Методика определения набора реберно-независимых маршрутов на неориентированных графах телекоммуникационных сетей

Edge independence route set identification technique for telecommunication network undirected graphs

Канаев / Kanaev A.
Андрей Константинович
(kanaevak@mail.ru)
доктор технических наук, профессор.
ФГБОУ ВО "Петербургский государственный университет путей сообщения Императора Александра I" (ПГУПС),
заведующий кафедрой "Электрическая связь".
г. Санкт-Петербург

Лукичев / Lukichev M.
Михаил Михайлович
(mixailspbpy@mail.ru)
ПГУПС,
аспирант кафедры "Электрическая связь".
г. Санкт-Петербург

Сахарова / Saharova M.
Мария Александровна
(saharova@itain.ru)
кандидат технических наук.
ЗАО «Институт телекоммуникаций»,
заместитель начальника НИО
по разработке систем связи.
г. Санкт-Петербург

Ключевые слова: телекоммуникационная сеть – telecommunication network; реберно-независимые маршруты – edge independence routes; граф сети связи – network graph; алгоритм Йена – Yen's algorithm.

В статье представлена методика определения реберно-независимых маршрутов на неориентированных графах сети связи, позволяющая повысить эффективность функционирования телекоммуникационных сетей связи (ТКС) путем использования оптимального набора маршрутов для рассматриваемой корреспондирующей пары КП на графе. В основе полученной методики лежат новые алгоритмы поиска всех возможных путей на неориентированном графе ТКС и получения оптимального подмножества из требуемого числа реберно-независимых маршрутов с учетом интегрального коэффициента.

The article represents a technique of identification of edge independence routes in telecommunication network undirected graphs. This technique allows enhancing the performance of telecommunication networks by utilizing the best possible set of routes for the pair of correspondents in the graph. The developed technique is based on new algorithms intended to search all possible routes in a telecommunication network undirected graph and to obtain the best possible subset of the desired amount of edge independence routes with regard to the integrated coefficient.

Анализ построения нескольких маршрутов

При построении и эксплуатации современных ТКС одной из важнейших задач является определение маршрутов прохождения потоков информации между узлами сети. Совокупность всех маршрутов, представленная на физическом графе сети, непосредственно определяет потоковую структуру, которая определяет физическую возможность реализации потребностей пользователей ТКС [1, 5].

С целью обеспечения эффективности распределения ресурсов пропускной способности, а также обеспечения надежности ТКС целесообразно применять метод многопутевой маршрутизации [2, 6]. Для его реализации требуется определить несколько кратчайших маршрутов между рассматриваемыми КП.

Как правило, для получения нескольких маршрутов используют алгоритм Йена, либо его модификации [7, 8], суть которого сводится к нахождению кратчайших маршрутов любым известным методам [3, 4]. Затем, исключают ребра кратчайшего маршрута из рассматриваемого графа для нахождения следующего маршрута, путем повторного использования алгоритма поиска кратчайшего маршрута на уже модифицированном графе. Данную операцию производят до тех пор, пока не выполнится условие надежности либо эффективности рассматриваемого графа.

Алгоритм Йена позволяет определить требуемое количество (если возможно) зависимых маршрутов и

3 2018.indd 26 26.09.2018 10:03:06

на каждом этапе исключает одно ребро из сети. Однако для ТКС требуется определить реберно независимые маршруты, независимые маршруты по узлам, либо оба варианта сразу. Для этого используют модифицированный алгоритм Йена [9], который многократно использует алгоритм Дейкстры для определения кратчайшего маршрута между двумя узлами.

Основную вычислительную сложность алгоритма составляет выполнение поиска кратчайших маршрутов. Так как в модификации алгоритма Йена из графа исключаются все ребра, входящие в маршрут, найденный алгоритмом поимка кратчайшего пути, то сложность модифицированного алгоритма Йена для поиска k реберно-независимых маршрутов, в основе которого используется алгоритм Дейкстры, составит $\Psi(k \times V^2)$, где k – количество путей от вершины источника, V – количество вершин (узлов) сети.

Однако применение модифицированного алгоритма Йена приводит к существенному увеличению нагрузки на ребра минимального веса, либо, при использовании оптимизационных алгоритмов построения потоковой структуры – к увеличению стоимости передачи информации для конкретных КП узлов.

Рассмотрим детально модифицированный алгоритм Йена на примере графа транспортной сети связи. Путь задан взвешенный неориентированный граф сети $G\{V,R\}$, где $V\{v_i\}$ – множество вершин графа, $i = \overline{1,10}$, количество узлов, $R\{v_i v_j\}$ – множество ребер, элементы которого $r_{vivj} = c_{i,j} \geq 0$ – вес данного ребра. Данный граф изображен на рис. 1 с указанием веса (расстояния между узлами) на ребрах.

Рассматриваемый граф (рис. 1) обладает минимальной валентностью равной трем, следовательно между каждой парой узлов существует не менее трех маршрутов. Тогда, используя модифицированный алгоритм Йена, найдем три реберно-независимых маршрута от узла V_3 к узлу V_8.

Для этого создадим массив расстояний:

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
V_1		4	6			6				
V_2	4		5		9					
V_3	6	5		5			3			
V_4			5		5				10	
V_5		9		5					14	
V_6	6						5	5		
V_7			3			5			11	11
V_8						5			11	8
V_9				10	14		11	11		6
V_{10}							11	8	6	

Затем применим алгоритм Дейкстры, начиная с узла V_3. Для этого каждой не посещенной вершине присвоим метку, равную $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} = 124$, а начальной вершине V_3 – метку равную 0. После выполнения алгоритма Дейкстры, получим массив меток, соответствующий

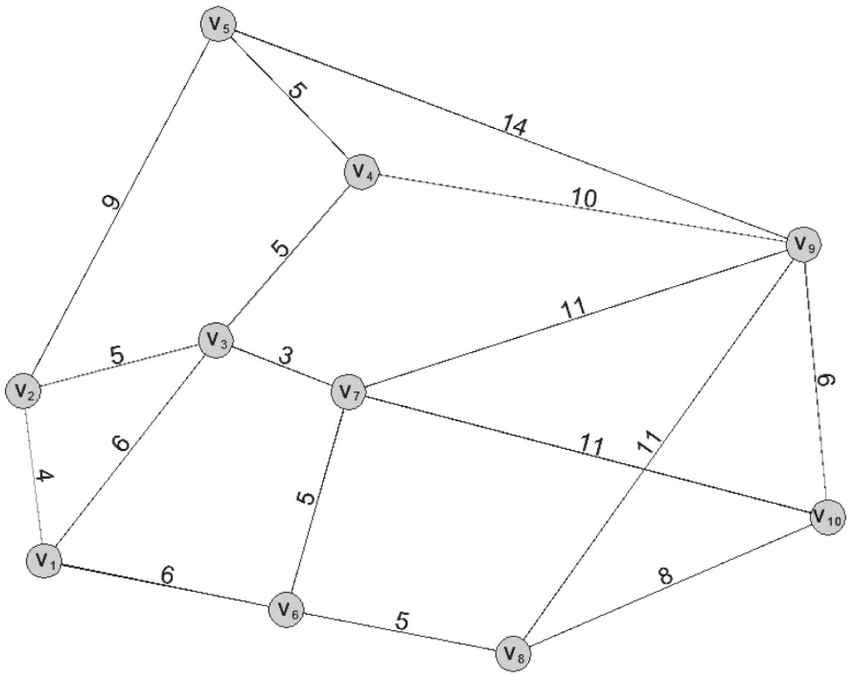

Рис. 1. Исходный граф ТКС

3 2018.indd 27 26.09.2018 10:03:09

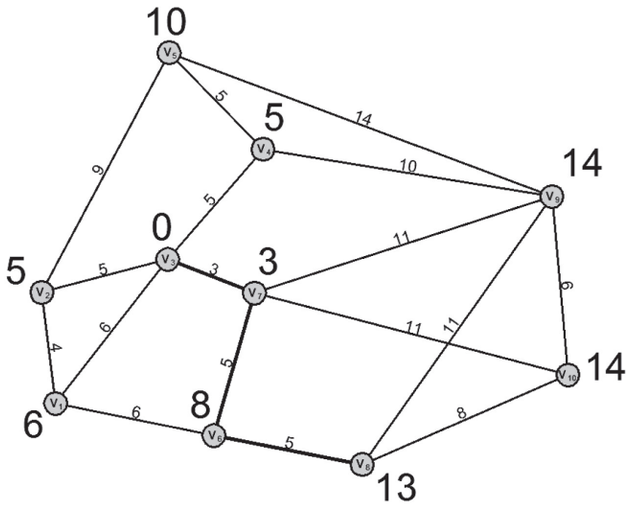

Рис. 2. Распределение меток, соответствующих расстоянию до каждого узла ТКС от вершины V_3

расстоянию от V_3 до каждой вершины ТКС. Графическое решение представлено на рис. 2.

После восстановления первого маршрута, получим вектор $M_{3,8}^{q=1} = \{V_3V_7, V_7V_6, V_6V_8\}$, при этом стоимость маршрута составит $M_{3,8}^{q=1} = 13$. Следующим шагом исключим из рассматриваемого графа ребра, входящие в маршрут $M_{3,8}^{q=1}$. На получившемся графе также выполним алгоритм Дейкстры. После восстановления маршрута получим $M_{3,8}^{q=2} = \{V_3V_4, V_4V_9, V_9V_8\} = 26$ и $M_{3,8}^{q=3} = \{V_3V_2, V_2V_5, V_5V_9, V_9V_{10}, V_{10}V_8\} = 42$.

Таким образом, путем использования модифицированного алгоритма Йена, в основе которого лежит алгоритм Дейкстры, для рассматриваемого графа сети получены три реберно-независимых маршрута между узлами V_3 и V_8:

$$\left.\begin{aligned} M_{3,8}^{q=1} &= \{V_3V_7, V_7V_6, V_6V_8\} = 13 \\ M_{3,8}^{q=2} &= \{V_3V_4, V_4V_9, V_9V_8\} = 26 \\ M_{3,8}^{q=3} &= \{V_3V_2, V_2V_5, V_5V_9, V_9V_{10}, V_{10}V_8\} = 42 \end{aligned}\right\} \sum_{i=1}^{k=3} M_{3,8}^{q=i} = 81$$

Если произвести суммирование длин всех полученных таким образом маршрутов, то получим: 81.

Разработанная методика позволяет за минимальное время выявить другие независимые маршруты, суммарным весом меньше чем маршруты, полученные модифицированным алгоритмом Йена. Результат представим ниже.

Методика определения набора реберно-независимых маршрутов

С целью оптимального использования ресурсов ТКС, построенных с применением многопутевой маршрутизации, требуется изменить подход к поиску независимых маршрутов между КП.

Для определения необходимого количества реберно-независимых маршрутов, оптимизированных по суммарной стоимости, необходимо решить две задачи:

1. Определение всех возможных маршрутов на ТКС;
2. Выделение из множества возможных маршрутов такого подмножества, которое бы удовлетворяло следующим условиям:

a. содержало количество маршрутов, равное заданному;
b. сумма всех маршрутов должна быть минимальной;
c. маршруты не должны иметь повторяющиеся ребра.

В крупных ТКС определение оптимальных маршрутов эмпирическим способом не представляется возможным, поэтому была разработана методика построения оптимальных реберно-независимых маршрутов по критерию суммарной стоимости. Кроме этого, небольшая модификация данного подхода позволяет определить оптимальные по суммарному весу зависимые маршруты с классификацией по степени повторного использования ребер сети.

Для описания функционирования методики определения k-независимых, оптимальных маршрутов по критерию минимального суммарного веса, предположим, что существует неориентированный граф ТКС $G(A,B,U,L)$, где $A = \{a_i\}, i = \overline{1,m}$ – совокупность узлов ТКС, $B = \{b_{ij}\}, i,j = \overline{1,n}$ – совокупность линий связи, $U = \{u_{ij}\}, i,j = \overline{1,n}$ – совокупность пропускных способностей линий передачи ТКС, а также совокупность веса линии связи по одному рассматриваемому параметру, $L'q = \{l_{ij}\}, i,j = \overline{1,n}$, где q – количество таких параметров (например, длинна линии, стоимость аренды канала, загруженность, уязвимость и т.д.).

Также определим пару узлов: s – источник потока и t – приемник.

Этап 1. На первом этапе определим основной параметр метрики ребра сети (линии связи), для этого

3 2018.indd 28 26.09.2018 10:03:12

будем использовать интегральные коэффициенты e_q, которые определяются индивидуально для каждой ТКС на основе приоритета того или иного параметра, таким образом получим: $L_{i,j} = \sum_{q=1}^{q} e_q \cdot L_q$.

Однако на практике вес ребра определяется одним из указанных выше параметров. Важно, чтобы все элементы $L_{i,j} \in R$, при этом чем меньше значение параметра $L_{i,j}$, тем предпочтительнее использовать ребро B_{ij} при построении маршрута. Простейшим случаем можно считать $\sum_{n=1}^{n} L_n = n$, тогда маршруты будут строиться по количеству ребер, т.е. с наименьшим количеством транзитных узлов сети.

Этап 2. Также необходимо построить массив связностей $C = \{c_{ij}\}$ размером

$m \times m$, таким, что

$$c_{ij} = \begin{cases} l_{i,j}, \text{если между } a_i \text{ и } a_j \text{ существует ребро } b_{i,j} \\ 0, \text{если между } a_i \text{ и } a_j \text{ нет ребра} \\ 0, \text{если i} = j \end{cases}.$$

Этап 3. После этого определим множество всех возможных маршрутов от источника потока *s* к приемнику *t*, путем использования алгоритма перебора с возвратом.

При переборе с возвратом создается дерево, в котором каждая вершина представляет частичное решение. Процесс создания такого дерева в точности повторяет процесс обхода графа в глубину с возвратом.

Для перечисления решений можно применять и обход в ширину, обход в глубину намного предпочтительнее, т.к. он занимает меньше места [9]. Текущее состояние поиска полностью представляется путем от корня к текущему узлу при обходе в глубину, что фактически может быть потенциальным решением и, следовательно, не требует модуля восстановления маршрута.

Функциональная блок-схема алгоритма представлена на рис. 3.

В качестве исходных данных выступает массив связностей *C*, а также начальный *s* и конечный *t* узел. Обход массива связностей начинается со строки, равной начальному узлу *s,* и первого столбца. Логически алгоритм перебора с возвратом можно представить в виде двух крупных функциональных блока поиска маршрута – блока перехода к следующему узлу сети и блока перехода к предыдущему узлу.

Результатом выполнения алгоритма служит массив $R^{s,t} = \{M^{s,t}{}_i\}$, $i = \overline{1,d}$, содержащий в себе все возможные пути от узла *s* к узлу *t*, где *d* – общее количество всех возможных маршрутов.

Сложность алгоритма соответствует сложности алгоритма поиска в глубину и составляет $\Psi(V^2)$ [10].

После получения массива $R^{s,t}$ необходимо выделить *k* независимых оптимальных маршрута. Математически данный процесс описывается как:

$$\sum_{k=1}^{k} \sum_{i=1,\, j=1}^{m} l_{i,j} \supset M_d^{s,t} \to \min \ldots \quad (1)$$

В случае крупных ТКС, где возможных маршрутов существенно больше, рекомендуется использовать распределение выполнения данной задачи по вычислительным ядрам системы, а также использовать для решения данной задачи серверный вариант платформы.

Функциональный алгоритм модуля определения *k* реберно-независимых оптимальных по критерию суммарной стоимости маршрутов можно представить в виде блок-схемы на рис. 4.

Применение разработанной методики позволило получить другой набор независимых трех маршрутов для варианта сети, представленной на рис. 1, а именно:

$$\left.\begin{aligned} M'^{q=1}_{3,8} &= \{V_3V_1, V_1V_6, V_6V_8\} = 17 \\ M'^{q=2}_{3,8} &= \{V_3V_4, V_4V_9, V_9V_8\} = 26 \\ M'^{q=3}_{3,8} &= \{V_3V_7, V_7V_{10}, V_{10}V_8\} = 22 \end{aligned}\right\} \sum_{i=1}^{k=3} M'^{q=i}_{3,8} = 65$$

В данном примере при использовании модифицированного метода поиска независимых маршрутов $\sum_{i=1}^{k=3} M'^{q=i}_{3,8} = 65$ на графе получено повышение эффективности использования ресурса ТКС на 26% при сравнении с результатами применения немодифицированного алгоритма Йена $\sum_{i=1}^{k=3} M'^{q=i}_{3,8} = 81$.

Таким образом, в ряде случаев, количество которых существенно увеличивается с ростом связности в графе (что особенно характерно для современных транспортных сетей), можно получить набор маршрутов, сумма которых значительно ниже, чем у маршрутов, полученных с использованием алгоритма Йена. Данный факт приводит в конечном итоге к существенному увеличению нагрузки на ребра минимального веса, либо, при использовании оптимизационных алгоритмов построения потоковой структуры, к увеличению стоимости передачи информации для конкретных корреспондирующих пар узлов.

Практическая реализация алгоритма поиска *k* оптимальных независимых маршрутов

С целью реализации построения потоковой структуры на основе оптимальных маршрутов был составлен алгоритм, позволяющий определить множество *k* оптимальных маршрутов из графа любой конфигурации.

Программная реализация данного алгоритма рассчитывает сумму весовых коэффициентов по каждому маршруту, а также распределяет заданный поток нагрузки по маршрутам согласно алгоритму путевого метода определения потоковой структуры.

В программе используется массив *A(i,j)* – массив связности, где $i, j = \overline{1,n}$, а *n* – количество узлов в сети. Также задается начальная вершина *s* и конечная – *t*. В ходе выполнения алгоритма заполняется массив *R(d,k)* путей

3 2018.indd 29 26.09.2018 10:03:14

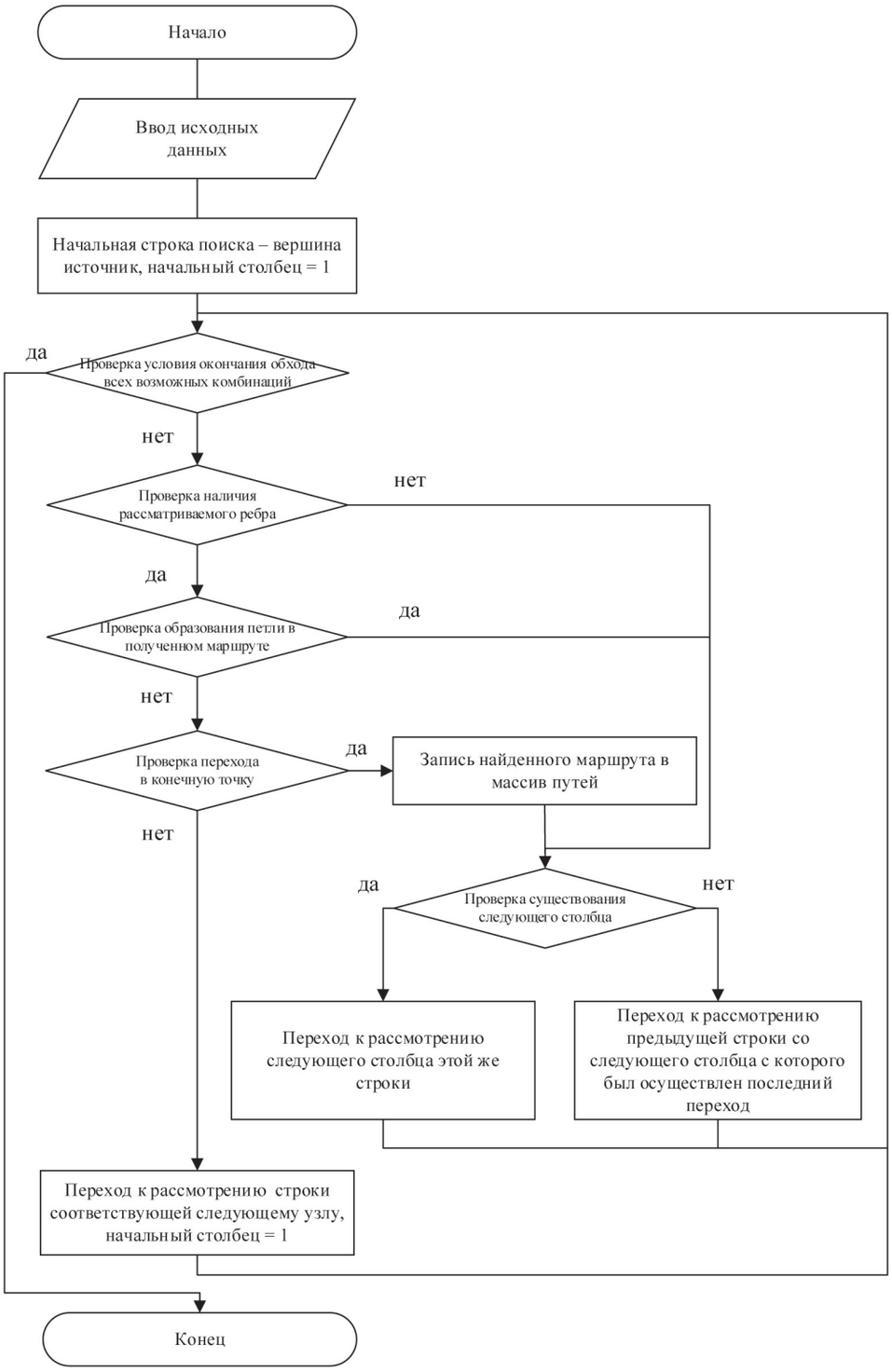

Рис. 3. Функциональная блок-схема алгоритма обхода графа в глубину без возврата

3 2018.indd 30 26.09.2018 10:03:16

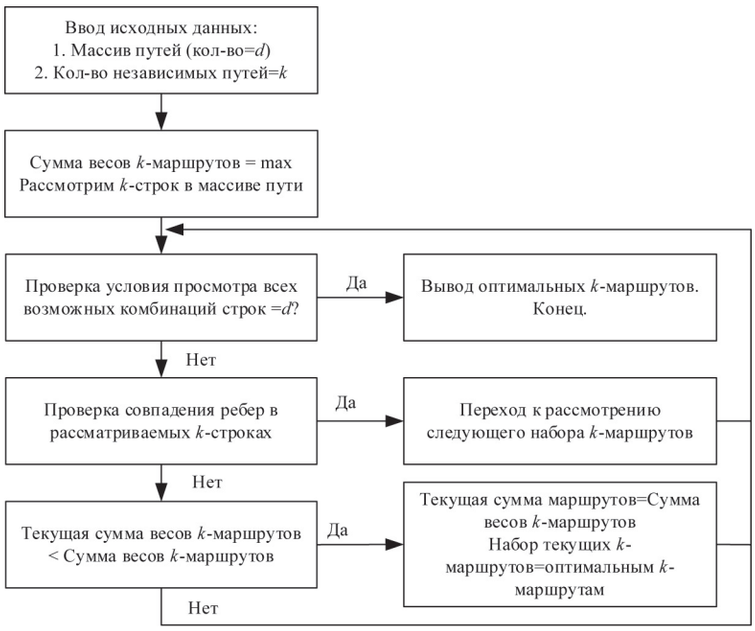

Рис. 4. Блок-схема определения k реберно-независимых оптимальных по критерию суммарной стоимости маршрутов

из вершины s в t, где d – количество всех возможных маршрутов, а $k \leq n+1$ – количество вершин, входящих в данный маршрут.

Первый элемент массива путей содержит сумму ребер, входящих в текущий маршрут. В качестве промежуточного массива сохранения результатов текущего обхода используется $M(p,1\ldots2)$ – массив текущего пути и пройденных столбцов. Первый элемент данного массива заполняется пройденными строками массива связности, а второй – столбцами.

Такой подход позволил обойти граф сети $A(i,j)$ всего один раз, без петель и повторных проверок связности узлов, соответственно сложность алгоритма $\Psi(n^2)$.

Таким образом, алгоритм поиска всех возможных маршрутов на графе может выполняться на персональном компьютере (ПК), а единственное ограничение накладывается на выделение памяти в vba для хранения найденных маршрутов. Максимальное количество маршрутов, которое допустимо для выполнения в vba, теоретически составляет $d = \frac{2^{31}-1}{n}$, однако на практике зависит от разрядности системы и оказалось равным порядка 2^{22} маршрутов. Например, для графа сети с числом узлов равным 40 и связностью равной 3 алгоритм нашел 655398 разных маршрутов за время порядка 15 сек.

Оценка времени получения всех маршрутов

Для определения области применения указанной программы был предложен тестовый макет сети, позволяющей увеличивать число узлов при неизменной валентности каждого узла рассматриваемой сети.

На каждой итерации определялись все возможные простые маршруты без петель от узла A_1 до A_j, а также фиксировалось время выполнения программы на ПК. Результаты работы программы представлены на рис. 5 и в табл.

Таким образом, составленная программа поиска всех возможных, неповторяющихся простых маршрутов работает достаточно эффективно для современных сетей, в рамках одной автономной системы.

Алгоритм формирования требуемого количества оптимального подмножества реберно-независимых маршрутов с учетом интегрального коэффициента

После определения множества маршрутов $R(d,k)$ необходимо получить подмножество из k маршрутов, удовлетворяющих условию (1). Как и указывалось выше, данная задача решается рекурсивной функцией, однако ввиду высокой вычислительной сложности рекурсивная функция была заменена на перебор всех возможных вариантов комбинаций маршрутов из не более 6 шт. В качестве примера модуля определения независимых маршрутов выбрано k=3.

С целью минимизации вычислений на каждом этапе рекурсивного цикла следует преобразовать полученный массив путей $R(d,k)$ в массив путей по ребрам $C(d,e+1)$ – где d – также количество всех возможных маршрутов, а е e – количество всех ребер графа. В первом столбце данного массива содержится сумма

3 2018.indd 31 26.09.2018 10:03:16

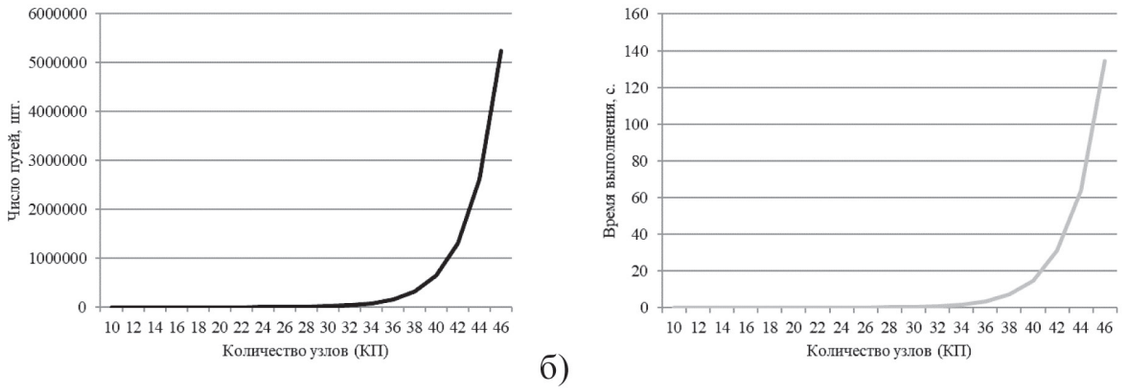

Рис. 5. График зависимости выявленного а) количества маршрутов (путей) и б) времени выполнения вычислений от количества узлов в сети

Таблица

Результаты экспериментов

N узлов	*M* маршрутов	*k* кол-во путей	Кол-во операций	Время выполнения
10	~100	3	10^6	0,019 сек
		4	10^8	1,917 сек
		5	10^{10}	195,72 сек
		6	10^{12}	~ 5,5 часов
15	~200	3	$8*10^6$	0,203 сек
		4	$2*10^9$	27,683 сек
		5	$3*10^{11}$	~1,1 час
20	~1000	3	10^9	17,429 сек
		4	10^{12}	~ 5,5 часов
25	~5000	3	10^{11}	40 мин.

весов ребер, входящих в данный маршрут, а также если ребро входит в рассматриваемый маршрут, то значение ячейки памяти $C(i,j)$ будет равно единице, в противном случае – нулю.

Для определения области применения алгоритма выделения из всех возможных несколько реберно независимых маршрутов по интегральному критерию был произведен анализ времени выполнения программы.

В качестве исходных данных были использованы наборы маршрутов, полученные в ходе оценки алгоритма определения всех возможных путей на неориентированном графе сети. Кроме этого, регистрировалось время выполнения программы, исполняемой на ПК. Результаты экспериментов приведены в таблице.

Таким образом, использование алгоритма выделения из всех возможных несколько реберно-независимых маршрутов по интегральному критерию существенно зависит от производительности ПК. Однако в большинстве практических задач по определению нескольких независимых маршрутов ограничиваются количеством маршрутизаторов доменной группы, обычно не превышающей 30 узлов для не более трех независимых маршрутов. Тогда использование полученного программного кода обосновано, т.к. время выполнения конечно и укладывается в рамки одного часа.

Кроме этого, оба полученных алгоритма написаны на одном языке программирования, а значит с целью удобства получения итогового результата легко объединяются в один программный комплекс по определению заданного количества реберно-независимых маршрутов на неориентированном графе сети. Такой комплекс сразу предоставляет необходимый результат за время последовательного выполнения двух описанных программ.

Выводы

Использование описанной методики позволяет определить требуемое количество оптимальных реберно-независимых маршрутов на неориентированном графе

3 2018.indd 32 26.09.2018 10:03:17

ТКС в автоматическом режиме, что существенно уменьшает время построения потоковых структур.

Выявлен выигрыш на 26 % от применения разработанной и программно-реализованной методики определения набора реберно-независимых маршрутов при сравнении результатов с результатами применения модифицированного алгоритма Йена.

Выполнен анализ времени выполнения задачи поиска оптимальных маршрутов между парой узлов разработанным программным комплексом. Выявлено, что за 40 мин. для сети из 25 узлов выполняется анализ около 5000 маршрутов.

Таким образом, применение описанных методических решений определения набора реберно независимых маршрутов при построении потоковой структуры позволяет повысить эффективность функционирования ТКС.

Литература

1. Алиев, Т. И. Моделирование и анализ подуровня агрегирования мультисервисной телекоммуникационной сети / Т.И. Алиев, И.Е. Никульский, В.О. Пяттаев // Техника связи. – 2009. – № 2. – С. 12–18.

2. Вишневский, В. М. Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. – М.: Техносфера, 2003. – 512 с.

3. Афанасьев, А. П. «Равномерные» алгоритмы последовательного заполнения потоковой сети потоками продуктов / А.П. Афанасьев, Я.Р. Гринберг, И.И. Курочкин // Труды ИСА РАН. – 2005. – Т. 14. – С. 118–140.

4. Малашенко, Ю. Е. Суперконкурентное распределение потоков в многопродуктовых сетях / Ю.Е. Малашенко, Н.М. Новикова // Дискретный анализ и исследование операций. – 1997. – № 2, Т. 4. – С. 34–54.

5. Канаев, А. К. Нейросетевая модель управления трафиком СПД с применением регулятора с предсказанием / А.К. Канаев, М.А. Сахарова // 69-я научно-техническая конференция, посвященная Дню радио. Труды конференции. – 2014. – С. 203–204.

6. Канаев, А. К. Моделирование процессов обработки потоков IP-пакетов с различными типами информационных составляющих на основе глубокого анализа трафика / А.К. Канаев, М.А. Сахарова // Бюллетень результатов научных исследований. – 2014. – № 3 (12). – С. 85–93.

7. Кристофидес, Н. Теория графов. Алгоритмический подход / Н. Кристофидес. М.: Мир, 1978. – 432 с.

8. Соколов, В. М. Результаты сравнения применения исходной и модифицированной методик синтеза структуры транспортной сети телекоммуникационной системы / В.М. Соколов, С.А. Ясинский // Информация и Космос. – 2016. – № 1. – С. 36–41.

9. Островерхий, С. М. Модифицированная методика синтеза структуры транспортной сети телекоммуникационной системы с использованием алгоритмов Йена и Прима / С.М. Островерхий, В.М. Соколов, С.А. Ясинский // Информация и Космос. – 2015. – № 4. – С. 18–21.

3 2018.indd 33 26.09.2018 10:03:17