УДК 621.391

Р.Б. Трегубов, С.Н. Лазарев, С.Ю. Андреев

АЛГОРИТМ НАХОЖДЕНИЯ *K* МАКСИМАЛЬНЫХ ПОТОКОВ

Трегубов Р.Б., Лазарев С.Н., Андреев С.Ю. **Алгоритм нахождения *k* максимальных потоков.**

Аннотация. В работе предлагается оригинальный алгоритм решения прикладной задачи теории графов о нахождении *k* максимальных потоков между двумя заданными вершинами графа. Описываемый подход представляет собой комплексное применение в едином оптимизационном цикле алгоритма Форда-Фалкерсона (Эдмондса-Карпа или Диница) и алгоритма построения усеченного дерева состояний в ширину.

Ключевые слова: сеть связи с коммутацией пакетов, пропускная способность соединения, задача о максимальном потоке.

Tregubov R.B., Lazarev S. N., Andreev S. Ju. **The algorithm of finding of *k* maximum flows.**

Abstract. In the paper the original algorithm of solution of an applied task of the graph theory on finding of *k* maximum flows between the two set count's tops is proposed. The approach described represents a complex application of Ford-Fulkerson (Edmonds-Karp or Dinitz) algorithm and the algorithm of creation of a truncated tree of states in width in an indivisible optimizing cycle.

Keywords: a communication network with packet switching, connection capacity, the maximum flow task.

1. Введение. Управление трафиком в сети связи (СС) с коммутацией пакетов (КП) является одной из ключевых задач, решаемых в ходе обеспечения требуемого качества обслуживания пользователей. Под управлением трафиком далее понимается совокупность действий (процедур), реализованных в узлах коммутации пакетов (УКП) и направленных на максимально эффективное функционирование СС с КП при условии обеспечения требуемого качества обслуживания поступающих заявок.

В системе управления трафиком можно условно выделить три составляющие [1]: подсистему составления и заключения соглашения между пользователем и СС с КП по необходимому объему и качеству предоставляемых услуг (SLA); подсистему управления потоками пакетов; подсистему контроля параметров потоков пакетов.

Базовой для подсистемы управления потоками пакетов является функция управления доступностью соединения [1, 2]. На этапе установления нового виртуального соединения (ВС) в СС с КП данная функция заключается в определении, имеется ли в сети ресурс для обслуживания с требуемым качеством поступающего объема трафика. Новое ВС может быть поддержано СС с КП только в том случае, если имеется в наличии запрошенный ресурс, и при этом не снизится ниже требуемого значения качество обслуживания уже существующих ВС.

Для примера ниже рассматривается СС с КП со структурой, представленной на рисунке 1. Пусть требуется определить максимальную полосу пропускания ВС, в котором источник информации подключен к 1-му УКП, а приемник информации – к 5-му УКП. Логичным в данных условиях будет воспользоваться классическим решением задачи о максимальном потоке (минимальном разрезе) [3–11]. На сегодняшний день существует множество методов решения данной задачи: линейное программирование [11], алгоритм Форда-Фалкерсона [3], алгоритм Эдмондса-Карпа [4], алгоритм Диница [5], алгоритм проталкивания предпотока и алгоритм "поднять в начало" [4]. Кроме того в 2010 году команде ученых из разных научных заведений, состоящей из Джонатана Кельнера, Александра Мэдри, Поля Кристиано, Даниэля Спельмана и Шангуа Тенга, удалось найти принципиально новый подход, позволивший улучшить работу алгоритма поиска максимального потока впервые за 10 лет [6–8].

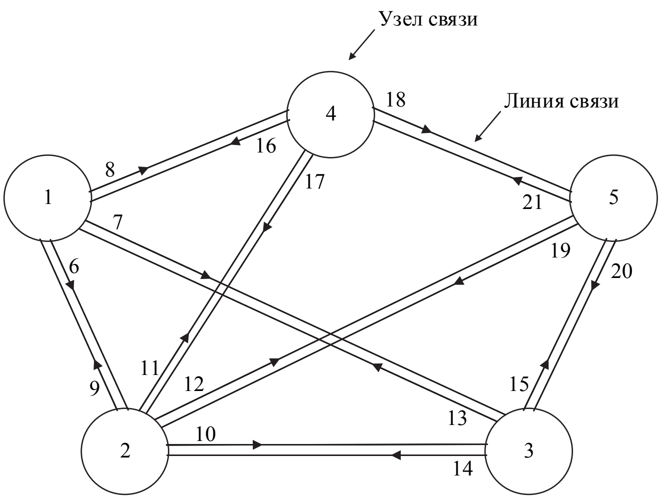

Рис. 1. Структура сети связи с пронумерованными элементами

Между тем в ходе реализации функций управления потоками пакетов в реальных СС важнее не просто получить список самых "узких" мест сети, а определить, как с наименьшими затратами их "расширить". Часто у администратора СС с КП имеется возможность увеличить пропускную способность отдельных направлений СС за счет дополнительной аренды/покупки ресурса у провайдера, однако рассмотренный выше инструментарий не позволяет определить параметры такого "расширения".

В свою очередь, известные решения поиска k максимальных потоков (минимальных разрезов), представленные в [12], такие как алгоритмы формирования множества простых разрезов методом сравнения и эвристическим методом обладают следующими недостатками: первый – требует значительных затрат для хранения промежуточных результатов вычисления, а второй – не исключает возможности неточных результатов в отдельных случаях.

Ниже предлагается алгоритм нахождения k максимальных потоков в порядке возрастания их пропускной способности лишенный данных недостатков.

2. Алгоритм нахождения *k* максимальных потоков. Граф СС с КП, структура которой показана на рисунке 1, целесообразно задать с помощью двух матриц смежности – в одной будут представлены номера элементов сети (узлов и линий связи), а в другой – их пропускная способность. Матрица смежности, в которой сосредоточены номера узлов сети и линий их соединяющих (если элемента нет, тогда его номер равен нулю) имеет вид:

$$M^{смежн}_{номера} = \begin{vmatrix} 1 & 6 & 7 & 8 & 0 \\ 9 & 2 & 10 & 11 & 12 \\ 13 & 14 & 3 & 0 & 15 \\ 16 & 17 & 0 & 4 & 18 \\ 0 & 19 & 20 & 21 & 5 \end{vmatrix}.$$

Матрица смежности, в которой представлены пропускные способности узлов и линий, их соединяющих (если элемента нет или вдоль него нельзя пустить поток, тогда его пропускная способность равна нулю; если пропускная способность элемента не определена, тогда она равна бесконечности) имеет вид:

$$M^{смежн}_{проп.спос} = \begin{vmatrix} \infty & 30 & 20 & 10 & 0 \\ 5 & \infty & 10 & 5 & 15 \\ 10 & 20 & \infty & 0 & 5 \\ 0 & 10 & 0 & \infty & 20 \\ 0 & 10 & 0 & 20 & \infty \end{vmatrix}.$$

Для удобства описания алгоритма введем ряд определений и одну процедуру.

Определение 1. Ограничивающее поток сечение (ОПС) – это такой набор элементов сети (узлов и/или соединительных линий), поток вдоль которых невозможно увеличить без изменения пропускной способности этих элементов.

Как правило, ОПС графа состоит из дуг, поток вдоль которых равен их пропускной способности. Такие дуги называют "насыщенными".

Определение 2. Ветвь дерева состояний – это последовательность элементов сети (узлов и/или соединительных линий), которая описывает группу состояний сети связи (последний элемент ветви – это пропускная способность ОПС для этой группы состояний, если ОПС не существует, тогда пропускная способность для этой группы состояний сети равна бесконечности).

Заметим, что часть элементов сети связи имеет ограниченную пропускную способность (положительные значения), а часть элементов – неограниченную пропускную способность (отрицательные значения) (рисунок 2).

Пример ветви дерева состояний:

$$vet_i = [8; -11; 12; 15; 40]. \tag{1}$$

Данная ветвь описывает группу состояний сети связи, которые представлены на рисунке 2.

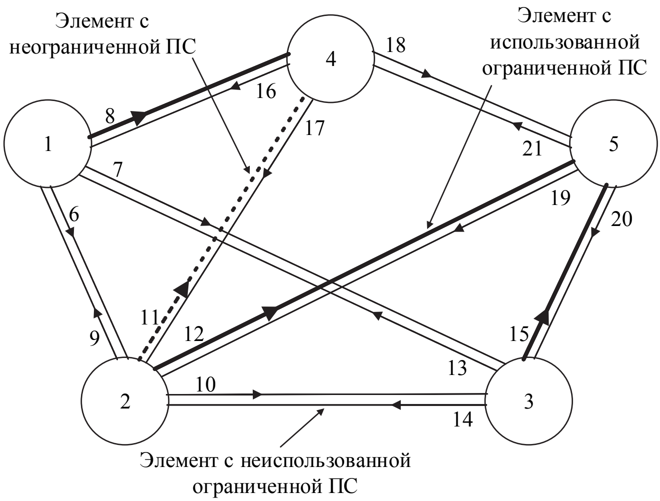

Рис. 2. Сеть связи, соответствующая ветви дерева состояний (выражение (1))

Определение 3. Ответвлением от ветви дерева состояний будем называть новую ветвь дерева состояний, которая получается путем перевода одного элемента с ограниченной пропускной способностью (положительное значение) в элемент с неограниченной пропускной способностью (отрицательное значение), при этом все последующие элементы, которые идут после него в исходной ветви из новой ветви исключаются. Пример формирования ответвления от ветви дерева состояний (см. выражение (1))

$$otk_i = [8; -11; 12; -15; \infty]. \tag{2}$$

Определение 4. Список ответвлений от ветви дерева состояний получается путем нахождения всех возможных ответвлений от элементов с ограниченной пропускной способностью исходной ветви, начиная с предпоследнего (последний элемент – это пропускная способность сети) и так далее, вплоть до первого элемента с неограниченной пропускной способностью. Если элементов с неограниченной пропускной способностью в исходной ветви дерева состояний нет, тогда вплоть до первого элемента исходной ветви.

Исходная ветвь дерева состояний

$$vet_j = [8; 11; 12; 15; \ 60]. \tag{3}$$

Список возможных ответвлений от исходной ветви

$$otk_1 = [8; 11; 12; -15; \ \infty], \tag{4}$$

$$otk_2 = [8; 11; -12; \infty], \tag{5}$$

$$otk_3 = [8; -11; \infty], \tag{6}$$

$$otk_4 = [-8; \infty]. \tag{7}$$

Процедура. Под формированием новых конкурирующих ветвей дерева состояний на основе списка ответвлений будем понимать следующую процедуру. Для каждого ответвления из списка, необходимо найти ОПС, используя алгоритм Форда-Фалкерсона (Эдмондса-Карпа, Диница или др.).

При этом необходимо учитывать, что часть элементов сети связи имеет неограниченную пропускную способность – это отрицательные номера элементов. Для таких элементов нужно заменить в исходной матрице $M_{проп.спос}^{смежн}$ их реальную пропускную способность значением бесконечность (после того, как ОПС будет найдено (или же не найдено), снова восстановить исходные значения пропускных способностей для соответствующих элементов). Если ОПС найдено, тогда с помощью его элементов необходимо дописать соответствующую строку списка ответвлений, при этом повторяющиеся элементы не пишутся, а в качестве последнего элемента ветви следует записать пропускную способность ОПС. Если путь не найден, тогда невозможно сформировать конкурирующую ветвь для такого состояния сети связи.

Ниже представлены новые ветви дерева состояний для списка ответвлений (см. выражения (4)–(7))

$$vet_k = [8; 11; 12; -15; 6; 7; 10; 60], \tag{8}$$

$$vet_{k+1} = [8; 11; -12; 6; 7; 14; 60], \tag{9}$$

$$vet_{k+2} = [8; -11; 12; 15; 18; 40], \tag{10}$$

$$vet_{k+3} = [-8; 12; 15; 17; 18; 40]. \tag{11}$$

Введенные выше определения и процедура позволяют сформулировать обобщенный алгоритм нахождения k максимальных потоков между двумя заданными вершинами графа.

I. Для исходной сети найти максимальный поток (минимальное сечение), используя алгоритм Форда-Фалкерсона (Эдмондса-Карпа, Диница или др.). Если такой поток будет равен бесконечности, тогда работа алгоритма заканчивается.

II. Найденное сечение необходимо записать в списки конкурирующих ветвей дерева состояний и конкурирующих сечений. Списки ответвлений и ОПС на этом шаге не содержат строк (эти списки пустые).

III. Подсчитать число строк в списке ОПС. Если оно равно числу k, то работа алгоритма заканчивается.

IV. Если список конкурирующих ветвей дерева состояний не содержит строк (список пустой), тогда работа алгоритма заканчивается. Иначе в списке конкурирующих ветвей необходимо найти строку с минимальной пропускной способностью (значение последнего элемента ветви дерева состояний). На ее основе сформировать список ответвлений и удалить эту ветвь из списка конкурирующих ветвей дерева состояний. Соответствующее ей сечение из списка конкурирующих сечений необходимо сравнить с уже имеющимися сечениями в списке ОПС, и, если такого сечения нет, записать его в список ОПС, а из списка конкурирующих сечений соответствующую строку удалить. Если же такое сечение уже есть (или это сечение является расширением уже имеющегося), то нужно удалить соответствующую строку из списка конкурирующих сечений.

V. Сформировать новые конкурирующие ветви дерева состояний на основе имеющегося списка ответвлений. При этом если для очередного ответвления сечение найдено, тогда оно записывается последним в список конкурирующих сечений, а полученная с его помощью новая ветвь дерева событий записывается в список конкурирующих ветвей в качестве последнего элемента (из списка ответвлений она удаляется). Если сечение не найдено, тогда соответствующая строка просто удаляется из списка ответвлений. По окончании работы данного шага алгоритма список ответвлений не должен содержать строк (список должен быть пустой).

VI. Переход на III шаг алгоритма.

Ниже работа данного алгоритма иллюстрируется на примере СС с КП, представленной на рисунке 1. Три ОПС между 1-м и 5-м узлом сети связи в порядке возрастания их пропускной способности находятся следующим образом.

Шаг 1. Список конкурирующих ветвей на первом шаге:

$$vet_1 * = [8; 11; 12; 15; 35]. \qquad (12)$$

Список ответвлений на первом шаге не содержит элементов:

$$otk_1 = [\varnothing]. \qquad (13)$$

Список конкурирующих сечений на первом шаге:

$$cut_1^* = [8; 11; 12; 15; 35]. \qquad (14)$$

Список ОПС на первом шаге:

$$ops_1 = [\varnothing]. \qquad (15)$$

Шаг 2. Список конкурирующих ветвей на втором шаге:

$$vet_1 = [\varnothing]. \qquad (16)$$

Список ответвлений на втором шаге:

$$otk_1 = [8; 11; 12; -15; \infty], \qquad (17)$$

$$otk_2 = [8; 11; -12; \infty], \qquad (18)$$

$$otk_3 = [8; -11; \infty], \qquad (19)$$

$$otk_4 = [-8; \infty]. \qquad (20)$$

Список конкурирующих сечений на втором шаге:

$$cut_1 = [\varnothing]. \qquad (21)$$

Список ОПС на втором шаге:

$$ops_1 = [8;\, 11;\, 12;\, 15;\, 35]. \tag{22}$$

Шаг 3. Список конкурирующих ветвей на третьем шаге:

$$vet_1 = [8;\, 11;\, 12;\, -15;\, 7;\, 10;\, 60], \tag{23}$$

$$vet_2 = [8;\, 11;\, -12;\, 6;\, 14;\, 60], \tag{24}$$

$$vet_3^* = [8;\, -11;\, 12;\, 15;\, 18;\, 40], \tag{24}$$

$$vet_4 = [-8;\, 12;\, 15;\, 17;\, 18;\, 40]. \tag{25}$$

Список ответвлений на третьем шаге:

$$otk_1 = [\varnothing]. \tag{26}$$

Список конкурирующих сечений на третьем шаге:

$$cut_1 = [7;\, 8;\, 10;\, 11;\, 12;\, 60], \tag{27}$$

$$cut_2 = [6;\, 8;\, 14;\, 60], \tag{28}$$

$$cut_3^* = [12;\, 15;\, 18;\, 40], \tag{29}$$

$$cut_4 = [12;\, 15;\, 17;\, 18;\, 40]. \tag{30}$$

Список ОПС на третьем шаге:

$$ops_1 = [8;\, 11;\, 12;\, 15;\, 35]. \tag{31}$$

Шаг 4. Список конкурирующих ветвей на четвертом шаге:

$$vet_1 = [8;\, 11;\, 12;\, -15;\, 7;\, 10;\, 60], \tag{32}$$

$$vet_2 = [8;\, 11;\, -12;\, 6;\, 14;\, 60], \tag{33}$$

$$vet_3 = [-8;\, 12;\, 15;\, 17;\, 18;\, 40]. \tag{34}$$

Список ответвлений на четвертом шаге:

$$otk_1 = [8;\, -11;\, 12;\, 15;\, -18;\, \infty], \tag{35}$$

$$otk_2 = [8;\, -11;\, 12;\, -15;\, \infty], \tag{36}$$

$$otk_3 = [8;\, -11;\, -12;\, \infty]. \tag{37}$$

Список конкурирующих сечений на четвертом шаге:

$$cut_1 = [7;\, 8;\, 10;\, 11;\, 12;\, 60], \tag{38}$$

$$cut_2 = [6;\, 8;\, 14;\, 60], \tag{39}$$

$$cut_3 = [12;\, 15;\, 17;\, 18;\, 40]. \tag{40}$$

Список ОПС на четвертом шаге:

$$ops_1 = [8; 11; 12; 15; 35], \quad (41)$$

$$ops_2 = [12; 15; 18; 40]. \quad (42)$$

Шаг 5. Список конкурирующих ветвей на пятом шаге:

$$vet_1 = [8; 11; 12; -15; 7; 10; 60], \quad (43)$$

$$vet_2 = [8; 11; -12; 6; 14; 60], \quad (44)$$

$$vet_3^* = [-8; 12; 15; 17; 18; 40], \quad (45)$$

$$vet_4 = [8; -11; 12; 15; -18; 6; 14; 60], \quad (46)$$

$$vet_5 = [8; -11; 12; -15; 6; 7; 10; 18; 60], \quad (47)$$

$$vet_6 = [8; -11; -12; 6; 14; 18; 60]. \quad (48)$$

Список ответвлений на пятом шаге:

$$otk_1 = [\varnothing]. \quad (49)$$

Список конкурирующих сечений на пятом шаге:

$$cut_1 = [7; 8; 10; 11; 12; 60], \quad (50)$$

$$cut_2 = [6; 8; 14; 60], \quad (51)$$

$$cut_3^* = [12; 15; 17; 18; 40], \quad (52)$$

$$cut_4 = [6; 8; 14; 60], \quad (53)$$

$$cut_5 = [6; 7; 10; 18; 60], \quad (54)$$

$$cut_6 = [6; 8; 14; 18; 60]. \quad (55)$$

Список ОПС на пятом шаге не изменяется, так как найденное сечение является расширением найденного ранее:

$$ops_1 = [8; 11; 12; 15; 35], \quad (56)$$

$$ops_2 = [12; 15; 18; 40]. \quad (57)$$

Шаг 6. Список конкурирующих ветвей на шестом шаге:

$$vet_1 = [8; 11; 12; -15; 7; 10; 60], \quad (58)$$

$$vet_2 = [8; 11; -12; 6; 14; 60], \quad (59)$$

$$vet_3 = [8; -11; 12; 15; -18; 6; 14; 60], \quad (60)$$

$$vet_4 = [8; -11; 12; -15; 6; 7; 10; 18; 60], \quad (61)$$

$$vet_5 = [8; -11; -12; 6; 14; 18; 60]. \quad (62)$$

Список ответвлений на шестом шаге:

$$otk_1 = [-8; 12; 15; 17; -18; \infty], \tag{63}$$

$$otk_2 = [-8; 12; 15; -17; \infty], \tag{64}$$

$$otk_3 = [-8; 12; -15; \infty], \tag{65}$$

$$otk_4 = [-8; -12; \infty]. \tag{66}$$

Список конкурирующих сечений на шестом шаге:

$$cut_1 = [7; 8; 10; 11; 12; 60], \tag{67}$$

$$cut_2 = [6; 8; 14; 60], \tag{68}$$

$$cut_3 = [6; 8; 14; 60], \tag{69}$$

$$cut_4 = [6; 7; 10; 18; 60], \tag{70}$$

$$cut_5 = [6; 8; 14; 18; 60]. \tag{71}$$

Список ОПС на шестом шаге:

$$ops_1 = [8; 11; 12; 15; 35], \tag{72}$$

$$ops_2 = [12; 15; 18; 40]. \tag{73}$$

Шаг 7. Список конкурирующих ветвей на седьмом шаге:

$$vet_1 = [8; 11; 12; -15; 7; 10; 60], \tag{74}$$

$$vet_2 = [8; 11; -12; 6; 14; 60], \tag{75}$$

$$vet_3 = [8; -11; 12; 15; -18; 6; 14; 60], \tag{76}$$

$$vet_4 = [8; -11; 12; -15; 6; 7; 10; 18; 60], \tag{77}$$

$$vet_5 = [8; -11; -12; 6; 14; 18; 60], \tag{78}$$

$$vet_6^* = [-8; 12; 15; -17; 18; 40], \tag{79}$$

$$vet_7 = [-8; 12; -15; 7; 10; 17; 18; 65], \tag{80}$$

$$vet_8 = [-8; -12; 6; 14; 17; 18; 80]. \tag{81}$$

Список ответвлений на седьмом шаге

$$otk_1 = [\varnothing]. \tag{82}$$

Список конкурирующих сечений на седьмом шаге:

$$cut_1 = [7; 8; 10; 11; 12; 60], \tag{83}$$

$$cut_2 = [6; 8; 14; 60], \quad (84)$$

$$cut_3 = [6; 8; 14; 60], \quad (85)$$

$$cut_4 = [6; 7; 10; 18; 60], \quad (86)$$

$$cut_5 = [6; 8; 14; 18; 60], \quad (87)$$

$$cut_6^* = [12; 15; 18; 40], \quad (88)$$

$$cut_7 = [7; 10; 12; 17; 18; 65], \quad (89)$$

$$cut_8 = [6; 14; 17; 18; 80]. \quad (90)$$

Список ОПС на седьмом шаге:

$$ops_1 = [8; 11; 12; 15; 35], \quad (91)$$

$$ops_2 = [12; 15; 18; 40]. \quad (92)$$

Шаг 8. Список конкурирующих ветвей на восьмом шаге:

$$vet_1 = [8; 11; 12; -15; 7; 10; 60], \quad (93)$$

$$vet_2 = [8; 11; -12; 6; 14; 60], \quad (94)$$

$$vet_3 = [8; -11; 12; 15; -18; 6; 14; 60], \quad (95)$$

$$vet_4 = [8; -11; 12; -15; 6; 7; 10; 18; 60], \quad (96)$$

$$vet_5 = [8; -11; -12; 6; 14; 18; 60], \quad (97)$$

$$vet_6 = [-8; 12; -15; 7; 10; 17; 18; 65], \quad (98)$$

$$vet_7 = [-8; -12; 6; 14; 17; 18; 80]. \quad (99)$$

Список ответвлений на восьмом шаге:

$$otk_1 = [-8; 12; 15; -17; -18; \infty]. \quad (100)$$

Список конкурирующих сечений на восьмом шаге:

$$cut_1 = [7; 8; 10; 11; 12; 60], \quad (101)$$

$$cut_2 = [6; 8; 14; 60], \quad (102)$$

$$cut_3 = [6; 8; 14; 60], \quad (103)$$

$$cut_4 = [6; 7; 10; 18; 60], \quad (104)$$

$$cut_5 = [6; 8; 14; 18; 60], \quad (105)$$

$$cut_6 = [7; 10; 12; 17; 18; 65], \quad (106)$$

$$cut_7 = [6; 14; 17; 18; 80]. \quad (107)$$

Список ОПС на восьмом шаге:

$$ops_1 \ = [8;\ 11;\ 12;\ 15;\ 35], \tag{108}$$

$$ops_2 \ = [12;\ 15;\ 18;\ 40]. \tag{109}$$

Шаг 9. Список конкурирующих ветвей на девятом шаге:

$$vet_1^* = [8;\ 11;\ 12;\ -15;\ 7;\ 10;\ 60], \tag{110}$$

$$vet_2 \ = [8;\ 11;\ -12;\ 6;\ 14;\ 60], \tag{111}$$

$$vet_3 \ = [8;\ -11;\ 12;\ 15;\ -18;\ 6;\ 14;\ 60], \tag{112}$$

$$vet_4 \ = [8;\ -11;\ 12;\ -15;\ 6;\ 7;\ 10;\ 18;\ 60], \tag{113}$$

$$vet_5 \ = [8;\ -11;\ -12;\ 6;\ 14;\ 18;\ 60], \tag{114}$$

$$vet_6 \ = [-8;\ 12;\ -15;\ 7;\ 10;\ 17;\ 18;\ 65], \tag{115}$$

$$vet_7 \ = [-8;\ -12;\ 6;\ 14;\ 17;\ 18;\ 80]. \tag{116}$$

Список ответвлений на девятом шаге:

$$otk_1 \ = [\varnothing]. \tag{117}$$

Список конкурирующих сечений на девятом шаге:

$$cut_1^* = [7;\ 8;\ 10;\ 11;\ 12;\ 60], \tag{118}$$

$$cut_2 \ = [6;\ 8;\ 14;\ 60], \tag{119}$$

$$cut_3 \ = [6;\ 8;\ 14;\ 60], \tag{120}$$

$$cut_4 \ = [6;\ 7;\ 10;\ 18;\ 60], \tag{121}$$

$$cut_5 \ = [6;\ 8;\ 14;\ 18;\ 60], \tag{122}$$

$$cut_6 \ = [7;\ 10;\ 12;\ 17;\ 18;\ 65], \tag{123}$$

$$cut_7 \ = [6;\ 14;\ 17;\ 18;\ 80]. \tag{124}$$

На девятом шаге не было найдено ни одного сечения.
Список ОПС на седьмом шаге:

$$ops_1 \ = [8;\ 11;\ 12;\ 15;\ 35], \tag{125}$$

$$ops_2 \ = [12;\ 15;\ 18;\ 40]. \tag{126}$$

Шаг 10. Список конкурирующих ветвей на десятом шаге:

$$vet_1 \ = [8;\ 11;\ -12;\ 6;\ 14;\ 60], \tag{128}$$

$$vet_2 \ = [8;\ -11;\ 12;\ 15;\ -18;\ 6;\ 14;\ 60], \tag{129}$$

$$vet_3 = [8; -11; 12; -15; 6; 7; 10; 18; 60], \tag{130}$$

$$vet_4 = [8; -11; -12; 6; 14; 18; 60], \tag{131}$$

$$vet_5 = [-8; 12; -15; 7; 10; 17; 18; 65], \tag{132}$$

$$vet_6 = [-8; -12; 6; 14; 17; 18; 80]. \tag{133}$$

Список ответвлений на десятом шаге:

$$otk_1 = [8; 11; 12; -15; 7; -10; \infty], \tag{134}$$

$$otk_2 = [8; 11; 12; -15; -7; \infty]. \tag{135}$$

Список конкурирующих сечений на десятом шаге:

$$cut_1 = [6; 8; 14; 60], \tag{136}$$

$$cut_2 = [6; 8; 14; 60], \tag{137}$$

$$cut_3 = [6; 7; 10; 18; 60], \tag{138}$$

$$cut_4 = [6; 8; 14; 18; 60], \tag{139}$$

$$cut_5 = [7; 10; 12; 17; 18; 65], \tag{140}$$

$$cut_6 = [6; 14; 17; 18; 80]. \tag{141}$$

Список ОПС на десятом шаге:

$$ops_1 = [8; 11; 12; 15; 35], \tag{142}$$

$$ops_2 = [12; 15; 18; 40], \tag{143}$$

$$ops_3 = [7; 8; 10; 11; 12; 60]. \tag{144}$$

Шаг 11. Работа алгоритма заканчивается, так как были найдены три максимальных потока между 1-м и 5-м узлом сети в порядке возрастания их пропускной способности.

3. Заключение. Обоснование эффективности предлагаемого в работе алгоритма основано на том очевидном факте, что длины новых ветвей дерева событий, получаемые на V шаге алгоритма, не могут быть меньше длины исходной ветви, ответвлениями от которой они являются. Предлагаемый в работе алгоритм может найти применение в СС с КП на этапе установления ВС для определения максимальной полосы пропускания в данном информационном направлении, а также в методах расчета структурной надежности сетей связи, использующих алгоритмы поиска простых сечений графа.

Литература

1. *Вегешна Ш.* Качество обслуживания в сетях IP: пер. с англ. // М.: Издательский дом "Вильямис". 2003. 368 с.
2. *Кучерявый Е.А.* Управление трафиком и качество обслуживания в сети Интернет // СПб.: Наука и Техника. 2004. 336 с.
3. *Ford L.R., Fulkerson Jr. Fulkerson, D.R.* Maximal Flow through a Network // Canadian Journal of Mathematics. 1956. pp. 399–404.
4. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ. 3-е издание // М.: "Вильямс". 2013. 1323 с.
5. *Dinitz Y.* Dinitz' Algorithm: The Original Version and Even's Version // Theoretical Computer Science: Essays in Memory of Shimon Even. Springer, 2006. pp. 218–240.
6. *Arora S., Hazan E., Kale S.* The multiplicative weights update method: A meta-algorithm and applications // Theory OF Computing. 2012. pp. 121–164.
7. *Christiano P., Kelner J., Madry A., Spielman D., Teng S.* Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs // Proceedings of the forty-third annual ACM symposium on Theory of computing (STOC 2011). 2011. pp. 273–282.
8. *Lee Y.T., Rao S., Srivastava N.* A new approach to computing maximum flows using electrical flows // Proceedings of the forty-fifth annual ACM symposium on Theory of computing (STOC 2013). 2013. pp. 755–764.
9. *Кристофидес Н.* Теория графов. Алгоритмический подход // М.: Мир. 1978. 432 с.
10. *Нечепуренко М.И., Попков В.К., Майнагашев С.М.* Алгоритмы и программы решения задач на графах и сетях // Новосибирск: Наука. Сиб. отделение. 1990. 515 с.
11. *Таха Х. А.* Введение в исследование операций. 6-е издание: пер. с англ. // М.: Издательский дом "Вильямс". 2001. 915 с.
12. *Дудник Б. Я., Овчеренко В. Ф., Орлов В. К. и др.* Надежность и живучесть систем связи / Под ред. Б. Я. Дудника // М.: Радио и связь. 1984. 216 с.

References

1. Vegeshna Sh. *Kachestvo obsluzhivanija v setjah IP* [Quality of service in networks IP] M.: Izdatel'skij dom "Vil'jamis". 2003. 368 p. (In Russ.).
2. Kucherjavyj E.A. *Upravlenie trafikom i kachestvo obsluzhivanija v seti Internet* [Traffic Management and quality of service in the Internet]. SPb.: Nauka i Tehnika. 2004. 336 p. (In Russ.).
3. *Ford L.R., Fulkerson Jr. Fulkerson, D.R.* Maximal Flow through a Network. Canadian Journal of Mathematics. 1956. pp. 399–404.
4. Kormen T., Lejzerson Ch., Rivest R., Shtajn K. *Algoritmy: postroenie i analiz. 3-e izdanie* [Algorithms: construction and analysis. 3rd edition]. M.: "Vil'jams". 2013. 1323 p. (In Russ.).
5. Dinitz Y. Dinitz' Algorithm: The Original Version and Even's Version. Theoretical Computer Science: Essays in Memory of Shimon Even. Springer, 2006. pp. 218–240.
6. Arora S., Hazan E., Kale S. The multiplicative weights update method: A meta-algorithm and applications. Theory OF Computing. 2012. pp. 121–164.
7. Christiano P., Kelner J., Madry A., Spielman D., Teng S. Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs. Proceedings of the forty-third annual ACM symposium on Theory of computing (STOC 2011). 2011. pp. 273–282.
8. Lee Y.T., Rao S., Srivastava N. A new approach to computing maximum flows using electrical flows. Proceedings of the forty-fifth annual ACM symposium on Theory of computing (STOC 2013). 2013. pp. 755–764.

9. Kristofides N. *Teorija grafov. Algoritmicheskij podhod* [Graph theory. Algorithmic approach]. M.: Mir. 1978. 432 p. (In Russ.).
10. Nechepurenko M.I., Popkov V.K., Majnagashev S.M. *Algoritmy i programmy reshenija zadach na grafah i setjah* [Algorithms and software of solutions problems on graphs and networks]. Novosibirsk: Nauka. Sib otdelenie, 1990. 515 p. (In Russ.).
11. Taha H. A. *Vvedenie v issledovanie operacija. 6-e izdanie: per. s angl.* [Introduction to Operations Research].M.: Izdatel'skij dom "Vil'jams". 2001. 915 p. (In Russ.).
12. Dudnik B. Ja., Ovcherenko V. F., Orlov V. K., et al. *Nadezhnost' i zhivuchest' sistem svjazi. Pod red. B. Ja. Dudnika* [Reliability and survivability of communication systems. Edited by Dudnik B. Ja.]. M.: Radio i svjaz'. 1984. 216 p. (In Russ.).

Трегубов Роман Борисович — к-т техн. наук, сотрудник Академии ФСО России. Область научных интересов: теория графов, теория массового обслуживания, теория вероятностей, применение методов математического моделирования в телекоммуникациях. Число научных публикаций — 61. treba@list.ru; Академия ФСО России, ул Приборостроительная, д. 35, г. Орел, 302034, РФ; р.т.: +7(4862)54-9731.

Tregubov Roman Borisovich — Ph.D., employee of the Academy of Federal Agency of protection of Russian Federation. Research interests: graph theory, waiting theory, probability theory, application of mathematical model approaches in telecommunications. The number of publications — 61. treba@list.ru; Academy of Federal Agency of protection of Russian Federation, Priborostroitelnaya Street, 35, Orel, 302034, Russia; office phone: +7(4862)54-9731.

Лазарев Сергей Николаевич — доцент, сотрудник Академии ФСО России. Область научных интересов: теория графов, теория массового обслуживания, теория вероятностей, применение методов математического моделирования в телекоммуникациях. Число научных публикаций — 60. serg.orel@mail.ru; Академия ФСО России, ул Приборостроительная, д. 35, г. Орел, 302034, РФ; р.т.: +7(4862)54-9731.

Lazarev Sergej Nikolaevich — associate professor, employee of the Academy of Federal Agency of protection of Russian Federation. Research interests: graph theory, waiting theory, probability theory, application of mathematical model approaches in telecommunications. The number of publications — 60. serg.orel@mail.ru; Academy of Federal Agency of protection of Russian Federation, Priborostroitelnaya Street, 35, Orel, 302034, Russia; office phone: +7(4862)54-9731.

Андреев Сергей Юрьевич — сотрудник Академии ФСО России. Область научных интересов: теория графов, теория массового обслуживания, теория вероятностей, применение методов математического моделирования в телекоммуникациях. Число научных публикаций — 1. us12a@mail.ru; Академия ФСО России, ул Приборостроительная, д. 35, г. Орел, 302034, РФ; р.т.: +7(4862)54-9731.

Andreev Sergej Jur'evich — employee of the Academy of Federal Agency of protection of Russian Federation. Research interests: graph theory, waiting theory, probability theory, application of mathematical model approaches in telecommunications. The number of publications — 1. us12a@mail.ru; Academy of Federal Agency of protection of Russian Federation, Priborostroitelnaya Street, 35, Orel, 302034, Russia; office phone: +7(4862)54-9731.

РЕФЕРАТ

Трегубов Р.Б., Лазарев С.Н., Андреев С.Ю. **Алгоритм нахождения *k* максимальных потоков.**

Управление трафиком в сети связи с коммутацией пакетов является одной из ключевых задач. В процессе ее решения, на этапе установления нового соединения осуществляется проверка, имеется ли в сети ресурс для обслуживания данного соединения с требуемым качеством. Однако в ходе реализации функций управления трафиком важно не просто знать список самых "узких" мест в сети, но и как их с наименьшими затратами "расширить".

В работе предлагается оригинальный алгоритм решения прикладной задачи теории графов о нахождении *k* максимальных потоков между двумя заданными вершинами графа, который позволяет решить данную проблему.

Описываемый подход представляет собой комплексное применение в едином оптимизационном цикле алгоритма Форда-Фалкерсона (Эдмондса-Карпа или Диница) и алгоритма построения усеченного дерева состояний в ширину. Особенности работы алгоритма демонстрируются на примере решения задачи нахождения *k* максимальных потоков для ориентированного графа.

Предлагаемый алгоритм может найти применение как на этапе установления нового соединения (для определения максимальной полосы пропускания в данном информационном направлении), так и в методах расчета структурной надежности сетей связи, использующих алгоритмы поиска простых сечений.

SUMMARY

Tregubov R.B., Lazarev S. N., Andreev S. Ju. **The algorithm of finding of k maximum flows.**

The traffic control in a communication network with switching of packages is one of the key tasks. To solve it, at the stage of establishment of new connection it is tested, whether the network has a resource to maintain this connection with the quality demanded. However while implementing functions of traffic control it is important to know the list of weak points of the network as well as the best way to overcome them.

To solve this problem the paper proposes the original algorithm of solution of an applied task of the graph theory on finding of *k* maximum flows between the two set count's tops.

The approach described represents a complex application of Ford-Fulkerson (Edmonds-Karp or Dinitz) algorithm and the algorithm of creation of a truncated tree of states in width in an indivisible optimizing cycle. The solution of the problem of finding of *k* of the maximum flows for the focused count demonstrates the peculiarities of algorithm functioning.

The algorithm suggested can be applied to define the maximum strip of a transmission in this information direction at the stage of establishment of new connection as well as to calculate the structural reliability of the communication networks using algorithms of simple sections search.