Г. Ш. РУБИНШТЕЙН

# О РЕШЕНИИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ БОЛЬШОГО ОБЪЕМА

В работе намечаются некоторые пути повышения размерности решаемых задач линейного программирования. В общем случае это достигается за счет модификации обычно используемых методов, позволяющей применять их непосредственно к рассматриваемой задаче, без предварительного сведения ее к одной из канонических форм. Для частных типов задач линейного программирования предлагаются приемы построения специальных алгоритмов, основанные на учете особенностей строения матриц систем линейных уравнений, которые могут встретиться в процессе решения задач рассматриваемого типа.

## § I. Некоторые сведения из общей теории

Рассмотрим два множества индексов

$$M=\{1,2,\dots,m\}, \quad N=\{1,2,\dots,n\},$$

каждое из которых разбито на три попарно непересекающихся подмножества, содержащих соответственно $m_1, m_2, m_3, n_1, n_2, n_3$ элементов:

$$M=M_1\cup M_2\cup M_3, \quad N=N_1\cup N_2\cup N_3.$$

Будем предполагать заданными вещественные числа:

$$a_{ij}\,(i\in M,\, j\in N),\ b_j\,(j\in N),\ c_i\,(i\in M),\ g_i>0\,(i\in M_3),\ h_j>0\,(j\in N_3).$$

Далее, для любого вещественного числа $a$ под $[a]^+$ будем понимать $\max\{0, a\}$.

Задача I. Определить вектор

$$x=(x_1, x_2, \ldots, x_m), \tag{1}$$

максимизирующий функцию

$$\mu(x)=\sum_{i\in M} c_i x_i - \sum_{j\in N_3} h_j\Big[\sum_{i\in M} a_{ij} x_i - b_j\Big]^+ \tag{2}$$

при ограничениях:

$$x_i\in\begin{cases}(-\infty, +\infty) \text{ для } i\in M_1,\\ [0, +\infty) \text{ для } i\in M_2,\\ [0, g_i] \text{ для } i\in M_3,\end{cases} \tag{3}$$

$$\sum_{i\in M} a_{ij}\begin{cases}= b_j \text{ для } j\in N_1,\\ \le b_j \text{ для } j\in N_2,\end{cases} \tag{4}$$

Задача II. Определить вектор

$$y=(y_1, y_2, \ldots, y_n), \tag{5}$$

минимизирующий функцию

$$\nu(y)=\sum_{j\in N} b_j y_j + \sum_{i\in M_3} g_i\Big[-\sum_{j\in N} a_{ij} y_j + c_i\Big]^+ \tag{6}$$

при ограничениях:

$$y_i\in\begin{cases}(-\infty, +\infty) \text{ для } j\in N_1,\\ [0, +\infty) \text{ для } j\in N_2,\\ [0, h_j] \text{ для } j\in N_3,\end{cases} \tag{7}$$

$$\sum_{j\in N} a_{ij} y_j \begin{cases}= c_i \text{ для } i\in M_1,\\ \ge c_i \text{ для } i\in M_2.\end{cases} \tag{8}$$

З а м е ч а н и е. Различие в постановке задач I и II лишь внешнее. Умножением заданных величин $a_{ij}$, $b_j$, $c_i$ на $-1$ первая из этих задач приводится к виду задачи II, а вторая – к виду задачи I.

Приведенные задачи представляют одну из форм пары двойственных задач линейного программирования. В частном случае, когда $M_1 = M_3 = N_1 = N_3 = \phi$ или $M_1 = M_3 = N_2 = N_3 = \phi$, получаем двойственные задачи в так называемых канонических формах. С другой стороны, задачи I и II, по существу, не являются более общими, так как формально они легко сводятся к каждому из отмеченных частных случаев. Однако при таком сведении размерность задач существенно повышается. Поэтому представляет практический интерес модификация алгорифмов, позволяющая применять их непосредственно к задачам I и II (ср. [I] , гл. 6).

Векторы (I) и (5) , удовлетворяющие условиям (3), (4) и (7), (8), называются допустимыми для задач I и II, а искомые векторы - оптимальными. Совокупности допустимых векторов в изучаемых задачах обозначим через $X$ и $Y$ и рассмотрим величины

$$\mu=\begin{cases}\sup\limits_{x\in X}\mu(x) & \text{при } X\neq\phi,\\ -\infty & \text{при } X=\phi,\end{cases}\qquad \nu=\begin{cases}\inf\limits_{y\in Y}\nu(y) & \text{при } Y\neq\phi,\\ +\infty & \text{при } Y=\phi.\end{cases}\tag{9}$$

Из общей теории линейного программирования вытекает, что для величин (9) имеет место неравенство

$$\mu\leqslant\nu,$$

причем, за исключением случая, когда $\mu=-\infty$ и $\nu=+\infty$ (ни в одной из задач нет допустимого вектора), в этом неравенстве достигается равенство. Далее, при конечном значении $\mu$ и $\nu$ знаки супрерума и инфимума в (9) могут быть заменены соответственно на максимум и минимум (в обеих задачах существуют оптимальные векторы). Таким образом, справедливы:

Теорема 1 (существования). Следующие утверждения равносильны:

а) задачи I и II имеют оптимальные векторы;

б) одна из рассматриваемых задач имеет оптимальный вектор;

в) задачи I и II имеют допустимые векторы;

г) задача I имеет допустимые векторы и функция (2) на множестве этих векторов ограничена сверху;

д) задача II имеет допустимые векторы и функция (6) на множестве этих векторов ограничена снизу.

Теорема 2 (признак оптимальности). Допустимый вектор (1) задачи I тогда и только тогда является оптимальным, когда существует допустимый вектор (5) задачи II такой, что $\mu(x)=\nu(y)$. При этом вектор (5) является оптимальным в задаче II.

Учитывая ограничения на допустимые векторы (условия (3), (4), (7), (8)) и вид функций (2), (6), признак оптимальности можно сформулировать в развернутом виде.

<u>Теорема 2'</u>. Для оптимальности допустимого вектора (1) задачи I необходимо и достаточно существование вектора (5), удовлетворяющего условиям:

а) $\sum\limits_{j\in N} a_{ij} y_j = c_i$, если $x_i$ лежит внутри д. и. (допустимого интервала);

б) $\sum\limits_{j\in N} a_{ij} y_j \geq c_i$, если $x_i$ совпадает с левым концом д. и.;

в) $\sum\limits_{j\in N} a_{ij} y_j \leq c_i$, если $x_i$ совпадает с правым концом д. и.;

г) $y_j = 0$, если $\sum\limits_{i\in M} a_{ij} x_i < b_j$;

д) $y_j = h_j$, если $\sum\limits_{i\in M} a_{ij} x_i > b_j$;

е) $y_i \geq 0$ при $j \in N_2$;

ж) $0 \leq y_j \leq h_j$ при $j \in N_3$.

При этом вектор (5) является оптимальным в задаче II.

Уже в первой работе по линейному программированию [2] для изучавшихся задач были сформулированы признаки оптимальности в форме, аналогичной теореме 2'. Компоненты соответствующих векторов (5) были названы разрешающими множителями, а основанный на их использовании алгорифм – методом разрешающих множителей. В настоящее время разработан ряд эффективных приемов решения задач линейного программирования. Все они опираются на признаки оптимальности и явно или неявно используют идею разрешающих множителей. Поэтому они могут рассматриваться как различные реализации общего метода разрешающих множителей. Одна из таких реализаций изучается в настоящей работе.

## § 2. Метод улучшения допустимого вектора

Идея метода состоит в следующем. Для некоторого исходного допустимого вектора (1) на основании части условий теоремы 2' составляется система линейных уравнений, из которой находится вектор (5). Если для него выполняются также остальные условия теоремы $2'$, то рассматриваемый вектор (1) является оптимальным в задаче I, а найденный вектор (5) – оптимальный в задаче II.

При нарушении одного из условий теоремы $2'$ строится новый допустимый вектор (1) такой, что значение функции (2) увеличивается. Через конечное число таких шагов получаем оптимальные векторы задач I и II либо убеждаемся в том, что в рассматриваемых задачах оптимальных векторов не существует.

Метод улучшения допустимого вектора обычно излагается для задач линейного программирования, приведенных к одной из канонических форм (см., например, [3], стр. 315-326). Здесь этот метод используется для непосредственного решения задач I и II.

Рассмотрим $n$-мерные векторы

$$\alpha_i=(a_{i1},a_{i2},\dots,a_{in}),\ i\in M,\ \beta=(b_1,b_2,\dots,b_n),$$

$$e_j=(\overbrace{0,\dots,0,1}^{j},0,\dots,0),\quad j\in N,$$

и соотношения вида

$$\sum_{i\in I} u_i\alpha_i+\sum_{j\in J_0} v_j e_j-\sum_{j\in J_h} w_j e_j=\beta-\sum_{i\in I_g} g_i\alpha_i, \qquad (10)$$

где

$$I=I_1\cup I_2\cup I_3,\quad I_1\subset M_1,\quad I_2\subset M_2,\quad I_3\subset M_3,\quad I_g\subset M_3,$$

$$I_3\cap I_g=\varphi,\quad J_0\subset N_2\cup N_3,\quad J_h\subset N_3,\quad J_0\cap J_h=\varphi\ ;$$

$$u_i\in\begin{cases}(-\infty,+\infty) & \text{при}\quad i\in I_1,\\ (0,+\infty) & \text{при}\quad i\in I_2,\\ (0,g_i) & \text{при}\quad i\in I_3,\end{cases}$$

$$v_j>0\ \ (j\in J_0),\qquad w_j>0\ \ (j\in J_h).$$

Каждому соотношению типа (10) отвечает допустимый вектор задачи I с компонентами

$$x_i=\begin{cases}u_i & \text{при}\quad i\in I,\\ g_i & \text{при}\quad i\in I_g,\\ 0 & \text{при}\quad i\in M\setminus(I\cup I_g).\end{cases}$$

Наоборот, каждому допустимому вектору (1) задачи I отвечает соотношение (10), в котором

$$I_g=\{i\in M_3: x_i=g_i\},$$

$$\{i\in M : x_i \neq 0\} \setminus I_g \subset I \subset \{i\in M : x_i \text{ лежит внутри д. и.}\},\ u_i = x_i,$$

$$J_0 = \{j\in N_2 \cup N_3 : \sum_{i\in M} a_{ij} x_i < b_i\}, \qquad v_j = b_j - \sum_{i\in M} a_{ij} x_i,$$

$$J_h = \{j\in N_3 : \sum_{i\in M} a_{ij} x_i > b_j\}, \quad w_j = \sum_{i\in M} a_{ij} x_i - b_j.$$

Основная часть алгорифма. Предположим вначале выполненным следующее условие:

(!) Число слагаемых в левой части любого соотношения типа (I0) не меньше $n$.

З а м е ч а н и е. При фиксированных множествах $I, J_0, J_h, I_g$ соотношение (I0) можно рассматривать как систему линейных уравнений относительно $u_i$, $v_j$, $w_j$. Если число неизвестных в системе меньше числа уравнений, то она, как правило, несовместна. Поэтому условие (!), вообще говоря, выполняется. Во всяком случае, его выполнения можно добиться сколь угодно малым изменением заданного вектора $\beta$.

Процесс начинается с некоторого соотношения типа (I0), в левой части которого ровно $n$ слагаемых. Опираясь на условие (!), можно показать, что соответствующие векторы

$$\alpha_i\,(i\in I), \qquad e_j\,(j\in J_0), \quad -e_j\,(j\in J_h) \tag{II}$$

образуют базис рассматриваемого $n$-мерного пространства. Благодаря этому система линейных уравнений относительно компонент вектора (5)

$$(\alpha_i, y) = c_i\ (i\in I), \quad (e_j, y) = 0\ (j\in J_0), \quad (-e_j, y) = -h_j\ (j\in J_h), \tag{I2}$$

составленная на основании части условий признака оптимальности (ср. условия "а", "г" и "д" теоремы 2$'$), имеет единственное решение. Находим это решение и проверяем остальные условия признака. Если они выполняются, то вектор (1), отвечающий рассматриваемому соотношению (I0), является оптимальным в задаче I, а найденный вектор (5) - оптимальный в задаче II (процесс окончен).

В случае нарушения, по крайней мере, одного из условий теоремы 2$'$ выбираем вектор

$$\alpha = \begin{cases} \alpha_{i_0}, & \text{если при } i_0\in M\setminus I_g : (\alpha_{i_0}, y) < c_{i_0}, \\ -\alpha_{i_0}, & \text{если при } i_0\in M_1 \cup I_g : (\alpha_{i_0}, y) > c_{i_0}, \\ e_{j_0}, & \text{если при } j_0\in N_2\cup N_3 : (e_{j_0}, y) < 0, \\ -e_{j_0}, & \text{если при } j_0\in N_3 : (-e_{j_0}, y) < -h_{j_0}, \end{cases}$$

и находим его разложение по базисным векторам (11):

$$\alpha=\sum_{i\in I}\bar{u}_i\alpha_i+\sum_{j\in J_0}\bar{v}_j e_j-\sum_{j\in J_h}\bar{w}_j e_j . \tag{13}$$

Рассматриваем соотношение, вытекающее из (10) и (13):

$$\begin{gathered}\varepsilon\alpha+\sum_{i\in I}(u_i-\varepsilon\bar{u}_i)\alpha_i+\sum_{j\in J_0}(v_j-\varepsilon\bar{v}_j)e_j-\\ -\sum_{j\in J_h}(w_j-\varepsilon\bar{w}_j)e_j=\beta+\sum_{i\in I_g}g_i\alpha_i .\end{gathered}\tag{14}$$

Если $\varepsilon>0$ удовлетворяет условиям:

$$\begin{gathered}u_i-\varepsilon\bar{u}_i\in\begin{cases}(-\infty,+\infty) & \text{при } i\in I_1 ,\\ [0,+\infty) & \text{при } i\in I_2 ,\\ [0,g_i] & \text{при } i\in I_3 ,\end{cases}\\ v_j-\varepsilon\bar{v}_j\geqslant 0\ \ (j\in J_0),\quad w_j-\varepsilon\bar{w}_j\geqslant 0\ \ (j\in J_h)\\ \varepsilon\leqslant h_{i_0},\ \text{если}\quad \alpha=\pm\alpha_{i_0},\ \text{причем}\ i_0\in M_3 ,\end{gathered}\tag{15}$$

то соотношению (14) отвечает допустимый вектор $x'$ задачи I. Максимизируемая функция (2) на этом векторе принимает большее значение, чем на векторе $x$, отвечающем исходному соотношению (10). А именно

$$\mu(x')=\mu(x)+\varepsilon\Delta ,$$

где

$$\Delta=\begin{cases}c_{i_0}-(\alpha_{i_0},y), & \text{если}\quad \alpha=\alpha_{i_0} ,\\ (\alpha_{i_0},y)-c_{i_0}, & \text{если}\quad \alpha=-\alpha_{i_0} ,\\ -(e_{j_0},y), & \text{если}\quad \alpha=e_{j_0} ,\\ (e_{j_0},y)-h_{j_0}, & \text{если}\quad \alpha=e_{j_0} .\end{cases}$$

Таким образом, мы заинтересованы в выборе возможно большего $\varepsilon$.

Если условия (15) выполняются при сколь угодно больших $\varepsilon$, функция (2) на множестве допустимых векторов не ограничена (сверху) и, в силу теоремы 1, в рассматриваемых задачах I и II оптимальных векторов не существует (процесс окончен).

В противном случае находим максимальное $\varepsilon$, удовлетворяющее условиям (15), и подставляем его в (14). При этом получаем новое соотношение типа (IO), в левой части которого, в силу условия (!), ровно $n$ слагаемых, и процесс можно продолжить.

Легко показать, что описанный процесс не может продолжаться неограниченно. Действительно, ввиду монотонности процесса мы не можем дважды прийти к одному и тому же соотношению (IO), каждое из них однозначно определяется соответствующим набором множеств $I$, $J_0$, $J_h$, $J_g$, а различных наборов такого рода - конечное число. Следовательно, через конечное число шагов будут получены оптимальные векторы для задач I и II либо будет установлено, что в рассматриваемых задачах оптимальных векторов не существует.

Для завершения описания метода остается рассмотреть вопросы, связанные с возможным нарушением предположения (!), а также указать приемы построения исходного соотношения (IO).

Проблема вырождения. В случае нарушения условия (!) в описанном процессе приходится допускать соотношения (IO), в которых векторы (11) образуют $n$-мерный базис, однако некоторые из коэффициентов $u_i$, $v_j$, $w_j$ выходят на границу указанных выше для них открытых интервалов (см. стр. 7). При этом возникают так называемые ситуации вырождения. Порядком вырождения называется число коэффициентов, вышедших на границу допустимых интервалов. Известно, что при вырождениях выше первого порядка рассматриваемый процесс, в принципе, может зациклиться. Однако, как показывает опыт, практически с этим вряд ли стоит считаться. Значительно больше шансов не окончить счета по другим причинам, например в связи с плохой обусловленностью встречающихся систем линейных уравнений.

Для уменьшения и без того малой вероятности зацикливания процесса можно предусмотреть в ситуациях вырождения незначительные изменения вышедших на границу коэффициентов, что соответствует малой вариации исходного вектора $\beta$.

Наконец, полностью гарантировать себя от зацикливания можно с помощью разработанного Чарнсом [4] приема бесконечно малой вариации вышедших на границу коэффициентов. Один из них изменяется на $\delta$, другой на $\delta^2$ и т. д. При этом, правда, коэффициенты $u_i$, $v_j$, $w_j$ в рассматриваемых соотношениях (IO) превращаются в полиномы от $\delta$ степени $r$, равной порядку вырождения. Для хранения этих полиномов в машине необходимо отвести специальное место, что при решении задач большого объема крайне затруднительно.

<u>Построение исходного допустимого вектора</u>. Исходный допустимый вектор (1) и отвечающее ему соотношение (10), необходимое для решения задач I и II описанным методом, могут быть получены с помощью того же метода, примененного к следующей вспомогательной задаче.

<u>Задача I</u>$'$. Определить $(m+n_1)$-мерный вектор $x$ с компонентами $x_i (i \in M)$, $x_{m+j} (j \in N_1)$, максимизирующий функцию

$$\mu(x) = -\sum_{j \in N_1} x_{m+j} - \sum_{j \in N_2} \left[ \sum_{i \in M} a_{ij} x_i - b_j \right]^+ \tag{16}$$

при ограничениях:

$$x_i \in \begin{cases} (-\infty, +\infty) & \text{для} \quad i \in M_1, \\ [0, +\infty) & \text{для} \quad i \in M_2, \\ [0, g_i] & \text{для} \quad i \in M_3, \end{cases}$$
$$x_{m+j} \in [0, +\infty) \quad \text{для} \quad j \in N_1,$$

$$\sum_{i \in M} a_{ij} x_i + \varepsilon_j x_{m+j} = b_j, \quad i \in N_1,$$

где

$$\varepsilon_j = \begin{cases} +1 & \text{при} \quad b_j \geq 0, \\ -1 & \text{при} \quad b_j < 0. \end{cases}$$

Ясно, что эта задача типа задачи I. В ней

$m' = m + n_1$, $n' = n_1 + n_2$, $M' = M \cup \{i = m+j : j \in N_1\}$, $N' = N_1 \cup N_2$,

$M_1' = M_1$, $M_2' = M_2 \cup (M' \setminus M)$, $M_3' = M_3$, $N_1' = N_1$, $N_2' = \phi$, $N_3' = N_2$,

$c_i' = 0$ для $i \in M$, $c_i' = -1$ для $i \in M' \setminus M$, $h_j' = 1$ для $j \in N_3'$.

Векторами $\alpha_i' (i \in M)$, $\alpha_i' (i \in M' \setminus M)$, $e_j' (j \in N')$, $\beta'$ здесь служат $(n_1 + n_2)$ -мерные вырезки из $n$ -мерных векторов $\alpha_i$, $\varepsilon_{i-m} e_{i-m}$, $e_j$, получающиеся удалением компонент с номерами $j \in N_3$.

В приведенной задаче I$'$ имеется допустимый вектор

$$x_i=\begin{cases} 0 & \text{при } i\in M, \\ |\beta_{i-m}| & \text{при } i\in M'\setminus M, \end{cases} \tag{17}$$

и максимизируемая функция (16) на множестве допустимых векторов ограничена сверху $(\mu(x)\leqslant 0)$. Следовательно, в рассматриваемой задаче существует оптимальный вектор (теорема 1), который может быть найден с помощью описанного процесса. При этом в качестве исходного может быть принято соотношение типа (10), отвечающее допустимому вектору (17). Это соотношение имеет вид

$$\sum_{i\in I'} u_i\alpha_i' + \sum_{j\in J_0} v_j e_j' - \sum_{j\in J_h'} w_j e_j' = \beta' - \sum_{i\in I_g'} g_i\alpha_i', \tag{18}$$

где

$$I'=M'\setminus M,\quad J_0'=\{j\in N_3' : \beta_j\geqslant 0\},\quad J_h'=\{j\in N_3' : \beta_j<0\},\quad I_g'=\phi,$$

$$u_i=|\beta_{i-m}|,\quad v_j=\beta_j,\quad w_j=-\beta_j .$$

Векторы $\alpha_i'(i\in I')$, $e_j'(j\in J_0')$, $-e_j'(j\in J_h')$, как нетрудно показать, образуют базис рассматриваемого $n'$-мерного пространства.

Исходя из указанного соотношения, через конечное число шагов получаем новое соотношение (18), отвечающее оптимальному вектору задачи I$'$. При этом могут встретиться три случая:

1. В полученном соотношении (18) $I'\subset M$, $J_h'=\phi$.

2. По крайней мере, одно из условий предыдущего пункта нарушается, причем для некоторого $i\in I'\setminus M$ или $j\in J_h'$ соответствующий коэффициент $u_i$ или $w_j$ отличен от нуля.

3. Условия пункта 1 нарушены, но для всех $i\in I'\setminus M$ и $i\in J_h'$ соответствующие коэффициенты $u_i$ и $w_j$ равны нулю.

В первом случае вектор (1) с компонентами

$$x_i=\begin{cases} u_i & \text{при } i\in I', \\ g_i & \text{при } i\in I_g', \\ 0 & \text{при } i\in M\setminus(I'\cup I_g') \end{cases}$$

является допустимым в исходной задаче I и отвечающее этому вектору соотношение (10) может быть принято за исходное при решении задач I и II методом улучшения допустимого вектора. В этом соотношении (IO)

$$I=I', \ I_g=I'_g, \ J_0=J'_0 \cup \{j\in N_3 : \sum a_{ij}x_i \leq b_j\},$$

$$J_h=\{j\in N_3 : \sum a_{ij}x_i > b_j\}$$

коэффициенты $u_i (i\in I)$, $v_j (j\in J'_0)$ совпадают с соответствующими коэффициентами полученного соотношения (I8), а коэффициенты добавленных слагаемых вычисляются по формулам:

$$v_j = b_j - \sum_{i\in M} a_{ij}x_i, \quad w_i = \sum_{i\in M} a_{ij}x_i - b_j .$$

Во втором случае условия (3) и (4) исходной задачи I несовместны (в этой задаче нет допустимого вектора), и потому ни в одной из рассматриваемых задач I и II нет оптимального вектора.

В последнем случае условия (3) и (4) совместны, однако при сколь угодно малом изменении величин $b_j$ эти условия становятся несовместными. Решать соответствующие задачи I и II вряд ли целесообразно.

## § 3. Вопросы практической реализации метода

При решении задач I и II методом улучшения допустимого вектора наиболее трудоемким является вычисление на каждом шаге соответствующего вектора (5) и разложение вводимого вектора $\alpha$ по базисным векторам (II), т. е. решение систем линейных уравнений типа (I2) и (I3). Число неизвестных в этих системах определяется количеством переменных в задаче II. С другой стороны, при рассмотрении пары двойственных задач в качестве задачи II всегда может быть принята та из этих задач, которая содержит меньшее количество переменных. Поэтому под размерностью задачи линейного программирования естественно понимать минимальное из чисел, выражающих количество переменных в данной и двойственной к ней задачах. При записи этих задач в форме задач I и II номера этих задач следует выбирать так, чтобы имело место неравенство $n \leq m$.

В общем случае для решения систем (I2) и (I3) можно, как это делается в модифицированном симплекс-методе (см., например,

[5], стр. 119–135), использовать обратную матрицу к матрице $A$, строками которой служат базисные векторы (11). При этом на каждом шаге описанного процесса в базисе (11) заменяется не более одного вектора, что позволяет новую обратную матрицу получать из матрицы предыдущего шага простым преобразованием.

З а м е ч а н и е. Вместо матрицы $A^{-1}$ порядка $n$ для решения систем (12) и (13) достаточно иметь обратную матрицу к матрице $\tilde{A}$, которая получается из $A$ исключением столбцов с номерами $j \in J_0 \cup J_h$ и строк, отвечающих векторам $e_j$ $(j \in J_0)$ и $-e_j$ $(j \in J_h)$. Порядок этой матрицы совпадает с числом элементов в множестве $I$ и от шага к шагу, вообще говоря, меняется, но не более чем на 1.

При сравнительно небольших задачах (размерности несколько меньшей корня квадратного из емкости оперативной памяти машины) матрица $A^{-1}$ (или $\tilde{A}^{-1}$) хранится в оперативной памяти. Кроме того, там выделяется небольшое рабочее поле и отводится место для коэффициентов $u_i$, $v_j$, $w_j$ соотношения (10) и отвечающих им индексов с отметками о их принадлежности множествам $J_0$, $J_h$, для чисел $c_i$ $(i \in I)$, $g_i$ $(i \in I \cap M_3)$, $h_j$ $(j \in N_3)$, для величин $y_j$ $(j \in N)$, представляющих решение системы (12), и, наконец, для коэффициентов $\bar{u}_i$, $\bar{v}_j$, $\bar{w}_j$ разложения (13). Векторы $\alpha_i$ (точнее, ненулевые компоненты этих векторов) и отвечающие им величины $c_i$, $g_i$ с отметками о принадлежности индекса $i$ множествам $M_1$, $M_2$, $M_3$, а также о включении этого индекса в $I_g$ находятся во внешней памяти. В оперативную память эти данные вводятся небольшими массивами при проверке условий признака оптимальности.

При решении задач I и II большей размерности матрицу $A^{-1}$ приходится размещать во внешних запоминающих устройствах (на магнитных барабанах или лентах). В этом случае оказывается удобным использовать мультипликативное представление обратной матрицы (см., например, [5], стр. 136–139). Матрица $A^{-1}$ хранится в виде произведения так называемых элементарных матриц, каждая из которых задается информацией, занимающей $n$ ячеек. На каждом шаге в описанном процессе в представлении матрицы $A^{-1}$ добавляется не более одного множителя.

Все сказанное относилось к задачам линейного программирования общего вида. Большим резервом для повышения размерности является разработка специальных алгорифмов для отдельных классов задач. Такие алгорифмы могут строиться на основе изложенного метода улучшения допустимого вектора. Специфика рассматриваемого класса задач учитывается лишь используемыми приемами решения встречающихся в процессе систем линейных уравнений. Эти системы решаются непосредственно, без использования громоздкого аппарата обратных матриц.

При разработке специальных алгорифмов оказывается полезным рассматриваемое здесь понятие ранга задачи[ж)].

Сложностью строения квадратной матрицы порядка $n$ условимся считать минимальное из чисел $r$ таких, что изменением нумерации строк и столбцов эта матрица сводится к виду $A=[a_{ij}]_{i,j}=1,2,\ldots,n$, где элементы $a_{ij}$ с номерами $i+j\leqslant n-r$ равны нулю.

Задачи линейного программирования некоторого частного вида будем называть задачами ранга $r$, если неособенные матрицы, составленные из векторов $\alpha_i (i\in M)$, $e_j\ (j\in N)$, в каждой из этих задач имеют сложность строения не выше $r$ и, по крайней мере, в одной из задач рассматриваемого вида имеются матрицы сложности $r$.

Отметим, что в определении ранга задачи допустимые интервалы для переменных не участвуют. Следовательно, при изменении этих интервалов и, в частности, при добавлении ограничений на отдельные переменные ранг задачи не меняется. В численных методах при этом уточнения требуют лишь некоторые детали.

Среди задач линейного программирования наиболее простыми являются так называемые транспортные задачи (в различных постановках). Нетрудно показать, что это задачи нулевого ранга. На этом факте основаны эффективные алгорифмы решения таких задач, в частности известный метод потенциалов (см. [6], а также [11]).

Рассмотренная в [7] задача об оптимальном использовании средств при выполнении нескольких видов работ - это задача первого ранга, и на этом базируется предложенный в [7] алгорифм.

---

ж) Это понятие было введено автором в докладе на 4-й Ленинградской конференции по применению математики в экономике в 1961 г.

В принципе этот алгорифм позволяет решать всевозможные задачи первого ранга. В уточнении нуждаются лишь отдельные детали. Иллюстрацией может служить статья [10], посвященная широкому классу задач первого ранга.

Для произвольных задач второго и более высоких рангов специальных алгорифмов в литературе до сих пор не встречалось. Это связано с тем, что приемы перенумерации строк и столбцов в соответствующих матрицах при повышении ранга задачи сильно усложняются. Между тем при увеличении размерности решаемых задач специальные алгорифмы начинают давать столь ощутимый эффект, что становится оправданным применение даже весьма сложных приемов перенумерации строк и столбцов встречающихся матриц. В настоящее время, как видно, уже назрела необходимость в разработке специальных алгорифмов, во всяком случае для задач второго ранга.

Ввиду трудности определения ранга рассматриваемых задач и сложности приемов перенумерации строк и столбцов в задачах высоких рангов нам представляется перспективной разработка упрощенных приемов, позволяющих матрицы с большим числом нулевых элементов приводить к рассмотренному выше виду, где $r$ достаточно мало, хотя и не минимальное из возможных. Даже примитивные приемы такого рода могут оказаться весьма полезными.

## § 4. Задачи линейного программирования с блочными матрицами

В [8] была рассмотрена задача линейного программирования, в которой все переменные разбивались на $m$ групп. На переменные каждой группы накладывалось одно ограничение и, кроме того, имелось $n$ ограничений, связывающих переменные различных групп. Иными словами, речь шла о задаче, в которой $m'=\sum_{i=1}^{m} l_i$ переменных ($l_i$ - число переменных в $i$-ой группе) были связаны ограничениями в количестве $n'=m+n$. Между тем анализ этой задачи проводился в $n$-мерном пространстве, и в соответствующем признаке оптимальности фигурировало лишь $n$ разрешающих множителей - компонент вектора $y$. Это означает, что при решении задач I и II указанного типа методом улучшения допустимого вектора встречающиеся системы линейных уравнений порядка $m+n$ сводятся к системам порядка $n$. На этой основе Р. А. Звягиной (см. настоящий сборник) был разработан алгорифм и составлена программа для решения задач линейного программирования, в которых матрица $A$, образованная упоминавшимися выше векторами (эти векторы служат строками матрицы $A$), имеет следующую блочную структуру:

$$A = \begin{pmatrix} \boxed{A_1} & & & 0 & \\ & \boxed{A_2} & & & \\ & & \ddots & & A_0 \\ 0 & & & \boxed{A_r} & \end{pmatrix}, \tag{19}$$

где $A_0$ – произвольная прямоугольная матрица размерности $m \times n_0$, а матрицы $A_1, A_2, \ldots, A_r$ имеют по одному столбцу. Эти задачи, содержащие $n = n_0 + r$ ограничений, решаются как задачи размерности $n_0$.

В. А. Булавский заметил, что указанный принцип можно обобщить и применить к задачам линейного программирования с блочными матрицами вида (19), где $A_1, A_2, \ldots, A_r$ – произвольные прямоугольные матрицы размерности $m_k \times n_k$ $(k = 1, 2, \ldots, r)$. Это позволяет в методе улучшения допустимого вектора вместо обратной матрицы порядка $n = \sum_{k=0}^{r} n_k$ ввести $r+1$ обратных матриц порядка $n_0, n_1, \ldots, n_r$. Соответствующий алгорифм и программа для ЭВМ были разработаны Р. А. Звягиной [12]. Этот алгорифм существенно отличается от предложенного ранее Данцигом и Вулфом [9] для задач того же типа и, как видно, является более удобным.

В [12] рассматриваются задачи I и II, в которых

$$M_1 = M_3 = N_3 = \phi .$$

Однако предложенные алгорифмы легко обобщаются на случай произвольных задач I и II с блочными матрицами соответствующей структуры.

Рассмотренные приемы конкретизации метода улучшения допустимого вектора с учетом ранга задачи и блочной структуры соответствующих матриц в некоторых случаях удается сочетать. Эффективность такого сочетания иллюстрируется в следующем параграфе на примере упоминавшейся задачи об оптимальном использовании средств при выполнении нескольких видов работ.

## § 5. Пример одновременного учета ранга задачи и блочного строения матрицы

Интересующая нас задача состоит в разыскании прямоугольной матрицы

$$x=[x_{jk}] \quad (j\in N=\{1,\dots,n\},\quad k\in P=\{1,\dots,p\}), \tag{20}$$

минимизирующей функцию

$$\mu(x)=\sum_{j\in N}\sum_{k\in P} c_{jk}x_{jk}$$

при ограничениях:

$$x_{jk}\geqslant 0,\quad j\in N,\quad k\in P, \tag{21}$$

$$\sum_{k\in P} a_{jk}x_{jk}\leqslant g_j,\quad j\in N, \tag{22}$$

$$\sum_{j\in N} x_{jk}=h_k,\quad k\in P, \tag{23}$$

где $c_{jk}$, $a_{jk}$, $g_j$, $h_k$ – заданные положительные числа.

Из приведенного выше (§ 1, теорема $2'$) признака оптимальности для общей задачи линейного программирования вытекает

Теорема $2''$. Для оптимальности допустимой матрицы (20) (удовлетворяющей условиям (21)-(23)), необходимо и достаточно существование вектора

$$y=(y_1,y_2,\dots,y_n,z_1,z_2,\dots,z_p) \tag{24}$$

такого, что

а) $y_j\geqslant 0,\ j\in N$;

б) $a_{jk}y_j+z_k\geqslant -c_{jk},\quad j\in N,\quad k\in P$;

в) $a_{jk}y_j+z_k=-c_{jk}$ при $(j,k)\in I=\{(j,k): x_{jk}>0\}$;

г) $y_j=0$ при $j\in J_0=\{j\in N: x_j=g_j-\sum_{k\in P}a_{jk}x_{jk}>0\}$.

В рассматриваемой задаче размерности $n+p$ каждой допустимой матрице (20) отвечает в $(n+p)$-мерном пространстве следующее соотношение типа (10):

$$\sum_{(j,\kappa)\in I} u_{j\kappa}\alpha_{j\kappa} + \sum_{j\in J_0} v_j e_j = \beta , \qquad (25)$$

где

$$\alpha_{j\kappa} = a_{j\kappa} e_j + e_{n+\kappa}, \quad j\in N, \quad \kappa\in P;$$

$e_j$ - орты соответствующих осей;

$$\beta = (g_1, g_2, \ldots, g_n, h_1, h_2, \ldots, h_p);$$

$$u_{j\kappa} = x_{j\kappa}, \qquad v_j = x_j = g_j - \sum_{\kappa\in P} a_{j\kappa} x_{j\kappa} .$$

При решении этой задачи изложенным методом улучшения допустимого вектора на каждом шаге имеем соотношение (25), в котором векторы

$$\alpha_{j\kappa}, \quad (j,\kappa)\in I, \quad e_j, \quad j\in I_0 \qquad (26)$$

образуют $(n+p)$-мерный базис. Из системы

$$(a_{j\kappa} e_j + e_{n+\kappa}, y) = -c_{j\kappa}, \quad (j,\kappa)\in I, \qquad (27)$$
$$(e_j, y) = 0, \quad j\in J_0 ,$$

определяется вектор (24), и для него проверяются условия "а" и "б" теоремы 2''. Если они выполнены, то отвечающая соотношению (25) допустимая матрица (20) является оптимальной. В противном случае необходимо найти разложение вектора

$$\alpha = \begin{cases} e_{j_0}, & \text{если} \quad (e_{j_0}, y) < 0, \\ a_{j_0\kappa_0} e_{j_0} + e_{n+\kappa_0}, & \text{если} \quad (\alpha_{j_0\kappa_0}, y) < c_{j_0\kappa_0}, \end{cases}$$

по базисным векторам (26), т. е. определить коэффициенты в соотношении

$$\alpha = \sum_{(j,\kappa)\in I} \bar{u}_{j\kappa}(a_{j\kappa} e_j + e_{n+\kappa}) + \sum_{j\in J_0} \bar{v}_j e_j . \qquad (28)$$

Благодаря блочному строению матрицы рассматриваемой задачи системы (27) и (28) могут быть сведены к системам меньшей размерности. Для этого множество пар $(j,k)\in I$ разбивается на подмножества $I_k$ с фиксированным вторым индексом. Эти подмножества, очевидно, непустые. Каждому $k\in P$ сопоставим некоторый индекс $j_k$ такой, что $(j_k,k)\in I_k$, и рассмотрим множества:

$$I'_k=I_k\setminus\{(j_k,k)\},\quad P'=\{k\in P: I'_k\neq\phi\},\quad I'=\bigcup_{k\in P'} I'_k .$$

Нетрудно проверить, что первые $n$ компонент вектора (24), представляющего решение системы (27), могут быть найдены из системы

$$\begin{aligned} &(a_{jk}e'_j-a_{j_k k}e'_{j_k}, y')=c_{j_k k}-c_{jk}, \quad (j,k)\in I', \\ &(e'_j,y')=0, \quad j\in J_0, \end{aligned} \tag{27'}$$

где $y'=(y_1,y_2,\ldots,y_n)$ - неизвестный вектор, $e'_j$ - орты соответствующих осей в $n$-мерном пространстве. Остальные компоненты вектора (24) находятся независимо друг от друга по формуле

$$z_k=-a_{j_k k}y_{j_k}-c_{j_k k}, \quad k\in P.$$

Коэффициенты соотношения (28), отличные от $\bar u_{j_k k} (k\in P)$, можно найти из системы

$$\sum_{(j,k)\in I'}\bar u_{jk}(a_{jk}e'_j-a_{j_k k}e'_{j_k})+\sum_{j\in J_0}\bar v_j e'_j=\alpha', \tag{28'}$$

где

$$\alpha'=\begin{cases} e'_{j_0}, & \text{если } \alpha=e_{j_0}, \\ a_{j_0 k_0}e'_{j_0}-a_{j_{k_0} k_0}e'_{j_{k_0}}, & \text{если } \alpha=\alpha_{j_0 k_0}. \end{cases}$$

Остальные коэффициенты разложения (28) при $\alpha=e_{j_0}$ определяются по формулам

$$\bar u_{j_k k}=\begin{cases} -\sum\limits_{(j,k)\in I'_k}\bar u_{jk} & \text{для } k\in P', \\ 0 & \text{для } k\in P\setminus P'. \end{cases} \tag{29}$$

При $\alpha=\alpha_{j_0 k_0}$

$$\bar u_{j_{k_0} k_0}=\begin{cases} 1-\sum\limits_{(j,k)\in I'_{k_0}}\bar u_{jk}, & \text{если } k_0\in P', \\ 1, & \text{если } k_0\in P\setminus P', \end{cases}$$

а $\bar u_{j_k k}$ для $k\neq k_0$ находятся по формулам (29).

Следовательно, рассматриваемая задача размерности $n+p$ решается как задача размерности $n$. Далее, матрицы систем (27) и (28) имели сложность строения, равную единице. Такую же сложность имеют матрицы систем (27′) и (28′). Это позволяет решение этих систем получать с помощью упрощенных приемов, аналогичных предложенным в работе [7]. Такие приемы для рассматриваемого случая детально описаны в статье [10], посвященной так называемым двухкомпонентным задачам.

Приведенные рассуждения показывают, что при решении задач рассматриваемого типа в оперативной памяти машин, помимо небольшого рабочего поля, достаточно отвести место для следующих величин:

$a_{j_k k}$, $u_{j_k k}$, $\bar{u}_{j_k k}$ $(k \in P')$ с указанием индексов $j_k$;

$v_j$, $\bar{v}_j$ $(j \in J_0)$, $a_{jk}$, $-c_{jk}+c_{j_k k}$, $u_{jk}$, $\bar{u}_{jk}$ $((j,k) \in I')$ с указанием индексов $j$ и расположения чисел $a_{j_k k}$, $u_{j_k k}$, $u_{j_k k}$;

$y_j$ $(j \in N)$ с информацией о порядке их вычисления.

Все эти данные легко размещаются на 8 полях по $n$ ячеек, что позволяет решать рассматриваемые задачи, в которых $n$ порядка 1/8 емкости оперативной памяти, а на $p$ не накладывается никаких ограничений.

В заключение заметим, что в случае необходимости указанную размерность можно повысить примерно вдвое. Для этого достаточно в оперативной памяти на каждом этапе процесса (при решении системы (27′), при проверке условий теоремы 2′, при решении системы (28′), при вычислении коэффициентов нового соотношения (25) ) сохранять лишь необходимую информацию, переписывая остальные данные, которые в дальнейшем используются, во внешнее запоминающее устройство.

## ЛИТЕРАТУРА


1. Г. Зойтендейк. Методы возможных направлений. Изд. иностр. лит., 1963.
2. Л. В. Канторович. Математические методы организации и планирования производства. Изд. ЛГУ, 1939.

3. Л. В. К а н т о р о в и ч. Экономический расчет наилучшего использования ресурсов. Изд. АН СССР, 1959.
4. C h a r n e s A. Optimality and degeneracy in linear programming. - Econometrica, 20, №2 (1952), стр. 160-170.
5. С. Г а с с. Линейное программирование. Физматгиз, 1961.
6. Л. В. К а н т о р о в и ч, М. К. Г а в у р и н. Применение математических методов в вопросах анализа грузопотоков. В сб. "Проблемы повышения эффективности работы транспорта". Изд. АН СССР, 1949, стр. 110-138.
7. М. К. Г а в у р и н, Г. Ш. Р у б и н ш т е й н, С. С. С у р и н. Об оптимальном использовании средств при выполнении нескольких видов работ. - Сибирский матем. журнал, т. 3, № 4 (1962), стр. 481-499.
8. Г. Ш. Р у б и н ш т е й н. Обобщение задачи о крайней точке пересечения оси с выпуклым многогранником. - Докл. АН СССР, т. 113, № 5 (1957), стр. 987-990.
9. G. B. D a n t z i g, P. W o l f e. Decomposition principle for linear programs. - J. Oper. Res. Soc. Am., 8 (1960), стр. 101-111.
10. М. А. Я к о в л е в а. Двухкомпонентные задачи линейного программирования. Настоящий сборник.
11. В. А. Б у л а в с к и й. Об одном алгорифме решения транспортной задачи. Настоящий сборник.
12. Р. А. З в я г и н а. Задачи линейного программирования с блочно-диагональными матрицами. Настоящий сборник.