УДК 512.25

АЛГОРИТМЫ С УПОРЯДОЧЕННЫМ БАЗИСОМ ДЛЯ ДВУХКОМПОНЕНТНОЙ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В.А.Булавский, М.А.Яковлева

При решении двухкомпонентных задач линейного программирования на ЭВМ приобретает большое значение рациональная организация массивов. Не последнюю роль при этом играет способ хранения информации о базисном графе. Этот вопрос рассматривается в ряде работ [1–5]. Настоящая статья посвящена дальнейшему развитию методов, в которых поддерживается специальное упорядочение информации о базисном графе. Алгоритмы строятся в рамках метода последовательного улучшения, который считается известным читателю.

Пусть заданы множества $M=\{1,2,\dots,m\}$ и $N=\{1,2,\dots,n\}$, строка $c[N]$, столбец $b[M]$, а также матрица $A[M,N]$, у которой каждый столбец $A[M,j]$ при $j\in N$ имеет не более двух отличных от нуля компонент. Мы будем рассматривать задачу линейного программирования в следующей стандартной форме: минимизировать величину $c[N]\cdot x[N]$ при ограничениях

$$A[M,N]\cdot x[N]=b[M],$$

$$x[N]\geqslant 0.$$

Каждой квадратной неособенной подматрице $A[M,J]$, $J\subset N$, полной матрицы $A[M,N]$ сопоставим граф, называемый б а з и с н ы м. Множеством вершин этого графа служит множество $M\cup\{0\}$, а множеством ребер – множество J. При этом будем считать, что ребро $j \in J$ соединяет вершины, являющиеся индексами ненулевых элементов столбца $A[M, j]$. При наличии в столбце лишь одной ненулевой компоненты условимся, что ребро соединяет соответствующую вершину с вершиной 0. Известно [1,2], что построенный таким способом базисный граф двухкомпонентной задачи в общем случае состоит из $\rho+1$ компоненты связности, причем одна из них является деревом с корнем в вершине 0, а каждая из остальных включает в себя замкнутый контур с отходящими от него ответвлениями в виде деревьев. Возможные структуры компонент связности изображены на рис. I При этом ориентация ребер базисного графа осуществляется по описанным ниже правилам.

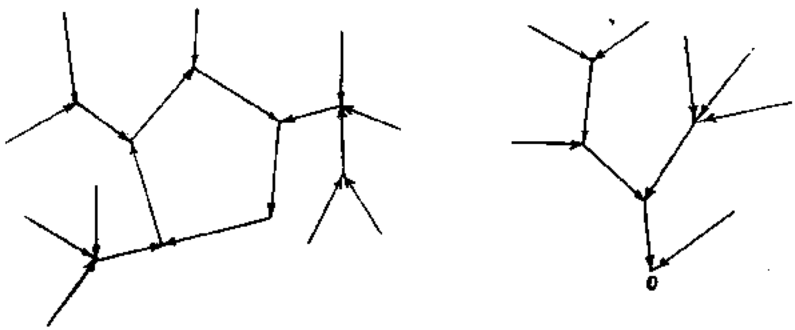


Рис. I

Для каждой вершины i, отличной от вершины 0 и не лежащей на замкнутом контуре, имеется единственный путь (без повторного прохождения вершин и ребер)

$$p_{t+1} \xleftarrow{q_t} p_t \cdots \xleftarrow{q_1} p_1 \xleftarrow{q_0} p_0 = i, \qquad \text{(I)}$$

ведущий либо в вершину 0, либо в вершину некоторого замкнутого контура. Тем самым для ребра j, не лежащего на замкнутом контуре, однозначно определяется вершина $i_в[j]$, путь из которой начинается ребром j. Эту вершину мы будем называть **в е р х н е й** для ребра j. Вторую же вершину ребра j будем называть **н и ж н е й** и обозначать символом $i_н[j]$. Ориентация ребра j выбрана в направлении от его верхней вершины к нижней.

На каждом из контуров выберем одно из двух возможных направлений его обхода, и ориентируем ребра в соответствии с этим направлением. При этом начальную и конечную вершину ориентированного ребра j назовем соответственно его верхней и нижней вершинами, сохранив для них символы $i_в[j]$ и $i_н[j]$. Замкнутый контур с выбранным направлением обхода будем называть **циклом**.

Для удобства обозначений при каждом $i \in M$ через $j[i]$ будем обозначать ребро, для которого вершина i является верхней. Это обозначение корректно, так как каждая вершина является верхней лишь для одного ребра. Ниже приводится описание трех алгоритмов решения поставленной двухкомпонентной задачи, различия в которых обусловлены хранением разной информации о базисном графе.

В первом из этих алгоритмов элементы множеств M и J упорядочиваются таким образом, чтобы нужные на каждом шаге метода решения систем

$$y[M]\cdot A[M,J]=c[J], \tag{2}$$

$$A[M,J]\cdot g[J]=A[M,j'], \tag{3}$$

$j'\in N\setminus J$ можно было найти, грубо говоря, немногим более, чем за один просмотр хранимой информации. Двойного просмотра при этом требует лишь информация о циклах. В этом алгоритме не учитывается, что правая часть системы (3) имеет не более двух отличных от нуля компонент. Компенсация за лишнюю работу состоит в простоте алгоритма, а также в том, что можно не хранить столбцы $x[J]$ и $g[J]$: подправка столбца $x[J]$ тогда просто не требуется, а определение номера $j''\in J$, выбывающего из числа базисных, можно осуществить при параллельном решении системы (3) и системы

$$A[M,J]\cdot x[J]=b[M]$$

с той же матрицей.

Во втором алгоритме учитывается специфика правой части системы (3). Это достигается благодаря хранению дополнительной информации, позволяющей для любого $i \in M$ экономно находить путь (I).

В третьем алгоритме вводятся более жесткие требования на упорядочение информации, что дает возможность довычислять лишь ту часть компонент строки $y[M]$, которая фактически изменяется при переходе к новому базисному множеству $\bar{J} = (J \setminus \{j''\}) \cup \{j'\}$.

Выберем на каждом цикле по одной произвольной вершине и будем называть эти вершины **н а ч а л ь н ы м и** для соответствующих циклов, обозначая их через $i_1^*, i_2^*, \ldots, i_p^*$. Пусть $Q = \{0, i_1^*, i_2^*, \ldots, i_p^*\}$. Упорядочение вершин

$$M \cup \{0\} = \{0, i_1, i_2, \ldots, i_m\} \tag{4}$$

будем называть **п р а в и л ь н ы м**, если для каждого $k = 1, 2, \ldots, m$ либо $i_k \in Q$, либо $i_H[j[i_k]] \in \{0, i_1, i_2, \ldots, i_{k-1}\}$.

Алгоритм с простым упорядочением

Потребуем дополнительно, чтобы вершины каждого цикла были расположены подряд в порядке, противоположном выбранному на цикле направлению обхода. Имея правильное упорядочение вершин (4), мы можем однозначно перейти к упорядочению ребер

$$J = \{j_1, j_2, \ldots, j_m\}, \qquad j_k = j[i_k], \tag{5}$$

которое также будем называть правильным. Очевидно, что ребра циклов окажутся при этом расположенными подряд.

В рассматриваемом алгоритме считается заданным правильное упорядочение (5) ребер базисного графа, причем для каждого ребра $j \in J$ указано, какая из двух его вершин является верхней, т.е. нам известны вершины $i_B[j]$ и $i_H[j]$. Кроме того, должны быть отмечены начальные и конечные ребра каждого цикла. Задание списка (5) можно осуществить следующим образом: для каждого ребра $j \in J$ задаются ребро $j_+[j]$, следующее в списке (5) за ребром j, и ребро $j_-[j]$, предшествующее ему. Для опознания начала и конца списка (5) дополнительно положим $j_-[j_1] = 0$, $j_+[0] = j_1$, $j_+[j_m] = 0$, $j_-[0] = j_m$.

При решении системы (2) примем $y[0] = 0$, $j_0 = 0$ и, последовательно для каждого $k = 1, 2, \ldots, m$, определим ребро $j_k = j_+[j_{k-1}]$, проверив, не является ли j_k начальным ребром некоторого цикла. Если это не так, то найдем

$$y[i_в[j_k]] = \frac{c[j_k] - y[i_н[j_k]] \cdot A[i_н[j_k], j_k]}{A[i_в[j_k], j_k]} . \tag{6}$$

Вершина $i_в[j_k]$ никогда не совпадает с вершиной O, в то же время может оказаться, что $i_н[j_k] = O$. Так как $y[O] = O$, то значение $A[O, j_k]$ безразлично. Можно, например, положить $A[O, j] = O$ при $j \in J$.

В случае, когда j_k является начальным ребром некоторого цикла, состоящего из ребер $j_k, j_{k+1}, \dots, j_{k+q}$, величина $y[i_н[j_k]]$ к моменту выхода на ребро j_k нам не известна. Поэтому определение $y[i_в[j_k]]$ является нестандартным. Если бы мы задали значение $y[i_в[j_k]]$ произвольно и применили последовательно формулу (6) для ребер $j_{k+1}, j_{k+2}, \dots, j_{k+q}, j_k$, то получили бы новое значение $\tilde{y}[i_в[j_k]] = \alpha + \rho \cdot y[i_в[j_k]]$. Потребовав выполнения равенства $\tilde{y}[i_в[j_k]] = y[i_в[j_k]]$, найдем

$$y[i_в[j_k]] = \frac{\alpha}{1-\rho} . \tag{7}$$

Заметим, что $\tilde{y}[i_в[j_k]] = \alpha$, если $y[i_в[j_k]] = O$. Кроме того, нетрудно проверить, что

$$\rho = \prod_{k=0}^{q} \left(- \frac{A[i_н[j_{k+r}], j_{k+r}]}{A[i_в[j_{k+r}], j_{k+r}]} \right) .$$

Таким образом, если положить $y[i_в[j_k]] = O$ и, просмотрев все ребра цикла, вычислить коэффициенты α и ρ, то истинное значение $y[i_в[j_k]]$ получается по формуле (7). После того, как это значение определено, продолжается стандартный просмотр списка (5), начиная с ребра $j_{k+1} = j_+[j_k]$, и вычисления проводятся по формуле (6) независимо от того, принадлежит ли исследуемое ребро циклу или нет.

Аналогично решаются системы вида $A[M, J] \cdot x[J] = F[M]$, но список (5) просматривается в обратном порядке. Именно, положим $j_{m+1} = O$ и для $k = m, m-1, \dots, 1$ определим ребро $j_k = j_-[j_{k+1}]$. Если это ребро не является последним в некотором цикле, то, в силу правильности упорядочения (5), $A[i_в[j_k], j_k] \neq O$ и $A[i_в[j_k], j_\nu] = O$ при $\nu = k-1, k-2, \dots, 1$. Поэтому, положив

$N_k = \{ \nu \geq k+1 \mid A[i_в[j_k], j_\nu] \neq 0 \}$, получаем

$$x[j_k] = \frac{F[i_в[j_k]] - \sum_{\nu \in N_k} A[i_в[j_k], j_\nu] \cdot x[j_\nu]}{A[i_в[j_k], j_k]}. \qquad (8)$$

Фактически числитель в последней формуле можно вычислять не в момент ее применения, а последовательно путем подправки правой части $F[M]$ по мере вычисления $x[j_\nu]$, $\nu = m, m-1, \ldots, k+1$.

Если ребро j_k является последним ребром некоторого цикла $j_{k-q}, j_{k-q+1}, \ldots, j_k$, то $i_н[j_{k-q}] = i_в[j_k]$, так что $A[i_в[j_k], j_{k-q}] \neq 0$, и формулу (8) применить нельзя: для этого нужно было бы знать $x[j_{k-q}]$. Заметим, однако, что приняв $x[j_k]$ за параметр и применив формулу (8) последовательно для определения $x[j_{k-1}]$, $x[j_{k-2}]$, ..., $x[j_{k-q}]$, мы получим зависимость вида

$$x[j_{k-q}] = \frac{\beta - A[i_в[j_k], j_k] \cdot \rho \cdot x[j_k]}{A[i_н[j_{k-q}], j_{k-q}]},$$

где

$$\rho = \prod_{r=0}^{q} \left(- \frac{A[i_н[j_{k-r}], j_{k-r}]}{A[i_в[j_{k-r}], j_{k-r}]} \right).$$

Таким образом, положив временно $x[j_k] = 0$, за один предварительный просмотр ребер встреченного цикла можно найти число β и одновременно подсчитать ρ. Истинное значение величины $x[j_k]$ можно найти из того условия, что

$$A[i_в[j_k], j_{k-q}] \cdot x[j_{k-q}] + A[i_в[j_k], j_k] \cdot x[j_k] +$$
$$+ \sum_{\nu \in N_k} A[i_в[j_k, j_\nu], j_\nu] \cdot x[j_\nu] = F[i_в[j_k]].$$

Напомним, что $i_н[j_{k-q}] = i_в[j_k]$. Подставив вместо $x[j_{k-q}]$ полученное выше его выражение через β, ρ и $x[j_k]$, будем иметь

$$x[j_k] = \frac{F[i_в[j_k]] - \sum_{\nu \in N_k} A[i_в[j_k, j_\nu], j_\nu] \, x[j_\nu] - \beta}{(1-\rho) \cdot A[i_в[j_k], j_k]}.$$

Этой формулой и надо воспользоваться вместо формулы (8). Затем можно продолжить стандартный просмотр списка (5), начиная с ребра $j_{k-1} = j_-[j_k]$.

Рассмотрим теперь вопрос о преобразовании графа при замене ребра j'' на ребро j' . Проследим прежде всего, как определяется путь, выходящий из некоторой вершины $i \in M$. Положим $p_0 = i$, $j_{m+1} = 0$ и будем сравнивать верхнюю вершину каждого из ребер $j_k = j_-[j_{k+1}]$, $k = m, m-1, \dots, 1$, с вершиной p_0 до тех пор, пока не обнаружим ребро j_{k_0} , для которого $i_в[j_{k_0}] = p_0$. Обозначим это ребро через q_0 . Если ребро q_0 не находится на цикле (что всегда можно определить с помощью отметок у начального и конечного ребер цикла), то возьмем в качестве p_1 нижнюю вершину ребра q_0 и будем отыскивать среди ребер $j_k = j_-[j_{k+1}]$, $k = k_0-1, k_0-2, \dots, 1$, ребро, для которого вершина p_1 является верхней. Продолжая этот процесс, мы построим путь

$$p_{t+1} \xleftarrow{q_t} p_t \cdots \xleftarrow{q_2} p_2 \xleftarrow{q_1} p_1 \xleftarrow{q_0} p_0 = i , \qquad (9)$$

у которого либо $p_{t+1} = 0$, либо следующее ребро q_{t+1} принадлежит некоторому циклу.

Расчленим операцию замены на три части. Сначала заменим выбывающее ребро j'' на ребро j''_0 , соединяющее вершину $i_в[j'']$ с корнем 0 . Затем ребро j''_0 заменим на ребро j'_0 , тоже примыкающее к вершине 0 . В качестве второй вершины ребра j'_0 берется вершина нового ребра j' , из которой имеется путь в вершину $i_в[j''_0]$. Заметим, что этим свойством всегда обладает хотя бы одна из вершин ребра j' . Выбранная вершина в дальнейшем будет выступать в качестве $i_в[j'_0]$ или $i_в[j']$. Последняя часть преобразований заключается в замене ребра j'_0 на ребро j' .

I. <u>Замена ребра j'' на ребро j''_0</u>. Если ребро j'' не принадлежит циклу, то порядок (5) перестройки не требует: ребро j''_0 может быть помещено на место ребра j'' . Если же ребро j'' принадлежит некоторому циклу, то старое упорядочение можно сохранить лишь в случае, когда j'' является начальным ребром цикла: нужно лишь стереть отметки о начале и конце цикла. Если же в цикле выбывает ребро, не являющееся начальным.

то предварительно можно выполнить надлежащую циклическую перестановку ребер этого цикла. Если j_1^* – начальное, а j_2^* – конечное ребра рассматриваемого цикла, то новый правильный порядок, отвечающий такой циклической перестановке, получится, если положить

$$\bar{j}_+[j_-[j_1^*]] = j'', \quad \bar{j}_+[j_2^*] = j_1^*, \quad \bar{j}_+[j_-[j'']] = j_+[j_2^*],$$
$$\bar{j}_-[j''] = j_-[j_1^*], \quad \bar{j}_-[j_1^*] = j_2^*, \quad \bar{j}_-[j_+[j_2^*]] = j_-[j''].$$

Здесь и в дальнейшем черта сверху обозначает новые значения величин, получающихся после проведенного преобразования.

2. Замена ребра j_0'' на ребро j_0'. При описании перестройки порядка будут использованы операции $S_1[q]$ и $S_2[q]$, аргументом которых служит некоторое ребро $q \in \bar{j}$. Операция $S_1[q]$ заключается в перемещении ребра q на первое место (если оно до этого не было первым) и сводится к выполнению следующих преобразований:

$$\bar{j}_+[0] = q, \quad \bar{j}_+[q] = j_+[0], \quad \bar{j}_+[j_-[q]] = j_+[q],$$
$$\bar{j}_-[q] = 0, \quad \bar{j}_-[j_+[0]] = q, \quad \bar{j}_-[j_+[q]] = j_-[q].$$

Операция $S_2(q)$ перемещает ребро q (если оно не последнее) на последнее место, что может быть достигнуто, если положить

$$\bar{j}_+[j_-[0]] = q, \quad \bar{j}_+[q] = 0, \quad \bar{j}_+[j_-[q]] = j_+[q],$$
$$\bar{j}_-[q] = j_-[0], \quad \bar{j}_-[0] = q, \quad \bar{j}_-[j_+[q]] = j_-[q].$$

Рассмотрим сначала частный случай замены ребра j_0'' на ребро j_0', когда путь из вершины $i_B[j_0']$ в вершину $i_B[j_0'']$ содержит лишь одно ребро, которое мы обозначим через q_1. Нетрудно проверить, что будет правильным порядок следования ребер, при котором на первом месте стоит ребро j_0', на втором – ребро q_1, а дальше без изменения их взаимного расположения

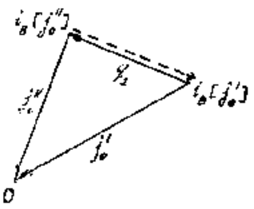


Рис. 2

следует остальные ребра из списка (5) с исключенными ребрами j_0'' и q_1. При этом у ребра q_1 меняются ролями верхняя и нижняя вершины.

В общем случае замена ребра j_0'' на ребро j_0' может быть представлена в виде последовательности преобразований описанного частного вида. Поместим ребро j_0' на место ребра j_0''. Если $i_{\text{в}}[j_0''] = i_{\text{в}}[j_0']$, то никакой перестройки порядка делать не надо. В противном случае проследим путь

$$i_{\text{в}}[j_0''] = s_0 \xleftarrow{q_1} s_1 \cdots \xleftarrow{q_{r-1}} s_{r-1} \xleftarrow{q_r} s_r = i_{\text{в}}[j_0'], \qquad (10)$$

ведущий из вершины $i_{\text{в}}[j_0']$ в вершину $i_{\text{в}}[j_0'']$, и будем попутно выполнять операции $S_2(j_0')$, $S_2(q_r)$, $S_2(q_{r-1}), \ldots, S_2(q_1)$. В полученном порядке на последнее место вынесена цепочка ребер $j_0', q_r, q_{r-1}, \ldots, q_1$. Чтобы получить правильный порядок, эту цепочку мы переставим во главу всего списка ребер, если только она уже там не стоит. Для этого, приняв полученный порядок за исходный, положим

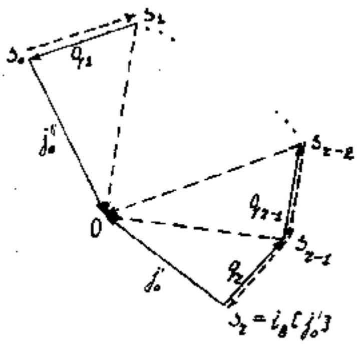


Рис. 3

$$\bar{j}_+[0] = j_0', \quad \bar{j}_+[j_-[0]] = j_+[0], \quad \bar{j}_+[j_-[j_0']] = 0,$$
$$\bar{j}_-[j_0'] = 0, \quad \bar{j}_-[j_+[0]] = j_-[0], \quad \bar{j}_-[0] = j_-[j_0'].$$

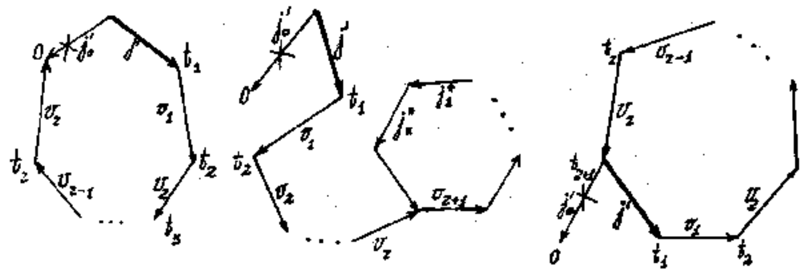
В результате сделанной замены у ребер $q_r, q_{r-1}, \ldots, q_1$ ориентация меняется на противоположную, т.е. верхняя и нижняя вершины меняются ролями.

ЗАМЕЧАНИЕ. Использование операций S_1 и S_2 удобно при описании, но при фактической реализации алгоритма экономнее не дробить процесс, а сразу получать окончательное упорядочение, имея в виду, что сначала должны следовать ребра $j_0', q_r, q_{r-1}, \ldots, q_1$, а затем без изменения взаимного расположения остальные ребра, входившие в перестраиваемый порядок.

3. Замена ребра j_0' на ребро j'. Ориентируем ребро таким образом, чтобы $i_B[j'] = i_B[j_0']$, поставим его на первое место взамен ребра j_0' и проследим путь

$$t_{z+1} \xleftarrow{v_z} t_z \cdots \xleftarrow{v_2} t_2 \xleftarrow{v_1} t_1 = i_H[j'], \tag{II}$$

ведущий из вершины $i_H[j']$. Последовательно для каждого из ребер $v_1, v_2, \ldots, v_z$ этого пути выполним операцию S_1 перестройки порядка: $S_1(v_1), S_1(v_2), \ldots, S_1(v_z)$. Следующие три возможные случая окончания прослеживаемого пути изображены на рис. 4:



а б в

Рис. 4

а) Путь кончается вершиной 0, то есть $i_H[v_z] = 0$; полученный порядок является правильным.

б) Следующее за v_z ребро пути принадлежит некоторому циклу, последнее ребро j_k^* которого мы могли запомнить при прослеживании пути. Пройдя цикл целиком, мы определим и его начальное ребро j_1^*. Для получения правильного порядка положим

$$\bar{j}_+[0] = j_1^*, \quad \bar{j}_+[j_k^*] = j_+[0], \quad \bar{j}_+[j_-[j_1^*]] = j_+[j_k^*],$$
$$\bar{j}_-[j_1^*] = 0, \quad \bar{j}_-[j_+[0]] = j_k^*, \quad \bar{j}_-[j_+[j_k^*]] = j_-[j_1^*].$$

в) Путь кончается верхней вершиной ребра j', то есть $t_{z+1} = i_B[j']$. Это означает, что в результате введения ребра j' образовался новый цикл, состоящий из ребер пути (II) и ребра

j'. В этом случае остается сделать отметки, указывающие, что ребро v_2 является начальным ребром вновь образованного цикла, а j' – его конечным ребром.

Алгоритм с двойным упорядочением

В этом алгоритме за основу взято правильное упорядочение вершин. Порядок следования задается указанием для каждой вершины $i \in M \cup \{0\}$ следующей за ней вершины $i_+[i]$ и предшествующей ей вершины $i_-[i]$. По существу, алгоритм не требует сплошного расположения вершин (и ребер) циклов. Однако при возникновении новых циклов их вершины автоматически оказываются расположенными подряд. Поэтому предположение об этом будет сохранено. При этом мы будем считать отмеченными все вершины циклов, а их начальные вершины отметим особо.

Входная информация о текущем базисе решаемой задачи в основном сводится к информации о ребрах из множества J и может быть расположена в произвольном порядке, соответствующем упорядочению

$$J = \{j_1, j_2, \ldots, j_m\}, \tag{12}$$

не соответствующему, вообще говоря, упорядочению вершин. Для каждой вершины $i \in M$ будем хранить порядковый номер $\nu[i]$ в списке (12) того ребра, для которого вершина i является верхней, так что $j_{\nu[i]} = j[i]$.

ЗАМЕЧАНИЕ. Зафиксировав порядок следования вершин, можно было бы упорядочить ребра таким образом, чтобы выполнялось равенство $j_{\nu[i]} = j_i$, $i \in M$. Тогда при задании информации о ребре $j_{\nu[i]}$ отпала бы необходимость в хранении номера вершины i (так как он совпадает с порядковым номером самого ребра). Так организованную информацию принято называть прямым списком [5].

С помощью номеров $\nu[i]$ облегчается отыскание пути (9), ведущего из некоторой вершины p_0, так как, зная вершину p_k, $k = 0, 1, \ldots$, можно сразу найти $q_k = j_{\nu[p_k]}$, а в качестве p_{k+1} следует взять вершину ребра q_k, не совпадающую с p_k.

Изменения в алгоритмах решения систем касаются только системы (3). Если через i'_1 и i'_2 обозначить номера отличных от нуля компонент столбца $A[M,j']$ и через $E[M,i'_1]$ и $E[M,i'_2]$ соответствующие орты, то правая часть этой системы может быть представлена в виде $A[M,j'] = A[i'_1,j']\cdot E[M,i'_1] + A[i'_2,j']\cdot E[M,i'_2]$, и дело сводится к решению двух систем с правыми частями $A[i'_1,j']\cdot E[M,i'_1]$ и $A[i'_2,j']\cdot E[M,i'_2]$ и сложению полученных результатов. Таким образом, нужно уметь решать системы вида

$$A[M,J]\cdot g[J] = a\cdot E[M,i]. \tag{13}$$

Проследим путь (9), ведущий из вершины i в корень $p_{t+1}=0$ или в вершину p_{t+1} цикла

$$p_{t+1} \xleftarrow{q_{t+z}} p_{t+z} \cdots \xleftarrow{q_{t+2}} p_{t+2} \xleftarrow{q_{t+1}} p_{t+1}.$$

В первом случае будем считать, что $z=0$. Заметим, что во втором случае путь (9) фактически может отсутствовать, если вершина i лежит на цикле. Формально это значит, что $t=-1$. Как известно, отличными от нуля в решении системы (13) являются лишь компоненты, соответствующие ребрам $q_0, q_1, \ldots, q_{t+z}$. Для ребер пути (9) компоненты решения можно вычислить по формулам:

$$g[q_0] = a/A[p_0,q_0],$$

$$g[q_k] = -(A[p_k,q_{k-1}]/A[p_k,q_k])\cdot g[q_{k-1}], \qquad k=1,2,\ldots,t.$$

Если $z=0$, то решение системы (13) этим заканчивается. Если же путь (9) выходит на цикл, то остальные ненулевые компоненты должны быть определены из системы

$$\begin{aligned} &A[p_{t+1},q_{t+1}]\cdot g[q_{t+1}] + A[p_{t+1},q_{t+z}]\cdot g[q_{t+z}] = \bar{a}, \\ &A[p_{t+\nu},q_{t+\nu}]\cdot g[q_{t+\nu}] + A[p_{t+\nu},q_{t+\nu-1}]\cdot g[q_{t+\nu-1}] = 0, \quad \nu=2,3,\ldots,z. \end{aligned} \tag{14}$$

Здесь $\bar{a}=a$ при $t=-1$ и $\bar{a}=-A[p_{t+1},q_t]\cdot g[q_t]$ при $t\geq 0$. Если принять временно $g[q_{t+1}]$ за параметр и из последних $z-1$ равенств в (14) выразить через него $g[q_{t+z}]$, то найдем, что $g[q_{t+z}] = -\rho\cdot A[p_{t+1},q_{t+1}]\cdot g[q_{t+1}]/A[p_{t+1},q_{t+z}]$. Поэтому из первого равенства в (14) можно определить

$$g[q_{t+1}] = \frac{\bar{a}}{(1-p)\cdot A[p_{t+1}, q_{t+1}]} .$$

После этого повторный просмотр цикла позволит найти последовательно $g[q_{t+2}],\ g[q_{t+3}], \ldots, g[q_{t+z}]$.

Выполнение замены ребра j'' на ребро j' снова расчленим на три части, определив вспомогательные ребра j''_0 и j'_0 так же, как в предыдущем алгоритме.

I. <u>Замена ребра j'' на ребро j''_0</u>. Перестраивать порядок следования вершин надо лишь в том случае, если ребро j'' принадлежит циклу. Имеющаяся информация позволяет определить, какая из двух вершин ребра j'' является верхней, так как должно выполняться равенство $j_{v[i_B[j'']]} = j''$. Если $i_B[j'']$ совпадает с начальной вершиной i^* цикла, то можно снова сохранить старый порядок, поместив в списке (I2) ребро j''_0 на место ребра j''. При этом необходимо ликвидировать отметки у всех вершин, принадлежащих разорванному циклу, а также метку, выделяющую начальную вершину этого цикла. Случай же, когда $i_B[j''] \neq i^*$ можно свести к предыдущему, выполнив предварительно циклическую перестановку вершин. По начальной вершине i^* можно легко определить конечную вершину i^*_k рассматриваемого цикла, так как эти вершины образуют ребро $j_{v[i^*]}$. После этого циклическая перестановка сводится к преобразованиям:

$$\bar{i}_+[i_-[i^*]] = i_B[j''], \quad \bar{i}_+[i^*_k] = i^*, \quad \bar{i}_+[i_H[j'']] = i_+[i^*_k],$$

$$\bar{i}_-[i_B[j'']] = i_-[i^*], \quad \bar{i}_-[i^*] = i^*_k, \quad \bar{i}_-[i_+[i^*_k]] = i_H[j''].$$

2. <u>Замена ребра j''_0 на ребро j'_0</u>. Мы снова будем использовать две вспомогательные операции $\tilde{S}_1(p)$ и $\tilde{S}_2(p)$, которые отличаются от ранее введенных операций $S_1(q)$ и $S_2(q)$ лишь тем, что в них перестраивается порядок следования вершин, а не ребер. Выполнение операции $\tilde{S}_1(p)$ состоит в том, что при $i_+[0] \neq p$ мы полагаем

$$\bar{i}_+[0] = p, \quad \bar{i}_+[p] = i_+[0], \quad \bar{i}_+[i_-[p]] = i_+[p],$$

$$\bar{i}_-[p] = 0, \quad \bar{i}_-[i_+[0]] = p, \quad \bar{i}_-[i_+[p]] = i_-[p].$$

Аналогично при $i_-[0] \neq p$ выполняется операция $\tilde{S}_2(p)$:

$$\tilde{i}_+[i_-[0]] = p, \quad \tilde{i}_+[p] = 0, \quad \tilde{i}_+[i_-[p]] = i_+[p],$$

$$\tilde{i}_-[p] = i_-[0], \quad \tilde{i}_-[0] = p, \quad \tilde{i}_-[i_+[p]] = i_-[p].$$

Поместим, прежде всего, ребро j_0' на место ребра j_0'' и рассмотрим путь (10), ведущий из вершины $i_B[j_0']$ в вершину $i_B[j_0'']$ (напомним, что в качестве верхней вершины ребра j_0' взята та вершина вводимого ребра j', из которой имеется путь в вершину $i_B[j_0'']$). Перенесем первую вершину $i_B[j_0']$ пути (10) в конец, т.е. выполним операцию $\tilde{S}_2(s_r)$. Кроме того, положим $\tilde{\nu}[i_B[j_0']] = \nu[i_B[j_0'']]$, указав тем самым, что вершина $i_B[j_0'] = s_r$ будет являться теперь верхней для ребра j_0' (стоящего в списке (12) на месте ребра j_0''). Затем последовательно для вершин s_k , $k = r-1, r-2, \ldots, 0$, пути (10) выполним операцию $\tilde{S}_2(s_k)$, полагая $\tilde{\nu}[s_k] = \nu[s_{k+1}]$. Примем полученный после этого порядок за исходный и, положив

$$\tilde{i}_+[0] = s_r, \quad \tilde{i}_+[i_-[0]] = i_+[0], \quad \tilde{i}_+[i_-[s_r]] = 0,$$

$$\tilde{i}_-[s_r] = 0, \quad \tilde{i}_-[i_+[0]] = i_-[0], \quad \tilde{i}_-[0] = i_-[s_r],$$

придем к правильному упорядочению вершин.

3. <u>Замена ребра j_0' на ребро j'</u>. Поместим ребро j' на место ребра j_0' и будем прослеживать путь (11) из вершины $t_1 = i_H[j']$, выполняя последовательно операции $\tilde{S}_1(t_1)$, $\tilde{S}_1(t_2), \ldots, \tilde{S}_1(t_r)$. Если $t_{r+1} = 0$, то на этом перестройка порядка заканчивается. Если же путь (11) привел к вершине t_{r+1} некоторого цикла

$$t_{r+1} \xrightarrow{v_{r+k}} t_{r+k} \cdots \xrightarrow{v_{r+2}} t_{r+2} \xrightarrow{v_{r+1}} t_{r+1}, \qquad (15)$$

то попутно с прослеживанием этого цикла выполним последовательно операции $\tilde{S}_1(t_{r+1})$, $\tilde{S}_1(t_{r+2}), \ldots, \tilde{S}_1(t_{r+k})$. Поскольку при этом произойдет циклическая перестановка вершин цикла, следует ликвидировать метку у бывшей начальной вершины этого цикла, выделив t_{r+k} в качестве новой начальной вершины. Наконец, возможен случай, когда $t_{r+1} = i_B[j']$, и образуется новый цикл.

Выполним операцию $\tilde{S}_1(t_{z+1})$. Для оформления этого цикла следует еще раз пройти путь (II), отметив вершины этого цикла и выделив t_{z+1} в качестве начальной его вершины.

Алгоритм с усиленным упорядочением

В этом алгоритме мы не будем требовать, чтобы в правильном упорядочении (4) образующие цикл вершины располагались подряд, но наложим другое дополнительное условие на правильное упорядочение вершин. Зафиксируем некоторую вершину $i \in M \cup \{0\}$ и рассмотрим направленный путь

$$i \xleftarrow{q_z} p_z \cdots \xleftarrow{q_1} p_1 \xleftarrow{q_0} p_0 = i'. \qquad (16)$$

Если вершина i не принадлежит циклу, то во всяком пути (16) вершины i, p_z, ..., p_1, i' расположены в том же порядке, что и в списке (4). Естественно такой путь называть монотонным. Если же вершина i принадлежит некоторому циклу с начальной вершиной i^*, то для монотонности пути (16) нужно, чтобы он не содержал ребра $j_{v[i^*]}$. Совокупность вершин i', для которых имеется монотонный путь (16), будем называть *верхним множеством* вершины i и обозначать символом $B(i)$. Саму вершину i мы также включим в ее верхнее множество.

Усиление упорядочения заключается в том, что для каждого $i \in M \cup \{0\}$ элементы множества $B(i)$ должны располагаться подряд, причем последнюю вершину этого множества обозначим через $\mu[i]$. Помимо списка $\mu[i]$, как и раньше, будем хранить списки $i_+[i]$, $i_-[i]$, $v[i]$, а также список ребер (12). Кроме того, будем считать отмеченными все вершины циклов и выделенными особо их начальные вершины. Замену ребра j'' на ребро j' снова расчленим на три части, введя вспомогательные ребра j_0'' и j_0' так же, как это было сделано выше.

I. <u>Замена ребра j'' на ребро j_0''</u>. Ребро j_0'' в списке (12) мы поместим на место ребра j'' и все вершины нового верхнего множества $\bar{B}(i_B[j_0''])$ поместим в конец нового верхнего множества $\bar{B}(0)$. Если ребро j'' не принадлежало циклу, то для этого (при $\mu[0]\neq\mu[i_в[j''_o]]$) нужно положить

$$\bar\mu[0]=\mu[i_в[j'']], \quad \bar i_+[\mu[0]]=i_в[j''], \quad \bar i_+[\mu[i_в[j'']]]=i_+[\mu[0]],$$

$$\bar i_+[i_-[i_в[j'']]]=i_+[\mu[i_в[j'']]], \qquad \bar i_-[i_в[j'']]=\mu[0], \tag{17}$$

$$\bar i_-[i_+[\mu[0]]]=\mu[i_в[j'']], \qquad \bar i_-[i_+[\mu[i_в[j'']]]]=i_-[i_в[j'']].$$

Для вершин ρ_0 , ρ_1 , ..., ρ_r пути

$$\rho_{r+1}\xleftarrow{q_r}\rho_r\cdots\xleftarrow{q_1}\rho_1\xleftarrow{q_0}\rho_0=i_н[j''],$$

конечная вершина ρ_{r+1} которого либо равна нулю, либо является начальной вершиной некоторого цикла, в случае выполнения равенства $\mu[\rho_k]=\mu[i_в[j'']]$ следует положить $\bar\mu[\rho_k]=$ $=i_-[i_в[j'']]$. Если $\rho_{r+1}\neq 0$, то при выполнении равенства $\mu[\rho_{r+1}]=\mu[i_в[j'']]$ положим также $\bar\mu[\rho_{r+1}]=i_-[i_в[j'']]$. В случае, когда $\mu[0]=\mu[i_в[j''_o]]$, никаких преобразований делать не нужно.

Если ребро j'' принадлежит циклу и $i_в[j'']$ совпадает с его начальной вершиной i^* , то нужно ликвидировать все отметки у вершин цикла, положить $\bar\mu[0]=\mu[i^*]$ и, если $i_+[\mu[0]]\neq i^*$, выполнить преобразования (17). Если же $i_в[j'']\neq i^*$, то предварительно следует сделать циклическую перестановку вершин с тем, чтобы новая начальная вершина цикла совпадала с $i_в[j'']$.

Для осуществления циклической перестановки построим путь

$$i^*=\rho_{s+1}\xleftarrow{q_s}\rho_s\cdots\xleftarrow{q_1}\rho_1\xleftarrow{q_0}\rho_0=i_н[j''], \tag{18}$$

ведущий из вершины $i_н[j'']$ в вершину i^* . Переместим сначала начальную вершину цикла из последней вершины этого пути ρ_{s+1} в его предпоследнюю вершину ρ_s , для чего положим

$$\bar i_+[i_-[\rho_{s+1}]]=\rho_s, \quad \bar i_+[\mu[\rho_s]]=\rho_{s+1}, \quad \bar i_+[i_-[\rho_s]]=i_+[\mu[\rho_s]],$$

$$\bar i_-[\rho_s]=i_-[\rho_{s+1}], \quad \bar i_-[\rho_{s+1}]=\mu[\rho_s], \quad \bar i_-[i_+[\mu[\rho_s]]]=i_-[\rho_s].$$

Кроме того, если $\mu[\rho_{s+1}]=\mu[\rho_s]$, то нужно положить $\bar\mu[\rho_{s+1}]=$ $=i_-[\rho_s]$ и $\bar\mu[\rho_s]=i_-[\rho_s]$, а если $\mu[\rho_{s+1}]\neq\mu[\rho_s]$, то следует лишь положить $\bar{\mu}[\rho_s]=\mu[\rho_{s+1}]$, сохранив старое значение $\mu[\rho_{s+1}]$. После этого можно переместить начальную вершину в ρ_{s-1}, ρ_{s-2} ... и т.д.

2. <u>Замена ребра j_0'' на ребро j_0'</u>. Запомним номер $j=\nu[i_B[j_0'']]$. Построим путь (10), ведущий из вершины $i_B[j_0']$ в вершину $i_B[j_0'']$.

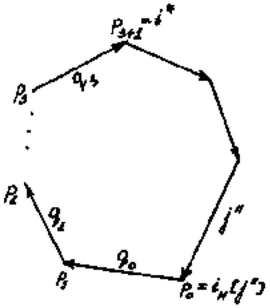


Рис. 5

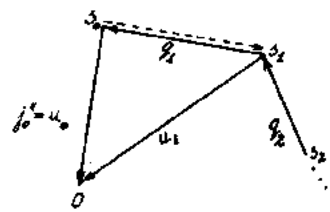


Рис. 6

Заменим сначала ребро $j_0''=u_0$ на ребро u_1, соединяющее вершину s_1 пути (10) с корнем 0. Затем подменим ребро u_1 на ребро u_2, верхняя вершина которого равна s_2 и так далее, пока не получим ребро $u_r=j_0'$. Один шаг этого процесса, состоящий в замене ребра u_k на ребро u_{k+1}, сводится к следующим преобразованиям. Нужно положить

$$\bar{i}_+[i_-[s_k]]=s_{k+1},\quad \bar{i}_+[\mu[s_{k+1}]]=s_k,\quad \bar{i}_+[i_-[s_{k+1}]]=i_+[\mu[s_{k+1}]],$$

$$\bar{i}_-[s_{k+1}]=i_-[s_k],\quad \bar{i}_-[s_k]=\mu[s_{k+1}],\quad \bar{i}_-[i_+[\mu[s_{k+1}]]]=i_-[s_{k+1}].$$

Кроме того, если $\mu[s_k]=\mu[s_{k+1}]$, то следует принять $\bar{\mu}[s_{k+1}]=i_-[s_{k+1}]$ и $\bar{\mu}[s_k]=i_-[s_{k+1}]$, а при $\mu[0]=\mu[s_{k+1}]$ также положить $\bar{\mu}[0]=i_-[s_{k+1}]$. Если же $\mu[s_k]\neq\mu[s_{k+1}]$, то достаточно лишь положить $\bar{\mu}[s_{k+1}]=\mu[s_k]$. Поскольку меняется направленность ребра q_{k+1}, то следует положить $\bar{\nu}[s_k]=\nu[s_{k+1}]$. Надо было бы еще поместить ребро u_{k+1} на место ребра u_k и указать, что $\bar{\nu}[s_{k+1}] = j$. Мы этого делать не будем, а лишь после выполнения всех шагов перестройки (при $k = 0, 1, \dots, r-1$) поместим ребро j_0' на место ребра j_0'' и укажем, что $\bar{\nu}[s_r] = j$.

8. <u>Замена ребра j_0' на ребро j'</u>. При выполнении описываемой части алгоритма следует помнить, что в полученном к этому моменту порядке $\mu[0] = \mu[i_B[j']]$. Построим путь

$$t_{r+1} \xleftarrow{v_r} t_r \cdots \xleftarrow{v_2} t_2 \xleftarrow{v_1} t_1 = i_H[j'], \tag{19}$$

в котором вершина t_{r+1} либо равна нулю, либо совпадает с начальной вершиной i^* некоторого цикла, либо равна верхней вершине ребра j' (см. рис. 4,в).

Поместим ребро j' на место ребра j_0'. В первых двух случаях, если $i_+[t_1] \neq i_B[j']$, положим $\bar{\mu}[0] = i_-[i_B[j']]$ и

$$\bar{i}_+[t_1] = i_B[j'], \quad \bar{i}_+[\mu[i_B[j']]] = i_+[t_1],$$

$$\bar{i}_+[i_-[i_B[j']]] = i_+[\mu[i_B[j']]], \quad \bar{i}_-[i_B[j']] = t_1,$$

$$\bar{i}_-[i_+[t_1]] = \mu[i_B[j']], \quad \bar{i}_-[i_+[\mu[i_B[j']]]] = i_-[i_B[j']].$$

Кроме того, для всех вершин t_k $(k = 1, 2, \dots, t_{r+1})$ пути (19), за исключением нулевой вершины, в которых выполнялось равенство $\mu[t_k] = t_1$, положим $\bar{\mu}[t_k] = \mu[i_B[j']]$.

Остался нерассмотренным последний случай, когда $t_{r+1} = i_B[j']$, и образовался новый цикл. Для оформления цикла нужно отметить вершины пути (19) как вершины, принадлежащие циклу, а вершину $i_B[j']$ объявить начальной его вершиной. Кроме того, следует положить $\bar{\mu}[0] = i_-[i_B[j']]$.

Как уже говорилось, описываемый алгоритм дает возможность довычислять те компоненты решения $y[M]$ системы (2), которые фактически изменяются при переходе к новому базису. Пусть мы знаем решение $y[M]$ системы (2), отвечающее базисному множеству J. При переходе к новому базисному множеству $\bar{J} = (J \setminus \{j^0\}) \cup \{j'\}$ изменятся лишь компоненты $y[M]$, отвечающие новому верхнему множеству вершины $i_B[j']$. Поскольку все вершины этого множества расположены подряд, то достаточно организовать просмотр лишь этой части общего списка вершин (4).

Если $i_B[j']$ не является начальной вершиной цикла, то положим $i_1 = i_B[j']$ и, переходя последовательно к $i_2 = i_+[i_1]$, $i_3 =$

$= i_+[i_2], \ldots, i_\ell = i_+[i_{\ell-1}] = \mu[i_s[j']]$, вычислим значения $y[i_k]$, $k = 1, 2, \ldots, \ell$ по формулам (6), имея при этом в виду, что $i_s[i_k] = i_k$, $j_k = j_{v[i_k]}$, а $i_н[j_k]$ – вершина ребра $j_{v[i_k]}$, не совпадающая с вершиной i_k.

Если же $i_s[j']$ является начальной вершиной вновь образованного цикла, то проследим путь (19) и вычислим значение $y[i_s[j']]$ так же, как это делалось в первом алгоритме. После этого можно вычислять последовательно компоненты $y[M]$, отвечающие остальным элементам $i_2 = i_+[i_1]$, $i_3 = i_+[i_2], \ldots, i_\ell = i_+[i_{\ell-1}] = \mu[i_s[j']]$ верхнего множества $B(i_s[j'])$.

Что касается системы (3), то ее решение определяется так же, как в алгоритме с двойным упорядочением.

Литература

1. ЕРУДНО А.Л. Решение транспортной задачи методом вычеркивающей нумерации. – В кн.: Применение цифровых вычислительных машин в экономике. М., 1962, с. 17-38.
2. БУЛАВСКИЙ В.А. Об одном алгоритме решения транспортной задачи. – В кн.: Оптимальное планирование. вып. 2, Новосибирск 1964, с. 41–49.
3. ЯКОВЛЕВА М.А. Некоторые методы решения задач линейного программирования. – Диссертация на соискание ученой степени кандидата физико-математических наук. Новосибирск, 1968, 125 с.
4. ЗМЫРЁВ В.И. Алгоритм решения одного класса задач линейного программирования большого объема. – В кн.: Оптимальное планирование. вып. 2, Новосибирск, с. 88–116.
5. КИМ Е... Об эффективности алгоритмов решения двухкомпонентных задач линейного программирования. – "Экономика и матем. методы", 1974, т. X, вып. 3, с. 621–631.
6. ГАВУРИН М.К., РУБИНШТЕЙН Г.Ш., СУРИН С.С. Об оптимальном использовании средств при выполнении нескольких видов работ. – "Сиб.мат.журн.", 1962, т. 3, № 4, с.481–499.
7. ЯКОВЛЕВА М.А. Двухкомпонентная задача линейного программирования. – В кн.: Оптимальное планирование. вып.2 Новосибирск, 1964, с. 23–40.

Поступила в ред.-изд. отд.
23. VI. 1976 г.