

УДК 004.272.2(045)

3973.002-02

Мартынова О. П. канд. техн. наук

## ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ МАРШРУТИЗАЦИИ НА ГРАФАХ И СЕТЯХ

Институт компьютерных технологий Национального авиационного университета

*Описан параллельный алгоритм маршрутизации на графах с положительными и/или отрицательными весами ветвей. Показано, что параллельный алгоритм обладает линейными оценками его временной и емкостной сложности.*

### Постановка проблемы

В настоящее время интенсивный рост сложности компьютерных сетей требует разработки параллельных алгоритмов маршрутизации для определения маршрутов передачи данных в сети, которые характеризуются максимальной пропускной способностью либо минимальным временем задержки. Математическая модель компьютерной сети может быть представлена в виде графа, вершины которого моделируют узлы-источники и узлы-приемники информации, а ветви графа соответствуют возможным каналам передачи данных. Ветвям графа присваивают вес пропорциональный пропускной способности канала передачи данных либо пропорциональный временной задержке в процессе передачи данных по этому каналу. В случае обеспечения максимальной пропускной способности необходимо решать задачу о длиннейшем пути между узлом-источником и узлом-приемником информации. Когда требуется минимизировать время задержки передачи данных необходимо решать задачу о кратчайшем пути между узлом-источником и узлом-приемником информации. Решение задачи о длиннейшем или кратчайшем пути на графе зависит от режима работы компьютерной сети. Задача о длиннейшем пути может решаться как задача о кратчайшем пути в ациклической сети с отрицательными весами всех ветвей. В связи с этим требуется разработка параллельного алгоритма о кратчайшем пути на графе с положительными и/или отрицательными весами ветвей, который характеризуется наилучшими оценками его временной и емкостной сложности.

### Анализ последних исследований и публикаций [1, 2, 7, 11]

позволяет сделать вывод о том, что задача маршрутизации решается только как задача о кратчайших путях на графе с положительным весом ветвей. Гибкий переход от решения задачи о кратчайшем пути к решению задачи о длиннейшем пути, а также решение задач о кратчайшем пути с положительными и отрицательными весами ветвей параллельные методы нахождения кратчайших маршрутов в сетях не обеспечивают.

**Цель работы** – разработка параллельного алгоритма о кратчайшем пути на графе с положительными и/или отрицательными весами ветвей.

Моделирование графов, как с положительным, так и с отрицательным весом имеет существенные особенности в случае применения принципа временной аналогии, когда вес ветви графа представляется временной задержкой сигнала в модели ветви. Моделирование отрицательного веса требует вместо временной задержки сигнала реализовать ускорение сигнала в данной ветви графа по отношению ко всем остальным ветвям. Реализация такой функции на моделируемом графе достаточно сложна и для графов значительных размеров практически нецелесообразна.

Рассмотрим процесс моделирования графа с положительным и отрицательным весом, когда граф можно представить в виде множества фрагментов, образованных пучком из  $m$  направленных ветвей, входящих в одну вершину (рис. 1) [1].

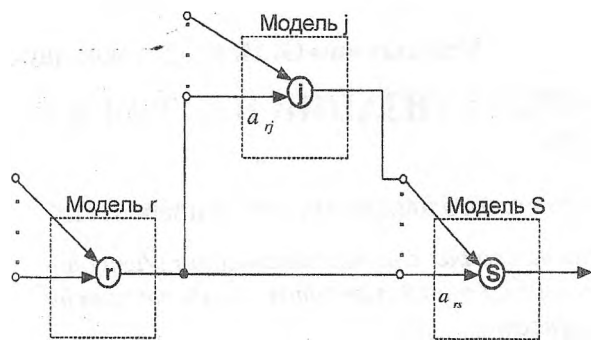


Рис. 1. Граф, представленный в виде множества фрагментов

Каждой  $l$ -й ветви  $S$ -го фрагмента графа сопоставим вес  $a_{ls}$ ,  $l=1,2,\dots,m$ , который может принимать отрицательное или положительное значение. Фрагменты графа соединяют между собой в соответствии с топологией моделируемого графа.

Присвоим  $S$ -тому узлу фрагмента графа вес

$$C_s(t_i + \Delta t) = \min_l [a_{ls} + b_{rls}(t)] \quad (1.1)$$

$$l=1, 2, \dots, m,$$

где  $b_{rls}$  – вес  $r$ -го узла фрагмента графа, соединенного с  $l$ -ой ветвью  $s$ -го фрагмента. Топология графа определяет схему соединения его фрагментов, которую можно описать соотношениями

$$C_r(t_i) = b_{rls}(t_i) \quad (1.2)$$

Полагаем, что моделируются графы, в которых отсутствуют циклы отрицательного веса, т.е. сумма весов ветвей по любому направленному циклу должна быть неотрицательной. Модель графа может находиться в двух состояниях: в начальном при  $t \leq t_0$  и в режиме моделирования при  $t > 0$ . Вес всех вершин графа формируется одновременно согласно параллельному алгоритму:

$$C_1 = 0; \quad C_s(t_i + \Delta t) = \begin{cases} \infty, & t \leq t_0; \\ \min_l [a_{ls} + b_{rls}(t)]; & S = 2, 3, \dots, q \end{cases} \quad (1.3)$$

где  $t > t_0$ ;  $q$  – количество вершин моделируемого графа,  $C_1$  – вес начальной вершины графа. Если фрагменты графа соединить согласно требуемой топологии (1.2) и в каждой модели, моделирующей фрагмент графа, реализовать алгоритм (1.3), то в модели графа в момент времени  $T$  достигается стационарное состояние, когда на выходах всех моделей установятся постоянные веса  $C_j(T) = C_j(T + \Delta t)$ ,  $j =$

$= 1, 2, \dots, q$ , где  $q$  – количество моделей,  $\Delta t$  – время реализации одного шага вычислений. В стационарном состоянии модели графа установившееся значение веса  $C_j(T)$  на выходе  $j$ -ой модели соответствует стоимости экстремального пути из начального узла  $C_1$  в  $j$ -тый узел, где  $j = 2, 3, \dots, q$ . Стоимость экстремального пути  $C_j(T)$  определяется минимальной алгебраической суммой весов ветвей графа, составленной вдоль всех допустимых путей из начального узла в  $j$ -тый узел.

В случае положительного веса всех ветвей моделируемого графа в стационарном состоянии значение веса  $C_j(T)$  на выходе  $j$ -ой модели равно длине кратчайшего пути из начального узла в  $j$ -тый, если вес ветвей графа равен длине пути из одного узла в другой.

Задача о длиннейшем пути решается как задача о кратчайшем пути в ациклической сети с отрицательными весами всех ветвей моделируемого графа. В этом случае в стационарном состоянии абсолютная величина веса  $C_j(T)$  на выходе  $j$ -ой модели равна длиннейшему пути из начального в  $j$ -тый, если абсолютная величина веса ветвей графа равна длине пути из одного узла в другой.

Покажем, что алгоритм (1.1) соответствует операции треугольника, на которой основан алгоритм Флойда [12], допускающий отрицательный вес ветвей графа. На рис. 1 изображена моделирующая структура из трех моделей. Операцией треугольника [12] для фиксированного узла  $j$  называется операция

$$C_s = \min \{a_{rs}, a_j + a_{js}\}, \quad (1.4)$$

для всех  $r, s = 1, 2, \dots, q$ , таких, что  $r, s \neq j$ . Эта операция для каждой пары  $r$  и  $s$  заменяет элемент  $a_{rs}$  алгебраической суммой  $a_{rj} + a_{js}$ , если она меньше  $a_{rs}$ .

В работе [12] по индукции доказано, что после последовательного выполнения операции треугольника для значений  $j = 1, 2, \dots, q$  каждый элемент  $C_s$  становится равным длине кратчайшего пути из  $r$  в  $s$  при положительном весе ветвей графа. Это утверждение обобщается на случай отрицательного веса ветвей, если граф не содержит циклов отрицательного веса.

Операция треугольника (1.4) является частным случаем алгоритма (1.1), ко-

торый для графа, изображенного на рис. 1 принимает вид

$$C_s = \min\{a_{rs} + C_r; a_{js} + a_{rj} + C_r\} \quad (1.5)$$

Выражение (1.4) следует из (1.5) при  $C_r = 0$ .

Алгоритм Флойда, как и другие известные последовательные алгоритмы [3,4] в процессе распараллеливания имеют временную сложность  $O(\log^2 q)$  при емкостной сложности  $O(q^3)$ , где  $q$  – количество вершин моделируемого графа. Следовательно, сокращение времени решения задачи о кратчайшем пути достигается при распараллеливании известных последовательных алгоритмов ценой полиномиального роста емкостной сложности (количества процессоров, моделирующих граф) [3, 4-6].

Покажем, что параллельный алгоритм (1.3) имеет временную сложность  $O(L)$ , где  $L$  – количество элементов графа вдоль кратчайшего пути и линейный рост емкостной сложности  $O(q)$ , где  $q$  – количество вершин моделируемого графа.

**Утверждение 1.** Параллельное выполнение операций (1.3) во всех вершинах графа, не содержащего циклов отрицательного веса, за конечное время устанавливает в вершинах графа стационарное значение веса, равное стоимости экстремального пути в эти вершины.

Факт достижения параллельным процессом выполнения операций (1.3) стационарного состояния устанавливается проверкой выполнения для всех узлов графа условий:

$$C_j(T) = C_j(T + \Delta t);$$

$$j = 1, 2, \dots, q, \quad (1.6)$$

где  $C_j$  – вес  $j$ -го узла графа,  $q$  – количество узлов графа,  $T$  – момент времени достижения стационарного состояния,  $\Delta t$  – время выполнения одного шага параллельных вычислений (1.3) во всех узлах графа.

Пусть дан граф, изображенный на рис. 1, где  $r$ -й узел является начальным и имеет вес  $C_r = 0$ . Если одновременно выполнить операции (1.3) для узлов  $j$  и  $s$ , то получим

$$C_j(t_1) = a_{rj}; C_s(t_1) = \min\{a_{js} + C_j(t_0); a_{rs}\} = a_{rs} \quad (1.7)$$

Так как в начальном состоянии  $C_j(t_0) = C_s(t_0) = \infty$ , а в результате выполнения операции (1.3) имеем согласно (1.7)  $C_j(t_1) = a_{rj}; C_s(t_1) = a_{rs}$ , то условия стационарного состояния (1.6) не выполняются

$$C_j(t_0) \neq C_j(t_1); C_s(t_0) \neq C_s(t_1),$$

где  $t_1 = t_0 + \Delta t$ .

Выполняем вновь одновременно для узлов  $j$  и  $s$  операции (1.3), после завершения которых, вес узлов определяется выражениями

$$C_j(t_r) = a_{rj};$$

$$C_s(t_r) = \min\{a_{rs}, a_{rj} + a_{js}\} \quad (1.8)$$

Вторая операция в выражении (1.8) соответствует операции треугольника (1.4) и дает следующий результат

$$C_s(t_2) = \begin{cases} a_{rs}, \text{ при } a_{rs} \leq a_{rj} + a_{js}, \\ a_{rj} + a_{js}, \text{ при } a_{rs} \geq a_{rj} + a_{js}. \end{cases} \quad (1.9)$$

Анализ выражений (1.7) и (1.9) показывает, что в случае  $C_s(t_1) = a_{rs}$  и  $C_s(t_2) = a_{rs}$  достигается стационарное состояние (1.6), так как  $C_j(t_2) = C_j(t_1) = a_{rj}$  и  $C_s(t_2) = C_s(t_1) = a_{rs}$ . Если согласно (1.9) имеет место случай, когда  $C_s(t_2) = a_{rj} + a_{js}$ , то  $C_s(t_2) \neq C_s(t_1)$  и необходимо выполнить следующий шаг вычислений, после которого достигается стационарное состояние по всем узлам графа

$$C_j(t_s) = C_j(t_2) = a_{rj};$$

$$C_s(t_3) = C_s(t_2) = a_{rj} + a_{js},$$

где  $t_2 = T_1, t_3 = T + \Delta t$ .

В стационарном состоянии установившееся значение веса узлов  $j$  и  $s$  графа определяется минимальной алгебраической суммой весов ветвей графа, составленной вдоль всех допустимых путей из начального  $r$ -того узла в узлы  $j$  и  $s$ . Минимальная алгебраическая сумма весов ветвей графа вдоль экстремального пути определяет его стоимость. Сформулированное утверждение для рассматриваемого случая справедливо.

Теперь рассмотрим произвольную часть исследуемого графа, изображенную на рис. 1, при условии, что вес  $r$ -ого узла графа  $C_r(t_i) \neq 0$  определяется к моменту

времени  $t_i$  стоимостью экстремального пути между начальным узлом и  $r$ -м узлом. Параллельное выполнение операций (1.3) одновременно для узлов  $j$  и  $s$  дает после первого шага вычислений следующий результат

$$C_j(t_i + \Delta t) = a_{rj} + C_r(t_i); C_s(t_i + \Delta t) = \min\{a_{js} + C_j(t_i); a_{rs} + C_r(t_i)\} \quad (1.10)$$

Так как  $C_j(t_i + \Delta t) \neq C_j(t_i)$ , то выполняем второй шаг вычислений согласно (1.3) после которого получим

$$\begin{aligned} C_j(t_i + 2\Delta t) &= a_{rj} + C_r(t_i + \Delta t) = \\ &= a_{rj} + C_r(t_i); \\ C_s(t_i + 2\Delta t) &= \\ &= \begin{cases} a_{js} + a_{rj} + C_r(t_i), a_{js} + a_{rj} \leq a_{rs} \\ a_{rs} + C_r(t_i), a_{rs} \leq a_{js} + a_{rj}. \end{cases} \quad (1.11) \end{aligned}$$

Выполнение последующих шагов вычислений не приводит к изменению значений  $C_r, C_j, C_s$ , что свидетельствует о достижении стационарного состояния (1.6). Так как из узла  $r$  в узел  $j$  ведет единственный путь, а  $C_r$  – стоимость экстремального пути между начальным и  $r$ -м узлом, то  $C_j(t_i + 2\Delta t)$  – стоимость между начальным и  $j$ -м узлом. Выражение (1.11) соответствует операции треугольника (1.4) и, следовательно,  $C_s(t_i + 2\Delta t)$  – стоимость между начальным и  $s$ -м узлом. Таким образом, в стационарном состоянии (1.6) веса узлов при выполнении операции (1.3), равны стоимости экстремального пути из начального узла во все остальные узлы графа, что доказывается по индукции. Предположим противное, в стационарном состоянии (1.6) значение  $C_r(t_i)$  в  $r$ -м узле больше стоимости  $\min C_k(t_i)$  экстремального пути из начального в  $r$ -й узел. В этом случае найдется  $k$ -й узел, соединенный направленной ветвью с  $r$ -м узлом, вес которого больше стоимости  $\min C_k(t_i)$  экстремального пути из начального в  $k$ -й узел и так далее до начального узла. В результате имеем ситуацию, изображенную на рис. 1, когда  $r$ -й узел является начальным, и которая уже рассматривалась выше. В этой ситуации, если значение  $C_s(t_i)$  в  $s$ -м узле больше стоимости  $\min C_s(t_i)$ , то после выполнения операции (1.3) в следующем шаге произойдет изменение стоимости  $C_s(t_i + \Delta t) < C_s(t_i)$ , что противоречит условию достижения стационарного состояния (1.6).

Характеристика быстродействия алгоритма (1.3) устанавливается следующим утверждением.

**Утверждение 2.** После  $L$  – шагов параллельного выполнения операций (1.3) во всех вершинах графа, не содержащего циклы отрицательного веса, достигается стационарное состояние (1.6), где  $L$  – количество вершин в экстремальном пути с максимальным количеством элементов.

Докажем сначала, что для достижения стационарного состояния (1.6) требуется не менее  $L$  – шагов выполнения операций (1.3) во всех узлах графа. Начнем рассмотрение с начального узла графа, который обозначим первым номером. Присвоим узлам номера от 1 до  $L$  в порядке следования узлов вдоль экстремального пути с максимальным количеством узлов от первого узла, соединенного с начальным узлом, до последнего узла в экстремальном пути. Анализ алгоритма (1.1)-(1.3) на фрагменте графа, изображенном на рис. 1, показывает, что, если  $r$ -й узел является начальным узлом с номером №1 и весом  $C_r = 0$ , а экстремальный путь с наибольшим количеством элементов проходит через  $s$ -й узел с номером №2, то на его выходе после первого шага вычислений установится стоимость экстремального пути, равная  $C_r = a_{12}$ . Если теперь предположим, что  $r$ -й узел является узлом с №2 и  $s$ -й узел – узлом экстремального пути с №3 с весом после первого шага превышающим минимальную стоимость экстремального пути в этот узел, то после второго шага вычислений (1.1) вес узла №3 станет равным стоимости экстремального пути в этот узел  $C_3 = a_{12} + a_{23}$ . Этот процесс распространяется последовательно вдоль экстремального пути с максимальным количеством узлов до тех пор, пока на  $L-1$  шаге вес последнего узла № $L$  не станет равным стоимости всего экстремального пути  $C_L = a_{12} + a_{23} + \dots + a_{L-1,L}$ . Для проверки условий достижения стационарного состояния необходимо сделать еще один  $L$ -й шаг вычислений. Таким образом, последовательный процесс установления стационарного состояния в узлах графа вдоль экстремального пути от начального узла в  $L$ -й требует не менее  $L$  шагов параллельного выполнения операций (1.6).

Теперь необходимо доказать, что для достижения стационарного состояния (1.6) требуется не более  $L$  - шагов параллельного выполнения операций (1.3).

Предположим, что стационарное состояние (1.6) в последнем  $L$  - м узле экстремального пути, имеющего наибольшее количество элементов, устанавливается после  $L + 1$ -го шага вычислений. Тогда стационарное состояние в предпоследнем  $L - 1$  - м узле установится после  $L$  - шагов и т.д. до тех пор, пока не придем к заключению о том, что в узле №2, связанным с начальным узлом №1 стационарное состояние установится после трех шагов. Эту ситуацию рассмотрим на рис.1 в предположении, что  $r$ -й узел является начальным, а  $s$ -й узел имеет №2 и принадлежит экстремальному пути с максимальным количеством элементов. Такой случай анализировался по выражениям (1.7)-(1.9), а результаты анализа показали, что стационарное состояние в узле установится после двух шагов, если экстремальный путь непосредственно связывает этот узел с начальным. Таким образом, имеем ситуацию, противоречащую сделанному допущению. Для установления стационарного состояния в узле №2 достаточно два шага вычислений, а если стационарное состояние в  $L$  - м узле установится после  $L + 1$ -го шага, то стационарное состояние устанавливается после трех шагов. Но стационарное состояние установится после трех шагов только тогда, когда между начальным узлом №1 и узлом №2 будет существовать дополнительный  $L + 1$ -й узел, что противоречит предположению о  $L$  узлах в экстремальном пути с максимальным количеством элементов. Приведенные рассуждения справедливы для всего дерева экстремальных путей графа, но так как экстремальный путь с максимальным количеством элементов имеет максимальное число  $L$ , то количество шагов для параллельного построения всего дерева экстремальных путей на графе с корнем в начальном узле определяется этим числом  $L$ .

Ветви дерева экстремальных путей определяются процессом распространения установившихся состояний от на-

чального во все остальные узлы графа. Если, например, в  $i$ -м узле вес  $C_i$  принял стационарное значение после установления стационарного состояния в предыдущем  $i - 1$ -м узле, связанным непосредственно с  $i$ -м узлом, то ветвь, соединяющая  $i - 1$ -й узел с  $i$ -м узлом, принадлежит дереву экстремальных путей. Объединив формулировки утверждений 1 и 2, получим следующее утверждение.

**Утверждение 3.** Параллельное выполнение операций (1.3) во всех вершинах графа, не содержащего циклов отрицательного веса, за  $L$  - шагов устанавливает в вершинах графа стационарное значение веса, равное стоимости экстремального пути в эти вершины, где  $L$  - количество вершин в экстремальном пути с максимальным количеством элементов. Из утверждения 3 следует, что алгоритм параллельного выполнения операций (1.3) является детерминированным и выполняется за  $L$  - шагов, где  $L$  - количество вершин в экстремальном пути с максимальным количеством элементов. Детерминированный характер предложенного алгоритма является существенным признаком, отличающим его от параллельного недетерминированного алгоритма, который работает бесконечно долго, так как в нем отсутствует ограничитель цикла локальной оптимизации выбранной вершины графа.

Утверждение 3.3 является следствием утверждений 3.1. и 3.2, которые справедливы в случае отсутствия в графах циклов отрицательного веса.

Если в графе содержится цикл отрицательного веса, то в результате работы алгоритма (1.3) веса вершин в цикле непрерывно уменьшаются, и стационарное состояние (1.6) не достигается.

На рис. 2 изображен цикл отрицательного веса.

Пусть узел 1 будет начальным с весом  $C_1(t_0) = 0$ , а начальное значение веса узлов 2 и 3 соответствует  $+\infty$ . Параллельная реализация алгоритма (1.11) — (1.13) на графе рис. 2 дает следующие результаты:

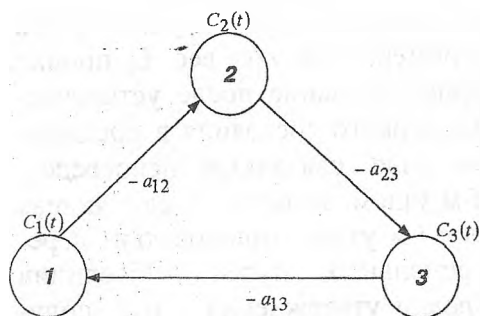


Рис. 2. Цикл отрицательного веса

$$C_2(t_1) = -a_{12},$$

$$C_3(t_2) = -a_{12} - a_{23},$$

$$C_1(t_3) = -(a_{12} + a_{23} + a_{13}),$$

$$C_2(t_4) = -2a_{12} - a_{23} - a_{13},$$

$$C_3(t_5) = -2a_{12} - 2a_{23} - a_{13},$$

$$\dots\dots\dots$$

$$C_1(t_v) = -k(a_{12} + a_{23} + a_{13}),$$

где  $v = 3k$  — количество шагов вычислений. Наличие цикла отрицательного веса приводит к тому, что при  $K \rightarrow \infty$  вес узлов  $C_1(t) \rightarrow -\infty$  и стационарное состояние (1.6) не достигается. Следовательно, если в результате применения алгоритма (1.1)-(1.3) стационарное состояние (1.6) не достигается за  $q$  шагов, где  $q$  — количество узлов графа, то моделируемый граф содержит циклы с отрицательным весом. Это свойство предложенного алгоритма имеет важное значение для приложений, так как позволяет установить факт существования циклов отрицательного веса в моделируемом графе. Установим зависимость времени реализации предложенного алгоритма (1.1)-(1.3) от размерности графа, которая оценивается числом его вершин  $q$ .

Из утверждения 2 следует, что время решения задачи об экстремальном пути предложенным алгоритмом (1.1)-(1.3) составляет

$$T = L\Delta t, \quad (1.12)$$

где  $L$  — количество узлов экстремального пути с максимальным количеством элементов,  $\Delta t$  — время выполнения одного шага вычислений (1.1) одновременно во всех узлах графа.

Временная сложность предложенного алгоритма согласно (1.12) составляет  $O(L)$ . В случае положительных весов временная сложность последовательного алгоритма Дейкстры [8] составляет  $O(q^2)$ , алгоритма Спирса  $O(q^2 \log_2 q)$ . Известный алгоритм Флойда [9,12] для графов с от-

рицательным весом, при условии отсутствия циклов с отрицательным весом, имеет оценку временной сложности порядка  $O(q^3)$ . Учитывая, что при  $q > 1$  имеет место  $L \leq q$  временная сложность предложенного алгоритма (1.1)-(1.3) составляет порядка  $O(q)$ . Емкостная сложность предложенного алгоритма оценивается в  $q$  параллельных вычислительных моделей, каждая из которых реализует алгоритм (1.3) и моделирует пучок ветвей графа, входящих в одну из его вершин, где  $q$  — количество вершин моделируемого графа.

В параллельных алгоритмах временная и емкостная сложности неразрывно связаны между собой [3-6]. Параллельные алгоритмы наименьшей временной сложности  $O(\log_2^2 q)$  имеют емкостную сложность порядка  $O(q^3)$ . Применение предложенного алгоритма позволяет вместо полиномиальной скорости роста емкостной сложности известных алгоритмов получить линейную зависимость скорости роста емкостной сложности от сложности моделируемого графа, оцениваемой количеством его вершин.

Решим задачу о кратчайшем пути [10] на графе с отрицательными весами, изображенном на рис. 3, параллельным алгоритмом (1.3).

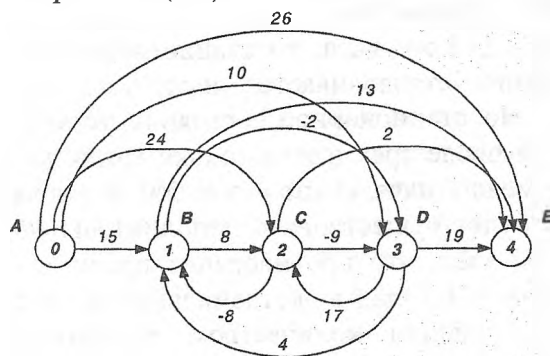
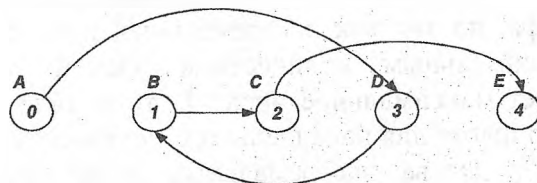


Рис. 3. Граф с отрицательными весами

Рис. 4. Кратчайший путь из вершины  $A_0$  в вершину  $E_4$ 

Результаты вычислений сведены в табл. 1. Кратчайший путь из вершины  $A_0$  в вершину  $E_4$  проходит через вершины  $A_0$

$D_3B_1C_2E_4$  и изображен на рис. 4. Последняя строка табл. 1, которая на пятом шаге вычислений повторяет результаты четвертого шага, дает значения стоимостей экстремального пути в каждую вершину графа, так как достигнуто стационарное

состояние (1.6). Задача о кратчайшем пути на графе с пятью вершинами решена предложенным параллельным алгоритмом за пять шагов, т.е. за количество шагов, равное количеству вершин графа вдоль кратчайшего пути  $A_0D_3B_1C_2E_4$ .

Таблица 1.

Вершины графа		Результаты вычислений				
		$A_0(C_1)$	$B_1(C_2)$	$C_2^{BT.}(C_3)$	$D_3(C_4)$	$E_4(C_5)$
№ шага		0	$\min \begin{cases} C_1 + 15 \\ C_3 - 8 \\ C_4 + 4 \end{cases}$	$\min \begin{cases} C_1 + 24 \\ C_2 + 8 \\ C_4 + 17 \end{cases}$	$\min \begin{cases} C_1 + 10 \\ C_2 - 2 \\ C_3 - 9 \end{cases}$	$\min \begin{cases} C_1 + 26 \\ C_2 + 13 \\ C_4 + 19 \end{cases}$
	0	0	$\infty$	$\infty$	$\infty$	$\infty$
	1	0	15	24	10	26
	2	0	14	23	10	26
	3	0	14	22	10	25
	4	0	14	22	10	24
	5	0	14	22	10	24

### Вывод

Предложенный параллельный алгоритм решения задачи маршрутизации в сети обеспечивает решение задачи о кратчайшем и/или длиннейшем пути на графе с положительными и/или отрицательными весами ветвей. Временная сложность параллельного алгоритма имеет оценку  $O(L)$ , где  $L$  – количество элементов графа вдоль кратчайшего пути. Емкостная сложность предложенного алгоритма имеет оценку  $O(q)$ , где  $q$  – количество вершин графа, моделирующей компьютерную сеть.

Перспективы дальнейших исследований связаны со схемотехническими способами реализации предложенного алгоритма.

### Список литературы

1. Жуков И.А., Мартинова О.П. Метод побудови паралельних структур для пошуку альтернативних маршрутів у комп'ютерних мережах // Вісник НАУ. – К.: НАУ, 2004. – № 1. – С. 14-17.
2. Мартинова О.П. Параллельные средства решения задачи альтернативной маршрутизации в компьютерных сетях // Інформаційно-діагностичні системи. – К.: НАУ, 2004. – Т.1. – С.13.49-13.52.
3. Иванов Е.А., Шевченко В.П. О параллельных вычислениях на графах // Кибернетика. – 1984. – № 3. – С. 89-94.
4. Анисимов А.В. Локальный алгоритм для задачи о кратчайшем пути из одного источника // Кибернетика. – 1986. – № 3. – С. 57-60.
5. Листровой С.В., Певнев В.Я. Вопросы построения параллельных вычислительных систем и параллельный алгоритм для решения задачи о кратчайшем пути // Электронное моделирование. – 1990. – №1. – С. 17-20.
6. Иванов Е.А. Параллельные алгоритмы на графах // Кибернетика. – 1981. – № 3. – С. 81-83.
7. Жуков И.А., Мартинова О.П. Системоаналоговое моделирование на графах параллельных маршрутов в сетях // Проблеми інформатизації та управління. – К.: НАУ, 2004. – Вип.10. – С. 40-45.
8. Dijkstra, W. "A Note on Two Problems in Connexion with Graphs", Numer. Math., vol. 1, pp. 269-271, Oct. 1959.
9. Ху Т. Целочисленное программирование и потоки в сетях. – М.: Мир, 1974. – 520с.
10. Вагнер Г. Основы исследования операций. – М.: Мир, 1973. – 330 с.
11. Жуков И.А., Мартинова О.П. Системоаналоговый метод багатопляхової маршрутизації на графах // Вісник Технологічного університету Поділля. – Хмельницький, 2004, Т.2(60), №2. – Ч.1. – С. 38-42.
12. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. – 512 с.