DOI: 10.25559/SITITO.020.202401.48-57
УДК 004.021

Оригинальная статья

Поиск схожих подграфов в невзвешенном неориентированном графе посредством вычисления изоморфных путевых наборов

В. В. Сысоев[1, 2*], А. Ю. Быков[2]
[1] Публичное акционерное общество «Сбербанк России», г. Москва, Российская Федерация
Адрес: 117312, Российская Федерация, г. Москва, ул. Вавилова, д. 19
[*] valsus88@mail.ru
[2] ФГБОУ ВО «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)», г. Москва, Российская Федерация
Адрес: 105005, Российская Федерация, г. Москва, ул. 2-я Бауманская, д. 5, к. 1

Аннотация
Тема поиска подграфов в суперграфе является до сих пор актуальной и фундаментальной задачей. Так как графы являются наиболее удачными моделями, как для визуализации, так и для обработки сложных взаимосвязей, представленных в виде семантических сетей, как например, исходный код программ. Одной из важнейших задач анализа таких моделей данных является, задача поиска схожих подграфов. Существующие на данный момент методы, предложения и подходы, базируются на поиске изоморфных подграфов, в том числе, в виде промежуточных результатов, что не дает возможность находить подграфы по общей части с искомым графом в виде частичного графа. Представленный алгоритм работает на основе построения и анализа путевых наборов графов, по оригинальным алгоритмам построения путевого каркаса и итерационного алгоритма поиска наикратчайших путей между двумя вершинами и находит схожие подграфы в суперграфе , выделяя общую часть с искомым графом в виде частичного графа , которая, в конечном счете, проходит проверку на изоморфизм.

Ключевые слова: сравнение графов, схожие графы, изоморфизм графов, кратчайший путь, оценка схожести

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Сысоев В. В., Быков А. Ю. Поиск схожих подграфов в невзвешенном неориентированном графе посредством вычисления изоморфных путевых наборов // Современные информационные технологии и ИТ-образование. 2024. Т. 20, № 1. С. 48-57. https://doi.org/10.25559/SITITO.020.202401.48-57

Original article

Searching for Similar Subgraphs in an Unweighted Undirected Graph by Computing Isomorphic Path Sets

V. V. Sysoev[a,b*], A. Yu. Bykov[b]
[a] Sberbank of Russia, Moscow, Russian Federation
Address: 19 Vavilova St., Moscow 117312, Russian Federation
[*] valsus88@mail.ru
[b] Bauman Moscow State Technical University, Moscow, Russian Federation
Address: 5, 2nd Baumanskaya St., building 2, Moscow 105005, Russian Federation

Abstract
The topic of subgraph search in a supergraph remains relevant and fundamental. Graphs are highly successful models for both visualization and handling complex relationships, such as those represented in semantic networks or even source code of programs. One of the most important tasks in analyzing such data models is the problem of finding similar subgraphs. Existing methods and approaches currently rely on searching for isomorphic subgraphs, including intermediate results, which does not allow for finding subgraphs based on a common part with the target graph in the form of a partial graph. The presented algorithm operates based on constructing and analyzing path sets of graphs, using original algorithms for constructing a path skeleton and an iterative algorithm for finding the shortest paths between two vertices. It identifies similar subgraphs in the supergraph by extracting the common part with the target graph in the form of a partial graph, which ultimately undergoes an isomorphism check.

Keywords: graphs comparison, graph similarity, isomorphism of graphs, shortest path problem, similarity assessment

Conflict of interests: The authors declare no conflict of interest.

For citation: Sysoev V.V., Bykov A.Yu. Searching for Similar Subgraphs in an Unweighted Undirected Graph by Computing Isomorphic Path Sets. *Modern Information Technologies and IT-Education.* 2024;20(1):48-57. https://doi.org/10.25559/SITITO.020.202401.48-57

Введение

Графы являются наиболее удачными моделями как для визуализации, так и для обработки сложных взаимосвязей, таких как социальные сети [1], объекты сетевой инфраструктуры [2, 3] и т.д. Соответствующие наборы данных сильно различаются по ключевым характеристикам, таки как количество графов в наборе данных, количество узлов в графе, средняя плотность графов, количество параметров вершин и пр.

Одной из главных задач в таких системах является поиск в суперграфе G подграфов F^Q, содержащих искомый граф Q. Нативный способ выполнить эту задачу – выполнить поиск изоморфизма подграфов q во всех графах в наборе данных G. Данный алгоритм был придуман Дж. Ульманом в 1976 году [4], впоследствии он существенно доработал свой алгоритм [5]. Данный подход получил дальнейшее развитие в работах многих других авторов, например, Корделлы [6] (алгоритм VF2), Бонници [7] (алгоритм RI).

К тому же, как показала практика, более актуальной темой является поиск схожих подграфов. Он нужен, например, для анализа структурных данных (web-документов) и знаний, представленных семантическими сетями[1], анализа исходного кода программ [8-11].

Существующие модели нахождения схожих подграфов, как, например, базирующиеся на нахождении максимально общего подграфа, или же, на подсчете расстояния редактирования между двумя графами[2]. Существуют, так же, оригинальные способы нахождения схожих подграфов, основанные на вычислении расположения цепей (кластеров) подграфов в суперграфе, такой подход работает только если графы имеют вид древоподобной структуры [12-14].

Так же, существуют методы опосредованного оценивания сходства структур на основе сопоставления инвариантов, различных дескрипторов[3] [15, 16]. Однако, данные методы хорошо себя показали только в химии и биологии, поскольку для их применения нужны выделяемые дескрипторы (например, ребро в виде ковалентной неполярной связи), которых в графах может и не быть [17-19].

Существуют другие методы, например, считающие евклидово расстояние между группой вершин[4], алгоритмы, основанный на поиске общих частей в графах и др. [20, 21]

Общим недостатком всех вышеописанных методов и алгоритмов нахождения схожих подграфов, является то, что все они или требуют нахождения изоморфных подграфов в промежуточных результатах, или требуют наличия явных дескрипторов, что не дает возможность находить схожие подграфы в виде частичного графа.

В данной статье ставится цель описания алгоритма выявления схожих подграфов F^Q в суперграфе G искомого графа Q, по новому методу, предусматривающему вычисление его путевых наборов. Алгоритм не предполагает вычисления изоморфизма подграфов, а основан на построении путевых каркасов [22] и вычислении наикратчайших путей между вершинами [23], и дает возможность выделить общие части в обоих графах в виде частичного графа [24, 25].

Математическая постановка задачи

Дано следующее:

$G = < E^G, V^G >$ - невзвешенный неориентированный суперграф, где

E^G – множество ребер суперграфа;

V^G – множество вершин суперграфа;

$Q = < E^Q, V^Q >$ - невзвешенный неориентированный искомый граф, где

E^Q – множество ребер искомого графа;

V^Q – множество вершин искомого графа;

Необходимо найти:

Пусть суперграф G можно представить как множество составляющих его подграфов $F^G = \{f_1^G, \ldots, f_n^G\}$, $n \in \mathrm{N}$ - количество всех возможных подграфов в суперграфе G. Необходимо найти подграфы $F^G(Q) \subseteq F^G$, $F^G(Q) = \{f_1^G(Q), \ldots, f_m^G(Q)\}$, которые имеют схожесть с искомым графом Q виде частичного графа: $f_j^G(Q) \sim Q$, где $m \in \mathrm{N}$– количество схожих подграфов в $F^G(Q)$, $j \in \{1,2, \ldots, \mathrm{m}\}$.

Описание алгоритма

Алгоритм поиска называется алгоритмом поиска путевых наборов. Алгоритм поиска схожего подграфа основан, отчасти, на алгоритме схожести двух невзвешенных неориентированных графов [22]. Дадим несколько определений, которые будут использоваться в дальнейшем:

Определение 1. *Наикратчайшие путевые наборы (НПН)* $W^H = \{W_1^{v_i v_j}, \ldots, W_k^{v_i v_j}\}$ – множество путей в произвольном графе H, состоящие из наикратчайших путей между всеми его двумя произвольными v_i и v_j, отличными друг от друга $v_i \neq v_j$, вершинами Алгоритм поиска всех наикратчайших путей подробно описан в статье [23].

Определение 2. *Максимальный путевой каркас (МПК) графа* K^H – это подграф графа H, который образован наикратчайшими путевыми наборами *(НПН)* между вершинами максимальной длины W_d^H. Алгоритм построения *МПК* графа подробно описан в разделе 2 статьи [22].

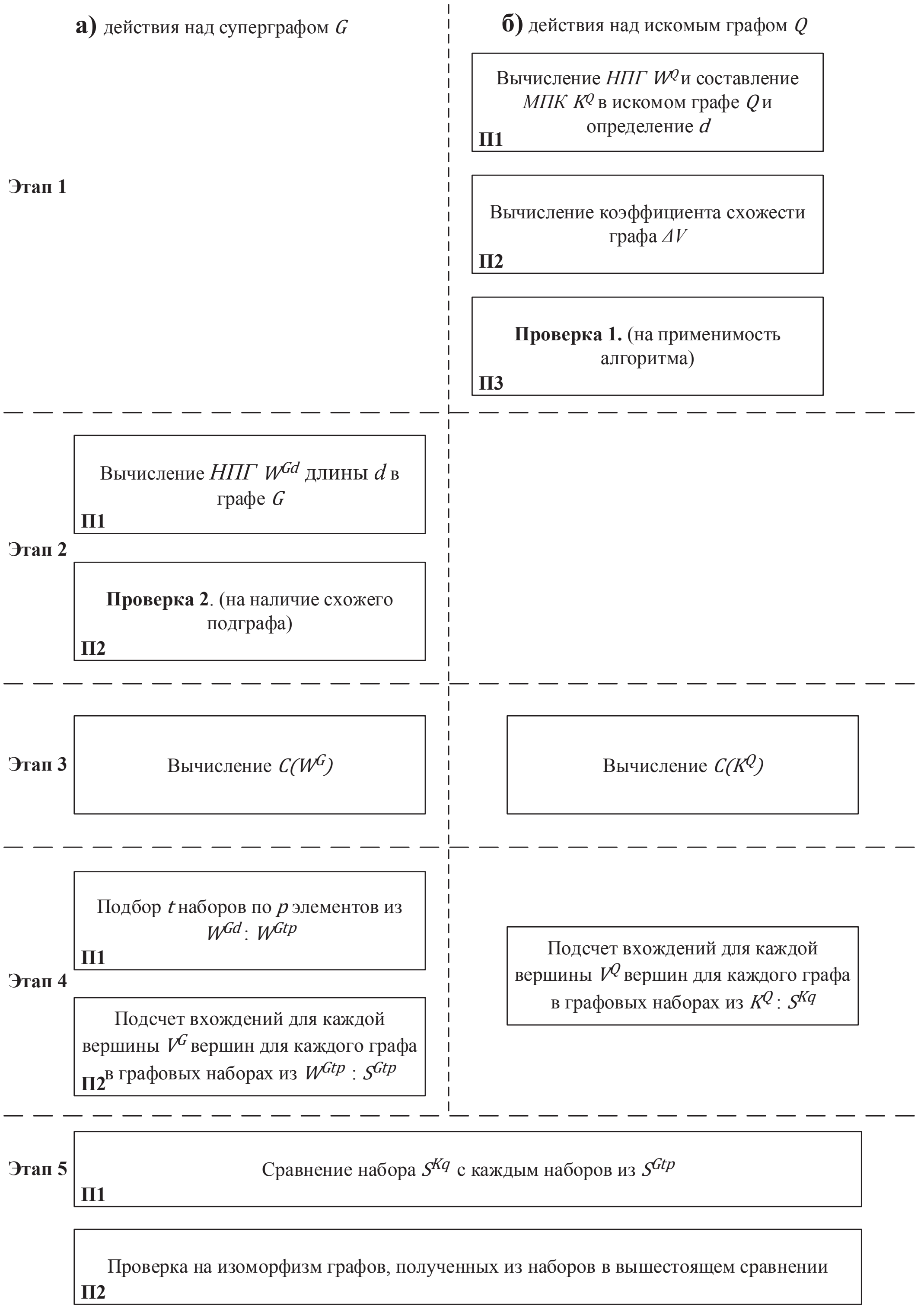
[1] Финн В. К. Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // ДСМ-метод автоматического порождения гипотез : логические и эпистемологические основания ; под общ. ред. О. М. Аншакова. М. : URSS, 2009. С. 10-50. EDN: RCXCZZ

[2] Bunke H., Dickinson P. J., Kraetzl M., Wallis W. D. A Graph-Theoretic Approach to Enterprise Network Dynamics (Progress in Computer Science and Applied Logic). Birkhauser, 2006.

[3] Varmuza K. Chemometrics in PracticalA applications. Rijeka, Croatia: InTech, 2012. 326 p.

[4] Саргсян С. С. Методы поиска клонов кода и семантических ошибок на основе семантического анализа программы : дис. … канд. физ.-мат. наук. Москва, 2016. 103 с. EDN: EMZWJY

Схематически, работу алгоритма можно отобразить на рисунке 1:



Р и с. 1. Схематическое изображение этапов работы алгоритма
F i g. 1. Schematic representation of the stages of the algorithm

Источник: здесь и далее в статье все рисунки составлены авторами.
Source: Hereinafter in this article all figures were drawn up by the authors.

Рассмотрим этапы работы алгоритма поиска подробнее:

Этап 1б:

П1. Для искомого графа Q составляем *МПК* графа $K^K = < E^K, V^K >$. Для этого вычисляются кратчайшее пути до каждой вершины в искомом подграфе Q:

$$W^Q = \cup_{\substack{v_i^Q \in V^Q \\ v_j^Q \in V^Q \\ v_i^Q \neq v_j^Q}} W^Q\left(v_i^Q, v_j^Q\right) = \{w_1^Q, \dots, w_l^Q\} \tag{1}$$

где $l \in \mathrm{N}$ – количество всех возможных наикратчайших путей в искомом графе Q. Среди множества W^Q выбираем такие путевые наборы, которые имеют наибольшую длину $d = \max_{k=1\dots l} D\left(w_k^Q\right)$, обозначим такие путевые наборы $w_i^Q(d)$, i – индекс путевого набора, все такие путевые наборы длины d образуют граф:

$$K^Q = \cup_{i=1,\dots,p} w_i^Q(d) \tag{2}$$

где $p \in \mathrm{N}$ – количество всех возможных НПН в искомом графе Q с длиной пути d, $D(H)$ – функция вычисления длины пути графа H, $K^Q \subset W^Q$.

П2. В силу того, что наша задача заключается в выделении общей части в обоих графах в виде частичного графа, то степень схожести следует считать по отношению количества вершин искомого графа Q, и его количества вершин в *МПК* K^Q. Тогда коэффициент схожести будет считаться по следующей формуле:

$$\Delta V = \frac{|V^K|}{|V^Q|} \tag{3}$$

Таким образом, находимые алгоритмом частичные графы F^Q, будут иметь степень схожести, равный коэффициенту ΔV.

Проверка 1. Коэффициент схожести между искомым графом и его *МПК* может быть очень низким, поэтому алгоритм применяется только, если коэффициент схожести имеет значение $\Delta V > 0.5$.

То есть, граф, образованный *МПК* K^Q, должен содержать не менее половины вершин искомого графа Q.

Этап 2а

Для каждой вершины v_i^G в суперграфе G вычисляются наикратчайшие путевые наборы (*НПН*) $w_r^G(v_i^G, v_j^G)$ до каждой вершины v_j^G, где $v_i^G, v_j^G \in V^G$, $v_i^G \neq v_j^G$. Из полученных *НПН* выбираем только те путевые наборы, которые имеют длину пути $D(w_r^G) = d$:

$$W^G(d) = \cup_{\substack{v_i^G \in V^G \\ v_j^G \in V^G \\ v_i^G \neq v_j^G \\ D(w_r^G) = d}} W^G\left(v_i^G, v_j^G\right) = \{w_1^G(d), \dots, w_k^G(d)\} \tag{4}$$

где $k \in \mathrm{N}$– количество всех возможных наикратчайших путей в суперграфе G длины d. Так же, очевидно, что $W^G \subseteq F^G$.

Проверка 2. На этом этапе, так же, проводиться проверка суперграфа G на наличие схожего подграфа Q. На этом этапе дается ответ на вопрос: возможно ли наличие в супеграфе схожих с искомым графом, подграфов, до проведения основных вычислений. Если наборов из W^G, имеющих длину путей d нет $|W^{Gd}| = 0$, то тогда схожих графов, соответственно, нет.

Этап 3а

Для каждого из НПН из $W^G(d)$ определяется количество путей С длины d между двумя вершинами

$$\mathrm{C}(W^G(d)) = \{\mathrm{c}(w_1^G(d)), \dots, \mathrm{c}(w_k^G(d))\} \tag{5}$$

Этап 3б

Для каждого из *НПН* из *МПК* K^Q определяется количество путей С:

$$\mathrm{C}(K^Q) = \left\{\mathrm{c}\left(k_1^Q\right), \dots, \mathrm{c}\left(k_p^Q\right)\right\} \tag{6}$$

Этап 4а:

П.1 Из всех *НПН* W^{Gd} подбираем t таких наборов $w_{p_1}^G$ по p элементов в каждом, какие удовлетворяют условию $C(K^Q) = C(W^G)$, составляя таким образом массив из наборов:

$$W^{Gtp} = \begin{Bmatrix} w_{P_1}^G \\ \vdots \\ w_{P_t}^G \end{Bmatrix} = \begin{Bmatrix} v_{11}^G & \dots & v_{1p}^G \\ \vdots & & \vdots \\ v_{t1}^G & \dots & v_{tp}^G \end{Bmatrix} \tag{7}$$

где v_{ij}^G – вершина из графа G, входящая в набор $w_{P_i}^G$, $p \in \mathrm{N}$– количество всех возможных *МПК* в искомом графе Q, $t \in \mathrm{N}$ – количество найденных наборов, удовлетворяющих условию из п.2).

П.2 Считаем, для каждой вершины из V^G для каждого графа, образованного *НПН* $w_{P_i}^G$, $i = \{0,1,\dots,p-1\}$ сколько раз она присутствует в выбранных графовых наборах:

$$S^{Gtp} = \begin{Bmatrix} S^{w_{P_1}^G} \\ \vdots \\ S^{w_{P_t}^G} \end{Bmatrix} = \begin{Bmatrix} \{s^{v_i^G}, s^{v_j^G}, \dots s^{v_{K_1}^G}\}_1 \\ \vdots \\ \{s^{v_i^G}, s^{v_j^G}, \dots s^{V_{K_t}^G}\}_t \end{Bmatrix} \tag{8}$$

где $s^{v_i^G}$ – количество присутствий вершины v_i^G в наборе $S^{w_{P_i}^G}$, K_i – количество вершин из V^G в *НПН* $W_{P_i}^{v_i^G v_j^G}$.

Этап 4б

Для каждой вершины из V^K в графе, образованном *МПК* K^Q определяем, сколько раз она присутствует в графовых наборах k_i^Q, $i = \{0,1,\dots,p-1\}$. В результате получим набор количеств вхождений произвольных вершин в *МПК* в произвольном порядке, где позиция вхождения не имеет значения:

$$S^{KQ} = \{s^{v_i^K}, s^{v_j^K}, \dots s^{v_M^K}\} \tag{9}$$

где M – количество вершин в *МПК* K^Q.

Этап 5

На данном этапе проводится сравнение полученных результатов. Граф, образованный из *НПН* $w_{P_i}^G$ будет схожим с графом, образованным и *МПК* K^Q в том случае, если :

П1.

I. $k \neq 0$ – количество *НПН* удовлетворяющих по диаметру *МПК* не нулевое.

II. $p \neq 0$ – количество наборов *НПН* удовлетворяющих по количеству путей в них, не нулевое.

III. Существуют такие наборы коэффициентов присутствия вершин, которые при произвольном расположении (перестановке) в S^{Gtp}, такие, что $S^{w_{P_i}^G} = S^{KQ}$, $i = \{0,1,\dots,t-1\}$.

П2.
Проверяем на изоморфизм графы, образованные путевыми наборами, лежащими в основе $S^{w^G_{P_i}}$ и S^{K^Q}:

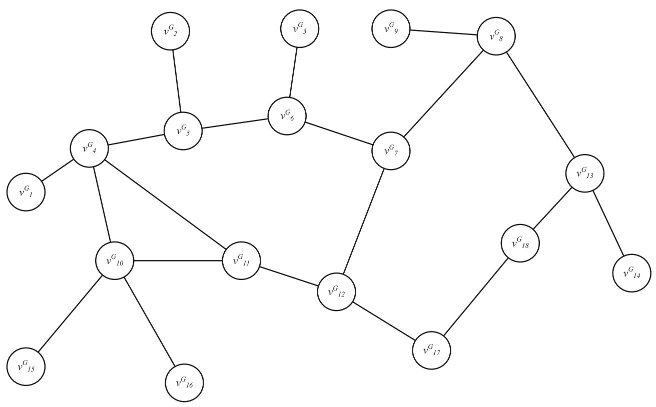
$$\left. \begin{array}{l} w^G_{P_i} \to \mathrm{H}^{Sw} \\ K^Q \ \to \mathrm{H}^{SK} \end{array} \right| \mathrm{H}^{Sw} \simeq \mathrm{H}^{SK} \tag{10}$$

При выполнении вышеописанных условий граф, образованный *НПН* $W^{v^G_I v^G_{II}}_{P_i}$ и граф, образованный *МПК* K^Q, будут схожими:

$$w^G_{P_i} \sim K^Q \ if{:} \left| \begin{array}{c} k \neq 0 \\ p \neq 0 \\ \in S^{w^G_{P_i}} = S^{KQ} \\ \mathrm{H}^{Sw} \simeq \mathrm{H}^{SK} \end{array} \right. \tag{11}$$

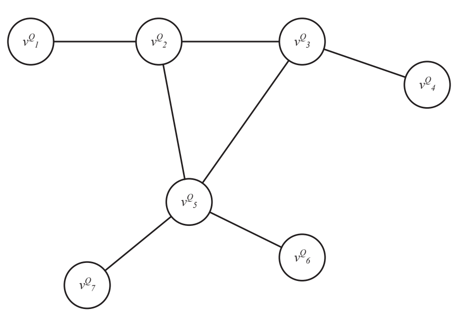
Таким образом, учитывая, что $W^G \subset F^G$ и $K^Q \subset W^Q$, при том, что все множество вершин и ребер W^Q, представляют собой граф Q можно сказать, что граф, образованный всеми вершинами и ребрами $w^G_{P_i}$, является подграфом из множества всех подграфов F^G суперграфа G, и он схож с искомым графом Q, так как Q имеет общий изоморфный граф, образованный *МПК* K^Q.

Пример работы алгоритма
Пусть дан произвольный суперграф G:



Р и с. 2. Схема суперграфа G
F i g. 2. Supergraph scheme G

И искомый подграф Q:



Р и с. 3. Схема искомого графа Q
F i g. 3. Scheme of the required graph Q

Этап 1б:

П1.
В искомом графе Q вычисляем W^Q:

$$W^Q = \{(v^Q_1 \to v^Q_2), (v^Q_1 \to v^Q_2 \to v^Q_5), (v^Q_1 \to v^Q_2 \to v^Q_3), \dots, (v^Q_1 \to v^Q_2 \to v^Q_4 \to v^Q_4)\} \tag{12}$$

Очевидно, что максимальная длина путей в графе Q, $d = 3$. Из W^Q отбираем такие пути, длина которых равна d:

$$K^Q = \left\{ \begin{array}{l} \left[(v^Q_1 \to v^Q_2 \to v^Q_5 \to v^Q_6)\right]^{v^Q_1 v^Q_6}_{D=3}, \\ \left[(v^Q_1 \to v^Q_2 \to v^Q_3 \to v^Q_4)\right]^{v^Q_1 v^Q_4}_{D=3}, \\ \left[(v^Q_1 \to v^Q_2 \to v^Q_5 \to v^Q_7)\right]^{v^Q_1 v^Q_7}_{D=3}, \\ \left[(v^Q_7 \to v^Q_5 \to v^Q_3 \to v^Q_4)\right]^{v^Q_7 v^Q_4}_{D=3}, \\ \left[(v^Q_4 \to v^Q_3 \to v^Q_5 \to v^Q_6)\right]^{v^Q_4 v^Q_6}_{D=3} \end{array} \right\} \tag{13}$$

Всего количество путей в наборе $t = 5$.

П2.
Вычислим степень схожести найденного подграфа:

$$\Delta V = \frac{7}{7} = 1 \tag{14}$$

Проверка 1.
Т.к. $\Delta V > 0.5$, то применение данного алгоритма будет оправдано.

Этап 2а:
В суперграфе G вычисляем W^G:

$$W^G = \left\{ \begin{array}{c} \left[\begin{array}{c} (v^G_1 \to v^G_4 \to v^G_5 \to v^G_6 \to v^G_7), \\ (v^G_1 \to v^G_4 \to v^G_{11}\ v^G_{12}\ v^G_7) \end{array}\right]^{v^G_1 v^G_7}, \dots, \\ \left[\begin{array}{c} (v^G_7 \to v^G_8 \to v^G_{13} \to v^G_{18}), \\ (v^G_7 \to v^G_{12} \to v^G_{17} \to v^G_{18}) \end{array}\right]^{v^G_7 v^G_{18}} \end{array} \right\} \tag{13}$$

Выбираем такие наборы путей из W^G, которые удовлетворяют условию $D(w_i) = 3$

$$W^{Gd} = \{[(v^G_7 \to v^G_8 \to v^G_{13} \to v^G_{18}), (v^G_7 \to v^G_{12} \to v^G_{17} \to v^G_{18})]^{v^G_7 v^G_{18}}_{D=3}, \dots, [(v^G_{15} \to v^G_{10} \to v^G_4 \to v^G_1)]^{v^G_{15} v^G_1}_{D=3}, \ [(v^G_{16} \to v^G_{10} \to v^G_4 \to v^G_1)]^{v^G_{16} v^G_1}_{D=3}\} \tag{14}$$

Проверка 2.
Т.к. $|W^{Gd}| > 0$, то наличие в супеграфе G схожих с искомым графом Q, подграфов возможно.

Этап 3а:
Для всех W^G, вычисляем $\mathrm{C}(w^G_i)$:

$$W^G = \left\{ \begin{array}{c} \left[\begin{array}{l} (v^G_1 \to v^G_4 \to v^G_5 \to v^G_6 \to v^G_7), \\ (v^G_1 \to v^G_4 \to v^G_{11} \to v^G_{12} \to v^G_7) \end{array}\right]^{v^G_1 v^G_7}_{\substack{D=4 \\ C=2}}, \dots, \\ \left[\begin{array}{l} (v^G_7 \to v^G_8 \to V^G_{13} \to v^G_{18}), \\ (v^G_7 \to v^G_{12} \to v^G_{17} \to v^G_{18}) \end{array}\right]^{v^G_7 v^G_{18}}_{\substack{D=3 \\ C=2}} \end{array} \right\} \tag{15}$$

Этап 3б:

Для всех K^Q вычисляем $C(k_i^Q)$:

$$K^Q = \left\{\begin{matrix} [(v_1^Q \to v_2^Q \to v_5^Q \to v_6^Q)]_{\substack{D=3\\ C=1}}^{v_1^Q v_6^Q}, \\ [(v_1^Q \to v_2^Q \to v_3^Q \to v_4^Q)]_{\substack{D=3\\ C=1}}^{v_1^Q v_4^Q}, \\ [(v_1^Q \to v_2^Q \to v_5^Q \to v_7^Q)]_{\substack{D=3\\ C=1}}^{v_1^Q v_7^Q}, \\ [(v_7^Q \to v_5^Q \to v_3^Q \to v_4^Q)]_{\substack{D=3\\ C=1}}^{v_7^Q v_4^Q}, \\ [(v_4^Q \to v_3^Q \to v_5^Q \to v_6^Q)]_{\substack{D=3\\ C=1}}^{v_4^Q v_6^Q} \end{matrix}\right\} \tag{16}$$

Этап 4а:

П1.

Составляем наборы по $t = 5$ путей в каждом, которые удовлетворяют условию $C(k_i^Q) = C(w_i^G)$:

$$W^{Gtp} = \left\{\begin{matrix} \left\{\begin{matrix} [(v_{12}^G \to v_7^G \to v_6^G \to v_3^G)]_{\substack{D=3\\ C=1}}^{v_{12}^G v_3^G}, \\ [(v_{12}^G \to v_7^G \to v_8^G \to v_9^G)]_{\substack{D=3\\ C=1}}^{v_{12}^G v_9^G}, \\ [(v_7^G \to v_8^G \to v_{13}^G \to v_{14}^G)]_{\substack{D=3\\ C=1}}^{v_7^G v_{14}^G}, \\ [(v_{14}^G \to v_{13}^G \to v_{18}^G \to v_{17}^G)]_{\substack{D=3\\ C=1}}^{v_{14}^G v_{17}^G}, \\ [(v_{14}^G \to v_{13}^G \to v_8^G \to v_9^G)]_{\substack{D=3\\ C=1}}^{v_{14}^G v_9^G} \end{matrix}\right\} \\ \vdots \\ \left\{\begin{matrix} [(v_{15}^G \to v_{10}^G \to v_4^G \to v_1^G)]_{\substack{D=3\\ C=1}}^{v_{15}^G v_1^G}, \\ [(v_{16}^G \to v_{10}^G \to v_4^G \to v_1^G)]_{\substack{D=3\\ C=1}}^{v_{16}^G v_1^G}, \\ [(v_1^G \to v_4^G \to v_{11}^G \to v_{12}^G)]_{\substack{D=3\\ C=1}}^{v_1^G v_{12}^G}, \\ [(v_{15}^G \to v_{10}^G \to v_{11}^G \to v_{12}^G)]_{\substack{D=3\\ C=1}}^{v_{15}^G v_{12}^G}, \\ [(v_{16}^G \to v_{10}^G \to v_{11}^G \to v_{12}^G)]_{\substack{D=3\\ C=1}}^{v_{16}^G v_{12}^G} \end{matrix}\right\} \end{matrix}\right\} \tag{17}$$

П2.

Считаем, для каждой вершины из V^G для каждого графа, образованного из путей $w_{P_i}^G$, сколько раз она присутствует в выбранных графах:

$$S^{Gtp} = \left\{\begin{matrix} \{v_{12}^G 2, v_7^G 3, v_6^G 1, v_3^G 1, v_8^G 2, v_9^G 2, v_{13}^G 3, v_{16}^G 2\} \\ \vdots \\ \{v_{15}^G 2, v_{16}^G 2, v_{10}^G 3, v_4^G 3, v_1^G 3, v_{11}^G 3, v_{12}^G 3\} \end{matrix}\right\} \tag{18}$$

Для каждой вершины из v^K для каждого графа, образованного из путей k_i^Q, сколько раз она присутствует в выбранных графах:

$$S^{K^Q} = \{v_1^Q 3, v_2^Q 3, v_5^Q 3, v_6^Q 2, v_3^Q 3, v_4^Q 3, v_7^Q 2\} \tag{19}$$

Этап 5:

П1.

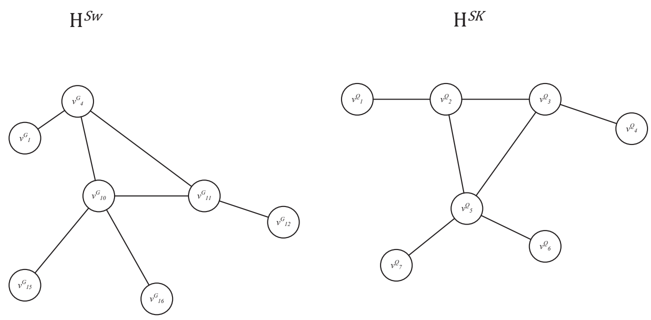
В наборах присутствия вершин S^{Gtp} ищем такие наборы коэффициентов, которые при перестановке выдают $S^{w_{P_i}^G} = S^{KQ}$. Очевидно, что в нашем случае это будет наборы:

$$\begin{matrix} S^{w_{P_1}^G} = \{v_{10}^G 3, v_4^G 3, v_1^G 3, v_{16}^G 2, v_{11}^G 3, v_{12}^G 3, v_{15}^G 2\} \\ S^{w_{P_2}^G} = \{v_{10}^G 3, v_4^G 3, v_1^G 3, v_{15}^G 2, v_{11}^G 3, v_{12}^G 3, v_{16}^G 2\} \end{matrix} \tag{20}$$

и $$S^{KQ} = \{v_1^Q 3, v_2^Q 3, v_5^Q 3, v_6^Q 2, v_3^Q 3, v_4^Q 3, v_7^Q 2\} \tag{21}$$

П2.

Путевые наборы, лежание в основе комбинаций $S^{w_{P_1}^G}$ и $S^{w_{P_2}^G}$, по сути, образуют один и тот же граф H^{Sw}, а *МПК* K^Q кторая обображает S^{KQ}, образует граф H^{SK}, что наглядно показано на Рисунке 4:



Р и с. 4. Наглядная демонстрация схожести графов, образованных комбинацией $S^{w_{P_1}^G}$ или и $S^{w_{P_2}^G}$ (H^{SW}) с графом, образованным комбинацией S^{KQ}(H^{SK})

F i g. 4. A visual demonstration of the similarity of graphs formed by a combination $S^{w_{P_1}^G}$ or with $S^{w_{P_2}^G}$ (H^{SW}) a graph formed by a combination S^{KQ}(H^{SK})

Рисунок 4 наглядным образом иллюстрирует изоморфизм графов $\mathrm{H}^{Sw} \simeq \mathrm{H}^{SK}$.

Исходя, из П1 и П2, суперграф G имеет в себе подграф H^{Sw}, образованный из вершин и ребер, путевых наборов, входящих в комбинацию или $S^{w_{P_1}^G}$ или $S^{w_{P_2}^G}$, схожий с искомым графом Q, поскольку искомый граф Q имеет изоморфный подграф H^{SK}, образованный его *МПК* K^Q: $\mathrm{H}^{Sw} \sim Q$.

Заключение

В статье предложен новый метод нахождения схожих невзвешенных неориентированных подграфов и подробно описан сам алгоритм. Алгоритм обладает рядом преимуществ:

- способен находить схожие графы путем выделения общей, изоморфной части в виде частичного графа, без вычисления изоморфизма подграфов и выявления дескрипторов.
- наличие проверок графов, которые позволяют обнаружить, что в суперграфе нет схожего подграфа, схожего с искомым графом, заранее, до проведения основных вычислений.

Однако, для полного раскрытия темы необходимо проведение следующих шагов.

- тестирование данного алгоритма в практической реализации, визуализация его работы и определение степени схожести найденных подграфов F^Q с искомым графом Q.
- поиск подграфа по заданному коэффициенту схожести. Формирование в искомом графе Q *НПН* по определенному правилу, и поиск их в суперграфе G.
- посвятить проблеме более полного обхвата искомого графа Q (зависимости МПК, его диаметра, и количества путевых наборов в нем).

И, в конечном счете, алгоритм планируется использовать для осуществления функции контроля исходного кода, оптимизации его структуры, визуального анализа фрагментов.

Список использованных источников

[1] Huang Z., Yuan L. Understanding Large-Scale Social Relationship Data by Combining Conceptual Graphs and Domain Ontologies // Discrete Dynamics in Nature and Society. 2021. Vol. 2021. Article number: 857611. https://doi.org/10.1155/2021/2857611

[2] Соколова О. Д. Графовые модели для задач функционирования современных сетей передачи данных // Проблемы информатики. 2014. № 4(25). С. 61-68. EDN: THSRHB

[3] Шахов В. В., Юргенсон А. Н., Соколова О. Д. Эффективный метод генерации случайных геометрических графов для моделирования беспроводных сетей // Прикладная дискретная математика. 2016. № 4(34). С. 99-109. https://doi.org/10.17223/20710410/34/8

[4] Ullmann J. R. An algorithm for Subgraph Isomorphism // Journal of the Association for Computing Machinery. 1976. Vol. 23, No. 1. P. 31-42. https://doi.org/10.1145/321921.321925

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

[6] A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs / L. Cordella [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Vol. 26, No. 10. P. 1367-1372. https://doi.org/10.1109/TPAMI.2004.75

[7] A subgraph isomorphism algorithm and its application to biochemical data / V. Bonnici [et al.] // BMC Bioinformatics. 2013. Vol. 14. Article number: S13. https://doi.org/10.1186/1471-2105-14-S7-S13

[8] Explainable Neural Subgraph Matching With Learnable Multi-Hop Attention / D. Q. Nguyen [et al.] // IEEE Access. 2024. Vol. 12. P. 130474-130492. https://doi.org/10.1109/ACCESS.2024.3458050

[9] Graph pattern matching: from intractable to polynomial time / W. Fan [et al.] // Proceedings of the VLDB Endowment. 2010. Vol. 3, No. 1-2. P. 264-275. https://doi.org/10.14778/1920841.1920878

[10] Зубов М. В., Пустыгин А. Н. Использование абстрактного цифрового автомата для получения универсального промежуточного представления исходного кода программ // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2015. № 4. С. 57-65. EDN: UYRZWV

[11] Лаздин А. В., Немолочнов О. Ф. Метод построения графа функциональной программы для решения задач верификации и тестирования // Научно-технический вестник информационных технологий, механики и оптики. 2002. № 6. С. 118-121.

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

[13] Wangmo C., Wiese L. An Experimental Evaluation of Summarisation-Based Frequent Subgraph Mining for Subgraph Searching // SN Computer Science. 2024. Vol. 5. Article number: 693. https://doi.org/10.1007/s42979-024-03006-w

[14] Кохов В. А., Ибрахим А. Р., Кохов В. В. Система моделей для анализа сходства графов с учетом расположения цепей // Вестник Московского энергетического института. 2009. № 5. С. 5-13. EDN: KXNBUR

[15] Khanna S., Putterman A., Sudan M. Near-Optimal Size Linear Sketches for Hypergraph Cut Sparsifiers // 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). Chicago, IL, USA : IEEE Computer Society, 2024. P. 1669-1706. https://doi.org/10.1109/FOCS61266.2024.00105

[16] Baskin I., Skvortsova M. On the basis of invariants of labeled molecular graphs // Journal of Chemical Information & Computer Sciences. 1995. Vol. 35, No. 3. P. 527-531.

[17] Stochastic Matching via In-n-Out Local Computation Algorithms / A. Azarmehr [et al.] // arXiv:2411.08805. 2024. https://doi.org/10.48550/arXiv.2411.08805

[18] Emmert Streib F., Dehmer M. Networks for systems biology: conceptual connection of data and function // IET Systems Biology. 2011. Vol. 5, issue 3. P. 185-207. https://doi.org/10.1049/iet-syb.2010.0025

[19] Dehmer M., Grabner M. The discrimination power of molecular identification numbers revisited // MATCH Communications in Mathematical and in Computer Chemistry. 2013. Vol. 69, No. 3. P. 785-794. URL: https://match.pmf.kg.ac.rs/electronic_versions/Match69/n3/match69n3_785-794.pdf (дата обращения: 19.01.2024).

[20] Molloy M., Pralat P., Sorkin G. B. Perfect matchings and loose Hamilton cycles in the semirandom hypergraph model // arXiv:2401.00559. 2024. https://doi.org/10.48550/arXiv.2401.00559

[21] Погребной В. К. Задача определения оценок сходства структур двух графов на основе выделения общих частей // Известия Томского политехнического университета. 2013. Т. 322, № 5. С. 194-199. EDN: QOXURJ

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

[23] 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. https://doi.org/10.25559/SITITO.17.202103.585-592

[24] Sudoso A. M. A Semidefinite Programming-Based Branch-and-Cut Algorithm for Biclustering // INFORMS Journal on Computing. 2024. https://doi.org/10.1287/ijoc.2024.0683

[25] Editorial: Experience with quantum annealing computation / N. Chancellor [et al.] // Frontiers in Computer Science. 2024. Vol. 6. Article number: 1481330. https://doi.org/10.3389/fcomp.2024.1481330

Поступила 19.01.2024; одобрена после рецензирования 15.02.2024; принята к публикации 12.03.2024.

Об авторах:

Сысоев Валентин Валерьевич, главный инженер Лаборатории кибербезопасности, Публичное акционерное общество «Сбербанк России» (117312, Российская Федерация, г. Москва, ул. Вавилова, д. 19); аспирант кафедры информационной безопасности факультета информатики и систем управления, ФГБОУ ВО «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)» (105005, Российская Федерация, г. Москва, ул. 2-я Бауманская, д. 5, к. 1), **ORCID: https://orcid.org/0000-0002-6157-5815,** valsus88@mail.ru

Быков Александр Юрьевич, доцент кафедры информационной безопасности факультета информатики и систем управления, ФГБОУ ВО «Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)» (105005, Российская Федерация, г. Москва, ул. 2-я Бауманская, д. 5, к. 1), кандидат технических наук, доцент, **ORCID: https://orcid.org/0000-0001-6336-5603,** abykov@bmstu.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

[1] Huang Z., Yuan L. Understanding Large-Scale Social Relationship Data by Combining Conceptual Graphs and Domain Ontologies. *Discrete Dynamics in Nature and Society*. 2021;2021:857611. https://doi.org/10.1155/2021/2857611

[2] Sokolova O.D. [Graph models for the problems of functioning of modern data transmission networks]. *Problems of Informatics.* 2014;(4):61-68. (In Russ.) EDN: THSRHB

[3] Shakhov V.V., Yurgenson A.N., Sokolova O.D. Fast method for generating random geometric graphs for wireless networks modeling. *Applied Discrete Mathematics*. 2016;(4):99-109. (In Russ., abstract in Eng.) https://doi.org/10.17223/20710410/34/8

[4] Ullmann J.R. An algorithm for Subgraph Isomorphism. *Journal of the Association for Computing Machinery*. 1976;23(1):31-42. https://doi.org/10.1145/321921.321925

[5] Ullmann J.R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *Journal of Experimental Algorithmics*. 2011;15:1.6. https://doi.org/10.1145/1671970.1921702

[6] Cordella L., 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. https://doi.org/10.1109/TPAMI.2004.75

[7] Bonnici V., Giugno R., Pulvirenti A., et al. A subgraph isomorphism algorithm and its application to biochemical data. *BMC Bioinformatics*. 2013;14(Suppl 7):S13. https://doi.org/10.1186/1471-2105-14-S7-S13

[8] Nguyen D.Q., Nguyen T.T., Jo J., Poux F., Anirban S., Quan T.T. Explainable Neural Subgraph Matching With Learnable Multi-Hop Attention. *IEEE Access.* 2024;12:130474-130492. https://doi.org/10.1109/ACCESS.2024.3458050

[9] Fan W., Li J., Ma S., Tang N., Wu Y., Wu Y. Graph pattern matching: from intractable to polynomial time. *Proceedings of the VLDB Endowment*. 2010;3(1-2):264-275. https://doi.org/10.14778/1920841.1920878

[10] Zubov M.V., Pustygin A.N. Use of finite-state automation for getting universal intermediate representation of program source code. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics.* 2015;(4):57-65. (In Russ., abstract in Eng.) EDN: UYRZWV

[11] Lazdin A.V., Nemolochnov O.F. [Method of constructing a graph of a functional program for solving verification and testing problems]. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics.* 2002;(6):118-121. (In Russ.)

[12] Broder A., et al. Graph structure in the Web. *Computer Networks*. 2000;33(1-6):309-320. https://doi.org/10.1016/S1389-1286(00)00083-9

[13] Wangmo C., Wiese L. An Experimental Evaluation of Summarisation-Based Frequent Subgraph Mining for Subgraph Searching. *SN Computer Science*. 2024;5:693. https://doi.org/10.1007/s42979-024-03006-w

[14] Kokhov V.A., Ibrahim A.R., Kokhov V.V. System of models for the analysis of graph's similarity with account of circuit arrangement. *Bulletin of Moscow Power Engineering Institute*. 2009;(5):5-13. (In Russ., abstract in Eng.) EDN: KXNBUR

[15] Khanna S., Putterman A., Sudan M. Near-Optimal Size Linear Sketches for Hypergraph Cut Sparsifiers. In: 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). Chicago, IL, USA: IEEE Computer Society; 2024. p. 1669-1706. https://doi.org/10.1109/FOCS61266.2024.00105

[16] Baskin I., Skvortsova M. On the basis of invariants of labeled molecular graphs. *Journal of Chemical Information & Computer Sciences*. 1995;35(3):527-531.

[17] Azarmehr A., Behnezhad S., Ghafari A., Rubinfeld R. Stochastic Matching via In-n-Out Local Computation Algorithms. *arXiv:*2411.08805. 2024. https://doi.org/10.48550/arXiv.2411.08805

[18] Emmert Streib F., Dehmer M. Networks for systems biology: conceptual connection of data and function. *IET Systems Biology.* 2011;5(3):185-207. https://doi.org/10.1049/iet-syb.2010.0025

[19] Dehmer M., Grabner M. The discrimination power of molecular identification numbers revisited. *MATCH Communications in Mathematical and in Computer Chemistry.* 2013;69(3):785-794. Available at: https://match.pmf.kg.ac.rs/electronic_versions/Match69/n3/match69n3_785-794.pdf (accessed 19.01.2024).

[20] Molloy M., Pralat P., Sorkin G.B. Perfect matchings and loose Hamilton cycles in the semirandom hypergraph model. *arXiv:*2401.00559. 2024. https://doi.org/10.48550/arXiv.2401.00559

[21] Pogrebnoy V.K. [The problem of determining the similarity estimates of the structures of two graphs based on the identification of common parts]. *Bulletin of the Tomsk Polytechnic University*. 2013;322(5):194-199. (In Russ.) EDN: QOXURJ

[22] Sysoev V.V. A Framework Similarity Estimation of Unweighted Undirected Graphs. *Modern Information Technologies and IT-Education.* 2022;18(3):655-665. (In Russ., abstract in Eng.) https://doi.org/10.25559/SITITO.18.202203.655-665

[23] 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.) https://doi.org/10.25559/SITITO.17.202103.585-592

[24] Sudoso A.M. A Semidefinite Programming-Based Branch-and-Cut Algorithm for Biclustering. *INFORMS Journal on Computing.* 2024. https://doi.org/10.1287/ijoc.2024.0683

[25] Chancellor N., McGeoch C.C., Mniszewski S., Niera D.D. Editorial: Experience with quantum annealing computation. *Frontiers in Computer Science*. 2024;6:1481330. https://doi.org/10.3389/fcomp.2024.1481330

Submitted 19.01.2024; approved after reviewing 15.02.2024; accepted for publication 12.03.2024.

About the authors:

Valentin V. Sysoev, Senior Engineer of the Cybersecurity Laboratory, Sberbank of Russia (19 Vavilova St., Moscow 117312, Russian Federation); Postgraduate Student of the Chair of Information Security, Faculty of the Computer science and control systems, Bauman Moscow State Technical University (5, 2nd Baumanskaya St., building 2, Moscow 105005, Russian Federation), **ORCID: https://orcid.org/0000-0002-6157-5815,** valsus88@mail.ru

Aleksandr Yu. Bykov, Associate Professor of the Chair of Information Security, Faculty of the Computer science and control systems, Bauman Moscow State Technical University (5, 2nd Baumanskaya St., building 2, Moscow 105005, Russian Federation), Cand. Sci. (Eng.), Associate Professor, **ORCID: https://orcid.org/0000-0001-6336-5603,** abykov@bmstu.ru

All authors have read and approved the final manuscript.