ПРЯМОЙ АЛГОРИТМ ПРОВЕРКИ ИЗОМОРФИЗМА ГРАФОВ

А.В. Пролубников[1]

[1]*Омский государственный университет, Омск, Россия*

Аннотация

Предлагается алгоритм решения задачи проверки изоморфизма графов. Алгоритм является прямым, в том смысле, что он не является модификацией схемы рекурсии с возвратом, на основе которой построены наиболее эффективные алгоритмы решения задачи. Решение задачи находится за число итераций алгоритма, не превышающее число вершин в графах. Показывается, что алгоритм является полиномиальным по числу используемых элементарных машинных операций и по используемой памяти. Исследуется класс графов, на котором алгоритм дает решение задачи.

Введение

Задача изоморфизма графов (ИГ) принадлежит к задачам, относительно которых нет ясности: являются ли они полиномиально разрешимыми или нет [1], тогда, как известно, эта задача полиномиально разрешима для некоторых классов графов. В частности для планарных графов, графов с ограниченной степенью вершин, графов с ограниченной кратностью собственных значений из спектра матрицы смежности [2], [3], [4] и некоторых других классов графов построены эффективные алгоритмы решения этой задачи.

Поскольку эти алгоритмы используют специфические структурные характеристики графов, область их эффективного применения ограничена. Возникает необходимость в алгоритме, который находил бы решение задачи ИГ для как можно более широкого класса графов, оставаясь при этом полиномиальным как по времени, так и по памяти.

Алгоритм, представленный в [5], работает со спектральными характеристиками графов и является полиномиальным алгоритмом решения задачи ИГ с наиболее широким классом применения. Но этот класс не покрывает определяемых спектром графов, которые попадают в область эффективного применения предлагаемого алгоритма. Граф называется определяемым спектром (ОС-графом), если ему изоморфен любой граф с таким же спектром матрицы, представляющей граф. Спектр матрицы графа так же называют спектром графа.

Спектр графа не является его полным инвариантом. Так, почти все деревья и сильно-регулярные графы не являются определяемыми спектром. Вместе с тем, спектр содержит в себе разнообразную информацию о структуре графа [6] – такую как число ребер в графе, степенная последовательность графа, род графа и др., что может быть использовано при построении эвристик для решения задачи ИГ. Алгоритм спектрального расщепления [7], работает не напрямую со спектром графа, а со следующим, связанным со спектром, инвариантом:

$$S(G)=\left(\prod\lambda_k,\prod\tilde{\lambda}_k^1,\prod\tilde{\lambda}_k^2,\ldots,\prod\tilde{\lambda}_k^n\right),$$

где λ_k – собственные значения модифицированной матрицы смежности графа, $\tilde{\lambda}_k^i$ – собственные значения матриц подграфов, полученных из исходного удалением i-ой вершины; произведения берутся по всем значениям из спектров. Будем называть граф определяемым спектрами графа и подграфов (ОС2-графом), если ему изоморфен любой граф с таким же значением S.

Как показывает эксперимент, многие графы, которые не являются ОС-графами, например, деревья являются ОС2-графами. Тогда как всякий ОС-граф является ОС2-графом.

В задаче проверки изоморфизма графов даны два графа G_A и G_B с множествами вершин $V(G_A)$, $V(G_B)$ и ребер $E(G_A)$, $E(G_B)$. Мощности множеств вершин и ребер графов равны. Требуется определить, существует ли такое биективное отображение (изоморфизм) $\varphi: V_A \to V_B$, что

$$(i,j)\in E(G_A) \Leftrightarrow (\varphi(i),\varphi(j))\in E(G_B),$$

и представить его, если графы изоморфны.

При работе со спектром графа могут рассматриваться различные варианты матриц, представляющих граф. Все они являются некоторыми модификациями матрицы смежности графа A_0. $A_0=(a_{ij})$, где

$$a_{ij}=\begin{cases}1, (i,j)\in E_A,\\ 0, \text{ иначе,}\end{cases}$$

$B_0=(b_{ij})$ – матрица смежности графа G_B.

Чем больше кратность собственных значений в спектрах графов, тем больше в графах симметрий, и тем более сложным является нахождение изоморфизма двух изоморфных графов с кратными собственными значениями. Предлагаемый алгоритм решения задачи проверки изоморфизма графов работает с видоизмененными матрицами смежности. В ходе его работы последовательными согласованными возмущениями диагональных элементов матрицы производится расщепление собственных значений графа, что позволяет решать задачу проверки ИГ для широкого класса графов за полиномиальное время в смысле количества используемых машинных операций.

Алгоритм является прямым, то есть решение производится без использования схемы рекурсии с возвратом, на которой основаны наиболее известные алгоритмы решения задачи ИГ без каких-либо ограничений на структуру проверяемых на изоморфизм графов – алгоритмы Ullman, NAUTY и VF, экспоненциальные для общего случая задачи.

Алгоритм

Пусть A_0 – матрица смежности графа $G_A = \langle V_A, E_A \rangle$, D_{A_0} – диагональная матрица следующего вида:

$$\begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{bmatrix},$$

где $d_i = \sum_{j=1}^{n} a_{ij}^0 + d = d_i + d$, d – максимальная степень вершин в графе G_A, d_i – степень вершины $i \in V_A$. Матрица D_{B_0} строится аналогично в соответствии с матрицей B_0. Алгоритм работает с матрицами следующего вида:

$$A = A_0 + D_{A_0}, B = B_0 + D_{B_0}. \qquad (1)$$

Матрицы вида (1) – положительно определенные матрицы с диагональным преобладанием. Матрицами графов далее будем называть матрицы вида (1). В ходе работы алгоритма производится возмущение матриц графов при помощи следующих матриц:

$$E^k = \begin{array}{cl} \begin{bmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix} & \begin{matrix} - \; 1 \\ \\ - \; k \\ \\ - \; n \end{matrix} \end{array}$$

На i-ой итерации будем производить следующие возмущения матриц графов:

$$A^i := A^{i-1} + \varepsilon E^i, B^i := B^{i-1} + \varepsilon E^j,$$

где j выбирается так, что $S(G_{A^i}) = S(G_{B^i})$. Если G_A и G_B – ОС2-графы, и G_A изоморфен G_B, то проведение таких возмущений возможно, поскольку имеет место очевидная лемма.

<u>Лемма 1.</u> Пусть G_A и G_B – изоморфные ОС2-графы, P – матрица перестановки, соответствующая изоморфизму $\varphi: V_B \to V_A$. Пусть A^j и B^j – матрицы, получаемые из A и B возмущениями их диагональных элементов в соответствии с последовательностями $\{j\}_{j=1}^n$ и $\{k_j\}_{j=1}^n$, где $\varphi(k_j) = j$. Тогда

$$A = PBP^{-1} \Leftrightarrow \forall j \; A^j = PB^jP^{-1}.$$

Обращение матриц производится путем решения систем линейных уравнений $A^i x = e_j, B^i x = e_j$, где $\{e_j\}_{j=1}^n$ – стандартный базис в R^n. Пусть $x_k = (x_{k1}, \dots, x_{kn})$ – k-й вектор-столбец матрицы $(A^i)^{-1}$. Тогда если графы G_A и G_B изоморфны, и φ – изоморфизм G_A на G_B, то:

$$\begin{bmatrix} x_{11} & \dots & x_{1i} & \dots & x_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \dots & x_{ii} & \dots & x_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{ni} & \dots & x_{nn} \end{bmatrix} =$$

$$= \begin{bmatrix} y_{\varphi(1)\varphi(1)} & \dots & y_{\varphi(1)\varphi(i)} & \dots & y_{\varphi(1)\varphi(n)} \\ \vdots & \ddots & \vdots & & \vdots \\ y_{\varphi(i)\varphi(1)} & \dots & y_{\varphi(i)\varphi(i)} & \dots & y_{\varphi(i)\varphi(n)} \\ \vdots & & \vdots & \ddots & \vdots \\ y_{\varphi(n)\varphi(1)} & \dots & y_{\varphi(n)\varphi(i)} & \dots & y_{\varphi(n)\varphi(n)} \end{bmatrix}.$$

Пусть

$$\Lambda_i \equiv \prod_{k=1}^{n-1} \tilde{\lambda}_k^i \Big/ \prod_{k=1}^{n} \lambda_k,$$

а $R(A) = \{\Lambda_{i_1}, \dots, \Lambda_{i_p}\}$ – это набор неповторяющихся значений Λ_i для матрицы A. Если $|R(A)| = n$, то есть если все диагональные элементы обратной матрицы к матрице A графа – разные, и G_A и G_B изоморфны, то, очевидно, изоморфизм может быть установлен без проведения возмущений по следующему правилу:

$$i = \varphi(j) \Leftrightarrow \Lambda_i = \Lambda_j.$$

Докажем следующую лемму.

<u>Лемма 2.</u> Пусть $A' = A + \varepsilon E^i$, $A' x_i = e_i$, $\Lambda_i = x'_{ii}$. Тогда может быть выбрано такое ε, что $\Lambda_i \neq \Lambda_j$ для любого $j \neq i$.

<u>Доказательство.</u> Пусть A_{pq} – алгебраическое дополнение элемента a_{pq} матрицы A.

$$x_{ii} = \frac{A_{ii}}{\det A}.$$

Поскольку элемент a_{ii} матрицы A – единственный изменяемый в ходе возмущения матрицы элемент, то $A'_{ii} = A_{ii}$, тогда как для любого $j \neq i$ получаем $A'_{jj} = A_{jj} + \varepsilon A_{ii,jj}$, где $A_{ii,jj}$ – определитель подматрицы матрицы A, получаемой удалением ее i-ой строки, i-го столбца, j-ой строки и j-го столбца. Изменение определителя при возмущении следующее: $\det A' = \det A + \varepsilon A_{ii}$. Получаем, что при надлежащем выборе ε для любого $j \neq i$

$$\Lambda_j = \frac{A_{jj} + \varepsilon A_{jj,ii}}{\det A + \varepsilon A_{ii}} \neq \frac{A_{ii}}{\det A + \varepsilon A_{ii}} = \Lambda_i .$$

Лемма доказана.

Учитывая леммы 1 и 2, можно утверждать, что не более чем за n последовательных согласованных возмущений матриц изоморфных ОС2-графов мы можем добиться того, что

$$| R(A^{n_0}) |=| R(B^{n_0}) |= n ,$$

где n_0 – необходимое число возмущений, $n_0 \le n$.

Принципиальная схема алгоритма

Шаг 0. $i := 1$, $A^0 := A$, $B^0 := B$.

Шаг 1. Если $| R(A^{i-1}) |= n$, то перейти на шаг 3, иначе перейти на шаг 2.1.

i -я итерация алгоритма:

Шаг 2.0. Выбор ε.

Шаг 2.1. $A^i := A^{i-1} + \varepsilon E^i$.

Шаг 2.2. Поиск j т. ч.

$$S(G_{A^i}) = S(G_{B^{i-1}+\varepsilon E^j}) .$$

Шаг 2.3. Если j не найден, то, либо поданные на вход графы не изоморфны, либо – не являются ОС2-графами. Работу алгоритма завершить, иначе перейти на следующий шаг.

Шаг 2.4. $i := i+1$. Перейти на шаг 1.

Шаг 3. Установить изоморфизм по правилу

$$i = \varphi(j) \Leftrightarrow \Lambda_i = \Lambda_j .$$

Вычислительная эффективность алгоритма

Локализация элементов множества R

Пусть $R(A^{(j)}) = \left\{\Lambda_1^{(j)}, \dots, \Lambda_p^{(j)}\right\}$ – набор R к j -ой итерации. Обозначим решение системы уравнений $A^j x = e_k$, как $x_k^{(j)}$, и определим множества $R_i(A^{(j)})$ как

$$R_i(A^j) = \left\{x_{i_1 i_1}^{(j)} \mid x_{i_1 i_1}^{(j)} = \Lambda_i^{(j)}\right\}, i = \overline{1, p}.$$

Рассмотрим множества $R_i(A^j)$ в процессе производимых в ходе работы алгоритма возмущений матрицы A. Будем говорить, что $R_i(A^{j-1})$ расщепляется на j -ой итерации алгоритма, если

$$x_{jj}^{(j-1)} \in R_i(A^{j-1}) \ , \ | R_i(A^{j-1}) |> 1$$

для некоторого i, но

$$x_{jj}^{(j)} \in R_k(A^j) \text{, и } | R_k(A^j) |= 1 .$$

Положим,

$$\Delta_{kl}^{(j)} \equiv \left| x_{k_1 k_1}^{(j)} - x_{l_1 l_1}^{(j)} \right|,$$

где $x_{k_1 k_1}^{(j)} \in R_k(A^j), x_{l_1 l_1}^{(j)} \in R_l(A^j)$, $x_{l_1 l_1}^{(j)} \in R_l(A^j)$. Будем рассматривать $\Delta_{kl}^{(j)}$ как расстояние между множествами $R_k(A^j)$ и $R_l(A^j)$.

Числом обусловленности матрицы A называется следующая величина $\mu(A)$:

$$\mu(A) = \sup_{x \neq 0, \xi \neq 0} \left\{ \frac{\|Ax\| \cdot \|\xi\|}{\|A\xi\| \cdot \|x\|} \right\},$$

где $\|\cdot\|$ – евклидова норма вектора в R^n. Пусть $\lambda_{\max}$ – наибольшее собственное значение матрицы A, $\lambda_{\min}$ – наименьшее.

$$\mu(A) = \frac{\sup\limits_{x \neq 0} \|Ax\| / \|x\|}{\sup\limits_{\xi \neq 0} \|A\xi\| / \|\xi\|} = \frac{\lambda_{\max}}{\lambda_{\min}} < \infty .$$

$\lambda_{\min} \neq 0$, поскольку A – положительно определенная матрица.

Рассмотрим систему $(A+C)y = f$, полученную из системы $Ax = f$ возмущением A при помощи матрицы C. Пусть

$$\frac{\|C\|}{\|A\|} \le \theta ,$$

где $\|A\| = \sup\limits_{x \neq 0} \dfrac{\|Ax\|}{\|x\|}$.

$\|A\| = \lambda_{\max}$ для положительно определенной матрицы A [8]. Имеет место следующая теорема [8].

Теорема 1. Если $\vartheta\mu(A) < 1$, то

$$\frac{\|y - x\|}{\|x\|} \le \frac{\theta\mu(A)}{1 - \theta\mu(A)} .$$

Для числа обусловленности симметричной матрицы выполняется [8]:

$$\mu(A) \le \frac{\eta(A)}{\chi(A)},$$

где

$$\eta(A) = \max_{1 \le i \le n} \left(a_{ii} + \sum_{j \neq i} | a_{ij} | \right),$$

$$\chi(A) = \min_{1 \le i \le n} \left(a_{ii} - \sum_{j \neq i} | a_{ij} | \right).$$

На j -ой итерации $a_{kk} = d + d_k + \varepsilon_k$, где ε_k – значение возмущения k -го диагонального элемента A, и $\varepsilon_k = 0$ если $k < j$. Пусть i – номер строки, на которой достигается $\eta(A)$, i_2 – номер строки, на которой достигается $\chi(A)$. Имеем

$$\eta(A) = a_{i_1 i_1} + \varepsilon_{i_1} + \sum_{j=1}^{n} |a_{i_1 j}| = d + \varepsilon_{i_1} + d_{i_1} +$$
$$+ d_{i_1} = d + \varepsilon_{i_1} + d + d = 3d + \varepsilon_{i_1},$$

$$\chi(A) = a_{i_2 i_2} + \varepsilon_{i_2} + \sum_{j=1}^{n} |a_{i_2 j}| =$$
$$= d + \varepsilon_{i_2} + d_{i_2} - d_{i_2} = d + \varepsilon_{i_2}.$$

Следовательно,

$$\mu(A) \le \frac{\eta(A)}{\chi(A)} = \frac{3d + \varepsilon_{i_1}}{d + \varepsilon_{i_2}}.$$

Если $\varepsilon_{i_1} = 1/n^{p_1}$, $\varepsilon_{i_2} = 1/n^{p_2}$, где $p_1, p_2 \in N$, то

$$\mu(A) \le \frac{3d + \varepsilon_{i_1}}{d + \varepsilon_{i_2}} = \frac{3d + 1/n^{p_1}}{d + 1/n^{p_2}} =$$

$$= n^{p_2 - p_1} \frac{3dn^{p_1} + 1}{dn^{p_2} + 1} \le n^{p_2 - p_1} \frac{4dn^{p_1}}{dn^{p_2}} = 4.$$

Таким образом, в нашем случае утверждение теоремы 1 принимает следующий вид:

Если $\theta < 1/4$, то

$$\|y - x\| \le \theta \frac{4}{1 - 4\theta} \|x\|. \tag{2}$$

Для гарантированного выделения $x_{jj}^{(j-1)}$ из $R_i(A^{j-1})$ на j-ой итерации возмущения должны быть достаточно малы, поскольку необходимо, чтобы после возмущения $x_{jj}^{(j)}$ был достаточно удален от всех остальных значений $x_{ii}^{(j)}$ в смысле введенного расстояния Δ.

Из (2) следует, что

$$\|x_i^{(j)} - x_i^{(j-1)}\| \le \frac{4\theta(\varepsilon_j)}{1 - 4\theta(\varepsilon_j)} \|x_i^{(j-1)}\|.$$

Поскольку

$$\frac{\|C\|}{\|A^{j-1}\|} = \frac{\|\varepsilon_j E^j\|}{\|A^{j-1}\|} = \frac{\varepsilon_j}{\|A^{j-1}\|},$$

мы можем положить

$$\theta(\varepsilon_j) = \frac{\varepsilon_j}{\|A^{j-1}\|}.$$

Полагая $\varepsilon_j = 1/n^p$, получаем

$$\|x_i^{(j)} - x_i^{(j-1)}\| < \frac{4/n^p}{\|A^j\| - 4/n^p} \|x_i^{(j-1)}\| =$$

$$= \frac{4/n^p}{\|A^j\| - 4/n^p} \|x_i^{(j-1)}\| = \tag{3}$$

$$= \frac{4}{\|A^j\| n^p - 4} \|x_i^{(j-1)}\| \le \frac{5}{\|A^j\| n^p} \|x_i^{(j-1)}\|.$$

Имеем

$$\mu(A^{j-1}) = \frac{\lambda_{\max}}{\lambda_{\min}} \le 4,$$

где $\lambda_{\max}$ – максимальное собственное значение A^{j-1}, $\lambda_{\min}$ – минимальное. По теореме Гершгорина [9] $\lambda_{\max} \ge d$. Учитывая, что $\lambda_{\min} \le \|A^{j-1}\|$, получаем

$$\|A^{j-1}\| > \lambda_{\min} > \lambda_{\max}/4 > d/4.$$

Таким образом, из (3) следует

$$\|x_i^{(j)} - x_i^{(j-1)}\| < \frac{20}{dn^p} \|x_i^{(j-1)}\|.$$

Так как все собственные значения A^j не меньше 1, и $\|e_k\| = 1$ для любого k, то $\|x_i^{(j-1)}\| < 1$. Из (3) следует, что для любого i

$$\|x_i^{(j)} - x_i^{(j-1)}\| < \frac{20}{dn^p}.$$

$$|x_{ii}^{(j)} - x_{ii}^{(j-1)}| < \|x_i^{(j)} - x_i^{(j-1)}\|,$$

следовательно, для любого i

$$|x_{ii}^{(j)} - x_{ii}^{(j-1)}| < \frac{20}{dn^p}. \tag{4}$$

Неравенство (4) дает оценку локализации векторов из множеств $R_i(A^j)$ в процессе работы алгоритма.

Расщепление множеств R_i

Производимые возмущения должны быть достаточны для расщепления множеств R_i. Алгоритм является вычислительно эффективным только в том случае, если для любого j и любых k и l, не равных друг другу, $\Delta_{kl}^{(j)}$ будет отличным от нуля машинным числом. Покажем, что необходимая точность решения систем линейных уравнений может быть обеспечена при помощи машинных чисел с длиной мантиссы, не превышающей n.

Пусть $x_{ii}^{(j-1)}, x_{jj}^{(j-1)} \in R_i(A^{j-1})$ для некоторого i, то есть перед j-ой итерацией в этом множестве имеется по меньшей мере два элемента, препятствующих установлению взаимнооднозначного соответствия между вершинами графов.

<u>Теорема 2.</u> Если $x_{ii}^{(j-1)} = x_{jj}^{(j-1)}$ и $0 < \varepsilon_j < 1$, то

$$|x_{ii}^{(j)} - x_{jj}^{(j)}| > \frac{1}{3^n d^2 (3d/\varepsilon_j + 1)}.$$

<u>Доказательство.</u> Пусть $A = A^{j-1}$, $A' = A^j$, $x_i = x_i^{(j-1)}$, $x_i' = x_i^{(j)}$, $x_j = x_j^{(j-1)}$, $x_j' = x_j^{(j)}$. Если $x_j' = Px_i'$, то $x_{jj}' = x_{ii}'$.

$$x'_{jj} = \frac{A'_{jj}}{\det A'}, x'_{ii} = \frac{A'_{ii}}{\det A'}.$$

После возмущения

$$\det A' = \det A + \varepsilon_j A_{jj}, A'_{jj} = A_{jj},$$

так как a_{jj} – единственный изменяемый элемент A на итерации алгоритма. Поскольку

$$A'_{ii} = A_{ii} + \varepsilon_j A_{ii,jj},$$

и $A_{ii} = A_{jj}$, так как $x_{ii} = x_{jj}$, получаем

$$|x'_{jj} - x'_{ii}| = \left| \frac{A'_{jj}}{\det A'} - \frac{A'_{ii}}{\det A'} \right| =$$

$$= \left| \frac{A_{jj}}{\det A'} - \frac{A_{ii} + \varepsilon_j A_{ii,jj}}{\det A'} \right| = \frac{\varepsilon_j A_{ii,jj}}{\det A + \varepsilon_j A_{jj}}. \quad (5)$$

$A_{ii,jj}$ – определитель подматрицы A, получаемой удалением из нее i-х строки и столбца и j-х строки и столбца. A – симметричная положительно определенная матрица с диагональным преобладанием, для которой выполняются условия Адамара:

$$H_k \equiv |a_{kk}| - \sum_{l \neq k} |a_{kl}| = d_k > 0, i = \overline{1,n},$$

где

$$d'_k = \begin{cases} d + \varepsilon_k, \text{ если } 1 \leq k \leq j, \\ d, \quad \text{если } 1 \leq k \leq j. \end{cases}$$

Следовательно,

$$\begin{aligned} A_{ii,jj} &\geq H_1 \times \ldots \times \overline{H}_i \times \ldots \times \overline{H}_j \times \ldots \times H_n = \\ &= d'_1 \times \ldots \times \overline{\widehat{d'}_i} \times \ldots \times \overline{\widehat{d'}_j} \times \ldots \times d'_n > d^{n-2}, \end{aligned} \quad (6)$$

черта сверху означает невключение в произведение.

С другой стороны, по теореме Гершгорина максимальное собственное значение $\lambda_{\max}$ матрицы A можно оценить так:

$$\lambda_{\max} \leq 3d + \max_{1 \leq k \leq j} \varepsilon_k < 3d + 1,$$

если $\varepsilon_k < 1$ для любого k. Поэтому

$$\det A = \prod_{k=1}^{n} \lambda_k < \lambda_{\max}^n < (3d+1)^n, \quad (7)$$

$$A_{jj} = \prod_{k=1}^{n-1} \lambda'_k < \lambda_{\max}^{n-1} < (3d+1)^{n-1}. \quad (8)$$

Пусть $\lambda'_{\max}$ – максимальное собственное значение подматрицы, получаемой из A удалением строк и столбцов с номерами i и j. $\lambda'_{\max} \leq \lambda_{\max}$ [9]. Используя (6), (7) и (8), из (5) получаем:

$$\begin{aligned} |x'_{ii} - x'_{jj}| &= \frac{\varepsilon_j A_{ii,jj}}{\det A + \varepsilon_j A_{jj}} > \\ &> \frac{\varepsilon_j d^{n-2}}{(3d+1)^n + \varepsilon_j (3d+1)^{n-1}} > \\ &> \frac{\varepsilon_j d^{n-2}}{(3d)^{n+1} + \varepsilon_j (3d)^n}, \end{aligned}$$

то есть

$$\left| x_{ii}^{(j)} - x_{jj}^{(j)} \right| > \frac{1}{3^n d^2 (3d/\varepsilon_j + 1)}. \quad (9)$$

Теорема доказана.

Пусть решение систем линейных уравнений $A^j x = e_k$ и $B^j x = e_k$ $(j,k = \overline{1,n})$ осуществляется методом Гаусса-Зейделя. Метод Гаусса-Зейделя позволяет оценить необходимую длину мантиссы используемых машинных чисел. В действительности не принципиально, каким методом мы решаем системы уравнений, необходимо только, чтобы решение находилось с нужной точностью за приемлемое время.

Имеет место [10] следующая теорема.

Теорема 3. Пусть

$$\sum_{j \neq i} |a_{ij}| \leq \gamma |a_{ii}|, \ \gamma < 1,$$

для каждого $i = \overline{1,n}$. Тогда

$$\left\| x - \tilde{x}^s \right\| \leq \gamma \left\| x - \tilde{x}^{s-1} \right\|,$$

где x – точное решение системы уравнений $Ax = b$, $\tilde{x}_i^s$ – приближенное решение этой же системы уравнений на s-ой итерации метода Гаусса-Зейделя.

Если $A \equiv A^j$, то $\gamma \leq 1/2$. Следовательно, на s-ой итерации метода Гаусса-Зейделя

$$\left| x_{ii} - \tilde{x}_{ii}^s \right| \leq \left\| x_i - \tilde{x}_i^s \right\| < \frac{1}{2^s} \delta^0,$$

$$\left| x_{jj} - \tilde{x}_{jj}^s \right| \leq \left\| x_j - \tilde{x}_j^s \right\| < \frac{1}{2^s} \delta^0,$$

где δ^0 – ошибка начального приближения. Если $\varepsilon_j = 1/n^p$ на j-ой итерации, то, учитывая (9), получаем

$$\left| x_{jj}^{(j)} - x_{ii}^{(j)} \right| > \frac{1}{3^n d^2 (3dn^p + 1)}.$$

Значит, если

$$\frac{1}{2^{s-1}} \delta^0 \leq \frac{1}{3^n d^2 (3dn^p + 1)}, \quad (10)$$

то разница $\left| x_{jj}^{(j)} - x_{ii}^{(j)} \right|$ может быть зафиксирована машинными числами с длиной мантиссы, не превышающей n. Оценим количество итераций метода

Гаусса-Зейделя s, необходимых для фиксирования этого значения.

Формула (10) эквивалентна

$$3^n d^2(3dn^p+1)\delta^0 \le 2^{s-1}. \qquad (11)$$

Логарифмируя (11), получаем

$$n\log_2 3+\log_2 d^2(3dn^p+1)+\log_2\delta^0 \le s-1,$$

то есть число необходимых итераций метода Гаусса-Зейделя может быть оценено как

$$s \ge n\log_2 3+p\log_2 n+3\log_2 d+\log_2\delta^0+3. \qquad (12)$$

Пусть $p>0$ определяет значение возмущения $\varepsilon_j = 1/n^p$. Пусть $x_{jj}^{(j-1)}$ – значение, выделяемое из $R_i(A^{j-1})$ на j-ой итерации. Выберем p так, что

$$\frac{20}{dn^p} < \frac{\Delta}{n},$$

то есть

$$p > \log_n \frac{20}{d\Delta}+1,$$

где $\Delta = \min_{k\ne l}\Delta_{kl}^{(j)}$.

Из (4) следует, что для любого i

$$\left|x_{ii}^{(j-1)}-x_{ii}^{(j)}\right| < \frac{20}{dn^p} < \frac{\Delta}{n}.$$

Получаем

$$\frac{1}{3^n d^2(3d/\varepsilon_j+1)} < \left|x_{ii}^{(j)}-x_{jj}^{(j)}\right| < \frac{\Delta}{n}$$

после j-ой итерации. Это означает, что на j-ой итерации мы можем выбрать такое возмущение, которое давало бы выделение $x_{jj}^{(j-1)}$ из содержащего его множества $R_i(A^{j-1})$ (если его мощность не равна 1), фиксируемое при помощи машинных чисел с длиной мантиссы равной n при сохранении локализации значений x_{ii}, $i=\overline{1,n}$. То есть могут быть заданы интервалы, в пределах которых будут происходить расщепления множеств R_i. Возмущения могут проведены таким образом, что после выделения x_{jj} из некоторого $R_i(A^{j-1})$, x_{jj} не попадает на следующей итерации в другое такое множество с мощностью отличной от 1.

В результате, если для решения систем линейных уравнений производится число итераций метода Гаусса-Зейделя, определяемое (12), то при $n_0 \le n$ мы получаем $|R(A^{n_0})| = n$, и расщепление элементов из R фиксируется машинными числами с длиной мантиссы, не превышающей n.

Общая вычислительная сложность алгоритма

На каждой итерации алгоритма мы проводим вычисление наборов R для обоих графов, для чего мы обращаем матрицы графов. Для обращения матриц необходимо решить $2n$ систем линейных уравнений. Пусть $O(N_X)$ – число элементарных машинных операций, которые необходимо совершить для решения систем линейных уравнений с заданной точностью. Тогда сложность процедуры обращения матриц составит $2n\cdot O(N_X)$.

Проверка равенства значений из $R(A)$ и $R(B)$ (проверка равенства значений $S(A)$ и $S(B)$) для пары матриц A и B осуществима за $O(n)$ элементарных машинных операций. Следовательно, общая сложность одной итерации алгоритма составляет

$$2n\cdot O(N_X)+O(n). \qquad (13)$$

Оценим $O(N_X)$ для метода Гаусса-Зейделя. На j-ой итерации алгоритма, находя ε_j, мы вычисляем p такое, что $20/dn^p < \Delta_j/n$, где $\Delta_j = \min_{k\ne l}\Delta_{kl}^{(j)}$, то есть

$$p = \log_n \frac{16}{d\Delta_j}+1.$$

После чего, в соответствии с (12), выполняется s итераций метода Гаусса-Зейделя. Пусть N_X^j – сложность нахождения решений систем линейных уравнений на j-ой итерации алгоритма проверки изоморфизма графов. Тогда, с учетом (13),

$$N_X^j = O(n^2)\cdot\left(n\log_2 3+\left(\log_n\frac{1}{\Delta_j}+1\right)\log_2 n+\right.$$
$$\left.+3\log_2 d+\log_2\delta^0+3\right),$$

поскольку одна итерация метода Гаусса-Зейделя требует $O(n^2)$ элементарных машинных операций. Точные решения систем линейных уравнений принадлежат отрезку $[0,1]^n \in R^n$, следовательно, мы можем положить $\delta^0 = O(\sqrt{n})$. С учетом (13) общая вычислительная сложность алгоритма может быть оценена как

$$O\left(n\times\max_{1\le j\le n}\left(n^2 n+\log_n\frac{1}{\Delta_j}\log_2 n+\log_2 n\right)\right)= \qquad (14)$$
$$=O(n^4).$$

Заключение

В класс ОС2-графов, для которых алгоритм решает задачу проверки изоморфизма графов, как и в класс ОС-графов, не попадают известные серии сильно-регулярных графов и некоторые другие, традиционно сложные для решения задачи ИГ графы.

Но, как показывает вычислительный эксперимент, в класс ОС2-графов попадают не только ОС-графы, но и многие графы, таковыми не являющиеся. Так, в библиотеке тестовых задач [11], использовавшейся для сравнения алгоритмов решения задачи ИГ на основе рекурсии с возвратом, не найдено примеров неправильного решения задачи ИГ представляемым алгоритмом для деревьев, случайных, планарных графов, регулярных n-мерных сеток ($n \le 4$) и некоторых других. Установлено также, что алгоритм решает и те задачи ИГ из этой библиотеки, на которых время работы алгоритмов Ullman и NAUTY становится экспоненциальным.

Несмотря на то, что в обосновании алгоритма длина мантиссы предполагается равной n, при проведении вычислительных экспериментов достаточным было использование числового типа extended, реализованного в языках Object Pascal и C++, позволяющего работать с машинными числами в диапазоне от $3{,}6\times10^{-4951}$ до $1{,}1\times10^{4932}$.

Литература

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. - М.: Мир, 1982.
2. Hopcroft, Wong A linear time algorithm for isomorphism of planar graphs // Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, 1974. - P. 172-184.
3. Luks Isomorphism of graphs of bounded valence can be tested in polynomial time // Proceedings of the 21-st IEEE FOCS Symp., 1980. - 42, 49,
4. Hoffmann Group-Theoretic Algorithms and Graph Isomorphism // Lecture Notes in Computer Science 1982. - Chapter V. - P.127-138,
5. Babai L., Grigoryev D., Mount D. Isomorphism of graphs with bounded eigenvalue multiplicity // Proc. 14th ACM symp. On theory of comput, STOC, 1982. - P. 310-324.
6. Цветкович Д. и др. Спектры графов. Теория и применение. - Киев: Наукова думка, 1984.
7. Faizullin R., A. Prolubnikov An algorithm of the spectral splitting for the double permutation cipher // Pattern recognition and image analysis. MAIK, Nauka. 2002. - Vol. 12. - No. 4. - P. 365-375.
8. Годунов С.К. и др. Гарантированная точность решения систем линейных уравнений в евклидовых пространствах. - Новосибирск: Наука, 1988.
9. Гантмахер Ф.Р. Теория матриц. - М.: Наука, 1976.
10. Бахвалов Н.С. Численные методы. - М.: Наука, 1975. - Т.1.
11. Foggia P., Sansone C., Vento M. A Database of graphs for isomorphism and sub-graph isomorphism benchmarking // Proc. of the 3rd IAPR TC-15 international workshop on graph-based representations, Italy, 2001. - P.157-168.

THE DIRECT ALGORITHM FOR CHECKING GRAPHS ISOMORPHISM

A.V. Prolubnikov[1]
[1]*Omsk State University, Omsk, Russia*

Abstract

This paper presents an algorithm for solving the graphs isomorphism problem. The algorithm is direct in the sense that it is not a modification of the backtracking recursion scheme upon which the most efficient algorithms are built to solve the problem. The solution of the problem is out of the number of algorithm iterations not exceeding the number of nodes of graphs. It is shown that the algorithm is polynomial by the number of the used elementary computer operations and the used memory. The graph class where the algorithm provides solution to the problem is studied.

Keywords: graphs isomorphism problem, backtracking recursion scheme, nodes of graphs, polynomial algorithm

Citation: Prolubnikov AV. The direct algorithm for checking graphs isomorphism [In Russian]. Computer Optics 2007; 31(3): 86-92.

References:

[1] Garey M, Johnson D. Computers and Intractability: A Guide to the Theory of NP-Completeness [In Russian]. Moscow: “Mir” Publisher, 1982.
[2] Hopcroft Wong. A linear time algorithm for isomorphism of planar graphs. Proceedings of the Sixth Annual ACM Symposium on Theory of Computing 1974; 172-184.
[3] Luks Isomorphism of graphs of bounded valence can be tested in polynomial time. Proceedings of the 21-st IEEE FOCS Symp. 1980; 42: 49.
[4] Hoffmann Group-Theoretic Algorithms and Graph Isomorphism. Lecture Notes in Computer Science 1982; V: 127-138.
[5] Babai L, Grigoryev D, Mount D. Isomorphism of graphs with bounded eigenvalue multiplicity. Proc. 14th ACM symp. On theory of comput. STOC 1982; 310-324.
[6] Cvetkovic D, Doob M, Sachs H. Spectra of graphs. Theory and application [In Russian]. Kiev: Naukova Dumka, 1984.
[7] Faizullin R, Prolubnikov A. An algorithm of the spectral splitting for the double permutation cipher. Pattern recognition and image analysis 2002; 12(4): 365-375.
[8] Godunov SK, Antonov AG, Kirilyuk OP, Kostin VI. Guaranteed accuracy of solving linear systems in Euclidean spaces [In Russian]. Novosibirsk: “Nauka” Publisher, 1988.
[9] Gantmacher FR. The theory of matrices [In Russian]. Moscow: “Nauka” Publisher, 1976.
[10] Bakhvalov NS. Numerical methods [In Russian]. Moscow: “”Nauka” Publisher, 1975; 1.
[11] Foggia P, Sansone C, Vento M. A Database of graphs for isomorphism and sub-graph isomorphism benchmarking. Proc. of the 3rd IAPR TC-15 international workshop on graph-based representations, Italy, 2001; 157-168.