

ОБ ИСПОЛЬЗОВАНИИ СПЕЦИФИКИ УСЛОВИЙ ЗАДАЧИ В МЕТОДЕ УЛУЧШЕНИЯ ПЛАНА

К. В. КИМ

Статья посвящена обобщению метода улучшения плана, позволяющему строить экономные алгоритмы для ряда специальных задач линейного программирования. Примером такой специальной задачи может служить следующая транспортная модель.

Имеется некоторая сеть, содержащая N узлов и M дуг. Рассматриваются перевозки L различных продуктов. Заданы:

c_{ijk} — стоимость перевозки k -го продукта по дуге (ij) ;

a_{ik} — значение производства (или потребления) в i -м узле для k -го продукта;

h_{ij} — пропускная способность дуги (ij) .

Обозначим через x_{ijk} объем перевозки k -го продукта по дуге (ij) . Тре-

буется найти $\min \sum c_{ijk}x_{ijk}$ при

$$\left. \begin{aligned} \sum_j x_{ijk} - \sum_s x_{sik} &= a_{ik}, \\ x_{ijk} &\geq 0, \end{aligned} \right\} \quad (1)$$

$$\sum_k x_{ijk} \leq h_{ij}. \quad (2)$$

Рекомендации, содержащиеся в работе, могут быть использованы в любом случае, когда ограничения задачи представимы в виде двух групп ограничений, причем, если учитывать ограничения только первой группы, то решение задачи могло бы сильно упроститься за счет их специфики. [В нашем примере соотношения (1) составляют первую группу ограничений, соотношения (2) — вторую группу]. Сущность предлагаемого приема состоит в том, чтобы по возможности использовать и сохранять простую структуру ограничений первой группы на каждом шаге. Этот прием в некоторой степени является новым подходом к анализу блочных задач, для которых один из методов решений был указан Данцигом и Вульфом в [1].

I. Будем рассматривать задачу: найти $\max CX$ при $AX = P$, $X \geq 0$, где $C = \|c_1, c_2, \dots, c_n\|$,

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Алгоритм последовательного улучшения плана можно представить в следующем виде. Пусть

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \quad Z_s = \begin{pmatrix} z_{1s} \\ \vdots \\ z_{ms} \end{pmatrix}, \quad A_s = \begin{pmatrix} a_{1s} \\ \vdots \\ a_{ms} \end{pmatrix},$$

$$U = \|u_1, u_2, \dots, u_m\|, \quad L = \|c_{j_1}, c_{j_2}, \dots, c_{j_m}\|.$$

Предполагается, что задано подмножество индексов $T = \{j_1, j_2, \dots, j_m\}$, такое, что матрица

$$B = \|A_{j_1}, A_{j_2}, \dots, A_{j_m}\|$$

невыврожденная, а также $BY = P$, причем $Y \geq 0$. Тогда шаг алгоритма состоит из следующих операций:

1. Находим вектор U из условия $UB = L$.
2. Если $UA_j - c_j \geq 0$ для всех $j = 1, 2, \dots, n$, то задача решена. Если же существует s такое, что $UA_s - c_s < 0$, то переходим к следующему пункту.
3. Находим Z_s из условия $BZ_s = A_s$.
4. Если $z_{is} < 0$ для всех $i = 1, 2, \dots, m$, то задача не имеет решения. В противном случае находим k такое, что

$$\theta = \min_{z_{is} > 0} \frac{y_i}{z_{is}} = \frac{y_k}{z_{ks}}.$$

5. В подмножестве J заменим значение j_h значением s . В результате получается новая матрица $\tilde{B}$ и новый вектор $\tilde{Y}$, причем вектор $\tilde{Y}$ находим по формулам

$$\tilde{y}_i = y_i - \theta z_{is},$$

$$\tilde{y}_h = \theta.$$

Таким образом, на каждом шаге нам приходится решать две системы уравнений:

$$BZ_s = A_s \quad \text{и} \quad UB = L. \quad (3)$$

Практически эти системы решают на каждом шаге только в случае, если матрица A чрезвычайно проста (например, в транспортной задаче). В большинстве случаев хранят матрицу B^{-1} . Поскольку на каждом шаге в матрице B меняется только один столбец, матрица B^{-1} меняется тоже незначительно, а именно $\tilde{B}^{-1} = \Phi B^{-1}$, где

$$\Phi = \begin{pmatrix} 1 & 0 & \dots & \varphi_{1k} & \dots & 0 \\ 0 & 1 & \dots & \varphi_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \varphi_{mk} & \dots & 0 \end{pmatrix},$$

$$\varphi_{ik} = -\frac{z_{is}}{z_{ks}} \quad \text{для } i \neq k,$$

$$\varphi_{kk} = \frac{1}{z_{ks}}.$$

В некоторых случаях матрицу B^{-1} хранят не явно, а в виде совокупности ее элементарных множителей. Если на некотором шаге

$$B^{-1} = \Phi^{(t)} \Phi^{(t-1)} \dots \Phi^{(1)} E,$$

то решение системы (3) сводится к последовательному умножению вектора на элементарные матрицы

$$Z_s = \Phi^{(1)}(\dots \Phi^{(2)}(\Phi^{(1)}A_s) \dots), \\ U = (\dots (L\Phi^{(1)}) \dots \Phi^{(2)})\Phi^{(1)}.$$

II. Нас будут интересовать задачи, в которых матрицу A по каким-либо причинам удобно представить в виде

$$A = \left\| \frac{A'}{A''} \right\|.$$

Пусть матрица A' содержит l строк. Условимся сохранять для всех клеток матрицы A сквозную нумерацию. Отсюда строки матрицы A'' пронумерованы от $l+1$ до m .

Разобьем матрицу B на клетки соответствующего размера

$$B = \left\| \frac{D}{r} \middle| \frac{d}{Q} \right\|$$

так, чтобы клетка D была невырожденной. Соответственно разбиваются все векторы:

$$P = \left\| \frac{P'}{P''} \right\|, Z = \left\| \frac{Z'}{Z''} \right\|, Y = \left\| \frac{Y'}{Y''} \right\|, A_s = \left\| \frac{A'_s}{A''_s} \right\|, \\ U = \|U', U''\|, L = \|L', L''\|.$$

Рассмотрим систему $BZ_s = A_s$. Ее можно представить в виде

$$\left\| \frac{D}{r} \middle| \frac{d}{Q} \right\| \times \left\| \frac{Z'_s}{Z''_s} \right\| = \left\| \frac{A'_s}{A''_s} \right\|.$$

Умножим обе части равенства слева на матрицу

$$\left\| \frac{D^{-1}}{-rD^{-1}} \middle| \frac{0}{E} \right\|.$$

Получим

$$\left\| \frac{E}{0} \middle| \frac{D^{-1}d}{R} \right\| \times \left\| \frac{Z'_s}{Z''_s} \right\| = \left\| \frac{D^{-1}A'_s}{A''_s - rD^{-1}A'_s} \right\|,$$

где $R = Q - rD^{-1}d$. Легко видеть, что клетка R невырожденная. Приравняв нижние части векторов полученного равенства, имеем

$$RZ''_s = A''_s - rD^{-1}A'_s.$$

Верхняя часть равенства сводится к

$$DZ'_s = A'_s - dZ''_s.$$

Отсюда

$$Z''_s = R^{-1}(A''_s - rD^{-1}A'_s), \\ Z'_s = D^{-1}(A'_s - dZ''_s). \quad (4)$$

Аналогично может быть выражено решение системы $UB = L$. Умножим обе части равенства

$$\|U', U''\| \times \left\| \frac{D}{r} \middle| \frac{d}{Q} \right\| = \|L', L''\|$$

справа на матрицу

$$\left\| \begin{array}{c|c} D^{-1} & -D^{-1}d \\ \hline 0 & E \end{array} \right\|.$$

Получим

$$\|U', U''\| \times \left\| \begin{array}{c|c} E & 0 \\ \hline rD^{-1} & R \end{array} \right\| = \|L'D^{-1}, L'' - L'D^{-1}d\|.$$

Отсюда

$$\begin{aligned} U'' &= (L'' - L'D^{-1}d)R^{-1}, \\ U' &= (L' - U''r)D^{-1}. \end{aligned} \quad (5)$$

Таким образом, если известны матрицы D^{-1} и R^{-1} , то нетрудно найти решение систем $UB = L$ и $BZ_s = A_s$.

Рассмотрим вопрос о том, как изменяются матрицы D^{-1} и R^{-1} при выполнении одного шага. Легко проверить, что

$$B^{-1} = \left\| \begin{array}{c|c} D^{-1} + D^{-1}dR^{-1}rD^{-1} & -D^{-1}dR^{-1} \\ \hline -R^{-1}rD^{-1} & R^{-1} \end{array} \right\|.$$

Обозначим $-D^{-1}d$ через H , $D^{-1} + D^{-1}dR^{-1}rD^{-1}$ через F , $-R^{-1}rD^{-1}$ через S . Тогда более коротко:

$$B^{-1} = \left\| \begin{array}{c|c} F & HR^{-1} \\ \hline S & R^{-1} \end{array} \right\|.$$

Известно, что $\tilde{B}^{-1} = \Phi B^{-1}$. Разобьем матрицу Φ на соответствующие клетки:

$$\Phi = \left\| \begin{array}{c|c} \Phi_1 & \Phi_2 \\ \hline \Phi_3 & \Phi_4 \end{array} \right\|.$$

Тогда

$$\tilde{B}^{-1} = \left\| \begin{array}{c|c} \Phi_1 F + \Phi_2 S & (\Phi_1 H + \Phi_2) R^{-1} \\ \hline \Phi_3 F + \Phi_4 S & (\Phi_3 H + \Phi_4) R^{-1} \end{array} \right\|.$$

Можно показать, что матрицы D^{-1} и R^{-1} на каждом шаге алгоритма улучшения плана подвергаются достаточно простым преобразованиям.

1. Пусть $k > l$. Тогда матрица D не изменяется, а матрица Φ_3 нулевая. Матрица R^{-1} изменяется по формуле

$$\tilde{R}^{-1} = \left\| \begin{array}{cccccc} 1 & 0 & \dots & \varphi_{l+1,k} & \dots & 0 \\ 0 & 1 & \dots & \varphi_{l+2,k} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \varphi_{m,k} & \dots & 1 \end{array} \right\| \times R^{-1}.$$

2. Пусть $k \leq l$. В этом случае матрица D изменяется, а именно, k -й столбец заменяется на A'_s . Пусть

$$D^{-1} A'_s = \left\| \begin{array}{c} \alpha_{1s} \\ \alpha_{2s} \\ \vdots \\ \alpha_{ls} \end{array} \right\|.$$

Если $\alpha_{ks} \neq 0$, то матрица $\tilde{D}$ остается невырожденной, и тогда

$$\tilde{D}^{-1} = \begin{vmatrix} 1 & 0 & \dots & \beta_{1k} & \dots & 0 \\ 0 & 1 & \dots & \beta_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \beta_{lk} & \dots & 1 \end{vmatrix} \times D^{-1},$$

где

$$\beta_{ik} = -\frac{\alpha_{is}}{\alpha_{ks}} \quad \text{для } i \neq k,$$

$$\beta_{kk} = \frac{1}{\alpha_{ks}}.$$

Матрица $\tilde{R}$ также невырожденная, и

$$\tilde{R}^{-1} = (\Phi_3 H + \Phi_4) R^{-1}.$$

Заметим, что $\Phi_4 = E$, а Φ_3 содержит лишь один отличный от нуля столбец. Поэтому можно записать

$$\tilde{R}^{-1} = (\Phi_3 H^{(k)} + E) R^{-1},$$

где $H^{(k)}$ содержит лишь одну отличную от нуля k -ю строку, а именно:

$$(\alpha_{kj_{l+1}}, \alpha_{kj_{l+2}}, \dots, \alpha_{kj_m}).$$

3. Несколько сложнее тот случай, когда $k \leq l$ и $\alpha_{ks} = 0$. В этом случае при замене k -го столбца на A_s' матрица D становится вырожденной. Однако тогда существует $t > l$ такое, что $\alpha_{kj_t} \neq 0$. Действительно, предположим, что для всех $l < t \leq m$ $\alpha_{kj_t} = 0$. Из формул (4) следует, что

$$z_{ks} = \alpha_{ks} - \sum_{t=l+1}^m \alpha_{kj_t} z_{ts} = 0,$$

но $z_{ks} \neq 0$!

Итак, пусть $\alpha_{kj_t} \neq 0$. Поменяем в матрице $\tilde{B}$ местами столбцы A_{j_k} и A_{j_t} . Тогда матрица $\tilde{D}$ получится путем замены в матрице D k -го столбца на A_{j_t}' , и так как $\alpha_{kj_t} \neq 0$, $\tilde{D}$ станет невырожденной.

Получим

$$\tilde{D}^{-1} = \begin{vmatrix} 1 & 0 & \dots & \beta_{1k} & \dots & 0 \\ 0 & 1 & \dots & \beta_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \beta_{lk} & \dots & 1 \end{vmatrix} \times D^{-1},$$

где

$$\beta_{ik} = -\frac{\alpha_{ij_t}}{\alpha_{kj_t}} \quad \text{для } i \neq k,$$

$$\beta_{kk} = \frac{1}{\alpha_{kj_t}}.$$

В матрице $\tilde{B}^{-1}$ надо поменять местами k -ю и t -ю строки. В результате этого в матрице $\tilde{R}^{-1}$ t -я строка заменяется на k -ю строку матрицы

$(\Phi_1 H + \Phi_2) R^{-1}$. Так как $\Phi_2 = 0$, а k -я строка матрицы Φ_1 представляет собой

$$\| 0, 0, \dots, \varphi_{kk}, \dots, 0 \|,$$

то искомая строка находится по формуле

$$\varphi_{kk} \times \| \alpha_{kj_{l+1}}, \alpha_{kj_{l+2}}, \dots, \alpha_{kj_n} \| \times R^{-1}$$

Учитывая эту замену, можно представить преобразование матрицы R^{-1} в виде

$$\tilde{R}^{-1} = (\Phi_3' H^{(k)} + E^{(t)}) R^{-1},$$

где

$$\Phi_3' = \begin{pmatrix} \varphi_{l+1, k} \\ \vdots \\ \varphi_{l-1, k} \\ \varphi_{k, k} \\ \varphi_{l+1, k} \\ \vdots \\ \varphi_{m, k} \end{pmatrix}, \quad H^{(k)} = \| \alpha_{kj_{l+1}}, \alpha_{kj_{l+2}}, \dots, \alpha_{kj_m} \|,$$

а $E^{(t)}$ — матрица, совпадающая с единичной, кроме t -го столбца; t -й столбец нулевой. (Напоминаем, что нумерация клеток сквозная, т. е. в матрице $E^{(t)}$ строки и столбцы пронумерованы от $l+1$ до m).

Таким образом, матрица D^{-1} подвергается элементарным преобразованиям. Преобразование матрицы R^{-1} немного сложнее — если матрица D^{-1} изменяется в один просмотр, то R^{-1} изменяется в два просмотра. Заметим, что в некоторых случаях, когда матрица A' очень проста, системы (4) и (5) лучше решать непосредственно, и матрица D^{-1} в этих случаях не нужна.

III. Рассмотренная модификация может быть применена и в простейшей форме метода последовательного улучшения плана, когда не хранят матрицу B^{-1} . Известно, что простейший алгоритм улучшения плана сводится к последовательному преобразованию матрицы

$$\left\| \begin{array}{c|c} B^{-1}A & B^{-1}P \\ \hline LB^{-1}A - C & \end{array} \right\|.$$

Мы будем рассматривать матрицу

$$\left\| \begin{array}{c|c} D^{-1}A & B^{-1}P \\ \hline \frac{R^{-1}(A' - rD^{-1}A')}{LB^{-1}A - C} & \end{array} \right\|.$$

Запишем ее в виде

$$\left\| \begin{array}{ccc|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} & y_1 \\ \dots & & & & \vdots \\ \alpha_{l1} & \alpha_{l2} & & \alpha_{ln} & \\ \hline z_{l+1,1} & z_{l+1,2} & \dots & z_{l+1,n} & \\ \dots & \dots & \dots & \dots & \\ z_{m,1} & z_{m,2} & & z_{m,n} & y_m \\ \hline z_{m+1,1} & z_{m+1,2} & & z_{m+1,n} & \end{array} \right\|.$$

Тогда модифицированный алгоритм сведется к следующим операциям:

1. Если $z_{m+1,j} \geq 0$ для всех $j = 1, 2, \dots, m$, то задача решена. В противном случае находим s такое, что $z_{m+1,j} < 0$.

2. Находим z_{is} для $i = 1, 2, \dots, l$ по формуле

$$z_{is} = \alpha_{is} - \sum_{t=l+1}^m \alpha_{ij_t} z_{ts}.$$

3. Если $z_{is} \leq 0$ для всех $i = 1, 2, \dots, m$, то задача не имеет решения. В противном случае находим такое k , что

$$\theta = \min_{z_{is} > 0} \frac{y_i}{z_{is}} = \frac{y_k}{z_{ks}}$$

4. Изменяем Y по формуле

$$\begin{aligned} \tilde{y}_i &= y_i - \theta z_{is} \quad \text{для } i \neq k, \\ \tilde{y}_k &= 0. \end{aligned}$$

5. Если $k \leq l$, то переходим к пункту 6. Если $k > l$, то элементы α_{ij} не изменяются, а элементы z_{ij} меняются по формулам

$$\begin{aligned} \tilde{z}_{ij} &= z_{ij} - \frac{z_{kj} z_{is}}{z_{ks}} \quad \text{для } i \neq k, \quad i > l, \\ \tilde{z}_{kj} &= \frac{z_{kj}}{z_{ks}}. \end{aligned}$$

В подмножестве J значение j_k заменяем значением s и переходим снова к пункту 1.

6. Находим z_{hf} для $f = 1, \dots, n$ по формуле

$$z_{hf} = \alpha_{hf} - \sum_{t=l+1}^m \alpha_{hj_t} z_{tf}.$$

7. Если $\alpha_{hs} = 0$, то переходим к пункту 8. Если же $\alpha_{hs} \neq 0$, то элементы матрицы изменяются по формулам

$$\begin{aligned} \tilde{\alpha}_{ij} &= \alpha_{ij} - \frac{\alpha_{kj} \alpha_{is}}{\alpha_{ks}} \quad \text{для } i \neq k, \quad 1 \leq i \leq l, \\ \tilde{\alpha}_{kj} &= \frac{\alpha_{kj}}{\alpha_{ks}}, \\ \tilde{z}_{ij} &= z_{ij} - \frac{z_{kj} z_{is}}{z_{ks}} \quad \text{для } l < i \leq m+1. \end{aligned}$$

В подмножестве J значение j_k заменяем значением s и переходим к пункту 1.

8. Находим такое q , что α_{hjq} . Тогда в подмножестве J значение j_k заменяется значением j_q , а j_q принимает значение s . Матрица изменяется по формулам

$$\begin{aligned} \tilde{\alpha}_{ij} &= \alpha_{ij} - \frac{\alpha_{kj} - \alpha_{ijq}}{\alpha_{kj_q}} \quad \text{для } i \neq k, \quad 1 \leq i \leq l, \\ \tilde{\alpha}_{kj} &= \frac{\alpha_{kj}}{\alpha_{kj_q}}, \end{aligned}$$

$$\tilde{z}_{ij} = z_{ij} - \frac{z_{kj}z_{is}}{z_{ks}} \quad \text{для } l < i \leq m+1, \quad i \neq q,$$

$$\tilde{z}_{qj} = \frac{z_{kj}}{z_{ks}}.$$

Затем переходим к пункту 1.

Применение этого алгоритма может стать выгодным при решении блочных задач, для которых матрица A' распадается на блоки, т. е. имеет вид:

$$A' = \left\| \begin{array}{c|c|c|c|c} A_1' & & & & 0 \\ \hline & A_2' & & & \\ \hline & & \ddots & & \\ \hline & & & A_M' & \\ \hline 0 & & & & \end{array} \right\|.$$

В этом случае хотя увеличивается число просмотров матрицы за счет пункта 6 при $k \leq l$, зато преобразование матрицы $D^{-1}A'$ сводится к преобразованию одного блока, так как матрица эта сохраняет блочную структуру.

ЛИТЕРАТУРА

1. G. B. Dantzig and P. Wolfe. Decomposition principle for Linear Programms. Opns. Res., 1960, vol. 8, No. 1, p. 101—111.

Поступила в редакцию
15 VII 1964