УДК 004.021
DOI: 10.25559/SITITO.18.202203.655-665

Оригинальная статья

Каркасный алгоритм оценки схожести невзвешенных неориентированных графов

В. В. Сысоев
ПАО «Сбербанк России», г. Москва, Российская Федерация
Адрес: 117312, Российская Федерация, г. Москва, ул. Вавилова, д. 19
Sysoev.V.V@sberbank.ru

Аннотация

Общеизвестно, что проблема алгоритмов оценки схожести графов упирается в сложность их вычисления. Большинство предлагаемых алгоритмов оценки схожести графов основаны на алгоритме Ульмана. Философия алгоритма Ульмана заключается в работе с матрицами смежности. Алгоритм Ульмана имеет высокую сложность вычисления, применение параллельных вычислений, в том числе, на видеокартах, оставляет алгоритм Ульмана *NP* – сложным. При этом алгоритм Ульмана дает только информацию, изоморфны ли графы или является ли один из них подграфом другого, а для получения численной оценки схожести необходимо проводить дополнительные вычисления. Предлагаемый в статье каркасный алгоритм оценки схожести призывает отойти от философии алгоритма Ульмана, а именно, работы с матрицами смежности. Новый алгоритм основан на построении и изучении каркасов сравниваемых графов. В статье приводится определение и алгоритм построения каркаса графа, который, в свою очередь, строятся на основе ранее опубликованном итерационном алгоритме нахождения всех наикратчайших путей между двумя вершинами. Каркасный алгоритм имеет асимптотическую сложность O $(|V|^2)$, способен выявить гомоморфизм графов и дать количественную оценку схожести двух сравниваемых графов.

Ключевые слова: алгоритм поиска, граф, кратчайший путь, сравнение графов, алгоритм Ульмана, оценка схожести

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Для цитирования: Сысоев В. В. Каркасный алгоритм оценки схожести невзвешенных неориентированных графов // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 3. С. 655-665. doi: https://doi.org/10.25559/SITITO.18.202203.655-665

Original article

A Framework Similarity Estimation of Unweighted Undirected Graphs

V. V. Sysoev
PJSC "Sberbank of Russia", Moscow, Russian Federation
Address: 19 Vavilov St., Moscow 117312, Russian Federation
Sysoev.V.V@sberbank.ru

Abstract

It is well known that the problem of algorithms for estimating the similarity of graphs rests on the complexity of their computation. Most proposed algorithms for estimating graph similarity are based on Ullman's algorithm. The philosophy of Ullmann's algorithm is to work with adjacency matrices. Ulman's algorithm has high computational complexity, and the application of parallel computations, including on video cards, leaves Ullman's algorithm NP-complex. In addition, Ullmann's algorithm just provides information on whether graphs are isomorphic or whether one of them is a subgraph of the other, and additional computations are needed to obtain a numerical estimate of similarity.

The wireframe similarity estimation algorithm proposed in the article calls to move away from the philosophy of Ullmann's algorithm, namely, work with adjacency matrices. The new algorithm is based on the construction and study of frameworks of compared graphs. The article provides a definition and an algorithm for constructing a graph framework, which, in turn, are built on the basis of a previously published iterative algorithm for finding all the shortest paths between two vertices. The wireframe algorithm has asymptotic complexity O ($|V|^2$), it is able to detect graph homomorphism and quantify the similarity of two compared graphs.

Keywords: search algorithm, graph, shortest path, graph comparison, Ullman's algorithm, similarity assessment

Conflict of interests: The author declares no conflict of interest.

For citation: Sysoev V.V. A Framework Similarity Estimation of Unweighted Undirected Graphs. *Modern Information Technologies and IT-Education.* 2022;18(3):655-665. doi: https://doi.org/10.25559/SITITO.18.202203.655-665

Введение

Численная оценка схожести и определение как изоморфизма, так и гомоморфизма графов являются фундаментальными, открытыми задачами как в математической, так и в компьютерных науках. Задача определения изоморфизма, как и задача об определении гомоморфизма, заслуживает отдельного места в области оценки сложности, так как алгоритма, работающего за полиномиальное время, до сих пор не было найдено.

Задача нахождения метрик для оценки схожести графов является сложной, даже несмотря на то, что активно исследуется последние десятилетия. Основой для данного направления является алгоритм Ульмана [1, 2], используемый для определения изоморфизма/гомоморфизма графов, и оценки схожести графов (в различных вариациях). Однако он имеет высокую сложность вычисления, например, если сравниваются два графа G_1 и G_2, и V_1, V_2 – множества вершин в первом и втором графе соответственно, то сложность алгоритма Ульмана будет иметь вид $O(|V_1|^{|V_2|})$[1]. Хотя, существуют работы, которые предлагают использовать параллельное вычисление [3-5], в результате чего скорость работы алгоритма увеличивается до 50%, алгоритм остается сложным вычислимым. Кроме того, алгоритм Ульмана дает ответы на бинарные вопросы: изоморфен ли граф или один из них является подграфом другого и не способен давать численную оценку подобия графов. Для того, чтобы дать численную оценку подобия графов, необходимо провести дополнительные вычисления, которые так же являются сложно вычислимыми. Поэтому данная работа просвещена оценке схожести невзвешенных неориентированных графов на новых принципах, отличными от принципа алгоритма Ульмана, а именно, операциями над матрицами смежности.

Созданный алгоритм дает численную оценку схожести, ответ на вопрос, гомоморфные ли сравниваемые графы.

Суть алгоритма заключается в том, что для каждого графа G_i во множестве графов $G = \{G_1, \dots, G_n\}$, где $G_i =< E_i, V_i >$ – невзвешенный неориентированный граф, $i \in N$, вычисляется каркас $K(G_i) =< E(G_i), V(G_i) >$, с помощью алгоритма итерационного поиска всех кратчайших путей [6].

Каркас графа $K(G_i)$ – это такой подграф графа G_i, который состоит из всех максимальных наикратчайших путей $W = \{W_{i,\alpha}, \dots, W_{i,\delta}\}$, где каждый путь представляет собой граф $W_i = < V(G_i), E(G_i) >$. Алгоритм построения каркаса графа подробно описан в разделе 2.

При попарном сравнении графов из множества графов G сравниваются соответствующие каркасы из множества $K(G) = \{K(G_1), \dots, K(G_n)\}$ и на основе этого вычисляется коэффициент подобия двух графов $J = J(G_1, G_n)$, где $J \in (0,1]$, и, чем больше коэффициент подобия, тем больше графы похожи друг на друга.

При $J = 1$ графы гомоморфные при условии $|V_{i_1}| = |V_{i_2}|$, при $|V_{i_1}| \neq |V_{i_2}|$ граф с меньшим количеством вершин будет подграфом графа с большим количеством вершин.

Сложность представленного алгоритма зависит от алгоритма поиска всех наикратчайших путей [6], который формирует каркас графа, и в худшем случае составляет $O(|V|^2)$. Подробнее о выведении сложности написано разделе 3.

Немаловажным преимуществом алгоритма является тот момент, что, найдя каркасы $K(G)$ для каждого графа из G, можно дать качественную оценку схожести любой паре графов из G практически без затраты дополнительных ресурсов.

Обзор существующих методов

В данной части рассмотрены существующие алгоритмы оценки схожести помимо уже упомянутого во введении алгоритма Ульмана и приведены их преимущества и недостатки.

Существует алгоритм подсчета расстояния редактирования между двумя графам [7, 8], заключающийся в том, что мы последовательно редактируем сравниваемые графы до тех пор, пока они не станут изоморфными. Количество действий редактирования и будет являться численной оценкой схожести графов. Однако, такой подход крайне сложно вычислимый при графах больших размеров, так как это NP - сложная задача, хотя ряд авторов утверждают, что при выполнении математической аппроксимации, удается перевести задачу в – сложную задачу, ценой снижения качества оценки схожести графов.

В алгоритме оценки схожести по общим подграфам [9, 10, 11] авторами заявлена точность алгоритма в 90–95%. Однако, поскольку алгоритм поиска подграфов основан на алгоритме Ульмана, он не может быть быстрым, в силу того, что кроме выполнения самого алгоритма Ульмана, требуется выполнение анализа полученных подграфов.

Алгоритм подсчета размера пересечений между ребрами и вершинами графов [12] является вычислимым за полином времени, но, в то же время, авторами указывается, что в текущем виде алгоритм пока не обладает должной точностью, но у него есть перспективы к улучшению.

Алгоритм подсчета максимального общего подграфа [13, 14, 15] является одним из лучших в плане точности, способным дать качественно оценку схожести графов. Работает быстрее, чем алгоритм оценки схожести по общим подграфам. Однако это алгоритм, несмотря на возросшую скорость работы, все еще позиционирует себя как NP – полная задача.

Существуют и другие алгоритмы, как способные находить изоморфизм сравниваемых графов [16, 17, 18], так и способные находить общие подграфы [19-22]. Несмотря на то, что вышеперечисленные алгоритмы работают эффективнее базового алгоритма Ульмана, они так же сложны в плане вычисления.

Резюмируя, можно составить таблицу, отражающие преимущества и недостатки имеющихся алгоритмов:

[1] Garey M. R., Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. NY : W. H. Freeman and Com-pany, 1979. 340 p.

Т а б л и ц а 1. **Сравнение алгоритмов оценки графов между собой**

T a b l e 1. **Comparison of graph estimation algorithms with each other**

Алгоритм	Возможности алгоритма	Краткая суть работы	Достоинства	Недостатки
Алгоритм Ульмана	Выявляет изоморфизм сравниваемых графов, ответить на вопрос, является ли один граф подграфом другого.	Выполняется путем сравнения матриц смежности сравниваемых графов.	Дает точные ответы о изоморфизме графов и о том, является ли один граф подграфом другого.	Высокая вычислительная сложность, не дает оценку схожести двух графов.
Алгоритм подсчета расстояния редактирования между двумя графам	Выявляет изоморфизм, дает оценку схожести сравниваемых графов.	Путем выполнения операций редактирования на обоих графов, до наступления изоморфизма сравниваемых графов.	Работает, в том числе, и с ориентированными графами.	Оценка изоморфизм графов – вероятностная величина, высокая вычислительная сложность в больших графах, задача NP - типа.
Алгоритм оценки схожести по общим подграфам	Дает оценку схожести сравниваемых графов.	Используя алгоритм Ульмана, находит подграфы в обоих графах. На основе найденных подграфов дает оценку схожести графов.	Высокая заявленная точность алгоритма, 90-95%.	Высокая вычислительная сложность в больших графах, задача NP – типа.
Алгоритм подсчета размера пересечений между ребрами и вершинами графов	Дает оценку схожести сравниваемых графов.	Вычисляет параметры графа (ребра и вершины) по своей методике и дает оценку схожести двух графов.	Быстрота работы.	Низкая точность оценки схожести в больших графах.
Алгоритм подсчета максимального общего подграфа	Дает оценку схожести сравниваемых графов.	Схож с алгоритмом схожести по общим подграфам, оценка деется по максимально общему подграфу.	Хорошая точность алгоритма.	Высокая вычислительная сложность в больших графах, задача NP – типа.
Каркасный алгоритм оценки схожести	Дает оценку схожести сравниваемых графов, выявляет гомоморфизм графов.	Строит каркасы сравниваемых графов, анализирует их свойства и дает заключение по сравниваемым графам.	Хорошая точность алгоритм, быстрота работы.	На конечном этапе оценки, вероятно, есть замечания к полноте результатов работы алгоритма.

Описание алгоритма

Пусть дано несколько невзвешенных неориентированных графов $G = \{G_1, \dots, G_n\}$, где

$G_i =< E_i, V_i >$ – невзвешенный неориентированный граф,

E_i – непустое множество неупорядоченных ребер графа G_i,

V_i – непустое множество вершин графа G_i,

n – количество графов,

$i = \{1 \dots n\},\ i\ \in N$ – индекс графа.

В каждом i – ом графе из G, для каждой j – ой вершины $V_{i,j}$ составляется список итераций $R_{G_{i,j}} = \{R_1, \dots, R_t\}$, где

$R_w \subseteq R_{G_{i,j}}$ – множество вершин из V_i, равноудаленных от $V_{i,w}$ на расстоянии w, $w \in N$,

$V_i = \{V_{i,j}, \dots, V_{i,k}\}$ – множество всех вершин в графе G_i,

t – удаленность от вершины $V_{i,j}$,

k – количество вершин в графе G_i,

$j = \{1 \dots k\},\ j\ \in N$ = индекс вершины в графе G_i.

Далее, определяется максимально возможное расстояние в графе G_i между вершинами t_{max} и создаётся множество пар вершин $V_i^{(R)} = \{[V_{i,\alpha}, V_{i,\beta}]\}$, где V_α, V_β – вершины, входящие в R_w, находящихся на максимальном удалении $t = t_{max}$.

С помощью итерационного алгоритма [6] между каждыми парами в $V_i^{(R)}$ вычисляются кратчайшие пути $W = \{W_{i,\alpha}, \dots, W_{i,\delta}\}$, где каждый путь представляет собой граф $W_i \coloneqq (V(G_i), E(G_i))$, и $\alpha, \beta = \{1 \dots n\}$, $\alpha, \beta \in N, \alpha \neq \beta$ – индексы графов.

На основании вышеописанный результатов для каждого G_i составляется каркас графа $K(G_i) = < E(G_i), V(G_i) >$, где

$V(K) = V(W_1) \cup \dots \cup V(W_m)$ – объединение вершин всех путей W графа G_i, m – количество графов в W.

$E(K) = E(W_1) \cup \dots \cup E(W_l)$ – объединение ребер всех путей W графа G_i, l – количество всех ребер в W.

Далее, для каждого графа G_i на основании советующего каркаса из $K(G_i)$:

- определяется диаметр графа $D(G_i)$ равный расстоянию любого из путей $K(G_i)$, $D(G_i) = t_{max}$,

- определяется количество путей в каркасе каждого графа $k(G_i) = |W(G_i)|$,

- определяется отношение количества вершин в каркасе $|V(K)|$ к общему количеству вершин в графе $|V(G_i)|$, или коэффициент плотности каркаса графа, по формуле:

$$\Delta V(G_i) = \frac{|V(K)|}{|V(G_i)|} \tag{1}$$

Коэффициент подобия J вычисляется при попарном сравнении графов, для этого для каждой пары графов G_1 и G_2 определяем:

- соотношение диаметров графов

$$\Delta D(G_1, G_2) = \frac{MIN(D(G_1), D(G_2))}{MAX(D(G_1) D(G_2))} \tag{2}$$

- соотношение количеств путей графов

$$\Delta k(G_1, G_2) = \frac{MIN(k(G_1), k(G_2))}{MAX(k(G_2), k(G_2))} \tag{3}$$

- соотношения коэффициентов плотности каркасов графов

$$\Delta V(G_1, G_2) = \frac{MIN(\Delta V(G_1) \Delta V(G_2))}{MAX(\Delta V(G_1), \Delta V(G_2))} \tag{4}$$

Коэффициент подобия вычисляется по формуле:

$$J = J(G_1, G_2) = \Delta D(G_1, G_2) \cdot \Delta k(G_1, G_2) \cdot \Delta V(G_1, G_2) \tag{5}$$

Коэффициент подобия отдаленно напоминает поведение вероятностной величины, $J \in (0, 1]$.

В отношении коэффициента подобия J действуют следующие правила:

1. чем ближе коэффициент J подобия к единице тем более похожи графы G_1, G_2 между собой.

2. при значении $J = 1$ графы G_1, G_2:

- гомоморфные, если $|V(G_1)| = |V(G_2)|$, т.е. количество вершин одинаково,

- граф с меньшим количеством вершин является подграфом с большим, то есть, к примеру, если $|V(G_1)| > |V(G_2)|$, то G_2 является подграфом G_1.

Вывод и обоснование сложности алгоритма

Определим асимптотическую сложность [2] по верхнему пределу O [23-25]. Так как основная нагрузка приходится на итерационный алгоритм, который в худшем случае, выдает сложность (G_i – полный граф)[3] $O(|V|^2)$, тогда для худшего случая асимптотическая сложность по верхнему пределу при сравнении двух графов имеет вид G_1, G_2:

$$O(O_{G_1}(|V_1|^2 + |V_1|^2 + 1) + O_{G_2}(|V_2|^2 + |V_2|^2 + 1) + 1) \approx O(|V|^2), \tag{6}$$

где:

Каждое слагаемое $O_{G_i}(|V_i|^2 + |V_i|^2 + 1)$ – асимптотическая сложность работы построения каркасов графов, где

- первое слагаемое – время на построения итераций для каждой вершины,

- второе слагаемое – время работы итерационного алгоритма,

- третье слагаемое – время вычисления характеристик графа.

Последнее слагаемое в общем уравнении (6) – асимптотическая сложность вычисления и сравнения характеристик каркасов графов G_1, G_2, представляющая собой простейшие арифметические операции, независящие от характеристик сравниваемых графов.

$|V|$ – наибольшее количество вершин из сравниваемых графов.

Однако, в реальности (в компьютерных сетях, инфраструктуре, компьютерных играх и т.д.), полные графы встречаются крайне редко, чаще всего графы имеют относительно невысокую плотность, при которой $|E| \ll |V|^2$. Если принять данное допущение, что каждый сравниваемый граф из G удовлетворяет вышеописанному условию, то сложность алгоритма становится:

$$O(|V|), \ \forall G: E_i \ll V_i^2 \tag{7}$$

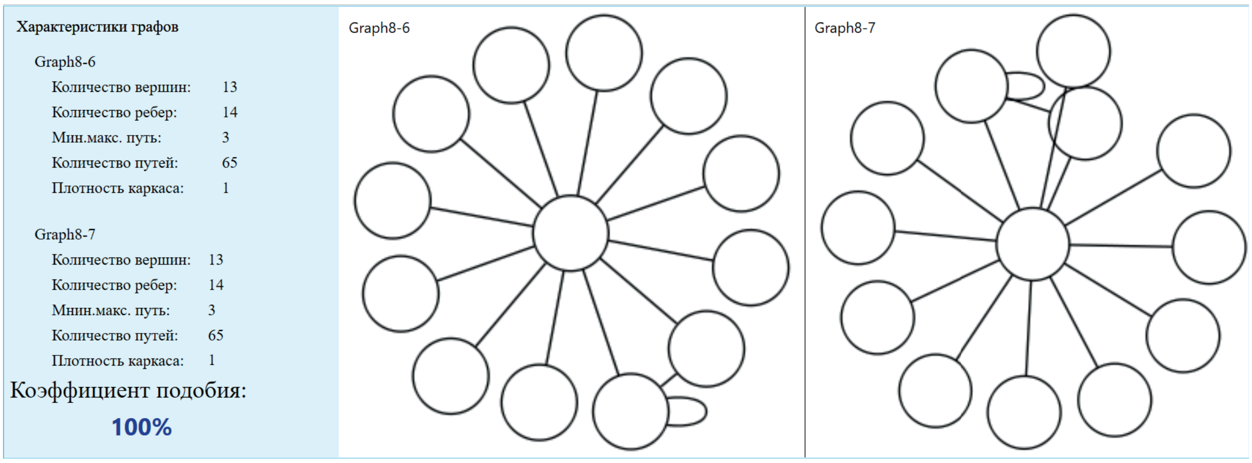
[2] Белоусов А. И., Ткачев С. Б. Дискретная математика. 7-е изд., испр. М. : Изд-во МГТУ им. Н. Э. Баумана, 2021. 704 с. URL: https://www.elibrary.ru/item.asp?id=49992555 (дата обращения: 30.08.2022); McConnell J. Analysis of Algorithms. 2nd ed. Jones & Bartlett Learning, 2007. 451 p.

[3] Harary F., Palmer E. M. Graphical Enumeration. NY : Academic Press, 1973. 218 p. doi: https://doi.org/10.1016/C2013-0-10826-4

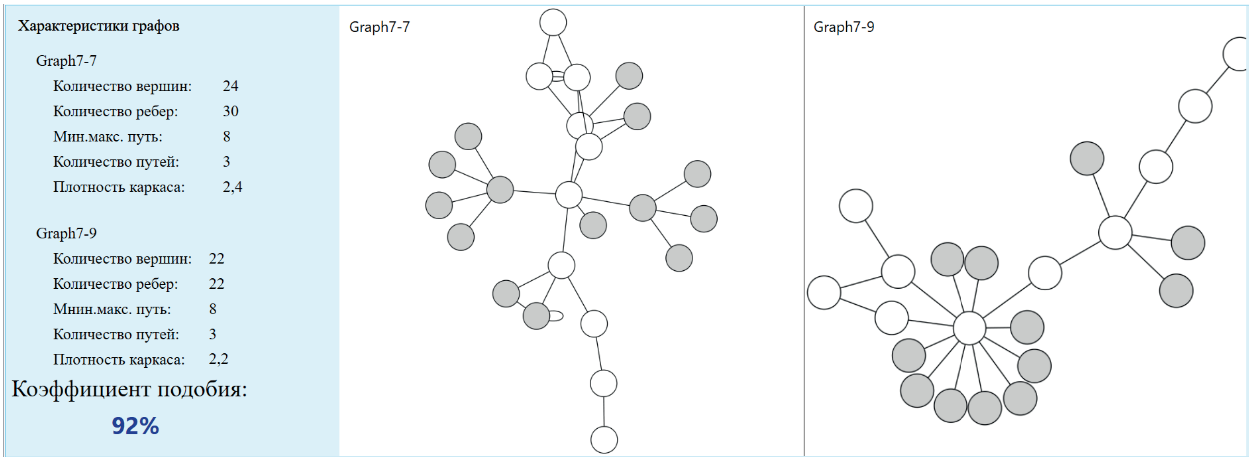
Экспериментальная часть

При проведении эксперимента были выбраны специальные условия для того, чтоб облегчить визуализацию работ данного алгоритма в рамках статьи:
- количество графов должно быть небольшим,
- сами графы должны быть, так же, небольшими,
- все графы должны быть разными по топологии,
- желательно, все графы должны быть взяты из реально существующих систем.

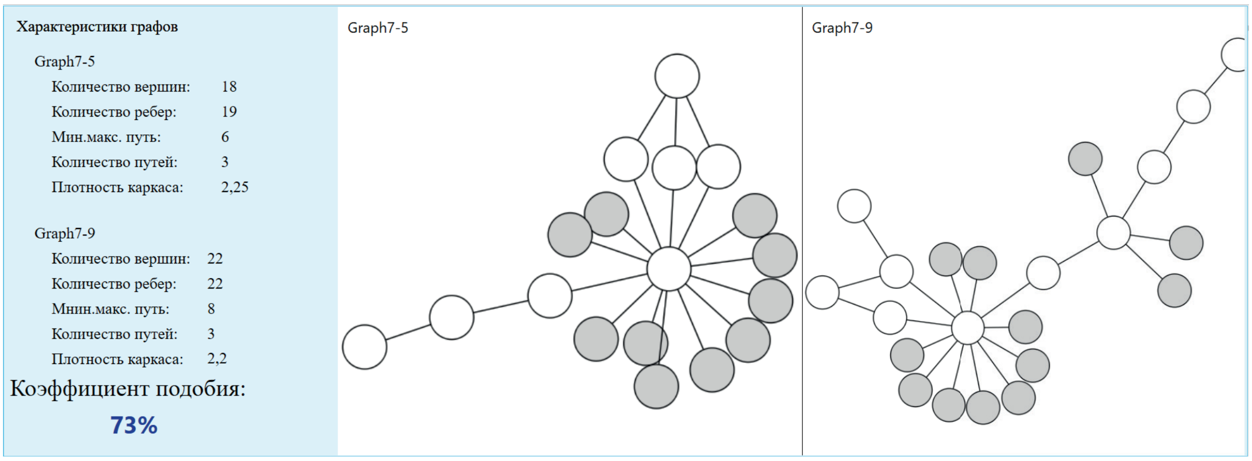
Данные для эксперимента были взяты из рабочей системы по транзакции денежных средств и полностью обезличены.
Так же для данной статьи была написана программа, реализующая алгоритм сравнения графов. Ниже показаны некоторые примеры результата работы программы, реализующей каркасный алгоритм сравнения графов, где белые вершины – вершины, входящие в каркасы соответствующих графов.
На рисунке 1 представлена визуализация сравнения гомоморфных графов. На рисунке 2 сравниваются схожие графы. В данном случае высокая оценка схожести двух графов дается за схожую топологию и количественную характеристику каркасов.
Менее схожие графы представлены на рисунке 3. Из данной визуализации видно, что по мере того, как каркасы графов меньше начинают быть схожими, оценка схожести падает.
На рисунке 4 сравниваются графы с каркасами, оставляющими почти весь граф. Однако, они содержат существенно разное количество вершин, поэтому оценка их схожести не столь высока.
Еще одно подобное сравнение представлено на рисунке 5. Схожая топология каркасов на данной визуализации обуславливает схожесть около 60%.
Для графов, представленных на рисунке 6, высокая оценка схожести обусловлена тем, что сравниваемые графы имеют похожие, «ветвистые» каркасы.
На рисунках 7 и 8 представлена оценка не схожих графов. Данных визуализациях сравниваются пары несхожих графов с кардинально разной структурой. В первом случае (рис. 7) сравниваются графы разного размера. На втором сравнении (рис. 8) показывается, что вне зависимости от размера каркасов сравниваемых графов, оценка схожести так же будет равна 0%.



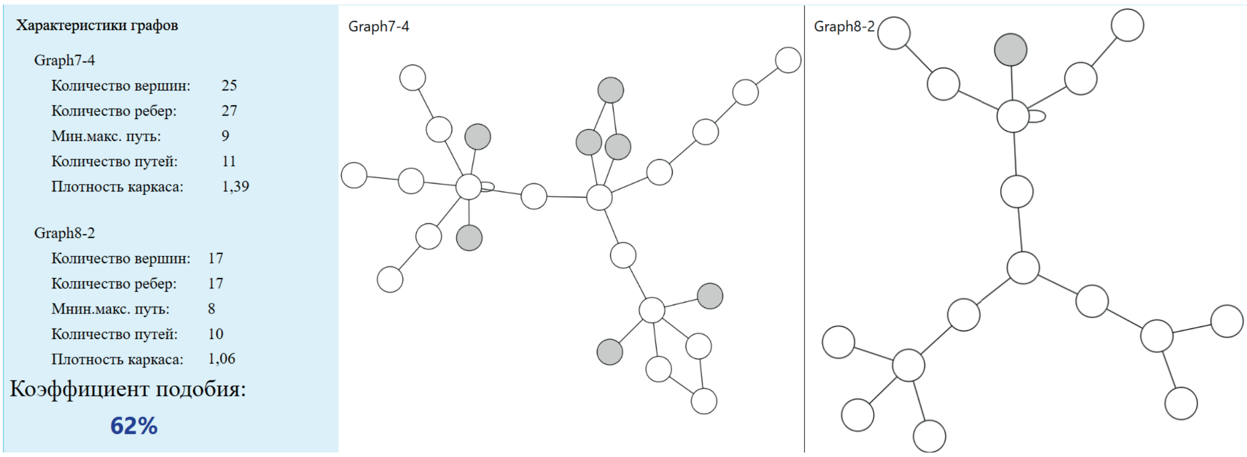
Р и с. 1. Визуализация сравнения гомоморфных графов
F i g. 1. Visualization of comparison of homomorphic graphs



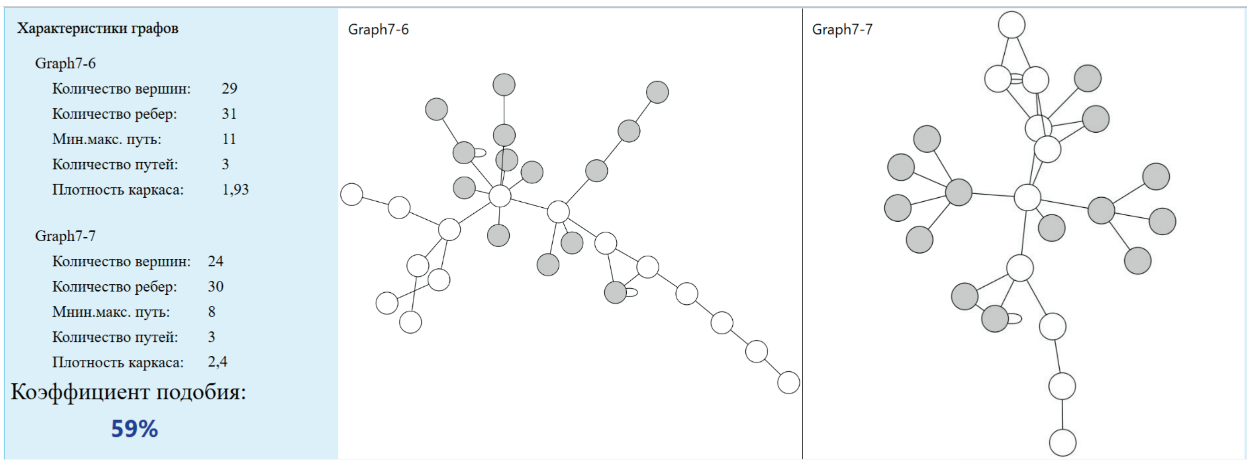
Р и с. 2. Визуализация сравнения схожих графов
F i g. 2. Visualization of comparison of similar graphs



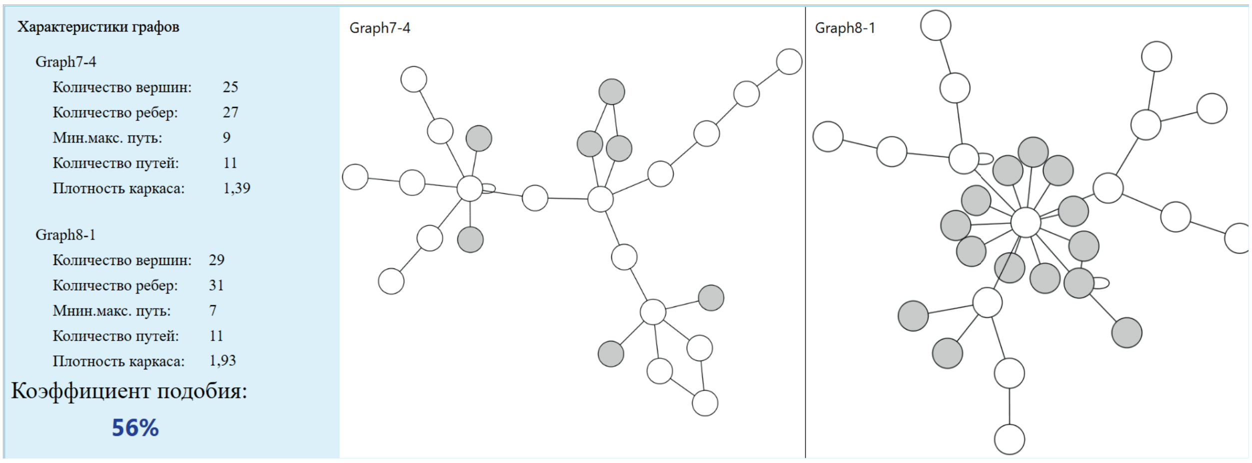
Р и с. 3. Визуализация сравнения графов с похожими подграфами
F i g. 3. Visualization of comparison of graphs with similar subgraphs



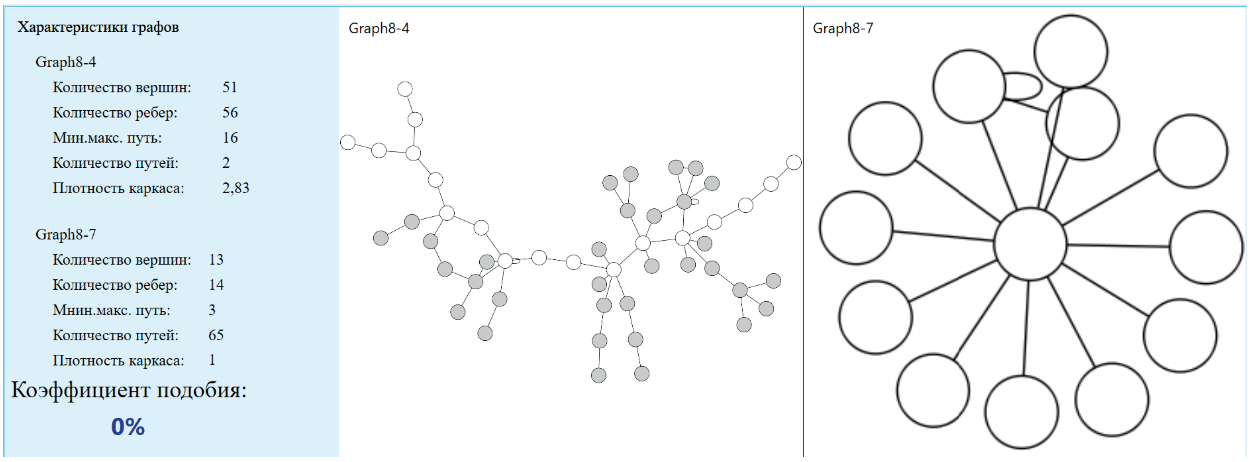
Р и с. 4. Визуализация сравнения графов с каркасами, оставляющими почти весь граф
F i g. 4. Visualization of comparison of graphs with wireframes leaving almost the entire graph



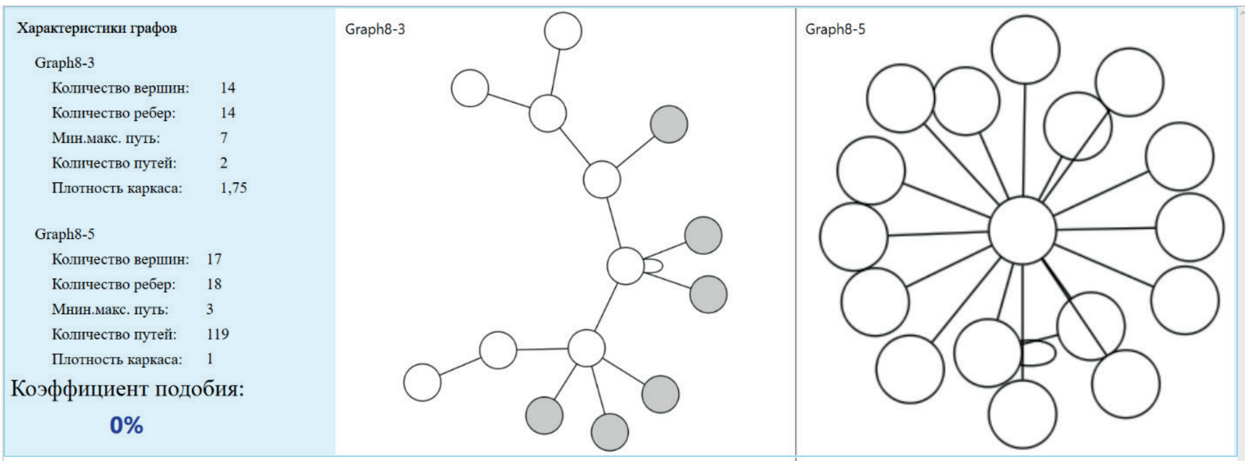
Р и с. 5. Визуализация сравнения графов с общими элементами структуры
F i g. 5. Visualization of comparison of graphs with common structure elements



Р и с. 6. Визуализация сравнения графов с общими элементами структуры
F i g. 6. Visualization of comparison of graphs with common structure elements



Р и с. 7. Визуализация сравнения несхожих графов
F i g. 7. Visualization of comparison of dissimilar graphs



Р и с. 8. Визуализация сравнения несхожих графов
F i g. 8. Visualization of comparison of dissimilar graphs

Как видно из визуализаций экспериментальных данных, алгоритм дает весьма правдоподобное и довольно точную оценку схожести двух графов.

Заключение

В статье проведено исследование современных алгоритмов сравнения графов, обозначены как их преимущества, так и недостатки. Описан новый каркасный алгоритм сравнения невзвешенных неориентированных графов, основанный на построении и сравнении каркасов графов. В том числе, был описан алгоритм построения каркаса графа. Презентуемый алгоритм показал хорошие результаты в ходе экспериментов, имеет хорошую асимптотическую сложность и способен использоваться в различных областях, требующих выявление схожих графовых элементов, как например, мошеннические группы, схожие финансовые транзакции, общие группы лиц, интересов групп лиц и т.д.

Возможным недостатком данного алгоритма является то, что он способен обнаружить только гомоморфизм графов, в то время, как базовый алгоритм Ульмана способен обнаружить изоморфизм сравниваемых графов. Так же, нахождение ответа на вопрос, является один сравниваемый граф подграфом другого не является основной задачей представленного алгоритма, а является его следствием работы.

Список использованных источников

[1] Ullmann J. R. An Algorithm for Subgraph Isomorphism // Journal of the ACM. 1976. Vol. 23, issue 1. P. 31-42. doi: https://doi.org/10.1145/321921.321925

[2] Ullmann J. R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism // ACM Journal of Experimental Algorithmics. 2011. Vol. 15. Article number: 1.6. P. 1.1-1.64. doi: https://doi.org/10.1145/1671970.1921702

[3] Fomin F. V., Kratsch D., Woeginger G. J. Exact (Exponential) Algorithms for the Dominating Set Problem // Graph-Theoretic Concepts in Computer Science. WG 2004. Lecture Notes in Computer Science ; ed. by J. Hromkovič, M. Nagl, B. Westfechtel. Vol. 3353. Berlin, Heidelberg: Springer, 2004. doi: https://doi.org/10.1007/978-3-540-30559-0_21

[4] On scalable parallel recursive backtracking / F. N. Abu-Khzam [и др.] // Journal of Parallel and Distributed Computing. 2015. Vol. 84. P. 65-75. doi: http://doi.org/10.1016/j.jpdc.2015.07.006

[5] An Efficient Graph Isomorphism Algorithm Based on Canonical Labeling and Its Parallel Implementation on GPU / R. Wang [и др.] // 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing. Zhangjiajie, China: IEEE Computer Society, 2013. P. 1089-1096. doi: http://doi.org/10.1109/HPCC.and.EUC.2013.154

[6] Сысоев В. В. Итерационный алгоритм поиска кратчайшего пути в невзвешенном неориентированном графе // Современные информационные технологии и ИТ-образование. 2021. Т. 17, № 3. С. 585-592. doi: https://doi.org/10.25559/SITITO.17.202103.585-592

[7] A Graphtheoretic Approach to Enterprise Network Dynamics / H. Bunke [и др.] // Progress in Computer Science and Applied Logic. Vol. 24. Birkhäuser Boston, MA, 2007. 226 p. doi: https://doi.org/10.1007/978-0-8176-4519-9

[8] Kruskal J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis // Psychometrika. 1964. Vol. 29, issue 1. P. 1-27. doi: https://doi.org/10.1007/BF02289565

[9] Papadimitriou P., Dasdan A., Garcia-Molina H. Web graph similarity for anomaly detection // Journal of Internet Services and Applications. 2010. Vol. 1, issue 1. P. 19-30. doi: https://doi.org/10.1007/s13174-010-0003-x

[10] Kanigalpula S., Naveen S. Subgraph Similarity Search in Large Graphs // The Eighth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2016). IARIA, 2016. P. 84-93. doi: https://doi.org/10.48550/arXiv.1512.05256

[11] Yang X., Qiao H., Liu Z. -Y. An Algorithm for Finding the Most Similar Given Sized Subgraphs in Two Weighted Graphs // IEEE Transactions on Neural Networks and Learning Systems. 2018. Vol. 29, no. 7. P. 3295-3300. doi: https://doi 10.1109/TNNLS.2017.2712794

[12] Graph structure in the Web / A. Broder [и др.] // Computer Networks. 2000. Vol. 33, issue 1-6. P. 309-320. doi: https://doi.org/10.1016/S1389-1286(00)00083-9

[13] Huang X., Lai J., Jennings S. F. Maximum common subgraph: some upper bound and lower bound results // BMC Bioinformatics. 2006. Vol. 7. Article number: S6. doi: https://doi.org/10.1186/1471-2105-7-S4-S6

[14] Москин Н. Д. Метрика для сравнения графов с упорядоченными вершинами на основе максимального общего подграфа // Прикладная дискретная математика. 2021. № 52. С. 105-113. doi: https://doi.org/10.17223/20710410/52/7

[15] Bunke H., Shearer K. A graph distance metric based on the maximal common subgraph // Pattern Recognition Letters. 1998. Vol. 19, issue 3-4. P. 255-259. doi: https://doi.org/10.1016/S0167-8655(97)00179-7

[16] Бойтяков А. А., Егоров Ю. С. Модель передачи данных на основании сравнения близости графов и графовых структур геометрических моделей // Современные наукоемкие технологии. 2019. № 6. С. 20-25. URL: https://elibrary.ru/item.asp?id=38597151 (дата обращения: 30.08.2022).

[17] Сайфуллина Е. Ф. Последовательности сравнения инвариантов графов в задаче проверки изоморфизма // Вектор науки Тольяттинского государственного университета. 2013. № 3(25). С. 87-90. URL: https://elibrary.ru/item.asp?id=21458448 (дата обращения: 30.08.2022).

[18] Тужилкин А. Ю., Захаров А. А. Нахождение соответствий на изображениях с использованием приближенного сравнения графов // Научно-технический вестник Поволжья. 2015. № 4. С. 130-132. URL: https://elibrary.ru/item.asp?id=24115121 (дата обращения: 30.08.2022).
[19] Solving Graph Isomorphism Using Parameterized Matching / J. Mendivelso [и др.] // String Processing and Information Retrieval. SPIRE 2013. Lecture Notes in Computer Science ; ed. by O. Kurland, M. Lewenstein, E. Porat. Vol. 8214. Cham: Springer, 2013. P. 230-242. doi: https://doi.org/10.1007/978-3-319-02432-5_26
[20] A subgraph isomorphism algorithm and its application to biochemical data / V. Bonnici [и др.] // BMC Bioinformatics. 2013. Vol. 14. Article number: S13. doi: https://doi.org/10.1186/1471-2105-14-S7-S13
[21] A Survey on Distributed Graph Pattern Matching in Massive Graphs / S. Bouhenni [и др.] // ACM Computing Surveys. 2021. Vol. 54, no. 2. Article number: 36. doi: https://doi.org/10.1145/3439724
[22] A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs / L. P. Cordella [и др.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Vol. 26, issue 10. P. 1367-1372. doi: https://10.1109/TPAMI.2004.75
[23] Optimization of Subgraph Matching over Knowledge Graph Based on Subgraph Indexing / L. Lv [и др.] // 2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD). Chengdu, China: IEEE Computer Society, 2022. P. 543-546. doi: https://doi.org/10.1109/ICAIBD55127.2022.9820592
[24] Llados J., Marti E., Villanueva J. J. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002. Vol. 23, issue 10. P. 1137-1143. doi: https://doi.org/10.1109/34.954603
[25] Прихожий А. А. Оптимизация размещения данных в иерархической памяти для блочных алгоритмов поиска кратчайших путей // Системный анализ и прикладная информатика. 2021. № 3. С. 40-50. doi: https://doi.org/10.21122/2309-4923-2021-3-40-50

Поступила 30.08.2022; одобрена после рецензирования 23.09.2022; принята к публикации 04.10.2022.

Об авторе:

Сысоев Валентин Валерьевич, главный инженер лаборатории кибербезопасности, Блок «Технологии», ПАО «Сбербанк России» (117312, Российская Федерация, г. Москва, ул. Вавилова, д. 19), **ORCID: https://orcid.org/0000-0002-6157-5815,** Sysoev.V.V@sberbank.ru

Автор прочитал и одобрил окончательный вариант рукописи.

References

[1] Ullmann J.R. An Algorithm for Subgraph Isomorphism. *Journal of the ACM.* 1976;23(1):31-42. doi: https://doi.org/10.1145/321921.321925
[2] Ullmann J.R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *ACM Journal of Experimental Algorithmics.* 2011;15(1.6):1.1-1.64. doi: https://doi.org/10.1145/1671970.1921702
[3] Fomin F.V., Kratsch D., Woeginger G.J. Exact (Exponential) Algorithms for the Dominating Set Problem. In: Hromkovič J., Nagl M., Westfechtel B. (eds.) Graph-Theoretic Concepts in Computer Science. WG 2004. Lecture Notes in Computer Science. Vol. 3353. Berlin, Heidelberg: Springer; 2004. doi: https://doi.org/10.1007/978-3-540-30559-0_21
[4] Abu-Khzam F.N., Daudjee K., Mouawad A.E., Nishimura N. On scalable parallel recursive backtracking. *Journal of Parallel and Distributed Computing*. 2015;84:65-75. doi: http://doi.org/10.1016/j.jpdc.2015.07.006
[5] Wang R., Guo L., Ai C., Li J., Ren M., Li K. An Efficient Graph Isomorphism Algorithm Based on Canonical Labeling and Its Parallel Implementation on GPU. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing. Zhangjiajie, China: IEEE Computer Society; 2013. p. 1089-1096. doi: http://doi.org/10.1109/HPCC.and.EUC.2013.154
[6] Sysoev V.V. Iterative Algorithm for Finding the Shortest Ways in an Unweighted Undirected Graph. *Modern Information Technologies and IT-Education.* 2021;17(3):585-592. (In Russ., abstract in Eng.) doi: https://doi.org/10.25559/SITITO.17.202103.585-592
[7] Bunke H., Dickinson P.J., Kraetzl M., Wallis W.D. A Graphtheoretic Approach to Enterprise Network Dynamics. In: Progress in Computer Science and Applied Logic. Vol. 24. Birkhäuser Boston, MA; 2007. 226 p. doi: https://doi.org/10.1007/978-0-8176-4519-9
[8] Kruskal J.B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika.* 1964;29(1):1-27. doi: https://doi.org/10.1007/BF02289565
[9] Papadimitriou P., Dasdan A., Garcia-Molina H. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications.* 2010;1(1):19-30. doi: https://doi.org/10.1007/s13174-010-0003-x
[10] Kanigalpula S., Naveen S. Subgraph Similarity Search in Large Graphs. In: The Eighth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2016). IARIA; 2016. p. 84-93. doi: https://doi.org/10.48550/arXiv.1512.05256

[11] Yang X., Qiao H., Liu Z. -Y. An Algorithm for Finding the Most Similar Given Sized Subgraphs in Two Weighted Graphs. *IEEE Transactions on Neural Networks and Learning Systems*. 2018;29(7):3295-3300. doi: https://doi 10.1109/TNNLS.2017.2712794
[12] Broder A., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., Wiener J. Graph structure in the Web. *Computer Networks*. 2000;33(1-6):309-320. doi: https://doi.org/10.1016/S1389-1286(00)00083-9
[13] Huang X., Lai J., Jennings S.F. Maximum common subgraph: some upper bound and lower bound results. *BMC Bioinformatics*. 2006. Vol. 7. Article number: S6. doi: https://doi.org/10.1186/1471-2105-7-S4-S6
[14] Moskin N.D. Metric for comparing graphs with ordered vertices based on the maximum common subgraph. *Applied Discrete Mathematics.* 2021;(52):105-113. (In Russ., abstract in Eng.) doi: https://doi.org/10.17223/20710410/52/7
[15] Bunke H., Shearer K. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters.* 1998;19(3-4):255-259. doi: https://doi.org/10.1016/S0167-8655(97)00179-7
[16] Boytyakov A.A., Egorov Yu.S. A data transfer model based on comparison of proximity graphs and graph structures of geometric models. *Modern high technologies.* 2019;(6):20-25. Available at: https://elibrary.ru/item.asp?id=38597151 (accessed 04.08.2022). (In Russ., abstract in Eng.)
[17] Sayfullina E.F. Consistent comparison of graph invariants in the problem of verification of isomorphism. *Science Vector of Togliatti State University*. 2013;(3):87-90. Available at: https://elibrary.ru/item.asp?id=21458448 (accessed 04.08.2022). (In Russ., abstract in Eng.)
[18] Tuzhilkin A.Yu., Zakharov A.A. Finding matches in the images using inexact comparison of graphs. *Scientific and Technical Volga region Bulletin.* 2015;(4):130-132. Available at: https://elibrary.ru/item.asp?id=24115121 (accessed 04.08.2022). (In Russ., abstract in Eng.)
[19] Mendivelso J., Kim S., Elnikety S., He Y., Hwang S., Pinzon Y. Solving Graph Isomorphism Using Parameterized Matching. In: Kurland O., Lewenstein M., Porat E. (eds.) String Processing and Information Retrieval. SPIRE 2013. Lecture Notes in Computer Science. Vol. 8214. Cham: Springer; 2013. p. 230-242. doi: https://doi.org/10.1007/978-3-319-02432-5_26
[20] Bonnici V., Giugno R., Pulvirenti A., Shasha D., Ferro A. A subgraph isomorphism algorithm and its application to biochemical data. *BMC Bioinformatics*. 2013. Vol. 14. Article number: S13. doi: https://doi.org/10.1186/1471-2105-14-S7-S13
[21] Bouhenni S., Yahiaoui S., Nouali-Taboudjemat N., Kheddouci H. A Survey on Distributed Graph Pattern Matching in Massive Graphs. *ACM Computing Surveys*. 2021;54(2):36. doi: https://doi.org/10.1145/3439724
[22] 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;26(10):1367-1372. doi: https://10.1109/TPAMI.2004.75
[23] Lv L., Liu J., Li Q., Li J. Optimization of Subgraph Matching over Knowledge Graph Based on Subgraph Indexing. In: 2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD). Chengdu, China: IEEE Computer Society; 2022. p. 543-546. doi: https://doi.org/10.1109/ICAIBD55127.2022.9820592
[24] Llados J., Marti E., Villanueva J.J. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002;23(10):1137-1143. doi: https://doi.org/10.1109/34.954603
[25] Prihozhy A.A. Optimization of data allocation in hierarchical memory for blocked shortest paths algorithms. *System analysis and applied information science*. 2021;(3):40-50. doi: https://doi.org/10.21122/2309-4923-2021-3-40-50

Submitted 30.08.2022; approved after reviewing 23.09.2022; accepted for publication 04.10.2022.

About the author:

Valentin V. Sysoev, Senior engineer of the Cybersecurity Laboratory, PJSC “Sberbank of Russia” (19 Vavilov St., Moscow 117312, Russian Federation), **ORCID: https://orcid.org/0000-0002-6157-5815,** Sysoev.V.V@sberbank.ru

The author has read and approved the final manuscript.