АЛГОРИТМИЧЕСКИЙ АНАЛИЗ СТРУКТУРНОЙ ИНФОРМАЦИИ
(Вычислительные системы)
1985 год Выпуск II2

УДК 519.I

О РЕШЕНИИ ЗАДАЧИ ПОИСКА НАИБОЛЬШИХ ПЕРЕСЕЧЕНИЙ ГРАФОВ НА ОСНОВЕ АНАЛИЗА ПРОЕКЦИЙ ПОДГРАФОВ МОДУЛЬНОГО ПРОИЗВЕДЕНИЯ

Ю.Е. Бессонов

Задачи, связанные с поиском наибольших пересечений двух графов, возникают в различных приложениях дискретной математики и известны как "трудноршаемые" [I]. В ряде приложений теории графов (например, в химии, биологии, электронике) графы, используемые в качестве моделей, как правило, образуют сравнительно узкие классы. Например, почти все графы, представляющие молекулярные структуры в химии, планарны, имеют степени вершин, ограниченные сверху числом 4, обладают сравнительно большой неплотностью. Наличие структурных особенностей у графов позволяет надеяться на возможность построения алгоритмов поиска их пересечений, значительно менее трудоемких по сравнению с общим случаем.

В настоящей работе для поиска пересечений двух графов используется метод, состоящий в том, что исходная задача сводится к задаче поиска максимальных клик в модульном произведении этих графов [2,3], которая решается при помощи рекурсивного разбора [4].

В статье исследованы свойства подграфов модульных произведений, определяемых вполне несвязными (либо полными) подграфами в исходных графах (§4). Эти свойства позволяют сокращать объем вычислений при реализации схемы рекурсивного разбора, в связи с чем такая структурная особенность графов, как наличие достаточно большого независимого множества вершин в каждом из подграфов, характерная для молекулярных графов в химии, для эволюционных деревьев в биологии и для моделей схем в электронике, дает существенно меньшую трудоемкость поиска пересечений по сравнению с общим случаем.

§I. Основные определения

В статье рассматриваются конечные неориентированные графы без петель и кратных ребер[ж)]. Примем следующие обозначения:

$G(V,X)$ – граф с множеством вершин V и множеством ребер X;

$p(G)$ – порядок графа G (число его вершин);

$G'(V',X')\subseteq G(V,X)$ – граф G' является подграфом в G, т.е. $V' \subseteq V$ и X' – максимально возможное подмножество X; будем говорить также, что G' порожден множеством вершин V', и обозначать это в виде $G' = \langle V' \rangle$;

$\Gamma_G(v)$ – окружение вершины v в графе G, т.е. множество всех вершин, смежных с v;

$\alpha(G)$ – неплотность графа G, т.е. порядок наибольшего вполне несвязного подграфа в G.

Вполне несвязным называется граф, не имеющий ребер. Множество вершин вполне несвязного подграфа называется независимым. Кликой называется максимальный по включению полный подграф. Под пересечением графов G_1 и G_2 будем понимать любой изоморфизм максимальных по включению подграфов $G_1' \subseteq G_1$ и $G_2' \subseteq G_2$. Порядком пересечения назовем порядок графов G_1' и G_2'. Наибольшим будем считать пересечение, имеющее максимальный порядок. Понятия и обозначения теории графов, не определяемые далее в тексте, можно найти в [5].

§2. Задачи поиска пересечений графов

В данной работе предлагаются алгоритмы для решения задач следующего класса.

<u>Задача I.</u> Определить, существует ли пересечение графов G_1 и G_2, имеющее порядок не менее k, где k – заданное число.

<u>Задача 2.</u> Определить порядок наибольшего пересечения графов G_1 и G_2.

<u>Задача 3.</u> Найти наибольшее пересечение графов G_1 и G_2.

<u>Задача 4.</u> Перечислить все наибольшие пересечения графов G_1 и G_2.

Задачи перечислены в порядке возрастания "информативности" решений, т.е. решение каждой последующей задачи дает решение всех

ж) Результаты настоящей статьи можно распространить и на случаи помеченных мультиграфов, а также орграфов.

предыдущих. Задача I является NP-полной [I] и полиномиально эквивалентна задаче 2. Каждая из перечисленных задач имеет самостоятельное прикладное значение.

§3. Модульные произведения графов

Операции модульного произведения были введены в работах [2,3] для сведения задач распознавания изоморфизма, изоморфного вхождения и поиска пересечений графов к задаче выделения клик. Модульным произведением $G_1 \nabla G_2$ графов $G_1(V_1, X_1)$ и $G_2(V_2, X_2)$ называется граф $G(V, X)$ с множеством вершин $V = V_1 \times V_2$ и множеством ребер $X = \{\{u,v\} \mid u=(u',u''),\ v=(v',v''),\ u',v' \in V_1,\ u'',v'' \in V_2,\ u' \neq v' \ \&\ u'' \neq v'' \ \&\ (\{u',v'\} \in X_1 \ \&\ \{u'',v''\} \in X_2 \vee \{u',v'\} \notin X_1 \ \&\ \{u'',v''\} \notin X_2)\}$. Отметим, что настоящее определение дано для непомеченных униграфов*), но это не уменьшает общности дальнейшего изложения.

Для решения задачи поиска максимальных пересечений графов используется следующее основное свойство модульных произведений: каждой клике $\langle\{(u_1,v_1),(u_2,v_2),\ldots,(u_n,v_n)\}\rangle$ (где $u_i \in V_1$, $v_i \in V_2$, $i=\overline{1,n}$) соответствует пересечение $\varphi = \begin{pmatrix} u_1 u_2 \ldots u_n \\ v_1 v_2 \ldots v_n \end{pmatrix}$ графов G_1 и G_2, и, наоборот, каждому пересечению графов G_1 и G_2 соответствует клика в $G_1 \nabla G_2$.

ПРИМЕР. Для $G_1 = K_2 \cup K_1$, $G_2 = P_3$ модульное произведение $G = K_1 \cup C_8$ показано на рис.I. Клики $\langle\{w_1, w_5\}\rangle$ и $\langle\{w_1, w_9\}\rangle$ в G соответствуют пересечениям $\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}$ и $\begin{pmatrix} u_1 u_3 \\ v_1 v_3 \end{pmatrix}$ исходных графов.

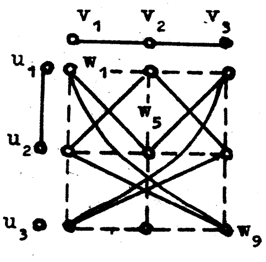

Рис.I

Вершины модульного произведения поместим в узлы прямоугольной решетки на плоскости таким образом, чтобы горизонтальные ряды занимали вершины из множеств $\{u\} \times V_2$ (где $u \in V_1$), а вертикальные ряды – вершины из множеств $V_1 \times \{v\}$, где $v \in V_2$ (см.рис.I). Из определения модульного произведения непосредственно вытекает следующее свойство: любые две вершины, лежащие в одном ряду, несмежны.

Порядок модульного произведения равен произведению порядков исходных графов. Наибольшее число клик $n!$ имеют модульные про –

---

*) Операция модульного произведения обобщается на случай помеченных графов с кратными ребрами [5] и ориентированных графов [3].

изведения вида $K_n \nabla K_n$ (или $\bar{K}_n \nabla \bar{K}_n$). Отсюда видно, что непосредственная реализация метода, предложенного в [2,3] для поиска наибольших пересечений графов, будет практически неприменимой, ввиду большого времени счета для графов, порядок которых превосходит 10.

§4. Подграфы модульного произведения. Проекции

Пусть $W = \{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$ - некоторое подмножество вершин модульного произведения. Проекцией множества W на граф G_1 назовем множество $U_1 = \{u_1, u_2, \ldots, u_n\} \subseteq V_1$, а на граф G_2 - множество $U_2 = \{v_1, v_2, \ldots, v_n\} \subseteq V_2$. Проекциями подграфа $\langle W \rangle$ на G_1 и G_2 будем считать подграфы $\langle U_1 \rangle \subseteq G_1$ и $\langle U_2 \rangle \subseteq G_2$. Рангом множества W назовем число r, равное максимальному количеству вершин, не лежащих попарно в одном ряду. Пусть $W_r \subseteq W$ - такое множество и U_1^r и U_2^r - его проекции. Множество W_r определяет подстановку φ: $U_1^r \to U_2^r$, при этом $\langle U_1^r \rangle \cong \langle U_2^r \rangle$ тогда и только тогда, когда $\langle W_r \rangle$ - клика.

Предположим, что обе проекции U_1 и U_2 множества W состоят из вершин, попарно несмежных между собой (т.е. $\langle U_1 \rangle$ и $\langle U_2 \rangle$ - вполне несвязные подграфы). Тогда любое подмножество $W_r \subseteq W$ ранга r порождает полный подграф в $G_1 \nabla G_2$, который будет максимальной[x)] кликой в $\langle W \rangle$ и который определяет подстановку φ: $U_1^r \to U_2^r$. Кроме того, граф $\langle W \rangle$ обладает следующими свойствами:

а) любые две вершины, не лежащие в одном ряду, смежны;

б) любые две вершины, лежащие в одном ряду, несмежны.

Отметим, что указанными свойствами будет обладать и подграф $\langle W \rangle \subseteq G_1 \nabla G_2$, обе проекции которого являются полными подграфами в G_1 и G_2.

Подграф модульного произведения, обладающий свойством "а", будем называть насыщенным. Граф назовем насыщенным, если множество его вершин можно расположить в позициях некоторой регулярной решетки на плоскости таким образом, чтобы выполнялись условия "а" и "б". Насыщенные графы обладают свойствами, позволяющими значительно сокращать перебор при решении задачи поиска наибольших пересечений графов.

---

x) Максимальной называется клика графа, имеющая наибольшее число вершин.

§5. Свойства насыщенных графов

Следующее утверждение будет использовано нами при поиске максимальных клик в насыщенных графах, а также при вычислении ранга подмножеств вершин модульного произведения.

ТЕОРЕМА I. *В насыщенном графе каждая вершина с максимальной степенью принадлежит максимальной клике.*

ДОКАЗАТЕЛЬСТВО. Пусть вершины насыщенного графа размещены в позициях регулярной решетки на плоскости так, что выполнены условия "а" и "б". Доказательство будем вести от противного. Пусть v – вершина с максимальной степенью. Предположим, что она не принадлежит ни одной максимальной клике. Рассмотрим некоторую максимальную клику K, пусть она состоит из k вершин. Вершина v лежит в одном ряду в точности с двумя вершинами v_1 и v_2 клики K, причем сами v_1 и v_2 не лежат в одном ряду (см. рис. 2). Действительно, если v не лежит в одном ряду с вершинами из K, то граф $v+K$ – полный и, значит, K не клика. Если v лежит в одном ряду только с одной вершиной, скажем с v_1, то $v+(K-v_1)$ – максимальная клика, содержащая v. Противоречие. Таким образом, v должна быть смежна с $(k-2)$ вершинами $v_3, v_4, \ldots, v_k$ клики K. Существует множество S, состоящее из l, $(l \geq 1)$, (не принадлежащих K) вершин, с которыми смежна вершина v (в противном случае степень v равна $k-2$ и любая вершина из K имеет большую степень). Имеем следующее $\deg v = k-l+2$; каждая из l вершин $u_i \in S$ лежит в одном ряду хотя бы с одной вершиной из K (иначе K не клика); ни одна из вершин множества S не лежит в одном ряду с v. Рассмотрим два случая.

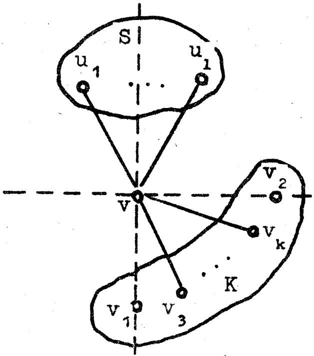

Рис. 2

I. Хотя бы одна вершина из множества S лежит в одном ряду с вершинами v_1 и/или v_2. Тогда, будучи смежной с v, эта вершина смежна также по меньшей мере с $(k-2)$ вершинами $v_3, \ldots, v_k$, с которыми смежна v, а значит, v принадлежит максимальной клике. Противоречие исходному предположению.

2. Ни одна вершина из S не лежит в одном ряду ни с v_1, ни с v_2. Но тогда, например, v_1 смежна с этими l вершинами из S, поэтому $\deg v_1 \geq k+l-1$. Противоречие с условием теоремы. Теорема доказана.

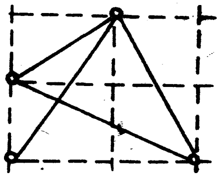

Рис.3

Отметим, что утверждение, обратное доказанному, неверно: на рис.3 изображен насыщенный граф, максимальная клика которого содержит вершины немаксимальной степени.

Насыщенными графами являются графы из следующих классов: K_n, $\bar{K}_n$, $K_n \nabla K_m$, $\bar{K}_n \nabla \bar{K}_m$, $K_{n_1, n_2, \ldots, n_m}$ (и, в частности, графы Муна-Мозера $K_{3,3,\ldots,3}$). Очевидны также следующие утверждения: если G_1 и G_2 насыщенные, то и $G_1 + G_2$ насыщенный; если граф насыщенный, то и любой его подграф насыщенный.

Последнее утверждение и теорема I позволяют находить максимальную клику насыщенного графа при помощи следующего алгоритма.

Алгоритм MACL (результатом алгоритма является множество K вершин насыщенного графа G, порождающее в G максимальную клику).

I. $K:=\emptyset$. Перейти к п.2.

2. Выбрать вершину v с максимальной степенью в G; $K:=K\cup\{v\}$. Перейти к п.3.

3. Выделить окружение $\Gamma_G(v)$ вершины v в G; $G:=\langle \Gamma_G(v) \rangle$. Перейти к п.4.

4. Если $G=\emptyset$, то перейти к п.5, иначе перейти к п.2.

5. Конец.

С помощью алгоритма MACL можно вычислять ранг подмножеств вершин в модульном произведении, если все вершины, не лежащие в одном ряду, соединить ребрами.

Нетрудно убедиться, что трудоемкость алгоритма MACL имеет порядок $O(p^3(G))$.

§6. Схема рекурсивного разбора для поиска клик в модульном произведении графов

Разбиением графа G на пояса [4] относительно вершины v назовем упорядоченную пару графов $\langle \Gamma_G(v) \rangle$ и $G-v$. Первый граф будем называть левым, а второй – правым поясом графа G.

Под рекурсивным разбором будем понимать последовательное разбиение исходного графа на пояса: исходный граф разбивается на два пояса, затем каждый пояс, в свою очередь, разбивается на два пояса и т.д. Общая схема рекурсивного разбора изображена на рис.4 и определяется следующими четырьмя правилами:

О – правило остановки заключается в проверке условия, при выполнении которого дальнейший разбор пояса не происходит;

В – правило выбора вершины, относительно которой граф разбивается на пояса;

У – правило упрощения определяет преобразование пояса к некоторому более "простому" графу;

Н – правило накопления – заключается в анализе и занесении результатов разбора в некоторый накопитель информации.

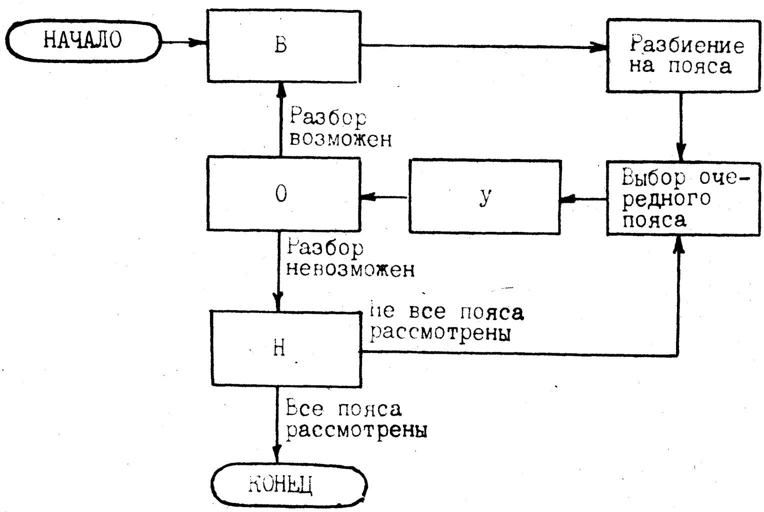

Рис. 4

П р а в и л о о с т а н о в к и. Для поиска в графе G_1 G_2 максимальных клик, соответствующих максимальным пересечениям, необходимо уметь прекращать дальнейший разбор тех поясов, в которых не содержится клик большего порядка, чем порядок максимальных клик, найденных на предыдущих шагах разбора. Пусть на некотором шаге разбора получен пояс $\langle W \rangle$. Если $k_В$ и $k_Н$ – соответственно нижняя и верхняя оценки порядка максимальной клики, которая может быть получена при исследовании $\langle W \rangle$, то основанием для прекращения разбора $\langle W \rangle$ может быть неравенство $k_В \le k_Н$. Ясно, что чем точнее эти оценки, тем быстрее будет работать алгоритм. Рассмотрим способы получения верхних и нижних оценок. Из построения схемы рекурсивного разбора следует, что W получается в результате применения к исходному графу G операций двух типов: выделения окружений вершин $v_1, v_2, \dots, v_k$ и удаления вершин $u_1, u_2, \dots, u_l$. Пусть $S = \{v_1, \dots, v_k\}$ (возможен случай, когда $S = \emptyset$, тогда $\langle W \rangle$ получен только удалением вершин $u_1, \dots, u_l$). Поскольку $v_i \in \bigcap_{t=1}^{i-1} \Gamma(v_t)$, $i = \overline{2,k}$, то $\langle S \rangle$ – полный подграф. Поэтому если k – максимальная клика в W, то $K + \langle S \rangle$ – максимальная клика в G и, значит, оценкой максимальной клики в G к моменту анализа пояса $\langle W \rangle$ будет $k + h(W)$, где $h(W)$ – оценка порядка максимальной клики в $\langle W \rangle$.

Рассмотрим способы получения оценок.

I. Н и ж н и е о ц е н к и.

Способ Н1 (тривиальный). Положим $k_Н$ равной порядку максимального полного подграфа, полученного на предыдущих шагах разбора; для исходного графа G $k_Н$ положим равной I.

Способ Н2. Этот способ основан на оценке, данной в [6]. Пусть $W = \{w_1, w_2, \dots, w_n\}$, а d_i – степень вершины w_i. Порядок максимальной клики в $\langle W \rangle$ больше либо равен $\sum_{i=1}^{n} \frac{1}{|W| - d_i}$.

2. В е р х н и е о ц е н к и.

Способ В1 (тривиальный). Положим $k_В = |W| + k$.

Способ В2. Положим $k_В = k + r$, где r – ранг множества W.

Способ В3. Положим $k_В = \min(|U_1|, |U_2|)$, где U_1 и U_2 – проекции множества W.

Предположим теперь, что $k_В > k_Н$ и граф $\langle W \rangle$ – насыщенный (т.е. обе проекции $\langle W \rangle$ либо вполне несвязны, либо полные). Тогда без дальнейшего разбора графа $\langle W \rangle$ можно определить порядок его максимальной клики или найти его максимальную клику за полиномиальное время (см.§5). Если же требуется перечислить все максимальные пересечения графов G_1 и G_2, то можно применить алгоритм, перечисляющий все максимальные клики в насыщенном графе (см.ниже) и работающий более эффективно по сравнению с общим случаем.

Алгоритм 0 (результатом алгоритма 0 является величина A, равная I, если дальнейший разбор пояса $\langle W \rangle$ возможен, и равная 0 – в противном случае).

I. Вычислить k_H и k_B. Если $k_H \geq k_B$, то перейти к п.3, иначе перейти к п.2.

2. Если обе проекции пояса $\langle W \rangle$ либо одновременно полные, либо одновременно вполне несвязные, то перейти к п.3, иначе перейти к п.4.

3. Положить $A = 0$. Перейти к п.5.

4. Положить $A = I$. Перейти к п.5.

5. Конец.

Если решается задача I, то алгоритм 0 необходимо модифицировать, добавив к п.I проверку условия $k_H \geq k$.

П р а в и л о в ы б о р а. Правило выбора может оказать заметное влияние на трудоемкость рекурсивного разбора [4].

Рассмотрим пояс $\langle W \rangle$, полученный на некотором шаге разбора. Пусть на предыдущих шагах разбора найдено максимальное пересечение, имеющее порядок t. Исследование пояса $\langle W \rangle$ должно привести к одному из двух результатов:

а) установлено, что не существует максимального пересечения, имеющего порядок больше чем t;

б) найдено максимальное пересечение, имеющее порядок больше t.

Правило выбора, следовательно, должно быть организовано таким образом, чтобы оно как можно быстрее приводило к одному из результатов "а" или "б".

При определении правила выбора мы будем руководствоваться соображениями, основанными на результатах §4: поскольку в насыщенных графах можно эффективно выделять максимальные клики, то необходимо, чтобы насыщенные графы возникали на как можно более ранних шагах и чтобы их порядки были как можно бóльшими.

Рассмотрим проекции $\langle U_1 \rangle$ и $\langle U_2 \rangle$ пояса $\langle W \rangle$. Пусть $N_1 \subseteq U_1$ и $N_2 \subseteq U_2$ – такие множества вершин, что оба подграфа $\langle N_1 \rangle$ и $\langle N_2 \rangle$

либо полные, либо вполне несвязные. Тогда подграф $\langle W \cap (N_1 x N_2) \rangle$ будет насыщенным. Поэтому если при дальнейшем разборе пояса $\langle W \rangle$ выбирать вершины из множества $Y_W = W \setminus (N_1 \times N_2)$, то это будет гарантировать нам, что через $|Y_W|$ шагов вправо по дереву разбора будет получен насыщенный подграф. Чем меньше $|Y_W|$, т.е. чем больше множества N_1 и N_2, тем будет меньше объем перебора при исследовании пояса $\langle W \rangle$. Таким образом, для того чтобы наилучшим образом сформировать множество Y_W, необходимо выделить максимальные независимые множества либо максимальные клики в графах $\langle U_1 \rangle$ и $\langle U_2 \rangle$. Задачи нахождения максимальных клик (или максимальных независимых множеств) известны как "труднорешаемые" [I]. Однако поскольку точность решения данных задач не окажет влияния на корректность всего алгоритма разбора в целом, то нас устроит какой-либо достаточно быстродействующий эвристический алгоритм поиска максимальных клик или максимальных независимых множеств. Примером может служить следующий алгоритм.

Алгоритм INSET (результатом алгоритма является независимое множество N вершин заданного графа G, максимальное или близкое к максимальному[ж)]).

I. $N := \emptyset$. Перейти к п.2.

2. Выбрать вершину $v \in V$ с минимальной степенью. $N := N \cup \{ v \}$. Перейти к п.3.

3. Удалить из V вершину v вместе с ее окружением и обозначить через V оставшееся множество вершин. Если $V = \emptyset$, то перейти к п.4, иначе перейти к п.2.

4. Конец.

Поскольку клика в G является вполне несвязным подграфом в $\bar{G}$, то алгоритм INSET может быть применен и для нахождения максимальных клик.

Итак, если исследование пояса $\langle W \rangle$ не может привести к бо́льшему пересечению, чем ранее найденное, то принцип, по которому формируется множество Y_W, приведет к данному результату быстрее по сравнению с тем случаем, когда вершины выбираются произвольно, и это подтверждается экспериментами (см.ниже §8).

ж) Если G - ациклический граф, то алгоритм INSET выделяет максимальное независимое множество [7]; в общем случае для $|N|$ известна нижняя оценка [6].

Предположим теперь, что при исследовании пояса $\langle W \rangle$ можно выделить пересечение $\begin{pmatrix} u_1 u_2 \dots u_n \\ v_1 v_2 \dots v_n \end{pmatrix}$ бо́льшего порядка, чем найденное на предыдущих шагах. Пусть $\langle\{w_1, w_2, \dots, w_n\}\rangle$ – клика, соответствующая данному пересечению. Возникает вопрос: каков должен быть принцип выбора вершин w_i, чтобы эту клику найти за наименьшее число шагов?

Сформулируем три принципа выбора, которые обеспечивают большую вероятность выделения максимального пересечения, по сравнению со случайным выбором.

Принцип В1. Выбрать вершину, имеющую максимальную степень.

В графах из некоторых классов (например, в насыщенных, в дополнениях ациклических графов [7]) данный принцип действительно позволяет выделить максимальную клику за наименьшее число шагов.

Принцип В2. Выбрать вершину, сумма степеней проекций которой максимальна.

Принцип В3. Выбрать вершину, сумма степеней проекций которой минимальна.

Принципы В2 и В3 в отличие от В1 предполагают использование информации об исходных графах и отражают интуитивное представление о том, что в наибольшем пересечении соответствующие друг другу вершины не должны сильно отличаться по степеням.

Можно предложить и другие принципы выбора, например, основанные на сравнении метрических характеристик вершин в проекциях пояса, необходимо только, чтобы алгоритмическая реализация принципа не требовала слишком много вычислений, так как правило выбора выполняется рекурсивно.

В §8 дается описание эксперимента по исследованию эффективности принципов В1-В3.

Алгоритм В (результатом алгоритма является вершина v пояса $\langle W \rangle$, относительно которой граф $\langle W \rangle$ будет разбит на пояса).

1. Если $\langle W \rangle$ – правый пояс, то перейти к п.4 (в этом случае множество Y_W уже сформировано на предыдущих шагах разбора), иначе перейти к п.2.

2. Определить проекции пояса $\langle W \rangle$, в которых выделить множества N_1 и N_2 с помощью алгоритма INSET. Перейти к п.3.

3. Положить $Y_W = W \setminus (N_1 \times N_2)$. Перейти к п.4.

4. Выбрать в Y_W вершину v в соответствии с некоторым принципом (например, в соответствии с B1,B2 или B3). Перейти к п.5.

5. Конец.

П р а в и л о у п р о щ е н и я. Правило упрощения определяет преобразование пояса к некоторому более "простому" графу с сохранением искомого свойства. Мы рассмотрим правило упрощения Y_1, которое заключается в удалении вершин пояса, не попадающих в клику заданного порядка. Пусть к моменту исследования пояса $\langle W \rangle$ установлено, что порядок максимальной клики графа $G_1 \nabla G_2$ не менее k_M, а сам пояс $\langle W \rangle$ получен выделением окружений l вершин. Тогда при исследовании $\langle W \rangle$ клика большего порядка в $G_1 \nabla G_2$ будет получена в том случае, когда порядок максимальной клики в $\langle W \rangle$ больше $k_M - 1$. Любая вершина графа $\langle W \rangle$, степень которой меньше k_M-1, заведомо не войдет в клику порядка $t > k_M-1$ и может быть удалена. В результате удаления всех таких вершин в полученном графе вновь могут появиться вершины со степенями, меньшими k_M-1. Таким образом, процедура Y_1 будет итеративной и должна приводить или к графу, в котором все степени вершин больше k_M-1, или к пустому графу. На рис.5 для $k_M-1 = 4$ иллюстрируется процедура Y_1.

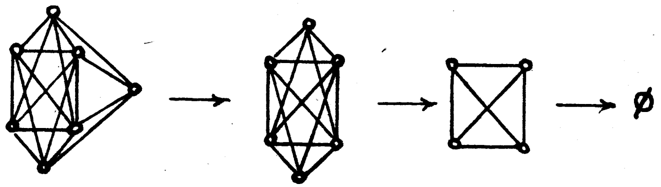

Рис. 5

Тождественное преобразование графов условимся также считать упрощением и обозначим его символом Y_0.

Целесообразность использования того или иного правила упрощения не очевидна, так как они применяются на каждом шаге рекурсии и, уменьшая число шагов рекурсии, могут тем не менее приводить к увеличению общего времени работы алгоритма (см.ниже §8).

П р а в и л о н а к о п л е н и я. Накопление результатов разбора может происходить в том случае, когда пояс $\langle W \rangle$ удовлетворяет условиям остановки. Пусть при получении пояса $\langle W \rangle$ были выполнены операции выделения окружений вершин $v_1, v_2, \dots, v_k$, а до момента начала анализа пояса $\langle W \rangle$ порядок найденного максимального пересечения графов G_1 и G_2 равен t (перед началом разбора $t = 0$), и в накопителе информации хранится клика порядка t.

Алгоритм H_1.

I. Если разбор пояса $\langle W \rangle$ прекращен вследствие бесперспективности (т.е. $k_H \geq k_B$, см. п.I алгоритма 0), то перейти к п.4, иначе перейти к п.2.

2. Выделить максимальную клику $\langle\{w_1, w_2, \dots, w_n\}\rangle$ в $\langle W \rangle$ при помощи алгоритма MACL (поскольку $\langle W \rangle$ – насыщенный граф). Перейти к п.3.

3. Если $n+k > t$, то занести в накопитель информации новую клику $\langle\{v_1, v_2, \dots, v_k, w_1, w_2, \dots, w_n\}\rangle$ взамен старой и перейти к п.4, иначе – сохранить старую клику и перейти к п.4.

4. Конец.

Правило накопления H_1 предназначено для использования в алгоритме разбора при решении задачи 3. Для решения задач I и 2 правило H_1 модифицируется таким образом, что в накопитель информации заносятся только порядки клик.

Рассмотрим теперь задачу 4 перечисления всех максимальных пересечений двух графов. В этом случае правило накопления должно включать в себя процедуру, эффективно перечисляющую максимальные клики в насыщенном графе. Идея построения такой процедуры состоит в следующем. Пусть H – насыщенный граф. Рассмотрим множество графов $H_{1,1}, H_{1,2}, H_{1,3}, H_{2,3}, \dots, H_{1,n}, H_{2,n}$, где $H_{1,1} = H$, а $H_{1,i}$ и $H_{2,i}$ – левый и правый пояса графа $H_{1,i-1}$ $(i=\overline{2,n})$ в разбиении его относительно вершин с максимальными степенями, граф $H_{1,n}$ – пустой. Тогда порядок максимальной клики графа H равен $n-1$, а сама клика состоит из всех вершин, относительно которых строились разбиения на пояса. Поиск максимальных клик графа H продолжим, применяя разбиение на пояса, аналогичное рассмотренному, поочередно к поясам $H_{2,n}, H_{2,n-1}, \dots, H_{2,2}$. Поскольку теперь порядок максимальной клики графа H нам известен, а графы $H_{2,i}$ насыщенные, то за $p(H_{2,i})$ шагов рекурсивного разбора мы можем определить, приведет ли исследование пояса $H_{2,i}$ к новым максимальным кликам графа H. Если ответ отрицательный, то исследование пояса $\langle W \rangle$ следует прекратить. Ниже, в §7, дается анализ трудоемкости данной процедуры перечисления максимальных клик.

§7. О трудоемкости алгоритмов

Под трудоемкостью алгоритмов здесь, так же как и в [4], будем понимать число элементарных операций (арифметических и логических операций над парами величин, операций пересылки единиц данных), необходимых для выполнения алгоритмов. Для анализа трудоемкости алгоритмов в качестве модели вычислительного процесса будем использовать дерево разбора [4].

Дерево разбора является корневым деревом, в котором вершины соответствуют поясам, причем корневая вершина соответствует исходному графу. Каждая вершина w дерева разбора представляет некоторый пояс $\langle W(w) \rangle$, полученный в процессе разбора. Если $\langle W(w) \rangle$ удовлетворяет условиям остановки, то w – висячая вершина, в противном случае она имеет два последователя, из которых левый последователь представляет левый пояс графа $\langle W(w) \rangle$, а правый последователь – правый пояс. Введем следующие обозначения:

p – число вершин графа G;

T – дерево разбора;

V_T – множество вершин дерева T; $n_T \equiv |V_T|$;

V_T' – множество висячих вершин дерева T;

$\langle W(w) \rangle$ – пояс, соответствующий вершине w дерева T;

$\sigma_O(w), \sigma_B(w), \sigma_Y(w)$, $\sigma_H(w)$ – количества операций при выполнении алгоритмов O, B, Y, H на графе $\langle W(w) \rangle$;

$\sigma_p(w)$ – количество операций, необходимое для разбиения на пояса графа $\langle W(w) \rangle$.

Число операций ξ при выполнении алгоритма разбора можно представить в виде:

$$\xi = \sum_{w \in V_T \setminus V_T'} (\sigma_O(w)+\sigma_B(w)+\sigma_Y(w)+\sigma_p(w)) + \sum_{w \in V_T'} (\sigma_O(w)+\sigma_H(w)). \qquad (1)$$

Нетрудно показать (например, способами, использованными в [4] для оценок числа операций), что имеют место следующие оценки:

$$\sigma_O(w) \le O(p^2), \quad \sigma_B(w) \le O(p^2), \quad \sigma_p(w) \le O(p^2),$$

$$\alpha_Y(w)^{*)} \le \begin{cases} O(1) & \text{при} \quad Y = Y_0, \\ O(p^3) & \text{при} \quad Y = Y_1. \end{cases}$$

Оценим величину $\sigma_H(w)$. При решении задач I–3 алгоритм H заключается в нахождении максимальной клики в насыщенном графе путем последовательного выделения окружений вершин с максимальной степенью (см. §6). Если порядок максимальной клики равен n, то необходимо n раз выполнить следующие операции, имеющие верхние оценки трудоемкости $O(p^2)$: в графе выбрать вершину с максимальной степенью и выделить ее окружение. Поскольку p равно произведению порядков исходных графов, то $n \le O(\sqrt{p})$ и, значит, $\sigma_H(w) \le O(p^{2,5})$. При решении задачи 4 алгоритм H должен выделять все максимальные клики в насыщенном графе. Процесс выделения клик в этом случае можно представить в виде схемы рекурсивного разбора. Пусть на некотором шаге разбора выделена максимальная клика. Оценим число шагов разбора, которые необходимо выполнить прежде, чем будет выделена следующая максимальная клика. Рассмотрим фрагмент дерева разбора (рис.6). Исследование начинается из вершины w и приводит через n шагов к максимальной клике M, которой соответствует вершина w_0. Далее необходимо выполнить исследования поочередно из вершин $w_1, w_2, \ldots, w_n$. Худшим будет случай, когда следующая максимальная клика будет найдена из вершины w_n. Поскольку графы $\langle W(w_i) \rangle$ насыщенные, то исследование из вершин w_i потребует не более $O(n)$ шагов $(i = \overline{1,n})$ и, значит, общее число шагов, необходимое для получения следующей после M клики, ограничено сверху величиной $O(n^2)$. Учитывая, что число операций, соответствующее каждому шагу, имеет порядок $O(p^2)$, а $n \le O(\sqrt{p})$, получаем $\sigma_H(w) \le O(p^3 \cdot \mu_W)$, где μ_W – число максимальных клик в $\langle W(w) \rangle$.

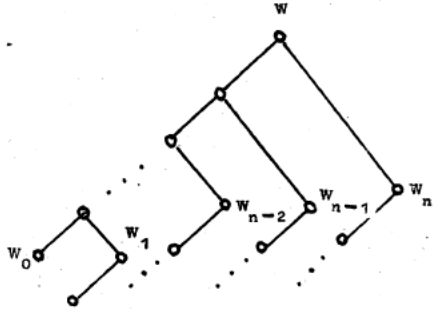

Рис. 6

---

*) Запись $f(n) \le O(g(n))$ означает, что существует такая константа c, что для всех $n = 1,2, \ldots$ выполняется неравенство $f(n) \le c \cdot g(n)$.

Таким образом, из (I) имеем, что при решении задач I-3:

$$\xi \le \begin{cases} O(p^{2,5}\cdot \eta_T), & \text{если} \quad Y = Y_0, \\ O(p^3\cdot \eta_T), & \text{если} \quad Y = Y_1, \end{cases}$$

и при решении задачи 4:

$$\xi \le \begin{cases} O(p^2\cdot \eta_T) + \sum\limits_{w\in V_T^!} O(p^3\cdot \mu_w), & \text{если} \quad Y = Y_0, \\ O(p^3\cdot \eta_T) + \sum\limits_{w\in V_T^!} O(p^3\cdot \mu_w), & \text{если} \quad Y = Y_1. \end{cases}$$

Вершину $w \in V_T$ назовем правой, если ей соответствует правый пояс $\langle W(w) \rangle$, и левой – в противном случае. Цепь $P_w = w, w_1, \ldots, w_n$, соединяющую вершину w с висячей вершиной w_n, назовем правой, если каждая вершина w_i – правая $(i = \overline{1,n})$. Множество всех левых вершин дерева T обозначим через $V_T^{Л}$. Если условиться считать правой цепью P_w для висячей вершины w саму эту вершину, то можно записать: $V_T = \bigcup\limits_{w\in V_T^{Л}} P_w$, где P_w – правые цепи (корневая вершина считается левой).

Алгоритм В (правило выбора) строит для каждого левого пояса $\langle W(w) \rangle$ множество выбора $Y_{W(w)}$ таким образом, что через $|Y_{W(w)}|$ шагов вправо по дереву разбора выполняется условие остановки, поэтому длина правой цепи P_w равна $|Y_{W(w)}|$ и, значит,

$$\eta_T = \sum_{w\in V_T^{Л}} |Y_{W(w)}|.$$

Множество $Y_{W(w)}$ определяется как $W(w)\setminus(N_1 \times N_2)$, где $\langle N_1 \rangle$ и $\langle N_2 \rangle$ – либо вполне несвязные, либо полные подграфы в проекциях $\langle W \rangle$. Отсюда ясно, что на графах G_1 и G_2 предложенные алгоритмы будут работать тем эффективнее, чем сильнее будет выражена их следующая особенность: каждый из подграфов в G_1 и в G_2 имеет большую плотность (т.е. порядок максимальной клики) или большую неплотность. Свойство графа называется наследственным, если оно выполняется для каждого его подграфа. Свойством иметь наследственно большую неплотность ($\alpha(G') \ge \frac{p(G')}{2}$ для каждого $G' \subseteq G$) обладают графы G, используемые в качестве моделей молекулярных структур (в химии), процессов эволюции (в биологии), электронных схем (в электронике). Отсюда видна прикладная направленность предложенных алгоритмов.

§8. Результаты экспериментального исследования алгоритмов

Для исследования эффективности предложенных алгоритмов был разработан комплекс экспериментальных программ. В цели эксперимента входило:

- сравнение эффективности различных вариантов правил выбора и остановки;
- проверка эффективности правила упрощения Y_1;
- изучение возможностей применения алгоритмов для решения реальных прикладных задач.

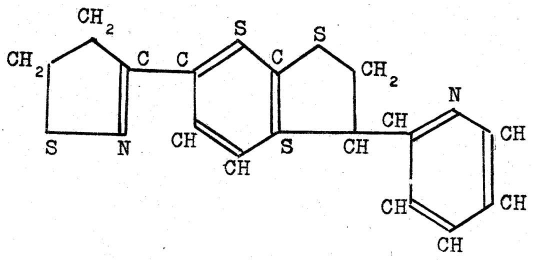

Рис. 7

Эксперимент проводился на графах из следующих классов:

I) молекулярных графов органических соединений (рис.7);

2) деревьев с непомеченными вершинами;

3) деревьев с помеченными висячими вершинами (эволюционные деревья в генетике [8], рис.8);

4) двудольных графов.

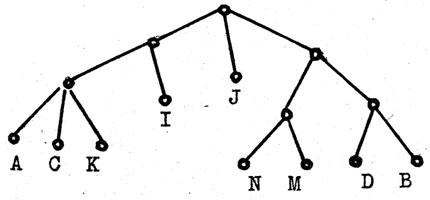

Рис. 8

Некоторые усредненные характеристики указанных графов представлены в таблице.

Т а б л и ц а

Характеристики графов, использованных для эксперимента

Графы	Характеристики				
	Порядок	Число ребер	Максимальн. степени вершин	Неплотность	Число типов меток
Молекулярные графы	20	25	4	I5	5
Непомеченные деревья	I7	I6	3	I0	-
Эволюционные деревья	20	I9	3	I4	I0
Двудольные графы	30	40	5	I5	I6

Для сравнения алгоритмов была использована инвариантная величина η_T - число вершин дерева разбора.

Преимущества использования правил выбора и остановки, учитывающих особенности графов, показаны при решении задачи 2 следующими тремя алгоритмами. Первый алгоритм А1 есть реализация схемы рекурсивного разбора с правилом остановки, в котором нижняя оценка порядка наибольшего пересечения вычисляется способом L1, а верхняя оценка - способом H1 (§6). Правило выбора состоит в выделении вершины с наименьшим номером. По сути, алгоритм А1 есть простейшая реализация метода, предложенного В.Г.Визингом [2] и Леви [3]. Второй алгоритм А2 отличается от А1 правилом выбора, основанном на построении множеств Y_W (§6). Третий алгоритм А3 отличается от А2 правилом остановки, в котором нижняя оценка порядка наибольшего пересечения вычисляется способом L1, а верхняя - способом H3. Величина η_T и время t (на ЕС 1050), характеризующие выполнение алгоритмов на двух семивершинных деревьях, имеют следующие значения:

η_T = 1035, t = 164,32 сек (для А1); η_T = 193, t = 52,16 сек (для А2); η_T = 3, t = 1,02 сек (для А3).

В дальнейшем исследовались различные модификации алгоритма А3 при решении задачи I. Модификации заключались в использова -

нии различных способов получения верхних и нижних оценок в правиле остановки, различных принципов выбора вершин, а также во введении правила упрощения Y_1.

Исследования показали, что

I) наиболее эффективным способом получения нижних оценок порядка наибольшего пересечения, является Н1, а верхних – Н3;

2) принципом выбора вершин, обеспечивающих наименьшее число шагов для достижения наибольшего пересечения, является В2;

3) введение правила Y_1 приводит к уменьшению величины η_T (в среднем на 20%), но и к увеличению времени счета (поскольку трудоемкость операции упрощения есть $O(p^3)$);

4) без использования правила Y_1 время счета[x)], приходящееся на одну вершину дерева разбора, увеличивается с ростом порядка модульного произведения от 0,I сек (для $p = 50$) до I мин (для $p = 1024$);

5) время счета при решении задач нахождения максимальных пересечений для эволюционных деревьев составляет в среднем около I мин, для молекулярных графов – от I до I0 мин.

Заключение

В работе рассмотрены методы построения алгоритмов поиска наибольших пересечений двух графов, включающие сведение исходной задачи к поиску клик в модульном произведении при помощи рекурсивного разбора с использованием такой особенности исходных графов, как наличие в каждом их подграфе сравнительно больших множеств попарно несмежных (либо попарно смежных) вершин.

Разработанные экспериментальные программы поиска наибольших пересечений графов исследованы на графах, используемых в качестве моделей структур в химии, биологии и электронике. Результаты исследования показывают возможность использования предложенных методов для построения эффективных прикладных алгоритмов.

Литература

I. ГЭРИ М., ДЖОНСОН Д. Вычислительные машины и труднорешаемые задачи. –М.: Мир, I982. – 367 с.

2. ВИЗИНГ В.Г. Сведение проблемы изоморфизма и изоморфного вложения к задаче нахождения неплотности.–В кн.: Труды Ш Всесоюз. конф. по проблемам теоретической кибернетики. Новосибирск, 1974, с.I24.

[x)] На ЭВМ ЕС I050.

3. LEVI G. A note of the derivation of maximal common subgraphs of two directed or undirected graphs.-Calcolo,1972,N9,p.341.

4. БЕССОНОВ Ю.Е., СКОРОБОГАТОВ В.А. Об одном семействе схем рекурсивного разбора графов. -В кн.: Машинные методы обнаружения закономерностей, анализа структур и проектирования (Вычислительные системы, вып. 92). Новосибирск, 1982, с. 3-49.

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

6. КОЧЕТОВ А.А. О некоторых стратегиях нахождения независимого множества в графах. -В кн.: Комбинаторно-алгебраические методы в прикладной математике. Горький, 1980, с.78-87.

7. ВАХОВСКИЙ Е.В. Об одном способе демонтажа графа. -Сиб.мат. журн., 1968, т.IX, № 2, с.255-263.

8. HENDY M.D. Establishing the minimality of phylogenetic trees from protein sequences.- Lect.Notes Math.,1979,N 829,p.165-171.

Поступила в ред.-изд.отд.
28 июня 1985 года