

УДК 004.02 + 519.1

Полная характеристика структуры неориентированного графа

Иванова Г. С.^{1,*}, Овчинников В. А.¹

* gsivanova@gmail.com

¹МГТУ им. Н.Э. Баумана, Москва, Россия

Предметом исследования является распознавание изоморфизма неориентированных графов как модели задачи установления идентичности структур сложных систем. Цель работы – получение за полиномиальное время структурной характеристики графа, которая могла бы быть использована для решения указанной задачи. Обоснована возможность сведения задачи установления изоморфизма неориентированных графов к задаче определения изоморфизма их расщепления на деревья всех кратчайших, в смысле числа ребер, цепей из каждой вершины в остальные. В качестве интегральной характеристики графа предлагается множество кодов корней деревьев кратчайших цепей. Показано, что вычислительная сложность получения кодов корней деревьев не хуже $O(n^3)$. Разработана программа получения указанных кодов. Выполнены экспериментальные исследования, которые подтвердили теоретические положения.

Ключевые слова: идентичность структур, изоморфизм графов, деревья кратчайших цепей, коды вершин деревьев

Введение

Проблема получения полной характеристики графа связана в основном с задачей определения идентичности структур сложных систем. Эта задача часто возникает в различных предметных областях. Структуры систем идентичны, если входы и выходы их подсистем соединены одинаково, а типы подсистем совпадают. При формализованном установлении идентичности сложных систем их структуры представляют различного вида графами: ультраграфами $HU(X, U)$, гиперграфами $H(X, U)$, ориентированными $G^{\rightarrow}(X, U)$ и неориентированными $G^{\sim}(X, U)$ графами, где X – множество вершин, U – множество ребер [1]. Если граф является адекватной моделью системы в смысле полноты и правильности отображения информации, полностью характеризующей систему, задача установления идентичности систем сводится к определению изоморфизма их моделей.

Алгоритмы определения изоморфизма графов используются при решении таких прикладных задач, как распознавание образов, формализованный контроль соответствия между исходной электрической схемой и ее топологической реализацией при проектировании больших интегральных схем и др.

Широко известны алгоритмы распознавания изоморфизма графов, основанные на использовании локальных характеристик вершин и реализующие метод поиска в глубину с возвращением [2–6]. В последние годы появились эвристические алгоритмы, использующие комбинации локальных инвариантов [7–9]. Все эти алгоритмы не накладывают ограничений на структуру графа и имеют невысокую емкостную сложность. Однако, чем больше кратность характеристик вершин графов, тем больше число вариантов возможных подстановок вершин при распознавании изоморфизма таких графов. В связи с этим вычислительная сложность этих алгоритмов может меняться от полиномиальной до экспоненциальной в зависимости от числа вершин с одинаковыми характеристиками.

Известно, что задача изоморфизма полиномиально разрешима для некоторых классов графов [10–12]. В частности для планарных графов, графов с ограниченной степенью вершин, графов с ограниченной кратностью собственных значений из спектра матрицы смежности построены эффективные алгоритмы решения этой задачи. Поскольку эти алгоритмы используют специфические структурные характеристики графов, область их эффективного применения ограничена указанными классами графов.

В работах [13, 14] предлагается алгоритм распознавания изоморфизма ОС2-графов полиномиальный как по времени, так и по памяти. Этот алгоритм работает со спектральными характеристиками графов и имеет вычислительную сложность не хуже $O(n^4)$. Алгоритм использует инвариант, представляющий собой кортеж произведений собственных значений модифицированной матрицы смежности графа и собственных значений матриц подграфов, полученных из исходного удалением i -ой вершины. Алгоритм относится к классу алгоритмов ограниченного применения.

В работах [15–20] предложен метод интеграции структурных различий – вычисления для каждой вершины графа уникальной структурной характеристики. Интегральный описатель структуры графа в виде вектора кодовых чисел вершин используется в качестве его полного инварианта. Разработаны полиномиальные алгоритмы распознавания изоморфизма неориентированных графов и гиперграфов. Указанные работы внесли существенный вклад в развитие методов инвариантного представления графов и анализа их структур. В работе [19] приведена предельная оценка суммарного объема вычислений при определении полного инварианта неориентированного графа с n вершинами и m ребрами в виде $L(G) < \tau(n, m) n^2$, где $\tau(n, m)$ – объем вычислений, затрачиваемый для выполнения одного шага интеграции. Если один шаг интеграции без учета упорядочивания вектора кодовых чисел требует n^2 операций, то асимптотическая оценка вычислительной сложности алгоритма вычисления полного инварианта неориентированного графа равна $O(n^4)$.

В работе [21] предложен метод определения изоморфизма графов с квазиполиномиальной оценкой вычислительной сложности, где степень n является функцией $\log n$.

Цель настоящей работы – предложить и обосновать такую характеристику графа, которая определяла бы его структуру с точностью до изоморфизма, и разработать алгоритм ее получения, имеющий вычислительную сложность меньше $O(n^4)$.

В предлагаемой работе рассматриваются неориентированные $G(X, U)$ графы, не имеющие весов вершин и ребер, определяемых характеристиками или свойствами отображаемых компонентов системы – подсистем и их связей. Так как в таком графе каждое ребро инцидентно только двум вершинам, его достаточно задавать множеством вершин X и множеством их образов относительно предиката смежности $F_1(X, X)$, т. е. в форме $G(X, F_1X)$.

1. Обоснование полной структурной характеристики неориентированного графа

Обоснованием полной характеристики структуры неориентированного графа являются изложенная ниже теорема о возможности сведения задачи установления изоморфизма неориентированных графов к задаче определения изоморфизма их расщепления на деревья *всех* кратчайших, в смысле числа ребер, цепей из каждой вершины в остальные и способ кодирования структуры дерева [22].

Начнем с предположения, что полной характеристикой графа, определяющей его структуру с точностью до изоморфизма, является совокупность *кратчайших*, в смысле числа ребер, цепей из каждой вершины во все остальные. Таким образом, индивидуальной характеристикой каждой вершины графа будет дерево таких кратчайших цепей с корнем в этой вершине. Отсюда следует, что деревья двух сопоставленных друг другу вершин изоморфных графов должны иметь одинаковую структуру, не зависящую от порядка записи вершин в их образах относительно предиката смежности $F_1(X, X)$. Нетрудно убедиться в том, что предположение ошибочно, рассмотрев рис. 1. Деревья №1 и №2 получены процедурой *BFS* просмотра графа в ширину [23] при указанных порядках записи вершин в образе F_1x_1 (напомним, что при построении деревьев кратчайших цепей этой процедурой вершины в дереве не повторяются). Порядок записи вершин в образах остальных вершин графа одинаковый. Деревья №3 и №4 – это деревья *всех* простых цепей из той же вершины при тех же данных, их структура совпадает.

Теорема. Задача распознавания изоморфизма двух неориентированных графов эквивалентна задаче установления изоморфизма деревьев *всех* кратчайших, в смысле числа ребер, цепей из каждой вершины в остальные.

Неформальное доказательство. Графы считаются изоморфными в том случае, если они имеют одинаковую структуру. Предположим, что для графа $G(X, U)$ найдется граф $G(Y, V)$, отличающийся от него структурой и имеющий такое же количество вершин и ребер. Граф $G(Y, V)$ можно получить из графа $G(X, U)$ только изменением отношения инцидентности вершин и ребер, что приведет и к изменению отношения смежности вершин.

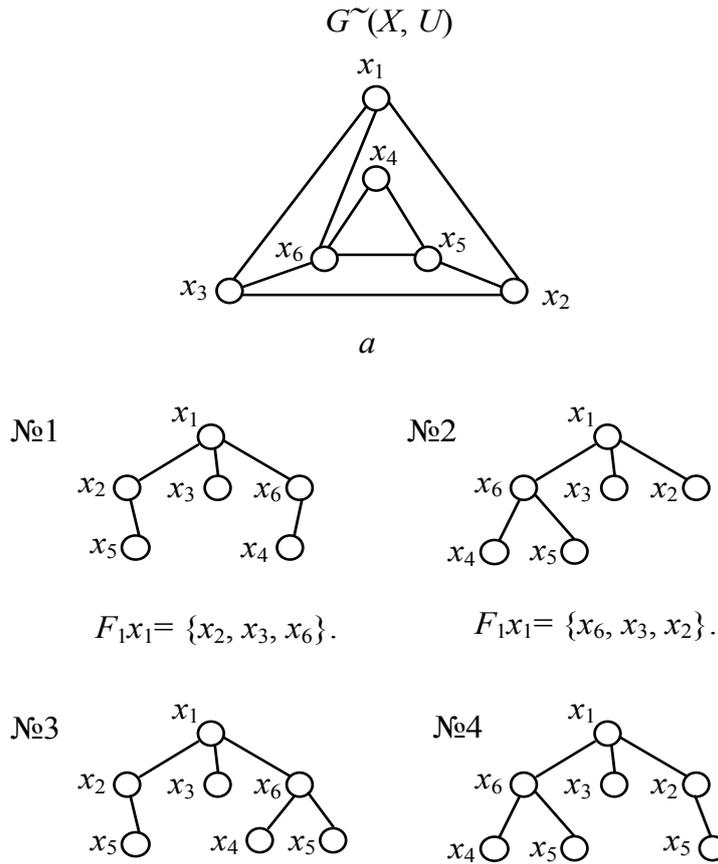


Рис. 1. Неориентированный граф и деревья: №1 и №2 – кратчайших цепей из вершины x_1 во все остальные; №3 и №4 – всех кратчайших цепей из вершины x_1 во все остальные

Если при этом будет получен граф с другой структурой, то изменится последовательность вершин хотя бы в одной цепи и, следовательно, структура хотя бы одного дерева, т.е. в множестве деревьев кратчайших цепей графов $G^{\sim}(X, U)$ и $G^{\sim}(Y, V)$ будут существовать неизоморфные деревья.

На рис. 2 показаны графы $G^{\sim}(X, U)$, $G^{\sim}(Y, V)$, $G^{\sim}(Z, W)$ и их деревья всевозможных кратчайших цепей из каждой вершины до всех остальных. Граф $G^{\sim}(Y, V)$ был получен из графа $G^{\sim}(X, U)$ такой перестановкой ребра, которая не изменила структуры, и, следовательно, должен быть изоморфен ему. Перестановка ребра при получении графа $G^{\sim}(Z, W)$ изменила структуру, следовательно, графы $G^{\sim}(X, U)$ и $G^{\sim}(Z, W)$ должны быть неизоморфны.

Нетрудно видеть, что графы $G^{\sim}(X, U)$ и $G^{\sim}(Y, V)$ – изоморфны и для каждого дерева кратчайших цепей графа $G^{\sim}(X, U)$ существует имеющее такую же структуру дерево кратчайших цепей графа $G^{\sim}(Y, V)$. В то же время деревья кратчайших цепей не изоморфного им графа $G^{\sim}(Z, W)$ имеют другую структуру.

Формальное доказательство. Обозначим цепь, соединяющую вершины x_i и x_l , как $Ch_{i,l}(x_i, u_j, x_k, u_r, \dots, x_t, u_f, x_l)$ или $Ch_{i,l}(X_{i,l}, U_{j,f})$, где $X_{i,l} = \{x_i, x_k, \dots, x_l\}$ и $U_{i,l} = \{u_j, u_r, \dots, u_f\}$. Определяя все кратчайшие цепи, связывающие каждую вершину графа $G^{\sim}(X, U)$ со всеми

остальными, мы получаем его представление в виде множества $A = \{A_i \{Ch((x_i, x_i), (u_j, u_j)), x_l \in X \setminus x_i\}\}$.

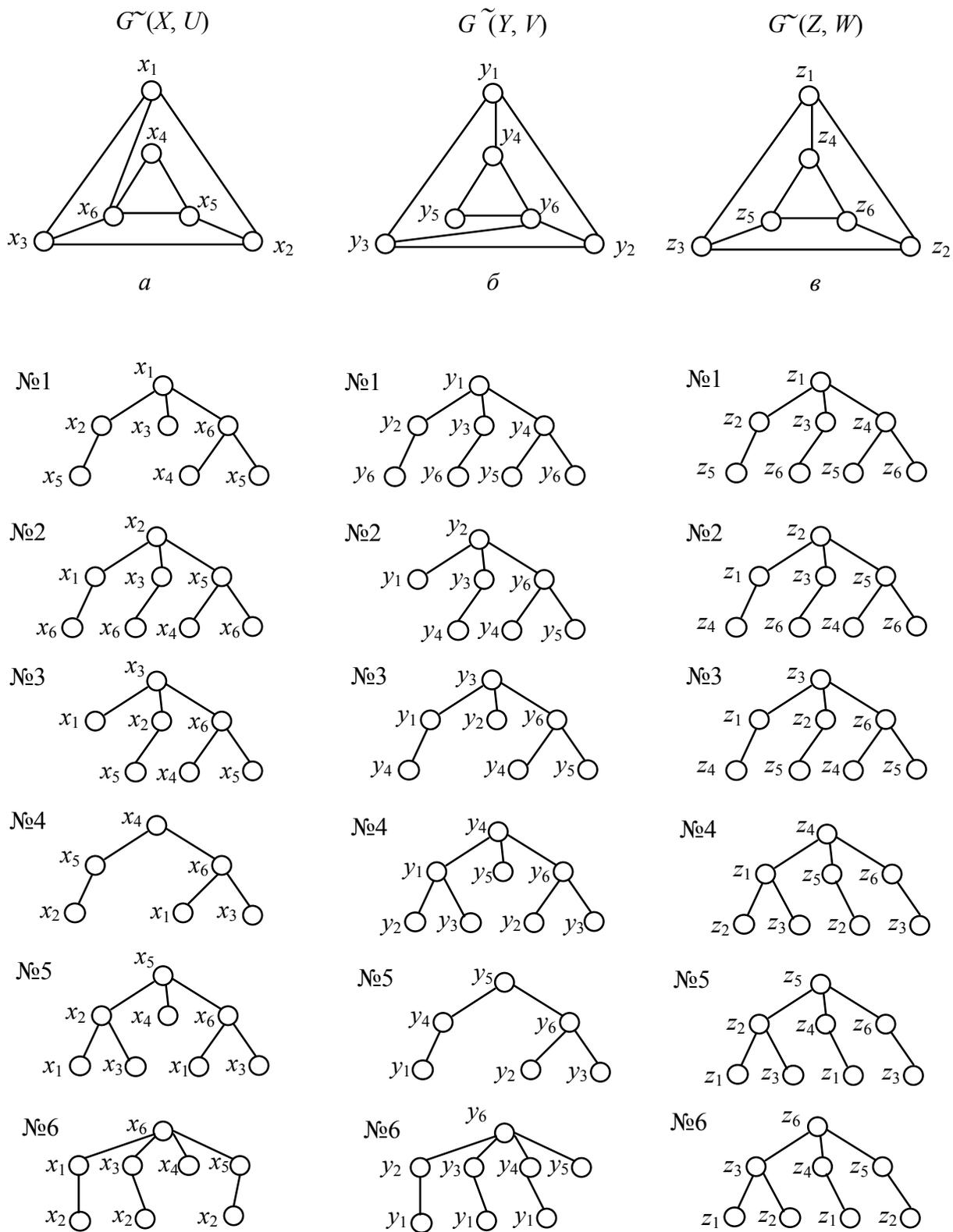


Рис. 2. Граф $G(X, U)$, изоморфный $G(Y, V)$ и не изоморфный $G(Z, W)$ графы, полученные из него перестановкой ребер, и их деревья всех кратчайших цепей

Для совокупности цепей $A_i\{Ch((x_i, x_l), (u_j, u_l)), x_l \in X \setminus x_i\}$ любой вершины x_i по определению должны выполняться следующие условия:

$$\bigcup_{l=1}^n \bigcup_{i=1}^{n-1} X_{i,l} = X \text{ и } \bigcup_{l=1}^n \bigcup_{i=1}^{n-1} U_{i,l} = U, \quad n = |X|. \quad (1)$$

Дерево всех кратчайших цепей из некоторой вершины во все остальные является объединением указанных цепей. Следовательно, для множества этих деревьев также справедливо условие (1).

В общем случае графы $\tilde{G}(X, U)$ и $\tilde{G}(Y, V)$ будут *изоморфными*, если существует взаимно однозначное соответствие $X \leftrightarrow Y, U \leftrightarrow V$, такое, что если вершины $x_i, x_j \in X$ соответствуют вершинам $y_k, y_l \in Y$, то для всех ребер, соединяющих вершины x_i, x_j , существуют ребра, соединяющие вершины y_k, y_l [2]:

$$(\forall u_t = (x_i, x_j) \in U)(\exists u_r = (y_k, y_l) \in V)(x_i \equiv y_k \& x_j \equiv y_l). \quad (2)$$

Поскольку множество деревьев всех кратчайших цепей сохраняет отношения инцидентности вершин и ребер, следовательно, и смежности вершин графа, для графов $\tilde{G}(X, U)$ и $\tilde{G}(Y, V)$ с учетом справедливости (1) условие (2) будет выполняться. Таким образом, изоморфизм множества всех кратчайших цепей вершин графов до всех остальных определяет изоморфизм этих графов. Теорема доказана.

На рис. 3 показан граф и деревья всех кратчайших цепей из каждой вершины во все остальные. Как видно из рисунка деревья имеют различную структуру. Для таких графов установление изоморфизма их деревьев позволяет получить подстановку вершин. Структура деревьев №1 и №3 графа, показанного на рис. 2, совпадает. В этом случае установление изоморфизма деревьев двух графов означает изоморфизм последних, в то время как равенство локальных степеней вершин двух неориентированных графов не является достаточным условием их изоморфизма. Однако для определения подстановки вершин необходимо проверить $k!$ вариантов возможного соответствия, где k – количество деревьев, имеющих одинаковую структуру.

Оценим вычислительную сложность построения деревьев из каждой вершины графа во все остальные. Так как кратчайших цепей из одной вершины в другую может быть несколько, необходимо определить диапазон изменения n_T числа вершин дерева. Минимальное число вершин дерева кратчайших цепей, равное $n = |X|$, будет, если граф $\tilde{G}(X, U)$ полный или является циклом с нечетным числом вершин. Число m_T ребер дерева и максимальное число n_T вершин дерева меньше числа ребер графа m (напомним, что у полного графа $m = n(n-1)/2$). Отсюда вычислительная сложность построения всех деревьев лежит в диапазоне от $O(n^2)$ до $O(n^3)$.

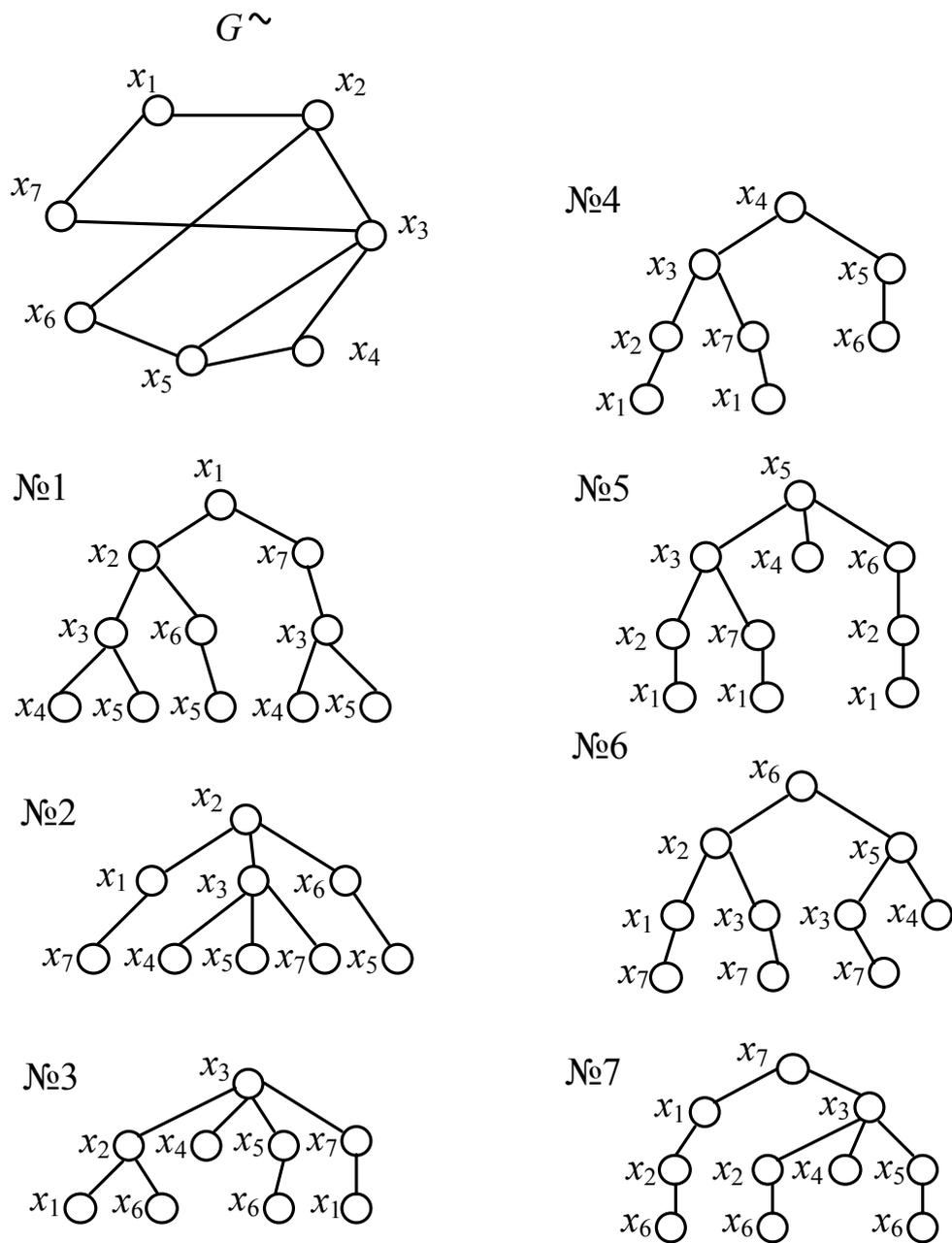


Рис. 3. Граф, все деревья кратчайших цепей которого имеют различную структуру

2. Кодирование структуры деревьев

Кодирование дерева использует идею приписывания вершинам дерева кодов, изложенную в [22]. Код корня описывает структуру дерева и является его полным инвариантом – деревья изоморфны тогда и только тогда, когда коды их корней совпадают.

Коды корней строятся, начиная с листьев дерева. Каждому листу дерева присваивается код, состоящий из открывающей и закрывающей скобок «()». Коды остальных вершин представляют собой заключенную в круглые скобки последовательность отсортированных в лексикографическом порядке кодов корней поддеревьев данной вершины. Например, вершины дерева №1 рис. 2 будут закодированы так, как показано на рис. 4.

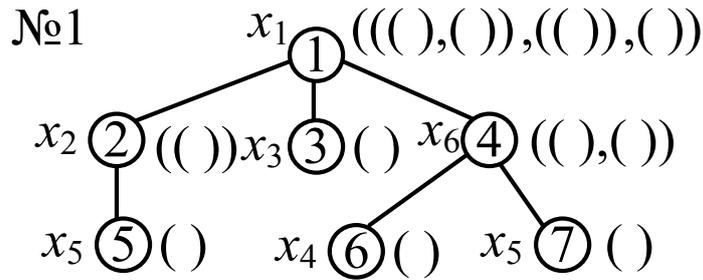


Рис. 4. Пример кодирования вершин дерева

В вершинах дерева указаны их номера в порядке получения, справа от вершин указаны их коды.

На основании доказанной выше теоремы и положений работы [22] в качестве интегральной характеристики графа предлагается множество кодов корней деревьев всех кратчайших цепей из каждой вершины в остальные.

3. Алгоритм определения кодов корней деревьев кратчайших цепей

Определение кодов корней деревьев кратчайших цепей выполняется в два этапа: сначала строится дерево, а потом получается код его корня. Построение деревьев осуществляется по стратегии декомпозиции в ширину. При получении кодов вершин выполняется обход дерева по методу в глубину с возвращением.

Как указано выше граф G задан множествами вершин $X = \{x_i / i = \overline{1, n}\}$ и образов относительно предиката смежности $F_1X = \{F_1x_i / i = \overline{1, n}\}$. При этом для каждой вершины x_j дерева фиксируется соответствующий ей номер вершины графа num_j , количество $kson_j$ ее сыновей и массив S_j номеров сыновей в дереве. Эта информация однозначно описывает дерево кратчайших цепей и используется на втором этапе при получении кода корня дерева.

Множество S_j сыновей вершины x_j для q -го уровня дерева определяется как множество смежных ей вершин F_1x_j без вершин, которые уже были найдены на предыдущих $q-1$ уровнях дерева поиска кратчайших путей X^{q-1} :

$$S_j = F_1x_j \setminus X^{q-1}.$$

На этапе построения кода корня дерева каждой вершине x_j сопоставляется ее код $code_j$, который потом участвует в построении кодов родительских вершин до корневой вершины включительно.

Таким образом, каждой вершине x_j сопоставляется кортеж вида:

$$\langle num_j, kson_j, S_j, code_j \rangle.$$

Результат работы алгоритма – вектор $Code$ кодов корней деревьев кратчайших цепей размером n . Схема алгоритма показана на рис. 5.

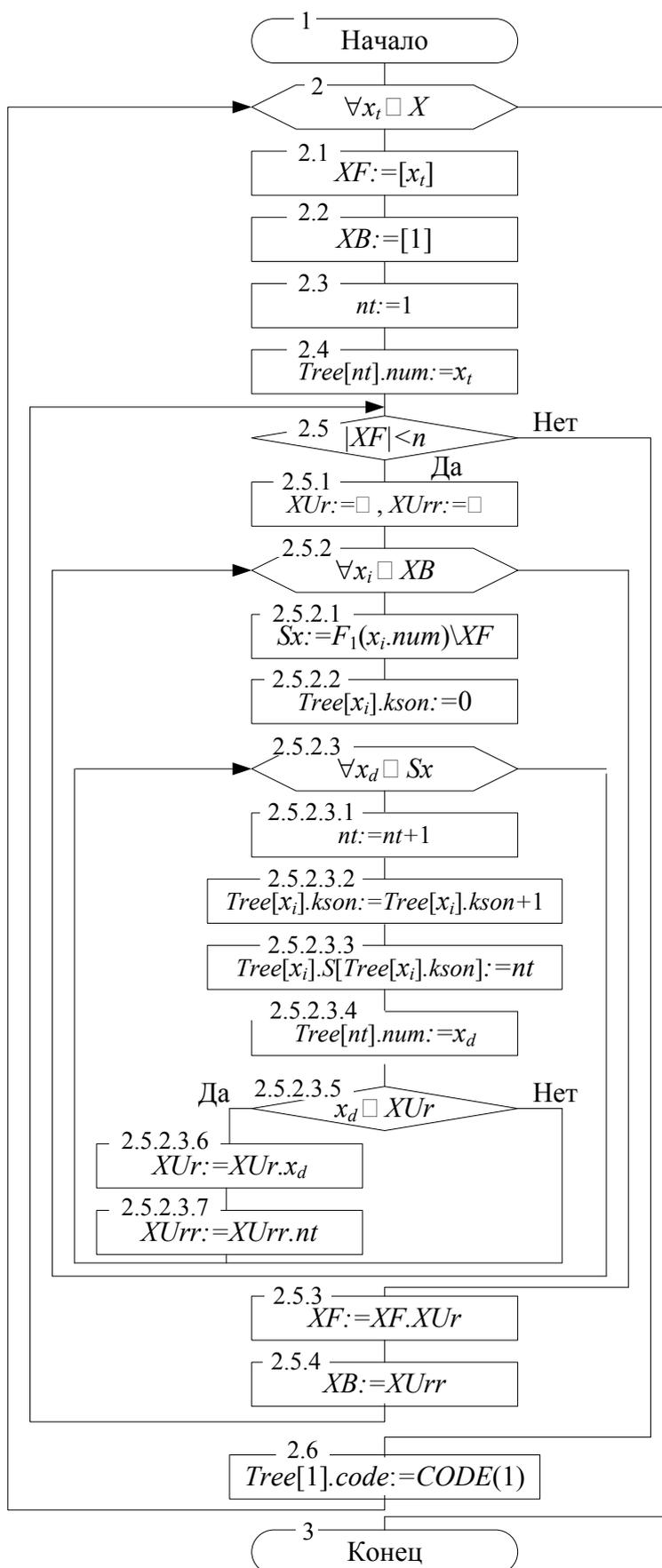


Рис. 5. Схема алгоритма получения кодов корней деревьев кратчайших цепей

Условные обозначения, используемые в описании алгоритма:

XF – множество вершин графа G^{\square} , к которым уже найдены кратчайшие пути;

XB – множество вершин графа G^{\square} , из которых будут строиться ветви на очередном шаге декомпозиции в ширину;

XUr – множество вершин графа G^{\square} , найденных на рассматриваемом уровне;

Ur – номер текущего уровня дерева декомпозиции в ширину;

Sx – множество сыновей вершины;

$CODE(x_i)$ – функция, рекурсивно вычисляющая коды вершин дерева по алгоритму, предложенному в [22]. В ходе работы алгоритма выполняется лексикографическая сортировка кодов вершин деревьев, обеспечивающая независимость кодов от порядка нумерации вершин деревьев. Согласно [22] вычислительная сложность получения кода – кортежа корня дерева не превышает $O(n_T)$, т. е. $O(n^2)$. Таким образом, вычислительная сложность построения деревьев кратчайших цепей и получения кодов их корней для всех вершин графа не превышает $O(n^3)$.

4. Экспериментальные исследования

Для выполнения экспериментальных исследований приведенный выше алгоритм был реализован в среде Turbo Delphi.

Очевидно, что число вершин дерева кратчайших цепей и, соответственно, временная сложность алгоритма зависят от плотности исходного графа D , которая определяется отношением числа ребер графа m к числу ребер полного графа. Для получения оценки временной сложности алгоритма «в худшем» потребовалось определить значение плотности графа, при котором количество вершин деревьев, а, следовательно и время их построения максимально. На рис. 6 показана экспериментально полученная зависимость числа вершин дерева кратчайших цепей в функции от плотности графа для числа его вершин $n = 600$. Несложно видеть, что максимальное число вершин деревьев достигается при плотности графа $60\% \leq D \leq 70\%$.

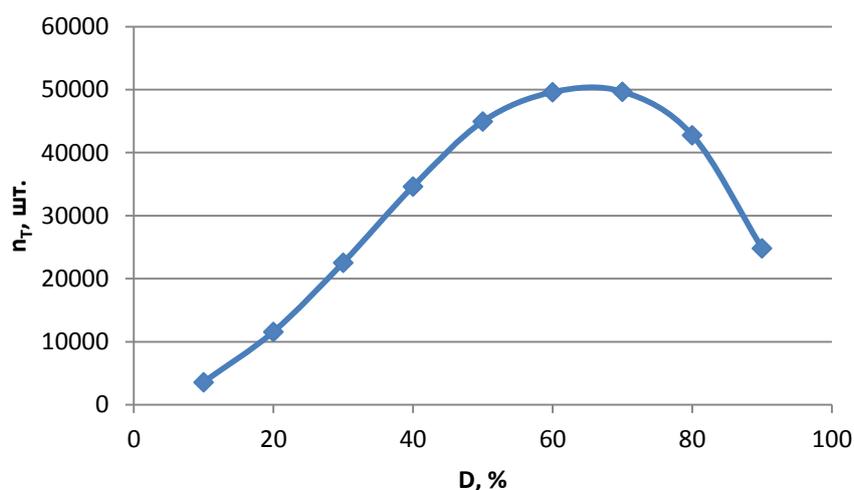


Рис.6. Зависимость числа вершин дерева кратчайших цепей от плотности исходного графа

Анализ полученной зависимости числа вершин дерева всех кратчайших цепей от числа вершин графа при его плотности $D = 65\%$ показал (рис. 7), что она с большой точностью аппроксимируется квадратичной функцией $n_T = n(n-1)/7$, что согласуется с теоретической оценкой раздела 1.

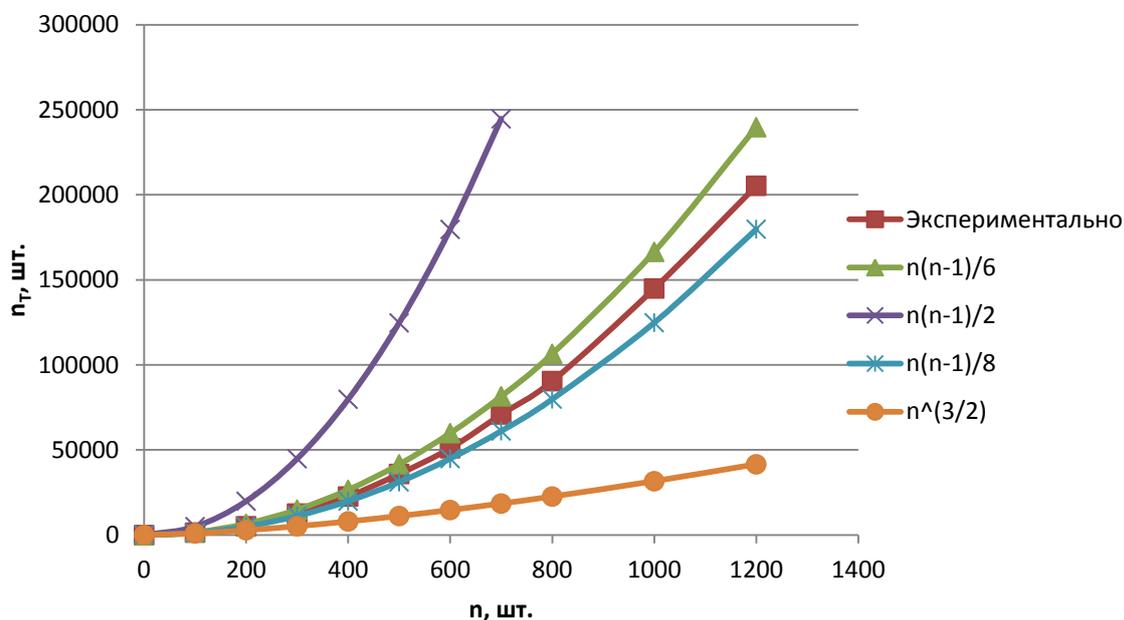


Рис. 7. Зависимость числа вершин дерева кратчайших цепей от числа вершин исходного графа при его плотности $D=60\%$

На рис. 8 показана зависимость времени выполнения программы, реализующей предлагаемый алгоритм, в функции от плотности графа для шести значений числа вершин. Несложно видеть, что максимальное время требуется для построения деревьев при плотности графа $D \approx 60\%$, причем данная оценка не зависит от числа вершин графа.

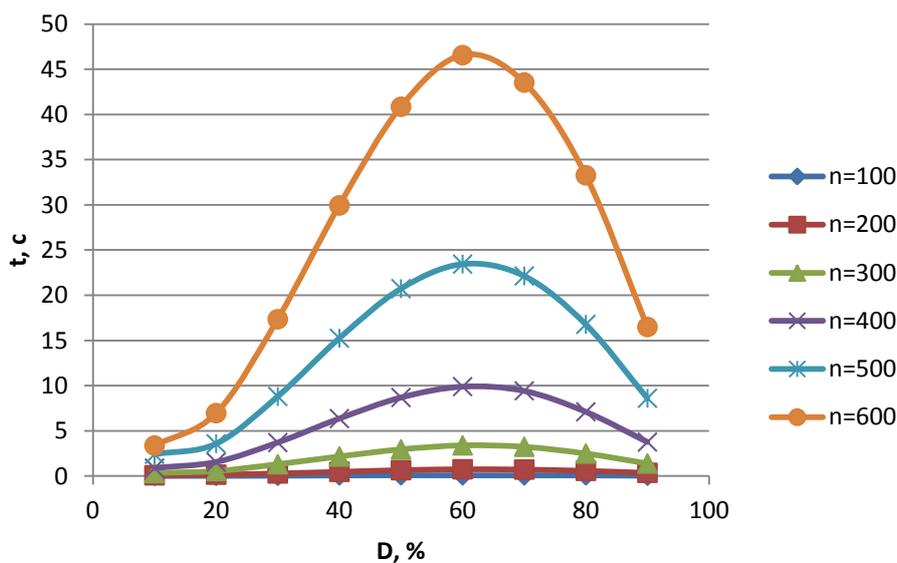


Рис. 8. Зависимость времени выполнения программы, реализующей предлагаемый алгоритм, в функции от плотности графа

Зависимость временной сложности алгоритма от числа вершин исходного графа при его плотности $D = 30\%$ (в часто встречающемся) и 60% (в худшем случае) показана на рис. 9.

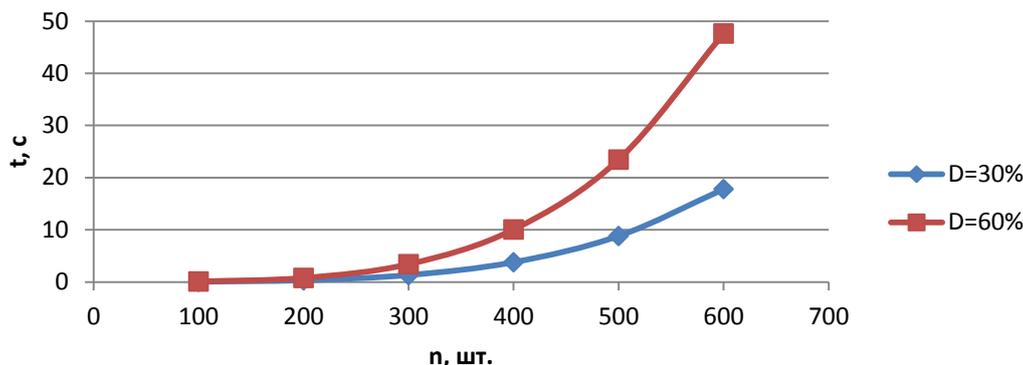


Рис. 9. Зависимость временной сложности алгоритма от числа вершин исходного графа при его плотности D равной 30% и 60%

Вид зависимости временной сложности от числа вершин исходного графа для алгоритма получения деревьев и определения кодов их корней согласуется с теоретическими оценками, полученными в разделах 1 и 3.

Заключение

В статье предложена и обоснована полная характеристика структуры неориентированного графа в виде совокупности кодов корней деревьев всех кратчайших цепей его вершин. Показано, что вычислительная сложность построения деревьев кратчайших цепей и получения кодов их корней для всех вершин графа не превышает $O(n^3)$. Указанная характеристика позволяет определить изоморфизм графов и получить подстановку их вершин, если в разложении графов нет деревьев с одинаковой структурой. В том случае, когда имеются k деревьев с одинаковой структурой, устанавливается изоморфизм графов и определяется частичная подстановка их вершин. Для получения полной подстановки необходимо исследовать $k!$ вариантов подстановок. Применение предложенного в статье подхода к ориентированным, гипер- и ультраграфам является предметом дальнейших исследований.

Список литературы

1. Овчинников В.А. Графы в задачах анализа и синтеза структур сложных систем. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 423 с.
2. Овчинников В.А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем: учеб. для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 288 с.

3. Овчинников В.А. Вычислительная сложность алгоритмических моделей задач идентификации и покрытия схем ЭВМ // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 1992. № 2. С. 31-42.
4. Ullmann J.R. An Algorithm for Subgraph Isomorphism // Journal of the Association for Computing Machinery (JACM). 1976. Vol. 23, no. 1. P. 31-42. DOI: [10.1145/321921.321925](https://doi.org/10.1145/321921.321925)
5. Cordella L.P., Foggia P., Sansone C., Vento M. A (sub)graph isomorphism algorithm for matching large graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Vol. 26, no. 10. P. 1367-1372. DOI: [10.1109/TPAMI.2004.75](https://doi.org/10.1109/TPAMI.2004.75)
6. Cordella L.P., Foggia P., Sansone C., Vento M. An Improved Algorithm for Matching Large Graphs // Proc. of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen, 2001. P. 149-159.
7. Dehmer M., Grabner M., Mowshowitz A., Emmert-Streib F. An efficient heuristic approach to detecting graph isomorphism based on combinations of highly discriminating invariants // Advances in Computational Mathematics. 2013. Vol. 39, no. 2. P. 311-325. DOI: [10.1007/s10444-012-9281-0](https://doi.org/10.1007/s10444-012-9281-0)
8. McKay B.D., Piperno A. Practical graph isomorphism, II // Journal of Symbolic Computation. 2014. Vol. 60. P. 94-112. DOI: [10.1016/j.jsc.2013.09.003](https://doi.org/10.1016/j.jsc.2013.09.003)
9. Попов А.Ю. Реализация алгоритмов обхода графов в вычислительной системе с многими потоками команд и одним потоком данных // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 10. С. 453-472. DOI: [10.7463/1015.0820736](https://doi.org/10.7463/1015.0820736)
10. Luks E.M. Isomorphism of graphs of bounded valence can be tested in polynomial time // Proc. of the 21st IEEE FOCS Symp. 1980. P. 42-49.
11. Hoffmann C.M. Group-Theoretic Algorithms and Graph Isomorphism // Lecture Notes in Computer Science. 1982. Chap. V. P. 127-138.
12. Babai L., Grigoryev D.Yu., Mount D.M. Isomorphism of graphs with bounded eigenvalue multiplicity // Proceedings of the 14th annual ACM symposium on Theory of computing (STOC '82). 1982. P. 310-324. DOI: [10.1145/800070.802206](https://doi.org/10.1145/800070.802206)
13. Faizullin R.T., Prolubnikov A.V. An Algorithm of the Spectral Splitting for the Double Permutation Cipher // Pattern Recognition and Image Analysis. 2002. Vol. 12, no. 4. P. 365-375. Режим доступа: http://www.omg.ru/general_information/faculties/radio_engineering_department/department_of_quot_integrated_protection_of_information_quot/files/theses/publications/ProlubnikovFaizullinPattern.pdf (дата обращения 01.03.2016).
14. Пролубников А.В. Прямой алгоритм проверки изоморфизма графов // Компьютерная оптика. 2007. Т. 31, № 3. С. 86-92.
15. Погребной В.К. Алгоритм решения задачи определения изоморфизма гиперграфов // Известия Томского политехнического университета. 2010. Т. 317, № 5. С. 16-21. Ре-

- жим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2010/v317/i5/03.pdf (дата обращения 01.03.2016).
16. Погребной В.К. Метод интеграции структурных различий в графовых моделях и его применение для описания структур // Известия Томского политехнического университета. 2011. Т. 318, № 5. С. 10-16. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2011/v318/i5/02.pdf (дата обращения 01.03.2016).
 17. Погребной В.К. Решение задачи определения изоморфизма графов, представленных атрибутными матрицами // Известия Томского политехнического университета. 2012. Т. 321, № 5. С. 52-56. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2012/v321/i5/11.pdf (дата обращения 01.03.2016).
 18. Погребной В.К., Погребной А.В. Полиномиальный алгоритм вычисления полного инварианта графа на основе интегрального описателя структуры // Известия Томского политехнического университета. 2013. Т. 323, № 5. С. 152-159. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2013/v323/i5/25.pdf (дата обращения 01.03.2016).
 19. Погребной А.В. Полный инвариант графа и алгоритм его вычисления // Известия Томского политехнического университета. Информационные технологии. 2014. Т. 325, № 5. С. 110-122. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2014/v325/i5/14.pdf (дата обращения 01.03.2016).
 20. Погребной В.К., Погребной А.В. Исследование полиномиальности метода вычисления интегрального описателя структуры графа // Известия Томского политехнического университета. 2013. Т. 323, № 5. С. 146-151. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2013/v323/i5/24.pdf (дата обращения 01.03.2016).
 21. Babai L. Graph Isomorphism in Quasipolynomial Time // [ArXiv.org](http://arxiv.org): website. Режим доступа: <http://arxiv.org/abs/1512.03547> (дата обращения 15.02.2016).
 22. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Построение и анализ вычислительных алгоритмов: пер. с англ. М.: Мир, 1979. 536 с.
 23. Кормен Т., Лейзерсон Ч., Риверст Р. Алгоритмы: построение и анализ: пер. с англ. М.: МЦНМО, 2000. 960 с.

Completely Described Undirected Graph Structure

G.S. Ivanova^{1,*}, V.A. Ovchinnikov¹

[*gsivanova@gmail.com](mailto:gsivanova@gmail.com)

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: identity structures, graph isomorphism, tree of the shortest paths, codes vertices of trees

The objects of research are undirected graphs. The paper considers a problem of their isomorphism. A literature analysis of its solution, has shown that there is no way to define a complete graph invariant in the form of unique structural characteristics of each its vertex, which has a computational complexity of definition better than $O(n^4)$.

The work objective is to provide the characteristics of the graph structure, which could be used to solve the problem of their isomorphism for a time better than $O(n^4)$. As such characteristics, the paper proposes to use the set of codes of tree roots of *all* the shortest - in terms of the number of edges - paths from each vertex to the others, uniquely defining the structure of each tree. It proves the theorem that it is possible to reduce the problem of isomorphism of the undirected graphs to the isomorphism problem of their splitting into the trees of *all* the shortest - in terms of the number of edges - paths of each vertex to the others. An algorithm to construct the shortest paths from each vertex to all others and to compute codes of their vertices has been developed. As the latter, are used Aho-codes, which find application in recognising the isomorphism of trees. The computational complexity to obtain structural characteristics of vertices has been estimated to be about $O(n^3)$.

The pilot studies involved the full-scale experiment using the developed complex programmes to generate raw data, i.e. analytic representation of the graph with the number of vertices equal to 1200, and a programme to provide codes of the tree roots. To have an estimate of "the worst" in terms of time - complexity of expansion algorithm of graphs into trees of the shortest paths and define the codes of their roots has been an experimentally studied how the number of tree vertices depends on the graph density. For the worst case was obtained a dependence of the number of tree vertices on the number of graph vertices, approximated by the quadratic function, which is consistent with the theoretical estimates. The experimental results of studies, in general, prove those of theoretical estimates.

References

1. Ovchinnikov V.A. *Grafy v zadachakh analiza i sinteza struktur slozhnykh sistem* [Graphs in problems of analysis and synthesis of structures of difficult systems]. Moscow, Bauman MSTU Publ., 2014. 423 p. (in Russian).
2. Ovchinnikov V.A. *Algoritmizatsiya kombinatorno-optimizatsionnykh zadach pri proektirovanii EVM i sistem* [Algorithmization of combinatorial and optimization problems in the design of computers and systems]. Moscow, Bauman MSTU Publ., 2001. 288 p. (in Russian).
3. Ovchinnikov V.A. Computational complexity of algorithmic models of problems of identification and coverage schemes of the computer. *Vestnik MGTU im. N.E. Baumana. Ser. Priborostroenie = Herald of the Bauman Moscow State Technical University. Ser. Instrument Engineering*, 1992, no. 2, pp. 31-42. (in Russian).
4. Ullmann J.R. An Algorithm for Subgraph Isomorphism. *Journal of the Association for Computing Machinery (JACM)*, 1976, vol. 23, no. 1, pp. 31-42. DOI: [10.1145/321921.321925](https://doi.org/10.1145/321921.321925)
5. Cordella L.P., Foggia P., Sansone C., Vento M. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 10, pp. 1367-1372. DOI: [10.1109/TPAMI.2004.75](https://doi.org/10.1109/TPAMI.2004.75)
6. Cordella L.P., Foggia P., Sansone C., Vento M. An Improved Algorithm for Matching Large Graphs. *Proc. of the 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, Cuen, 2001, pp. 149-159.
7. Dehmer M., Grabner M., Mowshowitz A., Emmert-Streib F. An efficient heuristic approach to detecting graph isomorphism based on combinations of highly discriminating invariants. *Advances in Computational Mathematics*, 2013, vol. 39, no. 2, pp. 311-325. DOI: [10.1007/s10444-012-9281-0](https://doi.org/10.1007/s10444-012-9281-0)
8. McKay B.D., Piperno A. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 2014, vol. 60, pp. 94-112. DOI: [10.1016/j.jsc.2013.09.003](https://doi.org/10.1016/j.jsc.2013.09.003)
9. Popov A.Yu. The implementation of graph traversal algorithms on the multiple instruction and single data stream computer system. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 10, pp. 453-472. DOI: [10.7463/1015.0820736](https://doi.org/10.7463/1015.0820736) (in Russian).
10. Luks E.M. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Proc. of the 21st IEEE FOCS Symp.* 1980, pp. 42-49.
11. Hoffmann C.M. Group-Theoretic Algorithms and Graph Isomorphism. *Lecture Notes in Computer Science*, 1982, chap. 5, pp. 127-138.
12. Babai L., Grigoryev D.Yu., Mount D.M. Isomorphism of graphs with bounded eigenvalue multiplicity. *Proceedings of the 14th annual ACM symposium on Theory of computing (STOC '82)*. 1982, pp. 310-324. DOI: [10.1145/800070.802206](https://doi.org/10.1145/800070.802206)

13. Faizullin R.T., Prolubnikov A.V. An Algorithm of the Spectral Splitting for the Double Permutation Cipher. *Pattern Recognition and Image Analysis*, 2002, vol. 12, no. 4, pp. 365-375. Available at: http://www.omgtu.ru/general_information/faculties/radio_engineering_department/department_of_quot_integrated_protection_of_information_quot/files/theses/publications/ProlubnikovFaizullinPattern.pdf , accessed 01.03.2016.
14. Prolubnikov A.V. Direct algorithm for testing isomorphism of graphs. *Komp'yuternaya optika = Computer Optics*, 2007, vol. 31, no. 3, pp. 86-92. (in Russian).
15. Pogrebnoi V.K. Algorithm for solving problem of determining isomorphism of hypergraphs. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2010, vol. 317, no. 5, pp. 16-21. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2010/v317/i5/03.pdf , accessed 01.03.2016. (in Russian).
16. Pogrebnoi V.K. Method of integration of structural differences in graph models and its application for describing structures. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2011, vol. 318, no. 5, pp. 10-16. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2011/v318/i5/02.pdf , accessed 01.03.2016. (in Russian).
17. Pogrebnoi V.K. Solving the problem of determining graph isomorphism represented by the attributive matrices. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2012, vol. 321, no. 5, pp. 52-56. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2012/v321/i5/11.pdf , accessed 01.03.2016. (in Russian).
18. Pogrebnoi V.K., Pogrebnoi A.V. Polynomial algorithm of computing complete graph invariant on the basis of integral structure descriptor. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. 152-159. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2013/v323/i5/25.pdf , accessed 01.03.2016. (in Russian).
19. Pogrebnoi A.V. Complete graph invariant and algorithm of its computation. *Izvestiya Tomskogo politekhnicheskogo universiteta. Informatsionnye tekhnologii = Bulletin of the Tomsk Polytechnic University. Information technology*, 2014, vol. 325, no. 5, pp. 110-122. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2014/v325/i5/14.pdf , accessed 01.03.2016. (in Russian).
20. Pogrebnoi V.K., Pogrebnoi A.V. Polynomiality of method for computing graph structure integral descriptor. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2013, vol. 323, no. 5, pp. 146-151. Available at: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2013/v323/i5/24.pdf , accessed 01.03.2016. (in Russian).

21. Babai L. *Graph Isomorphism in Quasipolynomial Time*. [ArXiv.org](https://arxiv.org/abs/1512.03547): website. Available at: <http://arxiv.org/abs/1512.03547> , accessed 15.02.2016.
22. Aho A.V., Hopcroft J.E., Ullman J.D. *The design and analysis of computer algorithms*. Addison-Wesley, 1976. (Russ. ed.: Aho A.V., Hopcroft J.E., Ullman J.D. *Postroenie i analiz vychislitel'nykh algoritmov*. Moscow, Mir Publ., 1979. 536 p.).
23. Cormen T.H., Leiserson C.E., Rivest R.L. *Introduction to Algorithms*. MIT Press, 1990. (Russ. ed.: Cormen T.H., Leiserson C.E., Rivest R.L. *Algoritmy: postroenie i analiz*. Moscow, MTsNMO Publ., 2000. 960 p.).