Выпуск 46(63) ОПТИМИЗАЦИЯ 1989 г.

Численные методы

519.853

УЧЕТ ПОГРЕШНОСТИ МАШИННЫХ ВЫЧИСЛЕНИЙ В МЕТОДЕ КАРМАРКАРА

Е.А.Пузанова

С точки зрения практического использования метода Кармаркара представляет интерес, какое влияние оказывает ограниченная точность выполнения арифметических операций вычислительной машиной на работу метода. В статье рассматриваются погрешности, с которыми точки строящейся методом Кармаркара последовательности $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$, сходящейся к оптимальному плану, удовлетворяют ограничениям решаемой задачи. Рассматривается влияние погрешностей на вычисляемые значения целевой функции и предложены определенные способы корректировки работы метода, позволяющие, в частности, предотвратить рост погрешности в ограничениях задачи при приближении к оптимальному плану.

§1. Введение

Н.Кармаркаром [1] предложен полиномиальный алгоритм для задач линейного программирования:

$$\min\{c^Tx \mid Ax = 0,\ e^Tx = 1,\ x \geq 0\},$$

где $c, x \in R^n$; $A \in R^{m\times n}$; $e^T = (1, \dots, 1) \in R^n$. Причем предполагается, что $c^Tx^* = 0$, где x^* – оптимальный план.

Для применения алгоритма необходимо задать начальную допустимую точку $x^{(0)} > 0$ и два параметра: q – натуральное число, α – вещественное. Параметр q указывает, во сколько раз должно быть уменьшено значение $c^Tx^{(0)}$; параметр $\alpha \in]0, 1[$ – множитель длины шага.

Алгоритм K строит последовательность допустимых точек $x^{(0)}, x^{(1)}, \ldots, x^{(k)}, \ldots$ следующим образом.

К.I°. Начальное состояние. Заданы $x^{(0)}$, $\mathcal{D}_0 = diag\{x_1^{(0)}, \ldots$ $\ldots, x_n^{(0)}\}$, $k = 0$ (номер итерации).

К.2°. Определение направления сдвига c_p. c_p удовлетворяет условию $Bc_p = 0$, где $B = \left[\begin{array}{c} A\mathcal{D}_k \\ \hline e^T \end{array}\right]$,

$$c_p = (I - B^T(BB^T)^{-1}B)\mathcal{D}_k c, \tag{I}$$

Здесь I - единичная $n \times n$ -матрица.

К.3°. Вычисление следующей точки. Определяем

$$b' = \frac{1}{n} e - \frac{\alpha}{\sqrt{n(n-1)}} \frac{c_p}{\|c_p\|}, \tag{2}$$

$$x^{(k+1)} = \frac{\mathcal{D}_k b'}{e^T \mathcal{D}_k b'},$$

$$\mathcal{D}_{k+1} = diag\{x_1^{(k+1)}, \ldots, x_n^{(k+1)}\}.$$

К.4°. Завершение. Если $c^T x^{(k+1)} / c^T x^{(0)} \leq 2^{-q}$, то завершить работу алгоритма; за решение принимается $\tilde{x} = x^{(k+1)}$. В противном случае положить $k = k+1$ и перейти к п.К.2°.

Это основная схема алгоритма. В [I] предложена также ее модификация (алгоритм М), которая обеспечивает сокращение числа операций при вычислении обратной матрицы на каждой итерации.

М.I°. Начальное состояние. Заданы $x^{(0)}$, $x'^{(0)} = x^{(0)}$, $\mathcal{D}_0 = \mathcal{D}'_0 = diag\{x_1^{(0)}, \ldots, x_n^{(0)}\}$, $\mathcal{M} = A(\mathcal{D}'_0)^2 A^T$, $\mathcal{M}_1 = \mathcal{M}^{-1}$, $k = 0$.

М.2°. Определение направления сдвига. Вводим матрицу $E = \mathcal{D}_k^{-1} \mathcal{D}'_k$, тогда

$$c_p = (I - E^2 B^T (BE^2B^T)^{-1} B) E^2 \mathcal{D}_k c,$$

где

$$BE^2B^T = \left[\begin{array}{c|c} A(\mathcal{D}'_k)^2 A^T & A\mathcal{D}'_k Ee \\ \hline (A\mathcal{D}'_k Ee)^T & e^T E^2 e \end{array}\right] = \left[\begin{array}{c|c} \mathcal{M} & d \\ \hline d^T & s \end{array}\right].$$

Вычисляя $(BE^2B^T)^{-1}$ по известной формуле

$$\left[\begin{array}{c|c} \mathcal{M} & d \\ \hline d^T & s \end{array}\right]^{-1} = \frac{1}{s-d^T\mathcal{M}^{-1}d}\left[\begin{array}{c|c} (s-d^T\mathcal{M}^{-1}d)\mathcal{M}^{-1}+(\mathcal{M}^{-1}d)(\mathcal{M}^{-1}d)^T & \mathcal{M}^{-1}d \\ \hline -(\mathcal{M}^{-1}d)^T & 1 \end{array}\right]$$

вместо $\mathcal{M}^{-1}$ используем $\mathcal{M}_1$.

М.3°. Вычисление следующей точки.

М.3а. Не отличается от п.К.3° основного алгоритма.

М.3б. Вычисление $\mathcal{D}'_{k+1}$ и преобразование $\mathcal{M}_1$. Определяем

$$\omega^{(k)} = \frac{1}{n}\sum_j x_j^{(k+1)}/x_j^{(k)} ,$$

$$x'^{(k+1)} = \omega^{(k)} x'^{(k)} ,$$

$$\mathcal{M}_1 = \frac{1}{(\omega^{(k)})^2}\mathcal{M}_1 .$$

Для $i=1,2,\dots,n$ выполняем действия:

если

$$(x_i'^{(k+1)}/x_i^{(k+1)})^2 \notin \left[\tfrac{1}{2}, 2\right],$$

то

$$g_i = (x_i^{(k+1)})^2 - (x_i'^{(k+1)})^2 ,$$

$$x_i'^{(k+1)} = x_i^{(k+1)} ,$$

$$\mathcal{M}_1 = (\mathcal{M}+g_i A_{*i} A_{*i}^T)^{-1} = \mathcal{M}_1 - g_i \frac{(\mathcal{M}_1 A_{*i})(\mathcal{M}_1 A_{*i})^T}{1+g_i A_{*i}^T \mathcal{M}_1 A_{*i}} , \qquad (3)$$

где A_{*i} – i-й столбец матрицы A .

Определяем $\mathcal{D}'_{k+1} = diag\{x_1'^{(k+1)},\dots,x_n'^{(k+1)}\}$.

М.4°. Завершение. Не отличается от п.К.4° основного алгоритма.

Опыт решения задач на ЭВМ методом Кармаркара показал [2], что по мере приближения точек последовательности $\{x^{(k)}\}$ к оптимальному плану x^* наблюдается рост погрешности в ограничениях $Ax=0$.

В §2 анализируются причины этого явления. В §3 рекомендуются определенные способы корректировки работы алгоритма с учетом возникающих погрешностей. В §4 приведены результаты вычислительных экспериментов с изложенным выше алгоритмом М и алгоритмом М с уточнениями.

Рассматриваются невырожденные задачи и предполагается, что матрица AA^T имеет полный ранг и достаточно хорошо обусловлена, т.е. исключена возможность получить на какой-либо k-й итерации матрицу $A\mathcal{D}_k^2 A^T$ (или $A(\mathcal{D}_k')^2 A^T$), близкую к матрице неполного ранга.

§2. Влияние погрешности машинных вычислений на работу алгоритма

2.1. <u>Погрешность в ограничениях задачи</u>. Как правило, в ЭВМ используется представление чисел с "плавающей запятой". В этом случае разрядность машины характеризуется параметрами ε_1 и ε_2, имеющими следующее содержание:

$1+\varepsilon_1$ – наименьшее машинное число, большее единицы,

$1/\varepsilon_2$ – наибольшее по модулю машинное число.

Известно [3], что абсолютная погрешность при машинном вычислении скалярного произведения n-мерных векторов a и b удовлетворяет неравенству

$$|a^T b - (a^T b)_{выч}| \le 2\varepsilon_1 \|a\| \, \|b\|, \tag{4}$$

если выполнены условия

$$n < 1/\varepsilon_1, \quad \|a\| > \sqrt{n\varepsilon_2/\varepsilon_1}, \quad \|b\| > \sqrt{n\varepsilon_2/\varepsilon_1} .$$

Отсюда для произведения матрицы $A\ (m \times n)$ на вектор b легко получить оценку

$$\|Ab - (Ab)_{выч}\| \le 2\varepsilon_1 \|A\|_E \|b\|, \tag{5}$$

где

$$\|A\|_E = \sqrt{\sum_{i,j} a_{ij}^2} .$$

При машинной реализации алгоритма точки последовательности $\{x^{(k)}\}$ удовлетворяют ограничениям задачи с некоторыми погрешностями, т.е.

$$(Ax^{(k)})_{\text{выч}} = \nu^{(k)},$$
$$(e^T x^{(k)})_{\text{выч}} = 1 + \nu_0^{(k)},$$

где $\nu^{(k)} \in R^m$, $\nu_0^{(k)} \in R^1$.

А направление сдвига c_p будет удовлетворять равенству

$$(Bc_p)_{\text{выч}} = \begin{bmatrix} \eta \\ \eta_{m+1} \end{bmatrix}, \quad \eta \in R^m, \quad \eta_{m+1} \in R^1.$$

ЛЕММА I. Векторы $\nu^{(k)}, \eta, \nu^{(k+1)}$ связаны равенством

$$\nu^{(k+1)} = \varkappa_1 \nu^{(k)} + \varkappa_2 \frac{\alpha}{\|c_p\|} \eta + \zeta, \tag{6}$$

где $\zeta \in R^m$, $\|\zeta\| \leq 2\varepsilon_1 \|A\|_E \|x^{(k+1)}\|$, а величины $\varkappa_1$ и $\varkappa_2$ ограничены: $\varkappa_1 \in [c_1, c_2]$, $\varkappa_2 \in [-c_3, -c_4]$; c_1, c_2, c_3, c_4 – положительные константы, зависящие от ε_1, n, α.

ДОКАЗАТЕЛЬСТВО. Учитывая п.К.3⁰ алгоритма К (или М.3а алгоритма М), получим

$$\nu^{(k+1)} = (Ax^{(k+1)})_{\text{выч}} = \frac{AD_k b'}{e^T D_k b'} = \frac{1}{e^T D_k b'} \left(\frac{1}{n} \nu^{(k)} - \frac{\alpha}{\sqrt{n(n-1)}} \frac{\eta}{\|c_p\|} \right) + \zeta, \tag{7}$$

$$(e^T x^{(k+1)})_{\text{выч}} = 1 + \nu_0^{(k+1)},$$

где ζ характеризует погрешность машинного произведения матрицы A на вектор $x^{(k+1)}$, а $\nu_0^{(k+1)}$ – погрешность скалярного произведения, т.е. (см. (4), (5))

$$\|\zeta\| \leq 2\varepsilon_1 \|A\|_E \|x^{(k+1)}\|,$$
$$|\nu_0^{(k+1)}| \leq 2\varepsilon_1 \|e\| \, \|x^{(k+1)}\| \leq 2\varepsilon_1 \sqrt{n}. \tag{8}$$

Оценка (8) верна для всех $k = 1, 2, \ldots$

Заметим, что

$$e^T D_k b' = \frac{1 + \nu_0^{(k)}}{n} - \frac{\alpha}{\sqrt{n(n-1)}} \frac{e^T D_k c_p}{\|c_p\|},$$

и так как

$$\frac{|e^T \mathcal{D}_\kappa c_p|}{\|c_p\|} \leq e^T \mathcal{D}_\kappa e \max \frac{|(c_p)_i|}{\|c_p\|} \leq 1 + \nu_0^{(\kappa)},$$

то

$$\frac{1+\nu_0^{(\kappa)}}{n}\left(1-\alpha\sqrt{n/(n-1)}\right) \leq e^T \mathcal{D}_\kappa b' \leq \frac{1+\nu_0^{(\kappa)}}{n}\left(1+\alpha\sqrt{n/(n-1)}\right).$$

Воспользовавшись оценкой (8), получим

$$\frac{(1-2\varepsilon_1\sqrt{n})}{n}\left(1-\alpha\sqrt{n/(n-1)}\right) \leq e^T \mathcal{D}_\kappa b' \leq \frac{1+2\varepsilon_1\sqrt{n}}{n}\left(1+\alpha\sqrt{n/(n-1)}\right).$$

Обозначив

$$\varkappa_1 = \frac{1}{n e^T \mathcal{D}_\kappa b'}, \quad \varkappa_2 = -\frac{1}{\sqrt{n(n-1)}\, e^T \mathcal{D}_\kappa b'},$$

из равенства (7) получим

$$\nu^{(\kappa+1)} = \varkappa_1 \nu^{(\kappa)} + \varkappa_2 \frac{\alpha}{\|c_p\|} \cdot \eta + \zeta,$$

причем

$$\varkappa_1 \in [c_1, c_2], \quad \varkappa_2 \in [-c_3, -c_4],$$

где

$$c_1 = \frac{1}{(1+2\varepsilon_1\sqrt{n})(1+\alpha\sqrt{n/(n-1)})},$$

$$c_2 = \frac{1}{(1-2\varepsilon_1\sqrt{n})(1-\alpha\sqrt{n/(n-1)})},$$

$$c_3 = \frac{\sqrt{n/(n-1)}}{(1-2\varepsilon_1\sqrt{n})(1-\alpha\sqrt{n/(n-1)})},$$

$$c_4 = \frac{\sqrt{n/(n-1)}}{(1+2\varepsilon_1\sqrt{n})(1+\alpha\sqrt{n/(n-1)})}. \quad \blacksquare$$

ЗАМЕЧАНИЕ. Константы c_1, c_2, c_3, c_4 определены и положительны, если выполнены условия $\alpha\sqrt{n/(n-1)} < 1$, $2\varepsilon_1\sqrt{n} < 1$.

Из равенства (6) видно, что погрешность в ограничениях может расти, особенно при малых значениях $\|c_p\|$, что характерно при приближении к x^*. Тогда главной частью погрешности становится составляющая $x_2 \frac{\alpha}{\|c_p\|} \eta$, и для устойчивости счета важно добиваться малой погрешности η, которая зависит от способа вычисления направления сдвига c_p.

Вычисление c_p по формуле (I) можно представить в виде:

$c_p = \mathcal{D}_k c - B^T Y$, где $Y = (BB^T)^{-1} B\mathcal{D}_k c$.

Поэтому для погрешности η верно

$$\begin{bmatrix} \eta \\ \hline \eta_{m+1} \end{bmatrix} = (B\mathcal{D}_k c - BB^T Y)_{выч},$$

т.е. $\begin{bmatrix} \eta \\ \hline \eta_{m+1} \end{bmatrix}$ характеризует, насколько точно вектор удовлетворяет системе $BB^T Y = B\mathcal{D}_k c$.

Наименьшую оценку этой погрешности получим, если будем считать точными участвующие в вычислении векторы $B\mathcal{D}_k c, Y$ и матрицу BB^T. Тогда η_i – погрешность машинного произведения $(BB^T)_{i*}$ (i-й строки матрицы BB^T) на вектор Y, т.е. (см. (4), (5))

$$|\eta_i| \le 2\varepsilon_1 \|(BB^T)_{i*}\| \, \|Y\|, \quad i = 1, 2, \dots, m+1,$$

и

$$\left\| \begin{matrix} \eta \\ \hline \eta_{m+1} \end{matrix} \right\| \le 2\varepsilon_1 \|BB^T\|_E \, \|Y\|. \qquad (9)$$

Аналогичным образом можно оценить $\begin{bmatrix} \eta \\ \hline \eta_{m+1} \end{bmatrix}$ для модифицированного алгоритма М:

$$|\eta_i| \le 2\varepsilon_1 \|(BE^2B^T)_{i*}\| \, \|Y\|, \quad i = 1, 2, \dots, m+1,$$

($Y = (BE^2B^T)^{-1} BE^2\mathcal{D}_k c$; $(BE^2B^T)_{i*}$ – i-я строка матрицы BE^2B^T),

$$\left\| \begin{matrix} \eta \\ \hline \eta_{m+1} \end{matrix} \right\| \le 2\varepsilon_1 \|BE^2B^T\|_E \, \|Y\|. \qquad (10)$$

Выполнения неравенств (9), (I0) всегда можно добиться, уточняя решения систем $BB^T Y = B\mathcal{D}_k c$ и $BE^2B^T Y = BE^2\mathcal{D}_k c$ методом итераций.

В то же время ясно, что даже если на каждом шаге погрешность ℓ имеет наилучшую при данном способе вычисления c_p оценку (9) (или (10)), тем не менее нельзя гарантировать, что погрешность $\nu^{(k+1)}$ не будет возрастать, когда $\|c_p\|$ становится маленькой, так как оценки (9) и (10) не зависят от $\|c_p\|$.

2.2. Погрешность вычисляемых значений целевой функции и корректировка условия завершения. Наличие погрешностей в вычислениях влияет и на получаемые значения целевой функции, а значит, и на условие завершения работы алгоритма.

В п.К.3^0 алгоритма К определяется точка b', которая по существу является решением следующей задачи:

- минимизировать $c'^T z$ (11)

при ограничениях

$$B z = \left[\frac{0}{1}\right], \|z - \frac{1}{n} e\| \le \frac{\alpha}{\sqrt{n(n-1)}}, \ z \geq 0, \tag{12}$$

где $c' = D_k c$. Напомним, что $b' = \frac{1}{n} e - \frac{\alpha}{\sqrt{n(n-1)}} \hat{c}_p$ (здесь $\hat{c}_p = \frac{c_p}{\|c_p\|}$), а $c_p = c' - B^T Y$.

В силу неточности вычислений $B\hat{c}_p = \gamma$ ($\gamma = \frac{1}{\|c_p\|}\left[\frac{\ell}{\ell_{m+1}}\right]$) в результате целевая функция (11) в точке b' имеет значение

$$c'^T b' = \frac{1}{n} c'^T e - \frac{\alpha}{\sqrt{n(n-1)}} c'^T \hat{c}_p =$$

$$= \frac{1}{n} c'^T e - \frac{\alpha}{\sqrt{n(n-1)}} (\|c_p\| + Y^T B \hat{c}_p) =$$

$$= \frac{1}{n} c'^T e - \frac{\alpha}{\sqrt{n(n-1)}} (\|c_p\| + Y^T \gamma).$$

При приближении к оптимальному плану $\|c_p\| \to 0$, а $\gamma = \frac{1}{\|c_p\|}\left[\frac{\ell}{\ell_{m+1}}\right]$, напротив, возрастает по норме, и тогда значение $c'^T b'$, а значит, и $c^T x^{(k+1)}$, в значительной степени зависит от погрешности вычислений γ. Если эта погрешность такова, что $|Y^T \gamma| > \|c_p\|$, то продолжать вычисления не имеет смысла, так как в этом случае точка b' заведомо не является решением задачи (11)-(12) и гарантировать дальнейшую правильную работу алгоритма нельзя.

Для модифицированного алгоритма аналогичное неравенство имеет вид

$$|Y^T\gamma| \;>\; \frac{c_p^T E^{-2} c_p}{\|c_p\|}\,. \qquad (13)$$

§3. Алгоритм Кармаркара с уточнениями

3.1. <u>Уточнение направления сдвига</u>. Как отмечалось выше, вблизи оптимальной точки x^* гласной частью погрешности $v^{(k+1)}$ становится составляющая $\varkappa_2 \frac{\alpha}{\|c_p\|}\eta$. Поэтому, чтобы предотвратить возрастание по абсолютной величине $\frac{1}{\|c_p\|}\eta$, желательно уметь уменьшать $|\eta_i|$ в соответствии с убыванием $\|c_p\|$.

В п.2.1 указывалось, что, уточняя вектор Y , можно добиться выполнения оценки (9) (или (10)). Опишем уточнение вектора Y методом итераций. Имеем Y , $c_p = \mathcal{D}_k c - B^T Y$ и

$$BB^T Y = B\mathcal{D}_k c - \begin{bmatrix} \eta \\ \eta_{m+1} \end{bmatrix}.$$

Вычисляем вектор

$$\tilde{z} = (BB^T)^{-1}\begin{bmatrix} \eta \\ \eta_{m+1} \end{bmatrix}$$

и полагаем $Y = Y + \tilde{z}$, при этом $c_p = c_p - B^T\tilde{z}$.

Таким образом, уточнению Y соответствует уточнение направления сдвига c_p , и уточнение Y можно заменить уточнением c_p .

Что дает такая замена? Уточняя c_p , можно добиться выполнения оценки

$$\left\|\begin{matrix} \eta \\ \eta_{m+1} \end{matrix}\right\| = \|(Bc_p)_{выч}\| < 2\varepsilon_1 \|B\|_E \|c_p\|, \qquad (14)$$

которая определяется погрешностью произведения матрицы на вектор, или $|\eta_i| \le 2\varepsilon_1 \|B_{i*}\| \, \|c_p\|$, $i = 1, 2, \ldots, m+1$ (B_{i*} - i-я строка матрицы B).

Тогда для $\gamma = B\hat{c}_p = \frac{1}{\|c_p\|} Bc_p$ верны оценки

$$|\gamma_i| \leq 2\varepsilon_1 \|B_{i*}\|, \qquad i=1,2,\ldots,m+1,$$

$$\|\gamma\| \leq 2\varepsilon_1 \|B\|_E, \tag{15}$$

и равенство (6) примет вид

$$\nu^{(k+1)} = \varkappa_1 \nu^{(k)} + \varkappa_2 \alpha \gamma' + \zeta, \tag{16}$$

где γ' уже не зависит от $\|c_p\|$ (γ' - часть γ : $\gamma = \begin{bmatrix} \gamma' \\ \gamma_0 \end{bmatrix}$, $\gamma' \in R^m$).

Таким образом, схему алгоритма К можно изменить: вычислить предварительно вектор $Y = (BB^T)^{-1} B\mathcal{D}_0 c$, а на каждой итерации вычислять c_p, используя имеющийся к данному моменту Y с последующим одновременным уточнением c_p и Y. Причем уточнение Y будет заканчиваться после достижения оценки

$$\|BB^T Y - B\mathcal{D}_k c\| \leq 2\varepsilon_1 \|BB^T\|_E \|Y\|,$$

а уточнение c_p - после выполнения (14).

Перед вычислением c_p и его уточнением можно изменить Y_{m+1}, так как известно, что его точное значение на $(k+1)$-й итерации есть $Y_{m+1} = \dfrac{c^T x^{(k)}}{n}$.

3.2. <u>Устранение накопленной погрешности</u>. Из равенств (6) и (16) видно, что $\nu^{(k+1)}$ зависит от накопленной ранее погрешности $\nu^{(k)}$. Эту погрешность также можно устранить путем внесения определенной поправки в вычисления.

Если в п.К.3^0 формулу (2) заменить на

$$b' = \frac{1}{n}(e - B^T \tilde{z}) - \frac{\alpha}{\sqrt{n(n-1)}} \frac{c_p}{\|c_p\|},$$

где $B'B^T\tilde{z} = \begin{bmatrix} \nu^{(k)} \\ 0 \end{bmatrix}$ (для алгоритма М соответственно $BE^2B^T\tilde{z} = \begin{bmatrix} \nu^{(k)} \\ 0 \end{bmatrix}$), то, повторив выкладки леммы I, получим равенство

$$\nu^{(k+1)} = \varkappa_2 \alpha \frac{1}{\|c_p\|} \ell + \zeta.$$

А если c_p вычисляется с уточнением, то (16) примет вид

$$\nu^{(k+1)} = \varkappa_2 \alpha \gamma' + \zeta ,$$

где γ' удовлетворяет (I5).

В исходном алгоритме без уточнения C_p введение поправки не приведет к уменьшению погрешности $\nu^{(k+1)}$, так как сохраняется ее главная часть $\varkappa_2 \frac{\alpha}{\|C_p\|} \eta$. Но таким образом можно поправить полученное решение $\tilde{x}$. Если $A\tilde{x} = \tilde{\nu}$, то, вычислив $b' = \frac{1}{n}(e - B^T \tilde{z})$ и $\tilde{\tilde{x}} = \frac{\tilde{D} b'}{e^T \tilde{D} b'}$ (где $\tilde{D} = diag\{\tilde{x}_1, \ldots, \tilde{x}_n\}$, $B = \left[\frac{A\tilde{D}}{e^T}\right]$, $\tilde{z} = (BB^T)^{-1}\left[\frac{\tilde{\nu}}{0}\right]$), получим $\tilde{\tilde{x}}$ такое, что $A\tilde{\tilde{x}} = \tilde{\tilde{\nu}}$ и $\|\tilde{\tilde{\nu}}\| \leq 2\varepsilon_1 \|A\|_E \|\tilde{\tilde{x}}\|$.

Конечно, погрешность $\tilde{\nu}$ должна быть все-таки достаточно мала, чтобы вычисленный вектор b', а значит, и $\tilde{\tilde{x}}$, был неотрицательным.

В алгоритме с уточнением C_p устранение накопленной погрешности целесообразно проводить, когда $\|\nu^{(k)}\| > 2\varepsilon_1 \|A\|_E \|x^{(k)}\|$.

3.3. <u>Учет погрешности в обратной матрице</u>. Рассмотрим, как меняется погрешность матрицы $\mathcal{M}_1$, используюшейся вместо $\mathcal{M}^{-1}$, при проведении одной модификации по формуле (3). Пусть до проведения модификации $(\mathcal{M}\mathcal{M}_1)_{выч} = I + Q$, где матрица Q характеризует погрешность $\mathcal{M}_1$, I - единичная $m \times m$ -матрица.

В формуле (3) используется вектор $\mathcal{M}_1 A_{*i}$, вместо него введем вектор $\tilde{d}$, который удовлетворяет равенству $\mathcal{M}\tilde{d} = A_{*i} + \tau$, где $\tau \in R^m$ и является погрешностью, с которой $\tilde{d}$ удовлетворяет системе $\mathcal{M} \cdot \mu = A_{*i}$ $(\mu \in R^m)$.

Модификация $\mathcal{M}_1$ заключается в вычислении

$$\bar{\mathcal{M}}_1 = \mathcal{M}_1 - \frac{g_i \tilde{d} \tilde{d}^T}{1 + g_i A_{*i}^T \tilde{d}} .$$

Это соответствует преобразованию $\mathcal{M}$ в $\bar{\mathcal{M}} = \mathcal{M} + g_i A_{*i} A_{*i}^T$.

ЛЕММА 2. Матрица погрешностей $\bar{Q} = (\bar{\mathcal{M}}\bar{\mathcal{M}}_1)_{выч} - I$ удовлетворяет равенству

$$\bar{Q} = Q + Q_1 + Q_2 + \tilde{Q} ,$$

где $\tilde{Q}$ - погрешность машинного произведения матриц $\bar{\mathcal{M}}$ и $\bar{\mathcal{M}}_1$.

$$Q_1 = g_i A_{*i}(A_{*i}^T \mathcal{M}_1 - \tilde{d}^T),$$

$$Q_2 = -\frac{g_i \tau \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}}.$$

ДОКАЗАТЕЛЬСТВО. Проводится путем непосредственного перемножения $\bar{\mathcal{M}}, \bar{\mathcal{M}}_1$ с учетом того, что при машинном вычислении произведения матриц получим некоторую погрешность $\tilde{Q}$. Имеем

$$(\bar{\mathcal{M}}\bar{\mathcal{M}}_1)_{выч} = I + Q + g_i A_{*i} A_{*i}^T \mathcal{M}_1 - \frac{g_i \mathcal{M} \tilde{d} \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} - g_i^2 A_{*i}^T \tilde{d}\,\frac{A_{*i}\tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} + \tilde{Q} =$$

$$= I + Q + g_i A_{*i} A_{*i}^T \mathcal{M}_1 - \frac{g_i A_{*i} \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} - \frac{g_i \tau \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} - g_i^2 A_{*i}^T \tilde{d}\,\frac{A_{*i}\tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} + \tilde{Q} =$$

$$= I + Q + g_i A_{*i} A_{*i}^T \mathcal{M}_1 - g_i A_{*i}\,\frac{1+g_i A_{*i}^T \tilde{d}}{1+g_i A_{*i}^T \tilde{d}}\,\tilde{d}^T - \frac{g_i \tau \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} + \tilde{Q} =$$

$$= I + Q + g_i A_{*i}(A_{*i}^T \mathcal{M}_1 - \tilde{d}^T) - \frac{g_i \tau \tilde{d}^T}{1+g_i A_{*i}^T \tilde{d}} + \tilde{Q} = I + Q + Q_1 + Q_2 + \tilde{Q}. \ \blacksquare$$

Таким образом, новая матрица погрешностей $\bar{Q}$ имеет несколько составляющих. Составляющая Q_2 зависит от τ-погрешности вектора $\tilde{d}$. Используя метод итераций, можно добиться, чтобы

$$\|\tau\| \leq 2\varepsilon_1 \|\mathcal{M}\|_E \|\tilde{d}\|. \qquad (17)$$

Это наименьшая оценка погрешности τ при машинных вычислениях. Ей будет соответствовать наименьшее по абсолютной величине значение всех элементов матрицы Q_2.

Но, уточняя $\tilde{d}$, мы увеличиваем абсолютную величину элементов составляющей Q_1, так как тогда увеличивается различие между векторами $\tilde{d}^T$ и $A_{*i}^T \mathcal{M}_1$.

С другой стороны, составляющую Q_1 можно устранить, взяв $\tilde{d} = \mathcal{M}_1 A_{*i}$ и проводя вычисления $\mathcal{M}_1$ таким образом, чтобы обеспечить равенство $\mathcal{M}_1 = \mathcal{M}_1^T$, т.е. не допуская вычислительных погрешностей, нарушающих симметрию $\mathcal{M}_1$. Но такое вычисление $\tilde{d}$ может привести к росту по абсолютной величине элементов τ и соответственно элементов матрицы Q_2.

Одновременно уменьшить абсолютные величины элементов Q_2 и устранить Q_1 можно, если проводить вычисление $\bar{\mathcal{M}}_1$ по формуле

$$\bar{\mathcal{M}}_1 = \mathcal{M}_1 - \frac{g_i \tilde{\mathcal{X}} A_{*i}^T \mathcal{M}_1}{1 + g_i A_{*i}^T \tilde{\mathcal{X}}}, \qquad (18)$$

где $\tilde{\mathcal{X}}$ - уточненный вектор, для которого верна оценка (17).

Повторив выкладки леммы 2 для формулы (18), получим $\bar{Q} = Q - Q_2 + \tilde{Q}$. Таким образом, вычисления по формуле (18) позволяют стабилизировать погрешность $\bar{Q} = (\bar{\mathcal{M}}, \bar{\mathcal{M}}_1)_{выч} - I$, хотя и ведут к нарушению симметрии $\bar{\mathcal{M}}_1$. Но в алгоритме важно, чтобы произведение $\bar{\mathcal{M}}\bar{\mathcal{M}}_1$ было близко к единичной матрице, и нигде не используется произведение $\bar{\mathcal{M}}_1 \bar{\mathcal{M}}$, поэтому нарушение симметрии $\bar{\mathcal{M}}_1$ не влияет на работу алгоритма.

3.4. <u>Алгоритм, учитывающий погрешность машинных вычислений.</u>

КТ.1^0. Начальное состояние. Заданы

$$x^{(0)},\ \mathcal{D}_0 = diag\{x_1^{(0)}, \ldots, x_n^{(0)}\},$$

$$B = \left[\frac{A\mathcal{D}_0}{e^T}\right],$$

$$Y = (BB^T)^{-1} B\mathcal{D}_0 c,$$

$$k = 0.$$

КТ.2^0. Определение направление сдвига.

КТ.2^0.1.

$$B = \left[\frac{A\mathcal{D}_k}{e^T}\right],$$

$$Y_{m+1} = \frac{c^T x^{(k)}}{n},$$

$$c_p = \mathcal{D}_k c - B^T Y,$$

$$\left[\frac{\ell}{\ell_{m+1}}\right] = B c_p .$$

Пока выполняется условие

$$\left\| \frac{\ell}{\ell_{m+1}} \right\| > 2\varepsilon_1 \|B\|_E \|c_p\|,$$

повторять следующие вычисления:

$$\tilde{z}=(BB^T)^{-1}\left[\frac{\ell}{\ell_{m+1}}\right];$$

$$c_p=c_p-B^T\tilde{z};$$

если $\left\|\frac{\ell}{\ell_{m+1}}\right\|>2\varepsilon_1\|BB^T\|_E\|Y\|$, то $Y=Y+\tilde{z};\ \left[\frac{\ell}{\ell_{m+1}}\right]=Bc_p$.

КТ.2^0.2. $\gamma=\frac{1}{\|c_p\|}\left[\frac{\ell}{\ell_{m+1}}\right]$.

Если $|Y^T\gamma|>\|c_p\|$, то перейти к КТ.4^0.

КТ.3^0. Вычисление следующей точки. Пусть

$$\tilde{z}=0;$$

$$v^{(k)}=Ax^{(k)};$$

если $\|v^{(k)}\|>2\varepsilon_1\|A\|_E\|x^{(k)}\|$, то $\tilde{z}=(BB^T)^{-1}\left[\frac{v^{(k)}}{0}\right]$;

$$b'=\frac{1}{n}(e-B^T\tilde{z})-\frac{\alpha}{\sqrt{n(n-1)}}\frac{c_p}{\|c_p\|};$$

$$x^{(k+1)}=\frac{\mathcal{D}_k b'}{e^T\mathcal{D}_k b'};$$

$$\mathcal{D}_{k+1}=diag\{x_1^{(k+1)},\dots,x_n^{(k+1)}\}.$$

КТ.4^0. Завершение. Если $|Y^T\gamma|>\|c_p\|$, то завершить работу алгоритма; полученное приближение $\tilde{x}=x^{(k)}$ является наилучшим при данной точности вычислений. Если $c^Tx^{(k+1)}/c^Tx^{(0)}\leq$ $\leq 2^{-q}$, то завершить работу алгоритма, за решение принимается $\tilde{x}=x^{(k+1)}$. Если $|Y^T\gamma|\leq\|c_p\|$ и $c^Tx^{(k+1)}/c^Tx^{(0)}>2^{-q}$, то положить $k=k+1$ и перейти к п.КТ.2^0.

Точно так же преобразуется алгоритм М в алгоритм МТ'. МТ повторяет КТ, только вместо BB^T используется матрица BE^2B^T, вместо векторов $B\mathcal{D}_kc$, $\mathcal{D}_kc$ используются векторы $BE^2\mathcal{D}_kc$, $E^2\mathcal{D}_kc$ соответственно и добавляется п.МТ.3б, аналогичный М.3б, где формула (3) заменена на (18).

Кроме того, в МТ вместо условия $|Y^T\gamma| > \|c_p\|$ используется условие (I3).

ЗАМЕЧАНИЕ. В данных алгоритмах с уточнениями проверку условий $|Y^T\gamma| > \|c_p\|$ и (I3) в пп.КТ.4^0 и МТ.4^0 можно заменить проверкой неравенства $|c^T x^{(k+1)}| < 2\varepsilon_1 \|c\| \, \|x^{(k+1)}\|$ и при его выполнении положить $\hat{x} = x^{(k+1)}$.

§4. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на задачах вида

$$\min\{c^T x \mid Ax = 0,\ e^T x = 1,\ x \geq 0\} \qquad \text{(I9)}$$

с выполнением условий $c^T x^* = 0$, $x^{(0)} = \frac{1}{n} e$. Задачи генерировались с помощью алгоритма, предложенного в [4], с использованием датчика равномерно распределенных случайных чисел. Программы были написаны на языке ПЛ-I и реализованы на ЭВМ ЕС-I06I. Вычисления велись в арифметике двойной точности, т.е. $\varepsilon_1 = 2^{-52} = 2{,}220446 \cdot 10^{-16}$.

Рассматривались алгоритмы М и МТ с параметром $\alpha = 0{,}9$ и максимальным шагом h вместо шага $1/\sqrt{n(n-1)}$. Величина шага выбиралась из условия $\frac{1}{n} e - h \frac{c_p}{\|c_p\|} \geq 0$, т.е. $h = \frac{\|c_p\|}{n \cdot \max\limits_i (c_p)_i}$.

Результаты экспериментов содержатся в таблице, где приведены значения соответствующих величин на последней итерации алгоритма. В графе "$m \times n$" указаны размеры матрицы А, определяющей размеры задачи (I9).

Параметр q не задавался, и работа алгоритма завершалась, если выполнялось условие (I3). Как указывалось в п.2.2, выполнение условия (I3) свидетельствует, что дальнейшую правильную работу алгоритма гарантировать нельзя. Из графы "число итераций" видно, что в алгоритме М условие (I3) было выполнено гораздо раньше, чем в МТ. Соответствующая погрешность в вычисляемой функции (II) приведена в графе "$|Y^T\gamma|$".

Несмотря на небольшое количество итераций алгоритма М, погрешность в ограничениях (графа "$\|A\tilde{x}\|$") во много раз превзошла оценку $2\varepsilon_1 \|A\|_E \|\tilde{x}\|$, в то время как в МТ вычисления поправок по условию $\|Ax^k\| > 2\varepsilon_1 \|A\|_E \|x^{(k)}\|$ (п.МТ.3^0) не потребовалось, так как $\|Ax^{(k)}\|$ не выходила за пределы данной оценки. Число уточнений вектора c_p в алгоритме МТ на всех итерациях не больше трех.

	Алгоритм	Число итераций	$\Vert\tilde{x}-\tilde{x}^*\Vert$	$c^T\tilde{x}$	$\vert y^T\gamma\vert$	$\Vert C_p\Vert$	$\Vert A\tilde{x}\Vert$
I0x40	М	8	3,I IO^{-3}	5,I IO^{-4}	2,3 IO^{-4}	6,8 IO^{-5}	2,0 IO^{-7}
	МТ	26	2,6 IO^{-II}	3,5 IO^{-I2}	3,0 IO^{-I3}	2,2 IO^{-I3}	I,3 IO^{-I5}
I5x40	М	I2	I,7 IO^{-5}	5,8 IO^{-5}	3,4 IO^{-5}	I,2 IO^{-5}	9,3 IO^{-7}
	МТ	32	5,4 IO^{-I4}	2,4 IO^{-I3}	4,2 IO^{-I4}	3,9 IO^{-I4}	I,9 IO^{-I4}
20x80	М	I2	2,9 IO^{-4}	2,9 IO^{-4}	5,4 IO^{-5}	3,0 IO^{-5}	5,7 IO^{-7}
	МТ	32	4,3 IO^{-I3}	3,I IO^{-I3}	4,0 IO^{-I4}	4,0 IO^{-I4}	6,9 IO^{-I5}
30x400	М	8	I,8 IO^{-2}	I,5 IO^{-2}	6,4 IO^{-4}	3,I IO^{-5}	7,7 IO^{-7}
	МТ	24	5,6 IO^{-II}	4,I IO^{-II}	8,5 IO^{-I3}	7,I IO^{-I3}	I,0 IO^{-I3}
35xI45	М	9	I,7 IO^{-3}	6,2 IO^{-3}	4,I IO^{-3}	7,I IO^{-4}	I,I IO^{-5}
	МТ	26	I,3 IO^{-II}	2,0 IO^{-I0}	I,9 IO^{-II}	I,4 IO^{-II}	2,3 IO^{-I4}
50 I50	М	5	3,I IO^{-2}	I,8 IO^{-2}	I,I IO^{-3}	I,I IO^{-3}	2,0 IO^{-6}
	МТ	26	4,9 IO^{-I0}	2,2 IO^{-II}	7,5 IO^{-I2}	4,3 IO^{-I2}	3,7 IO^{-II}
50x200	М	9	4,7 IO^{-4}	7,2 IO^{-3}	2,6 IO^{-3}	6,4 IO^{-4}	2,2 IO^{-5}
	МТ	28	4,4 IO^{-I3}	I,3 IO^{-II}	2,4 IO^{-I2}	2,2 IO^{-I2}	2,4 IO^{-I3}
I00x200	М	II	I,8 IO^{-3}	6,4 IO^{-3}	2,2 IO^{-3}	8,I IO^{-4}	2,I IO^{-6}
	МТ	35	I,9 IO^{-I2}	4,3 IO^{-II}	I,4 IO^{-I2}	3,4 IO^{-I3}	I,3 IO^{-I4}

Алгоритм МТ позволяет получить значительно более точное решение по сравнению с М, что видно по значениям $c^T\tilde{x}$ и $\|\tilde{x} - x^*\|$.

В обоих алгоритмах модификации $\mathcal{M}_1$ проводились по формуле (I8) и на всех итерациях погрешность вектора y удовлетворяла оценке (I0).

Автор благодарит В.А.Булавского за постановку задачи и внимание к работе.

Литература

I. Karmarkar N. A new polinomial-time algorithm for linear programming// Combinatorica. - I984. - V.4, N 4.- P.373-395.

2. Андрусенко С.К., Нурминский Е.А., Стецюк П.И. Численные эксперименты с новым классом алгоритмов ь линейном программировании// Журн. вычисл. математики и математич. физики. - I987. - Т.27, №3. - С.394-356.

3. Годунов С.К. Решение систем линейных уравнений. - Новосибирск: Наука, I980.

4. Немировский А.С. Об одном алгоритме типа Кармаркара// Изв. АН СССР. Техн. кибернетика. - I987. - №I. - С.I05-II8.

Поступила в ред.-изд. отдел