

УДК 681.32

А. А РЕВА

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

МЕТОД ПОИСКА АЛЬТЕРНАТИВНЫХ МАРШРУТОВ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ

Разработан метод поиска альтернативных маршрутов для сетей передачи данных, который базируется на алгоритме поиска всех или заданного числа кратчайших отличающихся хотя бы одним ребром маршрутов между парой вершин взвешенного графа. Рассматриваемая задача относится к классу задач структурной оптимизации теории графов. Разработанный метод имеет существенно меньшую оценку алгоритмической сложности. Рассмотрена работа алгоритма для поиска маршрутов отличающихся дугами на примере взвешенного графа с целочисленными неотрицательными весами дуг. На практике метод может быть использован для выбора маршрутов второго и третьего выбора при отказах узлов коммутации или каналов связи.

Ключевые слова: альтернативные маршруты, передача данных.

Введение

Для сложных распределенных систем одной из основных задач является обеспечение надежности, под которой понимается наличие связи между удаленными объектами. Для сетей связи требуется, чтобы между узлами существовало несколько независимых маршрутов передачи информации.

Для повышения надежности при передаче данных в сетях используются обходные и резервные маршруты [1]. Чтобы повысить эффективность оперативного управления необходимо в кратчайшие сроки определить новый оптимальный маршрут передачи данных при отказах узлов коммутации или каналов связи. На случай выхода из строя узлов и/или каналов связи наивысшего класса предварительно формируются обходные маршруты. Это позволяет оперативно перенаправить трафик без необходимости тратить временной ресурс на поиск обходных маршрутов.

Можно заранее сформировать множество альтернативных маршрутов, упорядоченное по их стоимости. Под стоимостью маршрута обычно понимается задержка передачи единичного объема информации между источником и потребителем.

Указанная задача [2] может быть решена в терминах теории графов путем последовательного перебора всех возможных топологий графа, возникающих при единичных отказах. Однако оценка сложности такого алгоритма будет равна $O(n)^5$. Предлагаемый алгоритм имеет существенно меньшую оценку сложности равную $O(Kn^2)$, где K – ко-

личество резервных маршрутов, а n – количество вершин исходного графа. Обычно, на практике, K не превышает трех: основной маршрут, маршрут второго и третьего выбора.

Постановка задачи

В зависимости от условия, определяющего, чем следующий маршрут должен отличаться от предыдущих, возникает несколько постановок задач:

1. Каждый следующий маршрут должен отличаться хотя бы одним ребром. В графе с R ребрами таких маршрутов может быть не более чем $(R-2)$.

2. Каждый последующий маршрут не должен содержать ребра предыдущих маршрутов (поиск независимых по ребрам маршрутов).

Данная статья посвящена решению первой задачи.

Задано:

– граф сети $G(V, E)$, $|V|=n$ – количество вершин графа;

– матрица весов ребер $D(n, n)=[d(i, j)]$;

– начальная v_s и конечная v_t вершины.

Найти:

N или все отличающиеся хотя бы одним ребром кратчайшие маршруты из v_s в v_t .

Для решения поставленной задачи используется алгоритм Дейкстры [3] – алгоритм поиска кратчайшего маршрута от заданной вершины ко всем остальным вершинам графа. Его сложность $O(n^2)$, где n – количество вершин графа.

Алгоритм поиска K или всех кратчайших маршрутов, отличающихся хотя бы одним ребром

1. Определение кратчайших путей из вершины v_s во все остальные v_i ($i \neq s, v_i \in V$) вершины графа $G(V, E)$ с помощью алгоритма Дейкстры. В результате получается дерево $T_s(V, E_s)$, каждая вершина v_i которого имеет пометку $|l_s(i)|$ - вес кратчайшего (s, i) - пути.

2. Нахождение кратчайших путей из всех вершин v_i ($i \neq t, v_i \in V$) до v_t . Для этого применяется алгоритм Дейкстры для графа, в котором матрица весов транспонируется

$$D' = D^T \text{ (} d'(i, j) = d(j, i); i = 1, |V|; j = 1, |V| \text{)}.$$

3. В получившемся дереве $T_t(V, E_t)$, каждой вершине v_i соответствует метка $|l_t(i)|$ - вес кратчайшего (i, t) - пути.

4. Заполнение матрицы R_{st} :

4.1. Инициализация элементов матрицы

$$R_{st}: r(i, j) = \infty \text{ (} i = 1, |V|; j = 1, |V| \text{)}.$$

4.2. Для каждого ребра (v_i, v_j) графа $G(V, E)$ вычисляется элемент матрицы R_{st} :

$$r(i, j) = |l_s(i)| + d(i, j) + |l_t(j)|$$

- длина кратчайшего маршрута из v_s в v_i , проходящего через ребро (v_i, v_j) .

5. Последовательное определение кратчайших маршрутов из v_s в v_t :

5.1. Результирующее множество отличающихся хотя бы одним ребром кратчайших маршрутов из вершины v_s в вершину v_t $M_{st} = \emptyset$

5.2. Поиск минимального элемента матрицы R_{st} $r_{\min} = r_{\min}(i_{\min}, j_{\min})$. Если нет отличных от ∞ элементов, то переход на п.6, иначе пометка $r_{\min}(i_{\min}, j_{\min}) = \infty$.

5.3. Определение кратчайшего маршрута из v_s в v_t , проходящего через $d(i_{\min}, j_{\min})$:

$$m_{i_{\min} j_{\min}} = l_s(i_{\min}) \cup d(i_{\min}, j_{\min}) \cup l_t(j_{\min}).$$

5.4. Если $M_{st} \cup m_{i_{\min} j_{\min}} = M_{st}$ (получившийся маршрут уже добавлен в результирующее множество), то переход на п. 5.2.

5.5. Если в получившемся маршруте имеются циклы (то есть хотя бы одна вершина встречается более одного раза), то переход на п. 5.2.

5.6. Кратчайший маршрут из v_s в v_t , проходя-

щий через $(v_{i_{\min}}, v_{j_{\min}})$ успешно добавлен в результирующее множество. При поиске N отличающихся хотя бы одним ребром кратчайших маршрутов проверяется дополнительное условие:

Если $|M_{st}| = N$, то переход на п. 5, иначе переход на п.4.2.

6. Конец работы алгоритма.

Вычислительную сложность алгоритма можно определить следующим образом. Двукратное применение алгоритма Дейкстры и поиск минимального элемента в матрице R_{st} $O(2n^2 + n^2) = O(3n^2)$, то есть вычислительная сложность остается квадратичной.

Рассмотрим применение алгоритма на примере.

Задан граф (рис. 1), необходимо найти все кратчайшие отличающиеся хотя бы одним ребром маршруты из вершины 1 в вершину 8.

Матрица $D(8,8)$:

	1	2	3	4	5	6	7	8
1	-	3	7	5	∞	∞	∞	∞
2	3	-	2	∞	∞	6	∞	∞
3	7	2	-	1	2	5	∞	∞
4	5	∞	1	-	∞	∞	10	∞
5	∞	∞	2	∞	-	5	4	3
6	∞	6	5	∞	5	-	∞	7
7	∞	∞	∞	10	4	∞	-	2
8	∞	∞	∞	∞	3	7	2	-

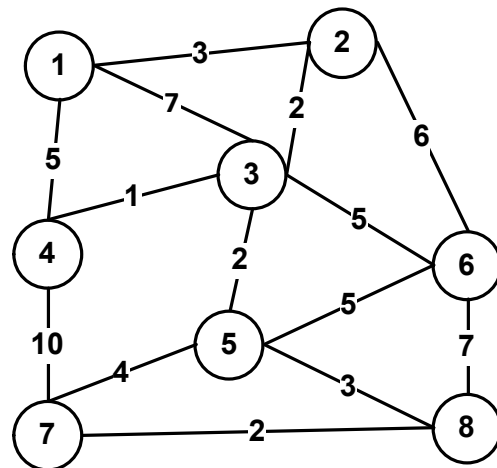


Рис. 1. Исходный граф

После первого шага алгоритма получим дерево $T_1(V, E_1)$ (рис. 2) – дерево кратчайших маршрутов от 1-й вершины ко всем остальным. Веса вершин равны

i	1	2	3	4	5	6	7	8
l ₁ (i)	0	3	5	5	7	9	11	10

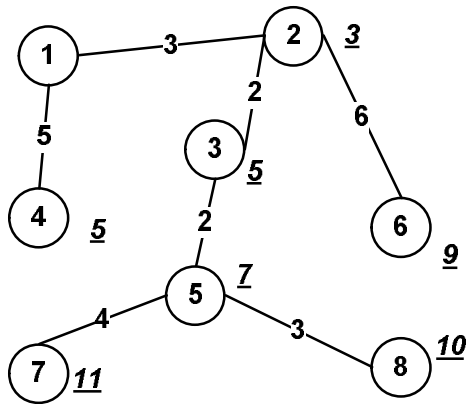


Рис. 2. Дерево T₁(V, E₁)

После второго шага алгоритма получим дерево T₈(V, E₈) (рис. 3) – дерево кратчайших маршрутов от всех вершин к вершине 8. Веса вершин равны

i	1	2	3	4	5	6	7	8
l ₈ (i)	10	7	5	6	3	7	2	0

После четвертого шага алгоритма получим матрицу R_{st} длин всех кратчайших маршрутов из v_s в v_t.

Матрица R_{st}:

	1	2	3	4	5	6	7	8
1	–	10	12	11	∞	∞	∞	∞
2	16	–	10	∞	∞	16	∞	∞
3	22	15	–	12	10	17	∞	∞
4	20	∞	11	–	∞	∞	17	∞
5	∞	∞	14	∞	–	19	13	10
6	∞	22	19	∞	17	–	∞	16
7	∞	∞	∞	27	18	∞	–	13
8	∞	∞	∞	∞	16	24	14	–

M₁₈=∅.

Минимальный элемент r(1,2)=10.

Маршрут от 1 вершины до 8, проходящий через дугу (1,2):

$$m_{12}=l_1(1) \cup d(1,2) \cup l_8(2)$$

$$m_{12}=(1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8).$$

$$M_{18} = M_{18} \cup m_{12}$$

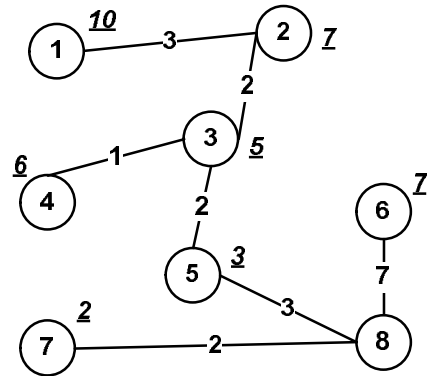


Рис. 3. Дерево T₈(V, E₈)

Первый получившийся кратчайший маршрут является глобально кратчайшим.

После исключения элемента r(1,2) (r(1,2)=∞), матрица R_{st} примет вид:

	1	2	3	4	5	6	7	8
1	–	∞	12	11	∞	∞	∞	∞
2	16	–	10	∞	∞	16	∞	∞
3	22	15	–	12	10	17	∞	∞
4	20	∞	11	–	∞	∞	17	∞
5	∞	∞	14	∞	–	19	13	10
6	∞	22	19	∞	17	–	∞	16
7	∞	∞	∞	27	18	∞	–	13
8	∞	∞	∞	∞	16	24	14	–

Следующий минимальный элемент r(2,3)=10.

$$m_{23}=l_1(2) \cup d(2,3) \cup l_8(3)$$

$$m_{23}=(1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8).$$

M₁₈ ∪ m₂₃= M₁₈, следовательно, маршрут m₂₃ - уже включен в M₁₈, поэтому он не добавляется в результирующее множество как новый маршрут.

Последующие итерации сведены в табл. 1. В первом столбце записаны текущие минимальные значения в матрице R_{st}. Во втором столбце записан сам маршрут. В столбце "Результат" символ "+" означает, что найденный маршрут добавляется в результирующее множество; "-" – маршрут не выбирается, так как он был ранее добавлен; "цикл" – не добавляется, потому что в маршруте имеются циклы.

Таким образом, получено результирующее множество маршрутов из вершины 1 в вершину 8, упорядоченное по не убыванию длин маршрутов:

Таблиця 1
Порядок просмотра дуг матрицы R_{st}

$r_{\min}(i,j)$	Маршрут	Результат
$r(3,5)=10$	1→2→3→5→8	–
$r(5,8)=10$	1→2→3→5→8	–
$r(1,4)=11$	1→4→3→5→8	+
$r(4,3)=11$	1→4→3→5→8	–
$r(1,3)=12$	1→3→5→8	+
$r(3,4)=12$	1→2→3→4→3→5→8	цикл
$r(5,7)=13$	1→2→3→5→7→8	+
$r(7,8)=13$	1→2→3→5→7→8	–
$r(8,7)=14$	1→2→3→5→8→7→8	цикл
$r(5,3)=14$	1→2→3→5→3→5→8	цикл
$r(3,2)=15$	1→2→3→2→3→5→8	цикл
$r(2,1)=16$	1→2→1→2→3→5→8	цикл
$r(2,6)=16$	1→2→6→8	+
$r(8,5)=16$	1→2→3→5→8→5→8	цикл
$r(6,8)=16$	1→2→6→8	–
$r(3,6)=17$	1→2→3→6→8	+
$r(4,7)=17$	1→4→7→8	+
$r(6,5)=17$	1→2→6→5→8	+
$r(7,5)=18$	1→2→3→5→7→5→8	цикл
$r(5,6)=19$	1→2→3→5→6→8	+
$r(6,3)=19$	1→2→6→3→5→8	+
$r(4,1)=20$	1→4→1→2→3→5→8	цикл
$r(3,1)=22$	1→2→3→1→2→3→5→8	цикл
$r(6,2)=22$	1→2→6→2→3→5→8	цикл
$r(8,6)=24$	1→2→3→5→8→6→8	цикл
$r(7,4)=27$	1→2→3→5→7→4→3→5→8	цикл

$$M_{18} = \{(1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8), (1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 8), (1 \rightarrow 3 \rightarrow 5 \rightarrow 8), (1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8), (1 \rightarrow 2 \rightarrow 6 \rightarrow 8), (1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 8), (1 \rightarrow 4 \rightarrow 7 \rightarrow 8), (1 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 8), (1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 8), (1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 5 \rightarrow 8)\}.$$

Количество найденных маршрутов равно десяти. Каждый из маршрутов содержит минимум одно новое ребро. Ни один из маршрутов не является суперпозицией других маршрутов.

Заключение

Предложенный метод предназначен для формирования множества альтернативных маршрутов, упорядоченных по их стоимости, и имеет квадратичную оценку сложности. Метод базируется на разработанном алгоритме поиска всех отличающихся хотя бы одним ребром кратчайших маршрутов.

Литература

1. Capone, A. Dynamic online QoS routing schemes: performance and bounds [Text] / A. Capone, L. Fratta, F. Martignon // Computer Networks. – 2006. – № 50. – P. 966–981.
2. Бертсекас, Д. Сети передачи данных. [Текст]: пер. с англ. / Д. Бертсекас, Р. Галлагер. – М.: Мир, 1989. – 544с.
3. Филлипс, Д. Методы анализа сетей. [Текст]: пер. с англ. / Д. Филлипс, А. Гарсиа-Диас. – М.: Мир, 1984. – 496 с.

Поступила в редакцию 20.01.2014, рассмотрена на редколлегии 12.02.2014

Рецензент: д-р техн. наук, профессор, заведующий кафедрой Е. А. Дружинин, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», г. Харьков.

МЕТОД ПОШУКУ АЛЬТЕРНАТИВНИХ МАРШРУТІВ У МЕРЕЖАХ ПЕРЕДАЧІ ДАНИХ

О. А. Рева

Розроблено метод пошуку альтернативних маршрутів для мереж передачі даних, який базується на алгоритмі пошуку всіх або заданого числа найкоротших відрізняючихся хоча б одним ребром маршрутів між парою вершин зваженого графа. Розглянута задача відноситься до класу задач структурної оптимізації теорії графів. Розроблений метод має істотно меншу оцінку алгоритмічної складності. Розглянуто роботу алгоритму для пошуку маршрутів, які відрізняються дугами на прикладі зваженого графа з цілочисельними невід'ємними вагами дуг. На практиці метод може бути використано для вибору маршрутів другого і третього вибору при відмовах вузлів комутації або каналів зв'язку.

Ключові слова: альтернативні маршрути, передача даних.

METHOD SEEKING ALTERNATIVE ROUTES IN DATA NETWORKS

О. А. Reva

Developed a method for finding alternative routes for data networks, which is based on an algorithm for finding all or a specified number of shortest differing by at least one edge of routes between a pair of vertices weighted graph. The problem under consideration belongs to the class of structural optimization problems in graph theory. This method has a much lower estimate of algorithmic complexity. Considered by the algorithm to find different routes arcs in Example weighted graph with nonnegative integer weights of arcs. In practice the method can be used to select the second and third route selection during the failure of switching nodes or links.

Keywords: alternative routes, the data transmission.

Рева Александр Анатольевич – канд. техн. наук, с.н.с., доцент кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков.