

УДК 004.421.2: 519.178

МЕТОД ДЕАНОНИМИЗАЦИИ ГРАФОВ СОЦИАЛЬНЫХ СЕТЕЙ НА ОСНОВЕ ТЕОРИИ ПЕРКОЛЯЦИИ

Нургалиев И.И., Садех Нобари

АНО ВО «Университет Иннополис», Иннополис, e-mail: i.nurgaliev@innopolis.ru, nobari@innopolis.ru

Проблема сопоставления графов огромных размеров является очень сложной и прикладной в разных сферах, в частности, в социальных сетях. Сложность ее связана с тем, что задача сопоставления графов в практике зачастую сталкивается с неэквивалентными графами, в которых может присутствовать разное количество вершин и ребер, таким образом, проблема сопоставления подобных графов сводится к поиску максимально общих подграфов, которая в свою очередь является NP-полной. В данной статье идет описание существующих подходов для решения задачи неполного сопоставления графов и особое внимание уделяется рассмотрению нового класса алгоритмов сопоставления графов, основанного на теории перколяции (просачивания), в связи с тем, что подобный класс хорошо проявил себя в работе с гигантскими графами. Все же, на основе проведенного формального анализа современных алгоритмов основанных на данной теории, мы выявили слабые стороны и критические случаи перколяции над графами для их сопоставления. Как результат, мы предлагаем способы их решения и обоснование возникающих ошибок, что будет полезным для дизайна улучшенного алгоритма сопоставления огромных графов.

Ключевые слова: графы, изоморфизм, сопоставление графов, социальные сети, протейновая сеть, теория перколяции, перколяция над графами

GRAPHS DEANONIMIZATION METHOD OF SOCIAL NETWORKS ON THE BASIS OF PERCOLATION THEORY

Nurgaliev I.I., Sadeh Nobari

Innopolis University, Innopolis, e-mail: i.nurgaliev@innopolis.ru, nobari@innopolis.ru

Graph Matching (GM) is a fundamental part of social network analysis. However, not only the complexity of the GM itself but also the growing size of graphs and the ubiquity of gigantic graphs further improves the complexity of the problem. As a result, among the many GM solutions very few, recently, are proposed to process large graphs. The complexity leads from the fact that in practice Graph Matching faces with non-equivalent graphs that could have different number of vertices and edges, thus the Graph Matching problem could be reduced to Maximum Common Induced Subgraph for revealing maximum count of common vertices that represent users uniquely, this problem is considered as NP-complete. In this article after a careful analysis of the state-of-the-art solutions for Graph Matching, we focused on the new class of GM algorithms that are based on the theory of percolation. This class of algorithms shows both theoretical and practical feasibility of processing gigantic graphs. However, we find that the existing solutions result to false positive matches under certain conditions that undermines the matching precision. Therefore, we conduct a formal analysis of the percolation based GM where we have identified some weak points and critical cases of the percolation based solution. Finally, thanks to the discovered causes of the weakness, we propose solutions that could overcome the shortcomings and substantially improve the precision of GM algorithms for gigantic graphs.

Keywords: graphs, isomorphism, graph matching, social network, ppi-network, theory of percolation, graph percolation

В связи с растущим количеством информации, представленной в виде графовых структур, их анализ становится центральной проблемой компьютерных наук в области анализа данных. Для некоторых задач величины графов превзошли возможности современных алгоритмов. Подобная тенденция ведет к нескольким интересным задачам, одной из которых является масштабирование алгоритмов с размером графа, если оригинальный алгоритм очень медленный.

В данной статье мы концентрируемся на решении проблемы сопоставления пользователей и их связей, представленных в графовых структурах разных социальных сетей больших размеров, принимая во внимание только топологии графовых

структур, в связи с тем, что один из подаваемых графов полностью анонимизирован. Под большими графами мы понимаем графы с количеством вершин более двух миллионов и средним ветвлением больше 30. К дополнительным ограничениям относится тот факт, что доля пользователей представленных уникальными вершинами в двух графах могут иметь разный набор ребер и более того, пользователь может отсутствовать во второй сети, таким образом задачу сопоставления графов уже нельзя будет свести к задаче изоморфизма двух графов в связи с тем, что целевая функция сопоставления $f': V_M \rightarrow V_D$ не является биективной. Данную задачу можно определить как неполное сопоставление графов (далее НСГ), стойкую к некоторому малому

набору ошибочных сопоставлений: даны два графа $G_M = (V_M, E_M)$ и $G_D = (V_D, E_D)$, $|V_M| < |V_D|$, задача заключается в том чтобы найти инъективную функцию $f': V_M \rightarrow V_D$, такую что $(u, v) \in E_M \leftrightarrow (f(u), f(v)) \in E_D$. Соответствующая задача разрешимости приведённой к задаче изоморфизма G_M и подграфа G_D является NP-полной, вследствие того, что это частный случай поиска максимального общего подграфа и поэтому не имеет полиномиального решения, даже если алгоритм ограничен по области применения (в предположении $P \neq NP$). Применение НСГ над огромными графами можно найти в таких задачах, как деанонимизация графа социальных сетей [6], чем на данный момент занимаются большие ИТ-компании для улучшения своей социальной сети за счет деанонимизации других социальных сетей, еще одно применение можно найти в сопоставлении сети взаимодействия протеинов разных видов животных для выявления общих частей, отвечающих за схожее функционирование в живом организме [3]. Метод решения проблемы НСГ, учитывающий только топологию графов, является по своей сути основой для будущих алгоритмов сопоставления графов в связи с ее обобщенностью над структурой графов.

Подходы и обоснование

Поиск структурного сопоставления в теории графов, известного как решение проблемы изоморфизма, имеет более общую задачу – сопоставление графов допускающая неточности (НСГ). Ниже приводим список самых известных подходов на данный момент для решения проблемы сопоставления небольших графов [2].

1. Редактирование дистанции: этот подход основан на интуитивно понятном измерении расхожести на графах при помощи поддерживаемого «расстояния редактирования». Этот подход имеет решения для точного системного сопоставления структуры и для подхода устойчивого к ошибкам с верхней границей точного расстояния редактирования. Целью является трансформация одного графа к другому путем некоторого числа операций (добавление, удаление, замещение вершин или ребер и перестановка ребер). Каждая операция ассоциируется со стоимостью, таким образом данный метод стремится найти последовательность операций минимизирующие оценку сопоставления графов. Несмотря на то, что подобный подход хорошо подходит для НСГ, его вычислительная сложность экспоненциально зависит от числа узлов в графах. Таким образом, для решения нашей проблемы подобный метод считается неприменимым.

2. Редактирование дистанции учитывая вероятность: идея заключается в моделировании распределения операции редактирования в пространстве имен вершин. Так как мы решаем задачу сопоставления анонимизированных графов, то данный способ тоже не применим вследствие наличия анонимизированного графа.

3. Вычисление максимального подграфа: основано на решении проблемы поиска максимальной клики в ассоциированном графе. Этот подход имеет решения для точного системного сопоставления структуры и для подхода допускающий мелкие погрешности, однако проблема поиска максимальной клики в свою очередь является NP-полной.

Самые масштабируемые методы СГ используют идеи из теории перколяции. Этот класс алгоритмов требует начальный набор изначально правильно сопоставленных вершин (далее семена), которые впоследствии используются как катализатор процесса одновременной перколяции над двумя графами. При процессе перколяции происходит массовая генерация пространства решений как пар кандидатов и далее применяются функции выбора соответствующих пар за счет контекстных данных каждой пары. Таким образом, подобные алгоритмы отличаются большой скоростью и динамическим сопоставлением, что можно будет использовать впоследствии над динамическими графами. Для деанонимизации сети были выбраны два самых эффективных алгоритма – НСГ на основе теории перколяции ExpandWhenStuck [5], который в свою очередь является расширением PercolationGM (PGM) [7]. Далее был проанализирован метод решения НСГ, основанный на теории перколяции, и были выявлены слабые стороны данного метода. Мы предлагаем способы их решения и обоснование возникающих ошибок, что будет полезным для дизайнера улучшенного алгоритма НСГ графов на основе теории перколяции.

Постановка задачи

На вход подаются два простых неориентированных графа $G_1(V_1, E_1)$ – анонимизированный и $G_2(V_2, E_2)$ вершины $V_{1,2}$ которые представляют одну и ту же сущность из множества сущностей $N = \{v_i \in V_1 \wedge v'_i \in V_2 \wedge (v_i, v'_i) \in N\}$. Далее, мы убираем метки с вершин и ребер G_2 . Так же на вход подается множество изначально сопоставленных пар $A_0 \subset N$ (далее семена), определенного размера $|A_0| = a_0$. Процесс сопоставления использует только структуру графов, которая представлена множеством вершин V и ребер E .

Оценка результатов НСГ проводится следующим образом:

$$1. \text{Точность} = \frac{\wedge(M^*)}{\wedge(M^*) + \Psi(M^*)},$$

где $\wedge(M^*)$ – количество правильных сопоставлений и $\Psi(M^*)$ – количество неправильных сопоставлений.

$$2. \text{Отклик} = \frac{\wedge(M^*)}{n_{\text{общ}}},$$

где $n_{\text{общ}}$ – число пересечений вершин в обоих графах.

$$3. F_{1\text{-оценка}} = 2 \frac{\text{точность} \times \text{отклик}}{\text{точность} + \text{отклик}}.$$

$F_{1\text{-оценка}} \in [0, 1]$ представляет из себя комбинацию оценок Точности и Отклика. Если $F_{1\text{-оценка}} \rightarrow 1$ тогда Точность $\rightarrow 1$, что говорит о том, что доля ошибок в M^* очень мала, таким же образом Отклик $\rightarrow 1$ говорит о том, что большая часть вершин присутствующих в $G_{1,2}$ сопоставлены правильно. Наша основная задача – максимизировать целевую функцию – $F_{1\text{-оценка}}$.

Модель случайного дублирования графа для тестирования алгоритма

Удобный подход для оценки алгоритмов СГ – это проверка его эффективности над моделью $G(n, p, t, s)$, которая позволяет генерировать два графа с пересечениями [5]. Биграфовая модель $G(n, p, t, s)$ генерирует два графа $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ следующим образом:

1. Генерируется граф $G(V, E)$ по заранее выбранной модели случайного графа [1]: Эрдеша – Реньи, Барабаши – Альберта либо Чунг-Лу.

2. Множество вершин $V_{1,2}$ выбирается случайным образом из множества V с вероятностью t .

3. Множество ребер $E_{1,2}$ выбирается случайным образом из множества E с вероятностью s . Результат $F_{1\text{-оценка}}$ мы получаем из $G_{1,2}$ путем верифицирования результирующих сопоставлений M^* через $G(V, E)$.

Метод деанонимизации графов на основе перколяции

Граф G представлен как V – набор вершин и E – набор ребер. Дается два графа: $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$. Степень ветвления вершин i и j в графах G_1 и G_2 обозначается как $d_{1,i}$ и $d_{2,j}$. Ребро в графе представлено как $(i, i') \in E_{1,2}$. Пара вершин представляется как $[i, j]: i \in V_1 \wedge j \in V_2$. Каждая пара $[i, j]$ имеет соседа $[i', j']: (i, i') \in E_1 \wedge (j, j') \in E_2$. Пара $[i, j] \in V_1 \times V_2$ соответствует какой-то единственной сущности (к примеру, че-

ловек), с чем алгоритм и работает. Нужно найти соответствующие пары вершин $[i, j] \subseteq V_1 \times V_2$, основываясь только на топологии двух сетей (непомеченные вершины и ребра), притом $V_{1,2}$ и $E_{1,2}$ могут не пересекаться полностью.

Изначально у нас имеются активированные пары A_0 в размере $|A_0| = a_0$ и граничный уровень $r \geq 2$, что является минимально возможной границей перколяции. В процессе сопоставления вершин графов, другими словами, выбора пары из множества кандидатов S , пополняется множество результатов M (сопоставленных пар), начальное значение которого $M \leftarrow A_0$. Так как мы ищем инъективную функцию $f': V_1 \rightarrow V_2$ в предположении, что $|V_1| \leq |V_2|$, то $V_1(M) = \{i: \exists j, i \in V_1 \wedge j \in V_2 \wedge [i, j] \in M\}$ и $V_2(M)$ не могут иметь повторяющиеся вершины.

Процесс перколяции над графами – это процесс активации вершин в графе, в нашем случае мы проводим перколяцию над двумя графами одновременно, таким образом – это процесс активации пары вершин $[i, j]$. $A(\tau)$ – активированные и $Z(\tau)$ – использованные пары во время τ , изначально $Z(0) = \emptyset$ и $A(0) = A_0$. На каждом временном шаге τ одна выбранная пара (далее активированная пара) $[i, j]: [i, j] \in A(\tau) \wedge [i, j] \notin Z(\tau) \wedge i \notin V_1(M) \wedge j \notin V_2(M)$ распространяет баллы по соседним парам, раздавая по одному баллу каждому из $[i', j']$ и далее переходит во множество результирующих сопоставлений M и после этого переходит во множество использованных пар Z . Пара из множества кандидатов S активируется во время τ , если по крайней мере r число соседних пар были активированы и использованы ранее. Оценка пары из множества кандидатов A определяется количеством полученных баллов в процессе перколяции графа, то есть в процессе раздачи баллов активизированными парами. В большинстве случаев среди пар в A встречается несколько пар с максимальной оценкой, таким образом, для максимизации вероятности правильного сопоставления пары мы выбираем $[i, j]$ такую, что разность ветвлений вершин представленных в паре имеет наименьшее значение: $\min(|d_{1,i} - d_{2,j}|)$, в этом случае вероятностью того, что $i = j$ больше чем $i \neq j$ [5]. Результатом является множество сопоставленных пар M^* , которое мы получаем при затухании процесса перколяции (остановка алгоритма), число правильно сопоставленных пар обозначается $\wedge(M^*)$, неправильных $\Psi(M^*)$.

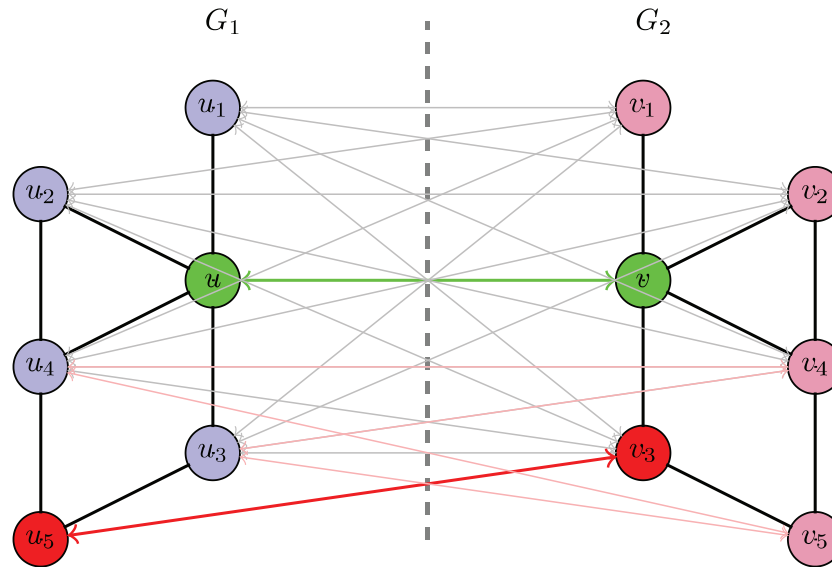


Рис. 1. Вершины u_1 , u_2 , u_3 и u_4 не сопоставленные соседи вершины u . Вершины u_4 и u_3 не сопоставленные соседи неправильно сопоставленной вершины u . После генерации новых кандидатов из соседних пар сопоставленной пары $[u, v]$ и разброса баллов по всем соседним парам мы получаем $[u, v_4]$ и $[u, v_3]$ кандидатов на сопоставление, как результат, будет выбрана пара $[u, v_4]$ имея большую оценку и $|d_{1,u_4} - d_{2,v_4}| = 0$

Параметр $a_{t,s,r}$ определяет число начальных семян для пересечения уровня фазового перехода в модели случайного графа Эрдеши – Реньи, примененной к дублированию графа, описанном выше. Для того, чтобы в результате иметь только корректные сопоставления над идентичными графами ($t=1, s=1$), достаточно иметь границу перколяции $r=4$ и, как минимум $|A_0| = a_{1,1,4}$ [4], что удалось доказать только для случайного графа Эрдеши – Реньи.

$$a_{t,s,r} = \left(1 - \frac{1}{r}\right) b_{t,s,r} \text{ (минимальный размер семян на вход алгоритму),}$$

$$\text{где } b_{t,s,r} = \left[\frac{(r-1)!}{nt^2 (ps^2)^r} \right]^{\frac{1}{r-1}}.$$

Для $\Psi(A_0) = 0$ нужно $a_{1,s,4}$ ($r=4$) семян.

Теорема об устойчивости к шумным активизированным парам [5] гарантирует, что множество семян размера $a_{1,s,2}$ ($r=2$) достаточно для сопоставления всех вершин с низким $\Psi(M_\tau)$ что требует значительно меньше начальных сопоставлений для начала перколяции $\frac{a_{1,s,4}}{a_{1,s,2}} \rightarrow \infty$. Доказательство структурировано на поиске нижней и выс-

шей границы доли ошибок, генерируемых за процесс перколяции [5].

Проведение экспериментов

Данные для эксперимента были получены путем генерации случайных графов Барабаши – Альберта, Чунг – Лу [1] и одного реального графа социальной сети Facebook небольшого размера. Дублирование графов для оценки алгоритмов проводится с параметрами $t=1$ и $s=1$. Для экспериментальных целей было выбрано 2 алгоритма, основанных на прочной идее, описанной в статье «Bootstrap percolation on the random graph $G(n, p)$ » [4]. Результаты представлены в таблице.

1. Percolation Graph Matching (PGM) [7] с пониженной фазовой границей $r=2$. Данный алгоритм по большей части зависит от начальных сопоставлений A_0 в связи с тем, что в процессе перколяции не происходит генерации семян.

2. ExpandWhenStuck [5] имеет пониженную планку к A_0 , за счет самогенерации семян и стабильности к неправильным сопоставлениям. Всякий раз, когда у алгоритма не имеется кандидатов со счетом в 2 балла и выше, он не прекращает перколяцию как PGM, в свою очередь, он продолжает перколяцию за счет расширения множества семян парами из множества неиспользованных Z и не входящих во множество результатов M , полученных от смежных уже сопоставлен-

ных вершин M : $A \leftarrow$ соседние пары $[i, j]$ от каждой пары из

$$M : [i, j] \notin Z \wedge i \in V_1(M) \wedge j \in V_2(M).$$

Слабые стороны алгоритмов перколяции и общие подходы для их решения

НСГ на основе перколяции неспособно сопоставить вершины с единичной степенью ветвления вследствие того, что границы фазового перехода $r = 2$. Стало очевидным, что граф, имеющий некоторое множество листьев, никогда не получит полный Отклик

(рис. 2, а, вершина 5). Более того, это влечет к невозможности сопоставить вершину-мост в случае, если этот мост – единственное соединение двух кластеров и перколяция происходит только в одном кластере, как результат, второй кластер не приступит к перколяции, что повлечет к очень низкому Отклику (рис. 2, а, вершина 6). Стоит отметить, что несопоставленные листья будут находиться во множестве кандидатов S , причем каждая из этих пар с высокой вероятностью является правильным сопоставлением, таким образом, возможно при полной остановке перколяции пополнить

$$A \leftarrow [i, j] : [i, j] \in S \wedge [i, j] \notin Z \wedge i \in M(V_1) \wedge j \in M(V_2)$$

и заново запустить алгоритм перколяции. Таким образом, листья будут сопоставлены и тем же образом перколяция пройдет через пару-мост к кластеру, где и продолжится перколяция.

Как было выявлено в теореме об устойчивости к ошибочным семенам, алгоритм СГ может продолжать процесс перколяции даже с ошибочными семенами и прийти к хорошим результатам. Все же, стоит уделить внимание снижению влияния неправильных сопоставлений до самого ми-

нимума. Процесс перколяции может сталкиваться с таким случаем, как состояние неподкрепленности принятия решения, которое основано лишь на оценках пары и разности ветвлений ее вершин. Во множестве кандидатов A при выборе пар с максимальной оценкой далее проверяется ветвление вершин в паре на минимальную разность, и вполне возможен случай, когда таких пар несколько. В итоге это ведет к случайному неподкрепленному выбору пары из A во множество сопоставлений M .

$$\{[i, j] : [i, j] \in A \wedge \text{оценка}([i, j]) = \max(\text{оценка}(A)) \wedge \min(d_{1,i} - d_{2,j})\} \vee \geq 2.$$

Пример приведен на рис. 2, б

$$(v_1, v'_1) \in M \wedge (v_1, v'_1) \notin Z$$

и

$$(v_2, v'_2) \in M \wedge (v_2, v'_2) \notin Z,$$

после распространения баллов по соседним парам (v_1, v'_1) и (v_2, v'_2) мы пополняем множество кандидатов $\{(v_3, v'_3), (v_3, v'_4), (v_4, v'_3), (v_4, v'_4)\}$, пары данного множества получают одинаковое количество баллов, притом вершины каждой из этих пар имеют одинаковую степень ветвления = 5. Таким образом, существующий алгоритм просто не в силах сделать взвешенный выбор для кандидата на сопоставление во множество M , что скажется на снижении оценки Точности. Как решение, в таких случаях можно создавать «обобщающую вершину» на каждом из двух графов, которая будет представлена как вершина кодирующая в себе несколько вершин, и далее пара из данных обобщающих вершин $[i_s, j_s] \rightarrow M$ (рис. 2, в).

Подобный подход еще интересен тем, что можно будет разработать в дальнейшем алгоритм архивации на основе перколяции.

Выводы

Был проведен анализ алгоритмов НСГ на основе перколяции с генерацией семян для поддержки процесса перколяции. В процессе анализа перколяции для сопоставления графов, были выявлены слабые моменты:

1. Процессу перколяции по крайней мере требуется граница перехода $r \geq 2$. В связи с этим листья не будут сопоставлены, что влечет плохой Отклик.

2. Невозможность сопоставить вершину-мост в случае, если этот мост – единственное соединение двух кластеров и перколяция происходит только в одном кластере, как результат, второй кластер не будет сопоставлен, что влечет более сильное ухудшение Отклика.

3. Состояние неподкрепленности принятия решения, встречаемое в процессе перколяции, влечет понижение оценки Точности.

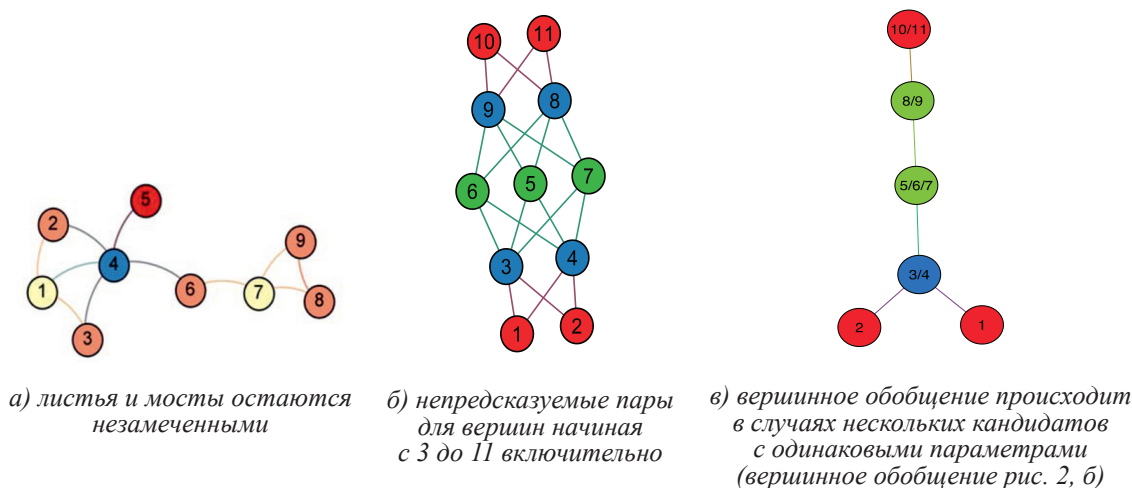


Рис. 2. Примеры случаев, с которыми не справляются современные алгоритмы НСГ на основе перколяции

Результаты представлены как (минимальное количество начальных семян / [F1-оценка * 100%]) для двух синтетических графов и одного реального

случ. граф алгоритм	Барабаши – Альберта $ V = 10^6$	Чунг-Льу $ V = 2 \cdot 10^5$	Facebook $ V = 4039$
PGM ($r = 2$)	185 / 98,8	4 / 99,9	135 / 0,91
ExpandWhenStuck	10 / 99,7	1 / 99,9	2 / 0,95

В данной работе были предложены причины и пути решения найденных проблем НСГ, основанного на теории перколяции. На основе проделанной работы мы усовершенствуем алгоритм НСГ и в дальнейшем планируем применить этот подход для реализации параллельной перколяции на платформе GPU для решения проблемы сопоставления графов.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-37-60089.

Список литературы

1. Берновски М.М., Кузюрин Н.Н. Случайные графы, модели и генераторы безмасштабных графов // Труды института системного программирования РАН. – 2012. – Т. 22.

2. Cook D.J., Holder L.B. (ed.). Mining graph data. – John Wiley & Sons, 2006.

3. Fabiana C. et al. Social Network De-anonymization Under Scale-free User Relations // IEEE-ACM TRANSACTIONS ON NETWORKING.

4. Janson S. et al. Bootstrap percolation on the random graph $G(n, p)$ // The Annals of Applied Probability. – 2012. – P. 1989–2047.

5. Kazemi E., Hassani S.H., Grossglauser M. Growing a graph matching from a handful of seeds // Proceedings of the VLDB Endowment. – 2015. – Т. 8, № 10. – P. 1010–1021.

6. Singh R., Xu J., Berger B. Global alignment of multiple protein interaction networks with application to functional orthology detection // Proceedings of the National Academy of Sciences. – 2008. – Т. 105, № 35. – P. 12763–12768.

7. Yartseva L., Grossglauser M. On the performance of percolation graph matching // Proceedings of the first ACM conference on Online social networks. – ACM, 2013. – P. 119–130.