УДК 519.10

НОВЫЙ АЛГОРИТМ ПОИСКА КРИТИЧЕСКИХ ПУТЕЙ В ГРАФЕ И ЕГО ПРИЛОЖЕНИЯ

Канд. физ.-мат. наук, доц. КОРЗНИКОВ А. Д.

Белорусский национальный технический университет

Математической моделью широкого круга проблем, связанных с задачами упорядочения, теории расписаний и сетевого планирования, является поиск критических путей между вершинами графа. Известные алгоритмы решения этих задач [1, 2] используют правильную нумерацию вершин графа и их пометки, т. е. существенным образом основаны на графическом представлении последнего. Однако даже в задачах сетевого планирования, где изначально весь комплекс операций задан их продолжительностью и отношением предшествования, построение графической модели является самостоятельной задачей. Указанные недостатки значительно усложняют программную реализацию этих алгоритмов и делают их малопривлекательными, если математической моделью является граф с большим количеством вершин и дуг.

Идея осуществления тернарных операций над вершинами графа для отыскания кратчайших путей, впервые предложенная в [3], оказалась достаточно плодотворной. Обобщение этого аппарата позволило получить эффективные алгоритмы решения широкого круга задач оптимизации на графах [4, 5]. Ниже приводится алгоритм отыскания путей максимального веса в многополюсной сети, позволяющей получить критические пути между любыми парами вершин в графе, не содержащем ориентированных циклов.

Рассмотрим ориентированный граф $G(V, U)$, не содержащий циклов с n вершинами $\left(|V| = n\right)$ и множеством дуг U. Каждой дуге $(i,\ j) \in U$ поставлено в соответствие число c_{ij} – вес дуги $(i,\ j) \in U$ ($c_{ij} = \infty$, если $(i,\ j) \notin U$). Под весом любого $(i - j)$-пути, ведущего из вершины i в вершину j $\left((i-j) = \{(i,\ i_1),\ (i_1,\ i_2), ..., (i_k,\ j)\}\right)$, будем понимать суммарный вес всех дуг $\sum_{(i,j)\in(i-j)} c_{ij}$, входящих в этот путь. Заметим, что граф $G(V, U)$ полностью описывается матрицей дуговых весов $C = \left\| c_{ij} \right\|_{n\times n}$. Определим тернарную операцию над матрицей C по элементу k следующим образом:

$$c_{ij} := \begin{cases} c_{ij}, & \text{если} \quad c_{ij} \neq \infty \quad \text{и} \quad c_{ij} \geq c_{ik} + c_{kj}, \\ & \text{либо} \quad c_{ik} + c_{kj} = \infty; \\ c_{ik} + c_{kj}, & \text{если} \quad c_{ik} + c_{kj} \neq \infty \quad \text{и} \quad c_{ij} < c_{ik} + c_{kj}, \\ & \text{либо} \quad c_{ij} = \infty \end{cases} \qquad (1)$$

для всех $i \neq j \neq k$.

Утверждение 1. Выполнения тернарных операций по всем элементам $k = 1, ..., n$ матрицы $C = \left\| \tilde{n}_{ij} \right\|_{n\times n}$ дает матрицу, элемент которой c_{ij} равен максимальному из весов всех путей, ведущих из вершины i в вершину j.

Доказательство. Действительно, рассмотрим путь максимального веса, ведущий из вершины l в вершину p. Выполнение тернарной операции по элементу k, если вершина k не входит в этот путь, не изменит его, так как в противном случае, если для некоторого k имеем $c_{ik} + c_{kj} > c_{ij}$, где i, j – две соседние вершины пути, то, заменив дугу (i, j) на дуги (i, k), (k, j), получим $(l - p)$-путь большего веса. Но это противоречит тому, что рассматриваемый путь имеет максимальный вес.

Пусть i_0 – промежуточная вершина с минимальным индексом в рассматриваемом пути, а i_1 и i_2 – соседние с ней вершины. Выполнение тернарной операции (1) по элементу i_0 матрицы C дает значение $c_{i_1 i_2} := c_{i_1 i_0} + c_{i_0 i_2}$. Заменив две дуги ($i_1$, i_0) и (i_0, i_2) на одну дугу (i_1, i_2), получим путь такого же веса, но содержащего на одну вершину меньше. Продолжая подобным образом, получаем доказательство нашего утверждения.

Заметим, что выполнение тернарных операций по всем вершинам сети дает матрицу весов максимальных путей, но не сами пути. Для отыскания этих путей воспользуемся вспомогательной матрицей $R = \|r_{ij}\|_{n\times n}$, элементы которой на начальном этапе: $r_{ij} = j, \quad i = \overline{1,n}, \quad j = \overline{1,n}.$ Одновременно с осуществлением тернарных операций по элементу k матрицы C элементы матрицы R будут изменяться следующим образом:

$$r_{ij} := \begin{cases} r_{ij}, & \text{если} \quad c_{ij} := c_{ij}; \\ r_{ik}, & \text{если} \quad c_{ij} := c_{ik} + c_{kj}. \end{cases} \qquad (2)$$

Утверждение 2. После проведения тернарных операций по всем вершинам сети элемент r_{ij} матрицы R будет указывать индекс вершины i_1, следующей за вершиной i в (i – j)-пути максимального веса. Докажем это утверждение.

Доказательство. Рассмотрим некоторый (i – j)-путь максимального веса, а i, i_1, i_2 – первые три вершины этого пути. Поскольку элементы матрицы R меняются только при изменении элементов матрицы C, они не будут изменяться при проведении тернарной операции по элементам матрицы C, если соответствующие им вершины не входят в рассматриваемый путь. При проведении тернарной операции по вершине i_1 получим $c_{ii_1} + c_{i_1 i_2} > c_{ii_2}$, поэтому $r_{ii_2} = r_{ii_1} = i_1$. Заменим дуги $(i,\ i_1)$, $(i_1,\ i_2)$ на одну дугу (i_1, i_2), получим новый максимальный путь того же веса, содержащий на одну вершину меньше. При дальнейшем выполнении тернарных операций величина r_{ii_2} не изменится. Продолжая подобным образом, получаем требуемое доказательство.

После осуществления тернарных операций (1) по всем элементам $k = 1, 2, ..., n,$ выполняя одновременно операции (2), получаем матрицу максимальных весов путей между всеми парами вершин. Сами пути находим с помощью вспомогательной матрицы R следующим образом.

Для любого (i – j)-пути последовательно находим $r_{ij} = i_1, \quad r_{i_1 j} = i_2, ..., \quad r_{i_k j} = j$. Искомый путь максимального веса между вершинами i и j проходит через вершины: $i,\ i_1,\ i_2,\ \ldots,\ i_k,\ j$.

Проиллюстрируем работу алгоритма на следующем примере. Зададим граф матрицей дуговых весов

$$C = \begin{bmatrix} \infty & 5 & 3 & 7 & \infty & \infty \\ \infty & \infty & \infty & 1 & \infty & \infty \\ \infty & 4 & \infty & \infty & 6 & \infty \\ \infty & \infty & \infty & \infty & 5 & 6 \\ \infty & \infty & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}.$$

Последовательно по всем элементам k = 1, 2, 3, 4, 5, 6 осуществим тернарные операции (1), изменяя одновременно элементы матрицы R по формулам (2). В результате получим:

$$C = \begin{bmatrix} \infty & 7 & 3 & 8 & 13 & 15 \\ \infty & \infty & \infty & 1 & 6 & 8 \\ \infty & 4 & \infty & 5 & 10 & 12 \\ \infty & \infty & \infty & \infty & 5 & 7 \\ \infty & \infty & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix};$$

$$R = \begin{bmatrix} 1 & 3 & 3 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 & 4 \\ 1 & 2 & 3 & 2 & 2 & 2 \\ 1 & 2 & 3 & 4 & 5 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}.$$

Критический путь (путь максимального веса) ведет из вершины 1 в вершину 6 и имеет вес 15 единиц. С помощью вспомогательной матрицы R находим:

$$r_{16} = 3; \quad r_{36} = 2; \quad r_{26} = 4; \quad r_{46} = 5; \quad r_{56} = 6,$$

т. е. критический путь состоит из дуг (1, 3), (3, 2), (2, 4), (4, 5), (5, 6). Аналогичным образом можно найти путь максимального веса между любой парой вершин графа. Поскольку элементы первого столбца и шестой строки равны бесконечности, вершина с номером 1 не является концевой ни для одной дуги (начальная вершина), а из вершины с номером 6 не выходит ни одна дуга (конечная вершина). С помощью найденных критических путей и матрицы *C* легко получить графическое изображение заданного графа (рис. 1), числа над дугами равны их весам.

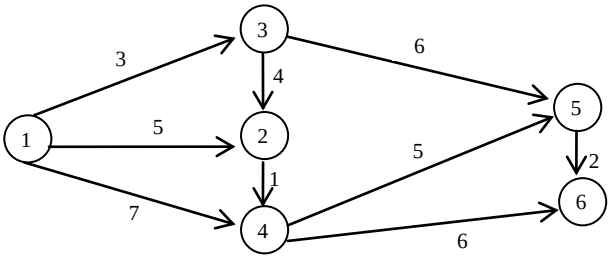

Рис. 1

Задачи упорядочения носят самый общий характер. Они возникают повсюду, где существует возможность выбора той или иной очередности выполнения некоторых операций.

Зачастую в задачах теории расписаний и сетевого планирования описание комплекса работ требует задания довольно сложного графа, определяющего отношение порядка и характеризующего ограничения на предшествование среди операций данного комплекса. Графы можно использовать для задания отношения порядка как между вершинами, так и между дугами, и их построение уже является самостоятельной задачей.

Рассмотрим теперь задачу упорядочения элементов некоторого конечного множества $U = \{a_i\}$ мощности *n*. Для каждого элемента a_i определено отношение порядка, т. е. задано множество $U_i \subseteq U$ элементов, которым элемент a_i непосредственно предшествует $\left(a_i \prec a_j \quad \forall a_j \in U_i\right)$. Если каждому элементу $a_i \in U$ поставить в соответствие вершину *i* некоторого графа и соединить ее с вершиной *j* ориентированной дугой (*i*, *j*) для всех $a_j \in U_i$, то наличие пути из вершины i_1 в i_k, содержащего дуги $(i_1, i_2), (i_2, i_3), ..., (i_{k-1}, i_k)$, указывает на существование отношений порядка между элементами

$$a_{i_1} \prec a_{i_2} \prec ... \prec a_{i_k}.$$

В этом случае говорят, что a_{i_1} предшествует a_{i_k} $\left(a_{i_1} \prec\prec a_{i_k}\right)$.

Таким образом, заданный выше описанным способом граф с *n* вершинами может быть использован для определения отношения порядка между элементами конечного множества. Для этого достаточно присвоить всем дугам (*i*, *j*) одинаковые веса (например, равные 1), если $a_i \prec a_j$, а веса остальных дуг положить равными бесконечности.

Применив тернарные операции (1) к матрице весов и выполнив операции (2), получим матрицу критических путей между всеми парами вершин (*i*, *j*). Если элемент c_{ij} конечен, то это означает, что элемент a_i предшествует элементу a_j $\left(a_i \prec\prec a_j\right)$, а вся цепочка непосредственно предшествующих друг другу элементов (критический путь) восстанавливается с помощью вспомогательной матрицы *R*, как это было показано выше. Причем для установления отношения предшествования между элементами множества *U* достаточно информации, содержащейся в матрицах *C* и *R*. Так, если $c_{i,j} = 1$, то $a_i \prec a_j$, если элемент $c_{i_1,j}$ равен $k\left(1 < k < n\right)$, то $a_i \prec\prec a_j$, причем вся цепочка непосредственно предшествующих друг другу элементов содержит их ровно *k*: $a_{i_1} \prec a_{i_2} \prec ... \prec a_{i_{k-1}} \prec \prec a_{i_k} \prec a_j$, где $i_2 = r_{i_1 j}$, $i_3 = r_{i_2 j}, ...,$ $i_k = r_{i_{k-1} j}$, $r_{i_p j}$ – соответствующие элементы матрицы *R*.

В качестве иллюстрации работы алгоритма упорядочения элементов конечного множества рассмотрим следующий пример. Отношение непосредственного предшествования задано табл. 1.

Таблица 1

i	1	2	3	4	5	6	7	8
U_i	2,5	7	6,8	6,8	6,8	7	–	–

Сформируем начальную матрицу весов *C*

$$C=\begin{bmatrix}\infty & 1 & \infty & \infty & 1 & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty\\ \infty & \infty & \infty & \infty & \infty & 1 & \infty & 1\\ \infty & \infty & \infty & \infty & \infty & 1 & \infty & 1\\ \infty & \infty & \infty & \infty & \infty & 1 & \infty & 1\\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty\end{bmatrix}.$$

После осуществления операций (1), (2) получим матрицы C и R:

$$C=\begin{bmatrix}\infty & 1 & \infty & \infty & 1 & 2 & 3 & 2\\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty\\ \infty & \infty & \infty & \infty & \infty & 1 & 2 & 1\\ \infty & \infty & \infty & \infty & \infty & 1 & 2 & 1\\ \infty & \infty & \infty & \infty & \infty & 1 & 2 & 1\\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty\end{bmatrix};$$

$$R=\begin{bmatrix}1 & 2 & 3 & 4 & 5 & 5 & 5 & 5\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 6 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 6 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 6 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\end{bmatrix}.$$

Таким образом, имеем:

$a_1 \prec\prec a_6 \quad (r_{16}=5,\ \ r_{56}=6\ )$ и $a_1 \prec a_5 \prec a_6$;

$a_1 \prec\prec a_7\,(r_{17}=5,\ \ r_{57}=6,\ \ r_{67}=7),\ \ a_1 \prec a_5 \prec a_6 \prec a_7$;

$a_1 \prec\prec a_8\,(r_{18}=5,\ \ r_{58}=8),\ \ a_1 \prec a_5 \prec a_8$;

$a_3 \prec\prec a_7 \quad (r_{37}=6,\ \ r_{67}=7),\ \ a_3 \prec a_6 \prec a_7$;

$a_4 \prec\prec a_7\,(r_{47}=6,\ \ r_{67}=7),\ \ a_4 \prec a_6 \prec a_7$;

$a_5 \prec\prec a_7 \quad (r_{57}=6,\ \ r_{67}=7),\ \ a_5 \prec a_6 \prec a_7$.

Учитывая отношение непосредственного предшествования и полученные результаты, легко построить граф, задающий отношение порядка между вершинами (элементами a_i множества U):

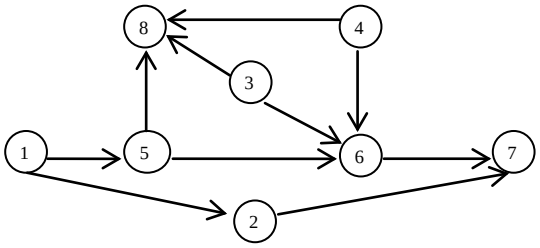

Рис. 2

Поскольку элементы седьмой и восьмой строк первого, третьего и четвертого столбцов матрицы C равны бесконечности, элементы a_7, a_8 не предшествуют никаким другим элементам (конечные вершины графа) и нет элементов, предшествующих элементам a_1, a_3, a_4 (начальные вершины).

В задачах сетевого планирования выполнения комплекса операций (работ) для каждой операции i, $i=\overline{1,n}$, известна ее продолжительность t_i, $i=\overline{1,n}$, и задано отношение предшествования, т. е. определено множество U_i, $i=\overline{1,n}$, операций, выполнению которых непосредственно предшествует выполнение операции i, $i=\overline{1,n}$. Ориентированный граф, представляющий взаимосвязь отдельных операций (сетевой график), будет содержать $2n$ вершин (каждой операции i поставлена в соответствие дуга, т. е. пара вершин $(i, n+i)$, $i=\overline{1,n}$). Матрицей дуговых весов $C_{2n\times 2n}$ на начальном этапе будет блочная матрица порядка $2n\times 2n$

$$C_{2n\times 2n}=\left[\begin{array}{c|c} C^{11}_{n\times n} & C^{12}_{n\times n}\\ \hline C^{21}_{n\times n} & C^{22}_{n\times n}\end{array}\right].$$

Причем полагаем элементы $c_{i,n+i}$ подматрицы $C^{12}_{n\times n}$, равными t_i, $i=\overline{1,n}$, элементы $c_{n+i,j}$ подматрицы $C^{21}_{n\times n}$, равными нулю, если $j\in U_i$ (операция i непосредственно предшествует операции $j\in U_i$), $i=\overline{1,n}$. Все остальные элементы матрицы $C_{2n\times 2n}$ равны бесконечности (т. е. достаточно большое число).

Из содержания задачи, для которой построена сетевая модель, следует, что рассматриваемый граф не может содержать циклов. После осуществления тернарных операций (1) и (2) последовательно для всех k = 1, 2, …, 2n, получим матрицу $C = \|c_{ij}\|_{2n\times 2n}$ и вспомогательную матрицу $R = \|r_{ij}\|_{2n\times 2n}$, с помощью которых можно получить все числовые характеристики сетевого графика, а при необходимости – и его графическое изображение. Так, все вершины i, $i = \overline{1, n}$, для которых $c_{n+j,i} = \infty$ для любого $j = \overline{1, n}$, являются начальными вершинами сетевого графика (множество V_0), а соответствующие им операции – начальными. Вершины $n+i$, $i = \overline{1, n}$, для которых $c_{n+i,j} = \infty$ для всех $j = \overline{1, n}$, являются конечными вершинами сетевого графика (множество V_k), соответствующие же им операции – завершающими. Критическое время выполнения всего комплекса операций

$$t_k = \max_{i\in V_0,\, j\in V_k} \left\{ c_{ij} \,\middle|\, c_{ij} \neq \infty \right\} c_{ij} = c_{i_0 j_k}.$$

Критические операции, составляющие критический путь $P_k = \{i_0, i_1, i_2, ..., i_k\}$, находятся с помощью вспомогательной матрицы $R: i_1 = r_{n+i_0,i_k}$, $i_2 = r_{n+i_1,i_k}$, ..., $i_k = r_{n+i_{k-1},i_k}$.

Ранний срок начала выполнения любой операции j равен

$$T_j^{\text{рн}} = \begin{cases} \max\limits_{i\in V_0} \{c_{ij} / c_{ij} \neq \infty\}, & j \notin V_0; \\ 0, & j \in V_0. \end{cases}$$

Соответственно ранний срок ее окончания

$$T_j^{\text{ро}} = T_j^{\text{рн}} + t_j,$$

или

$$T_j^{\text{ро}} = \max_{i\in 1,n} \{c_{i,n+j} / c_{i,n+j} \neq \infty\},\ j = \overline{1, n}.$$

Поздним сроком начала $T_j^{\text{пн}}$ и окончания $T_j^{\text{по}}$ любой операции j называется наибольшее допустимое время начала и окончания этой операции без нарушения срока завершения всего проекта. Понято, что:

$$T_j^{\text{рн}} = T_j^{\text{пн}},\ T_j^{\text{ро}} = T_j^{\text{по}},\ \forall j \in P_k;$$

$$T_j^{\text{по}} = t_k,\ T_j^{\text{пн}} = t_k - t_j,\ n + j \in V_k.$$

Для операций j таких, что $U_j \bigcap P_k = \{l\}$ $T_j^{\text{по}} = T_l^{\text{рн}}$, $T_j^{\text{пн}} = T_j^{\text{по}} - t_j$.

Таким образом, свободные резервы времени r_j могут иметь только конечные операции и операции, непосредственно предшествующие критическим; $r_j = T_j^{\text{по}} - T_j^{\text{ро}}$, если $n + j \in V_k$ или $U_j \bigcap P_k \neq \varnothing$.

Любая операция, предшествующая критическим, принадлежит пути (возможно, нескольким), ведущим из начальных вершин в начальную вершину критической операции. Поэтому полный резерв времени таких операций равен минимальному значению разности между ранним сроком начала выполнения соответствующей критической операции и длиной этого пути:

$$R_i = \min_{(i,n+i)\in(l-k)} \left(T_k^p - c_{lk} \right),\ l \in V_0,\ k \in P_k.$$

Если операция i не предшествует ни одной из критических операций (т. е. предшествует завершающим), то ее свободный резерв времени равен минимальному значению разности между критическим временем и ранним сроком завершения конечных операций, которым она предшествует. Таким образом, если $i \prec k (k \in P_k)$, но $i \prec\prec l$, $n + l \in V_k$, то $R_i = \min\limits_{n+l\in V_k} (t_k - C_{i,n+l})$.

Для иллюстрации описанного выше алгоритма построения сетевого графика и получения его числовых характеристик рассмотрим следующий пример. Информация о комплексе из 10 операций задана табл. 2.

Таблица 2

Номер операции i	1	2	3	4	5	6	7	8	9	10
Каким операциям предшествует υ_i	2	3	–	7,10	6,8	7,10	9	9	–	3
Продолжительность t_i	2	2	3	4	1	4	4	3	2	9

Выполнив тернарные операции (1) по всем элементам $k = 1, 2, ..., 2n$ начальной матрицы дуговых весов, полученной описанным выше способом, и осуществив одновременно операции (2) над элементами матрицы $R = \|r_{ij}\|_{2n\times 2n}$, $r_{ij} = j$, $i = \overline{1, 2n}$, $j = \overline{1, 2n}$, получим:

$$C = \begin{array}{c|cccccccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
\hline
1 & & 2 & 4 & & & & & & & & 2 & 4 & 7 & & & & & & & \\
2 & & & 2 & & & & & & & & & 2 & 5 & & & & & & & \\
3 & & & & & & & & & & & & & 3 & & & & & & & \\
4 & & & 13 & & & & 4 & & 8 & 4 & & & 16 & 4 & & & 8 & & 10 & 13 \\
5 & & & 14 & & & 1 & 5 & 1 & 9 & 5 & & & 17 & & 1 & 5 & 9 & 4 & 11 & 14 \\
6 & & & 13 & & & & 4 & & 8 & 4 & & & 16 & & & 4 & 8 & & 10 & 13 \\
7 & & & & & & & & & 4 & & & & & & & & 4 & & 6 & \\
8 & & & & & & & & & 3 & & & & & & & & & 3 & 5 & \\
9 & & & & & & & & & & & & & & & & & & & 2 & \\
10 & & & 9 & & & & & & & & & & & & & & & & & 9 \\
\hline
11 & & 0 & 2 & & & & & & & & & 2 & 5 & & & & & & & \\
12 & & & 0 & & & & & & & & & & 3 & & & & & & & \\
13 & & & & & & & & & & & & & & & & & & & & \\
14 & & & 9 & & & & 0 & & 4 & 0 & & & 12 & & & & 4 & & 6 & 9 \\
15 & & & 13 & & & 0 & 4 & 0 & 8 & 4 & & & 16 & & & 4 & 8 & 3 & 10 & 13 \\
16 & & & 9 & & & & 0 & & 4 & 0 & & & 12 & & & & 4 & & 6 & 9 \\
17 & & & & & & & & & 0 & & & & & & & & & & 2 & \\
18 & & & & & & & & & 0 & & & & & & & & & & 2 & \\
19 & & & & & & & & & & & & & & & & & & & & \\
20 & & & 0 & & & & & & & & & & 3 & & & & & & &
\end{array};$$

$$R = \begin{array}{c|cccccccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
\hline
1 & & 11 & 11 & & & & & & & & & & & 11 & & & & & & \\
2 & & & 12 & & & & & & & & & & & 12 & & & & & & \\
3 & & & & & & & & & & & & & & & & & & & & \\
4 & & & 14 & & & & 14 & & 14 & 14 & & & & 14 & & & 14 & & 14 & 14 \\
5 & & & 15 & & & 15 & 15 & 15 & 15 & 15 & & & & 15 & & & 15 & 15 & 15 & 15 \\
6 & & & 16 & & & & 16 & & 16 & 16 & & & & 16 & & & 16 & & 16 & 16 \\
7 & & & & & & & & & 17 & & & & & & & & & & 17 & \\
8 & & & & & & & & & 18 & & & & & & & & & & 18 & \\
9 & & & & & & & & & & & & & & & & & & & & \\
10 & & & 20 & & & & & & & & & & & 20 & & & & & & \\
\hline
11 & & & 2 & & & & & & & & & & 2 & 2 & & & & & & \\
12 & & & & & & & & & & & & & & 3 & & & & & & \\
13 & & & & & & & & & & & & & & & & & & & & \\
14 & & & 10 & & & & & & 7 & & & & & 10 & & & 7 & & 7 & 10 \\
15 & & & 6 & & & & 6 & & 6 & 6 & & & & 6 & & 6 & 6 & 8 & 6 & 6 \\
16 & & & 10 & & & & & & 7 & & & & & 10 & & & 7 & & 7 & 10 \\
17 & & & & & & & & & & & & & & & & & & & 9 & \\
18 & & & & & & & & & & & & & & & & & & & 9 & \\
19 & & & & & & & & & & & & & & & & & & & & \\
20 & & & & & & & & & & & & & & 3 & & & & & &
\end{array}.$$

Здесь пробелы в матрице C означают, что соответствующий ее элемент $c_{ij} = \infty$, а в матрице R – это $r_{ij} = j$.

Начальные операции $V_0 = \{1, 4, 5\}$, конечные $V_k = \{3, 9\}$. Ранние сроки начала выполнения операций:

$$T_1^{\text{рн}} = T_4^{\text{рн}} = T_5^{\text{рн}} = 0,\ T_2^{\text{рн}} = 2,\ T_3^{\text{рн}} = 14;$$
$$T_6^{\text{рн}} = 1,\ T_7^{\text{рн}} = 5,\ T_8^{\text{рн}} = 1,\ T_9^{\text{рн}} = 9,\ T_{10}^{\text{рн}} = 5,$$

соответственно ранние сроки окончания всех операций:

$$T_1^{\text{ро}} = 2;\quad T_2^{\text{ро}} = 4;\quad T_3^{\text{ро}} = 17;$$
$$T_4^{\text{ро}} = 4;\quad T_5^{\text{ро}} = 1;\quad T_6^{\text{ро}} = 5;\quad T_7^{\text{ро}} = 9;$$
$$T_8^{\text{ро}} = 4;\quad T_9^{\text{ро}} = 11;\quad T_{10}^{\text{ро}} = 14.$$

Критическое время $t_k = \max\{C_{5,13}, C_{5,19}\} = C_{5,13} = 17$.

Критические операции определяются с помощью матрицы R

$$r_{15,3} = 6;\quad r_{16,3} = 10;\quad r_{20,3} = 3.$$

Таким образом, полученные в результате осуществления тернарных операций (1) и (2) матрицы C и R позволяют определить все параметры сетевого графика.

В Ы В О Д

Разработан новый эффективный алгоритм отыскания критических путей в графе, не использующий графического представления последнего, а основанный на введенном понятии осуществления тернарных операций над матрицей дуговых весов.

Полученный алгоритм может использоваться для решения широкого круга задач, допускающих формулировку в терминах теории графов. Рассмотрены его приложения для решения задач упорядочения элементов конечного множества и задач сетевого планирования.

Л И Т Е Р А Т У Р А

1. **Форд, Л. Р.** Потоки в сетях / Л. Р. Форд, Д. Р. Фалкерсон. – М.: Мир, 1966. – 276 с.

2. **Конвей, Р. В.** Теория расписаний / Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер. – М.: Наука, 1975. – 360 с.

3. **Floyd, R. W.** Algorithm 97: Shortest Path / R. W. Floyd // Communication of ACM. – 1962. – № 5(6). – 345 p.

4. **Корзников, А. Д.** Тернарные операции в задачах оптимального преобразования сети / А. Д. Корзников // Современные прикладные задачи и технологии обучения в математике и информатике: сб. науч. ст. – Минск, 2004. – С. 92–98.

5. **Корзников, А. Д.** Оптимизация системы маршрутов в транспортных сетях / А. Д. Корзников // Проблема прогнозирования и государственного регулирования социально-экономического развития. – Минск, 2004. – С. 46–49.

Поступила 6.11.2007

УДК 532.135:532.5.013.4

О РЕАЛИЗАЦИИ МЕТОДА КОРРЕКЦИИ ДАВЛЕНИЯ ПРИ РЕШЕНИИ УРАВНЕНИЙ ГИДРОДИНАМИКИ НЕСЖИМАЕМОЙ ЖИДКОСТИ

МАКАРОВ И. А.

ЗАО «БелХард»

Во многих исследований в области вычислительной гидродинамики значительную часть составляют работы, рассматривающие модель несжимаемой жидкости. Причина этого – многообразие технологических процессов, в которых применима данная модель, в частности при добыче нефти в Республике Беларусь, в том числе нефти с аномальной вязкостью