

КЛАСС ГРАФОВ, ЗАДАЧА ПРОВЕРКИ ИЗОМОРФИЗМА ДЛЯ КОТОРЫХ РАЗРЕШИМА ЗА ПОЛИНОМИАЛЬНОЕ ВРЕМЯ АЛГОРИТМОМ СПЕКТРАЛЬНОГО РАСЩЕПЛЕНИЯ

А.В. Пролубников, Р.Т. Файзуллин

We propose an algorithm for solving the graph isomorphism problem. The problem is not known to be *NP*-complete or to be solvable in polynomial time for every pair of input graphs [1, p. 194]. Main heuristic used by the algorithm is inverse matrices of graphs' adjacency matrices which modified to positive defined matrices. These modified adjacency matrices can be viewed as adjacency matrices of graphs with weighted loops. The two initial input graphs are isomorphic if and only if the two obtained graphs with weighted loops are isomorphic. The graphs' automorphism groups are consequently simplified and finally become trivial during iterations of the algorithm by perturbations of graphs (perturbations of their modified adjacency matrices). So we can solve the isomorphism problem both for perturbed graphs with weighted loops and so for the initial input graphs. Some results relating to computational efficiency are considered. It is shown that our algorithm can solve the isomorphism problem for the class of graphs for which the problem is solvable at polynomial time by the most universal algorithm designed for the graph isomorphism problem [2, p. 310–324].

Введение

Задача проверки изоморфизма графов принадлежит к задачам, относительно которых нет ясности: являются ли они полиномиально разрешимыми или нет [1, с. 194]. В то же время известно, что задача проверки изоморфизма графа является полиномиально разрешимой для некоторых классов графов. В частности, для планарных графов, регулярных графов с ограниченной степенью вершин, графов с ограниченной кратностью собственных значений их матриц смежности и некоторых других построены эффективные алгоритмы для решения этой задачи [2, с. 310–324], [3, с. 172–184], [4, с. 42–49].

Предлагаемый нами алгоритм является эвристическим алгоритмом для проверки изоморфизма невзвешенных неориентированных графов. Алгоритм рабо-

© 2003 А.В. Пролубников, Р.Т. Файзуллин

E-mail: faizullin@univer.omsk.su, prolubnikov@math.omsu.omskreg.ru

Омский государственный университет

тает с модифицированными до положительно определенных матрицами смежности графов и основан на решении связанных с ними систем линейных уравнений, задающих обратные матрицы. Алгоритм не является комбинаторным в том смысле, что при решении задачи проверки изоморфизма графов в ходе работы алгоритма не происходит перебора вариантов в соответствии с некоторым деревом поиска, как это делается в модификациях процедуры backtracking, предназначенных для решения данной задачи [5, с. 110–119], [6, с. 31–42], [7, с. 433–445]. Построение изоморфизма происходит без неконтролируемого ветвления на всех итерациях алгоритма: на каждой итерации проверяется не более n возможных вариантов для установления соответствия между вершинами. В случае если графы изоморфны, то устанавливаемое соответствие – изоморфизм. Изоморфны графы или нет устанавливается в худшем случае за n полных итераций алгоритма, где n – число вершин в графе. В случае изоморфности графов предъявляется один из возможных изоморфизмов.

Задача проверки изоморфизма невзвешенных неориентированных графов формулируется следующим образом.

Постановка задачи № 1. Даны два невзвешенных неориентированных графа $G_A\langle V_A, E_A \rangle$ и $G_B\langle V_B, E_B \rangle$, где V_A, V_B – множества вершин графов, E_A, E_B – множества ребер графов. $|V_A| = |V_B|$, $|E_A| = |E_B|$.

Графы $G_A\langle V_A, E_A \rangle$ и $G_B\langle V_B, E_B \rangle$ изоморфны ($G_A\langle V_A, E_A \rangle \cong G_B\langle V_B, E_B \rangle$) тогда и только тогда, когда существует такое биективное отображение $\varphi : V_A \rightarrow V_B$, что $(i, j) \in E_A \Leftrightarrow (\varphi(i), \varphi(j)) \in E_B$. Требуется найти такое φ , или доказать, что его не существует.

Очевидно, графы G_A и графы G_B с одинаковым количеством вершин изоморфны тогда и только тогда, когда их вершины можно занумеровать так, что соответствующие матрицы смежности будут равны. Отношение изоморфизма является отношением эквивалентности, разбивающим множества всех графов на классы эквивалентности, которые можно рассматривать как абстрактные графы. Таким образом, изоморфные графы представляют собой один и тот же абстрактный граф.

Абстрактный граф однозначно определяется своей матрицей смежности, но обратное утверждение, вообще говоря, не имеет места, так как упорядочение (нумерация) вершин графа произвольное: каждому графу соответствует класс матриц смежности; две матрицы A и B принадлежат одному классу тогда и только тогда, когда существует матрица перестановки P такая, что $A = PBP^{-1}$. (Учитывая, что матрицей обратной к матрице перестановки P является матрица P^T , это условие равносильно $A = PBP^T$.) А это значит, что матрица A может быть получена из матрицы B при помощи некоторой перестановки строк матрицы вместе с точно такой же перестановкой ее столбцов. Соединение перестановки строк в квадратной матрице с такой же перестановкой ее столбцов будем в дальнейшем называть *перестановкой рядов* матрицы.

Биективному отображению $\varphi : V_A \rightarrow V_B$ однозначно соответствует перестановка P_φ , действующая на множестве вершин графа G_A . Если P_φ представлена матрицей перестановки (также обозначаемой P_φ), то P является изоморфизмом

тогда и только тогда, когда $A = P_\varphi B P_\varphi^{-1}$, где A – матрица смежности графа G_A , B – матрица смежности графа G_B .

Пусть A_0 – матрица смежности невзвешенного неориентированного графа G_A , то есть $A_0 = (a_{ij}^0)$, где

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

B_0 – матрица смежности графа G_B .

Любому биективному отображению $\varphi : V_A \rightarrow V_B$ может быть однозначно поставлена в соответствие некоторая матрица перестановки $P_\varphi = (p_{ij})$, задаваемая следующим образом.

$$p_{ij} = \begin{cases} 1, & \text{если } i = \varphi(j), \\ 0, & \text{иначе.} \end{cases}$$

Умножение некоторой матрицы слева на матрицу P_φ задает перестановку ее строк, при которой j -я строка переходит в i -ю. Обратной матрицей к матрице P_φ будет матрица P_φ^{-1} – матрица, получаемая из матрицы P транспонированием, то есть матрица со следующими элементами p'_{ij} :

$$p'_{ij} = \begin{cases} 1, & \text{если } j = \varphi(i), \\ 0, & \text{иначе.} \end{cases}$$

Умножение некоторой матрицы справа на матрицу P_φ^{-1} задает перестановку ее столбцов, при которой j -й столбец переходит в i -й.

Значит, следующая постановка задачи проверки изоморфизма невзвешенных неориентированных графов эквивалентна постановке задачи №1.

Постановка задачи №2. Даны два невзвешенных неориентированных графа $G_A \langle V_A, E_A \rangle$ и $G_B \langle V_B, E_B \rangle$. $|V_A| = |V_B|$, $|E_A| = |E_B|$. A_0 , B_0 – матрицы смежности графов G_A , G_B .

Требуется найти матрицу перестановки P такую, что $A_0 = P B_0 P^{-1}$, или показать, что ее не существует. В постановке задачи выше матрица P – мат-

рица перестановки P_φ соответствующая изоморфизму $\varphi : V_A \rightarrow V_B$, биекции, существование или отсутствие которой равносильно изоморфности или, соответственно, неизоморфности графов G_A и G_B . Везде ниже, если будет рассматриваться некоторый один изоморфизм φ , то индекс φ матрицы P_φ будет опускаться.

Таким образом, между биективными отображениями $\varphi : V_A \rightarrow V_B$ и матрицами перестановки P размерности $n \times n$ может быть установлено взаимнооднозначное соответствие. При этом если $G_A \cong G_B$, то

$$\varphi : V_A \rightarrow V_B \text{ – изоморфизм} \Leftrightarrow A_0 = P_\varphi B_0 P_\varphi^{-1}.$$

То есть

$$\begin{pmatrix} a_{11}^0 & \dots & a_{1j}^0 & \dots & a_{1n}^0 \\ \vdots & \ddots & \vdots & & \vdots \\ a_{j1}^0 & \dots & a_{jj}^0 & \dots & a_{jn}^0 \\ \vdots & & \vdots & \ddots & \vdots \\ a_{jn}^0 & \dots & a_{nj}^0 & \dots & a_{nn}^0 \end{pmatrix} = \begin{pmatrix} b_{\varphi(1)\varphi(1)}^0 & \dots & b_{\varphi(1)\varphi(j)}^0 & \dots & b_{\varphi(1)\varphi(n)}^0 \\ \vdots & \ddots & \vdots & & \vdots \\ b_{\varphi(j)\varphi(1)}^0 & \dots & b_{\varphi(j)\varphi(j)}^0 & \dots & b_{\varphi(j)\varphi(n)}^0 \\ \vdots & & \vdots & \ddots & \vdots \\ b_{\varphi(n)\varphi(1)}^0 & \dots & b_{\varphi(n)\varphi(j)}^0 & \dots & b_{\varphi(n)\varphi(n)}^0 \end{pmatrix}.$$

Если φ – изоморфизм, то $(i, j) \in E_A \Leftrightarrow (\varphi(i), \varphi(j)) \in E_B$. Значит, $a_{ij}^0 = 1 \Leftrightarrow b_{\varphi(i)\varphi(j)}^0 = 1$ (матрицы A_0 и PB_0P^{-1} поэлементно равны). Обратно, если $A_0 = PB_0P^{-1}$, то $a_{\varphi(i)\varphi(j)}^0 = 1 \Leftrightarrow b_{ij}^0 = 1$. То есть $A_0 = (b_{\varphi(i)\varphi(j)}^0)$, и $(i, j) \in E_A \Leftrightarrow (\varphi(i), \varphi(j)) \in E_B$.

Биективное отображение φ и соответствующая ему перестановка P действующие на множестве вершин графа G_A , называются *автоморфизмом графа G_A* , если они сохраняют смежность его вершин. Если P представлена матрицей перестановки (также обозначаемой P), то P является автоморфизмом тогда и только тогда, когда $A = PBP^{-1}$, где A – матрица смежности графа G_A . Автоморфизмы графа G_A образуют группу $\Gamma = \Gamma(G_A)$, называемую *группой автоморфизмов графа G_A* , в терминах которой выражаются симметрии этого графа относительно перестановок вершин.

Группа Γ автоморфизмов графа G_A реализуется множеством всех матриц перестановок P , которые коммутируют с матрицей смежности A графа G_A :

$$P \in \Gamma \Leftrightarrow AP = PA.$$

1. Алгоритм спектрального расщепления проверки изоморфизма графов

Видоизменяются матрицы до положительно определенных следующим образом. Пусть A_0 – матрица смежности графа $G_A \langle V_A, E_A \rangle$. В соответствии с матрицей A_0 строим диагональную матрицу D_{A_0} :

$$\begin{pmatrix} d_{11}^A & 0 & \dots & 0 \\ 0 & d_{22}^A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_{nn}^A \end{pmatrix},$$

со следующими элементами на диагонали:

$$d_{ii}^A = \sum_{j=1}^n a_{ij}^0 + d = d_i + d,$$

где d – максимальная степень вершин в графе G_A , а d_i – степень вершины i . Аналогично по матрице B_0 строится матрица D_{B_0} .

Матрицы, с которыми работает алгоритм, имеют следующий вид:

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

Рассмотрим еще одну постановку задачи проверки изоморфизма невзвешенных неориентированных графов.

Постановка задачи 3. Даны два невзвешенных неориентированных графа $G_A \langle V_A, E_A \rangle$ и $G_B \langle V_B, E_B \rangle$. $|V_A| = |V_B|$, $|E_A| = |E_B|$. A , B – матрицы вида (1) для графов G_A , G_B .

Требуется найти матрицу перестановки P такую, что $A = PBP^{-1}$, или показать, что ее не существует.

Докажем, что постановка №3 задачи проверки невзвешенных неориентированных графов эквивалентна постановке задачи №2, и, следовательно, постановке №1 также.

Лемма 1. Если $G_A \cong G_B$, то существует матрица перестановки P , такая что $A = PBP^{-1}$.

Доказательство. $G_A \cong G_B$, значит существуют такая биекция $\varphi : V_A \rightarrow V_B$ и соответствующая ей матрица перестановки $P = P_\varphi$, задающие изоморфизм графов. Если φ – изоморфизм, то φ сохраняет степень вершин. То есть если $\varphi(i) = j$, то $d_i^A = d_j^B$, где d_i^A – степень вершины i в графе G_A , а d_j^B – степень вершины j в графе G_B . Следовательно, сумма элементов i -ой строки матрицы A_0 равна $\sum_{j=1}^n a_{ij} = d_i^A$, а сумма элементов j -ой строки матрицы B_0 равна $\sum_{i=1}^n b_{ij} = d_j^B$, и если $d_i^A = d_{\varphi(i)}^B = d_j^B$, то и $d_{ii}^A = d_{jj}^B$, так как максимальная степень вершин d в изоморфных графах G_A и G_B одна и та же.

Перестановка рядов матрицы B влечет перестановку диагональных элементов B из соответствующих рядов и эквивалентна той же перестановке рядов диагональной матрицы D_{B_0} , с диагональными элементами матрицы B . А так как $\forall i \in V_A, \forall j \in V_B$, из $\varphi(i) = j$ следует, что $d_{ii}^A = d_{\varphi(i)\varphi(i)}^B = d_{jj}^B$, то тогда, если P – соответствующая изоморфизму φ перестановка, $D_{A_0} = PD_{B_0}P^{-1}$. Следовательно, $PBP^{-1} = P(B_0 + D_{B_0})P^{-1} = PB_0P^{-1} + PD_{B_0}P^{-1} = A_0 + D_{A_0} = A$, то есть $A = PBP^{-1}$. ■

Лемма 2. Если $G_A \not\cong G_B$, то не существует матрицы перестановки P такой, что $A = PBP^{-1}$.

Доказательство. В силу постановки задачи № 2, если $G_A \not\cong G_B$, то не существует такой матрицы перестановки P' , что $A_0 = P'B_0P'^{-1}$. Допустим, что при этом существует некоторая матрица перестановки P , такая что $A = PBP^{-1}$.

По построению $A = A_0 + D_{A_0}$, $B = B_0 + D_{B_0}$, и

$$A = A_0 + D_{A_0} = PBP^{-1} = P(B_0 + D_{B_0})P^{-1} = PB_0P^{-1} + PD_{B_0}P^{-1},$$

то есть

$$A_0 + D_{A_0} = PB_0P^{-1} + PD_{B_0}P^{-1}. \quad (2)$$

Перестановка рядов матрицы B влечет перестановку диагональных элементов B из соответствующих рядов и эквивалентна перестановке рядов диагональной матрицы D_{B_0} , диагональные элементы которой – диагональные элементы матрицы B . При этом диагональные элементы матрицы B переходят в диагональные элементы матрицы A , так как $A = PBP^{-1}$. То есть $D_{A_0} = PD_{B_0}P^{-1}$, и из (2) следует $A_0 = PB_0P^{-1}$. Но по условию утверждения леммы такой перестановки не существует. Из полученного противоречия следует, что допущение о существовании матрицы перестановки P такой, что $A = PBP^{-1}$ неверно. ■

Таким образом эквивалентность постановки задачи №3 постановкам №1 и №2 задачи проверки изоморфизма невзвешенных неориентированных графов доказана. Постановку задачи №3 можно рассматривать как сведение задачи проверки изоморфизма неориентированных графов G_A и G_B к задаче проверки изоморфизма неориентированных мультиграфов G'_A и G'_B с матрицами смежности (1), то есть их каждая вершина имеет петлю с определенным весом. Все остальные ребра построенного мультиграфа, кроме указанных петель, совпадающие с соответствующими ребрами графов G_A и G_B – не взвешенные и не кратные.

Рассмотрим следующие системы линейных уравнений:

$$Ax = e_j, \quad By = e_k, \quad (3)$$

где $e_j = (0, \dots, 0, 1, 0, \dots, 0)$ – j -й орт в пространстве R^n , а матрицы A и B – матрицы, построенные по матрицам смежности графов G_A и G_B так, как показано выше. Матрицы A и B – симметрические положительно определенные матрицы. Каждая система линейных уравнений из (3) имеет решение, и решение единственно, так как матрицы A и B – матрицы с диагональным преобладанием и их определители не равны нулю. Обозначим как x_j решения систем уравнений $Ax = e_j$, как y_k – решения систем уравнений $By = e_k$.

Решая системы уравнений (3), в качестве решений мы получаем столбцы обратных матриц к матрицам A и B , так как для i -ой компоненты вектора x_j справедливо

$$x_{ji} = \frac{A_{ij}}{|A|},$$

где A_{ij} – алгебраическое дополнение элемента a_{ij} матрицы A .

$A = PBP^{-1}$ тогда и только тогда, когда $A^{-1} = PB^{-1}P^{-1}$:

$$A = PBP^{-1} \Leftrightarrow AP = PB \Leftrightarrow P = A^{-1}PB \Leftrightarrow PB^{-1} = A^{-1}P$$

$$\Leftrightarrow PB^{-1}P^{-1} = A^{-1}.$$

Пусть φ – изоморфизм графов G_A и G_B , P – матрица перестановки, соответствующая φ , то есть матрица, задающая перестановку, определяемую подстановкой

$$\left(\begin{array}{ccccc} 1 & \dots & j & \dots & n \\ \varphi(1) & \dots & \varphi(j) & \dots & \varphi(n) \end{array} \right);$$

x_j – решение системы уравнений $Ax = e_j$, y_j – решение системы уравнений $Bu = e_j$. Так как x_j – j -й столбец матрицы A^{-1} , а y_j – j -й столбец матрицы B^{-1} , то

$$\begin{pmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{j1} & \dots & x_{jj} & \dots & x_{jn} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nj} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} y_{\varphi(1)\varphi(1)} & \dots & y_{\varphi(1)\varphi(j)} & \dots & y_{\varphi(1)\varphi(n)} \\ \vdots & \ddots & \vdots & & \vdots \\ y_{\varphi(j)\varphi(1)} & \dots & y_{\varphi(j)\varphi(j)} & \dots & y_{\varphi(j)\varphi(n)} \\ \vdots & & \vdots & \ddots & \vdots \\ y_{\varphi(n)\varphi(1)} & \dots & y_{\varphi(n)\varphi(j)} & \dots & y_{\varphi(n)\varphi(n)} \end{pmatrix}. \quad (4)$$

Значит, при изменении в системах линейных уравнений (3) k от 1 до n при фиксированном j , вектор x_j будет совпадать с некоторым вектором y_k с точностью до перестановки компонент в том случае, если элементы j -ой строки и j -о столбца матрицы A совпадают с точностью до перестановки компонент с элементами k -ой строки и k -о столбца матрицы B .

Ситуация, когда перестановка, задающая искомый изоморфизм φ , может быть однозначно получена покомпонентным сравнением некоторых векторов x_j и y_k , соответствует ситуации, когда графы изоморфны и их группы автоморфизмов тривиальны, и перестановка, задающая изоморфизм, может быть получена покомпонентным сравнением столбцов. Для этого необходимо:

1. Найти решение x_j системы линейных уравнений $Ax = e_j$ для некоторого произвольного $j = \overline{1, n}$.
2. Решив n систем линейных уравнений $Bu = e_k$, $k = \overline{1, n}$, найти $k_j : \exists P$ – матрица перестановки: $Px_j = y_{k_j}$, $j = \overline{1, n}$.
3. Покомпонентно сравнить полученные решения систем x_j и y_{k_j} и определить перестановку P , задающую изоморфизм графов G_A и G_B .

Однако, если группа автоморфизмов графа G_A не тривиальна, то необходимо из множества перестановок, которые могут быть получены покомпонентным сравнением некоторых двух векторов x_j и y_k , выделить одну, задающую изоморфизм.

Кроме того, возможна ситуация, когда помимо перестановок, задающих изоморфизм, векторы-решения могут получаться друг из друга и с помощью перестановок, изоморфизма графов не задающих, что иллюстрируется ниже на примере работы алгоритма.

Если перестановку, задающую изоморфизм, нельзя однозначно установить покомпонентным сравнением векторов x_j и y_k , отличающихся друг от друга только перестановкой компонент, то тогда существуют и векторы x_l , $l \neq j$, отличающихся от x_j только перестановкой компонент.

Лемма 3. Пусть матрицы A и B , такие что существует P – матрица перестановки: $A = BPB^{-1}$. Матрица перестановки P – не единственная такая матрица тогда и только тогда, когда некоторые из столбцов матрицы A^{-1} отличаются друг от друга только перестановкой компонент.

Доказательство. В самом деле, если P и P' такие, что $A = PBP^{-1}$ и $A = P'BP'^{-1}$, и соответствующие им биекции – φ и φ' , то

$$\begin{aligned} A = PBP^{-1} &\Leftrightarrow A^{-1} = PB^{-1}P^{-1}, \\ A = P'BP'^{-1} &\Leftrightarrow A^{-1} = P'B^{-1}P'^{-1}, \end{aligned}$$

а значит, учитывая (4), получаем:

$$\begin{pmatrix} x_{11} & \dots & x_{j1} & \dots & x_{n1} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{1j} & \dots & x_{jj} & \dots & x_{nj} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{1n} & \dots & x_{jn} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} y_{\varphi(1)\varphi(1)} & \dots & y_{\varphi(1)\varphi(j)} & \dots & y_{\varphi(1)\varphi(n)} \\ \vdots & \ddots & \vdots & & \vdots \\ y_{\varphi(j)\varphi(1)} & \dots & y_{\varphi(j)\varphi(j)} & \dots & y_{\varphi(j)\varphi(n)} \\ \vdots & & \vdots & \ddots & \vdots \\ y_{\varphi(n)\varphi(1)} & \dots & y_{\varphi(n)\varphi(j)} & \dots & y_{\varphi(n)\varphi(n)} \end{pmatrix}, \quad (5)$$

и

$$\begin{pmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{j1} & \dots & x_{jj} & \dots & x_{jn} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nj} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} y_{\varphi'(1)\varphi'(1)} & \dots & y_{\varphi'(1)\varphi'(j)} & \dots & y_{\varphi'(1)\varphi'(n)} \\ \vdots & \ddots & \vdots & & \vdots \\ y_{\varphi'(j)\varphi'(1)} & \dots & y_{\varphi'(j)\varphi'(j)} & \dots & y_{\varphi'(j)\varphi'(n)} \\ \vdots & & \vdots & \ddots & \vdots \\ y_{\varphi'(n)\varphi'(1)} & \dots & y_{\varphi'(n)\varphi'(j)} & \dots & y_{\varphi'(n)\varphi'(n)} \end{pmatrix}. \quad (6)$$

$P' \neq P \Leftrightarrow \varphi \neq \varphi'$, следовательно, $\exists j : \varphi(j) \neq \varphi'(j)$, и значит из равенства (5) следует, что столбец x_j равен столбцу $y_{\varphi(j)}$ с точностью до перестановки компонент, а из равенства (6), что x_j равен и некоторому другому ($\varphi(j) \neq \varphi'(j)$) столбцу $y_{\varphi'(j)}$ с точностью до перестановки компонент.

При перестановке рядов B^{-1} , задаваемой P , столбец $y_{\varphi'(j)}$ переходит в столбец с номером k , $k \neq j$, то есть $y_{\varphi'(j)} = x_k$. Получаем:

- а) x_j с точностью до перестановки компонент равен столбцу $y_{\varphi(j)}$;
- б) $y_{\varphi(j)}$ с точностью до перестановки компонент равен столбцу $y_{\varphi'(j)}$;
- в) $y_{\varphi'(j)}$ с точностью до перестановки компонент равен столбцу x_k .

Значит, столбец x_j с точностью до перестановки компонент равен столбцу x_k .

Обратно, если в матрице A^{-1} нет столбцов, отличающихся друг от друга только перестановкой компонент, то если (5) выполняется при некотором φ , которому соответствует матрица перестановки P , то (6) не может быть выполнено при $\varphi' \neq \varphi$, а значит и не существует $P' \neq P : A = P'BP'^{-1}$. ■

Замечание 1. Очевидно, что если в условии леммы 3 в качестве матрицы B положить матрицу A , то отсутствие столбцов, равных с точностью до перестановки компонент, влечет тривиальность группы автоморфизмов $\Gamma(G_A)$.

Введем следующие обозначения: $R(A) \stackrel{dn}{=} \{x_j\}_{j=1}^n$, $R(B) \stackrel{dn}{=} \{y_k\}_{k=1}^n$. Множество $R(A)$ ($R(B)$) будем называть *простым*, если $\forall j \forall P \nexists l : x_j = Px_l$ ($\forall k \forall P \nexists i : y_k = Py_i$), где P – матрица перестановки.

Отметим, что в силу (1) равенство $x_j = Px_l$ влечет равенство $x_{jj} = x_{ll}$.

На каждой итерации алгоритм работает не с исходными матрицами A и B (1), но с матрицами уже возмущенными в ходе предыдущих итераций. Будем производить возмущения матриц, поставленных в соответствие графам (1), при помощи матриц E^j с единственным ненулевым элементом – элементом на диагонали в j -ой строке, равным единице:

$$E^j = \begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} \begin{matrix} - & 1 \\ \vdots \\ - & j \\ \vdots \\ - & n \end{matrix}$$

Положим $A^0 = A$, $B^0 = B$. Находя соответствие j -ой вершины графа G_A некоторой вершине графа G_B , мы рассматриваем одну возмущенную матрицу A^j и $n - j + 1$ матриц B_k^j , получаемых возмущением матрицы B^{j-1} :

$$A^j = A^{j-1} + \varepsilon E^j, \quad B_k^j = B^{j-1} + \varepsilon E^j.$$

То есть на каждой j -ой итерации в системы линейных уравнений (3) вместо A подставляем матрицу A^j , а вместо матрицы B – поочередно $n - j + 1$ матриц B_k^j , полагая, что соответствие между вершинами от 1-ой до $(j-1)$ -ой графа G_A и вершинами $j_1, \dots, j_{k-1}$ графа G_B уже установлено.

Увеличивая на ε_j j -й диагональный элемент матрицы A^{j-1} , использовавшейся на предыдущей итерации, и получая матрицу A^j , мы поочередно возмущаем все диагональные элементы матрицы B^{j-1} , отвечающие в графе G_B вершинам, которые пока не поставлены в соответствие вершинам графа G_A . Выделяя таким образом среди них одну, которая будет поставлена в соответствие вершине j , формируем матрицу B^j .

Пусть $j \leftrightarrow k_j$ – обозначает устанавливаемое в ходе работы алгоритма соответствие. Принципиальная схема алгоритма спектрального расщепления проверки изоморфизма графов следующая.

Алгоритм спектрального расщепления проверки изоморфизма графов.

Схема 1

Шаг 0. $A^0 := A$, $B^0 := B$; $j := 1$.

Шаг 1.1. Если $j \leq n$, то перейти на шаг 1.1, иначе перейти на шаг 6.

Шаг 1.2. Выбор ε_j .

Шаг 1.3. $A^j := A^{j-1} + \varepsilon_j E^j$.

Шаг 2. Решение системы линейных уравнений $A^j x = e_j$. x_j – полученное решение. $k := 1$.

Шаг 3.1. Если $k \leq n$, то перейти на шаг 3.2, иначе – перейти на Шаг 4.

Шаг 3.2. $B_k^j := B^{j-1} + \varepsilon_j E^k$.

Шаг 3.3. Решение системы линейных уравнений $B_k^j y = e_k$. y_k – полученное решение.

Шаг 3.4. $k := k + 1$. Перейти на шаг 3.1.

Шаг 4. Сравнение норм векторов x_j и y_k , где k такие, что $\forall i < j \nexists i : i \leftrightarrow k$.

Если $\forall k \|x_j\| \neq \|y_k\|$, то графы G_A и G_B неизоморфны. Работу алгоритма завершить.

Если $\exists k \exists P : x_j = Py_k$, и $\forall i \exists l : x_{ji} = y_{kl}$, и $x_{jj} = y_{kk}$, то $k_j := k$.

Шаг 5. $B^j := B^{j-1} + \varepsilon_j E^{k_j}$. $j := j + 1$. Перейти на шаг 1.1.

Шаг 6. Работу алгоритма завершить. Полученное соответствие $j \leftrightarrow k_j$ – найденный изоморфизм графов G_A и G_B .

В ходе работы алгоритма последовательно возмущаем матрицы, представляющие графы G_A и G_B – матрицы графов с петлями G'_A и G'_B , до тех пор пока группы автоморфизмов графов G'_A и G'_B не станут тривиальными. После того, как группы автоморфизмов графов G'_A и G'_B становятся тривиальны, возможно однозначное установление изоморфизма графов.

Рассмотрим последовательности матриц, с которыми работает алгоритм: $\{A^j\}_{j=1}^n$ и $\{B^j\}_{j=1}^n$.

Лемма 4. P – матрица перестановки такая, что $A = PBP^{-1}$, тогда и только тогда, когда на каждой j -ой итерации алгоритма выполняется равенство

$$A^j = PB^jP^{-1},$$

если $k_j = \varphi(j)$, где φ – биекция, соответствующая P .

Доказательство. Пусть φ – соответствующая P биекция. Тогда

$$k_j = \varphi(j) \Leftrightarrow E^j = PE^{k_j}P^{-1},$$

а так как $A = PBP^{-1}$, то

$$A^j = PB^jP^{-1} \Leftrightarrow E^j = PE^{k_j}P^{-1} \text{ и } A^{j-1} = PB^{j-1}P^{-1}. \quad (8)$$

Значит, при $j = 1$ выполнение равенства $A^j = PB^jP^{-1}$ следует из равенства $A = PBP^{-1}$. Из (8) же следует, что оно имеет место и для $j = \overline{2, n}$. ■

По лемме 4 $G_A \cong G_B$ тогда и только тогда, когда $\exists P : \forall j = \overline{1, n} A^j = PB^jP^{-1}$, при выборе $k_j = \varphi(j)$ на каждой j -ой итерации, и, в частности, $A^n = PB^nP^{-1}$. То есть если $G_A \cong G_B$, то в ходе работы алгоритма при возмущении исходных матриц A^0 и B^0 , поставленных в соответствие графам G_A и G_B , на каждой j -ой итерации сохраняется следующее ключевое соотношение:

если $A^j = A^{j-1} + \varepsilon_j E^j$, то

$$A^j = PB^jP^{-1} \Leftrightarrow B^j = B^{j-1} + \varepsilon_j E^{\varphi(j)}.$$

Элементы обратной матрицы изменяются непрерывно относительно изменения элементов матрицы. После возмущения матрицы A прибавлением к ней матрицы $\varepsilon_j E^j$ и возмущения матрицы B прибавлением к ней матрицы $\varepsilon_j E^k$:

$$|A'| = |A| + \varepsilon_j A_{jj}, \quad |B'| = |B| + \varepsilon_j B_{kk},$$

и

$$\forall p \neq j : A'_{pp} = A_{pp} + \varepsilon_j A_{pp,jj}, \quad (9)$$

$$\forall q \neq k : B'_{qq} = B_{qq} + \varepsilon_j B_{qq,kk}, \quad (10)$$

где $A_{pp,jj}$, $B_{qq,kk}$ — обозначают миноры второго порядка, получаемые из миноров A_{pp} , B_{qq} последующим удалением из них рядов с номерами j и k соответственно. Все указанные миноры — определители положительно определенных подматриц матриц A и B , то есть они строго положительны. Следовательно, за n итераций алгоритма при надлежащем подборе ε_j мы можем получить матрицы A^n и B^n такие, что $A^n = PB^n P^{-1}$, и множества $R(A^n)$ и $R(B^n)$ — простые. Тогда матрица P , с учетом леммы 4, может быть получена однозначно проверкой на равенство с точностью до перестановки компонент векторов из множеств $R(A^n)$ и $R(B^n)$. И так как для P в ходе работы алгоритма выполняется $A^j = PB^j P^{-1}$, то за n итераций алгоритма может быть установлено соответствие $j \leftrightarrow k_j$, являющееся биекцией, задающей изоморфизм графов G_A и G_B .

При этом для проверки на каждой j -ой итерации сохранения возможности получения матрицы $(B^j)^{-1}$ из матрицы $(A^j)^{-1}$ при помощи некоторой перестановки ее рядов достаточно проверить равенство с точностью до перестановки компонент векторов x_j и y_k при некотором k . Это следует из леммы 5.

Лемма 5. Пусть для всех значений j и k определены матрицы A' и B' :

$$A' = A + \varepsilon_0 E^j, \quad B' = B + \varepsilon_0 E^k, \quad \varepsilon_0 > 0,$$

x_j , y_k — решения систем линейных уравнений $A'x = e_j$, $B'y = e_k$, такие что

$$x_j = Py_k \text{ и } x_{jj} = y_{kk}.$$

Тогда $\forall j \exists \varepsilon_0 > 0$:

$$A^{-1} = PB^{-1}P^{-1}$$

$$\Leftrightarrow$$

$$\exists! k : x_j = Py_k \text{ и } x_{jj} = y_{kk}, \text{ и } \forall j' \neq j \exists k' : x_{j'} = Py_{k'} \text{ и } x_{j'j'} = y_{k'k'}.$$

Доказательство.

Необходимость. Пусть x_i^0 и y_l^0 — решения систем линейных уравнений

$$Ax = e_i, \quad i = \overline{1, n},$$

$$By = e_l, \quad l = \overline{1, n},$$

с невозмущенными матрицами A и B . $x_i^0 = Py_l^0$.

$$x_i^0 = \left(\frac{A_{1i}}{|A|}, \dots, \frac{A_{ni}}{|A|} \right), \quad y_l^0 = \left(\frac{B_{1l}}{|B|}, \dots, \frac{B_{nl}}{|B|} \right).$$

При производимых нами возмущениях

$$A'_{jt} = A_{jt}, \quad B'_{kt} = B_{kt}, \quad t = \overline{1, n},$$

Следовательно, если $x_j^0 = Py_k^0$, то и $x_j = Py_k$, поскольку

$$x_j = \frac{|A|}{|A'|} \left(\frac{A_{j1}}{|A|}, \dots, \frac{A_{jn}}{|A|} \right) = \frac{|A|}{|A'|} x_j^0,$$

$$y_k = \frac{|B|}{|B'|} \left(\frac{B_{k1}}{|B|}, \dots, \frac{B_{kn}}{|B|} \right) = \frac{|B|}{|B'|} y_k^0.$$

Из (9) и (10) следует, что мы можем подобрать такое значение $\varepsilon_0 > 0$, что

$$\forall p \neq j : A'_{pp} \neq A'_{jj} = A_{jj},$$

и

$$\forall q \neq l : B'_{qq} \neq B'_{kk} = B_{kk},$$

при том, что $A_{jj} = B_{kk}$. Для этого ε_0 должно быть таким, что

$$\forall p \quad \varepsilon_0 \neq \frac{A_{jj} - A_{pp}}{A_{pp,jj}} \left(\forall q \quad \varepsilon_0 \neq \frac{B_{kk} - B_{qq}}{B_{qq,kk}} \right),$$

($A_{pp,jj} > 0$, $B_{qq,kk} > 0$). После возмущения матриц A и B для любого заданного j при выбранном ε_0 такое k , что $x_j = Py_k$ и $x_{jj} = y_{kk}$, будет единственным.

Выполнение условия $\forall j' \neq j \quad \exists k' : x_{j'} = Py_{k'}$ и $x_{j'j'} = y_{k'k'}$ следует из равенства $A^{-1} = PB^{-1}P^{-1}$.

Достаточность. Имеем:

$$A' = A + \varepsilon_0 E^j, \quad B' = B + \varepsilon_0 E^k, \quad \varepsilon_0 > 0;$$

$$x_j : A'x_j = e_j, \quad y_k : B'y_k = e_k;$$

и

$$y_k = Px_j, \quad x_{jj} = y_{kk},$$

и $\forall j \quad \exists k : x_j = Py_k$ и $x_{jj} = y_{kk}$.

Тогда $A'Py_k = e_j$. Так как $y_k = (B')^{-1}e_k$, то $A'P(B')^{-1}e_k = e_j$. Из $e_j = Pe_k$ следует $A'P(B')^{-1}e_k = Pe_k$, то есть $(A'P(B')^{-1} - P)e_k = 0$ для любого $\varepsilon_0 > 0$. При вводимых нами возмущениях матриц A и B изменения элементов матриц и элементов обратной матриц имеют непрерывную зависимость от ε_0 – (9), (10). Поскольку $\forall \varepsilon_0 > 0 \quad \forall j = \overline{1, n} :$

$$((A + \varepsilon_0 E^j)P(B + \varepsilon_0 E^k)^{-1} - P)e_k = 0,$$

и

$$APB^{-1} - P = \lim_{\varepsilon_0 \rightarrow 0} ((A + \varepsilon_0 E^j)P(A + \varepsilon_0 E^k)^{-1} - P),$$

то и линейный оператор $APB^{-1} - P$ будет занулять любой базисный вектор e_k в R^n , так как j – произвольное, и поскольку из равномерной сходимости линейных операторов в R^n следует их точечная сходимость. Значит, $APB^{-1} - P$ – нулевой оператор. То есть

$$A = PBP^{-1}.$$

■

2. Трудоемкость принципиальной схемы алгоритма спектрального расщепления проверки изоморфизма графов

Одна итерация алгоритма спектрального расщепления состоит из следующих этапов:

1 этап.

Решение системы линейных уравнений $A^j x = e_j$.

2 этап.

1. Решение в худшем случае n систем линейных уравнений $B_k^j y = e_k$, $k = \overline{1, n}$.
2. Сравнение компонент полученных решений: вектора x_j и (в худшем случае) n векторов $\{y_k\}_{k=1}^n$.

Трудоемкость процедуры решения систем линейных уравнений с заданной точностью зависит от метода, применяемого при решении систем линейных уравнений. Пусть $O(N_S)$ – количество элементарных машинных операций, которые необходимо совершить для решения систем линейных алгебраических уравнений с заданной точностью.

Процедура сравнения полученных решений может потребовать максимум $O(n \log n)$ элементарных операций.

Таким образом, общая трудоемкость одной итерации алгоритма составляет

$$O(N_S) + n(O(N_S) + O(n \log n)) = O(n(N_S + \log n)).$$

Поэтому общая трудоемкость алгоритма по всем итерациям составляет

$$n \cdot O(n(N_S + \log n)) = O(n^2(N_S + \log n)). \quad (11)$$

В худшем случае для установления взаимнооднозначного соответствия между вершинами поданных на вход алгоритма графов (представляющих их матриц) необходимо совершить n итераций.

Ниже приводится схема алгоритма, трудоемкость которой может быть значительно ниже схемы приведенной выше. В случае, если графы изоморфны, ситуация, когда матрица P может быть получена однозначно проверкой на равенство с точностью до перестановки компонент векторов из множеств $R(A^{n_0})$ и $R(B^{n_0})$, может возникнуть и ранее n -ой итерации. То есть множества $R(A^{n_0})$

и $R(B^{n_0})$ могут оказаться простыми к некоторой итерации n_0 , когда $n_0 < n$. Добавление шагов 6.1 - 6.4. в приведенной ниже схеме алгоритма позволяет во многих случаях существенно снизить число итераций алгоритма. Так, при установлении изоморфизма регулярных графов со степенью вершин равной 4, представляющих собой решетку на торе, множества $R(A^{n_0})$ и $R(B^{n_0})$ становятся простыми при n_0 не превышающем $\sqrt{n}$.

Отметим также, что задача установления изоморфизма регулярных графов представляет собой одну из тяжелейших задач при проверке изоморфизма графов для наиболее эффективных алгоритмов. Так, в частности, трудоемкость алгоритма NAUTY [8, с. 45–87], считающегося одним из наиболее эффективных алгоритмов решения задачи проверки изоморфизма графов, становится экспоненциальной уже при работе с регулярными графами, со степенью вершин равной 4 [9, с. 1172–1177].

Алгоритм спектрального расщепления проверки изоморфизма графов.

Схема 2

Шаг 0. $A^0 := A$, $B^0 := B$; $j := 1$.

Шаг 1.1. Если $j \leq n$, то перейти на шаг 1.1, иначе перейти на шаг 7.

Шаг 1.2. Выбор ε_j .

Шаг 1.3. $A^j := A^{j-1} + \varepsilon_j E^j$.

Шаг 2. Решение системы линейных уравнений $A^j x = e_j$. x_j – полученное решение. $k := 1$.

Шаг 3.1. Если $k \leq n$, то перейти на шаг 3.2, иначе – графы неизоморфны. Работу алгоритма завершить.

Шаг 3.2. $B_k^j := B^{j-1} + \varepsilon_j E^k$.

Шаг 3.3. Решение системы линейных уравнений $B_k^j y = e_k$. y_k – полученное решение.

Шаг 3.4. $k := k + 1$. Перейти на шаг 3.1.

Шаг 4. Сравнение норм векторов x_j и y_k , где k такие, что $\forall i < j \nexists i : i \leftrightarrow k$ (то есть если вершине k графа G_B не поставлена в соответствие никакая вершина графа G_A).

Если $\forall k \|x_j\| \neq \|y_k\|$, то графы G_A и G_B неизоморфны. Работу алгоритма завершить.

Если $\exists k \|x_j\| = \|y_k\|$, и $\forall i \exists l : x_{ji} = y_{kl}$, и $x_{jj} = y_{kk}$, то $k_j := k$.

Шаг 5. $B^j := B^{j-1} + \varepsilon_j E^{k_j}$. $j := j + 1$. Перейти на шаг 6.1.

Шаг 6.1. Решение систем линейных уравнений $A^j x = e_p$, где p такое, что $\nexists s : p \leftrightarrow s, 1 \leq s \leq n$. x_p – полученные решения. (То есть рассматриваются вершины p графа G_A , которым не поставлены в соответствие никакие вершины графа G_B .)

Шаг 6.2. Если $\forall p_1 \forall p_2 \|x_{p_1}\| \neq \|x_{p_2}\|$, то перейти на шаг 6.3, иначе перейти на шаг 1.1.

Шаг 6.3. Решение систем линейных уравнений $B^j y = e_q$, где q такое, что $\nexists t : t \leftrightarrow q, 1 \leq t \leq n$. y_q – полученные решения.

Шаг 6.4. Сравнение норм полученных векторов-решений. Если p_i и q_i такие, что

$\exists P : x_{p_i} = Py_{q_i}$ и $x_{p_i p_i} = y_{q_i q_i}$, то $\varphi(p_i) := q_i$.

Шаг 7. Работу алгоритма завершить. Полученное соответствие $j \leftrightarrow k_j$ – найденный изоморфизм графов G_A и G_B .

3. Пример, иллюстрирующий работу алгоритма спектрального расщепления проверки изоморфизма графов.

Рассмотрим пример, иллюстрирующий работу алгоритма. Пусть даны графы G_A и G_B (рис. 1). Модифицированные матрицы смежности вида (1) этих графов:

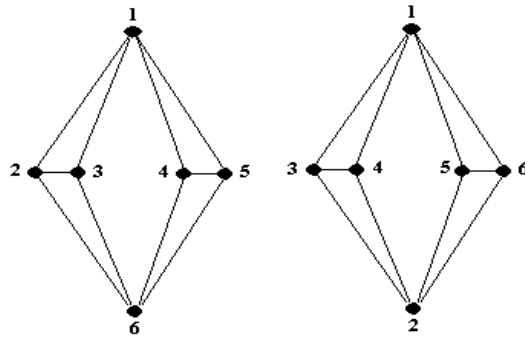
$$A = \begin{pmatrix} 8 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}, \quad B = \begin{pmatrix} 8 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}.$$

Пусть длина мантииссы машинных чисел на протяжении всей работы алгоритма равна n ($n = 6$). Такой длины мантииссы будет достаточно для эффективной работы алгоритма.

$$\text{Тогда } (A^0)^{-1} = \begin{pmatrix} 8 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,134 & -0,018 & -0,018 & -0,018 & -0,018 & 0,008929 \\ -0,018 & 0,15 & -0,016 & 0,004464 & 0,004464 & -0,018 \\ -0,018 & -0,016 & 0,15 & 0,004464 & 0,004464 & -0,018 \\ -0,018 & 0,004464 & 0,004464 & 0,15 & -0,016 & -0,018 \\ -0,018 & 0,004464 & 0,004464 & -0,016 & 0,15 & -0,018 \\ 0,008929 & -0,018 & -0,018 & -0,018 & -0,018 & 0,134 \end{pmatrix},$$

и

Рис. 1. Графы G_A и G_B

$$(B^0)^{-1} = \begin{pmatrix} 8 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,134 & 0,008929 & -0,018 & -0,018 & -0,018 & -0,018 \\ 0,008929 & 0,134 & -0,018 & -0,018 & -0,018 & -0,018 \\ -0,018 & -0,018 & 0,15 & -0,016 & 0,004464 & 0,004464 \\ -0,018 & -0,018 & -0,016 & 0,15 & 0,004464 & 0,004464 \\ -0,018 & -0,018 & 0,004464 & 0,004464 & 0,15 & -0,016 \\ -0,018 & -0,018 & 0,004464 & 0,004464 & -0,018 & 0,15 \end{pmatrix}.$$

Возможными решениями в данном случае являются матрицы перестановки, соответствующие следующим подстановкам:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 5 & 6 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 6 & 5 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 3 & 5 & 6 & 2 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 3 & 6 & 5 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 6 & 3 & 4 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 6 & 4 & 3 & 2 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 6 & 5 & 3 & 4 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 6 & 5 & 4 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 6 & 5 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 5 & 6 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 3 & 6 & 5 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 5 & 6 & 3 & 4 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 5 & 6 & 4 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 3 & 4 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 5 & 4 & 3 & 1 \end{pmatrix}.$$

Также укажем перестановку

$$P' = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 5 & 6 & 3 & 4 & 2 \end{pmatrix}$$

как задающую перестановку компонент первого вектор-столбца матрицы B^{-1} , дающую первый вектор-столбец матрицы A^{-1} , однако не задающую изоморфизма графов G_A и G_B .

До 5-ой итерации схема 1 и схема 2 алгоритма для данной задачи будут вести себя одинаково. А именно, при вводимых нами на каждой итерации возмущениях матриц множества $R(A^j)$ и $R(B^j)$ не будут простыми при $j = \overline{1, 4}$, и выполнение шагов 6.1 - 6.4 не будут приводить схему 2 к отходу от последовательных возмущений матриц A^j и B^j .

Итерация 1. Пусть на итерации 1 $\varepsilon_1 = 0, 1$, тогда

$$(A^1)^{-1} = \begin{pmatrix} 8 + \varepsilon_1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,132 & -0,018 & -0,018 & -0,018 & -0,018 & 0,008811 \\ -0,018 & 0,15 & -0,016 & 0,004433 & 0,004433 & -0,018 \\ -0,018 & -0,016 & 0,15 & 0,004433 & 0,004433 & -0,018 \\ -0,018 & 0,004433 & 0,004433 & 0,15 & -0,016 & -0,018 \\ -0,018 & 0,004433 & 0,004433 & -0,016 & 0,15 & -0,018 \\ 0,008811 & -0,018 & -0,018 & -0,018 & -0,018 & 0,134 \end{pmatrix},$$

и при $k_1 = 1$, получаем

$$(B^1)^{-1} = \begin{pmatrix} 8 + \varepsilon_1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,132 & 0,008811 & -0,018 & -0,018 & -0,018 & -0,018 \\ 0,008811 & 0,15 & -0,018 & -0,018 & -0,018 & -0,018 \\ -0,018 & -0,018 & 0,15 & -0,016 & 0,004433 & 0,004433 \\ -0,018 & -0,018 & -0,016 & 0,15 & 0,004433 & 0,004433 \\ -0,018 & -0,018 & 0,004433 & 0,004433 & 0,15 & -0,016 \\ -0,018 & -0,018 & 0,004433 & 0,004433 & -0,016 & 0,15 \end{pmatrix}.$$

То есть соответствие $j \leftrightarrow k_j$, $j = \overline{1, 6}$, которое строится в ходе работы алгоритма, на итерации 1 имеет следующий вид:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & . & . & . & . & . \end{pmatrix},$$

точками обозначены соответствия, которые еще будут установлены на последующих итерациях.

Итерация 2

Положив $\varepsilon_2 = 0,2$

$$(A^2)^{-1} = \begin{pmatrix} 8 + \varepsilon_1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 + \varepsilon_2 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,134 & -0,017 & -0,018 & -0,018 & -0,018 & 0,00875 \\ -0,017 & 0,146 & -0,016 & 0,004303 & 0,004303 & -0,017 \\ -0,018 & -0,016 & 0,15 & 0,004447 & 0,004447 & -0,018 \\ -0,018 & 0,004303 & 0,004447 & 0,15 & -0,016 & -0,018 \\ -0,018 & 0,004303 & 0,004447 & -0,016 & 0,15 & -0,018 \\ 0,00875 & -0,017 & -0,018 & -0,018 & -0,018 & 0,134 \end{pmatrix},$$

$$(B^2)^{-1} = \begin{pmatrix} 8 + \varepsilon_1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 + \varepsilon_2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}^{-1} =$$

$$= \begin{pmatrix} 0,132 & 0,00875 & -0,017 & -0,018 & -0,018 & -0,018 \\ 0,00875 & 0,134 & -0,017 & -0,018 & -0,018 & -0,018 \\ -0,017 & -0,017 & 0,146 & -0,016 & 0,004303 & 0,004303 \\ -0,018 & -0,018 & -0,016 & 0,15 & 0,004447 & 0,004447 \\ -0,018 & -0,018 & 0,004303 & 0,004447 & 0,15 & -0,016 \\ -0,018 & -0,018 & 0,004303 & 0,004447 & -0,016 & 0,15 \end{pmatrix}.$$

$j \leftrightarrow k_j$, $j = \overline{1,6}$ на итерации 2 имеет вид:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & . & . & . & . \end{pmatrix}.$$

Итерация 3

$\varepsilon_3 = 0,3$

$$\begin{aligned}
(A^3)^{-1} &= \begin{pmatrix} 8 + \varepsilon_1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 + \varepsilon_2 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 + \varepsilon_3 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}^{-1} = \\
&= \begin{pmatrix} 0,132 & -0,017 & -0,017 & -0,018 & -0,018 & 0,008659 \\ -0,017 & 0,146 & -0,015 & 0,004324 & 0,004324 & -0,017 \\ -0,017 & -0,015 & 0,144 & 0,004255 & 0,004255 & -0,017 \\ -0,018 & 0,004324 & 0,004255 & 0,15 & -0,016 & -0,018 \\ -0,018 & 0,004324 & 0,004255 & -0,016 & 0,15 & -0,018 \\ 0,008659 & -0,017 & -0,017 & -0,018 & -0,018 & 0,134 \end{pmatrix}, \\
(B^3)^{-1} &= \begin{pmatrix} 8 + \varepsilon_1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 + \varepsilon_2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 + \varepsilon_3 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}^{-1} = \\
&= \begin{pmatrix} 0,132 & 0,008659 & -0,017 & -0,017 & -0,018 & -0,018 \\ 0,008659 & 0,134 & -0,017 & -0,017 & -0,018 & -0,018 \\ -0,017 & -0,017 & 0,146 & -0,015 & 0,004324 & 0,004324 \\ -0,017 & -0,017 & -0,015 & 0,144 & 0,004255 & 0,004255 \\ -0,018 & -0,018 & 0,004324 & 0,004255 & 0,15 & -0,016 \\ -0,018 & -0,018 & 0,004324 & 0,004255 & -0,016 & 0,15 \end{pmatrix}.
\end{aligned}$$

$j \leftrightarrow k_j$, $j = \overline{1, 6}$ на итерации 3:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & . & . & . \end{pmatrix}.$$

Итерация 4

$\varepsilon_4 = 0,4$

$$\begin{aligned}
(A^4)^{-1} &= \begin{pmatrix} 8 + \varepsilon_1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 7 + \varepsilon_2 & 1 & 0 & 0 & 1 \\ 1 & 1 & 7 + \varepsilon_3 & 0 & 0 & 1 \\ 1 & 0 & 0 & 7 + \varepsilon_4 & 1 & 1 \\ 1 & 0 & 0 & 1 & 7 & 1 \\ 0 & 1 & 1 & 1 & 1 & 8 \end{pmatrix}^{-1} = \\
&= \begin{pmatrix} 0,132 & -0,017 & -0,017 & -0,017 & -0,018 & 0,008929 \\ -0,017 & 0,146 & -0,015 & 0,004079 & 0,004351 & -0,017 \\ -0,017 & -0,015 & 0,144 & 0,004014 & 0,004282 & -0,017 \\ -0,017 & 0,004079 & 0,004014 & 0,142 & -0,015 & -0,017 \\ -0,018 & 0,004351 & 0,004282 & -0,015 & 0,15 & -0,018 \\ 0,008541 & -0,017 & -0,017 & -0,017 & -0,018 & 0,134 \end{pmatrix},
\end{aligned}$$

$$(B^4)^{-1} = \begin{pmatrix} 8 + \varepsilon_1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 7 + \varepsilon_2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 7 + \varepsilon_3 & 0 & 0 \\ 1 & 1 & 0 & 0 & 7 + \varepsilon_4 & 1 \\ 1 & 1 & 0 & 0 & 1 & 7 \end{pmatrix}^{-1} =$$

$$\begin{pmatrix} 0,132 & 0,008541 & -0,017 & -0,017 & -0,017 & -0,018 \\ 0,008541 & 0,134 & -0,017 & -0,017 & -0,017 & -0,018 \\ -0,017 & -0,017 & 0,146 & -0,015 & 0,004079 & 0,004351 \\ -0,017 & -0,017 & -0,015 & 0,144 & 0,004014 & 0,004282 \\ -0,017 & -0,017 & 0,004079 & 0,004014 & 0,142 & -0,015 \\ -0,018 & -0,018 & 0,004351 & 0,004282 & -0,015 & 0,15 \end{pmatrix}.$$

$j \leftrightarrow k_j, j = \overline{1,6}$ на итерации 4:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 5 & . & . \end{pmatrix}.$$

К пятой итерации и схемы 1 и схемы 2 алгоритма мы получаем простые множества $R(A^4)$ и $R(B^4)$, а значит дальнейших возмущений матриц A^4 и B^4 можно не производить, и на шагах 6.1 - 6.4 схемы 2 алгоритма будет установлено то же соответствие по всем оставшимся j ($j = \overline{5,6}$), что и на итерациях 5 и 6 схемы 1 алгоритма. В результате, получаем следующее соответствие вершин – изоморфизм:

$j \leftrightarrow k_j, j = \overline{1,6}$:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 5 & 6 & 2 \end{pmatrix}.$$

4. Вычислительная эффективность алгоритма спектрального расщепления

Локализация групп решений

Введем следующие обозначения.

$$R_i(A) \stackrel{dn}{=} \{x_{i_1}, \dots, x_{i_k} : \forall p, q : 1 \leq p, q \leq k \exists P : x_{i_p} = Px_{i_q}, x_{i_p i_p} = x_{i_q i_q}\},$$

где индекс i присваивается этим множествам в соответствии с их упорядочением по невозрастанию норм векторов в них входящих. При этом $R(A) = \cup R_i(A)$, где объединение берется по всем таким группам векторов-столбцов обратной матрицы, в каждой из которых только те векторы, которые могут быть получены друг из друга при помощи некоторой перестановке их компонент. Пусть количество таких групп m . Тогда

$$R(A) = \bigcup_{i=1}^m R_i(A),$$

и

$$i \neq j \Rightarrow R_i \cap R_j = \emptyset.$$

Очевидно, что множество $R(A)$ является простым тогда и только тогда, когда $\forall i |R_i(A)| = 1$, и, соответственно, $m = n$. Рассмотрим эти множества в процессе возмущения матрицы A .

Если на j -ой итерации алгоритма после возмущения матрицы A^{j-1} будет происходить расщепление некоторого множества $R_i(A^{j-1})$, выделением из него хотя бы одного вектора-столбца x_j , то есть $|R_i(A^{j-1})| \geq |R_i(A^j)| + 1$, и при этом будет выполняться также $|R_l(A^{j-1})| \geq |R_l(A^j)|$ и для $l \neq i$, то за n_0 возмущений диагональных элементов матриц A и B ($n_0 \leq n$) мы придем к тому, что множества $R(A)$ и $R(B)$ будут простыми множествами $R(A^{n_0})$ и $R(B^{n_0})$, и, согласно леммам 3, 4, 5, будет возможно однозначное установление соответствия между вершинами графов G_A и G_B , являющегося изоморфизмом.

Пусть

$$\Delta_{kl}^j \stackrel{dn}{=} |||x_{k_1}|| - ||x_{l_1}||, \quad x_{k_1} \in R_k(A^j), \quad x_{l_1} \in R_l(A^j),$$

и

$$\Delta^j \stackrel{dn}{=} \min_{1 \leq i, j \leq m} \Delta_{ij}^j.$$

Если нормы векторов x_j отличны, то они принадлежат разным группам $R_i(A)$, поэтому расщепление этих множеств на приведенном примере работы алгоритма спектрального расщепления может быть проиллюстрировано следующим образом. (Везде ниже верхние индексы у Δ опущены.) Так перед итерацией 1 алгоритма:

$$R_1(A^0) = \{x_1, x_6\}, \quad R_2(A^0) = \{x_2, x_3, x_4, x_5\},$$

$$\Delta_{12} = 0,144,$$

$$\Delta = 0,144.$$

После 1-ой итерации:

$$R_1(A^1) = \{x_1\}, \quad R_2(A^1) = \{x_2, x_3, x_4, x_5\},$$

$$R_3(A^1) = \{x_6\},$$

$$\Delta_{12} = 0,144, \quad \Delta_{13} = 0,002, \quad \Delta_{23} = 0,142,$$

$$\Delta = 0,002.$$

После 2-ой итерации:

$$R_1(A^2) = \{x_1\}, \quad R_2(A^2) = \{x_2\}, \quad R_3(A^2) = \{x_3\},$$

$$R_4(A^2) = \{x_4, x_5\}, \quad R_5(A^2) = \{x_6\},$$

$$\Delta_{12} = 0,016, \quad \Delta_{13} = 0,0649, \quad \Delta_{14} = 0,016, \quad \Delta_{15} = 0,002,$$

$$\Delta_{23} = 0,0649, \quad \Delta_{24} = 0, \quad \Delta_{25} = 0,014,$$

$$\Delta_{34} = 0,0809, \Delta_{35} = 0,0669,$$

$$\Delta_{45} = 0,014,$$

$$\Delta = 0.002.$$

Отметим, что хотя $\Delta_{24} = 0$, но $x_2 \neq Px_4 \forall P$, и $x_2 \neq Px_5 \forall P$. После 3-ей итерации:

$$R_1(A^3) = \{x_1\}, R_2(A^3) = \{x_2\}, R_3(A^2) = \{x_3\},$$

$$R_4(A^2) = \{x_4, x_5\}, R_5(A^2) = \{x_6\},$$

$$\Delta_{12} = 0,012, \Delta_{13} = 0,016, \Delta_{14} = 0,016, \Delta_{15} = 0,002,$$

$$\Delta_{23} = 0,004, \Delta_{24} = 0,004, \Delta_{25} = 0,01,$$

$$\Delta_{34} = 0, \Delta_{35} = 0,014,$$

$$\Delta_{45} = 0,014,$$

$$\Delta = 0,002.$$

$\Delta_{34} = 0$, но $x_3 \neq Px_4 \forall P$, и $x_3 \neq Px_5 \forall P$.

После 4-ой итерации:

$$R_1(A^3) = \{x_1\}, R_2(A^3) = \{x_2\}, R_3(A^2) = \{x_3\},$$

$$R_4(A^2) = \{x_4\}, R_5(A^4) = \{x_5\}, R_6(A^2) = \{x_6\}.$$

$$\Delta_{12} = 0,012, \Delta_{13} = 0,01, \Delta_{14} = 0,008, \Delta_{15} = 0,016, \Delta_{16} = 0,002,$$

$$\Delta_{23} = 0,002, \Delta_{24} = 0,004, \Delta_{25} = 0,004, \Delta_{26} = 0,01,$$

$$\Delta_{34} = 0,002, \Delta_{35} = 0,006, \Delta_{36} = 0,008,$$

$$\Delta_{45} = 0,008, \Delta_{46} = 0,006,$$

$$\Delta_{56} = 0.014,$$

$$\Delta = 0,002.$$

Числом обусловленности квадратной матрицы называется величина

$$\mu(A) = \sup_{\substack{x \neq 0 \\ \xi \neq 0}} \left\{ \frac{\|Ax\| \|\xi\|}{\|A\xi\| \|x\|} \right\} \quad (9).$$

Пусть $\lambda_{\max}$ – максимальное собственное значение из $Sp(A)$, $\lambda_{\min}$ – минимальное собственное значение из $Sp(A)$. При $\lambda_{\min} \neq 0$

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

При решении системы линейных уравнений $Ax = f$ в связи с погрешностями вычислений мы в действительности решаем систему уравнений $A(x + \xi) = f + g$,

где ξ – возмущение решения, возникающее от возмущения g правой части, так что $A\xi = g$. Из (9) следует

$$\frac{\|\xi\|}{\|x\|} \leq \mu(A) \frac{\|A\xi\|}{\|Ax\|} = \mu(A) \frac{\|g\|}{\|f\|}. \quad (10)$$

Содержательно величина $\mu(A)$ характеризует относительную погрешность решения $\|\xi\|/\|x\|$ через относительную погрешность правой части $\|f\|/\|g\|$. $\mu(A)$ – это наименьшая константа, для которой справедливо (10).

Рассмотрим систему $(A + C)y = f + g$, полученную из системы $Ax = f$ возмущением g правой части и возмущением C матрицы A . Пусть

$$\frac{\|C\|}{\|A\|} \leq \theta, \quad \frac{\|g\|}{\|f\|} \leq \psi.$$

Имеет место [10, с. 39] следующая теорема:

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

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

■

Для числа обусловленности симметрической матрицы справедлива следующая оценка [10, с. 59]:

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

где

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

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

При заданной нами структуре матрицы: $a_{kk} = d + d_k$, где d_k – степень k -ой вершины графа, а d – максимальная степень вершин графа: $d = \max_{1 \leq j \leq n} d_j$. Пусть i_1 – номер строки, на которой достигается $\eta(A)$, а i_2 – номер строки на которой достигается минимум $\chi(A)$. Суммирование модулей элементов i -ой строки для матриц, с которыми работает алгоритм, дает степень i -ой вершины. Значит

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

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

И, следовательно,

$$\mu(A) \leq \frac{\eta(A)}{\chi(A)} = \frac{3d}{d} = 3.$$

Таким образом, для решений систем линейных уравнений, получаемых в ходе итераций алгоритма, приведенная выше теорема с учетом того, что возмущения правых частей решаемых систем линейных уравнений не происходит ($\psi = 0$), приобретает следующий вид.

Теорема 1. Если $\theta < 1/3$, то

$$\frac{\|y - x\|}{\|x\|} \leq \theta \frac{3}{1 - 3\theta}.$$

Или же

$$|\|y\| - \|x\|| \leq \theta \frac{3}{1 - 3\theta} \|x\|. \quad (11)$$

■

При реализации алгоритма необходимо, чтобы возмущения были достаточно малы, и выделяемые из одной группы $R_i(A)$ векторы гарантированно не попадали в другие группы. Это возможно, если группы $R_i(A)$ и $R_i(B)$ решений систем (2) достаточно удалены друг от друга в смысле удаленности друг от друга норм решений, входящих в группы.

Пусть x_i, x_k — соответственно решения систем уравнений $A^{j-1}x = e_i$ и $A^{j-1}x = e_k$ и $x_i, x_k \in R_l(A^{j-1})$ для некоторого l , а x'_i, x'_k — решения систем уравнений $A^j x = e_i$ и $A^j x = e_k$. A_i, A_j — векторы, компоненты которых являются соответственно элементами i -ой и j -ой строки матрицы A^{j-1} , A'_i, A'_j — векторы, компоненты которых являются соответственно элементами i -ой и j -ой строки матрицы A^j .

По теореме 1

$$|\|x'_i\| - \|x_i\|| \leq \theta(\varepsilon_j) \frac{3}{1 - 3\theta(\varepsilon_j)} \|x'_i\|,$$

где

$$\frac{\|C\|}{\|A\|} = \frac{\|\varepsilon_j E^j\|}{\|A\|} = \frac{|\varepsilon_j|}{\|A\|} \leq \theta(\varepsilon_j).$$

В [11, с. 382] показано, что

$$\|x'_i\| < \frac{\sqrt{10}}{3d},$$

то есть

$$|\|x'_i\| - \|x_i\|| \leq \theta(\varepsilon_j) \frac{3}{1 - 3\theta(\varepsilon_j)} \cdot \frac{\sqrt{10}}{3d},$$

Учитывая малость $\theta(\varepsilon_j)$, это значит, что при достаточно малом возмущении матриц A^j происходит и малое возмущение векторов x_i , $i = \overline{1, n}$, что не нарушает локализованности групп векторов $R_i(A)$. То есть если, например, Δ^{j-1} составляло бы $1/n$, а Δ_{ik}^{j-1} при возмущениях матриц A^j изменялись бы не более чем на $1/n^2$, то после j -ой итерации не произошло бы попадания вектора x_j , выделяемого на j -ой итерации алгоритма из некоторой группы R_i , в какую-либо другую группу. И таким образом, не более чем за n итераций алгоритма все

группы R_i , состоящие более чем из одного вектора, могли бы быть расщеплены и при этом никакой выделяемый вектор гарантированно не попал бы из одной группы в другую, с числом векторов ней отличным от единицы.

5. Расщепление групп решений систем линейных уравнений

Следствием ограниченности памяти машины является то обстоятельство, что в ней не могут быть представлены числа, мантиссы которых содержат бесконечную последовательность γ -х ненулевых цифр в своем γ -м представлении. Под каждое число с плавающей точкой отводится место (ячейка) в памяти ЭВМ, достаточное для размещения k γ -х цифр мантиссы. В той же ячейке памяти размещается знак числа z , порядок $p_\gamma(z)$. Это позволяет хранить в памяти машины числа, порядки которых удовлетворяют условию $p^- \leq p_\gamma(z) \leq p^+$, где $p^- < 0$ и $p^+ > 0$ некоторые числа не зависящие от z .

То есть *машинные числа* – это вещественные числа вида

$$z = \pm \gamma^p \left(\frac{z_1}{\gamma} + \frac{z_2}{\gamma^2} + \dots + \frac{z_k}{\gamma^k} \right),$$

где $p^- \leq p \leq p^+$, $1 \leq z_1 \leq \gamma - 1$, $0 \leq z_j \leq \gamma - 1$ ($2 \leq j \leq k$). Число нуль также является машинным числом ($p_\gamma(0) = 0$, $m_\gamma(0) = 0$).

При численной реализации алгоритма необходимо, чтобы возмущения, выбираемые на итерациях алгоритма, были достаточно велики для выделения векторов из групп. Необходимо, чтобы соответствующие компоненты векторов, равные до возмущения, были отделимы после возмущения как *машинные* числа. Иными словами, необходимо, чтобы точности нахождения векторов-решений систем линейных уравнений (3) и их представления машинными числами было достаточно для численной реализации расщепления групп $R_i(A)$.

Покажем, что на каждой j -ой итерации может быть выбрано возмущение $\varepsilon_j = \varepsilon_j(\Delta^j)$, не приводящее к попаданию выделяемого из некоторой группы вектора x_j в другую группу, и при этом длина мантиссы k обрабатываемых машинных чисел, обеспечивающая необходимую точность, может быть задана на старте алгоритма, оставаясь неизменной на протяжении всей его работы.

Пусть $i, j: x_i, x_j \in R_i(A)$, то есть в группе $R_i(A)$ имеется как минимум два элемента, препятствующие установлению взаимно однозначного соответствия алгоритмом спектрального расщепления на j -ой итерации. Для решений систем линейных уравнений (3), если $\exists P: x_i = Px_j$, то

$$x_{ii} = x_{jj}.$$

Если же после j -ой итерации

$$x_{ii} \neq x_{jj},$$

то это означает выделение из группы $R_i(A)$ вектора x_j , и уменьшения на единицу $|R_i(A)|$, а если $\forall i |R_i(A)| = 1$, то возможно установление взаимно однозначного соответствия между вершинами графов G_A и G_B .

Предложение 1. Если до j -ой итерации алгоритма $\exists P$ – матрица перестановки: $x_j = Px_i$, $x_{jj} = x_{ii}$, и возмущение на j -ой итерации алгоритма $\varepsilon_j > 0$, то

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

Доказательство.

Пусть, если не оговаривается дополнительно, обозначения со штрихом – векторы, матрицы и множества, в которые переходят соответствующие нештрихованные векторы и множества после проведения j -ой итерации, они же без штрихов – векторы, матрицы и множества, имеющиеся перед j -ой итерацией.

$$A \stackrel{dn}{=} A^{j-1}, A' \stackrel{dn}{=} A^j.$$

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

После возмущения:

$$|A'| = |A| + \varepsilon_j A_{jj}.$$

$$A'_{jj} = A_{jj},$$

так как возмущается единственный элемент матрицы A – диагональный элемент a_{jj} .

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

То есть

$$|x'_{jj} - x'_{ii}| = \left| \frac{A'_{jj}}{|A'|} - \frac{A'_{ii}}{|A'|} \right| = \left| \frac{A'_{jj}}{|A'|} - \frac{A_{ii} + \varepsilon_j A_{ii,jj}}{|A'|} \right| = \frac{\varepsilon_j A_{ii,jj}}{|A| + \varepsilon_j A_{jj}}. \quad (12)$$

$A_{ii,jj}$ – определитель подматрицы, получаемой удалением из матрицы A i -о и j -о рядов. A – симметрическая положительно определенная матрица со строгим диагональным преобладанием, то есть для нее выполняются условия Адамара:

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

где

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

Следовательно [12, с. 382],

$$A_{ii,jj} \geq H_1 \dots \widehat{H_i} \dots \widehat{H_j} \dots H_n = d'_1 \dots \widehat{d'_i} \dots \widehat{d'_j} \dots d'_n > d^{n-2}. \quad (13)$$

С другой стороны, по теореме Гершгорина [12, с. 390] для $\lambda_{\max}$ – максимального собственного значения матрицы A справедлива оценка:

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

если $\forall k \ \varepsilon_k < 1$. То есть

$$|A| = \prod_{k=1}^n \lambda_k < \lambda_{\max}^n < (3d+1)^n, \quad (14)$$

$$A_{jj} = \prod_{k=1}^{n-1} \lambda'_k < \lambda_{\max}^{n-1} < (3d+1)^{n-1}, \quad (15)$$

и $\lambda'_{\max}$ – максимальное собственное значение подматрицы, получаемой удалением из матрицы A i -о и j -о рядов, не превышает $\lambda_{\max}$ [12, с. 350]. Следовательно, учитывая (13), (14), (15), из (12) получаем:

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

то есть

$$|x'_{jj} - x'_{ii}| > \frac{1}{3^n d^2 (3d/\varepsilon_j + 1)}. \quad (16)$$

■

Замечание 2. Поскольку оценки (14) и (15) получены простым мажорированием произведения собственных значений матрицы A степенью максимального собственного значения матрицы $\lambda_{\max}$, то оценка (16) расщепления множества $R_i(A)$ справедлива вне зависимости от кратности собственных значений матриц (1) и их возмущенных матриц, и, соответственно, от кратности собственных значений матриц смежности графов. То есть класс графов, для которых задача проверки изоморфизма графов разрешима за полиномиальное время алгоритмом, представленным в [2], является таковым и для алгоритма спектрального расщепления.

Пусть решение систем линейных уравнений (3) осуществляется методом Гаусса-Зейделя. Помимо того, что для решаемых систем уравнений с заданными матрицами для этого метода присутствует геометрическая сходимость приближенных решений к решениям точным, применение этого метода позволяет оценить приближение по значащим разрядам мантииссы на каждой итерации. Справедлива [13, с. 363–369] следующая теорема:

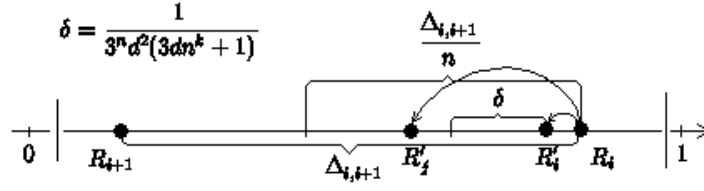
Теорема 2. Пусть при всех i

$$\sum_{j \neq i} |a_{ij}| \leq \mu |a_{ii}|,$$

$\mu < 1$. Тогда

$$\|x_i - \tilde{x}_i^{(s)}\| \leq \mu \|x_i - \tilde{x}_i^{(s-1)}\|,$$

где x_i – точное решение системы линейных уравнений $A^i x = e_i$, $\tilde{x}_i^{(s)}$ – его приближение на s -ой итерации метода Гаусса-Зейделя. ■

Рис. 2. Расщепление множества R_i

Для матриц вида (1) $\mu \leq 1/2$. Следовательно, на s -ой итерации метода Гаусса-Зейделя:

$$|x_{ii} - \tilde{x}_{ii}^s| \leq \|x_i - \tilde{x}_i^s\| < \frac{1}{2^s} \delta^0, \quad |x_{jj} - \tilde{x}_{jj}^s| \leq \|x_j - \tilde{x}_j^s\| < \frac{1}{2^s} \delta^0,$$

где δ_0 – погрешность начального приближения. Если возмущение на j -ой итерации $\varepsilon_j = 1/n^p$, то (16) принимает вид

$$|x'_{jj} - x'_{ii}| > \frac{1}{3^n d^2 (3dn^p + 1)}. \quad (17)$$

Следовательно, если

$$\frac{1}{2^{s-1}} \delta^0 \leq \frac{1}{3^n d^2 (3dn^p + 1)}, \quad (18)$$

то расщепление x_{ii} и x_{jj} будет вычислительно эффективным при заданной до запуска алгоритма длине мантиссы, которая может быть ограничена n .

Найдем количество итераций s , осуществление которых необходимо для вычислительно эффективного расщепления. (18) эквивалентно

$$3^n d^2 (3dn^p + 1) \delta^0 \leq 2^{s-1}. \quad (19)$$

Логарифмируя (19), получаем:

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

то есть число необходимых итераций может быть оценено снизу как

$$s \geq n \log_2 3 + p \log_2 n + 4 \log_2 d + \log_2 \delta^0 + 3. \quad (20)$$

$p > 0$ – целое число, определяющее возмущение $\varepsilon_j = 1/n^p$. p вычисляется с целью предотвращения попадания выделяемого из группы R_i элемента x_j в другую ближайшую к ней группу R_{i+1} , находящуюся от R_i на расстоянии $\Delta_{i,i+1}$: p определяется из неравенства

$$\frac{1}{n^p} < \frac{\Delta_{i,i+1}}{n},$$

то есть

$$p > \log_n \frac{1}{\Delta_{i,i+1}} + 1.$$

Если на j -ой итерации $\varepsilon_j = 1/n^p$, A^j – возмущенная матрица A на j -ой итерации, то

$$\begin{aligned} |||x'_i|| - ||x'_j||| &< \frac{\varepsilon_j}{\|A^j\|} \cdot \frac{3\|A^j\|}{\|A^j\| - 3\varepsilon_j} \|x'_i\| = \\ &= \frac{3\varepsilon_j}{\|A^j\| - 3\varepsilon_j} \|x'_i\| = \frac{3 \cdot 1/n^p}{\|A^j\| - 3 \cdot 1/n^p} \|x'_i\| = \frac{3}{\|A^j\|n^p - 3} \|x'_i\| \approx \frac{3\|x'_i\|}{\|A^j\|n^p}, \end{aligned}$$

и

$$|||x'_i|| - ||x'_j||| < \frac{3\|x'_i\|}{\|A^j\|n^p} < \frac{3\|x'_i\|}{(d + d_j)n^p}.$$

$$\|x'_i\| < \frac{\sqrt{10}}{3d}$$

[11, с. 139], и с учетом утверждения 1 получаем:

$$\frac{1}{3^n d^2 (3dn^p + 1)} < |x'_{ii} - x'_{jj}| < |||x'_i|| - ||x'_j||| < \frac{\sqrt{10}}{d(d + d_j)n^p}.$$

То есть интервал, в котором происходит расщепление, может быть задан так, чтобы гарантированно не происходило попадания выделенного из одной группы $R_i(A)$ вектора в другую группу $R'_{i+1}(A)$, имеющуюся на следующей итерации (Рис. 1).

Рассмотрим случай, когда графы G_A , G_B – полные графы. Тогда имеется одна группа решений, то есть $m = 1$, и $|R_1| = n$. Так как число векторов в одной группе R_1 равно n , то:

1. Выбрав $\varepsilon_1 = 1/n$ на первой итерации, на которой происходит расщепление этой группы, мы получим выделение первого вектора из этой группы:

не менее чем на $1/(3^n d^2 (3dn + 1))$ и не более чем $\sqrt{10}/(d(d + d_j)n)$,
необходимое число итераций – $s = n \log_2 3 + \log_2 n + 4 \log_2 d + \log_2 \delta^0 + 3$.

2. Выбрав $\varepsilon_2 = 1/n^2$ на второй итерации, на которой происходит расщепление группы, мы получим выделение второго вектора из этой группы:

не менее чем на $1/(3^n d^2 (3dn^2 + 1))$ и не более чем на $\sqrt{10}/(d(d + d_j)n^2)$,
необходимое число итераций – $s = n \log_2 3 + 2 \log_2 n + 4 \log_2 d + \log_2 \delta^0 + 3$.

Так, далее продолжая выделять векторы из этой группы, для n -о вектора получаем (n -я итерация): при выборе $\varepsilon_n = 1/n^n$, происходит выделение n -о вектора не менее чем на $1/(3^n d^2 (3dn^n + 1))$ и не более чем на $\sqrt{10}/(d(d + d_j)n^n)$,
необходимое число итераций – $s = n \log_2 3 + n \log_2 n + 4 \log_2 d + \log_2 \delta^0 + 3$.

То есть, задав длину мантиссы равной n и совершая на каждой итерации $s = n \log_2 3 + n \log_2 n + 4 \log_2 d + \log_2 \delta^0 + 3$ итераций метода Гаусса-Зейделя при решении систем линейных уравнений на каждой итерации алгоритма, мы получим расщепление всех групп R_i при эффективной численной реализации алгоритма. Отметим, что числовой тип `extended`, реализованный в языках `Object Pascal` и `C++`, позволяет работать с числами в диапазоне от $3,6 \times 10^{-4951}$ до $1,1 \times 10^{4932}$.

Поскольку ни в одном из доказательств лемм и утверждений не использовалось равенство ненулевых положительных элементов матрицы смежности

единице, то алгоритм может быть использован для проверки изоморфизма взвешенных графов с произвольными неотрицательными весами. К этой ситуации легко может быть сведен и общий случай проверки изоморфизма взвешенных неориентированных графов. Существует также модификация алгоритма для проверки изоморфизма ориентированных графов. Проведенные численные эксперименты подтверждают вычислительную эффективность алгоритма.

ЛИТЕРАТУРА

1. Гэри М., Джонсон Д. *Вычислительные машины и труднорешаемые задачи*. М.: Мир, 1982.
2. Babai, L., Grigoryev, D.Y., Mount, D.M. *Isomorphism of graphs with bounded eigenvalue Multiplicity* // Proceedings 14th ACM Symposium on Theory of Computing. 1982.
3. Hopcroft J., Wong J. *A linear time algorithm for isomorphism of planar graphs* // Proceedings of the Sixth Annual ACM Symposium on Theory of Computing. 1974.
4. Luks E.M. *Isomorphism of graphs of bounded valence can be tested in polynomial time* // Proc. 21st IEEE FOCS Symp. 1980.
5. Jiang X., Bunke H. *Including Geometry in Graph Representations: a Quadratic Time Isomorphism Algorithm and Its Applications* // Advances in Structural and Syntactic Pattern Recognition (Perner P., Wang P., Rosenfeld A., eds). Springer Verlag. Lecture Notes in Computer Science. 1996. V. 1121.
6. Ullmann J.R. *An Algorithm for Subgraph Isomorphism* // Journal of the Association for Computing Machinery. 1976. V. 23.
7. Schmidt D.C., Druffel L.E. *A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices* // Journal of the Association for Computing Machinery. 1976. V. 23.
8. McKay B.D. *Practical Graph Isomorphism* // Congressus Numerantium. 1981. V. 30.
9. Cordella L.P., Foggia P., Sansone C., Vento M. *Evaluating Performance of the VF Graph Matching Algorithm* // Proc. of the 10th International Conference on Image Analysis and Processing, IEEE Computer Society Press. 1999.
10. С.К. Годунов и др. *Гарантированная точность решения систем линейных уравнений в евклидовых пространствах*. Новосибирск: Наука, 1988.
11. Пролубников А.В., Файзуллин Р.Т. *О вычислительной эффективности алгоритма спектрального расщепления проверки изоморфизма графов* // Сборник «Компьютерная оптика». Под ред. В.Н. Сойфера. Издательство Самарского государственного университета. 2003. №24.
12. Гантмахер Ф.Р. *Теория матриц*. М.: Наука, 1976.
13. Бахвалов Н.С. *Численные методы*. М.: Наука, 1975. Т. 1.