

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 4

2000

УДК 519 : 681.3

В. А. Мелентьев

(Новосибирск)

**СКОБОЧНАЯ ФОРМА ОПИСАНИЯ ГРАФОВ
И ЕЕ ИСПОЛЬЗОВАНИЕ В СТРУКТУРНЫХ ИССЛЕДОВАНИЯХ
ЖИВУЧИХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ***

Предложен новый способ формального описания графов, основанный на совместном использовании отношений смежности (инцидентности, связности) и эквивалентности элементов графа (вершин, дуг или ребер). Показаны появившиеся в связи с этим новые возможности в анализе графов и в структурных исследованиях живучих вычислительных систем. Приведены методики и примеры таких исследований. Сформулированы и доказаны теоремы о совокупностях кратчайших маршрутов. Показано, что использование этих теорем значительно снижает объемы вычислений в алгоритмах поиска кратчайших маршрутов графа и в оценке структурной живучести вычислительных систем.

Введение. Живучесть вычислительной системы (ВС) определена как свойство, присущее ей в силу специально предпринятых мер и методов общесистемной (функциональной, структурной, алгоритмической и т. п.) организации и заключающееся в том, что любой отказ физических или логических компонентов не выводит систему за пределы допустимого диапазона значений показателей качеств ее функционирования, существенных для пользователей [1, 2]. Методы структурной организации живучих ВС основаны на использовании теории графов, при этом структура представляется графом $G(V, E)$, где между множеством вычислительных модулей и множеством V вершин графа, а также между множествами линий связи и ребер E графа существуют взаимно однозначные соответствия. Важнейшей характеристикой структуры живучей ВС является ее связность как мера защищенности графа от распада на не связанные между собой части при удалении вершин и(или) ребер [3]. Общеупотребительные показатели связности – вершинная (узловая) и реберная связности, характеризующие минимальный размер соответственно вершинного и реберного сечений. Под сечениями понимаются множества соответствующих элементов (вершин или ребер), удаление которых делает граф несвязным [4, 5].

Как показано в работах [1, 2], упомянутые выше показатели связности не несут в себе достаточной информации о способностях сохранения коммуникационных качеств в условиях деградации, вызванной отказами модулей и связей ВС (вершин и ребер графа). Поэтому в тех же работах были введены, а в [6, 7] получили развитие некоторые термины, характеристики и показатели, позволяющие использовать более полную информацию для оценки связности графа. Ниже приведено их краткое описание.

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 98-01-00402, № 99-01-00605).

Под соединением любой пары вершин $(s, t) \in V, s \neq t$, понимается такое множество всех непересекающихся маршрутов $M(s, t) = \{M_j(s, t)\}$, которое в совокупности обеспечивает протекание максимального потока минимальной стоимости из вершины-истока s в вершину-сток t . При этом маршрут $M_j(s, t)$ представляет собой простую цепь с начальной s и конечной t вершинами. Длины этих маршрутов $L_j(s, t)$ не должны превышать допустимых для данного соединения значений: $L_j(s, t) \leq L^*(s, t)$. Пересечение маршрутов также не допускается. Это означает, что любой из маршрутов множества $\{M_j(s, t)\}$ не содержит в качестве транзитных элементов цепи ни одного элемента, имеющегося в любом другом маршруте этого множества.

Далее предметом рассмотрения, как правило, являются неориентированные графы без петель с равновзвешенными (по длине и пропускной способности) ребрами. В этом случае максимальный поток определяется числом маршрутов в соединении вершин s и t $|\{M_j(s, t)\}|$, а минимальная стоимость – кратчайшими длинами этих маршрутов. Соединение характеризуется величиной $q(s, t)$, названной проводимостью и являющейся суммой проводимостей всех маршрутов, составляющих это соединение:

$$q(s, t) = \sum_{j=1}^{|M|} q_j(s, t) = \sum_{j=1}^{|M|} L_j^{-1}(s, t).$$

Введение понятий соединения и проводимости, используемых при оценке структурной живучести ВС (живучести графа ВС), поставило задачу поиска множества непересекающихся кратчайших маршрутов в соединении. В принципе при решении этой задачи «в лоб» можно итерационно использовать известные алгоритмы поиска кратчайших маршрутов, например алгоритм Флойда и Фалкерсона [8, 9], последовательно модифицируя граф для каждого соединения путем удаления вершин и ребер, уже использованных в предыдущей итерации поиска. Однако в этом случае процедура решения получается слишком объемной. Сложность решения существенно возрастает с увеличением размера и размерности ВС, а это, в свою очередь, делает неэффективным использование таких алгоритмов в оперативном управлении функционированием живучих ВС.

В данной работе предложен способ формального описания графов и показаны появившиеся в связи с этим новые возможности в их анализе и в структурных исследованиях вычислительных систем.

1. Скобочная форма описания графа. Общеизвестны две формы описания графа: графическая и матричная. Графическая форма представляет собой отображение на плоскости перечислимого множества вершин в виде узлов (точек) и линий связи (дуг, ребер), непосредственно соединяющих эти вершины. Матричная форма в зависимости от выбора образующих матрицу строк и столбцов носит название матрицы смежности (строки и столбцы соответствуют вершинам графа), матрицы инцидентности (строки – вершины, столбцы – ребра) и матрицы смежности ребер (элементами строк и столбцов являются ребра графа). Понятно, что в отличие от графической формы матричная предполагает введение однозначных обозначений соответствующих элементов графа, например нумерации.

Ниже предлагается скобочная форма описания графа, позволяющая значительно упростить процессы получения требуемых характеристик графа и

существенно снизить трудоемкость этих процессов за счет использования уже заложенных в этой форме неких промежуточных результатов.

Определим некоторые отношения между элементами графа, которые будут полезны для введения скобочной формы.

Отношение смежности связывает однотипные элементы графа: вершины или дуги (ребра). Если существует дуга из вершины a в вершину b , то можно записать $a(b)$, но не $b(a)$. Если связь вершин a и b лишена ориентации, то $a(b) = b(a)$. Такая же форма записи справедлива и в отношении смежных ребер или дуг. В последнем случае также необходимо учитывать их ориентацию. Заметим, что в отличие от общепринятого определения смежности в качестве бинарного отношения для некоторых пар элементов графа предполагаем бинарность этого отношения для некоторых пар объектов, каждый из которых может представлять собой множество эквивалентных элементов. Кроме того, для общего случая рассмотрения графов со взвешенными элементами введем понятие строгого отношения смежности, в котором учитывается ограничение на длину ребер или на вес вершин. При необходимости может быть использовано также отношение инцидентности, связывающее разнотипные элементы графа: вершины и дуги (ребра).

Отношение эквивалентности, определенное на множестве вершин V или ребер E графа $G(V, E)$, вычленяет в этом множестве некоторые классы элементов, объединенных общим свойством (смежности, инцидентности или связности) по отношению к одному или нескольким элементам графа. Если при этом учитывается отношение строгой смежности, то и отношение эквивалентности будет строгим. Таким образом, вершины a, b и c являются эквивалентными по отношению к вершине d , если каждая из них смежна с последней, т. е. запись $d(a, b, c)$ справедлива, если $d(a) \cap d(b) \cap d(c) = 1$.

Рассмотрим вышеизложенное на простом примере (рис. 1). Отношениями смежности для каждой вершины данного графа являются $0(1), 0(3), 1(0), 1(2), 1(3), 2(1), 2(3), 3(0), 3(1), 3(2)$. С вершиной 0 смежны вершины 1 и 3, поэтому они эквивалентны друг другу относительно вершины 0, т. е. $0(1, 3)$. Аналогично для остальных вершин можно записать: $1(0, 2, 3), 2(1, 3), 3(0, 1, 2)$. Запись вида $1(0, 2, 3)$ означает, что вершины 0, 2 и 3 эквивалентны в отношении их смежности с вершиной 1. Подобная запись, сочетающая в себе отношения смежности и эквивалентности, является симметричной для неориентированных графов. Возьмем пару вершин 1 и 3 из рассмотренного примера и определим множество вершин, эквивалентность которых заключается в их смежности с любой из оставшихся вершин графа: $(1(0, 2, 3) \cap 3(0, 1, 2)) = (1, 3)((0, 2, 3) \cap (0, 1, 2)) = (1, 3)(0, 2)$, т. е. вершины 0 и 2 эквивалентны в их конъюнктивной (одновременной, совместной) смежности с вершинами 1 и 3. Справедливо и обратное утверждение: вершины 1 и 3 эквивалентны в отношении их конъюнктивной смежности с вершинами 0 и 2.

В отдельных случаях полезно использовать *отношение связности*: вершины a и b связаны, если существует хотя бы один маршрут конечной длины из a в b , $L(a \rightarrow b) < \infty$. Если при этом имеется ограничение максимальной длины маршрута $L(a \rightarrow b) \leq L^*$, то отношение связности назовем строгим. Для неориентированных графов (как показано в разд. 3) отношение связности является симметрическим: $a \leftrightarrow b, L(a \rightarrow b) = L(b \rightarrow a)$.

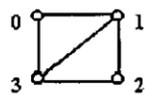
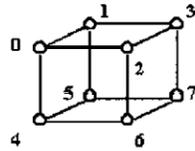


Рис. 1



$$G \left(\begin{array}{l} 0(1, 2, 4), \quad 1(0, 3, 5), \quad 2(0, 3, 6), \quad 3(1, 2, 7), \\ 4(0, 5, 6), \quad 5(1, 4, 7), \quad 6(2, 4, 7), \quad 7(3, 5, 6) \end{array} \right)$$

Рис. 2

Используя введенные понятия, рассматриваемый в качестве примера на рис. 1 граф опишем следующим образом: $G(0(1,3), 1(0,2,3), 2(1,3), 3(0,1,2))$. Учитывая, что граф неориентированный, можно исключить дублирование записи и минимизировать это описание:

$$G(0(1,3), 1(2,3), 2(3)) = G((0,2)(1,3), 1(3)) = G((0,2)(1(3), 3)).$$

На рис. 2 приведено описание куба в скобочной форме с числом записей, равным числу вершин куба. Минимизируя это описание, запишем

$$\begin{aligned} &G((0(4(5)), 3(7(6)))(1(5(7)), 2(6(4)))) \text{ или} \\ &G((5(1(0)), 6(2(3)))(4(0(2)), 7(3(1)))) \text{ или} \\ &G((4(5(7)), 2(3(1)))(0(1(5)), 6(7(3)))) \text{ или} \\ &G((1(0(2)), 7(6(4)))(3(2(6)), 5(4(0)))) \end{aligned}$$

Представленный на рис. 2 граф и нумерация его вершин соответствуют определению гиперкуба. Этот граф может быть описан еще более компактно через обобщенное описание смежности его вершин:

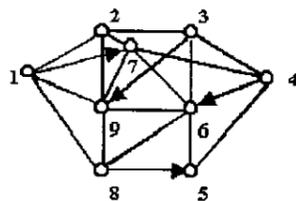
$$\bigvee_{i,j=0}^7 v_i(v_j | i \oplus j = 2^k, \quad 0 \leq k < 3).$$

Здесь использовано несколько модифицированное обозначение квантора общности: чтобы не загромождать подкванторное выражение, области определения i, j вынесены в обозначение самого квантора подобно обозначениям знаков алгебраической суммы, произведения, дизъюнкции и т. п. Представленная запись читается следующим образом: в описываемом графе любая вершина с номером от 0 до 7 смежна со всеми вершинами с номерами из той же области определения, если поразрядное сложение по модулю 2 номеров этих вершин представляет собой k -ю степень двойки с областью определения k от 0 до 2. Таким же образом может быть описан любой n -мерный гиперкуб

$$\bigvee_{i,j=0}^{2^n-1} v_i(v_j | i \oplus j = 2^k, \quad 0 \leq k < n)$$

и другие регулярные структуры.

Для описания ориентированных графов со взвешенными дугами в качестве элемента формального описания может быть предложена запись вида $A(B_b, C_c, D_d, \dots)$. Такая запись означает, что вершины B, C и D эквивалентны в отношении их смежности с вершиной A , причем дуги из A в эти вершины



Т а б л и ц а 1

	1	2	3	4	5	6	7	8	9
1		10					3	6	12
2	10		18				2		13
3		18		25		20			7
4			25		5	16	4		
5				5		10			
6			20		10		14	15	9

обладают весовыми характеристиками, обозначенными индексами b, c и d . В качестве примера приведем описание графа, иллюстрирующего в [8] алгоритмы Дейкстры, Флойда и Данцига поиска кратчайших путей (рис. 3 и табл. 1).

Приведенный здесь граф может быть описан с учетом весовых значений и ориентации дуг следующим образом:

$$G \left(\begin{array}{ccc} 1(7_3, 8_6, 2_{10}, 9_{12}), & 2(7_2, 1_{10}, 9_{13}, 3_{18}), & (9_7, 2_{18}, 6_{20}, 4_{25}), \\ 4(7_4, 5_5, 6_{16}, 3_{25}), & 5(4_5, 6_{10}), & 6(9_9, 5_{10}, 7_{14}, 8_{15}, 3_{20}), \\ 7(2_2, 4_4, 6_{14}, 9_{24}), & 8(9_5, 1_6, 6_{15}, 5_{23}), & 9(8_5, 6_9, 1_{12}, 2_{13}, 7_{24}) \end{array} \right).$$

Здесь эквивалентные вершины перечислены в порядке возрастания весов их дуг, исходящих из вершины, являющейся основой их эквивалентности. Подобная упорядоченность записи не является обязательной, но будет полезной при последующем более подробном рассмотрении этого графа.

Итак, наряду с общеупотребительным выражением вида $G(V, E)$, где отношения между множествами вершин V и ребер (дуг) E заданы графически либо соответствующими матрицами, графы могут быть описаны компактными и более информативными выражениями вида $G(V(V))$, $G(E(E))$ или $G(V(E))$, учитывающими соответственно смежность вершин и дуг (ребер) или инцидентность вершин и дуг (ребер).

2. Использование скобочной формы в структурных исследованиях вычислительных систем. Структурные исследования вычислительных систем включают в себя широкий круг задач, связанных с маршрутизацией (поиск совокупности кратчайших маршрутов), исследованием структурной живучести (поиск кратчайших непересекающихся маршрутов соединения), модификацией описания графа ВС в условиях ее деградации, синтезом оптимальных графов ВС и т. п. Для иллюстрации возможностей предлагаемого здесь способа описания графов ниже приведены методики и примеры решения некоторых из этих задач.

2.1. Поиск кратчайших маршрутов в соединениях графа. Задача состоит в определении всех кратчайших путей из каждой вершины во все остальные

вершины исходного графа. Известный алгоритм Дейкстры [9] и его модифицированные варианты (алгоритмы Форда, Флойда, Данцига и др.) используют в качестве исходной информации матричное описание графа со взвешенными дугами, веса которых могут принимать отрицательные значения. Для демонстрации возможностей предлагаемого здесь метода рассмотрим представленный выше граф (см. рис. 3, табл. 1) и опишем в скобочной форме все кратчайшие по длине маршруты из каждой вершины данного графа. Последовательность составления такой записи показана на примере вершины 1.

Итак, вершина 1 смежна (в смысле исходящих из нее дуг) с вершинами $7_3, 8_6, 2_{10}, 9_{12}$. Это может быть отражено в следующей записи и в соответствующей ей одномерной таблице (далее для наглядности отображения этапности поиска она развернута во времени):

$$1(7_3^3 \dots \begin{matrix} 8_6^6 \\ 2_{10}^{10} \\ 9_{12}^{12} \end{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 10 & & & & & 3 & 6 & 12 \end{matrix} \quad (1)$$

Здесь нижний индекс при номере вершины равен длине дуги из предшествующей вершины, а верхний – длине пути из исходной вершины. Равенство верхнего индекса вершины нижнему указывает на ее смежность с вершиной-источником. В пустую до начала процесса записи таблицу длин путей (номер столбца таблицы соответствует номеру вершины графа) вписаны значения верхних индексов, причем для исходной вершины 1 эта величина равна нулю.

Далее последовательно раскрываем запись (1), обновляя элементы таблицы меньшими значениями, если таковые появляются:

$$1(7_3^3 (2_2^5 \dots \begin{matrix} 8_6^6 & 4_4^7 \\ 2_{10}^{10} & 6_{14}^{17} \\ 9_{12}^{12} & \end{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 10 & & & & & 3 & 6 & 12 \\ & 5 & & 7 & & 17 & & & \end{matrix} \quad (2)$$

$$1(7_3^3 (2_2^5 (3_{18}^{23}, 4_4^7 (5_5^{12}, 6_{14}^{17}), 8_6^6 (9_5^{11})). \begin{matrix} 8_6^6 & 4_4^7 & 6_{14}^{17} \\ 2_{10}^{10} & 4_4^7 & 6_{14}^{17} \\ 9_{12}^{12} & 6_{14}^{17} & \end{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 10 & & & & & 3 & 6 & 12 \\ & 5 & & 7 & & 17 & & & \\ & & 23 & & & & & & \\ & & & 12 & & & & & \\ & & & & & & & & 11 \end{matrix}$$

Вершина 7 смежна с вершинами $2_2, 4_4, 6_{14}, 9_{24}$. Верхние индексы при указанных вершинах с учетом длины дуги из вершины 1 в вершину 7 будут выглядеть следующим образом: $2_2^5, 4_4^7, 6_{14}^{17}, 9_{24}^{27}$. В связи с тем, что длина пути к вер-

шине 9 через вершину 7 равна 27 и превышает значение, уже имеющееся в таблице и равное 12, то на этом этапе формирования записи вершина 9 игнорируется, запись и таблица получают дальнейшее развитие, приведенное в (2). Из таблицы видно, что вершина 2 изменила свой верхний индекс в сторону уменьшения (с 10 до 5), поэтому соответствующий элемент таблицы заменен значением длины более короткого пути через вершину 7, нежели более длинный непосредственный (не имеющий транзитов) маршрут. Аналогично поступаем по мере дальнейшего описания до тех пор, пока последовательность не прервется из-за отсутствия путей, более коротких, чем уже занесенные в таблицу. Окончательная запись кратчайших маршрутов из вершины 1 во все остальные вершины рассмотренного графа будет иметь вид:

$$1(7_3^3(2_2^5(3_{18}^{23}), 4_4^7(5_5^{12}), 6_{14}^{17}), 8_6^6(9_5^{11})). \quad (3)$$

Вычисление и описание всех кратчайших маршрутов, входящих в любую из вершин графа, проводится по той же методике. Однако в этом случае необходимо сменить ориентацию дуг на противоположную, т. е. использовать не строки, а столбцы матрицы смежностей. Результирующее описание входящих в вершину 1 маршрутов имеет вид:

$$\downarrow 1(8_6^6(9_5^{11}(3_7^{18}, 6_9^{20})), 2_{10}^{10}(7_2^{12}(4_4^{16}(5_5^{21}))))).$$

Для составления описаний входящих в вершину кратчайших маршрутов можно также использовать описания исходящих и наоборот. Аналогично (3) составим описания всех исходящих кратчайших маршрутов для остальных вершин графа:

$$\begin{aligned} & 2(7_2^2(4_4^6(5_5^{11}), 6_{14}^{16}), 1_{10}^{10}(8_6^{16}), 9_{13}^{13}, 3_{18}^{18}), \quad 3(9_7^7(8_5^{12}(1_6^{18}), 6_9^{16}(5_{10}^{26})), 2_{18}^{18}(7_2^{20}(4_4^{24}))), \\ & 4(7_4^4(2_2^6(1_{10}^{16}(8_6^{22}), 9_{13}^{19}, 3_{18}^{24})), 5_5^5(6_{10}^{15})), \quad 5(4_5^5(7_4^9(2_2^{11}(1_{10}^{21}, 3_{18}^{29}))), 6_{10}^{10}(9_9^{19}(8_5^{24}))), \\ & 6(9_9^9(8_5^{14}(1_6^{20})), 5_{10}^{10}(4_5^{15}), 7_{14}^{14}(2_2^{16}), 3_{20}^{20}), \quad 7(2_2^2(1_{10}^{12}(8_6^{18}), 9_{13}^{15}, 3_{18}^{20}), 4_4^4(5_5^9), 6_{14}^{14}), \\ & 8(9_5^5(6_9^{14}), 1_6^6(7_3^9(2_2^{11}(3_{18}^{29}), 4_4^{13}(5_5^{18}))), \quad 9(8_5^5(1_6^{11}(7_3^{14}(4_4^{18}))), 6_9^9(5_{10}^{19}, 3_{20}^{29}), 2_{13}^{13}). \end{aligned} \quad (4)$$

Используя полученные описания, построим граф G^* кратчайших маршрутов (рис. 4) и соответствующую ему матрицу смежностей (табл. 2).

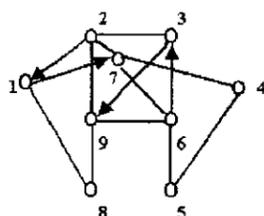


Рис. 4

Таблица 2

	1	2	3	4	5	6	7	8	9
1							3	6	
2	10		18				2		13
3		18							7
4					5		4		
5				5		10			
6			20		10		14		9
7		2		4		14			
8	6								5
9		13				9		5	

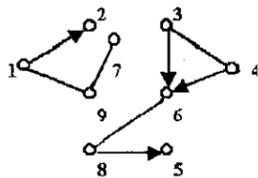


Рис. 5

Таблица 3

	1	2	3	4	5	6	7	8	9
1		10							12
2									
3				25		20			
4			25			16			
5									
6								15	
7									24
8					23	15			
9	12						24		

Перечислив связи полученного графа кратчайших маршрутов, запишем его в следующем виде:

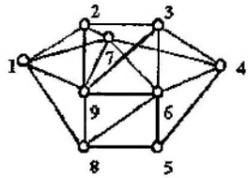
$$G^* \left(\begin{array}{l} 1(7_3, 8_6), \quad 2(7_2, 1_{10}, 9_{13}, 3_{18}), \quad 3(9_7, 2_{18}), \\ 4(7_4, 5_5), \quad 5(4_5, 6_{10}), \quad 6(9_9, 5_{10}, 7_{14}, 3_{20}), \\ 7(2_2, 4_4, 6_{14}), \quad 8(9_5, 1_6), \quad 9(8_5, 6_9, 2_{13}) \end{array} \right)$$

Сравнивая описание исходного графа (см. рис. 3) с описанием полученного графа кратчайших маршрутов (см. рис. 4), можно убедиться в избыточности первого. В связи с этим вполне правомерны постановка и решение задачи поиска графа $G' = G - G^*$ избыточных связей и ликвидации последних с целью оптимизации исходного графа. Имея полученные выше результаты, решить задачу в такой постановке не представляет сложности. Достаточно из исходной матрицы вычесть полученную. Результаты решения приведены на рис. 5 и в табл. 3. Перечень избыточных связей, подлежащих ликвидации в процессе оптимизации исходного графа, содержится в описании соответствующего графа G' :

$$G'(1(2_{10}, 9_{12}), 3(6_{20}, 4_{25}), 4(6_{16}, 3_{25}), 6(8_{15}), 7(9_{24}), 8(6_{15}, 5_{23}), 9(1_{12}, 7_{24})).$$

Структурные исследования вычислительных систем в рассмотренных выше постановках и решениях существенно проще. Это связано со спецификой представления структурной модели таких систем, как правило, в виде неориентированных графов без петель с равновзвешенными ребрами. В качестве примера далее приведено решение задачи о кратчайших маршрутах на рассмотренном выше графе, но дуги этого графа заменены ребрами и веса всех ребер одинаковы. Такой нерегулярный граф с различной связностью вершин взят в качестве примера из общих соображений, учитывающих возможность отказа вершин и ребер графа ВС в процессе ее эксплуатации, что, в свою очередь, приводит к неизбежному нарушению регулярности исходной структуры.

Итак, структура ВС представлена неориентированным графом без петель с равновзвешенными ребрами (рис. 6).



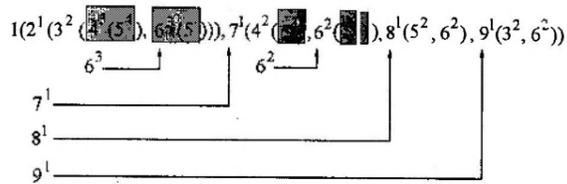
$$G \left(\begin{array}{l} 1(2, 7, 8, 9), \quad 2(1, 3, 7, 9), \quad 3(2, 4, 6, 9), \\ 4(3, 5, 6, 7), \quad 5(4, 6, 8), \quad 6(3, 4, 5, 7, 8, 9), \\ 7(1, 2, 4, 6, 9), \quad 8(1, 5, 6, 9), \quad 9(1, 2, 3, 6, 7, 8) \end{array} \right)$$

Рис. 6

Одной из возможных минимизированных форм представления данного графа является описание вида

$$G((1(2(3)), 6(3(4), 4(5), 5)))(7(2, 4, 9), 8(5, 9), 9(2, 3))).$$

В том, что это описание является достаточным и при этом не содержит в себе избыточной информации, легко убедиться, построив граф, соответствующий описанию. Задача состоит в нахождении всех кратчайших маршрутов для всех возможных вершинных паросочетаний. Применим использованную выше методику, при этом нижние индексы могут быть опущены в связи с их равенством единице. В целях систематизации процесса поиска перечисление эквивалентных вершин упорядочим в сторону увеличения их номеров. Табл. 4 отражает последовательность изменения ее содержимого в процессе поиска всех кратчайших маршрутов из вершины 1 в остальные вершины графа:



или в окончательном виде

$$1(2^1(3^2), 7^1(4^2, 6^2), 8^1(5^2, 6^2), 9^1(3^2, 6^2))). \quad (5)$$

Из (5) составим перечень всех исходящих из вершины 1 кратчайших маршрутов:

$$(1 \rightarrow 2) = 1(2), \quad (1 \rightarrow 3) = 1(2(3), 9(3)), \quad (1 \rightarrow 4) = 1(7(4)), \quad (1 \rightarrow 5) = 1(8(5)), \\ (1 \rightarrow 6) = 1(7(6), 8(6), 9(6)), \quad (1 \rightarrow 7) = 1(7), \quad (1 \rightarrow 8) = 1(8), \quad (1 \rightarrow 9) = 1(9).$$

Таблица 4

1	2	3	4	5	6	7	8	9
0	1					1	1	1
		2						
			■		■			
				4				
			2		2			
				■				
				7				

Представим описания кратчайших маршрутов из всех остальных вершин графа в результирующей форме:

$$\begin{aligned}
 &2(1^1(8^2(5^3)), 3^1(4^2(5^3), 6^2(5^3)), 7^1(4^2(5^3), 6^2(5^3)), 9^1(6^2(5^3), 8^2(5^3))), \\
 &3(2^1(1^2, 7^2), 4^1(5^2, 7^2), 6^1(5^2, 7^2, 8^2), 9^1(1^2, 7^2, 8^2)), \\
 &4(3^1(2^2, 9^2), 5^1(8^2), 6^1(8^2, 9^2), 7^1(1^2, 2^2, 9^2)), \\
 &5(4^1(3^2(2^3), 7^2(2^3)), 6^1(3^2(2^3), 7^2(2^3), 9^2(2^3)), 8^1(1^2(2^3), 9^2(2^3))), \quad (6) \\
 &6(3^1(2^2), 4^1, 5^1, 7^1(1^2, 2^2), 8^1(1^2), 9^1(1^2, 2^2)), \\
 &7(1^1(8^2), 2^1(3^2), 4^1(3^2, 5^2), 6^1(3^2, 5^2, 8^2), 9^1(3^2, 8^2)), \\
 &8(1^1(2^2, 7^2), 5^1(4^2), 6^1(3^2, 4^2), 9^1(3^2)), \\
 &9(1^1, 2^1, 3^1(4^2), 6^1(4^2), 7^1(4^2), 8^1(5^2)).
 \end{aligned}$$

Нетрудно убедиться в том, что максимальное значение индекса на множестве полученных описаний представляет собой максимальное кратчайшее расстояние между двумя вершинами, т. е. определяет диаметр графа. Здесь диаметр графа равен кратчайшему расстоянию между вершинами 2 и 5: $d(2, 5) = 3$. Перечень диаметрально простых цепей [4, 5] рассмотренного графа включает в себя следующие маршруты:

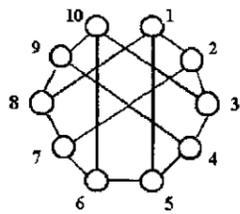
$$\begin{aligned}
 &2(1(8(5))), 2(3(4(5))), 2(3(6(5))), 2(7(4(5))), 2(7(6(5))), 2(9(6(5))), 2(9(8(5))), \\
 &5(4(3(2))), 5(4(7(2))), 5(6(3(2))), 5(6(7(2))), 5(6(9(2))), 5(8(1(2))), 5(8(9(2))).
 \end{aligned}$$

Анализируя полученные описания, несложно найти и многие другие структурные характеристики [4], например величину радиуса рассматриваемого графа, определить его центральные вершины, перечень радиальных простых цепей и др.

2.2. Поиск кратчайших непересекающихся маршрутов соединения. Как уже сообщалось выше, моделирование структурно живучей вычислительной системы основано на использовании графовой модели, в которой между множеством элементарных машин и множеством вершин графа, а также между множеством линий связи и множеством ребер графа существуют взаимно однозначные соответствия. Структурная живучесть ВС характеризуется совокупностью компонентов, оценивающих живучесть соединений всех вершинных паросочетаний. В качестве параметра, характеризующего потенциальное качество соединения (без учета алгоритмов маршрутизации, реконфигурации и т. п.), используется величина, определяемая в [1, 6] как проводимость. Под соединением пары вершин понимается множество всех непересекающихся кратчайших маршрутов между заданными вершинами; при необходимости вводится ограничение на максимальную длину маршрута. Задача поиска всех кратчайших непересекающихся маршрутов соединения пары вершин является одной из наиболее трудоемких даже для довольно небольших графов с числом вершин более 10. Именно эта задача в силу своей трудоемкости и послужила побудительной причиной введения скобочной формы и создания методик ее использования.

Итак, в качестве примера рассмотрим граф, представленный на рис. 7. Минимизированное представление данного графа может быть следующим:

$$G((1(5(4, 6)), 7(6(10)))(2(3(4, 10)), 8(9(4, 10)))).$$



$$G \begin{pmatrix} 1(2,5,8), & 6(5,7,10), \\ 2(1,3,7), & 7(2,6,8), \\ 3(2,4,10), & 8(1,7,9), \\ 4(3,5,9), & 9(4,8,10), \\ 5(1,4,6), & 10(3,6,9) \end{pmatrix}$$

Рис. 7

Задача поиска всех кратчайших непересекающихся маршрутов соединения пары вершин может быть сведена к приведенной выше задаче поиска кратчайших путей из вершин, смежных с истоком, ко второй вершине рассматриваемой пары. Накладываемое ограничение (поиск из смежных вершин) объясняется тем, что ни одна из вершин заданного паросочетания не может быть использована в качестве транзитной. Кроме того, ни одна из вершин, смежных с рассматриваемыми, также не может быть использована для транзитов при поиске кратчайших путей от любой другой вершины из множества смежных. Эти ограничения накладываются самой постановкой проблемы поиска кратчайших маршрутов, не имеющих пересечений.

В качестве примера рассмотрим процесс поиска таких маршрутов из вершины 1 ко всем остальным вершинам графа. Вершинами, смежными с истоком, являются вершины (2, 5, 8). Это видно из описания графа. Вычислим все кратчайшие маршруты из вершин 2, 5 и 8 описанным в п. 2.1 способом:

1	2	3	4	5	6	7	8	9	10
0	1			1			1		
1	0	1	2	3	2	1	2	3	2
1	3	2	1	0	1	2	3	2	3
1	2	3	2	3	2	1	0	1	2

Затененные в таблице столбцы соответствуют вершинам, запрещенным к использованию в качестве транзитных (в соответствии с наложенными выше ограничениями). Результатом поиска всех кратчайших маршрутов из вершин 2, 5 и 8 будет следующий:

$$\begin{aligned} &2(3^1(4^2(5^3, 9^3), 10^2(9^3)), 7^1(6^2(5^3), 8^2)), \\ &5(4^1(3^2(2^3), 9^2(8^3)), 6^1(7^2(2^3, 8^3), 10^2)), \\ &8(7^1(2^2, 6^2(5^3)), 9^1(4^2(3^3, 5^3), 10^2(3^3))). \end{aligned}$$

Используем полученные выражения для описания искомым маршрутов из вершины 1:

$$(1 \rightarrow 2) = 1(2^1, 5^1(4^2(3^3(2^4))), 6^2(7^3(2^4))), 8^1(7^2(2^3))).$$

В маршрутах $1(5^1(6^2(7^3(2^4))))$ и $1(8^1(7^2(2^3)))$ имеет место пересечение по вершине 7. Так как длина второго из этих маршрутов меньше длины первого (верхние индексы при вершине 2 равны соответственно 3 и 4), то, разрешив

транзит через вершину 7 для более короткого маршрута, необходимо запретить ее использование для более длинного. Окончательное выражение для описания искомым маршрутов из вершины 1 в вершину 2 имеет вид: $1(2^1, 5^1(4^2(3^3(2^4))), 8^1(7^2(2^3)))$. Опуская подробности поиска маршрутов в остальные вершины графа, получим

$$\begin{aligned}(1 \rightarrow 3) &= 1(2^1(3^2), 5^1(4^2(3^3)), 8^1(9^2(10^3(3^4))))), \\(1 \rightarrow 4) &= 1(2^1(3^2(4^3)), 5^1(4^2), 8^1(9^2(4^3))), \\(1 \rightarrow 5) &= 1(2^1(7^2(6^3(5^4))), 5^1, 8^1(9^2(4^3(5^4))))), \\(1 \rightarrow 6) &= 1(2^1(7^2(6^3)), 5^1(6^2), 8^1(7^2(6^3))).\end{aligned}$$

В соединении вершин 1 и 6 маршруты $1(2^1(7^2(6^3)))$ и $1(8^1(7^2(6^3)))$ имеют пересечение по вершине 7. Длины указанных маршрутов равны; запрет транзита через вершину 7 в любом из этих маршрутов (с целью исключения пересечения) приводит к его потере. Поэтому должен быть найден другой кратчайший путь, исключаяющий транзиты через вершины, уже используемые в других маршрутах этого соединения (в том числе и через вершину 7 в описываемом случае). К числу запрещенных для транзита вершин в рассмотренной выше таблице добавлен столбец, соответствующий вершине 7.

1	2	3	4	5	6	7	8	9	10
0	0			0		0			
0	0	1	2	0	3	0		3	2
0	0	3	2	0	3	0	0	1	2

В результате поиска получим

$$\begin{aligned}2(3^1(4^2(5^3, 9^3(8^4)), 10^2(6^3, 9^3)), 7^1) \text{ или} \\8(7^1, 9^1(4^2(5^3, 3^3(2^4))), 10^2(3^3, 6^3)).\end{aligned}$$

Используя эти выражения, представим итоговые описания всех кратчайших непересекающихся маршрутов между вершинами 1 и 6:

$$\begin{aligned}1(2^1(3^2(10^3(6^4))), 5^1(6^2), 8^1(7^2(6^3))) \text{ или} \\1(2^1(7^2(6^3)), 5^1(6^2), 8^1(9^2(10^3(6^4)))).\end{aligned}$$

Так как для остальных соединений подобных ситуаций не возникает, перечислим их без дополнительных комментариев:

$$\begin{aligned}(1 \rightarrow 7) &= 1(2^1(7^2), 5^1(6^2(7^3)), 8^1(7^2)), \\(1 \rightarrow 8) &= 1(2^1(7^2(8^3)), 5^1(4^2(9^3(8^4))), 8^1), \\(1 \rightarrow 9) &= 1(2^1(3^2(10^3(9^4))), 5^1(4^2(9^3)), 8^1(9^2)), \\(1 \rightarrow 10) &= 1(2^1(3^2(10^3)), 5^1(6^2(10^3)), 8^1(9^2(10^3))).\end{aligned}$$

Из приведенных описаний могут быть получены и другие полезные характеристики графа и его соединений, например, обход соединения $g(1, 7) = 4: 1(2^1(7^2(8^3(1^4))))$, окружение соединения $c(1, 7) = 5: 1(2^1(7^2(6^3(5^4(1^5))))$ или $1(8^1(7^2(6^3(5^4(1^5))))$. В отличие от [4] понятия обхода и окружения отнесены здесь не ко всему графу, а к его подграфам, включающим в себя совокупности вершин и ребер, образующих заданные соединения.

Для анализа структурной живучести большемасштабных вычислительных систем, и особенно для построения алгоритмов их статической (при деградации системы) и динамической (при отказах в подсистемах) реконфигураций, очень важным моментом является оперативное (актуализированное) внесение изменений в описание структуры, произошедших в результате отказов модулей ВС и (или) связей между ними. Очевидно, что для этого не обязательно проводить полный поиск всех маршрутов заново. Достаточно в описаниях исходящих маршрутов вычеркнуть отказавшие элементы и последующие участки цепей. Исчезнувшие в результате этих действий маршруты могут быть частично восполнены из оставшихся описаний, а частично сформированы в результате нового поиска. Обоснования таких возможностей даны в разд. 3 настоящей работы.

3. Оптимизация процесса поиска кратчайших маршрутов. Трудоемкость процесса поиска кратчайших маршрутов в графе может быть существенно снижена, если в каждой последующей итерации поиска будут учтены предшествующие результаты. Ниже сформулированы и обоснованы некоторые теоретические положения, позволяющие исключить из процесса поиска действия, ведущие к появлению дублированных результатов.

Теорема 1. Если в неориентированном графе маршрут $s \rightarrow t$ из вершины s в вершину t является кратчайшим, то маршрут $t \rightarrow s$ из t в s , представляющий собой последовательность вершин (ребер), обратную последовательности вершин (ребер) маршрута $s \rightarrow t$, также является кратчайшим.

Доказательство. В неориентированном графе по его определению любое ребро имеет одинаковый вес в любом из направлений. Следовательно, в силу коммутативности операции сложения сумма весов ребер (вершин), составляющих любой маршрут, равна сумме весов ребер (вершин), представляющих собой последовательность, обратную исходной. Таким образом, если маршрут $s \rightarrow t$ кратчайший и обладает длиной l , то обратный маршрут $t \rightarrow s$, состоящий из тех же вершин (ребер), обладает той же длиной l . Предположим, что существует другой маршрут ($t \rightarrow s$), обладающий длиной $l' < l$. В этом случае маршрут, обратный маршруту ($t \rightarrow s$), также должен обладать длиной $l' < l$. Однако маршрут $s \rightarrow t$ является кратчайшим, и по определению не существует других маршрутов с длиной, меньшей длины кратчайшего маршрута. Следовательно, предположение о возможности существования более короткого обратного маршрута является ложным.

Теорема 2. Если маршрут из вершины s в вершину t является кратчайшим, то любая простая цепь, входящая в этот маршрут, также является кратчайшим маршрутом.

Доказательство. Рассмотрим общий случай для любого связанного графа, в котором связи между вершинами взвешенны и могут быть как ориентированными (дуги), так и двунаправленными (ребра). Пусть при этом маршрут $s \rightarrow t$ из вершины s в вершину t является кратчайшим и число транзитных вершин в нем равно n . В общем случае он может быть не единственным, т. е. может существовать некоторое множество маршрутов между за-

данными вершинами, обладающих одинаковой и при этом минимальной длиной.

Множество транзитных вершин, входящих в состав заданного кратчайшего маршрута, обозначим через Ω , а сами вершины – через ω_i , где индекс указывает на порядковый номер в последовательности этих вершин на маршруте $s \rightarrow t$, причем $|\Omega| = n$. В целях сокращения последующих записей с каждой вершиной ω_i сопоставим подмножество $\Omega_i \subseteq \Omega, \Omega_i = \{\omega_{i+1}, \dots, \omega_n\}, 1 \leq i < n, \Omega_n = \Omega_{n-1} - \omega_n$. Совокупность всех простых цепей, составляющих заданный кратчайший маршрут, может быть представлена множеством

$$M(s \rightarrow t) = \{s \rightarrow \{\Omega, t\}, \omega_1 \rightarrow \{\Omega_1, t\}, \dots, \dots, \omega_i \rightarrow \{\Omega_i, t\}, \dots, \omega_{n-1} \rightarrow \{\Omega_{n-1}, t\}, \omega_n \rightarrow t\}.$$

Здесь $\omega_i \rightarrow \{\Omega_i, t\} = \{\omega_i \rightarrow \omega_{i+1}, \omega_i \rightarrow \omega_{i+2}, \dots, \omega_i \rightarrow \omega_n, \omega_i \rightarrow t\}$. Предположим, что хотя бы один из участков простой цепи $s \rightarrow t$, представляющий собой некоторый маршрут $\omega_i \rightarrow \omega_j$ из множества $M(s \rightarrow t)$, не является кратчайшим и может быть заменен более коротким маршрутом $(\omega_i \rightarrow \omega_j)'$, длина которого $l'_{ij} < l_{ij}$. Здесь $0 < i \leq n, \omega_0 = s, i < j \leq n+1, \omega_{n+1} = t$. Однако в этом случае маршрут из s в t изменит свою конфигурацию: $(s \rightarrow t)'$, $M(s \rightarrow t)'$. Длина нового маршрута определится выражением $L(s \rightarrow t)' = L(s \rightarrow t) - l_{ij} + l'_{ij}$ и окажется меньше, чем $L(s \rightarrow t)$, т. е. $L(s \rightarrow t)' < L(s \rightarrow t)$. Это утверждение противоречит как нашему условию ($s \rightarrow t$ – кратчайший маршрут), так и определению кратчайшего маршрута. Следовательно, ни один участок заданного маршрута не может быть заменен более коротким участком. Таким образом, любой участок, определенный на множестве всех маршрутов, являющихся частями кратчайшего, также является кратчайшим.

Следствие 1 (из теоремы 2). Если хотя бы один участок простой цепи из s в t не является кратчайшим, то и сама цепь $s \rightarrow t$ в целом, и ее компоненты, включающие в себя этот участок, также не являются кратчайшими.

Предположение о ложности данного утверждения означает, что в состав кратчайшего маршрута могут входить участки, не являющиеся кратчайшими, что, в свою очередь, противоречит доказанной выше теореме 2.

Данное следствие теоретически обосновывает методику поиска маршрутов, предложенную в разд. 2 настоящей работы, суть которой состоит в исключении из составляемого описания тех участков, длина которых на момент поиска перестала быть минимальной.

Следствие 2 (из теорем 1 и 2). Любая простая цепь, являющаяся частью кратчайшего маршрута между двумя вершинами неориентированного графа, также является кратчайшим маршрутом между концевыми вершинами этой цепи.

Следствие 3 (из теоремы 2). Кратчайший маршрут между любыми двумя вершинами графа содержит в себе как минимум $(n+1)(n+2)/2$ кратчайших маршрутов, где n – число транзитных вершин в заданном маршруте.

Для ориентированного графа величина $|M(s \rightarrow t)|$ представляет собой сумму арифметической прогрессии

$$|M(s \rightarrow t)| = \sum_{i=0}^n \sum_{j=i+1}^{i+1} |\{\omega_i \rightarrow \omega_j\}| = (n+1)(n+2)/2$$

и не требует дополнительных пояснений. Для смешанного же графа с дугами и ребрами $|M(s \rightarrow t)| > (n+1)(n+2)/2$.

Следствие 4 (из теоремы 1 и следствия 2). Кратчайший маршрут между любыми двумя вершинами неориентированного графа содержит в себе $(n+1)(n+2)$ кратчайших маршрутов, где n – число транзитных вершин в заданном маршруте.

Учитывая доказанные выше положения, можно существенно сократить процесс поиска кратчайших маршрутов графа. Например, помимо всех кратчайших маршрутов, исходящих из вершины 1, выражение (3) определяет следующие кратчайшие маршруты: $2(3_{18}^{18})$, $4(5_5^5)$, $7(2_2^2)$, $7(2_2^2(3_{18}^{20}))$, $7(4_4^4)$, $7(4_4^4(5_5^9))$, $7(6_{14}^{14})$. Для неориентированных графов выигрыш еще существеннее, так как в этот перечень добавляются также все обратные маршруты и их компоненты. Например, из выражения (6) для вершины 2 следует:

$$\downarrow 2(1^1(8^2(5^3))), 3^1(4^2(5^3)), 7^1(4^2(5^3), 6^2(5^3)), 9^1(6^2(5^3), 8^2(5^3))).$$

Из того же выражения (5) и из выражения, полученного для входящих в вершину 2 кратчайших маршрутов, могут быть выделены следующие кратчайшие маршруты:

$$\begin{aligned} &1(8^1), 1(8^1(5^2)), 3(4^1, 6^1), 3 \rightarrow 5 = 3((4^1, 6^1)5^2), 4(3^1, 5^1, 7^1), 5(4^1, 6^1, 8^1), \\ &5 \rightarrow 3 = 5((4^1, 6^1)3^2), 5(6^1(7^2, 9^2)), 6(3^1, 5^1, 7^1, 9^1), 7(4^1, 6^1), \\ &7 \rightarrow 5 = 7((4^1, 6^1)5^2), 8(1^1, 5^1). \end{aligned}$$

Таким образом, информация о многих маршрутах содержится в имеющихся описаниях в неявной форме и может быть использована в последующих процедурах поиска. Достаточно часто для ориентированных графов и практически всегда для неориентированных графов встречаются ситуации, когда найденные описания кратчайших маршрутов уже содержат полные описания искомым маршрутов для одной или нескольких еще не рассмотренных вершин. Например, из выражений (3), (4) для ориентированного графа (см. рис. 3) видно, что перечень маршрутов из вершины 7 полностью перекрывается описаниями кратчайших маршрутов из всех остальных вершин. Из выражений (6) для неориентированного графа (см. рис. 6) следует, что если известны описания для любых пяти вершин, то описания маршрутов из остальных четырех могут быть сформированы из уже полученных.

Заключение. Предложен новый способ формального описания графов и их компонентов (подграфов, маршрутов, соединений). В отличие от известной матричной формы описание графа в скобочной форме, помимо информации о смежности (инцидентности) в явном виде, несет в себе информацию о связности несмежных элементов (вершин, дуг, ребер), о коммуникационных качествах графа. При этом матричная форма может служить сырьевой основой для предложенного в работе более компактного и информативного представления графа. Процесс такого преобразования несложен в алгоритмической формализации и может быть автоматизирован по описанной в работе методике. Предложенный способ является необходимой логической базой для создания формализованного языка описания графов и структур вычислительных систем.

Рассмотрены возможности использования предложенных способов в исследованиях коммуникационных свойств графов и живучих вычислительных систем. Впервые поставлена и решена задача оптимизации графа путем поиска избыточных связей и их ликвидации. Даны методики поиска кратчайших маршрутов в графе и кратчайших непересекающихся маршрутов в его соединениях. Последнее имеет большое значение для анализа структурной живучести вычислительных систем. Предполагается, что использование предложенных методик значительно повысит оперативность алгоритмов реконфигурации системы при ее деградации, обусловленной как естественными причинами, так и внешними факторами.

Доказаны теоремы о совокупностях кратчайших маршрутов в графах, выведены следствия из этих теорем. Введенные теоретические положения служат обоснованием предложенных в работе путей оптимизации процессов автоматизированного анализа коммуникационных качеств графа.

СПИСОК ЛИТЕРАТУРЫ

1. Мелентьев В. А. О живучести вычислительных систем // Электронное моделирование. 1985. 7, № 4.
2. Мелентьев В. А. Алгоритм инвариантного программирования структуры вычислительной системы с учетом живучести соединений // Теория и применение моделирующих систем: Тр. Всесоюз. науч.-техн. конф. «Моделирование-85». Теория. Средства. Применение. Киев: Наукова думка, 1985.
3. Вычислительные сети и сетевые протоколы: Пер. с англ. Д. Дэвиса, Д. Барбера, У. Прайса, С. Соломонидеса. М.: Мир, 1982.
4. Оре О. Теория графов: Пер. с англ. М.: Наука, 1980.
5. Харари Ф. Теория графов: Пер. с англ. М.: Мир, 1973.
6. Мелентьев В. А. Оценка потенциальной и структурной живучести вычислительных систем // Электронное моделирование. 1995. 17, № 1.
7. Melentiev V., Griaznov N. Analysis of potential and structural robustness of distributed computer systems // Proc. 15th IMACS World Congress on Sci. Computation Modelling and Applied Mathematics. Berlin, August 1997. V. 6: Application in Modelling and Simulation. Berlin: Wissenschaft & Technik Verlag, 1997.
8. Кристофидес Н. Теория графов. Алгоритмический подход: Пер. с англ. М.: Мир, 1978.
9. Майника Э. Алгоритмы оптимизации на сетях и графах: Пер. с англ. М.: Мир, 1981.

*Институт физики полупроводников СО РАН,
E-mail: melva@isp.nsc.ru*

*Поступила в редакцию
19 августа 1999 г.*