Оптимальное планирование, вып. I0, (1968), 69-80.

УДК 512.25/.26+519.3:330.115

С.С.СУРИН

РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БОЛЬШИМ ЧИСЛОМ НУЛЕВЫХ ЭЛЕМЕНТОВ В ИСХОДНОЙ СИМПЛЕКСНОЙ ТАБЛИЦЕ

§ I. Постановка вопроса

Пусть дана задача линейного программирования: найти вектор $X=(x_1, x_2, \ldots x_n)$ из условий:

$$AX=B, \tag{I}$$

$$X \geq 0, \tag{2}$$

$$(C,X)-\min, \tag{3}$$

где A - матрица $[m \times n]$, $B-m$ - мерный вектор и $C=(c_1, c_2, \ldots c_n)$ - вектор цен.

Обозначим через $\tilde{A}$, $\tilde{X}$ и $\tilde{C}$ расширения соответственно матрицы A и векторов X и C, необходимые для решения задачи с использованием искусственного базиса. Точнее, $\tilde{A}$ - матрица A, дополненная справа единичной матрицей порядка m,

$$\tilde{X}=(x_1, x_2, \ldots, x_n, \ldots, x_{n+m}),$$

$$\tilde{C}=(c_1, c_2, \ldots, c_n, \underbrace{M, \ldots, M}_{m}),$$

где M - достаточно большое число.

Задачу (I) - (3) в новых обозначениях можно записать так:

$$\tilde{A}\tilde{X}=B\,, \qquad (\mathrm{I}')$$

$$\tilde{X}\geq 0\,, \qquad (2')$$

$$(\tilde{C},\tilde{X})=min. \qquad (3')$$

Пусть r - наибольшее число отличных от нуля элементов матрицы A, стоящих в одном столбце. Если отношение $\frac{r}{m}$ невелико, то, как будет ясно из дальнейшего, для решения на ЭВМ задачи (I)-(3) (соответственно, задачи (I')-(3'))по схеме последовательного улучшения плана возможно построить такое видоизменение этой схемы, которое позволит увеличить объем решаемых задач путем более экономичного использования запоминающего устройства машины. Подробно схема последовательного улучшения плана описана в [I] . Как известно, она состоит из следующих операций:

I. Строится начальное допустимое базисное решение задачи (I) - (3).

2. Пусть известно некоторое допустимое базисное решение X задачи (I) -(3). Это значит, что из матрицы $\tilde{A}$ можно выделить неособенную подматрицу B порядка m ж) такую, что

$$BX^*=B\,.$$

Здесь X^* - вектор, составленный из базисных компонент вектора $\tilde{X}$ жж)

Ищется вектор y (вектор оценок) из условий:

$$B^T y = C^*, \qquad (4)$$

где C^* - вектор базисных цен. Обозначим решение системы (4) через $\bar{y}$.

3. Проверяется выполнение условий оптимальности решения

$$(\bar{y},A_s)\leq C_s \qquad (s=1,2,\dots n), \qquad (5)$$

ж) Векторы, соответствующие столбцам матрицы B , составляют базис решения.

жж) Небазисные компоненты вектора $\tilde{X}$ равны нулю.

где A_s – s-й столбец матрицы A, а c_s –его цена.

Если все условия (5) выполнены, то X – искомое (оптимальное)решение задачи (I')-(3') (соответственно задачи (I)-(3)), и на этом процесс решения заканчивается. В противном случае в матрице A найдется такой столбец A_{s_0}, что введение в базис решения соответствующего ему вектора уменьшит значение линейной формы (3').

4. В базис решения X вводится вектор A_{s_0}. Для этого отыскиваются коэффициенты разложения этого вектора по векторам базиса, т.е. решается система линейных алгебраических уравнений:

$$BX_{s_0}=-A_{s_0}, \tag{6}$$

где $X_{s_0}=(x^1_{s_0}, x^2_{s_0}, \dots, x^m_{s_0})$ – вектор коэффициентов разложения. Новое решение ищется в виде:

$$\bar{X}=X+\varepsilon X_{s_0},$$

причем ε находится из условий:

$$\varepsilon=\min\left\{\frac{x^*_k}{|x^k_{s_0}|}\Big/ x^k_{s_0}<0\right\},$$

где $$(x^*_1, x^*_2, \dots, x^*_m)=x^*.$$

Решение $\bar{X}$, также как и решение X, обладает свойствами допустимости и базисности, и к нему применяются вторая, третья и, если нужно, четвертая операции схемы.

Эта схема решения не требует, как прямой симплекс-метод, одновременного хранения коэффициентов разложения по базису всех небазисных векторов, для чего при решении задачи на ЭВМ нужно отводить значительный объем запоминающего устройства. На каждом шаге алгорифма последовательного улучшения плана нужно иметь коэффициенты разложения по базису лишь одного небазисного вектора. Это позволяет матрицу A хранить в сжатом виде или записанной алгорифмически.

При выполнении пунктов 2 и 4 схемы приходится решать системы линейных алгебраических уравнений порядка m. Поэтому для того, чтобы иметь возможность решать на ЭВМ задачи линейного программирования большого объема по схеме последовательного улучшения плана, надо фактически уметь решать на ЭВМ системы линейных алгебраических уравнений высокого порядка, что связано с преодолением ограниченности объема внутреннего запоминающего устройства машины.

Когда дробь $\frac{z}{m}$ невелика, оказывается эффективным начинать решение системы (4) и (6) с приведения матрицы путем перестановки в ней строк и столбцов к треугольному или к почти треугольному виду. В первом случае все неизвестные вычисляются рекуррентно. Во втором-это удается сделать лишь для части неизвестных. Для остальных неизвестных приходится решать систему линейных алгебраических уравнений общего вида порядка ниже m, используя для этого общие методы линейной алгебры.

В § 2 описывается алгорифм приведения и схема вычисления неизвестных для системы (4).

Так как на одном шаге решения задачи по схеме последовательного улучшения плана матрица системы (6) получается из матрицы системы (4) с помощью операции транспонирования, то алгорифм приведения можно применять при решении лишь одной системы, например системы (4). Тогда при решении системы (6) можно воспользоваться информацией приведения, полученной при работе алгорифма. Этому посвящен § 3.

§ 2. Вычисление вектора оценок

Введем следующие обозначения:

y_i - i -тая компонента вектора оценок;

b_{ij} - элемент матрицы B, стоящий на пересечении i - той строки и j -того столбца матрицы.

В этих обозначениях система (4) запишется так:

$$\sum_{i=1}^{m} b_{ij} y_i = c_j^* \qquad (j = 1, 2, \ldots, m), \tag{7}$$

где c_j^* - j - тая компонента вектора C^*.

Введем обозначения:

n_j - j - тое уравнение системы (7);

$m_j = \{i / b_{ij} \neq 0\}$;

k - переменная, принимающая целые значения от I до m.

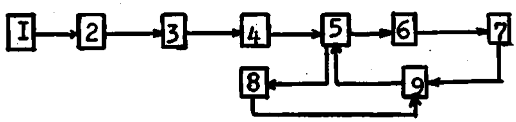
Символ "оп. q" следует читать; "оператор № q",

Выражение "оп. q" передает управление" оп. t" означает, что по выполнении оператора № q следует перейти к выполнению оператора № t.

Через $\{x\}$ будем обозначать множество, состоящее из одного элемента, а именно из элемента x .

Под символом $|X|$ будем понимать число элементов множества X, если X - множество, и абсолютную величину числа X, если X - число.

Переходим к описанию алгорифмов.



Черт. I.

На черт.I дана блок-схема алгорифма приведения. Ниже приводится описание отдельных операторов схемы.

Оп.I присваивает переменной k значение "I" и передает управление оп.2.

Оп.2 находит значение индекса j, удовлетворяющее условию:

$$|m_j| = \min_{1 \le j' \le m} |m_{j'}|,$$

а если таких значений несколько, то фиксирует наименьшее из них. Затем оп.2 передает управление оп.3.

Оп.3 для значения индекса j, найденного оп.2, находит значение индекса i, удовлетворяющее условию:

$$|b_{ij}| = \max_{i' \in m_j} |b_{i'j}|,$$

а если таких значений несколько, то фиксирует наименьшее из них. После этого оп.3 передает управление оп.4.

Оп.4 вводит в рассмотрение множества значений индексов M_1^k, M_2^k, N_1^k, N_2^k, полагая

$$M_1^k = \{i\},$$
$$M_2^k = m_j \setminus \{i\},$$
$$N_1^k = \{j\},$$
$$N_2^k = \Lambda,$$

и фиксирует упорядоченную тройку чисел $\tau = (i, j, k)$.
Здесь i и j - значения индексов, найденные соответственно оп.3 и 2. Затем оп.4 передает управление оп.5, полагая $k := k+1$. Оп.5 вводит в рассмотрение множества $\bar{m}_j^k$ и $\bar{\bar{m}}_j^k$, определяемые равенствами

$$\bar{m}_j^k = m_j \cap (M_1^{k-1} \cup M_2^{k-1}),$$
$$\bar{\bar{m}}_j^k = m_j \setminus \bar{m}_j^k,$$

и находит значение индекса j, удовлетворяющее условию

$$|\bar{\bar{m}}_j^k| = \min |\bar{\bar{m}}_{j'}^k| \,/\, j' \in (N_1^{k-1} \cup N_2^{k-1}), \quad (8)$$

а если таких значений несколько, то фиксирует наименьшее из них. Если минимум в (8) равен нулю, то оп.5 передает управление оп.8, в противном случае - оп.6.
Оп.6 для значения индекса j, найденного оператором № 5, находит значение индекса i, удовлетворяющее условию

$$|B_{ij}| = \max_{i' \in \bar{\bar{m}}_j^k} |B_{i'j}|,$$

а если таких значений несколько, то фиксирует наименьшее из них. Оп.6 передает управление оп.7.
Оп.7 полагает

$$M_1^k = \{i\} \cup M_1^{k-1},$$
$$M_2^k = (\bar{\bar{m}}_j^k \setminus \{i\}) \cup M_2^{k-1},$$
$$N_1^k = \{j\} \cup N_1^{k-1},$$
$$N_2^k = N_2^{k-1}$$

и фиксирует упорядоченную тройку чисел $\tau = (i, j, k)$.
Здесь i и j - значения индексов, найденные, соответственно, оп. 6 и 5. Оп.7 передает управление оп.9.
Оп.8 полагает:

$$M_1^k = M_1^{k-1},$$
$$M_2^k = M_2^{k-1},$$
$$N_1^k = N_1^{k-1},$$
$$N_2^k = \{j\} \cup N_2^{k-1}$$

Здесь j - значение соответствующего индекса, найденное оп.5. Оп. 8 передает управление оп.9.

Оп.9, если $k = m$, заканчивает работу алгорифма. Если $k < m$, оп. 9 передает управление оп.5, полагая

$$k := k + 1.$$

После работы алгорифма приведения все компоненты векто -

ра y оказываются разбитыми на два непересекающихся множества M_1^m и M_2^m, а все уравнения системы (7) -на два непересекающихся множества N_1^m и N_2^m. Кроме того,остается информация о новом порядке расположения уравнений в системе и компонент в векторе оценок,записанная в виде упорядоченных троек чисел $\tau=(\tau_1, \tau_2, \tau_3)$. Множество всех этих троек обозначим через T.

При вычислении решения системы (7) для нахождения компонент вектора y, входящих в множество M_2^m, будет решена система общего вида порядка $|N_2^m|$, полученная из уравнений множества N_2^m путем исключения из них переменных, входящих в множество M_1^m с помощью уравнений из множества N_1^m. Вслед за этим компоненты вектора y, входящие в множество M_1^m, будут последовательно вычислены по рекуррентным соотношениям, полученным из уравнений множества N_1^m.

Заметим,что описанный выше алгорифм приведения не гарантирует минимальности $|N_2^m|$. Однако для матриц, которые с помощью перестановок строк и столбцов можно привести к треугольному виду, он дает оптимальное разбиение $(|N_2^m|=0)$. Кроме того,в тех случаях, когда для системы (7) может быть указано максимальное число составляющих её независимых подсистем[ж)], величину N_2^m можно оценить так:

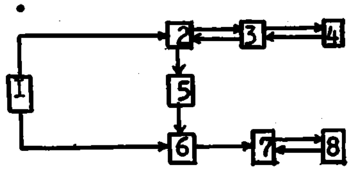
$$|N_2^m|<m+1-\frac{m-p}{\tau-1},$$

ж) Говорят,что система линейных алгебраических уравнений состоит из независимых подсистем,если матрицу системы с помощью перестановок в ней строк и столбцов можно привести к виду,в котором по главной диагонали будут стоять квадратные матрицы, а остальные элементы будут равны нулю. В частности, каждая система состоит из одной независимой подсистемы (из себя самой). Примером системы,состоящей из большого числа независимых подсистем, может служить система для определения потенциалов допустимого базисного решения обобщенной транспортной задачи [2], скелет которого распадается на несколько компонент связности.

где m - порядок системы (7), p - максимальное число независимых подсистем, составляющих систему (7), и r - наибольшее число отличных от нуля элементов матрицы A, стоящих в одном столбце.

Переходим к описанию алгорифма вычисления решения системы (7).

Обозначим через h переменную, принимающую целые значения от I до m.



Черт. 2.

На черт.2 представлена блок-схема алгорифма. Ниже следует описание отдельных операторов блок-схемы.

Оп.I переменной h присваивает значение "I" и полагает

$$N_2^{m-h+1} = N_2^{m}.$$

Если N_2^m пусто, то оп.I передает управление оп.6; в противном случае - оп.2.

Оп.2, если N_2^{m-h+1} пусто, передает управление оп. 5; в противном случае оп.2 находит $\tilde{j}$, удовлетворяющее условию:

$$\tilde{j} = \min j / j \in N_2^{m-h+1},$$

полагает:

$$N_2^{m-h} = N_2^{m-h+1} \setminus \{\tilde{j}\}$$

и передает управление оп.3.

Оп.3, если множество $m_{\tilde{j}} \cap M_1^m$ пусто, передает управление оп.2, полагая $h := h+1$. В противном случае оп.3 в множестве упорядоченных троек чисел T находит тройку $\tau = (\tau_1, \tau_2, \tau_3)$, удовлетворяющую условию:

$$\tau_3 = \max \tau_3' / \tau_1' \in m_{\tilde{j}} \cap M_1^m; \quad (\tau_1', \tau_2', \tau_3') = \tau'; \quad \tau' \in T,$$

и передает управление оп.4.

Оп.4 в уравнение $n_{\tilde{j}}$ ($\tilde{j}$ - значение индекса j , найденное оп. 2) вместо переменной $y_{\tilde{\iota}_1}$ ($\tilde{\iota}_1$ - значение индекса i , являющееся первой компонентой тройки $\tilde{\iota}$, найденной оп.3) подставляет ее выражение по формуле:

$$y_{\tilde{\iota}_1}=\frac{1}{b_{\tilde{\iota}_1\tilde{\iota}_2}}\left(c^*_{\tilde{\iota}_2}-\sum_{\substack{\ell=1\\ \ell\neq\tilde{\iota}_1}}^{m} b_{\ell\tilde{\iota}_2}\,y_\ell\right).$$

Здесь $\tilde{\iota}_2$ - значение индекса j , являющееся второй компонентой тройки $\tilde{\iota}$, найденной оп.3.

Пусть после этой подстановки уравнение $n_{\tilde{j}}$ имеет вид:

$$\sum_{i=1}^{m}\bar{b}_{i\tilde{j}}\,y_i=\bar{c}^*_{\tilde{j}} .$$

Оп. 4 полагает

$$m_{\tilde{j}}=\{i/\bar{b}_{i\tilde{j}}\neq 0\}$$

и передает управление оп.3.

Оп.5 работает,когда построена система уравнений

$$\sum_{i\in M_2^{mij}}\bar{b}_{ij}\,y_i=\bar{c}_j^*(j\in N_2^m).$$

Он решает эту систему любым из известных методов линейной алгебры и передает управление оп.6.

Оп.6 полагает

$$M_1^{m-h+1}=M_1^m$$

и передает управление оп.7.

Оп.7, если M_1^{m-h+1} - пусто,заканчивает работу алгорифма.Если M_1^{m-h+1} не пусто, то оп.7 в множестве упорядоченных троек чисел T находит тройку $\tilde{\iota}=(\tilde{\iota}_1,\tilde{\iota}_2,\tilde{\iota}_3)$, удовлетворяющую условию

$$\tilde{\iota}_3=\min \tilde{\iota}_3'/\tilde{\iota}_1'\in M_1^{m-h+1}; \qquad (\tilde{\iota}_1',\tilde{\iota}_2',\tilde{\iota}_3')=\tilde{\iota}',\ \tilde{\iota}'\in T,$$

полагает

$$M_1^{m-h}=M_1^{m-h+1}\setminus\{\tilde{\iota}_1\}$$

(здесь $\tilde{\iota}_1$ - значение индекса i ,являющееся первой компонентой найденной оператором 7 тройки) и передает управление оп. 8.

Оп.8 вычисляет переменную $y_{\tilde{\iota}_1}$ по формуле

$$y_{\tilde{\iota}_1} = \frac{1}{b_{\tilde{\iota}_1 \tilde{\iota}_2}} \left(c^*_{\tilde{\iota}_2} - \sum_{\substack{\ell=1 \\ \ell \neq \tilde{\iota}}}^{m} b_{\ell \tilde{\iota}_2} y_\ell \right).$$

Здесь $\tilde{\iota}_1$ и $\tilde{\iota}_2$ - значения индексов i и j, являющиеся, соответственно, первой и второй компонентами тройки $\tilde{\iota}$, найденной оператором 7.

Закончив вычисление, оп.8 передает управление оп.7, полагая $h := h+1$.

§ 3. Вычисление коэффициентов разложения

Как отмечалось в § I, матрица системы (6) для определения коэффициентов разложения получается из матрицы системы(4) транспонированием. Поэтому при вычислении коэффициентов разложения могут быть использованы результаты приведения матрицы системы (4).

Введем следующие обозначения в дополнение к обозначениям § I и 2:

$x^j_{s_0}$ - j - тая компонента вектора коэффициентов разложения X_{s_0} ;

$a^i_{s_0}$ - элемент матрицы A, стоящий на пересечении i -той строки и столбца A_{s_0} .

В этих и ранее принятых обозначениях система (6) запишется следующим образом:

$$\sum_{j=1}^{m} b_{ij} x^j_{s_0} = -a^i_{s_0}, \qquad i=1,2,\ldots,m. \tag{9}$$

Пусть

m_i - множество $\{j : B_{ij} \neq 0\}$;

n_i - i - тое уравнение системы (9) ;

h - переменная, принимающая целые значения от I до m .

Переходим к описанию алгорифма (см. черт. 2).

Оп.I присваивает переменной h значение "I" и полагает

$$M_2^{m-h+1} = M_2^m.$$

Если M_2^m пусто, то оп.I передает управление оп.6; в противном случае - оп.2.

Оп.2, если M_2^{m-h+1} пусто, передает управление оп. 5. Если M_2^{m-h+1} не пусто, то оп.2 находит $\hat{\iota}$, удов -

летворяющее условию

$$\tilde{i} = \min i' / i' \in M_2^{m-h+1},$$

полагает

$$M_2^{m-h} = M_2^{m-h+1} \setminus \{\tilde{i}\}$$

и передает управление оп.3.

Оп.3, если множество $m_{\tilde{\tau}} \cap N_i^m$ пусто, передает управление оп.2, полагая $h := h+1$. В противном случае оп.3 в множестве T находит тройку $\tilde{\tau} = (\tilde{\tau}_1, \tilde{\tau}_2, \tilde{\tau}_3)$, удовлетворяющую условию

$$\tilde{\tau}_3 = \min \tau_3' / \tau_2' \in N_1^m; \quad (\tau_1', \tau_2', \tau_3') = \tau'; \quad \tau' \in T,$$

и передает управление оп.4.

Оп.4 в уравнении $n_{\tilde{i}}$ ($\tilde{i}$ - значение индекса i, найденное оператором 2) вместо переменной $x_{s_0}^{\tilde{\tau}_2}$ ($\tilde{\tau}_2$ - значение индекса i, являющееся второй компонентой тройки $\tilde{\tau}$, найденной оператором 3) подставляет ее значение, выражаемое формулой:

$$x_{s_0}^{\tilde{\tau}_2} = \frac{1}{b_{\tilde{\tau}_1 \tilde{\tau}_2}} \left(-a_{s_0}^{\tilde{\tau}_1} - \sum_{\substack{e=1 \\ e \neq \tilde{\tau}_2}}^{m} b_{\tilde{\tau}_1 e} x_{s_0}^{e} \right).$$

Здесь $\tilde{\tau}_1$ - значение индекса i, являющееся первой компонентой тройки $\tilde{\tau}$, найденной оператором 3.

Пусть после этой подстановки уравнение $n_{\tilde{i}}$ имеет вид:

$$\sum_{j=1}^{m} \bar{b}_{\tilde{i} j} x_{s_0}^{j} = -\bar{a}_{s_0}^{\tilde{i}}.$$

Оп.4 полагает

$$m_{\tilde{i}} = \{ j / \bar{b}_{\tilde{i} j} \neq 0 \}$$

и передает управление оп.3.

Оп.5 работает, когда построена система уравнений:

$$\sum_{j \in N_2^m} \bar{b}_{ij} x_{s_0}^{j} = -\bar{a}_{s_0}^{i} \quad (i \in M_2^m).$$

Он решает эту систему любым из известных методов линейной алгебры и передает управление оп.6.

Оп.6 полагает

$$N_1^{m-h+1} = N_1^m$$

и передает управление оп.7.

Оп.7, если N_1^{m-h+1} пусто, заканчивает работу алгорифма. В противном случае оп.7 находит в множестве T тройку $\tau=(\tau_1, \tau_2, \tau_3)$, удовлетворяющую условию

$$\tau_3 = \max \tau_3' / \tau_2' \in N_1^{m-h+1}; \quad (\tau_1', \tau_2', \tau_3')=\tau'; \quad \tau' \in T_3,$$

полагает

$$N_1^{m-h} = N_1^{m-h+1} \setminus \{\tau_2\}.$$

(здесь τ_2 - значение индекса j, являющееся второй компонентой тройки τ, найденной оп.7) и передает управление оп.8.

Оп.8 вычисляет переменную $x_{s_0}^{\tau_2}$ по формуле:

$$x_{s_0}^{\tau_2} = \frac{1}{b_{\tau_1 \tau_2}} \Big(-a_{s_0}^{\tau_1} - \sum_{\substack{\ell=1 \\ \ell \neq \tau_2}}^{m} b_{\tau_1 \ell} x_{s_0}^{\ell}\Big).$$

Здесь τ_1 и τ_2 - значения индексов i и j, являющиеся, соответственно, первой и второй компонентами тройки τ, найденной оп.7. Закончив вычисление, оп.8 передает управление оп. 7, полагая $h := h+1$.

Описанная выше схема решения систем линейных алгебраических уравнений в настоящее время испытывается на ЭВМ. Но уже сейчас можно сказать, что далеко не каждая система легко решается по этой схеме, ибо она является одним из вариантов метода исключения. Однако в ряде задач экономического содержания, в которых уже известные схемы алгорифмов решения задач линейного программирования не достигают цели (главным образом, из-за ограниченности внутреннего запоминающего устройства машины), применение описанной схемы не встречает затруднений.

Литература

1. Г.Ш. Рубинштейн. О развитии и применениях линейного программирования в СССР. - Линейные неравенства и смежные вопросы, М, ИЛ, 1959.
2. М.К. Гавурин, Г.Ш.Рубинштейн, С.С.Сурин. Об использовании производственных средств при выполнении нескольких видов работ (Обобщенная транспортная задача) .- Сибмат. журнал, 1963, т.3, № 4.

Поступила в редакцию
20.III.1967 г.