

ЗАМЕТКИ И ПИСЬМА

О ВЫЧИСЛИТЕЛЬНОЙ СХЕМЕ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Ю. А. ЗАК

(Киев)

В задачах линейного программирования, связанных с проблемами оптимального планирования, управления производственными процессами и др., очень часто эвристическими методами удается найти план, удовлетворяющий исходной системе ограничений и достаточно близкий к оптимальному. В симплексном методе, который в качестве начального приближения использует лишь угловые точки многогранника, эта полезная информация не может быть использована. Предлагаемый градиентный метод лишен указанного недостатка.

Пусть требуется найти

$$L = \min_{X \in G} (C, X),$$

$$G = \{X = (x_j), \quad i \in J, \quad J = \{1, \dots, n\} \mid b_{1i} \leq \sum_j a_{ij} x_j \leq b_{2i}, \quad (1)$$

$$i \in I, \quad I = \{1, \dots, m\}; \quad d_{1\xi} \leq x_\xi \leq d_{2\xi}, \quad \xi \in J_1; \quad x_p \geq 0,$$

$$p \in J_2; \quad J_1, J_2 \subseteq J\}.$$

$X = (x_j), \quad j \in J$, — вектор-столбец размерностью n .

Рассматриваемый алгоритм решения (1) использует методы проектирования градиента. Он позволяет осуществить движение к экстремальной точке многогранника вдоль границы множества G под острым углом к антиградиенту и обеспечивает монотонное убывание целевой функции. В отличие от метода Розена [4] алгоритм благодаря использованию другой процедуры определения допустимого направления убывания целевой функции в точке $X^k, k = 0, 1, \dots$, не сводится к вычислительной схеме по методу обратной матрицы [2, 3] и по своей реализации существенно отличается от симплексных методов линейного программирования.

На каждом шаге алгоритма $k = 0, 1, \dots$ определяются: величина шага вдоль направления убывания целевой функции, коэффициенты нового приближения к решению задачи $X^k \in G, (C, X^k) < (C, X^{k-1})$ и направление луча, исходящего из X^k , принадлежащего границе множества G и составляющего острый угол с вектором $-C$.

В качестве начального приближения выбирается вектор $X^0 \in G$, что возможно, если $G \neq \emptyset$. Допустимое направление убывания целевой функции на k -м шаге задается уравнением луча

$$\begin{aligned} x_j &= x_j^k + \alpha_j^k t, \quad k = 0, 1, \dots; \quad j = 1, \dots, n, \\ \alpha_j^k &= \begin{cases} -c_j, & \text{если } k = 0, \\ x_j^k - x_j^k, & \text{или } z_j^k - x_j^k, & \text{если } k = 1, 2, \dots, \end{cases} \end{aligned} \quad (2)$$

где координаты x_j^k, z_j^k и x_j^k находятся на предыдущем шаге.

Опишем k -ю, $k = 0, 1, 2, \dots$, итерацию алгоритма.

1. Определим величину шага вдоль направления (2). Пересечение луча (2) с различными гиперплоскостями ограничений задачи

$$t_{1i} = \frac{\sum_{j=1}^n a_{ij} x_j^k - b_{1i}}{-\sum_{j=1}^n a_{ij} \alpha_j^k}, \quad t_{2i} = \frac{b_{2i} - \sum_{j=1}^n a_{ij} x_j^k}{\sum_{j=1}^n a_{ij} \alpha_j^k}, \quad i \in I,$$

* В результате этого не происходит с каждым шагом понижения размерности многообразия I^k , на которое проектируется антиградиент, и попадание через фиксированное число шагов в вершину многогранника G .

$$t_{1\xi} = \frac{x_{\xi}^k - d_{1\xi}}{-\alpha_{\xi}^k}, \quad t_{2\xi} = \frac{d_{2\xi} - x_{\xi}^k}{\alpha_{\xi}^k}, \quad \xi \in J_1. \quad (3)$$

$$t_p = \frac{x_p^k}{-\alpha_p^k}, \quad p \in I_2.$$

Обозначим I_+^k , $k = 0, 1, \dots$, множество индексов, для которых справедливы выражения

$$\left[\left(\sum_{j=1}^n a_{ij} x_j^k - b_{1i} \geq 0 \right) \wedge \left(- \sum_{j=1}^n a_{ij} \alpha_j^k > 0 \right) \right] \vee \left[\left(b_{2i} - \sum_{j=1}^n a_{ij} x_j^k \geq 0 \right) \wedge \right. \\ \left. \wedge \left(\sum_{j=1}^n a_{ij} \alpha_j^k > 0 \right) \right], \\ [(x_{\xi}^k - d_{1\xi} \geq 0) \wedge (-\alpha_{\xi}^k > 0)] \vee [(d_{2\xi} - x_{\xi}^k \geq 0) \wedge (\alpha_{\xi}^k > 0)]. \quad (4)$$

Определим допустимую величину шага в направлении (2)

$$t_k^* = \min_{\substack{t_{1i}, t_{2i}, t_{1\xi}, t_{2\xi}, t_p \\ i, \xi, p \in I_+^k}} (t_{1i}, t_{2i}, t_{1\xi}, t_{2\xi}, t_p). \quad (5)$$

2. Определим координаты нового приближения к решению задачи. Сделаем шаг вдоль направления луча (2), величина которого равна t_k^* . Если $t_k^* = \infty$, то линейная форма является неограниченной снизу

$$x_j^{k+1} = x_j^k + \alpha_j^k t_k^*, \quad k = 0, 1, \dots; j = 1, \dots, n. \quad (6)$$

3. В точке X^{k+1} определим направление убывания целевой функции вдоль грани множества G . Для этого спроектируем на границу многогранника G точку, лежащую за пределами множества G на луче, вдоль которого происходит убывание целевой функции. Эта точка может быть выбрана несколькими способами.

а) Определим точку Y^{k+1} на луче, параллельном вектор-антиградиенту целевой функции

$$y_j^{k+1} = x_j^k - c_j, \quad j = 1, \dots, n.$$

б) Возьмем любую точку на луче (2) вне многогранника, обозначим ее $\bar{X}^{k+1}$. Координаты $\bar{X}^{k+1}$ могут быть найдены, например, следующим образом

$$\bar{x}_j^{k+1} = x_j^k + (t_k^* + 1) \alpha_j^k, \quad j = 1, \dots, n. \quad (7)$$

В качестве точки $\bar{X}^{k+1}$ может быть принята также точка пересечения луча (2) с ближайшей гиперплоскостью ограничений вне многогранника.

Спроектируем эту точку, $\bar{X}^{k+1}$ или Y^{k+1} на границу многогранника. Обозначим I^k множество индексов, на котором достигается значение t_k^* (если I^k состоит из группы индексов, то эта грань есть линия пересечения конечного числа гиперплоскостей). Определим направление убывания целевой функции на грани соответствующей размерности w

$$w = n - \sum_{s=1}^{m+n} \gamma_s, \quad \gamma_s = \begin{cases} 1, & \text{если } s \in \tilde{I}^k, \\ 0, & \text{если } s \notin \tilde{I}^k, \end{cases} \quad (8)$$

где

$$I^k = \begin{cases} \tilde{I}^k, & \text{если } t_k^* > 0, \\ \tilde{I}^k \cup I^{k-1}, & \text{если } t_k^* = 0. \end{cases}$$

В отличие от метода Розена [1] на k -м шаге алгоритма переход к грани меньшей размерности происходит лишь в том случае, если не существует других направлений более быстрого убывания целевой функции, имеющих общие точки с множеством G , исключая точку $X^k(t_k^* = 0)$.

Уравнение этой грани определяется совместным решением недоопределенной системы линейных алгебраических уравнений вида

$$\begin{aligned} \sum_j a_{ij} x_j &= b_i, \quad i \in I^k, \\ x_{\xi} &= d_{\xi}, \quad \xi \in I^k, \\ x_p &= 0, \quad p \in I^k, \end{aligned} \quad (9)$$

где

$$b_i = \begin{cases} b_{1i}, & \text{если } t_{1i} = t_k^*, \\ b_{2i}, & \text{если } t_{2i} = t_k^*, \end{cases} \quad d_{\xi} = \begin{cases} d_{1\xi}, & \text{если } t_{1\xi} = t_k^*, \\ d_{2\xi}, & \text{если } t_{2\xi} = t_k^*. \end{cases} \quad (10)$$

Если X^{k+1} — вершина многогранника ((9) — система n уравнений с n неизвестными), то необходимо проверить, является ли эта точка решением задачи. В этом случае вектор-антиградиент целевой функции может быть представлен в виде линейной комбинации с неотрицательными коэффициентами нормалей ограничений, в которых лежит вектор X^{k+1} .

$$-c_j = \sum_{i=1}^q u_i a_i^j. \quad (11)$$

Если какие-либо из коэффициентов u_i отрицательны (множество индексов соответствующих гиперплоскостей обозначим Q^k), то X^{k+1} не является решением задачи, а уравнение грани, на которую необходимо спроектировать антиградиент целевой функции, на данном шаге определится в результате решения недоопределенной системы линейных алгебраических уравнений (9), в которой из I^k устранены все гиперплоскости, дающие отрицательные коэффициенты в разложении (11).

Применяя схему Гаусса к расширенной матрице системы (9), (10), получим уравнение, связывающее w каких-либо свободных координат системы (множество индексов этих координат обозначим J_1^k)

$$\sum_{j \in J_1^k} \bar{a}_j x_j = \bar{b}. \quad (12)$$

Остальные $v = n - w$ зависимых координат могут быть выражены как линейная комбинация первых и вычисленных уже при обратном ходе зависимых координат

$$x_r = \sum_{j \in J_1^k} l_{rj} x_j + \sum_{r_1=1}^r p_{rr_1} x_{r_1} + \bar{a}_r, \quad r \in J/J_1^k. \quad (13)$$

Свободные координаты точки пересечения нормалей из точек Y^{k+1} и X^{k+1} с гранью (12), (13) определяются выражением

$$\bar{x}_j^{k+1} = \bar{x}_j^{k+1} + \frac{1}{\sum_{j \in J_1^k} \bar{a}_j^2} \left(\bar{b} - \sum_{j \in J_1^k} \bar{a}_j x_j^{k+1} \right) \bar{a}_j, \quad j \in J_1^k, \quad (14)$$

$$z_j^{k+1} = y_j^{k+1} + \frac{1}{\sum_{j \in J_1^k} \bar{a}_j^2} \left(\bar{b} - \sum_{j \in J_1^k} \bar{a}_j y_j^{k+1} \right) \bar{a}_j, \quad j \in J_1^k. \quad (15)$$

Выражения зависимых координат точек пересечения получим, подставив значения $\bar{x}_j^{k+1}$, $j \in J_1^k$ и z_j^{k+1} , $j \in J_1^k$ в (13).

Если $Z^{k+1} = X^{k+1}$, то целевая функция достигает экстремального значения в любой точке грани, которой принадлежит вектор X^{k+1} , и алгоритм заканчивает свою работу. В противном случае направление убывания целевой функции вдоль границы

множества G в точке X^{k+1} определяется следующим образом

$$x_j = x_j^{k+1} + t(z_j^{k+1} - x_j^{k+1}), \quad (16)$$

$$x_j = \begin{cases} x_j^{k+1} + t(x_j^{k+1} - x_j^{k+1}), & \text{если } (C, X^{k+1}) > (C, \bar{X}^{k+1}), \\ x_j^{k+1} - t(x_j^{k+1} - x_j^{k+1}), & \text{если } (C, X^{k+1}) < (C, \bar{X}^{k+1}). \end{cases} \quad (17)$$

Если $(C, X^{k+1}) = (C, \bar{X}^{k+1})$, то для определения направления убывания целевой функции на этом шаге или проверки, получено ли решение, необходимо вычислить координаты вектора Z^{k+1} по формулам (13), (15). Если точка X^{k+1} не является решением задачи, то переходим к $(k+1)$ -му шагу.

Изложенный алгоритм имеет стандартную вычислительную схему относительно различных видов ограничений задачи и не требует предварительной подготовки системы. Сходимость алгоритма следует из сходимости методов возможных направлений [2], в которых движение происходит под острым углом к градиенту с выбором оптимальной величины шага вдоль направления убывания целевой функции. К преимуществам алгоритма следует отнести то, что в ходе вычислений работа производится с исходной матрицей задачи, вследствие чего не происходит накопления ошибки с каждым шагом.

По данному алгоритму составлена программа для ЭВМ Минск-22 и решен целый ряд числовых примеров. Решение всех примеров было получено за конечное число шагов.

Автор считает своим долгом выразить благодарность И. В. Стрельцовой за помощь в реализации предложенного метода на ЭВМ и проведении вычислительных экспериментов.

ЛИТЕРАТУРА

1. J. B. Boshen. The Gradient Projection Method for Nonlinear Programming. Part I, Linear Constraints. J. Soc. Industr. Appl. Math., 1960, v. 8, N. 1.
2. С. И. Зуховицкий, Л. И. Авдеева. Линейное и выпуклое программирование. М., «Наука», 1967.
3. Д. Б. Юдин, Е. Г. Гольштейн. Линейное программирование. Теория и конечные методы. М., Физматгиз, 1963.

Поступила в редакцию
7.V.1968

ОБ ОДНОЙ ЗАДАЧЕ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ

И. П. ЛЫТКИН

(Харьков)

1. Изучается следующая задача [1]. Известно заранее производственное задание заводу на месяц (в виде месячного номенклатурного плана). Требуется составить межцеховой календарный план выполнения работ по каждому изделию, вошедшему в номенклатурный план таким образом, чтобы как можно быстрее выполнить весь комплекс работ. Необходимо учесть технологическую последовательность выполнения работ и ограничения на располагаемые ресурсы.

Составим математическую модель задачи. Введем обозначения: i — тип изделия, $i = 1, \dots, m$; j — вид работ, $j = 1, \dots, n$; C_i — плановое количество изделий i -го типа; b_j — число рабочих мест j -го вида работ, имеющееся на заводе; $x_{ij}(t)$ — кривая распределения (во времени) рабочих мест j -го вида, занятых изготовлением i -го типа изделий; a_{ij} — производительность; $a_{ij} = 1/t_{ij}$. Тогда составление календарного плана, оптимального по быстродействию, можно свести к задаче минимизации функционала

$$I = \int_0^T dt \quad (1)$$