АЛГОРИТМЫ АНАЛИЗА СТРУКТУРНОЙ ИНФОРМАЦИИ
(Вычислительные системы)
1984 год Выпуск 103

УДК 519.1

МЕТОДЫ И АЛГОРИТМЫ АНАЛИЗА СИММЕТРИЙ ГРАФОВ

В.А.Скоробогатов, П.В.Хворостов

Решение многих задач обработки структурной информации в таких областях, как химия и электроника, тесно связано с проблемами распознавания изоморфизма и определения топологических симметрий графов. Поскольку алгоритмы, решающие эти задачи с полиномиальной оценкой сложности неизвестны, а переборные методы требуют слишком больших затрат машинного времени, возникает необходимость разрабатывать алгоритмы, с одной стороны, не требующие слишком больших вычислительных затрат, а с другой - позволяющие получить решение для достаточно широкого класса графов.

В данной работе приведено описание алгоритма поиска симметрий, имеющего полиномиальную оценку сложности для встречающихся в практических приложениях графов. Для графов типа сильно регулярных [16] в алгоритме предусмотрено осуществление этапа направленного перебора.

Рассматривается возможность применения этого алгоритма для распознавания изоморфизма и канонизации графов.

Основные определения. Пусть $G(V, X)$ - граф с множеством вершин V и множеством ребер X, $|V| = p$, $|X| = q$, а $\Gamma(G)$ - его группа автоморфизмов [1] - совокупность подстановок на множестве $V(G)$, сохраняющих смежность.

Вершины $v, u \in V(G)$ называются эквивалентными относительно $\Gamma(G)$ (обозначается $v \underset{\Gamma}{\sim} u$), если $\exists \varphi \in \Gamma(G) \mid \varphi v = u$. Такое отношение действительно является отношением эквивалентности, поскольку для любых $v, u, w \in V(G)$ выполняются следующие свойства:

1) $v \underset{\Gamma}{\sim} v$ - рефлексивность ($ev = v$, e - тождественная подстановка),

2) $v \underset{\Gamma}{\sim} u \Rightarrow u \underset{\Gamma}{\sim} v$ – симметричность ($\varphi v = u \Rightarrow \varphi^{-1} u = v$),

3) $v \underset{\Gamma}{\sim} u,\ u \underset{\Gamma}{\sim} w \Rightarrow v \underset{\Gamma}{\sim} w$ – транзитивность ($\varphi_1 v = u,\quad \varphi_2 u = w \Rightarrow \varphi_2 \circ \varphi_1\ v = w$).

Орбитами графа (областями транзитивности $\Gamma(G)$) называются классы эквивалентности θ_i, $i = \overline{1,s}$, разбиения множества вершин по отношению к $\Gamma(G)$. Такое разбиение обозначим $\hat{V}_\theta(G) = \{\theta_1, \ldots, \theta_s\}$. Будем говорить, что граф имеет симметрии, если существует хотя бы одна нетривиальная орбита, т.е. орбита, состоящая более чем из одной вершины графа.

Если рассматривать симметрию в терминах матрицы смежности $A(G)$, то условие $\varphi v = u$ означает, что матрица $A'(G)$, переупорядоченная в соответствии с подстановкой φ (переставлены соответствующие строки и столбцы), совпадает с $A(G)$.

Методы определения симметрий. Проблема нахождения орбит графа является по сложности эквивалентной распознаванию изоморфизма графов. Согласно [2,3] проблемы эквивалентны, если существование алгоритма полиномиальной сложности, решающего одну из них, влечет существование алгоритма полиномиальной сложности, решающего другую. Действительно, задачу определения орбит можно свести к нахождению изоморфизма графа на себя, а графы G и H изоморфны тогда и только тогда, когда в орбитах графа $G \cup H$ присутствуют одновременно вершины G и H.

В настоящее время неизвестны полиномиальные алгоритмы распознавания изоморфизма, а следовательно, и определения орбит графов.

Так как мы имеем дело с конечными графами, можно легко получить алгоритм, основанный на переупорядочении вершин графа и гарантирующий решение задачи. В таком алгоритме фиксируется матрица смежности $A(G)$ и рассматриваются матрицы $A'(G)$, полученные из $A(G)$ всевозможными одновременными перестановками строк и столбцов. Если $A = A'$, то соответствующие этим матрицам нумерации вершин графа образуют подстановку из $\Gamma(G)$. Таким образом, проделав все $p!$ перестановок, можно получить в явном виде группу автоморфизмов графа и его орбиты. Но большое число требуемых перестановок в матрицах делает алгоритм практически неприменимым.

Этим обусловлено возникновение многочисленных алгоритмов определения орбит графов [4–8], использующих разбиения множества вершин на классы эквивалентности относительно различных характе -

ристик вершин графа, например, степень вершины, число циклов и путей различной длины, проходящих через данную вершину, и т.д. Характеристики такого рода называются инвариантами вершин, и условие $\mathrm{inv}(v) = \mathrm{inv}(u)$, $v, u \in V$, является необходимым условием существования автоморфизма $\varphi \in \Gamma(G)$ такого, что $\varphi v = u$. До сих пор неизвестен набор инвариантов, определяющих граф с точностью до изоморфизма [9], поэтому легко сконструировать граф, для которого классы эквивалентности, найденные подобными алгоритмами, будут содержать по несколько орбит графа. Таким примером может служить граф G [I0], изображенный на рис.I и полученный из графа $L(K_4)$ (реберный граф полного графа K_4) заменой каждого ребра x деревом $T(x)$.

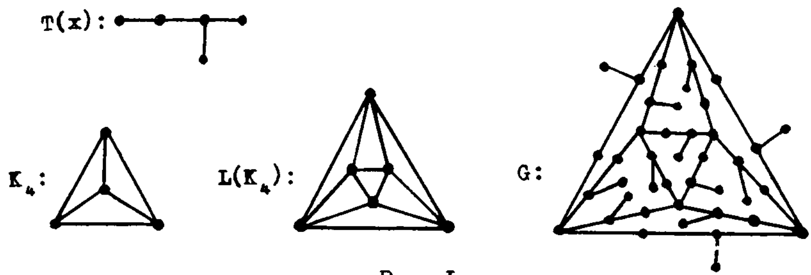

Рис. I

Так как группа автоморфизмов графа $L(K_4)$ транзитивна на вершинах и ребрах, то все характеристики типа числа путей и циклов, проходящих через данную вершину, у вершин с одинаковой степенью в графе G будут совпадать. Поэтому классы эквивалентности будут состоять из вершин с равными степенями: I2 вершин степени I; I2 вершин степени 2; I2 вершин степени 3 и 6 вершин степени 4 - всего четыре класса. На самом деле этот граф имеет I4 орбит, каждая из которых содержит три вершины.

Группу и симметрии графа $G(V, X)$ можно также найти посредством анализа клик графа модульного произведения [I5,29].

Модульным произведением (графом соответствий) обыкновенных [25] графов $G(V, X)$ и $H(U, Y)$ называется граф $S(W, E) = G \triangledown H$, множеством вершин которого является декартово произведение множеств V и U, т.е. $W = V \times U = \{w = (v, u),\ v \in V,\ u \in U\}$, а множество ребер E определяется следующим отношением смежности вершин $w = (v, u)$ и $w' = (v', u')$ графа S:

$$(w,w')\in E \to [(v,v')\in X,\ (u,u')\in Y]\vee$$
$$\vee[(v,v')\notin X,\ (u,u')\notin Y,\ v\neq v',\ u\neq u'] .$$

Граф $S(W,E)$ содержит полную информацию о возможных соответствиях вершин графа G вершинам графа H, и основное его свойство состоит в том, что каждая клика графа S соответствует максимальному общему подграфу графов G и H [I5,29]. Иначе говоря, если $\langle w_1,\dots,w_k\rangle$ – клика графа S, $w_i=(v_i,u_i)$, $v_i\in V$, $u_i\in U$, $i=\overline{1,k}$, то подграфы $\langle v_1,\dots,v_k\rangle\subseteq G$ и $\langle u_1,\dots,u_k\rangle\subseteq H$ максимальны и изоморфны.

Таким образом, если рассматривать модульное произведение $G\nabla G$, то каждая клика $K=\langle w_1,\dots,w_p\rangle$, $w_i=(v,v')$, $i=\overline{1,p}$, $v,v'\in V$, $p=|V|$, будет соответствовать подстановке $\tau(K)=$
$=\begin{pmatrix} v_1\dots v_p\\ v'_1\dots v'_p\end{pmatrix}$, являющейся автоморфизмом, и все множество клик порядка p будет определять группу $\Gamma(G)$.

Однако следует отметить тот факт, что задача поиска клик в графе является NP-полной [28].

Примером другого подхода к определению симметрий графа могут служить алгоритмы [II,I2] построения канонической нумерации вершин графа. Под канонической здесь понимается такая нумерация вершин графа, что двоичное число δ, полученное при выписывании подряд в одну строку всех строк соответствующей матрицы смежности, максимально.

В этих алгоритмах процесс построения возможных нумераций вершин графа можно представить в виде построения корневого дерева T, корень которого $T(0)=\emptyset$ в [II], либо $T(0)=\{V\}$ в [I2], а листья соответствуют различным нумерациям вершин. Анализ симметрий осуществляется посредством сравнения матриц смежности, соответствующих листьям дерева. Для сокращения числа листьев (в худшем случае оно может достигать значения $p!$, например, для графа K_p) используются приемы регулярного построения уровней дерева T_k, $1\le k\le p$, и отсечения некоторых ветвей.

В алгоритме [II] построение дерева T происходит следующим образом. В качестве уровня $T(1)$ рассматриваются все вершины графа, а уровни $T(k)$, $1<k\le p$, строятся так, что $A_k=\max A_{k-1,r}$, где A_k – матрица смежности порожденного подграфа на k вершинах, а

$$A_{k-1,r} = \begin{pmatrix} A_{k-1} & \begin{matrix} a_{1,r} \\ \vdots \\ a_{k-1,r} \end{matrix} \\ a_{r,1} \ \dots & a_{r,r} \end{pmatrix}$$

Матрица A больше матрицы A', если $\delta(A) > \delta(A')$. При совпадении матриц смежности на листьях дерева, вершины, образующие циклы подстановок, объединяются в классы эквивалентности. Если $v_i \underset{T}{\sim} v_i'$, $v_i, v_i' \in V$, то одну из ветвей $(v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_p)$ или $(v_1, \dots, v_{i-1}, v_i', v_{i+1}', \dots, v_p')$ можно исключить из дерева T. Применение такого отсечения ветвей для графа K_p потребует рассмотрения только p листьев дерева T вместо p! всех возможных нумераций.

В алгоритме [12] уровни дерева состоят из упорядоченных разбиений множества вершин графа на классы. Условием принадлежности вершин $v, u \in V$ к одному и тому же классу является совпадение числа соседей у вершин v и u в каждом из классов разбиения (более подробно построение таких разбиений будет описано при рассмотрении процедуры итеративной классификации вершин). Если число классов в текущем разбиении меньше p, то некоторая произвольная вершина из неединичного класса выделяется в отдельный класс (ветвление дерева) и т.д., до получения разбиения с числом классов равным p. Такие разбиения образуют листья дерева. Симметрии графа определяются, как и в алгоритме [11], при сравнении матриц смежности, соответствующих листьям.

Для отсечения ветвей дерева используется следующий прием. Пусть τ - разбиение множества вершин в листе дерева, получаемое через последовательность текущих разбиений $\pi_1 \geq \pi_2 \geq \dots \geq \pi_k = \tau$ ($\pi_i \geq \pi_j$, если каждый класс разбиения π_j содержится в некотором классе разбиения π_i). Каждому разбиению τ ставится в соответствие список $\Delta(\tau) = \{\lambda(\pi_1), \dots, \lambda(\pi_k)\}$, где $\lambda(\pi_i)$, $i = \overline{1,k}$, равно числу ребер графа, концы которых лежат в одном и том же классе разбиения π_i. Очевидно, что разбиения τ_1 и τ_2 не могут быть эквивалентными (т.е. порождать одинаковые матрицы смежности), если $\Delta(\tau_1) \neq \Delta(\tau_2)$, а каноническую нумерацию графа следует искать только среди множества разбиений, у которых $\Delta(\tau) = \Delta^*$, где $\Delta^* = \max\limits_{\tau}(\Delta(\tau))$, старшинство векторов Δ - лексикографическое.

Таким образом, если при построении текущего разбиения π_i' оказалось, что $\lambda(\pi_i') < \lambda(\pi_i)$ для некоторого уже построенного разбиения π_i, то ветвь π_i' удаляется из дерева.

Для рассмотренных алгоритмов [II,I2] не удается оценить число листьев и соответствующих им матриц смежности, которые необходимо рассмотреть в процессе анализа симметрий и определения канонической нумерации графа. Однако из схемы устранения ветвей дерева видно, что алгоритм [II] наиболее эффективно должен работать на графах с достаточно богатой группой автоморфизмов, так как большое число эквивалентных вершин позволяет значительно сократить дерево перебора. В то время как для обработки графов с малым числом симметричных вершин будет требоваться много времени на безрезультатное сравнение различных матриц смежности, соответствующих листьям, число которых для таких графов особенно велико. Так, для сравнительно небольших кубических графов с $p = 14$ и тождественной группой автоморфизмов число таких матриц зачастую более двухсот [I3,I4].

Алгоритм [I2], наоборот, более эффективно работает на графах с малым числом эквивалентных вершин, так как в этом случае разбиения множества вершин графа часто имеют различные значения функции Δ, что приводит к устранению большого числа ветвей дерева перебора. Более трудоемкими для этого алгоритма будут являться транзитивные графы, например K_p, где значения функции $\Delta(\tau)$ будут совпадать для всего множества разбиений в листьях дерева.

В предлагаемом алгоритме наряду с традиционным разбиением множества вершин графа на классы эквивалентности относительно некоторых инвариантов используются также сведения об орбитах стабилизаторов вершин, позволяющие "улучшить" начальное разбиение, полученное с помощью инвариантов.

Алгоритм определения симметрий графа. Будем рассматривать непомеченные связные графы без петель и кратных ребер, не являющиеся сильно регулярными [I6,I7]. В предлагаемом алгоритме определения орбит графа [I8,I9] используется идея построения и анализа разбиений множества вершин графа $G(V,X)$ на классы эквивалентности $top(V) = \{V_1^t, \ldots, V_{n_1}^t\}$ и $low(V) = \{V_1^l, \ldots, V_{n_2}^l\}$, для которых выполнены условия

$$\forall k \in \{1,\ldots,s\} \ \exists i \in \{1,\ldots,n_1\} \ |\theta_k \cap V_i^t = \theta_k \qquad (1)$$

и

$$\forall i \in \{1,\ldots,n_2\} \ \exists k \in \{1,\ldots,s\} \ |V_i^l \cap \theta_k = V_i^l, \qquad (2)$$

т.е. каждая орбита графа целиком вкладывается в некоторый класс разбиения top(V) , и каждый класс разбиения low(V) целиком вкладывается в некоторую орбиту.

Рассмотрим процесс построения этих разбиений.

П о с т р о е н и е р а з б и е н и я top(V) . При построении разбиения top(V) учитывается информация о положении и частичных степенях вершин в слоях относительного разбиения графа [20].

ОПРЕДЕЛЕНИЕ. Относительным разбиением графа $G(V,X)$ для вершины v называется разбиение $\hat{V}(v) = \{V_k(v),\ k = \overline{0,e(v)},\ v' \in V_k \Leftrightarrow d(v,v') = k\}$, где $e(v)$ – эксцентриситет вершины v, $d(v,v')$ – расстояние между вершинами v и v', $V_0 \equiv \{v\}$. Классы $V_k, k=\overline{0,e(v)}$, называются слоями графа.

Каждой вершине $v_i \in V(G)$ ставится в соответствие характеристика $\alpha(v_i) = (\alpha^0_{ij}, \alpha^1_{ij}, \alpha^2_{ij}, \alpha^3_{ij})$; $i, j = \overline{1,p}$, где α^m_{ij}, $m = 1,2,3$, равно числу ребер, соединяющих вершину $v_i \in V_k(v_j)$ в разбиении $\hat{V}(v_j)$ с вершинами из слоев V_{k-1}, V_k и V_{k+1} соответственно, а $\alpha^0_{ij} = k$ (рис.2). Если $k = 0$, то $\alpha^1_{ij}, \alpha^2_{ij} = 0$, а если $k = e(v_j)$, то $\alpha^3_{ij} = 0$.

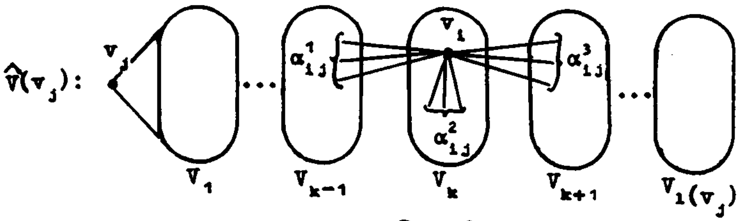

Рис. 2

Пример вычисления $\alpha(v_i)$ приведен на рис. 3.

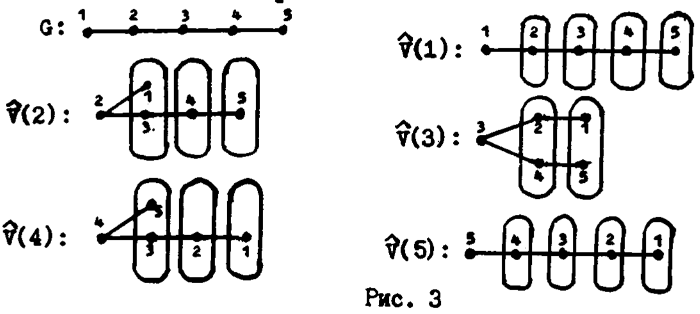

Рис. 3

Вершины будут иметь следующие характеристики:

$$\alpha(1)=\begin{bmatrix}0&0&0&1\\1&1&0&0\\2&1&0&0\\3&1&0&0\\4&1&0&0\end{bmatrix},\alpha(2)=\begin{bmatrix}1&1&0&1\\0&0&0&2\\1&1&0&1\\2&1&0&1\\3&1&0&1\end{bmatrix},\alpha(3)=\begin{bmatrix}2&1&0&1\\1&1&0&1\\0&0&0&2\\1&1&0&1\\2&1&0&1\end{bmatrix},\alpha(4)=\begin{bmatrix}3&1&0&1\\2&1&0&1\\1&1&0&1\\0&0&0&2\\1&1&0&1\end{bmatrix},\alpha(5)=\begin{bmatrix}4&1&0&0\\3&1&0&0\\2&1&0&0\\1&1&0&0\\0&0&0&1\end{bmatrix}.$$

Строки матриц $\alpha(v_i)$, $i=\overline{1,p}$, упорядочиваются лексикографически. Вершины v_i и v_j принадлежат одному и тому же классу, если $\alpha(v_i)=\alpha(v_j)$, а классы упорядочиваются по убыванию α. Так, для графа на рис.3

$$\alpha(1)=\alpha(5)=\begin{bmatrix}4&1&0&0\\3&1&0&0\\2&1&0&0\\1&1&0&0\\0&0&0&1\end{bmatrix},\quad \alpha(2)=\alpha(4)=\begin{bmatrix}3&1&0&1\\2&1&0&1\\1&1&0&1\\1&1&0&1\\0&0&0&2\end{bmatrix},\quad \alpha(3)=\begin{bmatrix}2&1&0&1\\2&1&0&1\\1&1&0&1\\1&1&0&1\\0&0&0&2\end{bmatrix}$$

и порядок классов следующий: $V_1=\{1,5\}$, $V_2=\{2,4\}$, $V_3=\{3\}$.

К полученному упорядоченному разбиению $\hat{V}=\{V_1,\dots,V_n\}$ применяется процедура итеративной классификации вершин [9], суть которой состоит в том, что каждой вершине $v\in V$ ставится в соответствие список $D(v)=(d_1(v),\dots,d_n(v))$, где $d_i(v)$, $i=\overline{1,n}$, есть число вершин из класса V_i, смежных с v. По этим спискам каждый класс V_i может быть подразбит на подмножества, состоящие из вершин с одинаковыми списками. Эти подмножества (только внутри классов V_i) упорядочиваются в соответствии с лексикографическим старшинством списков принадлежащих им вершин, и, таким образом, получается измельчение начального разбиения. Если все вершины в каждом подмножестве имеют одинаковые списки, то подразбиения классов не произойдет (стабилизация разбиения). Если измельчение классов произошло, то к текущему разбиению вновь применяем процедуру итеративной классификации до тех пор, пока не наступит стабилизация. Полученное разбиение и образует $\mathrm{top}(V)$.

ТЕОРЕМА I. Разбиение $\mathrm{top}(V)$ удовлетворяет условию (I).

ДОКАЗАТЕЛЬСТВО. Достаточно показать, что если вершины $v,v'\in V(G)$, $v\neq v'$, принадлежат некоторой орбите графа, то они попадут в один и тот же класс разбиения $\mathrm{top}(V)$. Рассмотрим относительное разбиение $\hat{V}(u)$, $u\in V$. Пусть $v\in V_i(u)$, поскольку автоморфизмы сохраняют смежность, то $\exists\varphi\in\Gamma(G)\,|\,\varphi v=v',\varphi u=u'$ и $\alpha^m_{vu}=\alpha^m_{v'u'}$, $m=\overline{0,3}$ (рис.4). Такое свойство справедливо для всех $u\in V(G)$, поэтому $\alpha(v)=\alpha(v')$ и вершины v,v' на этом этапе построения разбиения будут отнесены к одному и тому же классу.

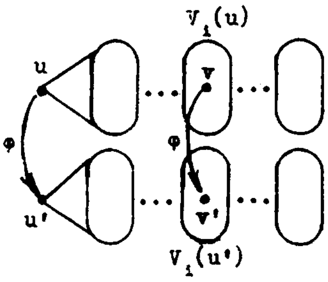

Рис. 4

Так как каждая орбита графа целиком лежит в некотором классе полученного разбиения и автоморфизмы сохраняют смежность, то в процессе итеративной классификации вершин графа используемые в качестве инвариантов списки $D(v)$ и $D(v')$ совпадают на каждой итерации. Следовательно, вершины v и v' будут принадлежать одному и тому же классу разбиения $top(V)$.

Построение разбиения $low(V)$. При построении разбиения $low(V)$ используются сведения об орбитах стабилизаторов вершин графа. Стабилизатором $v \in V(G)$ называется подгруппа подстановок $\Gamma(v) \subseteq \Gamma(G)$, оставляющих неподвижной вершину v. Орбитами стабилизатора $\Gamma(G)$ называются его области транзитивности.

Для получения орбит стабилизатора $\Gamma(v)$ используется следующий приближенный метод. Для фиксированного $v \in V$ строится относительное разбиение $\hat{V}(v)$, к которому применяется процедура итеративной классификации вершин. Полученное разбиение обозначается $\hat{V}_*(v) = \{V_*^1, \ldots, V_*^m\}$. Если каждый класс V_*^i разбиения $\hat{V}_*(v)$ целиком вкладывается в некоторый класс V_j^t разбиения $top(V)$, то разбиение $\hat{V}_*(v)$ остается без изменений. В противном случае, т.е. если $V_*^i \cap V_j^t \neq \emptyset$ и $V_*^i \cap V_j^t \neq V_*^i$, то V_*^i разбивается на подклассы $V_*^{i_1} = V_*^i \setminus (V_j^t \cap V_*^i)$ и $V_*^{i_2} = V_j^t \cap V_*^i$. Старшинство классов определяется в соответствии со старшинством классов в $top(V)$. Подкласс $V_*^{i_1}$ также проверяется на вложение в некоторый класс разбиения $top(V)$. Например, пусть разбиение $\hat{V}_*(1) = \{(1),(2,3,5,8),(4,6,7)\}$, а разбиение $top(V) = \{(1,2,3),(4,5),(6,7,8)\}$. Тогда класс (2,3,5,8) будет сначала подразбит на подклассы (2,3) и (5,8), а затем подкласс (5,8) – на (5) и (8). Класс (4,6,7) в свою очередь разобьется на подклассы (4) и (6,7).

Проверив и подразбив аналогичным образом все классы разбиения $\hat{V}_*(v)$ и проделав процедуру итеративной классификации вершин, получим некоторое новое разбиение, которое обозначим $\hat{V}_\Gamma(v)$.

Необходимость подразбиения классов в разбиении $\hat{V}_*(v)$ в соответствии с классами top(V) обусловлена тем, что каждая орбита $\Gamma(v)$ всегда лежит внутри некоторой орбиты графа.

Классы разбиения $\hat{V}_\Gamma(v)$ будем считать орбитами $\Gamma(v)$.

Предлагаемый метод построения орбит стабилизаторов является приближенным, так как не удается определить класс графов (кроме деревьев [21]), для которых он действительно приводит к получению орбит стабилизаторов. Но в проводимых исследованиях и практических задачах [13,22,23] не было встречено ни одного контрпримера.

Предположим, что в данном классе графов мы имеем разбиения $\hat{V}_\Gamma(v)$ на орбиты стабилизаторов для всех $v \in V(G)$. Множество таких разбиений служит базой для построения разбиения low(V). Рассмотрим $\hat{V}_\Gamma(v)$ и $\hat{V}_\Gamma(u)$, $v,u \in V$, $v \neq u$. Если некоторый класс $V^i_\Gamma(v) \subset \hat{V}_\Gamma(v)$ пересекается с некоторым классом $V^j_\Gamma(u) \subset \hat{V}_\Gamma(u)$: $V^i_\Gamma(v) \cap V^j_\Gamma(u) \neq \emptyset$, $V^i_\Gamma(v) \neq V^j_\Gamma(u)$, то вершины, принадлежащие этим классам, попадают в одну и ту же орбиту графа. Действительно, пусть вершины $w_1, w_2 \in V^i_\Gamma(v)$, а вершины $w_2, w_3 \in V^j_\Gamma(u)$. Так как каждый класс в этих разбиениях есть орбита стабилизатора, то $\exists \varphi_1$, $\varphi_2 \in \Gamma(G) \mid \varphi_1 w_1 = w_2$, $\varphi_2 w_2 = w_3$. Значит, $\varphi_2 \circ \varphi_1 w_1 = w_3$, т.е. существует автоморфизм, переводящий w_1 в w_3 .

Классу, состоящему из объединения классов $V^i_\Gamma(v)$ и $V^j_\Gamma(u)$, $V^i_\Gamma(v) \cap V^j_\Gamma(u) \neq \emptyset$, присваиваем номер, равный $\min(i,j)$.

Разбиение, полученное в результате операции объединения классов двух разбиений $\hat{V}_\Gamma(v)$ и $\hat{V}_\Gamma(u)$ при их непустых пересечениях, обозначим $\hat{V}_\Gamma(v) \cup \hat{V}_\Gamma(u)$.

В качестве low(V) используется разбиение, полученное в результате последовательного выполнения операции объединения разбиений на орбиты стабилизаторов: $\mathrm{low}(V) = \bigcup_{v \in V} \hat{V}_\Gamma(v)$. Очевидно, что для данного класса графов верна следующая

ТЕОРЕМА 2. Разбиение low(V) удовлетворяет условию (2).

<u>Анализ разбиений top (V) и low(V).</u> После того, как разбиения top(V) и low(V) построены, возможны следующие ситуации.

1. Каждый класс разбиения top(V) содержит только одну вершину графа $n_1 = p$. В этом случае граф не имеет симметрий (жесткий) и его обработка прекращается.

2. Некоторый класс V^*_i разбиения top(V) содержит только одну вершину: $V^*_i = \{v\}$, $i \in \{1,\ldots,n_1\}$, $n_1 < p$. В этом случае орбиты графа совпадают с орбитами стабилизатора $\Gamma(v)$.

3. Разбиения $top(V)$ и $low(V)$ совпадают: $V_i^t = V_i^l$, $i=\overline{1,n_1}$, $n_1 = n_2 = p$. Так как для разбиений $top(V)$ и $low(V)$ выполнены соотношения (I) и (2), то $\hat{V}_\theta(G) = top(V) = low(V)$.

4. В разбиении $low(V)$ существуют классы, являющиеся подмножествами классов разбиения $top(V)$: $\forall i \in \{1,\dots,n_2\}$ $\exists j \in \{1,\dots,n_1\} | V_i^l \subseteq V_j^t$, $n_1 < n_2$. При этом совпадающие классы разбиений $low(V)$ и $top(V)$ являются орбитами графа, а для вершин, принадлежащих различным классам разбиения $low(V)$, но одному классу в разбиении $top(V)$, осуществляется проверка существования автоморфизма, переводящего вершины из различных классов разбиения $low(V)$ друг в друга.

Пусть класс $V_i^t \subseteq top(V)$ представлен в разбиении $low(V)$ совокупностью классов $V_{j_1}^l, \dots, V_{j_r}^l$; $V_i^t = \bigcup\limits_{k=1}^{r} V_{j_k}^l$.

Достаточно рассматривать только по одной вершине из каждого класса $V_{j_k}^l$, $k=\overline{1,r}$, так как в любом таком классе вершины графа переводятся одна в другую автоморфизмами, и наличие автоморфизма, переводящего друг в друга хотя бы по одной вершине из различных классов, свидетельствует о том, что все вершины этих классов входят в одну орбиту графа.

Зафиксируем произвольные вершины $u \in V_{j_k}^l$ и $w \in V_{j_m}^l$, $1 \le k < m \le r$, и построим относительное разбиение $\hat{V}(u,w)$, такое разбиение определяется аналогично относительному разбиению $\hat{V}(v)$, только слой V_0 состоит из вершин u и w. Относительные разбиения для некоторых подмножеств вершин графа $V' \subset V$ рассматривались в [20,24]. К разбиению $\hat{V}(u,w)$ применяется процедура итеративной классификации вершин, и полученное разбиение проверяется на вложение классов в классы разбиения $top(V)$. Эта проверка аналогична описанной выше проверке для $\hat{V}_*(v)$. Полученное в результате разбиение обозначим $\hat{V}_*(u,w)$.

Если вершины u и w оказались в одном классе разбиения $\hat{V}_*(u,w)$, то в разбиении $low(V)$ происходит объединение классов и вместо $V_{j_k}^l$, $V_{j_m}^l$ теперь рассматривается один класс $V_{j_k}^l \cup V_{j_m}^l$ с номером j_k.

Такой анализ "парных" разбиений $\hat{V}_*(u,w)$ выполняется для всех вершин - представителей u и w (при объединении классов остается только один представитель на весь новый класс). Если в результате получено, что $low(V) = top(V)$, то, как и в случае 3, имеем $\hat{V}_\theta(G) = low(V) = top(V)$, иначе $\hat{V}_\theta(G) = low(V)$.

Структура алгоритма поиска симметрий графа изображена на рис.5, где блоки имеют следующее функциональное назначение:

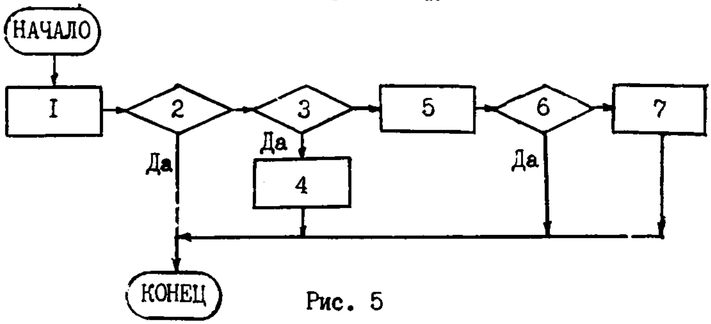

Рис. 5

I – построение разбиения top(V), 2 – проверка разбиения top(V) на дискретность (дискретным называется разбиение, каждый класс которого содержит ровно одну вершину), 3 – проверка существования в top(V) хотя бы одного единичного класса, 4 – построение орбит стабилизатора единичного класса разбиения top(V), 5 – построение разбиения low(V), 6 – проверка совпадения разбиений top(V) и low(V), 7 – выделение совпадающих классов разбиений top(V) и low(V) и анализ парных разбиений для представителей классов разбиения low(V).

Оценка сложности алгоритма. В процессе выполнения алгоритма можно выделить следующие этапы, определяющие трудоемкость решения задачи:

I) построение разбиения top(V),

2) построение разбиения low(V),

3) анализ разбиений top(V) и low(V).

На каждом из этих этапов применяется процедура итеративной классификации вершин, которая состоит из определения числа ребер, инцидентных каждой вершине графа,– число требуемых операций $O(q)$ и сортировки вершин по классам – число операций $O(p\log_2 p)$. Так как в худшем случае возможно p итераций и $q \sim p^2$, то оценка сложности процедуры итеративной классификации составит $Q_0 = p(O(q) + O(p\log_2 p)) = O(p^3)$.

Для построения разбиения top(V) требуется построить все относительные разбиения графа $\hat{V}(v)$, $v \in V$ (сложность построения каждого разбиения $O(q)$), в каждом разбиении определить частичные степени всех вершин (сложность $O(q)$), упорядочить вершины (сложность $O(p\log_2 p)$ и применить процедуру итеративной классификации. Значит, общая сложность построения $top(V)$ будет $Q_1 = p(O(q) + O(p\log_2 p)) + Q_0 = O(p^3)$.

Сложность построения разбиения $low(V)$ определяется необходимостью в худшем случае применить процедуру итеративной классификации для всех относительных разбиений $\hat{V}(v)$, $v \in V$, следовательно, $Q_2 = p(O(q) + Q_0) + O(p^2) = O(p^4)$, где $O(p^2)$ – слагаемое, возникающее при сравнении и объединении классов в разбиениях $\hat{V}_\Gamma(v)$, $v \in V$.

При анализе разбиений $top(V)$ и $low(V)$ наиболее трудоемким является гипотетический случай, когда $top(V)$ состоит из единственного класса, содержащего все вершины графа, а $low(V)$ состоит из единичных классов. Здесь возникает необходимость в худшем случае построить $\binom{p}{2}$ разбиений $\hat{V}(v,v')$, $v,v' \in V$, к которым затем применить процедуру итеративной классификации. Трудоемкость этого этапа составит $Q_3 = \binom{p}{2} \cdot (O(q) + Q_0) + O(p^3) = O(p^5)$.

Таким образом, общая оценка сложности алгоритма будет $Q = Q_1 + Q_2 + Q_3 = O(p^5)$.

Следует отметить, что в проводимых исследованиях [10,13,18, 23] для всех графов (кроме сильно регулярных) $\hat{V}_\Theta = top(V)$, и практическая оценка сложности алгоритма составляет $O(p^3)$.

Определение симметрий помеченных мультиграфов. Пусть $G(V,X)$ – граф с помеченными вершинами и взвешенными ребрами. Метки вершин и веса ребер можно задать при помощи однозначных отображений $m\colon V \to R' \subset R$, $f\colon X \to R'' \subset R$, где R – множество действительных чисел.

Алгоритм определения симметрий помеченных графов отличается от алгоритма для обыкновенных графов построением разбиения $top(V)$ и процедурой итеративной классификации вершин.

При построении разбиения $top(V)$ осуществляется этап предварительной классификации вершин по меткам – строится упорядоченное разбиение $\hat{V}_m$ множества $V(G)$ на классы эквивалентности относительно значений функции m. Вершины $v,u \in V$ принадлежат одному и тому же классу разбиения $\hat{V}_m$ тогда и только тогда, когда $m(v) = m(u)$. Класс $V_i \subset \hat{V}_m$ старше класса $V_j \subset \hat{V}_m$, если $m(v) > m(u)$, $\forall v \in V_i$, $\forall u \in V_j$.

Для вершин, попавших в один класс разбиения $\hat{V}_m$, вычисляются характеристики α, на основе которых осуществляется подразбиение классов разбиения $\hat{V}_m$: если $\alpha(v) \neq \alpha(u)$, $v,u \in V_i \subset \hat{V}_m$, то вершины v и u относятся к различным подклассам в классе V_i. Порядок подклассов внутри V_i определяется лексикографически старшинством матриц α.

К полученному подразбиению разбиения $\hat{V}_m$ применяется процедура итеративной классификации, а окончательное разбиение используется в качестве $top(V)$.

Поскольку вершины из одной орбиты графа должны иметь одинаковые метки, то, очевидно, что для определенного таким образом разбиения $top(V)$ справедлива теорема I.

Рассмотрим процедуру итеративной классификации вершин для графов со взвешенными ребрами. Пусть $x_1,\dots,x_k$ – ребра, инцидентные вершине $v \in V$ и вершинам из класса V_i текущего разбиения. В случае обыкновенных графов в качестве i-го элемента списка $D(v)$ рассматривалось число $d_i(v) = k$, а в случае графов со взвешенными ребрами i-м элементом списка $D(v)$ служит вектор $(f(x_1),\dots$ $\dots,f(x_k))$. При сравнении списков вершин такие векторы предварительно упорядочиваются по убыванию.

<u>Использование симметрий для распознавания изоморфизма и построения канонического кода графов.</u> При распознавании изоморфизма, хранении и поиске графов удобно использовать следующий прием. В каждом классе попарно изоморфных графов выбирается один граф, называемый каноническим видом любого графа данного класса. После этого распознавание изоморфизма графов сводится к построению и сравнению канонических видов [30,31].

Каноническим видом графа будем называть граф, матрица смежности которого переупорядочена в соответствии со свойствами вершин, не зависящими от их исходной нумерации. Нумерация вершин, соответствующая каноническому виду графа, называется канонической.

Для построения канонической нумерации вершин графа $G(V,X)$ рассматривается упорядоченное разбиение $\hat{V}_\theta(G) = \{V_1,\dots, V_s\}$ множества вершин графа на орбиты, $V_i = \theta_i$, $i = \overline{1,s}$. Граф G может быть как обыкновенным, так и помеченным мультиграфом.

Если $s = p$ (в каждом классе V_i содержится ровно одна вершина), то полученный порядок вершин в разбиении $\hat{V}_\theta$ и определяет каноническую нумерацию.

Если $s < p$, то выбираются первый по порядку класс V_i, $|V_i| > 1$, и произвольная вершина $v \in V_i$. К разбиению $\{V_1,\dots,V_i \setminus v,\dots, V_s, v\}$ применяется процедура итеративной классификации вершин. Если число полученных классов меньше p, то снова выбираем первый не -

единичный класс и повторяем процесс до тех пор, пока каждый класс разбиения не будет содержать ровно одну вершину. Порядок вершин в таком разбиении будет определять каноническую нумерацию графа.

Очевидно, что неизоморфные графы будут иметь различные канонические виды, ибо в противном случае из совпадения матриц смежности немедленно следовал бы изоморфизм, а канонические виды изоморфных графов совпадают, поскольку на каждом шаге процесса построения канонической нумерации сохраняется отношение изоморфизма между вершинами из соответствующих классов текущих разбиений.

Для компактного хранения канонических видов графов используется запись на языке ОГРА-30 [26], в которой описание графа представлено в виде его покрытия множеством цепей. Каждая цепь формируется следующим образом. Выбирается вершина v_1 с минимальным номером и записывается последовательность ребер: $(v_1,v_2),(v_2,v_3)$, ..., (v_{i-1},v_i),..., где v_i - вершина с минимальным номером, смежная с v_{i-1}. В этой последовательности также учитываются метки вершин и веса ребер. После записи в последовательность ребро помечается. Если у вершины v_i нет непомеченных инцидентных ребер, то цепь заканчивается. Далее выбирается вершина с минимальным номером, имеющая непомеченные инцидентные ребра, и начинается формирование новой цепи. Граф полностью покрыт, когда помечены все его ребра. Например, для графа G на рис.6 будет сформирована запись G*1CH$_3$- 8N- 7CO- 6N- 3CH- 2CH$_2$(2)*4CH$_2$- 5CO- 6*4- 9CO- 8** .

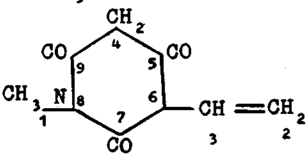

Рис. 6

Здесь * означает конец цепи, ** - конец описания графа. Метку вершины достаточно описать один раз. Вес ребра указывается в скобках.

Алгоритм канонизации можно применять и для распознавания изоморфизма графов. Пусть требуется определить, изоморфны ли графы G(V,X) и H(U,Y) . Для этого достаточно просто сравнить их канонические виды. При совпадении графы изоморфны и их канонические нумерации образуют подстановку изоморфизма, а при несовпадении - неизоморфны.

В случае, когда графы G и H неизоморфны, для ускорения работы алгоритма оказывается полезной проверка соответствия текущих разбиений множеств V и U . Если на некотором шаге построения канонических нумераций в разбиениях $\hat{V}$ и $\hat{U}$ различные количества классов либо классы с одинаковыми номерами содержат неодинаковое число вершин, либо вершины в этих классах имеют различные характеристики, то между разбиениями $\hat{V}$ и $\hat{U}$ нет соответствия, а графы G и H заведомо неизоморфны. Поэтому дальнейшее построение и сравнение канонических видов можно не производить.

Оценка сложности алгоритма канонизации будет отличаться от оценки сложности алгоритма определения симметрий только добавкой оценки сложности процедуры итеративной классификации, так как существует прямая зависимость между числом итераций до стабилизации в этой процедуре и числом необходимых выделений вершин в отдельный класс при канонизации.

Определение симметрий сильно регулярных графов. Использование инвариантов вершин графа для определения симметрий оказывается малоперспективным при анализе сильно регулярных и подобных им графов [3,16], так как значения инвариантов будут равными у неэквивалентных вершин.

Связный однородный граф $G(V,X)$ называется сильно регулярным, если каждая пара смежных вершин имеет ровно a общих соседей, а каждая пара несмежных вершин – ровно b общих соседей (рис.7), т.е. $\forall v,u \in V$, $|O(v) \cap O(u)| = a$, если $(v,u) \in X$; $|O(v) \cap O(u)| = b$, если $(v,u) \notin X$, где $O(v)$, $O(u)$ – окружения вершин v и u соответственно.

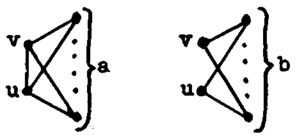

Рис.7

Для анализа таких графов применяется алгоритм направленного перебора (метод ветвей и границ) с использованием процедуры итеративной классификации вершин. Алгоритм основан на проверке существования автоморфизма $\varphi \in \Gamma(G)$ такого, что $\varphi v_1 = u_1$ для фиксированной пары вершин $v_1, u_1 \in V$ $v_1 \neq u_1$. Задача проверки существования такого автоморфизма сводится к нахождению изоморфизма графа на себя с фиксированным переходом вершины v_1 в вершину u_1.

Зафиксируем вершину v_1 и рассмотрим разбиения $\{V \setminus v_1, v_1\}$, $\{V \setminus u_1, u_1\}$, $v_1 \neq u_1$, и применим к ним процедуру итеративной классификации. Полученные разбиения обозначим $\hat{V}^{v_1} = \{V_1^{v_1}, \ldots, V_n^{v_1}\}$ и $\hat{V}^{u_1} = \{V_1^{u_1}, \ldots, V_n^{u_1}\}$. Если между разбиениями $\hat{V}^{v_1}$ и $\hat{V}^{u_1}$ нет соответствия, то вершины v_1 и u_1 неэквивалентны (не принадлежат одной орбите), и переходим к рассмотрению следующей вершины $w \neq v_1, u_1$.

Если между разбиениями $\hat{V}^{v_1}$ и $\hat{V}^{u_1}$ установлено соответствие, то возможны два случая:

1) $n = p$, тогда матрицы смежности графа G, переупорядоченные в соответствии с разбиениями $\hat{V}^{v_1} = \{v_1, v_2, \ldots, v_p\}$ и $\hat{V}^{u_1} = \{u_1, \ldots, u_p\}$, совпадают, значит, подстановка $\tau = \begin{pmatrix} v_1 v_2 \ldots v_p \\ u_1 u_2 \ldots u_p \end{pmatrix}$ является автоморфизмом графа и вершины, входящие в один и тот же независимый цикл подстановки τ, эквивалентны;

2) $n < p$, тогда выбираем классы $V_i^{v_1}$ и $V_i^{u_1}$, $i \in \{1, \ldots, n\}$, содержащие более одной вершины. Пусть $v' \in V_i^{v_1}$, $u' \in V_i^{u_1}$, рассмотрим разбиения $\{V_1^{v_1}, \ldots, V_i^{v_1} \backslash v', \ldots, V_n^{v_1}, v'\}$, $\{V_1^{u_1}, \ldots, V_i^{u_1} \backslash u_1, \ldots$ $\ldots, V_n^{u_1}, u_1\}$ (такую операцию выделения вершины в отдельный класс будем называть, как и в [27], ветвлением), к которым применяем процедуру итеративной классификации. При несоответствии полученных разбиений выбирается следующая вершина $u'' \in V_i^{u_1}$, $u'' \neq u'$, и процедура итеративной классификации применяется к разбиению $\{V_1^{u_1}, \ldots, V_i^{u_1} \backslash u'', \ldots, V_n^{u_1}, u''\}$. Если рассмотрен весь класс $V_i^{u_1}$, а соответствие между разбиениями, полученными из $\hat{V}^{v_1}$ и $\hat{V}^{u_1}$ с помощью ветвления, не установлено, то вершины v_1 и u_1 неэквивалентны. Для ускорения просмотра класс V_i выбирается минимальной мощности.

Если соответствие между полученными разбиениями установлено и $n < p$, то вновь выполняем ветвление с итеративной классификацией до тех пор, пока не получится $n = p$.

Пусть на некотором шаге возникло несоответствие разбиений, тогда возвращаемся к классу, в котором произошло последнее ветвление, и рассматриваем все варианты ветвлений в этом классе. Этап возвращения к последней точке ветвления обеспечивает полный перебор всех возможных соответствий между вершинами, поэтому в результате выполнения алгоритма будет найдена орбита графа, содержащая вершину v_1.

После того, как все вершины, эквивалентные v_1, найдены, приступаем к анализу оставшихся вершин графа, причем существенным образом используется информация об уже полученных эквивалентностях.

Так, если в классах с одинаковыми номерами содержится разное число вершин из одной и той же орбиты, то между разбиениями нет соответствия. Также между разбиениями не будет соответствия, если для ветвления выбрать вершины из разных орбит. Если установлена эквивалентность вершин v_1 и u_1, которые в свою очередь входят в некоторые классы эквивалентности $(v_1, v_2, \dots, v_k)$ и $(u_1, u_2, \dots, u_l)$ то, очевидно, все вершины этих классов эквивалентны между собой: $v_i \underset{\Gamma}{\sim} u_j$, $i = \overline{1,k}$, $j = \overline{1,l}$.

Алгоритм заканчивает работу, когда рассмотрены все возможные соответствия между вершинами графа. В худшем случае (жесткий граф) требуется просмотреть $\binom{p}{2}$ возможных соответствий пар вершин.

Алгоритмы определения симметрий, распознавания изоморфизма и получения канонического кода графов реализованы на языке ПЛ/I для ОС ЕС ЭВМ. Данные об их работе на конкретных графах приведены в [10].

Литература

1. ХАРАРИ Ф. Теория графов. -М.: Мир, 1973. -300 с.

2. КУК С. Сложность процедур вывода теорем. -Кибернетич.сб., нов.сер., 1975, вып. 12, с. 5-15.

3. ЗЕМЛЯЧЕНКО В.М., КОРНЕЕНКО Н.М., ТЫШКЕВИЧ Р.И. Проблема изоморфизма графов. -Записки научных семинаров ЛОМИ, 1982, т. 118. Теория сложности вычислений. I, с. 83-152.

4. GOLENDER V.E., DRBOGLAV V.V., ROSENBLIT A.B. Graph potentials method and its applications for chemical information processing.- J.Chem.Inf.Sys.,1981,v.21,p.196-204.

5. CORNEIL D.G., GOTLIEB C.C. An efficient algorithm for graph isomorphism.-Journal ACM,1970,v.17,p.51-64.

6. MORGAN H.L. The generation of a unique machine description for chemical structures - a technique developed at Chemical Abstract Service.- J.Chem.Doc.,1965,p.107-113.

7. RANDIC M. On canonical numbering of atoms in a molecule and graph isomorphism. - J.Cehm.Inf.Comp.Sci.,1977,p.171-180.

8. BALABAN A.T., MEKENYAN O., BONCHEV D. Ordering and numbering of atoms, and unique topological representation of chemical structures by algorithms НОС. -В кн.: Использование вычислительных машин в спектроскопии молекул и химических исследованиях: Тезисы докл. VI Всесоюз. конф., Новосибирск, 1983, с. 87-89.

9. READ R.C., CORNEIL D.G. The graph isomorphism disease. - J.of Graph Theory, 1977,N 1,p.339-363.

10. ДОБРЫНИН А.А. Сравнение программ обработки структурной информации. -Настоящий сборник, с. 90-99.

II. Алгоритм приведения конечных неориентированных графов к каноническому виду /Арлазаров В.Л., Зуев И.И., Усков А.В., Фараджев И.А. -Журн. вычислит. мат. и мат.физ., 1974, т. 14, № 3, с.737-743.

12. McKAY B.D. Computing algorithms and canonical labeling of graphs.-Lect.Notes.Math.,1977,v.686,p.223-232.

13. ХВОРОСТОВ П.В. Симметрия кубических графов.-В кн.: Машинные методы обнаружения закономерностей, анализа структур и проектирования (Вычислительные системы, вып. 92). Новосибирск, 1982, с. 80-141.

14. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Орбиты, клики, канонизация. -В кн.: Методы и программы решения оптимизационных задач на графах и сетях: Тезисы докл. Всесоюз. совещания. Новосибирск, 1980, с.85-87.

15. ВИЗИНГ В.Г. Сведение проблемы изоморфизма и изоморфного вложения к задаче нахождения неплотности. -В кн.: Труды Ш Всесоюзной конференции по проблемам теоретической кибернетики. Новосибирск, 1974, с. 124.

16. BOSE R.C. Strongly regular graphs, partial geometries and block designs.-Pacific J.Math.,1963,v.13,p.389-419.

17. WEISFEILER B. On construction and identification of graphs. - Lect.Notes Math.,1976, v.558.- 297 p.

18. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Об анализе комбинаторных симметрий химических структур.-В кн.: Использование вычислительных машин с спектроскопии молекул и химических исследованиях: Тезисы докл. У Всесоюз. конф., Новосибирск, 1980, с.10-11.

19. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Анализ симметрий графов.-В кн.: Методы и программы решения оптимизационных задач на графах и сетях. Ч.2: Тезисы докл. П Всесоюз. совещания. Новосибирск, 1982, с. 138-139.

20. СКОРОБОГАТОВ В.А. Относительные разбиения и слои графов. -В кн.: Вопросы обработки информации при проектировании систем (Вычислительные системы, вып. 69). Новосибирск, 1977, с.6-9.

21. CORNEIL D.G. Graph isomorphism. - Ph.D.thesis. Univ. of Toronto, 1968. - 135 p.

22. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Анализ структурной химической информации при предсказании биологической активности.-В кн.: Методы и средства обработки сложноструктурированной семантически насыщенной графической информации: Тезисы докл. I Всесоюз. конф. Горький, 1983, с. 28.

23. ХВОРОСТОВ П.В. Генерация подстановок при построении тестов. -В кн.: Вычислительная техника и дискретная математика: Тезисы докл. региональной научно-технической конф., посвященной Дню радио, Новосибирск, 1983, с. 27.

24. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Анализ метрических свойств графов. -В кн.: Методы обнаружения закономерностей с помощью ЭВМ (Вычислительные системы, вып. 91). Новосибирск, 1981, с. 1-20.

25. ЗЫКОВ А.А. Теория конечных графов. -Новосибирск: Наука, 1969. - 543 с.

26. КОЧЕТОВА А.А., СКОРОБОГАТОВ В.А.,ХВОРОСТОВ П.В. Язык описания структурной информации ОГРА-30.-В кн.: Машинные методы обнаружения закономерностей, анализа структур и проектирования (Вычислительные системы, вып. 92). Новосибирск, 1982, с. 70-79.

27. КУРЕЙЧИК В.М., КОРОЛЕВ А.Г. Об одном методе распознавания изоморфизма графов. -Кибернетика, 1977, № 2, с. 82-87.

28. ГЭРИ М., ДЖОНСОН Д. Вычислительные машины и труднорешаемые задачи. -М.: Мир, 1982. - 416 с.

29. СКОРОБОГАТОВ В.А. Нахождение общих частей в семействах графов.-В кн.: Прикладные задачи на графах и сетях: Материалы Всесоюз. совещания. Новосибирск, 1981, с. 117-132.

30. Алгоритмические исследования в комбинаторике.-М.: Наука, 1978. - 188 с.

31. COMPUTER investigation of cubic graphs /Bussemaker F.S., Cobeljic S., Cvetkovic D.M., Seidel J.J. - T.H.Rept. 76-WSK-01.Technol.Univ.of Eindhoven,1976. - 87 p.

Поступила в ред.-изд.отд.
3 июля 1984 года