного и квадратичного программирования (ЗЛП и ЗКП), матрицы ограничений или квадратичной формы которых имеют блочную с окаймлением структуру, а именно, первые строки являются разреженными [2] подматрицами, а далее по диагонали расположены непересекающиеся блоки, рис. 1 a).

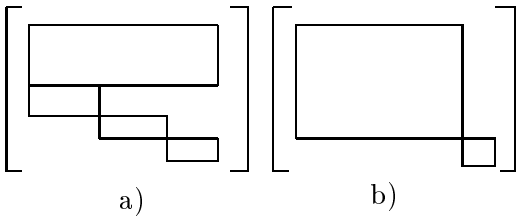

Рис. 1. a) блочная с окаймлением матрица, b) "почти заполненная матрица".

Подобные задачи возникают, например, при исследовании оптимальных стратегий функционирования иерархических систем. Существуют алгоритмы решения этих задач, основанные на методах декомпозиции ([3], [4]), а также алгоритмы, предложенные в [5]–[7]. Последние алгоритмы основываются на методе последовательного улучшения плана [8]. С тех пор появились новые методы, имеющие полиномиальную трудоемкость. В данной работе для решения задач математического программирования, имеющих блочную с окаймлением структуру, предлагаются модификации некоторых полиномиальных алгоритмов и одного алгоритма, для которого не доказана полиномиальная трудоемкость.

Для того чтобы оценить эффективность предложенных алгоритмов, в [1] был введен *коэффициент уменьшения трудоемкости* (КУТ), равный отношению трудоемкости алгоритма, учитывающего структуру задачи, к трудоемкости универсального алгоритма. Кроме того, были приведены оценки коэффициентов уменьшения трудоемкости для рассмотренных алгоритмов. При выводе этих оценок не накладывалось никаких условий на соотношение размерностей окаймления и блоков. Таким образом, не исключался случай, когда один из блоков будет иметь большую размерность, т. е. матрица будет "почти заполненной" (вид такой матрицы приведен на рис. 1 b)). В данной работе КУТ оцениваются для тех же алгоритмов, но при дополнительных ограничениях, исключающих ситуацию "почти заполненности".

Вычислительные схемы, предложенные в работе, основаны на четырех методах: Кармаркара, Ринальди, симплексных погружений Анциферова, методе Дикина. Выбор методов для исследования был обусловлен следующими соображениями: метод Кармаркара имеет наилучшую теоретическую оценку трудоемкости среди алгоритмов, в которых допустимая область

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проекты 03-01-00877 и 00-15-98884) и совета по грантам при президенте Российской Федерации № НШ-80.2003.6.

аппроксимируется эллипсоидами [9]. Он сформулирован для задач линейного программирования, ограничения и оптимальное значение целевой функции которых имеют специальный вид. Метод Ринальди [10] позволяет решать задачи более общего вида. Формулы, реализующие метод Дикина [11], предложенный в 1967 г., совпадают с формулами модификации метода Кармаркара для решения задач линейного программирования в каноническом виде [12].

Метод симплексных погружений [13] принадлежит к другому классу задач, в которых допустимая область аппроксимируется не эллипсоидами, а симплексами.

Существуют несколько работ, посвященных применению метода Кармаркара к решению ЗЛП со структурированными матрицами. В [14] предложена модификация метода Кармаркара для ЗЛП с разреженной матрицей произвольного вида. В [15] приводится модификация метода Кармаркара для решения ЗЛП с матрицей, имеющей некоторую (не уточненную) специальную структуру. При построении этой модификации использована идея декомпозиции Данцига–Вулфа. В [16] исследуется решение ЗЛП с блочной (без окаймления) матрицей. К сожалению, во всех этих работах вопрос об эффективности предложенных модификаций исследуется только экспериментально.

Введем обозначение для КУТ — Ω_L, где L идентифицирует метод, на основе которого построен новый алгоритм. Здесь $L = \mathrm{C}$ для метода симплексных погружений, $L = K$ для метода Кармаркара, $L = D$ для метода Дикина и $L = R$ для метода Ринальди.

Приведем теперь математическую формулировку решаемых задач.

2. Постановка задачи

Рассматриваются задачи линейного или квадратичного программирования 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

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

и

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

Предполагаем, что матрица A имеет блочную с окаймлением структуру: первые m строк есть матрица размера $m \times N$, а строки с $m+1$ по M — последовательность R непересекающихся блоков, расположенных по диагонали. Размеры блоков $m_r \times n_r$, $r = 1, \ldots, R$. Формально эти условия можно записать следующим образом: $A = (A_{ij})$,

$$A_{ij} = \begin{cases} a_{ij}, & \text{если } i \le m \text{ или } \Big(m + \sum\limits_{l=1}^{r-1} m_l < i \le m + \sum\limits_{l=1}^{r} m_l \bigwedge \\ & \qquad \bigwedge \sum\limits_{l=1}^{r-1} n_l < j \le \sum\limits_{l=1}^{r} n_l \text{ для некоторого } r\Big); \\ 0 & \text{в противном случае.} \end{cases} \tag{5}$$

Будем считать, что на размерности блоков наложены дополнительные ограничения

$$n_r = \frac{N}{R}(1+\nu_r), \quad m_r = \frac{M-m}{R}(1+\varepsilon_r), \quad \text{где } |\nu_r| < \frac{1}{2}, \quad |\varepsilon_r| < \frac{1}{2}, \quad r = 1, \ldots, R. \tag{6}$$

Выполнение последнего условия исключает ситуацию "почти заполненности". Кроме того, предполагаем, что матрица A имеет большую размерность: $N > 10^3$, $M > 10^3$.

3. Оценка КУТ вычислительной схемы, основанной на методе симплексных погружений

В этом и последующих разделах вначале приводится алгоритм решения ЗЛП либо ЗКП с матрицей общего вида, а далее — вычислительная схема, модифицированная для решения задачи, матрица ограничений или функционала которой удовлетворяет условиям (5).

Рассмотрим процесс решения задачи (2), (3) методом симплексных погружений (доказательство его полиномиальной трудоемкости — в [13]).

Пусть в R^N задан N-мерный симплекс S_N^k, определяемый матрицей

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

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

1) Найдем центр тяжести $\overline{x}^k$ симплекса S_N^k по формуле

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

2) определим вектор нормали

$$g^k = \nabla H(\overline{x}^k)$$

плоскости, отсекающей часть симплекса S_N^k, и получим усеченный симплекс

$$\widetilde{S}^k = \{x \in S_N^k : g^k (x - \overline{x}^k) \le 0\}$$

для его погружения в симплекс S_N^{k+1} минимального объема;

3) подсчитаем числа

$$\gamma_i^k = g^k (x_i^k - \overline{x}^k), \quad i = 1, \dots, N+1,$$

где x_i^k — i-я строка матрицы X^k; пусть $I^- = \{i : \gamma_i^k < 0\}$, $I^+ = \{i : \gamma_i^k > 0\}$;

4) определим индекс q из условия

$$\gamma_q^k = \min_{i \in I^-} \gamma_i^k,$$

очевидно, $\gamma_q^k < 0$;

5) найдем точку минимума t^* вогнутой функции

$$\Pi(t) = \prod_{1 \le i \le N+1,\ i \ne q} (1 + \gamma_i^k t)$$

на отрезке $0 \le t \le -1/\gamma_q^k$; пусть $q_k = 1/\Pi(t^*)$, следовательно, $q_k < 1$;

6) вычислим

$$v_i^* = 1 + t^* \gamma_i^k, \quad \tau_i^* = 1/v_i^*, \quad i = 1, \dots, N+1, \quad i \ne q;$$

7) пересчитаем вершины симплекса S_N^k по формуле

$$x_i^{k+1} = x_i^k + \tau_i^* (x_i^k - x_q^k), \quad i = 1, \dots, N+1, \quad i \ne q, \quad x_q^{k+1} = x_q^k;$$

новая матрица X^{k+1} будет соответствовать новому симплексу S_N^{k+1} минимального объема, содержащему усеченный симплекс $\widetilde{S}^k$;

8) подсчитаем 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 заменим k на $k+1$ и перейдем к п. 1).

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

$$\nabla_j H(x) = 2\sum_{i=1}^{M} a_{ij} \sum_{l=1}^{N} (a_{il}x_l^k - b_i), \quad j = 1, \ldots, N. \tag{7}$$

Обозначим выражение в скобках через δ_i. Если матрица A имеет блочную с окаймлением структуру, то

$$\delta_i = \begin{cases} \sum\limits_{j=1}^{N} (a_{ij}x_j - b_i) & \text{при } i \le m; \\ \sum\limits_{j=k_r+1}^{k_r+n_r} (a_{ij}x_j - b_i) & \text{при } i = m + i_0, \text{ где } \sum\limits_{l=1}^{r-1} m_l + 1 \le i_0 \le \sum\limits_{l=1}^{r} m_l, \quad k_r = \sum\limits_{l=1}^{r-1} n_l. \end{cases} \tag{8}$$

Вычислительная схема решения ЗКП (2), (3) с окаймленной блочной матрицей функционала таким образом совпадает с описанной выше за исключением п. 2), в котором градиент определяем по формуле

$$\nabla_j H(x) = 2\Big[\sum_{i=1}^{m} a_{ij}\delta_i + \sum_{i=k+1}^{k+m_r} a_{ij}\delta_i\Big], \tag{9}$$

где $k = \sum\limits_{l=1}^{r-1} m_l$, $\sum\limits_{l=1}^{r-1} n_l + 1 \le j \le \sum\limits_{l=1}^{r} n_r$.

Обозначим $\widehat{m} = \max\{m, \max\limits_{r=1,\ldots,R} m_r\}$ и для оценки трудоемкости предложенного алгоритма докажем следующее утверждение.

Лемма 1. *Коэффициент уменьшения трудоемкости* $\Omega_C \ge O(\widehat{m}/M)$.

Доказательство. Число итераций для реализации алгоритма, основанного на методе симплексных погружений, совпадает с числом итераций, необходимых для решения задачи общего вида этим методом. Поэтому число операций уменьшается за счет более экономного осуществления итерации. В ситуации, когда матрица A общего вида, для нахождения градиента по формуле (7) требуется $O(NM)$ операций, а все остальные действия для выполнения одной итерации можно проделать за $O(N^2)$ операций. Всего для выполнения одной итерации метода симплексных погружений необходимо $O(N^2 + NM)$ операций.

Из (8) и (9) следует, что число операций для нахождения градиента в блочном случае равно $O(Nm + Nm^r + n^r(M - m))$, где $n^r = \max\limits_{1\le i\le R}\{n_i\}$, а $m^r = \max\limits_{1\le i\le R}\{m_i\}$. Трудоемкость остальных действий для выполнения итерации совпадает с общим случаем. Поэтому трудоемкость выполнения итерации вычислительной схемы, основанной на методе симплексных погружений, равна $O(N^2 + Nm + Nm^r + n^r(M - m))$. Тогда

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

Так как

$$O\Big(\frac{N^2 + Nm + Nm^r}{N^2 + NM}\Big) > O\Big(\frac{m + m^r}{N + M}\Big) > O\Big(\frac{\widehat{m}}{M}\Big),$$

то

$$\Omega_C \ge O\Big(\frac{\widehat{m}}{M}\Big) + O\Big(\frac{n^r(M-m)}{N(M+N)}\Big) \ge O\Big(\frac{\widehat{m}}{M}\Big). \quad \square$$

4. Оценка КУТ для вычислительной схемы, основанной на методе Кармаркара

В [9] доказана сходимость метода Кармаркара, если решается ЗЛП в частном виде:

$$\sum_{j=1}^{N} c_j x_j \to \min,$$
$$Ax = 0, \quad \sum_{j=1}^{N} x_j = 1, \quad x_j \geq 0, j = 1, \ldots, N. \tag{10}$$

При этом предполагается, что

1) $x_0 = e/N$, где $e = (1, \ldots, 1)^T$, является допустимой точкой;
2) если x^* — оптимальное решение, то $cx^* = 0$.

Опишем k-ю итерацию алгоритма Кармаркара, считая, что точка x^k удовлетворяет (10).

Вначале на шагах 1.1–1.5 вычисляется следующая точка последовательности:

$$x^{k+1} = \phi(x^k).$$

1.1. Пусть

$$B = \begin{pmatrix} & AD^k & \\ 1 & \ldots & 1 \end{pmatrix},$$

где $D^k = \operatorname{diag}(x_1^k, \ldots, x_N^k)$;

1.2. вычисляем ортогональную проекцию $D^k\overline{c}$ в нуль-пространстве преобразования B

$$\overline{c_p} = [E - B^T(BB^T)^{-1}B]D\overline{c}, \tag{11}$$

обращение матрицы BB^T производится по формуле одноранговой модификации за $O(M^2\sqrt{N})$ операций;

1.3. нормируем вектор c_p

$$c = \frac{\overline{c_p}}{\|\overline{c_p}\|};$$

1.4. определяем сдвиг

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

где $\gamma \in (0, 1)$ — параметр;

1.5. применяем обратное проективное преобразование к $\phi'(x^k)$

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

Далее проверяем точку x^{k+1} на недопустимость.

Определим потенциальную функцию как

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

Если $f(x^{k+1}) > f(x^k) - \delta$, где $\delta = \ln(1 + \gamma)$, то вычисления останавливаются и делается заключение о том, что значение целевой функции строго положительно. Если каноническая форма задачи получена путем преобразования задачи линейного программирования в стандартной форме, то делается вывод, что исходная задача не имеет конечного оптимального решения.

В заключение проверяем точку x^{k+1} на оптимальность. Эта проверка производится периодически и включает переход из текущей точки в граничную точку без увеличения значения целевой функции и проверку этой граничной точки на оптимальность. Если $cx^{k+1} > 0$, повторяем итерацию алгоритма, т. е. переходим на шаг 1.1.

Модификацию метода Кармаркара будем строить для ЗЛП (10) такой, что ее матрица ограничений A имеет блочную с окаймлением структуру. Условия 1) и 2) для этой задачи также выполняются. Модификация состоит в более экономном умножении матрицы A на вектор-столбец p и векторы-строки q на матрицу A при определении ортогональной проекции по формуле (11). Эти умножения выполняются по формулам

$$(Ap)_i = \begin{cases} \sum\limits_{j=1}^{N} (a_{ij}p_j), & \text{если } i \le m; \\ \sum\limits_{j=k_r+1}^{k_r+n_r} (a_{ij}p_j), & \text{если } i = m + i_0, \text{ где } \sum\limits_{l=1}^{r-1} m_l + 1 \le i_0 \le \sum\limits_{l=1}^{r} m_l, \quad k_r = \sum\limits_{l=1}^{r-1} n_l, \end{cases}$$

$$(qA)_j = \sum_{i=1}^{m} a_{ij}q_i + \sum_{i=k+1}^{k+m_r} a_{ij}q_i, \quad \text{где} \quad k = \sum_{l=1}^{r-1} m_l, \quad \sum_{l=1}^{r-1} n_l + 1 \le j \le \sum_{l=1}^{r} n_r.$$

Таким образом, вычислительная схема предлагаемого алгоритма решения рассматриваемой ЗЛП совпадает с приведенной, кроме шага 1.2.

Справедлива

Лемма 2. *Коэффициент уменьшения трудоемкости* $\Omega_{\text{К}}$ *больше* $1/6$ *в случае, когда* $\sqrt{N} \le M$, *и не меньше* $O(\widehat{m}/M)$ *в противном случае.*

Доказательство. Оценим число операций, необходимых для выполнения итерации метода Кармаркара в случае, когда матрица A не обладает специальной структурой. Для построения матрицы B необходимо $O(NM)$ операций; вычисление $D\overline{c}$ из (11) требует $O(N)$ операций; вычисление $BD\overline{c}$ требует $O(2NM)$ операций (в отличие от общепринятой методики оставим у выражения NM коэффициент 2); вычисление $(BB^T)^{-1}$ требует $O(M^2\sqrt{N})$ операций (алгоритм с данной трудоемкостью предложен в [9]); вычисление $(BB^T)^{-1}BD\overline{c}$ требует $O(M^2)$ операций; вычисление $B^T(BB^T)^{-1}BD\overline{c}$ требует $O(2NM)$ операций. Всего в этой ситуации итерация выполняется за $O(5NM + M^2\sqrt{N})$ операций (величины порядка N и M^2 можем пропустить).

Число итераций в общем случае и для решения задач со специальной структурой одинаково. В случае блочной матрицы ограничений вычисление $BD\overline{c}$ требует $O(Nm + n^r(M - m))$ операций; вычисление $(BB^T)^{-1}$ требует $O(M^2\sqrt{N})$ операций; вычисление $(BB^T)^{-1}BD\overline{c}$ требует $O(M^2)$ операций; вычисление $B^T(BB^T)^{-1}BD\overline{c}$ требует $O(N(m+m^r))$ операций. Итого, итерация выполняется за $O(Nm + n^r(M - m) + M^2\sqrt{N} + N(m + m^r))$ операций. Следовательно,

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

и после оценки получим

$$\Omega_K \ge O\Big(\frac{M^2\sqrt{N}}{5NM + M^2\sqrt{N}}\Big) > 1/6 \quad \text{при} \quad M \ge \sqrt{N}.$$

Вторая ситуация заключается в том, что $M < \sqrt{N}$. Тогда

$$\Omega_K \ge O\Big(\frac{Nm + Nm^r}{5NM}\Big) \ge O\Big(\frac{\widehat{m}}{M}\Big). \quad \square$$

5. Оценка КУТ вычислительной схемы, основанной на методе Ринальди

В развитие идей Кармаркара в [10] рассматривается алгоритм решения задачи: минимизировать функцию (1) при ограничениях (4) и

$$0 \le x_j \le 1/N, \quad j = 1, \ldots, N. \tag{12}$$

Очевидно, задача (1), (3), (4) сводится к (1), (4), (12).

Рассмотрим задачу (1), (4), (12) при предположениях

1) $c^T x^* = 0$, где x^* — решение (1), (4), (12);
2) точка $\frac{1}{2N}e$ допустима для этой задачи и $c^T e > 0$;
3) A имеет ранг M.

Опишем алгоритм.

Шаг 1. Начинаем с точки $x^0 = \frac{1}{2N}e$; положим $D_0 = \frac{1}{2N}I$.

Шаг 2. Поиск направления. Положим

$$d^k = [I - J_k^T A^T (A J_k H_k^{-1} J_k^T A^T)^{-1} A J_k H_k^{-1}] D_k c,$$
$$b_k = [(d^k)^T H_k^{-1} d^k]^{-1/2} H_k^{-1} d^k,$$

где

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

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 $u_k v_k^T$, имеющих ранг, равный 1. Поэтому обращение матрицы H_k можно выполнить за $O(N^2)$ операций по формуле

$$(P + uv^T + vu^T)^{-1} = P^{-1} - \frac{1}{1+b} P^{-1}(uv^T + vu^T)P^{-1} + \frac{1}{(1+a)^2 - cd} P^{-1}(duu^T + cvv^T)P^{-1},$$

где P — диагональная матрица, $a = v^T P^{-1} u$, $b = a - cd/(1+a)$, $c = u^T P^{-1} u$, $d = v^T P^{-1} v$.)

Шаг 3. Положим

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

и

$$D_{k+1} = \mathrm{diag}(x_1^{k+1}, \ldots, x_N^{k+1}).$$

Шаг 4. Если $c^T x^{k+1} / c^T x^k \le 2^{-p}$, то процесс завершен; иначе переходим на шаг 1.

Структуру матрицы ограничений будем учитывать при вычислении матрицы $A J_k H_k^{-1} J_k^T A^T$, а также при умножении матриц A и A^T на N-мерные и M-мерные векторы соответственно.

Оценку КУТ метода Ринальди для блочной матрицы ограничений дает

Лемма 3. *Коэффициент уменьшения трудоемкости* $\Omega_R \ge O(m/M)$.

Для доказательства леммы 3, как и раньше, заметим, что число итераций для реализации метода при решении задачи (1), (4), (12) 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 $O(N^2)$ 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, а 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 общем случае $O(NM(N+M))$ операций; очевидно, эта матрица является симметричной и положительно определенной, если A имеет ранг, равный M. Ее обращение может быть выполнено за $O(M^3)$ операций. Общее число операций, необходимых для выполнения одной итерации метода Ринальди без учета структуры матрицы ограничений, равно $O(N^2M+NM^2+M^3)$.

В случае блочной матрицы A умножение ее на J_k слева требует $O\Big(N^2m+NM_r\sum\limits_{r=1}^{R}n_r\Big)=O(N^2(m+m^r))$ операций, а умножение на A^T справа матрицы размера $M\times N$ требует $O\Big(NMm+M\sum\limits_{r=1}^{R}n_rm^r\Big)=O(NM(m+m^r))$ операций. Трудоемкость остальных действий по выполнению итерации совпадает с общим случаем. Всего для выполнения одной итерации в блочном случае необходимо $O((N^2+NM)(m+m^r)+M^3)$ операций. Тогда

$$\Omega_R=\frac{O((N^2+NM)(m+m^r)+M^3)}{O(N^2M+NM^2+M^3)}=\frac{O(N^2m+NMm+M^3)}{O(N^2M+NM^2+M^3)}+$$
$$+\frac{O(N^2m_r+NMm^r)}{O(N^2M+NM^2+M^3)}\geq\frac{O(N^2\widehat{m}+NM\widehat{m}+M^3)}{O(N^2M+NM^2+M^3)}>O\Big(\frac{\widehat{m}}{M}\Big).\quad\square$$

6. Оценка КУТ вычислительной схемы, основанной на методе Дикина

Алгоритм метода внутренней точки (метода Дикина) для ЗЛП (1), (3), (4) состоит из двух этапов [11]. Первый этап заключается в нахождении допустимой точки, удовлетворяющей условиям (3), (4); второй — в нахождении оптимума. Опишем оба этих этапа.

I. Нахождение допустимой точки.

1.0. Определяется точка x^0, удовлетворяющая (3),

$$x_j^0=(\alpha_j+\beta_j)/2,\quad j=1,\ldots,N,\tag{13}$$

и далее для нахождения точки, лежащей в области (3), (4), выполняется следующий итерационный процесс;

1.1. вычисляются величины

$$\sigma_j^k=\min[(x_j^k-\alpha_j)^2,(\beta_j-x_j^k)^2],\quad j=1,\ldots,N;\tag{14}$$

1.2. решается система линейных уравнений

$$\sum_{t=1}^{M}b_{st}^ku_t^k=r_s^k,\quad s=1,\ldots,M,\tag{15}$$

где $b_{st}^k=\sum\limits_{j=1}^{N}\sigma_j^ka_{sj}a_{tj}$, $s,t=1,\ldots,M$, 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

$$\Phi^k = \sum_{j=1}^{N} \sigma_j^k (\delta_j^k)^2; \tag{16}$$

1.5. определяется направление спуска

$$s_j^k = \delta_j^k \sigma_j^k, \quad j = 1, \ldots, N; \tag{17}$$

1.6. осуществляется итерационный переход по формулам

$$x_j^{k+1} = x_j^k + \lambda_k s_j^k, \quad j = 1, \ldots, N, \text{ где} \tag{18}$$

$$\lambda_k = \min(1, \varphi_k),$$

$$\varphi_k = \max(\rho\mu_k, (\Phi^k)^{-1/2}), \quad 0.5 \le \rho \le 1, \tag{19}$$

$$\mu_k = \min(\mu_k', \mu_k''),$$

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

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

II. Нахождение оптимального решения ЗЛП (1), (3), (4) производится с помощью процесса 1.1–1.6, в котором

правые части СЛУ в п. 1.2 вычисляются по формуле $r_i^k = \sum\limits_{j=1}^{N} c_j \sigma_j^k a_{ij}$, $i = 1, \ldots, M$,

двойственные невязки в п. 1.3 находятся по формуле

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

итерационный переход выполняется по правилу

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

$$\lambda_k = \max(\rho\mu_k, \varphi_k),$$

а величина φ_k вычисляется по формуле (19).

Если $\sqrt{\Phi_k} < \varepsilon$, то вычисления прекращаются и вектор x^{k+1} полагается решением ЗЛП (1), (3), (4).

Для того чтобы построить эффективный алгоритм решения ЗЛП в случае, когда матрица A блочная с окаймлением, покажем сначала, что матрица СЛУ $\|b_{st}\|$ также имеет специальную структуру.

Лемма 4. *Если матрица A является блочной с окаймлением, то симметричная матрица $\|b_{st}^k\|$ является блочной с окаймлением, у которой первые m строк и столбцов — матрицы размера $M \times m$ и $m \times M$ соответственно, а матрица, составленная из последних $(M - m)$ строк и столбцов, блочно-диагональная, состоящая из R блоков размера $m_r \times m_r$.*

Доказательство. Если матрица A блочная, то в силу п. 1.2

$$b_{st}^k = \begin{cases} \sum\limits_{j=1}^{N} \sigma_j a_{sj} a_{tj}, & \text{если } 1 \le s,t \le m \quad (b_{st} \text{ из окаймления}); \\ \sum\limits_{j=l+1}^{l+n_r} \sigma_j a_{sj} a_{tj}, & \text{если } s = m+p, \quad \sum\limits_{k=1}^{r-1} m_k < p \le \sum\limits_{k=1}^{r} m_k \text{ или} \\ & t = m+q, \quad \sum\limits_{k=1}^{r-1} m_k < q \le \sum\limits_{k=1}^{r} m_k; \quad l = \sum\limits_{k=1}^{r-1} n_k \quad (b_{st} \text{ из } r\text{-го блока}); \\ 0 & \text{иначе,} \end{cases} \tag{20}$$

$s, t = 1, \ldots, M$.

Формулы (20) с очевидностью обосновывают утверждение леммы. □

Из этих формул также следует, что матрицу $B^k = \|b_{st}^k\|$ можно представить в виде

$$B = \begin{bmatrix} V & Y_1 & . & . & Y_R \\ Y_1^T & D_1 & . & . & . \\ . & . & . & . & . \\ Y_R^T & . & . & . & D_R \end{bmatrix},$$

где V, Y_r и D_r — матрицы размеров $m \times m$, $m \times m_r$, $m \times m_r$ соответственно.

Лемма 4 позволяет конструировать итеративный алгоритм решения блочных с окаймлением ЗЛП, основанный на методе Дикина.

Начальная точка в этом алгоритме вычисляется по формуле (13), а k-я итерация имеет следующий вид.

2.1. Величины σ_j^k находятся по формуле (14);

2.2. определяются правые части СЛУ (15): $r_i^k = b_i - \gamma_i^k$, где

$$\gamma_i^k = \begin{cases} \sum\limits_{j=1}^{N} a_{ij} x_j^k, & \text{если } i \le m; \\ \sum\limits_{j=s+1}^{s+n_r} a_{ij} x_j^k, & \text{если } i = m + i_0 \bigwedge \Big(\sum\limits_{l=1}^{r-1} m_l + 1 \le i_0 \le \sum\limits_{l=1}^{r} m_l \Big), \quad s = \sum\limits_{l=1}^{r-1} n_l; \end{cases}$$

2.3. элементы матрицы $\|b_{st}^k\|$ вычисляются по формулам (20);

2.4. решается система уравнений

$$\sum_{t=1}^{m} v_{st}^* y_t^k = e_s^k, \quad 1 \le s \le m,$$

где матрица $V^* = \|v_{st}^*\|$ определяется по формуле

$$V^* = V - \sum_{i=1}^{R} Y_i D_i^{-1} Y_i^T, \tag{21}$$

правые части считаются по формуле

$$e_s^k = r_s^k - \Big(\sum_{l=1}^{R} Y_l D_l^{-1} r^l \Big)_s, \quad s = 1, \ldots, m,$$

где r^l — m_l-мерный вектор, состоящий из компонент вектора r, соответствующих l-му блоку D_l матрицы $B^k = \|b_{st}^k\|$;

2.5. находятся векторы

$$f^i = D_i^{-1}(r^i - Y_i^T y^k), i = 1, \ldots, R;$$

2.6. определяются двойственные невязки

$$\delta_j = \sum_{i=1}^{m} a_{ij} u_i^k + \sum_{i=l}^{l+m_r} a_{ij} f_i^r,$$

где $l = m + \sum\limits_{s=1}^{r-1} m_s + 1$, если $\sum\limits_{s=1}^{r-1} n_s \le j \le \sum\limits_{s=1}^{r} n_s$;

2.7. вычисляется функционал Φ^k по формуле (16);

2.8. определяется направление спуска по формуле (17);

2.9. итерационный переход делается по формулам (18)–(19).

Критерий остановки поиска допустимой внутренней точки аналогичен критерию остановки процесса 1.1–1.6.

Нахождение решения ЗЛП осуществляется с помощью описанного выше процесса 2.1–2.9. При этом в п. 2.2 для вычисления величин r_i^k используются формулы

$$r_i^k = \begin{cases} \sum\limits_{j=1}^{N} a_{ij}\sigma_j^k c_j, & \text{если } i \le m; \\ \sum\limits_{j=s+1}^{s+n_r} a_{ij}\sigma_j^k c_j, & \text{если } i = m + i_0 \bigwedge \Big(\sum\limits_{l=1}^{r-1} m_l \le i_0 \le \sum\limits_{l=1}^{r} m_l \Big), \quad s = \sum\limits_{l=1}^{r-1} n_l, \end{cases}$$

а формулы п. 2.6 для нахождения двойственных невязок второго этапа необходимо дополнить вычитанием величин c_j из полученных в них сумм.

Лемма 5. *Матрица $V^* = \|v_{st}^*\|$ симметрична и положительно определена.*

Доказательство. Используя свойство симметричности матрицы B^k, из формулы (21) получаем

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

Следовательно, V^* симметрична.

Докажем теперь положительную определенность матрицы V^*. Очевидно, $V^* = V - Y D^{-1} Y^T$, где D — блочно-диагональная матрица размера $(M-m)\times(M-m)$, блоками которой являются матрицы D_r, а Y — матрица вида $[Y_1|\dots|Y_R]$.

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

$$P = \begin{bmatrix} E_m & -Y_1 D_1^{-1} & . & . & . \\ 0 & E_{m_1} & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & E_{m_R} \end{bmatrix},$$

а E_i — единичные матрицы размера $i \times i$. Тогда

$$P B^k P^T = \begin{bmatrix} V - Y D^{-1} Y^T & 0 \\ 0 & D \end{bmatrix}.$$

В левом верхнем углу матрицы PB^kP^T стоит матрица V^*. Пусть $x^* = (x_1^*,\dots,x_m^*)^T \ne 0$ — произвольный вектор. Положим $x = (x_1^*,\dots,x_m^*,0,\dots,0)^T$ — M-мерный вектор, а $x' = P^T x$. Тогда

$$(V^* x^*, x^*) = (PB^kP^T x, x) = (B^k P^T x, P^T x) = (B^k x', x') > 0$$

в силу положительной определенности матрицы B^k. Следовательно, матрица V^* положительно определена. □

Опираясь на результат леммы 5, делаем вывод о том, что СЛУ из п. 2.4 можно решать методом квадратного корня.

Оценим теперь коэффициент уменьшения трудоемкости для модифицированного алгоритма Дикина при предположениях (6).

Лемма 6. *Если* $N > 10^3$, $M > 10^3$ *и выполняется* (6), *то*

$$\Omega_D \le O\Big(\frac{\widehat{m}^2}{M^2} + \frac{M-m}{RM^2}\Big). \tag{22}$$

Доказательство. В ситуации, когда матрица A имеет общий вид, из формул 1.1–1.6 следует, что вычисление σ_j^k, $j = 1, \ldots, N$, требует $O(N)$ операций; построение матрицы B^k требует $O(NM^2)$ операций; подсчет правых частей СЛУ (15) требует $O(NM)$ операций; ее решение требует $O(M^3)$ операций; двойственные невязки 1.3 (с. 57) определяются за $O(NM)$ операций; все остальные действия в процессе 1.1–1.6 (сс. 57, 58) требуют $O(N)$ операций. Всего одна итерация выполняется за $O(NM^2 + M^3 + NM)$ операций.

Если матрица A имеет блочную структуру, то определение вектора r^k требует $O\Big(Nm + \sum\limits_{r=1}^{R} n_r m_r\Big)$ операций; матрица B^k по формулам (20) строится за $O\Big(Nm^2 + m\sum\limits_{r=1}^{R} n_r m_r\Big)$ операций; составление матрицы V^* по формуле (21) требует $O\Big(\sum\limits_{r=1}^{R} m_r^3 + m\sum\limits_{r=1}^{R} m_r^2 + m^2(M-m)\Big)$ операций; вектор правых частей e^k ищется за $O(m(M-m))$ операций; решение системы из 2.4 (с. 59) требует $O(m^3)$ операций; компоненты вектора f_i находятся за $O\Big(\sum\limits_{r=1}^{R} m_r^2\Big)$ операций; двойственные невязки считаются за $O\Big(Nm + \sum\limits_{r=1}^{R} n_r m_r\Big)$ операций; остальные действия имеют такую же трудоемкость, как в общем случае. Всего на выполнение одной итерации в блочном случае необходимо

$$O\Big(Nm^2 + m^3 + m\sum_{r=1}^{R} n_r m_r + \sum_{r=1}^{R} m_r^3 + m\sum_{r=1}^{R} m_r^2 + m^2(M-m) + \\ + \sum_{r=1}^{R} n_r m_r + \sum_{r=1}^{R} m_r^2 + m(M-m) + Nm\Big)$$

операций. Так как число итераций решения ЗЛП в общем виде равно числу итераций решения ЗЛП с блочной матрицей ограничений, КУТ равен

$$\Omega_D = O\Big(\frac{Nm^2 + m^2(M-m) + m^3 + Nm + m(M-m)}{NM^2 + NM + M^3}\Big) + \\ + O\Big(\frac{\sum\limits_{r=1}^{R} m_r^3 + m\sum\limits_{r=1}^{R} m_r^2 + m\sum\limits_{r=1}^{R} n_r m_r + \sum\limits_{r=1}^{R} n_r m_r + \sum\limits_{r=1}^{R} m_r^2}{NM^2 + NM + M^3}\Big). \tag{23}$$

Оценим первое слагаемое

$$O\Big(\frac{Nm^2 + m^2(M-m) + m^3 + Nm + m(M-m)}{NM^2 + NM + M^3}\Big) \le \\ \le O\Big(\frac{m^2}{M^2}\frac{N+M+m}{N+M+N/M} + \frac{m}{M^2}\frac{N+M}{N+M+N/M}\Big) \le \\ \le O\Big(\frac{m^2}{M^2}\frac{N+M+m}{N+M} + \frac{m}{M^2}\frac{N+M}{N+M+N/M}\Big) \le O\Big(\frac{m^2}{M^2}\Big).$$

Оценим второе слагаемое. Так как $m^r = \max\limits_{r=1,\dots,R}\{m_r\}$, то

$$O\Big(\frac{\sum\limits_{r=1}^{R} m_r^3 + m\sum\limits_{r=1}^{R} m_r^2 + m\sum\limits_{r=1}^{R} n_r m_r + \sum\limits_{r=1}^{R} n_r m_r + \sum\limits_{r=1}^{R} m_r^2}{NM^2 + NM + M^3}\Big) \le$$

$$\le O\Big(\frac{(m^r)^3 + m(m^r)(M-m) + Nmm^r + Nm^r + m^r(M-m)}{NM^2 + NM + M^3}\Big) \le$$

$$\le \frac{\widehat{m}^2}{M^2}\frac{m^r + M - m + N}{N + M} + \frac{m^r}{M^2}\frac{N + M - m}{N + M}.$$

Учитывая (6), получим $m^r < \frac{3}{2}\frac{M-m}{R}$. Тогда $\frac{m^r}{M^2} \le \frac{3}{2}\frac{M-m}{RM^2}$. Величины m и m^r малы по сравнению с величинами N и M. Поэтому $(N+M-m)/(N+M)$ и $(m^r+N+M-m)/(N+M)$ близки к единице. Следовательно, второе слагаемое из (23) можно оценить сверху величиной $O(\frac{\widehat{m}^2}{M^2} + \frac{M-m}{RM^2})$. Итак, (22) выполнено. □

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

Утверждение. *Наиболее эффективно учитывать структуру блочной с окаймлением матрицы ограничений позволяет вычислительная схема, основанная на методе Дикина.*

Доказательство с очевидностью следует из лемм 1, 2, 3 и 6.

Литература

1. Анцыз С.М., Пудова М.В. *Методы внутренней точки для решения задач со специальной структурой.* – Новосибирск: Изд-во ИМ СО РАН, 1997. – № 44. – 27 с.
2. Писсанецки С. *Технология разреженных матриц.* – М: Мир, 1988. – 410 с.
3. Данциг Дж. *Линейное программирование, его применение и обобщения.* – М.: Прогресс, 1966. – 600 с.
4. Корнаи И., Липтак Т. *Планирование на двух уровнях* // В кн. "Применение математики в экономических исследованиях". Под ред. В.Ц. Немчинова. – М.: Мысль, 1965. – Т. 3. – С. 495.
5. Рубинштейн Г.Ш. *О решении задач линейного программирования большого объема* // Оптимальное планирование. – Новосибирск: Наука, 1964. – Вып. 2. – С. 3–22.
6. Звягина Р.А. *Задачи линейного программирования с блочно-диагональными матрицами* // Оптимальное планирование. – Новосибирск: Наука, 1964. – Вып. 2. – С. 50–61.
7. Звягина Р.А. *Об общем методе решения задач линейного программирования блочной структуры* // Оптимизация. – Новосибирск, 1971. – Вып. 1. – С. 22–40.
8. Канторович Л.В. *Экономический расчет наилучшего использования ресурсов.* – М: Изд-во АН СССР, 1960. – 348 с.
9. Karmarkar N. *A new polynomial-time algorithm for linear programming* // Combinatorica. – 1984. – № 4. – P. 373–395.
10. Rinaldi G. *A projective method for linear programming with box-type constrains* // Algorithmica. – 1986. – № 1. – P. 517–527.
11. Дикин И.И. *Итеративное решение задач линейного и квадратичного программирования* // ДАН СССР. – 1967. – Т. 174. – С. 747–748.
12. Bernes E.K. *A variation on Karmarkar's algorithm for solving programming problems* // Math. Program. – 1986. – V. 36. – № 2. – P. 174–182.
13. Ащепков Л.Т., Белов Б.И., Булатов В.П. *Методы решения задач математического программирования и оптимального управления.* – Новосибирск: Наука, 1984. – 233 с.
14. Fujisawa K., Kojima M., Nakata K. *Exploiting sparsity in primal-dual interior-point methods for semidefinite programming* // Math. Program. – 1997. – V. 79. – P. 235–255.

15. Todd M.J. *Exploiting special structure in Karmarkar's linear programming algorithm* // Math. Program. – 1988. – V. 41. – № 1. – P. 97–113.
16. Choi I.C., Goldfarb D. *Exploiting special structure in a primal-dual path-following algorithm* // Math. Program. – 1993. – V. 58. – № 2. – P. 33–53.

Институт математики
Сибирского отделения
Российской Академии наук

Поступили
первый вариант 13.03.2000
окончательный вариант 29.01.2002