ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ ЗАДАЧИ РАСПОЗНАВАНИЯ ИЗОМОРФНЫХ ГРАФОВ

Дудов М.Х.

Карачаево-Черкесская государственная технологическая академия, г.Черкесск

dudovm@mail.ru

Предлагается метод распознавания изоморфных графов с полиномиальной вычислительной сложностью.

Пусть $G_1=(V_1,E_1)$ и $G_2=(V_2,E_2)$ - графы. Требуется определить, изоморфны ли G_1 и G_2, т.е. существует ли функция $f:V_1\to V_2$ такая, что $(u,v)\in E_1$ тогда и только тогда, когда $(f(u),f(v))\in E_2$ [1]. Вопрос о вычислительной сложности рассматриваемой задачи остается открытым [1,2], хотя для некоторых частных случаев построены эффективные алгоритмы с полиномиальной вычислительной сложностью [3-6]. Ниже впервые предлагается и обосновывается метод распознавания изоморфных графов, позволяющий построить алгоритмы с вычислительной сложностью не более $O(n^5)$.

Пусть $P=\|p_{ij}\|$, причем $p_{ij}=1$ при $j=\pi(i)$ и $p_{ij}=0$ при $j\neq\pi(i)$ для всех $i,j=\overline{1,n}$, где π - некоторая подстановка, т.е. матрица P может быть получена перестановкой столбцов единичной матрицы в соответствии с подстановкой π. Матрица P называется матрицей перестановок и обладает следующим свойством – если $B=P\ AP^T$, где $\tilde{N}$ - некоторая $n\times n$-матрица, то $b_{ij}=\tilde{n}_{\pi(i)\pi(j)}$, т.е. матрица B получается из матрицы $\tilde{N}$ перестановкой строк и столбцов в соответствии с подстановкой π.

Пусть $A=\|a_{ij}\|,i,j=\overline{1,n}$- матрица смежности графа G. Построим матрицу $M=\|m_{ij}\|,i,j=\overline{1,n}$ такую, что

$$\begin{cases} m_{ij}=-a_{ij}, i\neq j;\\ m_{ii}=deg\,v_i+1, \end{cases} \tag{1}$$

где $deg\,v_i$ - степень *i*-ой вершины графа G.

Матрица M называется далее схемной матрицей графа G [7]. Известно, что схемные матрицы симметрические и, в отличие от матриц смежности, всегда неособенные [8].

Пусть M_1 и M_2 - схемные матрицы графов G_1 и G_2. Обозначим через $m_{ij}^{(1)}$ и $m_{ij}^{(2)}$ элементы этих матриц, а через $m_k^{(1)}$ и $m_l^{(2)}$ - k-ый и l-ый столбцы матриц M_1 и M_2 соответственно.

Построим матрицу C^{kl} такую, что $c_{ij}^{kl}=1$, если $m_{ik}^{(1)}=m_{jl}^{(2)}$, все же остальные элементы матрицы C^{kl} равны нулю. Если с помощью перестановки элементов $m_l^{(2)}$ можно построить $m_k^{(1)}$, то на C^{kl} существует по меньшей мере один набор из n независимых единиц, т.е. по одной в каждом столбце и строке. Каждый такой набор однозначно задает порядок перенумерации, позволяющий из l-го столбца матрицы M_2 получить k-ый столбец матрицы M_1. Соответственно отсутствие такого набора означает, что перестановки, переводящей $m_l^{(2)}$ в $m_k^{(1)}$, не существует. Поэтому матрицу C^{kl} будем далее называть матрицей допустимости перестановки $m_l^{(2)}$ и $m_k^{(1)}$.

Рассмотрим теперь матрицу $C_{\langle 2\rangle} = C^{kl} + C^{st}$, $k \neq s, l \neq t$. Если на ней существует набор из n независимых двоек, то очевидно, что и на C^{kl}, и на C^{st} существует такой же набор из n независимых единиц. Следовательно, порядок перенумерации, позволяющий из l-го столбца матрицы M_2 получить k-ый столбец матрицы M_1 совпадает с порядком перенумерации, позволяющим из t-го столбца матрицы M_2 получить s-ый столбец матрицы M_1. Соответственно отсутствие такого набора означает, что перестановки, переводящей $m_l^{(2)}$ в $m_k^{(1)}$ и $m_t^{(2)}$ в $m_s^{(1)}$ одновременно, не существует.

Пусть теперь $C_{\langle n\rangle} = \sum_{i=1}^{n} C^{i\pi(i)}$. Если на $C_{\langle n\rangle}$ существует набор из n независимых «n-ок», то существует перестановка столбцов матрицы M_2, переводящая одновременно все $m_i^{(2)}$ в $m_{\pi(i)}^{(1)}$. Или, иначе говоря, существует матрица перестановок P такая, что $M_1 = PM_2P^T$.

Этот метод распознавания изоморных графов может быть реализован следующим алгоритмом. Предположим, что графы G_1 и G_2 изоморфны, и, следовательно, существует матрица перестановок P такая, что $M_1 = PM_2P^T$. Будем искать P следующим образом.

Предположим, что $p(k,l)=1$, т.е. l-ый столбец матрицы M_2 станет k-ым столбцом матрицы $M_1 = PM_2P^T$. Построим матрицу $C_{\langle 1\rangle}$ следующим образом: сформируем матрицу C^{kl} и вычеркнем k-ую строку и l-ый столбец, оставшаяся невычеркнутой часть матрицы C^{kl} есть $C_{\langle 1\rangle}$. Найдем один из ненулевых элементов $C_{\langle 1\rangle}$, пусть $c_{s,t}$. Построим матрицу $C^{kl} + C^{st}$ и оставшуюся после вычеркивания k-ой и s-ой строк и l-го и t-го столбцов часть этой матрицы назовем $C_{\langle 2\rangle}$. Если в какой-либо строке (столбце) матрицы $C_{\langle 2\rangle}$ нет двойки, то, по построению, $p(k,l)=1$ и $p(s,t)=1$ не могут быть одновременно равны «1», т.е. перестановки, переводящей $m_l^{(2)}$ в $m_k^{(1)}$ и $m_t^{(2)}$ в $m_s^{(1)}$ одновременно, не существует. В таком случае возвращаемся к матрице C^{kl} и выбираем следующий ненулевой элемент в l-ом столбце.

Если же во всех строках (столбцах) $C_{\langle 2\rangle}$ есть хотя бы по одному элементу «2», то выбираем один из них, пусть $c_{r,f}$, причем $r \notin \{k,s\}$, $f \notin \{l,t\}$ и находим $C_{\langle 3\rangle}$. Если в какой-либо строке (столбце) этой матрицы нет элемента «3», то утверждения «$p(r,f)=1$» и «$p(k,l)=1$, $p(t,s)=1$» несовместимы, тогда возвращаемся к матрице $C_{\langle 2\rangle}$ и выбираем другой элемент «2» в том же столбце f и продолжаем работу алгоритма. Если в столбце больше не осталось невыбранных двоек, то возвращаемся к матрице $C_{\langle 1\rangle}$.

Пусть уже построена матрица $C_{\langle m\rangle}$ и во всех строках (столбцах) её есть хотя бы по одному элементу «m». Если $m=n$, то верхние индексы матриц $C^{i\pi(i)}$, сумма которых определялась при построении $C_{\langle n\rangle}$, составляют искомый набор из n независимых «n-ок», т.е. соответствующую перестановку столбцов и строк матрицы M_2, переводящую одновременно все $m_i^{(2)}$ в $m_{\pi(i)}^{(1)}$.

Если в результате произошел возврат к матрице $C_{\langle 1\rangle}$ и в ней нет еще невыбранных элементов «1», то предположение, что $p(k,l)=1$, ложно, т.е. l-ый столбец матрицы M_2 не может стать k-ым столбцом матрицы $M_1 = PM_2P^T$. Тогда предположим, что $p(k,l+1)=1$, строим матрицу $C^{k,l+1}$, вычеркиваем k-ую строку и $(l+1)$-ый столбец, т.е. получаем новую матрицу $C_{\langle 1\rangle}$ и продолжаем, если возможно, работу алгоритма. Если же выяснилось, что все $p(k,j)=0,\ j=1,..,n$, т.е. ни один из столбцов матрицы M_2 не может стать k-ым столбцом матрицы $M_1=PM_2P^T$, то искомой перестановки (или перенумерации) столбцов и строк матрицы M_2, переводящей одновременно все $m_i^{(2)}$ в $m_{\pi(i)}^{(1)}$, не существует. Иначе говоря, матрицы перестановок P, такой что $M_1=PM_2P^T$, не существует, т.к. графы G_1 и G_2 неизоморфны.

При построении матриц $C_{\langle i\rangle}$ их размерность последовательно уменьшается, например, размерность $C_{\langle m\rangle}$ есть $(n-m)\times(n-m)$, поэтому количество возвратов алгоритма к матрице $C_{\langle m\rangle}$ не более $(n-m)$. После каждого возврата к матрице $C_{\langle m\rangle}$ требуется выполнить $O(n^2)$ операций для построения новой матрицы $C_{\langle m+1\rangle}$. Таким образом, каждый возврат к матрице $C_{\langle 1\rangle}$ может произойти после не более чем $O(n^4)$ элементарных операций. Для проверки всех $p(k,j),\ j=\overline{1,n}$ потребуется не более n раз построить новые матрицы $C_{\langle 1\rangle}$, поэтому вычислительная сложность предлагаемого метода распознавания изоморфных графов в худшем случае не превышает $O(n^5)$. Кроме того, если при получении первого же искомого решения не прерывать работу алгоритма, то будут найдены все возможные искомые перестановки.

Литература

1. *Гери М., Джонсон Д.* // Вычислительные машины и труднорешаемые задачи. – М.: Мир,1982.
2. *Карп Р.М.* // Сводимость комбинаторных задач. – Киб.сб. – М.: Мир, 1975.
3. *D. G. Corneil and C. C. Gotlieb,* // An Efficient Algorithm for Graph Isomorphism *J. ACM,* Vol. 17, 51—64 (1970).
4. *L. Weinberg,* // A Simple and Efficient Algorithm for Determining Isomorphism of Planar Triply Connected Graphs, *IEEE Trans. Circuit Theory,* Vol. CT-13, 142—148 (1966).
5. *J. E. Hopcroit and R. E. Tarjan,* A $V\log V$ // Algorithm for Isomorphism of Triconnected Planar Graphs, *J. Comput. Syst. Sci.,* Vol. 7, 323—331 (1973).
6. *J. E. Hopcroft and J. K. Wong,* // Linear Time Algorithm for Isomorphism on Planar Graphs, Proc. 6th Annual ACM Symp. on Theory of Computing. 1974, pp. 172—184.
7. *Веников В.А., Зуев Э.Н., Литкенс И.В., Маркович И.М., Мельников Н.А., Солдаткина Л.А., Строев В.А.* // Электрические системы. Математические задачи электроэнергетики. – М.: ВШ, 1981.
8. *Свами М., Тхуласираман К.* // Графы, сети и алгоритмы. – М.: Мир,1984.