

УДК 512.25/26

ОБ АЛГОРИТМАХ МЕТОДА ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ
ДЛЯ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ

В.И. Шмырёв

В настоящей работе рассматриваются вопросы алгоритмической конкретизации для задач квадратичного программирования метода последовательного улучшения, общая схема которого была предложена в [1]. Подробно излагается один из алгоритмов, который можно рассматривать как обобщение известного модифицированного симплекс-метода для решения задач линейного программирования. По сравнению с другими алгоритмами такого типа для задач квадратичного программирования, предлагаемый алгоритм отличается простотой и компактностью вычислительной схемы. Изложение иллюстрируется на численном примере.

§ I. Общая схема метода

Будем рассматривать задачу квадратичного программирования в следующей постановке.

Определить вектор $x = (x_1, \dots, x_n)$, минимизирующий функцию

$$f(x) = (p, x) + \frac{1}{2}(x, Cx) \quad (1)$$

при условиях

$$Ax = b, \quad (2)$$

$$x \geq 0. \quad (3)$$

Здесь $p=(p_1, \dots, p_n)$ и $v=(v_1, \dots, v_m)$ - заданные векторы, а $A=\{a_{ij}\}$ и $C=\{c_{ij}\}$ - заданные матрицы размером $m \times n$ и $n \times n$ соответственно, причем матрица C симметрична и неотрицательно определена.

Общая схема метода последовательного улучшения, изложенного в [1], может быть описана следующим образом.

Пусть к началу k -го шага процесса нам известен некоторый допустимый, т.е. удовлетворяющий условиям (2-3), вектор $x^k = (x_1^k, x_2^k, \dots, x_n^k)$. Через $J(x^k)$ будем обозначать множество

$$J(x^k) = \{j \in N : x_j^k > 0\},$$

где $N = \{1, 2, \dots, n\}$. Пусть также известно некоторое множество J_k такое, что $J(x^k) \subset J_k \subset N$ и среди векторов $A^j = (a_{1j}, \dots, a_{mj})$, $j \in J_k$, имеется m линейно независимых.

Рассматривается задача минимизации функции f на аффинном многообразии

$$L(J_k) = \{x \in R^n : \sum_{j \in J_k} A^j x_j = v, x_j = 0, j \notin J_k\}.$$

Заметим, что $x^k \in L(J_k)$. Ниже будет показано, что если функция (1) на аффинном многообразии $L(J_k)$ не достигает минимума, то существует вектор $g \in R^n$, обладающий свойствами:

- 1) луч $\Lambda = \{x \in R^n : x = x^k + tg, t \geq 0\}$ принадлежит $L(J_k)$;
- 2) на любом луче с направляющим вектором g , в частности, на луче Λ , функция (1) строго убывает.

Вопрос нахождения точки минимума функции f на $L(J_k)$ или вектора g с указанными свойствами будут рассмотрены ниже.

Таким образом, возможен один из следующих случаев:

- а) обнаружено, что минимум функции f на аффинном многообразии $L(J_k)$ реализуется в точке x^k ;
- б) определена точка \bar{x}^k , доставляющая минимум функции f на $L(J_k)$, причем $f(\bar{x}^k) < f(x^k)$;
- в) определен вектор $g \in R^n$, обладающий свойствами 1) и 2).

Если имеет место первый из этих случаев, то существует вектор $y = (y_1, \dots, y_m)$, удовлетворяющий системе линейных уравнений

$$(y, A^j) = \frac{\partial f}{\partial x_j}(x^k), \quad j \in J_k. \quad (4)$$

Так как по предположению среди векторов A^j , $j \in J_k$, имеется m линейно независимых, то указанный вектор y определяется единственным образом. Если для него выполняются условия

$$(y, A^j) \leq \frac{\partial f}{\partial x_j}(x^k), \quad j \in N \setminus J_k, \quad (5)$$

то точка x^k , согласно признаку оптимальности для задач выпуклого программирования, является решением исходной задачи, и процесс окончен.

В противном случае находим $j_0 \in N \setminus J_k$ такой, что

$$(y, A^{j_0}) - \frac{\partial f}{\partial x_{j_0}}(x^k) > 0,$$

и переходим к следующему шагу, полагая $x^{k+1} = x^k$, $J_{k+1} = J_k \cup \{j_0\}$. При этом (см. [1]) будет

$$\inf_{x \in L(J_{k+1})} f(x) < f(x^{k+1}),$$

и, следовательно, на $(k+1)$ -ом шаге будет иметь место случай в) или случай с).

Если на k -ом шаге процесса реализуется случай в) и компоненты вектора \bar{x}^k неотрицательны, то принимаем $x^{k+1} = \bar{x}^k$,

$J_{k+1} = J_k$ и переходим к следующему шагу, на котором будет иметь место случай а). Если же среди компонент вектора \bar{x}^k имеются отрицательные, то полагаем $g = \bar{x}^k \cdot x^k$ и находим максимальное t , равное t_0 , при котором выполняются условия

$$x_j^k + t g_j \geq 0, \quad j \in N,$$

т.е.

$$t_0 = \min_{g_j < 0} \left\{ -\frac{x_j^k}{g_j} \right\} = -\frac{x_{j'}^k}{g_{j'}}. \quad (6)$$

после чего переходим к следующему шагу, принимая $x^{k+1} = x^k + t_0 g$, $J_{k+1} = J_k \setminus \{j'\}$.

Пусть теперь реализуется случай с). Если при этом оказалось, что все компоненты вектора g неотрицательны, то любая точка на луче Λ соответствует некоторому допустимому вектору и, как легко видеть, оптимального вектора в исходной задаче не существует.

Если же среди компонент вектора g имеются отрицательные, то поступаем, как и в случае в).

Отметим, что при исключении элемента j' из множества J_k в случаях в) и с), ввиду того, что $g_{j'} \neq 0$, среди векторов A^j , $j \in J_{k+1}$, по-прежнему будет m линейно независимых.

Как показано в [1], описанный процесс заканчивается через конечное число шагов при условии невырожденности ограничений

(2-3), т.е. когда любой допустимый вектор имеет не менее m отличных от нуля компонент.

Применение изложенной схемы метода последовательного улучшения в случае квадратичного программирования характеризуется той особенностью, что вспомогательная задача минимизации функции f на аффинном многообразии $L(J_k)$, решаемая на k -ом шаге процесса, допускает точное решение, а именно: упомянутая задача введением множителей Лагранжа $y_i (i=1, \dots, m)$ сводится к рассмотрению следующей системы линейных алгебраических уравнений :

$$\sum_{j \in J_k} a_{ij} x_j = b_i, \quad i=1, \dots, m$$

$$\sum_{i=1}^m a_{ij} y_i - \sum_{i \in J_k} c_{ij} x_i = p_j, \quad j \in J_k. \quad (7)$$

Это обстоятельство и позволяет получить для задач квадратичного программирования эффективные алгоритмы метода последовательного улучшения.

Для того, чтобы обосновать возможность применения описанной схемы к задачам квадратичного программирования, следует показать, что если функция f не достигает минимума на некотором аффинном многообразии $L(J_k)$, то найдется вектор g , обладающий упомянутыми свойствами 1) и 2).

Для удобства изложения введем следующие обозначения. Пусть

$$D^j = (a_{1j}, \dots, a_{mj}, -c_{1j}, \dots, -c_{nj}),$$

$$C^j = (c_{1j}, \dots, c_{nj}).$$

Для произвольного вектора $w \in R^{m+n}$ через w_x и w_y будем обозначать векторы

$$w_x = (w_{m+1}, \dots, w_{m+n}),$$

$$w_y = (w_1, \dots, w_m).$$

Используя эти обозначения, можно записать: $D_x^j = -C^j$, $D_y^j = A^j$. Далее, для произвольного вектора $x \in R^n$ и некоторого упорядоченного множества $J \subset N$,

$$J = \{j_1, \dots, j_s\},$$

через $x(J)$ будем обозначать следующий вектор из R^v :

$$x(J) = (x_{j_1}, \dots, x_{j_s}).$$

Аналогично. для $w \in R^{m+n}$ через $w(\mathcal{J})$ будем обозначать вектор, составленный из векторов w_j и $w_{\mathcal{J}}(\mathcal{J})$, т.е. вектор

$$w(\mathcal{J}) = (w_1, \dots, w_m, w_{m+j_1}, \dots, w_{m+j_s}).$$

Кроме того, через $A(\mathcal{J})$ будем обозначать матрицу из векторов A^j , $j \in \mathcal{J}$, через $C(\mathcal{J})$ - матрицу с элементами c_{ij} , $i, j \in \mathcal{J}$, а через $H(\mathcal{J})$ - матрицу следующей структуры

$$H(\mathcal{J}) = \begin{pmatrix} 0 & A(\mathcal{J}) \\ A^T(\mathcal{J}) & -C(\mathcal{J}) \end{pmatrix}.$$

Под $H^{-1}(\mathcal{J})$ будем понимать матрицу $(H(\mathcal{J}))^{-1}$.

В дальнейшем предполагаем множество \mathcal{J}_k некоторым образом упорядоченным. Тогда, как легко видеть, $H(\mathcal{J}_k)$ представляет собой матрицу системы (7).

ЛЕММА I. Если ранг матрицы $A(\mathcal{J})$ равен m и для некоторых векторов λ' и λ'' выполняются условия:

$$A(\mathcal{J})\lambda'' = 0, \quad (8)$$

$$A^T(\mathcal{J})\lambda' - C(\mathcal{J})\lambda'' = 0, \quad (9)$$

то $\lambda' = 0$ и

$$\sum_{j_i \in \mathcal{J}} \lambda''_{j_i} C^{j_i} = 0. \quad (10)$$

ДОКАЗАТЕЛЬСТВО. Вычитая из (8), умноженного скалярно на λ' , соотношение (9), умноженное на λ'' , получим

$$(\lambda'', C(\mathcal{J})\lambda'') = 0. \quad (11)$$

Образует вектор $g \in R^n$ с компонентами

$$\begin{aligned} g_{j_i} &= \lambda''_{j_i}, & j_i \in \mathcal{J}, \\ g_{j_i} &= 0, & j_i \notin \mathcal{J}. \end{aligned} \quad (12)$$

Для этого вектора $(g, Cg) = (\lambda'', C(\mathcal{J})\lambda'') = 0$, что ввиду симметричности и неотрицательной определенности матрицы C , эквивалентно $Cg = 0$, т.е. выполняется требуемое равенство (10). Остается показать, что $\lambda' = 0$. Из (9) и (10) следует: $A^T(\mathcal{J})\lambda' = 0$, и так как ранг матрицы $A(\mathcal{J})$ равен m , то $\lambda' = 0$.

Предположим теперь, что на некотором аффинном многообразии $L(\mathcal{J}_k)$, где \mathcal{J}_k такое, что матрица $A(\mathcal{J}_k)$ имеет ранг m , функция f не достигает минимума. Это означает, что система (7) не имеет решения. Следовательно, существуют векторы λ' и λ'' , удовлетворяющие при $\mathcal{J} = \mathcal{J}_k$ системе (8-9) и такие, что

$$(\lambda', \theta) + (\lambda'', \rho(\mathcal{J}_k)) < 0.$$

Тогда по лемме I $\lambda' = 0$. Образовав вектор $g \in R^n$ с компонентами (12), будем иметь

$$\sum_{j \in \mathcal{J}_k} A^j g_j = 0. \quad (13)$$

$$\sum_{j \in \mathcal{J}_k} C^j g_j = 0, \quad (14)$$

$$(p, g) < 0. \quad (15)$$

Покажем теперь, что производная функции f по направлению, определяемому вектором g , не зависит от точки, в которой эта производная вычисляется, и отрицательна. Без ограничения общности можно считать, что вектор g единичной длины. Тогда

$$\frac{\partial f}{\partial g}(x^0) = (Cx^0, g) + (p, g) = (x^0, Cg) + (p, g).$$

Но из (14) и (12) следует $Cg = 0$. Поэтому

$$\frac{\partial f}{\partial g}(x^0) = (p, g) < 0.$$

Кроме того, если $x^0 \in L(\mathcal{J}_k)$, то из (12) и (13) следует, что и луч с вершиной в точке x^0 и направляющим вектором g принадлежит $L(\mathcal{J}_k)$. Следовательно, вектор g является требуемым.

§ 2. Алгоритм метода с использованием обратной матрицы

Изложенная в предшествующем параграфе схема метода последовательного улучшения допускает различные конкретизации при решении вспомогательной задачи минимизации функции f на аффинном многообразии. В зависимости от применяемого алгоритма для решения этой задачи будут получаться различные алгоритмы метода последовательного улучшения. В настоящем параграфе мы остановимся на таком алгоритме, в котором для определения точки минимума функции f на аффинном многообразии $L(\mathcal{J}_k)$ или луче, на котором f строго убывает, используется корректируемая шаг за шагом обратная матрица системы (7) (или некоторой

усеченной её подсистемы). Этот алгоритм является аналогом известного модифицированного симплекс-метода для задач линейного программирования (см. [2], стр. 208-223).

Рассмотрим подробно один шаг процесса.

Пусть

$$\begin{aligned} \omega &= (y_1, \dots, y_m, x_1, \dots, x_n), \\ B &= (b_1, \dots, b_m, p_1, \dots, p_n). \end{aligned}$$

Используя эти обозначения, систему линейных уравнений (7) для определения точки минимума функции f на аффинном многообразии $L(J_k)$ можно записать в виде

$$H(J_k) \omega(J_k) = B(J_k). \quad (7')$$

1⁰. Предположим, что матрица $H(J_k)$ неособенная и известна матрица $H^{-1}(J_k)$. Пусть, кроме того, известен вектор ω^k такой, что вектор $\omega^k(J_k)$ является решением системы (7'), т.е.

$$\omega^k(J_k) = H^{-1}(J_k) B(J_k),$$

причем вектор $\omega_{m+j}^k \geq 0, j \in J_k$, и $\omega_{m+j}^k = 0, j \notin J_k$. Ясно, что в этом случае вектор $x^k = \omega_x^k$ является допустимым вектором исходной задачи и доставляет минимум функции f на аффинном многообразии $L(J_k)$, т.е. реализуется случай а) — описание схемы метода. Системе (4) для определения вектора y , фигурирующего в критерии оптимальности, будет удовлетворять вектор $y = \omega_y^k$. Поэтому система неравенств (5) критерия оптимальности может быть переписана в виде

$$(\omega_y^k, A^j) \leq (\omega_x^k(J_k), C^j(J_k)) + p_j, \quad j \in N \setminus J_k,$$

т.е.

$$(\omega^k(J_k), D^j(J_k)) \leq p_j, \quad j \in N \setminus J_k. \quad (5')$$

Если все эти неравенства выполняются, то вектор x^k является решением исходной задачи. В противном случае, определив некоторое j_0 , для которого

$$\Delta_{j_0} = (\omega^k(J_k), D^{j_0}(J_k)) - p_{j_0} > 0, \quad (16)$$

и приняв $J_{k+1} = J_k \cup \{j_0\}$, $x^{k+1} = x^k$, мы должны перейти к следующему шагу метода. В рассматриваемом варианте алгоритма следует положить $\omega^{k+1} = \omega^k$ и перейти к выполнению п. 2⁰.

2⁰. Будем считать, что упорядоченное множество J_{k+1} получается из J_k добавлением j_0 в качестве последнего элемента.

Тогда вектор $w^{k+1}(\mathcal{F}_{k+1})$ имеет следующую структуру

$$w^{k+1}(\mathcal{F}_{k+1}) = \begin{pmatrix} w^{k+1}(\mathcal{F}_k) \\ x_{j_0}^{k+1} \end{pmatrix}$$

и удовлетворяет системе

$$H(\mathcal{F}_k) w^{k+1}(\mathcal{F}_k) + D^{j_0}(\mathcal{F}_k) x_{j_0}^{k+1} = B(\mathcal{F}_k), \quad (17)$$

$$(D^{j_0}(\mathcal{F}_k), w^{k+1}(\mathcal{F}_k)) - c_{j_0 j_0} x_{j_0}^{k+1} = p_{j_0} + \Delta_{j_0}. \quad (18)$$

При этом если п. 2⁰ выполняется непосредственно после п. 1⁰, то $x_{j_0}^{k+1} = 0$.

Аналогично матрица $H(\mathcal{F}_{k+1})$ получается окаймлением матрицы $H(\mathcal{F}_k)$:

$$H(\mathcal{F}_{k+1}) = \left(\begin{array}{c|c} H(\mathcal{F}_k) & h \\ \hline h^T & -c_{j_0 j_0} \end{array} \right),$$

где $h = D^{j_0}(\mathcal{F}_k)$.

Если $H(\mathcal{F}_{k+1})$ неособенная, то обратная матрица $H^{-1}(\mathcal{F}_{k+1})$ может быть получена из $H^{-1}(\mathcal{F}_k)$ по формулам метода окаймления (см. [3], стр. 187-192):

$$H^{-1}(\mathcal{F}_{k+1}) = \left(\begin{array}{c|c} \bar{H} & \frac{1}{z} d \\ \hline \frac{1}{z} d^T & -\frac{1}{z} \end{array} \right),$$

где $d = H^{-1}(\mathcal{F}_k) D^{j_0}(\mathcal{F}_k)$, $z = c_{j_0 j_0} + (D^{j_0}(\mathcal{F}_k), d)$, $\bar{H} = H^{-1}(\mathcal{F}_k) - \frac{1}{z} d d^T$.

А. Получим вектор d и величину z .

а) В случае $z = 0$ матрица $H(\mathcal{F}_{k+1})$ будет особенной. Покажем, что в этом случае функция f не достигает минимума на аффинном многообразии $L(\mathcal{F}_{k+1})$.

Вектор d очевидным образом удовлетворяет такой системе

$$-H(\mathcal{F}_k) d + D^{j_0}(\mathcal{F}_k) = 0, \quad (19)$$

$$-(D^{j_0}(\mathcal{F}_k), d) - c_{j_0 j_0} = -z. \quad (20)$$

Введем векторы $\lambda' \in R^m$ и $\lambda'' \in R^{\ell+1}$, где ℓ - число элементов в множестве \mathcal{F}_k , следующим образом.

$$\lambda'_i = -d_i, \quad i = 1, \dots, m,$$

$$\lambda''_j = -d_{m+j}, \quad j = 1, \dots, \ell,$$

$$\lambda_{i+m}'' = 1.$$

Так как $\gamma = 0$, то из (19-20) следует, что векторы λ' и λ'' удовлетворяют условиям леммы I предшествующего параграфа при $\tilde{J} = \tilde{J}_{k+1}$. Поэтому $\lambda' = 0$ и вектор $g \in R^n$ с компонентами (12), т.е.

$$\begin{aligned} g_{ji} &= -d_{m+i}, & j_i \in \tilde{J}_k, \\ g_{j_0} &= 1, \\ g_{ji} &= 0, & j_i \in \tilde{J}_k \cup \{j_0\}, \end{aligned} \quad (21)$$

удовлетворяет условиям (13-14). Чтобы показать, что вектор g определяет направление, не выводящее из $L(\tilde{J}_{k+1})$ и такое, что функция f строго убывает по этому направлению в любой точке, нужно лишь проверить выполнение неравенства (15).

Умножая равенство (17) скалярно на вектор d и вычитая его из (18), получим, ввиду (19-20):

$$0 = -(B(\tilde{J}_k), d) + p_{j_0} + \Delta_{j_0}.$$

Остается лишь заметить, что

$$\begin{aligned} -(B(\tilde{J}_k), d) + p_{j_0} &= \sum_{i=1}^m \beta_i \lambda'_i + \sum_{j_i \in \tilde{J}_k} p_{j_i} \lambda''_i + p_{j_0} = \\ &= \sum_{j \in \tilde{J}_k} p_j g_j + p_{j_0} = \sum_{j \in \tilde{J}_{k+1}} p_j g_j = \sum_{j=1}^n p_j g_j. \end{aligned}$$

Поэтому

$$\sum_{j=1}^n p_j g_j = -\Delta_{j_0} < 0,$$

и неравенство (15) выполняется.

Следовательно, функция f не достигает минимума на аффинном многообразии $L(\tilde{J}_{k+1})$, т.е. реализуется случай с) в описании схемы метода, а вектор g с компонентами (21) является требуемым.

β) Вернемся теперь к случаю $\gamma \neq 0$. В этом случае матрица $H(\tilde{J}_{k+1})$ неособенная и функция f достигает минимума на аффинном многообразии $L(\tilde{J}_{k+1})$. Покажем, что в этом случае вектор g с компонентами (21) определяет направление из точки x^{k+1} в точку минимума функции f на $L(\tilde{J}_{k+1})$.

Систему линейных уравнений для определения точки минимума функции f на $L(\tilde{J}_{k+1})$ в данном случае можно записать в следующем виде

$$H(\mathcal{J}_k) w(\mathcal{J}_k) + D^{j_0}(\mathcal{J}_k) x_{j_0} = B(\mathcal{J}_k), \quad (22)$$

$$(D^{j_0}(\mathcal{J}_k), w(\mathcal{J}_k)) - c_{j_0 j_0} x_{j_0} = p_{j_0}. \quad (23)$$

При этом $x_j = w_{m+j} = 0$, $j \in \mathcal{J}_{k+1}$, а искомой точкой минимума будет $x = w_x$.

Введем обозначения $\Delta w(\mathcal{J}_k) = w(\mathcal{J}_k) - w^{k+1}(\mathcal{J}_k)$, $\Delta x_{j_0} = x_{j_0} - x_{j_0}^{k+1}$. Заметим, что если п. 2⁰ выполняется непосредственно после п. 1⁰, то $x_{j_0}^{k+1} = 0$ и $\Delta x_{j_0} = x_{j_0}$. Из (22-23) и (17-18) получаем следующую систему для $\Delta w(\mathcal{J}_k)$ и Δx_{j_0} :

$$H(\mathcal{J}_k) \Delta w(\mathcal{J}_k) + D^{j_0}(\mathcal{J}_k) \Delta x_{j_0} = 0, \quad (24)$$

$$(D^{j_0}(\mathcal{J}_k), \Delta w(\mathcal{J}_k)) - c_{j_0 j_0} \Delta x_{j_0} = -\Delta p_{j_0}. \quad (25)$$

Из (24) следует

$$\Delta w(\mathcal{J}_k) = -H^{-1}(\mathcal{J}_k) D^{j_0}(\mathcal{J}_k) \Delta x_{j_0} = -d \Delta x_{j_0},$$

и, принимая во внимание (19), получим

$$w_x = x^{k+1} + \Delta x_{j_0} g.$$

Остается показать, что $\Delta x_{j_0} > 0$. Подставим выражение для $\Delta w(\mathcal{J}_k)$ в (25):

$$-\Delta x_{j_0} ((D^{j_0}(\mathcal{J}_k), d) + c_{j_0 j_0}) = -\Delta p_{j_0},$$

т.е.

$$\Delta x_{j_0} \gamma = \Delta p_{j_0} \quad (26)$$

Теперь для доказательства $\Delta x_{j_0} > 0$ достаточно показать, что $\gamma > 0$. А это следует из того, что $-1/\gamma$ является диагональным элементом η_{ii} , $i > m$, матрицы $H^{-1}(\mathcal{J}_{k+1})$, и следующей леммы.

ЛЕММА 2. Диагональные элементы η_{ii} , $i > m$, матрицы $H^{-1}(\mathcal{J})$, $\mathcal{J} \in N$, неположительны. При этом если $\eta_{ii} = 0$, $i > m$, то

$$\eta_x^i = 0,$$

где η^i — i -ая строка матрицы $H^{-1}(\mathcal{J})$.

ДОКАЗАТЕЛЬСТВО. Легко видеть, что

$$\eta_{ii} = (\eta^i, H(\mathcal{J}) \eta^i) = 2(\eta^i, A(\mathcal{J}) \eta_x^i) - (\eta_x^i, C(\mathcal{J}) \eta_x^i),$$

и так как $i > m$, то

$$A(\mathcal{J}) \eta_x^i = 0 \quad (27)$$

и

$$\eta_{ii} = -(\eta_x^i, C(\mathcal{J})\eta_x^i). \quad (28)$$

Но матрица C неотрицательно определена, и поэтому $\eta_{ii} \leq 0$. Предположим теперь, что $\eta_{ii} = 0$. Тогда из (28) следует

$$C(\mathcal{J})\eta_x^i = 0,$$

что вместе с (27) ввиду неособенности матрицы $H(\mathcal{J})$ возможно лишь при $\eta_x^i = 0$. Тем самым доказано, что $\Delta x_{j_0} > 0$ и, следовательно, вектор g определяет направление из точки x^{k+1} в точку минимума функции f на аффинном многообразии $L(\mathcal{J}_{k+1})$. При этом, как следует из (26), $x^{k+1} + tg$ будет являться точкой минимума при $t = \Delta j_0 / \gamma$.

В. Таким образом, для определения точки x^{k+2} следует смещаться из точки x^{k+1} в направлении вектора g с компонентами (21) вне зависимости от того, какой из двух случаев, — $\gamma = 0$ или $\gamma > 0$, реализуется. Следовательно, в соответствии с общей схемой метода следует определить максимальное t , равное t_0 , при условиях

$$x_j^{k+1} + tg_j \geq 0, \quad j \in \mathcal{J}_{k+1},$$

а в случае $\gamma > 0$ при дополнительном условии, вытекающем из (26),

$$t \leq \frac{\Delta j_0}{\gamma},$$

т.е.

$$t_0 = \min \left\{ \frac{\Delta j_0}{\gamma}, \min_{j \in \mathcal{J}_{k+1}} \frac{x_j^{k+1}}{g_j} \right\},$$

или, в других обозначениях,

$$t_0 = \min \left\{ \frac{\Delta j_0}{\gamma}, \min_{d_{m+i} > 0} \frac{w_{m+i}^{k+1}(\mathcal{J}_k)}{d_{m+i}} \right\}.$$

i) Если оказалось $t = +\infty$, т.е. все d_{m+i} неположительны и γ равно нулю, то функция f неограничена снизу на множестве допустимых векторов рассматриваемой задачи.

В противном случае получаем вектор $w^{k+2}(\mathcal{J}_{k+1})$, дополняя вектор $w^{k+1}(\mathcal{J}_k) - t_0 d$ компонентой $x_{j_0}^{k+2} = x_{j_0}^{k+1} + t_0$.

ii) Если $t_0 = \Delta j_0 / \gamma$, то $\mathcal{J}_{k+2} = \mathcal{J}_{k+1}$ и x^{k+2} доставляет минимум функции f на аффинном многообразии $L(\mathcal{J}_{k+2})$. Процесс

может быть повторен, начиная с п. 1⁰. Следует лишь предварительно получить матрицу $H^{-1}(\mathcal{J}_{k+2})$, что можно осуществить по упомянутым формулам метода окаймления, используя матрицу $H^{-1}(\mathcal{J}_k)$ и вектор d .

iii). Рассмотрим теперь случай, когда

$$t_0 = \frac{w_{m+i}^{k+1}(\mathcal{J}_k)}{d_{m+i}} < \frac{\Delta d_0}{2}$$

В этом случае элемент $j' = j'_i$ исключается из множества \mathcal{J}_{k+1} , т.е. $\mathcal{J}_{k+2} = \mathcal{J}_{k+1} \setminus \{j'\} = \{j_0\} \cup \mathcal{J}_k \setminus \{j'\}$. При этом будем считать, что множество \mathcal{J}_{k+2} получается из множества \mathcal{J}_k заменой элемента j' элементом j_0 без нарушения порядка остальных элементов. Для простоты изложения будем предполагать, что j' является последним элементом множества \mathcal{J}_k .

Пусть $\mathcal{J}'_k = \mathcal{J}_k \setminus \{j'\}$ и $\Delta'_{j_0} = \Delta_{j_0} - t_0 \tau$. Из (17-18) и (19-20) следует, что вектор $w^{k+2}(\mathcal{J}_{k+2})$ удовлетворяет системе (17-18), если заменить \mathcal{J}_k и Δ_{j_0} на \mathcal{J}'_k и Δ'_{j_0} соответственно, т.е. системе

$$H(\mathcal{J}'_k) w^{k+2}(\mathcal{J}'_k) + D^{j_0}(\mathcal{J}'_k) x_{j_0}^{k+2} = B(\mathcal{J}'_k), \quad (17')$$

$$(D^{j_0}(\mathcal{J}'_k), w^{k+2}(\mathcal{J}'_k)) - c_{j_0} x_{j_0}^{k+2} = p_{j_0} + \Delta'_{j_0}. \quad (18')$$

Следовательно, можно вернуться к п. 2⁰, получив предварительно матрицу $H^{-1}(\mathcal{J}'_k)$, если только $H(\mathcal{J}'_k)$ неособенная. Отличие от рассмотренного случая будет состоять лишь в том, что возможно $x_{j_0}^{k+2} \neq 0$.

α). Матрица $H(\mathcal{J}'_k)$ будет неособенной, если диагональный элемент η_{ss} , $s = m+i'$, матрицы $H^{-1}(\mathcal{J}_k)$ отличен от нуля и, следовательно, по лемме 2 отрицателен. Если считать, что j' является последним элементом в множестве \mathcal{J}_k , то матрица $H(\mathcal{J}_k)$ получается окаймлением матрицы $H(\mathcal{J}'_k)$. При этом обратная матрица $H^{-1}(\mathcal{J}_k)$ имеет структуру

$$H^{-1}(\mathcal{J}_k) = \begin{pmatrix} \tilde{H} & \eta \\ \eta^T & \eta_{ss} \end{pmatrix},$$

и, как следует из формул метода окаймления, для матрицы $H^{-1}(\mathcal{J}'_k)$ будем иметь

$$H^{-1}(\mathcal{J}'_k) = \tilde{H} - \frac{1}{\eta_{ss}} \eta \eta^T. \quad (29)$$

Следовательно, при $\eta_{ss} \neq 0$, получив матрицу $H^{-1}(\mathcal{J}'_k)$, про-

цесс можно возобновить, начиная с п.2⁰. При этом вектор $d' = H^{-1}(\mathcal{J}_k') D^{\delta_0}(\mathcal{J}_k')$ может быть получен подправкой вектора d следующим образом.

Пусть $\eta^s = \langle \eta, \eta_{ss} \rangle$ - последний столбец матрицы $H^{-1}(\mathcal{J}_k)$. Как следует из (29), матрица

$$L = H^{-1}(\mathcal{J}_k) - \frac{1}{\eta_{ss}} \eta^s (\eta^s)^T$$

будет иметь вид

$$L = \begin{pmatrix} H^{-1}(\mathcal{J}_k') & 0 \\ 0 & 0 \end{pmatrix}.$$

Поэтому компоненты вектора d' будут совпадать с первыми $s-1$ компонентами вектора $\bar{d}' = L D^{\delta_0}(\mathcal{J}_k)$, для которого имеем

$$\bar{d}' = (H^{-1}(\mathcal{J}_k) - \frac{1}{\eta_{ss}} \eta^s (\eta^s)^T) D^{\delta_0}(\mathcal{J}_k) = d - \frac{d_s}{\eta_{ss}} \eta^s. \quad (30)$$

Аналогично для $z' = c_{j_0 j_0} + (D^{\delta_0}(\mathcal{J}_k'), d')$ будет

$$z' = z - \frac{d_s^2}{\eta_{ss}}. \quad (31)$$

Действительно, из (30) следует $\bar{d}'_s = 0$. Поэтому

$$(d', D^{\delta_0}(\mathcal{J}_k')) = (\bar{d}', D^{\delta_0}(\mathcal{J}_k)) = (d, D^{\delta_0}(\mathcal{J}_k)) - \frac{d_s^2}{\eta_{ss}},$$

откуда и получаем (31).

Таким образом, процесс можно возобновить, начиная с определения величины t_0 , т.е. с п. 2⁰ В.

Отметим, что так как $d_s > 0$, то $z' > 0$ и, следовательно, случай $z = 0$, т.е. случай с) в описании схемы метода, не может реализоваться подряд дважды.

β) Рассмотрим теперь случай $\eta_{ss} = 0$. Покажем, что в этом случае точка x^{k+2} будет являться точкой минимума функции f на аффинном многообразии $L(\mathcal{J}_{k+2})$, и, следовательно, реализуется случай а) в описании метода.

ЛЕММА 3. Если $\eta_{ss} = 0$, то для вектора \bar{g} , удовлетворяющего условиям

$$\sum_{j \in \mathcal{J}_k} A^j \bar{g}_j + A^{\delta_0} \bar{g}_{j_0} = 0, \quad (32)$$

$$\bar{g}_j = 0, \quad j \in \mathcal{J}_k \cup \{j_0\}$$

и в

следует $\bar{g}_{j_0} = 0$, $j' = j_{s-m}$ (32')

ДОКАЗАТЕЛЬСТВО. Предположим противное: $\bar{g}_{j'} = 0$, $\bar{g}_{j_0} \neq 0$. Без ограничения общности можно считать $\bar{g}_{j_0} = 1$. Заметим, что условию (32) удовлетворяет также рассматривавшийся выше вектор g с компонентами (2I), причем $g_{j'} \neq 0$. Тогда для вектора $\lambda = g - \bar{g}$ будем иметь

$$\sum_{j \in J_k} A^j \lambda_j = 0$$

и $\lambda_{j'} \neq 0$. Отсюда, ввиду того, что ранг матрицы $A(J_k)$ равен m , получаем, что ранг матрицы $A(J'_k)$, $J'_k = J_k \setminus \{j'\}$, также равен m . Но тогда матрица $H(J'_k)$ неособенная. Действительно, если бы $H(J'_k)$ оказалась особенной, то по лемме I были бы линейно зависимы векторы D^j , $j \in J'_k \subset J_k$ и, следовательно, была бы особенной матрица $H(J_k)$, что неверно.

Таким образом, неособенная матрица $H(J_k)$ получается окаймлением неособенной матрицы $H(J'_k)$. Но тогда из формул метода окаймления следует, что диагональный элемент η_{ss} не равен нулю, что противоречит предположению. Лемма доказана.

Для доказательства того, что x^{k+2} доставляет минимум функции f на $L(J_{k+2})$, покажем, что производная этой функции по любому направлению \bar{g} , не выводящему из $L(J_{k+2})$, т.е. удовлетворяющему условию (32), равна нулю.

Действительно,

$$\frac{\partial f(x^{k+2})}{\partial \bar{g}} = (\bar{g}, Cx^{k+2}) + (\bar{g}, p)$$

и, ввиду (32),

$$\frac{\partial f(x^{k+2})}{\partial \bar{g}} = (\bar{g}(J_{k+2}), C(J_{k+2})x^{k+2}(J_{k+2})) + (\bar{g}(J_{k+2}), p(J_{k+2})) \quad (33)$$

Из (I7' - I8') можем записать

$$A^T(J'_k)w_y^{k+1} - C(J'_k)x^{k+2}(J'_k) - C^{j_0}(J'_k)x_{j_0}^{k+2} = p(J'_k), \quad (34)$$

$$(A^{j_0}, w_y^{k+2}) - (C^{j_0}(J'_k), x^{k+2}(J'_k)) - c_{j_0 j_0} x_{j_0}^{k+2} = p_{j_0} + \Delta'_{j_0} \quad (35)$$

Умножая (34) скалярно на $\bar{g}(J'_k)$ и прибавляя к равенству (35), умноженному на \bar{g}_{j_0} , получим ввиду (32)

$$-(\bar{g}(J_{k+2}), C(J_{k+2})x^{k+2}(J_{k+2})) = (p(J_{k+2}), \bar{g}(J_{k+2})) + \Delta'_{j_0} \bar{g}_{j_0}$$

и, учитывая (33),

$$\frac{\partial f(x^{k+2})}{\partial \bar{g}} = -\Delta'_{j_0} \bar{g}_{j_0}.$$

Но в силу леммы 3 $\bar{g}_{j_0} = 0$, и утверждение доказано.

Таким образом, в случае $\eta_{ss} = 0$ можно вернуться к п. I⁰, получив предварительно матрицу $H^{-1}(\mathcal{J}_{k+2})$ и вектор

$$\bar{w}^{k+2} = H^{-1}(\mathcal{J}_{k+2}) B(\mathcal{J}_{k+2}).$$

Заметим, что матрица $H(\mathcal{J}_{k+2})$ получается из матрицы $H(\mathcal{J}_k)$ заменой S -того столбца вектором $D^{j_0}(\mathcal{J}_k)$ и S -ой строки вектором $D^{j_0}(\mathcal{J}_{k+2}) = D^{j_0}(\mathcal{J}_k) + e^S(-c_{j_0 j_0} + c_{j j_0})$, где e^S - S -ый орт, т.е.

$$e^S = (\overbrace{0, \dots, 0}^S, 1)$$

Поэтому матрица $H^{-1}(\mathcal{J}_{k+2})$ может быть получена из матрицы $H^{-1}(\mathcal{J}_k)$ последовательным применением двух преобразований метода пополнения (см. [3] стр. 198-203) следующим образом.

Пусть \tilde{H} - матрица, получаемая из $H(\mathcal{J}_k)$ заменой S -го столбца вектором $D^{j_0}(\mathcal{J}_k)$. Как и прежде, предполагаем, что S является номером последнего столбца матрицы $H(\mathcal{J}_k)$. Тогда

$$\tilde{H}^{-1} = L_1 H^{-1}(\mathcal{J}_k), \quad L_1 = \begin{pmatrix} 1 & & & -\frac{d_1}{d_s} \\ & 1 & 0 & -\frac{d_2}{d_s} \\ & & \ddots & \\ 0 & & & 1 \\ & & & & \frac{1}{d_s} \end{pmatrix},$$

где d_i ($i=1, \dots, S$) - компоненты вектора $d = H^{-1}(\mathcal{J}_k) D^{j_0}(\mathcal{J}_k)$. Аналогично, так как $H(\mathcal{J}_{k+2})$ получается из \tilde{H} заменой S -ой строки вектором $D^{j_0}(\mathcal{J}_{k+2})$, то $H^{-1}(\mathcal{J}_{k+2}) = \tilde{H}^{-1} L_2$,

$$L_2 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & 0 \\ & -\frac{\tilde{d}_1}{d_s} & -\frac{\tilde{d}_2}{d_s} & \ddots \\ & & & \frac{1}{d_s} \end{pmatrix},$$

где \tilde{d}_i ($i=1, \dots, S$) - компоненты вектора $\tilde{d} = D^{j_0}(\mathcal{J}_{k+2}) \tilde{H}^{-1}$. Для вектора \tilde{d} можно записать

$$\tilde{d} = D^{j_0}(\mathcal{J}_{k+2}) L_1 H^{-1}(\mathcal{J}_k).$$

Но

$$\begin{aligned} D^{j_0}(\mathcal{J}_{k+2}) L_1 &= (D^{j_0}(\mathcal{J}_k) + e^S(-c_{j_0 j_0} + c_{j j_0})) L_1 = D^{j_0}(\mathcal{J}_k) - D^{j_0}(\mathcal{J}_k) e^S - \\ &- \left(\frac{1}{d_s} \sum_{j=1}^{S-1} d_j D^{j_0}(\mathcal{J}_k) \right) e^S + \frac{1}{d_s} D^{j_0}(\mathcal{J}_k) e^S + \frac{1}{d_s} (c_{j j_0} - c_{j_0 j_0}) e^S = \end{aligned}$$

$$= D^{j_0}(\tilde{\gamma}_k) - \frac{1}{\alpha_s} \left(\sum_{j=1}^S d_j D_j^{j_0}(\tilde{\gamma}_k) + c_{j_0 j_0} \right) e^s = D^{j_0}(\tilde{\gamma}_k) - \frac{\gamma}{\alpha_s} e^s.$$

Поэтому

$$\tilde{d} = (D^{j_0}(\tilde{\gamma}_k) - \frac{\gamma}{\alpha_s} e^s) H^{-1}(\tilde{\gamma}_k) = d - \frac{\gamma}{\alpha_s} \eta^s, \quad (36)$$

где η^s — S -ая строка матрицы $H^{-1}(\tilde{\gamma}_k)$.

Получим теперь формулы для вектора

$$\overline{w}^{k+2}(\tilde{\gamma}_{k+2}) = H^{-1}(\tilde{\gamma}_{k+2}) B(\tilde{\gamma}_{k+2}) = L_1 H^{-1}(\tilde{\gamma}_k) L_2 B(\tilde{\gamma}_{k+2}).$$

Так как $B(\tilde{\gamma}_{k+2}) = B(\tilde{\gamma}_k) + (\rho_{j_0} - \rho_{j_1}) e^s$, то

$$\begin{aligned} L_2 B(\tilde{\gamma}_{k+2}) &= B(\tilde{\gamma}_k) - B_s(\tilde{\gamma}_k) e^s - \frac{1}{\alpha_s} \left(\sum_{j=1}^{S-1} B_j(\tilde{\gamma}_k) \tilde{d}_j \right) e^s + \\ &+ \frac{1}{\alpha_s} B_s(\tilde{\gamma}_k) e^s + \frac{1}{\alpha_s} (\rho_{j_0} - \rho_{j_1}) e^s = \\ &= B(\tilde{\gamma}_k) - \frac{1}{\alpha_s} ((B(\tilde{\gamma}_k), \tilde{d}) - \rho_{j_0}) e^s. \end{aligned} \quad (37)$$

Далее,

$$H^{-1}(\tilde{\gamma}_k) B(\tilde{\gamma}_k) = w^{k+1}(\tilde{\gamma}_k) + x_{j_0}^{k+1} d \quad (38)$$

и, ввиду (36),

$$\begin{aligned} (B(\tilde{\gamma}_k), \tilde{d}) &= (B(\tilde{\gamma}_k), d) - \frac{\gamma}{\alpha_s} (B(\tilde{\gamma}_k), \eta^s) = (B(\tilde{\gamma}_k), H^{-1}(\tilde{\gamma}_k) D^{j_0}(\tilde{\gamma}_k)) - \\ - \frac{\gamma}{\alpha_s} (B(\tilde{\gamma}_k), \eta^s) &= (w^{k+1}(\tilde{\gamma}_k), D^{j_0}(\tilde{\gamma}_k)) + x_{j_0}^{k+1} (d, D^{j_0}(\tilde{\gamma}_k)) - \frac{\gamma}{\alpha_s} (w_s^{k+1}(\tilde{\gamma}_k) - x_{j_0}^{k+1} d_s). \end{aligned}$$

Но

$$(w^{k+1}(\tilde{\gamma}_k); D^{j_0}(\tilde{\gamma}_k)) = \rho_{j_0} + \Delta_{j_0} + c_{j_0 j_0} x_{j_0}^{k+1}.$$

Поэтому

$$\begin{aligned} (B(\tilde{\gamma}_k), \tilde{d}) &= \rho_{j_0} + \Delta_{j_0} + x_{j_0}^{k+1} (c_{j_0 j_0} + (d, D^{j_0}(\tilde{\gamma}_k))) - \\ - \frac{\gamma}{\alpha_s} (w_s^{k+1}(\tilde{\gamma}_k) + x_{j_0}^{k+1} d_s) &= \rho_{j_0} + \Delta_{j_0} - \frac{\gamma}{\alpha_s} w_s^{k+1}(\tilde{\gamma}_k). \end{aligned}$$

Обозначим $\tilde{\Delta} = \Delta_{j_0} - \frac{\gamma}{\alpha_s} w_s^{k+1}(\tilde{\gamma}_k)$. Теперь (37) можно упростить:

$$L_2 B(\tilde{\gamma}_{k+2}) = B(\tilde{\gamma}_k) - \frac{\tilde{\Delta}}{\alpha_s} e^s.$$

В результате, если учесть (38), получаем

$$\begin{aligned} \overline{w}^{k+2}(\tilde{\gamma}_{k+2}) &= L_1 H^{-1}(\tilde{\gamma}_k) (B(\tilde{\gamma}_k) - \frac{\tilde{\Delta}}{\alpha_s} e^s) = \\ &= L_1 w^{k+1}(\tilde{\gamma}_k) + x_{j_0}^{k+1} L_1 d - \frac{\tilde{\Delta}}{\alpha_s} L_1 \eta^s, \end{aligned}$$

и, так как $L, d = e^s$, то окончательно

$$\bar{w}^{k+2}(\bar{J}_{k+2}) = \bar{w} + x_{j_0}^{k+1} e^s - \frac{\bar{\Delta}}{d_s} \bar{\eta}^s,$$

где

$$\bar{w} = L, w^{k+1}(\bar{J}_k) \quad \text{и} \quad \bar{\eta}^s = L, \eta^s.$$

Заметим, что если случай $\eta_{ss} = 0$ реализуется при выполнении п. 2^о непосредственно после очередного выполнения п. 1^о, то $x_{j_0}^{k+1} = 0$. Кроме того, $\bar{\eta}^s = \eta^s$, и так как по лемме 3 $\eta_x^s = 0$, то

$$\bar{w}_x^{k+2}(\bar{J}_{k+2}) = (L, w^{k+1}(\bar{J}_k))_x + x_{j_0}^{k+1} e_x^s = w_x^{k+2}(\bar{J}_{k+2}).$$

Завершая рассмотрение алгоритма, остановимся на вопросе получения начального вектора v^0 , множества J_0 и матрицы $H^{-1}(J_0)$.

Для получения начального допустимого вектора x^0 и множества J_0 можно воспользоваться обычно применяемым для этой цели в задачах линейного программирования приемом искусственного базиса (см. [2], стр. 103-110). Если при этом применить модифицированный симплекс-метод, то в результате, при условии совместности ограничений задачи, будет получен допустимый вектор x^0 , множество $J_0 = J(x^0)$ и матрица G , обратная к матрице $A(J_0)$. для получения вектора w^0 определим вектор

$$y^0 = (p(J_0) + C(J_0) x^0(J_0)) G$$

и образуем вектор $w^0(J^0)$, составив векторы y_0 и $x^0(J_0)$. Матрица $H^{-1}(J_0)$, как легко убедиться, будет иметь такую структуру

$$H^{-1}(J_0) = \begin{pmatrix} G^T C(J_0) G & G^T \\ G & 0 \end{pmatrix}$$

§ 3. Вычислительная схема алгоритма

Вычислительная схема описанного алгоритма во многом схожа с вычислительной схемой модифицированного симплекс-метода для задач линейного программирования. Величины, необходимые для выполнения одного шага процесса, схематически удобно объединить в следующую таблицу

K	$w(\mathcal{J})$	$H^{-1}(\mathcal{J})$				u	d	$D^{\circ}(\mathcal{J})$
j_0	$w_{j_0}^K$	$\eta_{1,1}$			$\eta_{1,m+l}$		d_1	a_{j_0}
\vdots	\vdots	\vdots			\vdots		\vdots	\vdots
j_m	$w_{j_m}^K$	$\eta_{m,1}$			$\eta_{m,m+l}$		d_m	$a_{m j_0}$
x_{j_0}	$w_{x_{j_0}}^K$	$\eta_{m+1,1}$			$\eta_{m+1,m+l}$		d_{m+1}	$-c_{j_0}$
\vdots	\vdots	\vdots			\vdots		\vdots	\vdots
x_{j_l}	$w_{x_{j_l}}^K$	$\eta_{m+l,1}$			$\eta_{m+l,m+l}$		d_{m+l}	$-c_{j_0}$
Δ	Δ_{j_0}	d_1			d_{m+l}		γ	$c_{j_0 j_0}$
x_{j_0}	$w_{x_{j_0}}^K$							

Рис. 1.

В этой таблице первая строка и первый столбец указывают лишь наименование соответствующих величин и введены для наглядности. Столбец u введен для удобства выполнения преобразований по формулам метода окаймления.

При выполнении алгоритма информация, содержащаяся в таблице, будет преобразовываться следующим образом.

1⁰. К моменту выполнения п. 1⁰ в таблице находятся лишь вектор $w^K(\mathcal{J}_K)$ и матрица $H^{-1}(\mathcal{J}_K)$. После того как при проверке неравенств (5') определено j_0 , для которого соответствующее неравенство из (5') нарушается, в таблицу на указанные места заносятся величины Δ_{j_0} , $c_{j_0 j_0}$ и вектор $D^{\circ}(\mathcal{J}_K)$. Кроме того, в дополнительную строку записываются: в столбец наименований — x_{j_0} , а в столбец $w(\mathcal{J})$ — ноль.

2⁰. Умножая матрицу $H^{-1}(\mathcal{J})$ на вектор $D^{\circ}(\mathcal{J})$, заполняем столбец d . После этого перемножаем скалярно столбцы d и $D^{\circ}(\mathcal{J})$ и добавляем к полученному произведению величину $c_{j_0 j_0}$, получая величину γ .

Для определения величины t_0 находим минимальное среди отношений элементов столбца $w(\mathcal{J})$ к положительным соответствующим элементам столбца d , начиная с $(m+1)$ -ой строки до Δ -строки включительно. Если оказалось, что в этой части столбца d нет положительных элементов, то $t_0 = +\infty$ и процесс окончен: минимизируемая функция f не ограничена снизу на множестве допустимых векторов задачи.

В случае, когда минимальное из указанных отношений находится в Δ -строке, т.е. когда реализуется случай ii) в описании алгоритма, необходимые преобразования таблицы будут состоять в

следующем. В дополнительный столбец u рядом с матрицей $H^{-1}(J)$ вписываются нули, а в Δ -строку вписывается -1 . После этого часть таблицы, содержащая $w(J)$, $H^{-1}(J)$ и дополнительный столбец u по Δ -строку включительно, преобразуется, как в модифицированном симплекс-методе: Δ -строка делится на γ , в результате чего получается x_{j_0} -строка новой таблицы, и вычитается из остальных строк, умноженная на соответствующие элементы столбца d . Преобразование таблицы завершается добавлением элемента столбца $w(J)$, стоящего в дополнительной строке, к соответствующему элементу новой x_{j_0} -строки. Описанное преобразование кратко будем называть преобразованием окаймления. Получив новую таблицу, возвращаемся к выполнению п. 1°.

В случае, когда минимальное из отношений при определении величины t_0 находится не в Δ -строке, а в некоторой другой строке с номером S , т.е. когда реализуется случай *iii*) в описании алгоритма, преобразования таблицы проводятся различно в зависимости от того, равен ли нулю элемент η_{ss} или нет.

Если $\eta_{ss} \neq 0$, то умноженные на t_0 элементы столбца d до Δ -строки включительно вычитаются из соответствующих элементов столбца $w(J)$. После этого строки матрицы $H^{-1}(J)$, дополненные элементами столбца d , и соответствующая часть Δ -строки преобразуются, как в модифицированном симплекс-методе, чтобы в результате в S -ом столбце получился вектор e^S , а именно: S -ая строка делится на η_{ss} и вычитается из остальных строк, умноженная на соответствующие элементы S -го столбца. После этого S -ый столбец и S -ая строка из таблицы исключаются. В заключение к элементу столбца $w(J)$, находящемуся в дополнительной x_{j_0} -строке, добавляется величина t_0 . Описанное преобразование кратко будем называть преобразованием сужения. Получив новую таблицу, повторяем п. 2°, начиная с определения величины t_0 . Столбец $D^{j_0}(J)$ в новой таблице опускается.

Если $\eta_{ss} = 0$, то необходимые преобразования проводятся в два этапа. Вначале часть таблицы, содержащая $w(J)$, $H^{-1}(J)$ до Δ -строки включительно, преобразуется, как в модифицированном симплекс-методе: S -ая строка делится на d_S и вычитается из остальных строк, умноженная на соответствующие элементы столбца d . Затем выполняем аналогичные преобразования

над столбцами полученной таблицы: S -ый столбец делится на элемент \tilde{a}_s , находящийся в S -ом столбце и Δ -строке новой таблицы, и вычитается из остальных столбцов (включая и w -столбец), умноженный на соответствующие элементы Δ -строки.

В заключение элемент столбца $w(j)$, стоящий в дополнительной x_{j_0} -строке, добавляется к соответствующему элементу S -ой строки, после чего S -ая строка становится x_{j_0} -строкой, что и отмечается в столбце наименований. Описанное преобразование кратко будем называть преобразованием замещения. После выполнения этого преобразования возвращаемся к п. 1⁰.

Резюмируя изложение, приводим для наглядности блок-схему алгоритма.

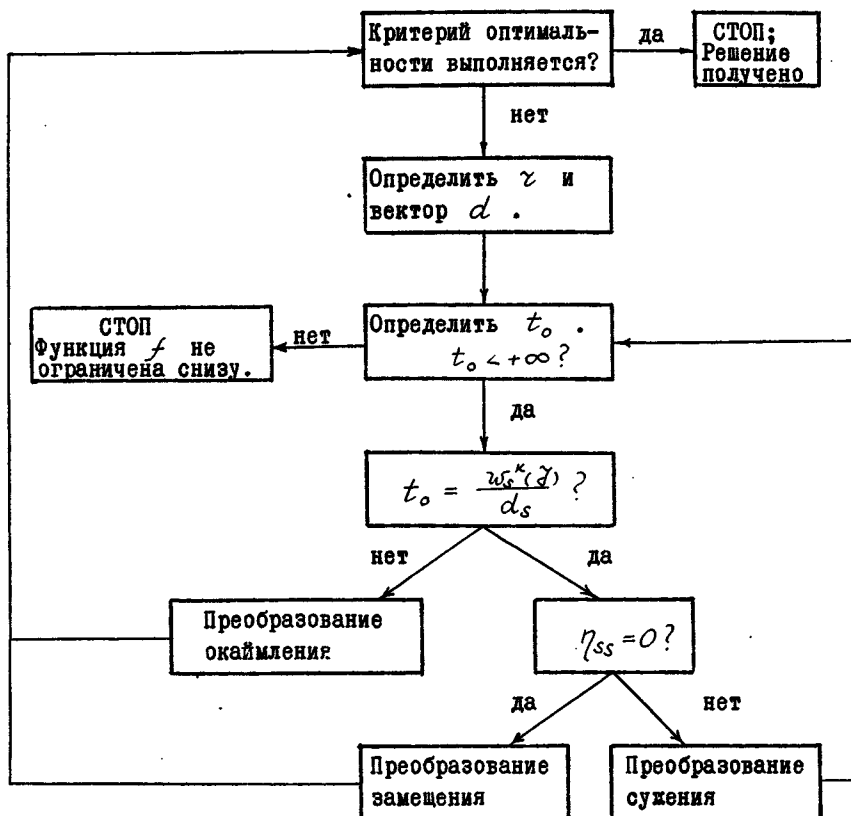


Рис. 2. Блок-схема алгоритма

В заключение этого параграфа рассмотрим численный пример, иллюстрирующий описанную выше схему.

ПРИМЕР. Минимизировать

$$f(x) = -\frac{1}{3}x_1 - 2x_2 + \frac{1}{2}(x_1^2 + x_2^2)$$

при условиях

$$2x_1 + 3x_2 + x_3 = 4\frac{2}{3},$$

$$x_1 + x_2 + x_4 = 1\frac{1}{3}.$$

В качестве начального решения возьмем

$$x^0 = (0, 0, 4\frac{2}{3}, 1\frac{1}{3})$$

Тогда $J_0 = \{3, 4\}$ и

$$A^{-1}(J_0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Поэтому получаем $y^0 = (0, 0)$, $w^0(J_0) = (0, 0, 4\frac{2}{3}, 1\frac{1}{3})$,

$$H^{-1}(J_0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Дальнейший ход вычислений явствует из приводимых таблиц, сопровождаемых необходимыми пояснениями.

I-ый шаг.

θ	$w(J)$	$H^{-1}(J)$				u	d	$D'(J)$
y_1	0	0	0	1	0	0	0	2
y_2	0	0	0	0	1	0	0	1
x_3	$4\frac{2}{3}$	1	0	0	0	0	2	0
x_4	$1\frac{1}{3}$	0	1	0	0	0	1	0
Δ	$\frac{1}{3}$	0	0	2	1	-1	①	1
x_1	0							

При проверке признака оптимальности нарушается неравенство (5') при $J_0 = 1$. Минимальное отношение при определении величины t_0 находится в Δ - строке: $t_0 = \frac{1}{3}$ (соответствующий элемент столбца d отмечен). Поэтому выполняется преобразование окаймления.

2-ой шаг.

1	w_j	$H^{-1}(J)$					u	d	\bar{w}_j
y_1	0	0	0	1	0	0	0	3	
y_2	0	0	0	0	1	0	0	1	
x_3	4	1	0	-4	-2	2	3	0	
x_4	1	0	1	-2	-1	1	①	0	
x_5	$\frac{1}{3}$	0	0	2	1	-1	0	0	
Δ	2	0	0	3	1	0	1	1	
x_2	0								

На этом шаге $j_0 = 2$, но при определении t_0 минимальное отношение находится в 4-ой строке: $t_0 = 1$. При этом $\eta_{44} = -1 \neq 0$. Выполняется преобразование сужения.

3-ий шаг.

2	w_j	$H^{-1}(J)$					u	d
y_1	0	0	0	1	0			0
y_2	0	0	1	-2	1			1
x_3	1	1	-2	0	0			1
x_4	$\frac{1}{3}$	0	1	0	0			①
Δ	1	0	1	1	1			2
x_2	1							

При определении t_0 минимальное отношение находится в 4-ой строке: $t_0 = \frac{1}{3}$. Но $\eta_{44} = 0$, поэтому выполняется преобразование замещения (в два этапа).

\bar{w}_j	$\bar{H}^{-1}(J)$				
0	0	0	1	0	
$-\frac{1}{3}$	0	0	-2	1	
$\frac{2}{3}$	1	-3	0	0	
$\frac{1}{3}$	0	1	0	0	
$\frac{1}{3}$	0	-1	1	1	

4-ый шаг.

3	w_j	$H^{-1}(J)$				
y_1	0	0	0	1	0	
y_2	$-\frac{2}{3}$	0	1	-3	1	
x_3	$\frac{2}{3}$	1	-3	0	0	
x_2	$\frac{4}{3}$	0	1	0	0	

Все неравенства (5') выполняются. Следовательно, вектор

$$x^3 = (0, \frac{4}{3}, \frac{2}{3}, 0)$$

является решением рассматриваемой задачи.

§ 4. Другие алгоритмы

В этом параграфе, не касаясь подробностей, кратко остановимся на некоторых других подходах к построению алгоритмов метода последовательного улучшения для задач квадратичного программирования.

При разработке алгоритмов, предназначенных для реализации на ЭВМ, одним из основных факторов является объем оперативной памяти машины, необходимой для реализации алгоритма. В связи с этим отметим, что при реализации рассмотренного алгоритма, ввиду симметрии фигурирующих в процессе матриц $H(\tilde{J}_k)$, в оперативной памяти ЭВМ можно хранить лишь половину матрицы $H^{-1}(\tilde{J}_k)$, учитывая вторую половину алгоритмическим путем.

Ещё более рациональное использование памяти ЭВМ получается, если для решения систем линейных уравнений, фигурирующих в процессе, воспользоваться клеточным аналогом метода исключения Гаусса, храня в оперативной памяти и преобразуя от шага к шагу обратные матрицы диагональных клеток. Примером алгоритма такого типа может служить алгоритм для решения частной задачи квадратичного программирования, когда матрица C единичная (см. [5]). В этом случае система (7) имеет вид

$$A(\tilde{J}_k)x(\tilde{J}_k) = b$$

$$A^T(\tilde{J}_k)y - x(\tilde{J}_k) = p(\tilde{J}_k)$$

и эквивалентна следующей системе с квазистреугольной матрицей:

$$A(\tilde{J}_k) A^T(\tilde{J}_k)y = b + A(\tilde{J}_k)p(\tilde{J}_k);$$

$$A^T(\tilde{J}_k)y - x(\tilde{J}_k) = p(\tilde{J}_k)$$

Для решения этой системы достаточно иметь матрицу, обратную к матрице $A(\tilde{J}_k) A^T(\tilde{J}_k)$, преобразование которой от шага к шагу можно осуществлять по методу пополнения.

Однако непосредственное обобщение такого алгоритма на случай произвольной положительно определенной матрицы C дает малую экономию оперативной памяти ЭВМ (даже если хранить половину

матрицы $C^{-1}(\mathcal{J}_k)$ по сравнению с реализацией рассмотренного алгоритма, когда хранится половина матрицы $H^{-1}(\mathcal{J}_k)$. Поэтому в общем случае указанный подход требует реализации по более сложной схеме, например, следующей.

Так как ранг матрицы $A(\mathcal{J}_k)$ равен m , то множество \mathcal{J}_k можно разбить на два подмножества \mathcal{J}_k' и \mathcal{J}_k'' так, чтобы матрица $A(\mathcal{J}_k')$ была квадратной и неособенной. Поэтому при надлежащей упорядоченности множества \mathcal{J}_k матрица $H(\mathcal{J}_k)$ будет иметь структуру

$$H(\mathcal{J}_k) = \begin{pmatrix} O & A(\mathcal{J}_k'') & A(\mathcal{J}_k') \\ A^T(\mathcal{J}_k'') & C_{11} & C_{12} \\ A^T(\mathcal{J}_k') & C_{12}^T & C_{22} \end{pmatrix}$$

где $C_{11} = -C(\mathcal{J}_k'')$, $C_{22} = -C(\mathcal{J}_k')$ и C_{12} - соответствующие клетки матрицы $-C(\mathcal{J}_k)$. Легко видеть, что при

$$L = \begin{pmatrix} E_1 & O & O \\ O & E_2 & O \\ O & -A^T(\mathcal{J}_k')A(\mathcal{J}_k'') & E_3 \end{pmatrix},$$

где E_1 , E_2 , E_3 - единичные матрицы требуемых размеров, матрица $\bar{H} = L^T H(\mathcal{J}_k) L$ будет квазитреугольной:

$$\bar{H} = \begin{pmatrix} O & O & A(\mathcal{J}_k') \\ O & \bar{C}_{11} & \bar{C}_{12} \\ A^T(\mathcal{J}_k') & \bar{C}_{12}^T & C_{22} \end{pmatrix}$$

Теперь если заменить систему (7')

$$H(\mathcal{J}_k) w(\mathcal{J}_k) = B(\mathcal{J}_k)$$

ей эквивалентной системой

$$L^T H(\mathcal{J}_k) L L^{-1} w(\mathcal{J}_k) = L^T B(\mathcal{J}_k),$$

то для получения $w(\mathcal{J}_k)$ достаточно решить систему с матрицей \bar{H}

$$\bar{H} \bar{w}(\mathcal{J}_k) = L^T B(\mathcal{J}_k),$$

после чего для $w(\mathcal{J}_k)$ имеем

$$w(\mathcal{J}_k) = L \overline{w}(\mathcal{J}_k).$$

При реализации такого алгоритма в оперативной памяти ЭВМ необходимо будет хранить матрицы: $A'(\mathcal{J}_k')$, $A'(\mathcal{J}_k')A(\mathcal{J}_k)$, \overline{C}_{11}' , C_{12} , C_{22} . Экономия памяти достигается за счет того, что в практических задачах матрица C , как правило, мало заполненная. Однако и сложность алгоритма по сравнению с рассмотренным существенно возрастает.

Следует отметить, что получаемый таким образом алгоритм фактически реализует схему метода [4], что указывает на эквивалентность методов [1] и [4] в случае задач квадратичного программирования.

В заключение заметим, что для минимизации квадратичной функции на аффинных многообразиях можно воспользоваться также методами сопряженных направлений, позволяющими получить точку минимума за конечное число шагов. При этом получим другую группу алгоритмов метода последовательного улучшения для задач квадратичного программирования. Примером алгоритма такого типа является известный алгоритм [6].

Л и т е р а т у р а

1. Рубинштейн Г.Ш., Шмырёв В.И. Методы минимизации квази-
впуклой функции на выпуклом многограннике. "Оптимизация" I (18),
1971, 82-117.
2. Данциг Дж. Линейное программирование, его применения и
обобщения, "Прогресс", М., 1966.
3. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной
алгебры, Физматгиз, М., 1960.
4. Булавский В.А., Рубинштейн Г.Ш. О решении задач выпуклого
программирования с линейными ограничениями методом последо-
вательного улучшения допустимого вектора. ДАН СССР, 150 : 2
(1963), 231-234.
5. Булавский В.А. Один специальный алгоритм квадратичного
программирования. Настоящий сборник, 23-36.

Поступила в редакцию
15.П. 1972 г.