

УДК 530.12; 519.178; 536.91

ПОИСК СООБЩЕСТВ С ИСПОЛЬЗОВАНИЕМ ПУЧКА ГЕОДЕЗИЧЕСКИХ НА ГРАФЕ

А. К. Илющенко, О. В. Иванов

*За последние десятилетия накоплен большой опыт применения различных подходов к решению задачи поиска сообществ в графах. Широкую известность имеет алгоритм Гирвана–Ньюмана, основанный на вычислении меры центральности рёбер *Betweenness*, который демонстрирует хорошие результаты на относительно небольших графах. В данной работе предложен новый алгоритм, основанный на вычислении центральности *Attendance*, объединяющий идеи случайных невожатных блужданий на графе, построения минимальных остовных деревьев, ранжирования вершин по величине центральности. Алгоритм имеет некоторые сходства с алгоритмом Гирвана–Ньюмана, однако некоторые его особенности позволили значительно увеличить эффективность работы алгоритма. В работе приведено описание предложенного алгоритма, проведен анализ сопоставимости величин *Betweenness* и *Attendance*, а также рассмотрены результаты работы алгоритма на *SBM* графах.*

Ключевые слова: граф, поиск сообществ, минимальное остовное дерево, степень посредничества.

1. *Введение.* Удобным инструментом для изучения системы взаимодействующих объектов является представление системы в виде графа, то есть набора вершин и соединяющих их рёбер, что позволяет исследовать, например, топологию получаемого графа, в том числе с точки зрения объединения объектов в сообщества. Сообществом в графе обычно называют группу вершин, более плотно связанных между собой, нежели с вершинами других групп. Такие объединения вершин в сообщества активно исследовались

ФИАН, 119991 Россия, Москва, Ленинский пр-т, 53; e-mail: a.ilyushchenko@lebedev.ru.

в различных сюжетах, таких как сложные комплексы химических реакций, взаимодействие белков и метаболические системы, сообщества соавторов научных публикаций, распространение эпидемий, различные задачи в социологии и многое другое [1–5], так как это позволяет выявить закономерные или, наоборот, аномальные взаимодействия объектов сюжета.

За последние десятилетия накоплен большой опыт применения различных подходов к решению проблемы поиска сообществ. Было предложено множество алгоритмов для обнаружения сообществ, среди них выделяются такие алгоритмы как оптимизация модулярности [6] и кодирование траектории на графе [7, 8, 11], вероятностные модели [9, 10, 13], методы, основанные на разметке вершин [14–16], использование остовных деревьев для нахождения сообществ [22–25], а также множество алгоритмов, использующие случайные невозвратные блуждания [12, 17, 18]. Среди существующих методов стоит отметить алгоритм, предложенный в работе Гирвана–Ньюмана [26], основанный на вычислении степени посредничества (величины *Betweenness*) рёбер (или вершин), которая определяется как доля кратчайших путей между парами вершин, проходящих через ребро (или вершину), относительно общего числа кратчайших путей. Таким образом, ребра, имеющие высокую величину *Betweenness*, вероятнее всего, относятся к ребрам, соединяющим вершины разных сообществ. Этот алгоритм хорошо зарекомендовал себя на множестве примеров, однако в качестве главного недостатка этого алгоритма указывается времязатратность на графах, количество вершин в которых больше 10000, поэтому актуальным остаётся вопрос о том, возможно ли придумать более эффективный способ расчета величины, похожей на *Betweenness* в том смысле, что наборы наиболее значимых вершин при ранжировании с использованием данных подходов были бы схожи. В данной работе предложен алгоритм с использованием величины, которую мы называем “Посещаемость” (или *Attendance*) вершины. Для тестирования алгоритма были использованы графы, сгенерированные методом стохастической блочной модели (SBM) [19], на которых предлагаемый подход показал более высокую эффективность, чем алгоритм Гирвана–Ньюмана.

2. *Связь предлагаемого метода с известными в литературе алгоритмами.* Алгоритм поиска сообществ, предлагаемый в данной работе, основан на идее построения случайных невозвратных траекторий для ранжирования вершин графа. Ранжирование вершин с помощью построений траекторий производится и в алгоритме Ньюмана–Гирвана, где рассчитываемая в этом алгоритме величина *Betweenness* ребер (вершин) связана с числом кратчайших путей между всеми парами вершин. При расчете вели-

чины Betweenness рассматриваются все кратчайшие траектории между всеми парами вершин. В данной работе построение случайных невозвратных траекторий производится таким способом, что искомые траектории также являются кратчайшими путями между вершинами, однако в расчетах оказывается рассмотрено некоторое подмножество кратчайших путей между вершинами, что будет описано далее.

Опишем способ построения случайных невозвратных траекторий. Он представляет собой построение минимального остовного дерева (Minimum Spanning Tree – MST). Минимальное остовное дерево – дерево, состоящее из всех вершин графа. Построение дерева включает в себя выбор корневой вершины и последовательное построение ярусов так, чтобы в результате дерево состояло из всех вершин исходного графа. Мы подразумеваем, что граф является связным, в противном случае считаем, что рассматривается какая-то из связных компонент исходного графа, для которой ставится задача поиска сообществ. В результате такого построения кратчайшими являются траектории от корневой вершины до любой другой вершины. В частности, при расчетах предлагается учитывать траектории, начинающиеся в корневой вершине и заканчивающиеся в терминальных вершинах, без возврата, то есть из одного построенного дерева оказывается сгенерировано столько траекторий, сколько терминальных вершин в полученном дереве. Более подробно способ построения будет описан в разделе 2.1.

Провести ранжирование вершин предлагается путем организации некоторого числа остовных деревьев (что влечет за собой построение достаточно большого количества случайных невозвратных траекторий) и для каждого дерева U производить подсчет для каждой вершины i , кроме стартовой, число траекторий K_i^U , соединяющих корневую вершину с терминальными вершинами и проходящих через неё. При этом для более эффективной генерации траекторий предлагается строить MST парно: для первого дерева в паре корневая вершина выбирается случайным образом, а для второго дерева в паре предлагается в качестве корневой выбрать одну из терминальных вершин первого дерева, наиболее удаленных от корневой вершины. Похожая идея построения MST парно уже встречалась в работе [24], однако в указанной работе алгоритм устроен иначе: сначала вычисляется матрица, называемая матрицей расстояний, элементы которой определяют веса рёбер, после чего производится построение двух MST: 1-е MST исходного графа G представляет собой древовидный подграф G , который включает все узлы исходного графа и общее расстояние минимально; 2-е MST представляет собой древовидный подграф без повторяющихся ребер из 1-го MST и общее расстояние минимально. После построения каждого из этих MST удаляются ребра с наибольшей

величиной соответствующего элемента матрицы расстояний, после чего граф разделяется на сообщества. Однако стоит обратить внимание, что при вычислении матрицы расстояний рассматривались только пути длиной 1, 2 или 3 ребра, в то время как в данной работе не ставятся ограничения на длину пути. После завершения процедуры генерации траекторий для каждой вершины i рассматривается величина Attendance, равная $\sum_U K_i^U$, то есть включающая только количество путей, прошедших через вершину i . Тогда вершины с наибольшей величиной Attendance являются кандидатами на роль мостиковых вершин (мостиковыми названо минимальное подмножество вершин графа, при удалении которого граф разбивается на отдельные компоненты).

2.1. Описание предлагаемого алгоритма. Таким образом, предлагаемый в данной работе алгоритм состоит из следующих шагов:

- расчет величины Attendance для каждой из вершин исходного графа,
- ранжирование вершин по убыванию величины Attendance,
- удаление из графа одной или нескольких вершин с наибольшей величиной Attendance. Это может привести к разделению графа на две или более компоненты.

Этот алгоритм по своей структуре аналогичен более ранней версии алгоритма Гирвана–Ньюмана без пересчета величин центральности [27], однако этот алгоритм в последующих работах был модифицирован.

Рассмотрим первый шаг алгоритма более подробно.

Без ограничения общности дальнейшие рассуждения строятся в предположении, что исходный граф является связным. В случае необходимости применения алгоритма к несвязному графу следует рассматривать каждую из его компонент отдельно.

Расчет величины Attendance для вершины i основан на подсчете количества случайных невозвратных траекторий, прошедших через эту вершину. Организовать процесс построения невозвратных случайных траекторий предлагается путем построения минимальных остовных деревьев на этом графе, то есть построения древовидного подграфа, содержащего каждую из вершин исходного графа. Алгоритм построения MST и расчета величины Attendance описан псевдокодом ниже. На первом шаге построения дерева выбирается его корневая вершина (строки 3–6). Напомним, что предлагается построение деревьев парно, выбирая корневую вершину в зависимости от четности номера MST: если номер нечетный, то вершина выбирается случайным образом из всех вершин графа, а если четный, то дерево строится парным к предыдущему, то есть его корневая вершина выбирается случайно из терминальных вершин (наиболее удаленных от корневой) предыдущего дерева. Строки 7–10 описывают построение каждого из уровней будущего

го дерева. Вершинами первого уровня оказываются все вершины, соседние с корневой. Вершины уровня l — вершины, соседние вершинам уровня $l - 1$, не задействованные на предыдущих уровнях. Для каждой вершины q уровня l рассматриваются все ребра, соединяющие вершину q с вершинами уровня $l - 1$ и случайно выбирается одно из них (строки 7–10). Построение продолжается, пока есть незадействованные вершины. Как было отмечено ранее, идея алгоритма включает в себя построение случайных невозвратных траекторий: этими траекториями являются траектории, соединяющие корневую вершину с каждой из терминальных вершин, откуда следует, что эти траектории невозвратные, а случайными они являются из-за случайности выбора ребер каждого следующего уровня.

Количество повторений обозначено в псевдокоде числом $Npath$. По завершению построений $Npath$ деревьев для каждой вершины i производится подсчет величины Attendance, то есть числа траекторий, начинающихся в корневых вершинах и заканчивающихся на концевых вершинах соответствующих деревьев, проходящих через вершину i . Отметим, что если какая-то вершина при построении какого-то из $Npath$ деревьев являлась корневой, то это дерево не дает вклада в величину Attendance этой вершины. Другими словами, при подсчете числа путей в конкретном дереве вклад в Attendance его корневой вершины не учитывается, так как иначе при малом значении $Npath$ величина Attendance корневых вершин искусственно увеличивается. Таким образом, величина Attendance вершины i — это количество случайных траекторий, построенных описанным способом, прошедших через вершину i .

Algorithm 1 Attendance

Input Граф, количество остовных деревьев $Npath$

Output Количества посещений каждой вершины

```

1:  $i := 1$ 
2: while  $i \leq Npath$  do
3:   if  $i \not\equiv 2$  then
4:     случайный выбор вершины  $s$ 
5:   else
6:     случайный выбор наиболее удаленной концевой вершины  $s$  предыдущего дерева
7:   while есть непосещенные вершины do
8:     отмечается набор вершин  $Q$ , соседних к вершинам  $P$  текущего уровня
9:     for каждого  $q \in Q$  do
10:      случайный выбор ребра, соединяющего  $q$  с  $p \in P$ 
11:    $i$  gets  $i + 1$ 
12: подсчет «Attendance» каждой из вершин, кроме  $stp$ 
  
```

2.2. Связь величин *Betweenness* и *Attendance* для значимых вершин графов, в которых есть сообщества. То, что при описанном выше построении траекторий от корневой вершины до каждой из вершин представляют собой кратчайшие пути, может указывать на то, что величины *Attendance* и *Betweenness* сопоставимы: *Betweenness* данной вершины предполагает расчет доли кратчайших путей, проходящих через данную вершину, от числа всех кратчайших путей между всеми парами вершин, тогда как *Attendance* – число некоторого подмножества кратчайших путей (способ их построения описан в предыдущем разделе), проходящих через данную вершину. Однако особенность величины *Attendance*, в отличие от *Betweenness*, состоит в том, что в условиях многократного запуска алгоритма для одного и того же графа при неизменной величине *Npath* для каждой из вершин возможно получение различных значений величины *Attendance*, что вызвано случайностью выбора корневых вершин и случайностью при построении деревьев. Также различные значения *Attendance* получаются при различных выборах числа *Npath*. Эта особенность, в частности, и приводит к необходимости анализа сопоставимости этих величин.

Для сравнительного анализа величин *Attendance* и *Betweenness* был применен коэффициент ранговой корреляции Спирмена ρ при увеличении *Npath* по формуле (1):

$$\rho = 1 - \frac{6 \cdot \sum_{i=1}^q d_i^2}{q^3 - q}, \quad (1)$$

где q – количество вершин, для которых определено значение *Betweenness*, d_i – разность между рангами наблюдения i .

Отметим, что в работе [26] алгоритм подразумевает пересчет величин *Betweenness* после каждого удаления ребра с наибольшей *Betweenness*. Сравнительный анализ величин производится в случае однократного расчета величин *Betweenness*. Предлагаемый в этой работе алгоритм с пересчетом величин *Attendance* будет рассмотрен в разделе 3.

Для сравнительного анализа величин *Attendance* и *Betweenness* была проведена серия экспериментов на модельных графах известной структуры (рис. 1): для каждого из графов при фиксированном *Npath* собиралась статистика получаемых значений ρ , среднее значение фиксировалось. Таким образом были получены графики зависимостей $\rho(Npath)$ для модельных графов (рис. 2). По полученным графикам видно, что при некотором *Npath* значение ρ становится больше 0.7. Граф из Примера 2 не отображен на графике, так как уже при $Npath = 2$ значение ρ достигает значения 0.9. Примеры 2 и 3 отличаются тем, что в примере 2 каждое из двух сообществ представляет собой полный граф из 99 вершин, а в примере 3 в каждом из двух сообществ 99 вершин, но

на 1% связей меньше, чем в полном графе. Этим можно объяснить разницу в числе $Npath$, при котором ρ достигает значения больше 0.9: особенность Betweenness заключается в том, что ее значение может быть определено не для каждой из вершин, что при $Npath > 1$ неверно для Attendance. Так, в графе Примера 2 значение Betweenness определено только для вершины номер 100, в то время как значение Attendance определено для каждой из вершин графа. Для расчета значения ρ каждый раз выбирались первые q наибольших значений Attendance и Betweenness. В примере 3 значение Betweenness определено для каждой из вершин, поэтому для большего значения ρ нужно увеличивать $Npath$. Однако в этом примере при расчете ρ для первых 10% вершин, $Npath = 2$ оказывается достаточным для $\rho > 0.9$. Примеры 4 и 5 сгенерированы методом SBM [19]. Генерация графа методом SBM включает следующие шаги: 1) для каждой вершины определяют номер сообщества, которому она принадлежит, 2) ребро между парами вершин i и j генерируется с вероятностью ω_{rs} , где r – номер сообщества вершины i , s – номер сообщества вершины j . Для рассмотрения был выбран симметричный случай, когда $\omega_{rr} = \omega_{in}$, $\omega_{rs} = \omega_{out}$, где $r \neq s$. Число вершин графов в примерах 4 и 5 равно 200, число сообществ равны 3 и 2, соответственно, $\omega_{in} = 1$, $\omega_{out} = 0.0003$.

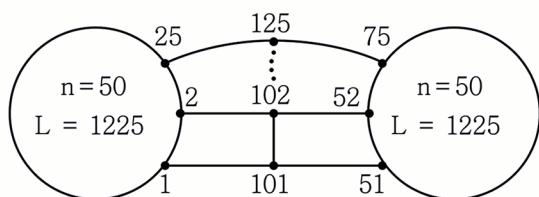
3. *Обсуждение результатов.* Эффективность работы предложенного алгоритма зависит от того, какое значение $Npath$ необходимо выбрать для того, чтобы удаление из графа вершин с наибольшей величиной Attendance было достаточным для того, чтобы граф разделился на две или несколько компонент.

Обозначим за N количество вершин в графе. Покажем, что достаточным окажется выбор $Npath$ значительно меньше N .

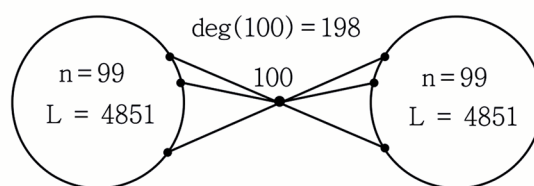
Для того чтобы показать этот факт, были сгенерированы SBM графы с количеством сообществ 2, 3 и 4, в каждом из которых 800 вершин, для генерации были взяты следующие вероятности: $\omega_{in} = 0.8$, $\omega_{out} = 0.00003$. В результате такой генерации мостиковые вершины считались известными для дальнейшей обработки результатов выполнения алгоритма. Мостиковыми ребрами называем ребра, соединяющие вершины разных сообществ, а соответствующие им вершины – мостиковыми вершинами.

Для каждого графа собиралась следующая статистика: пары MST генерировались последовательно так, что корневые вершины полученных MST различны, затем вычислялось число вершин, выделенных как значимые. Если число выделенных вершин оказывалось меньше половины искомым, то генерировалась следующая пара. Затем фиксировался номер пары, после генерации которой впервые количество значимых вершин становилось больше или равным половине числа искомым мостиковых вершин.

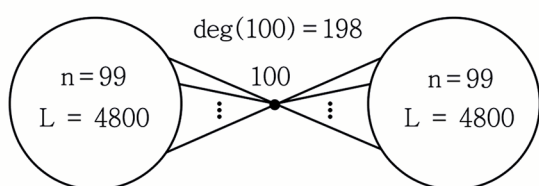
Пример 1



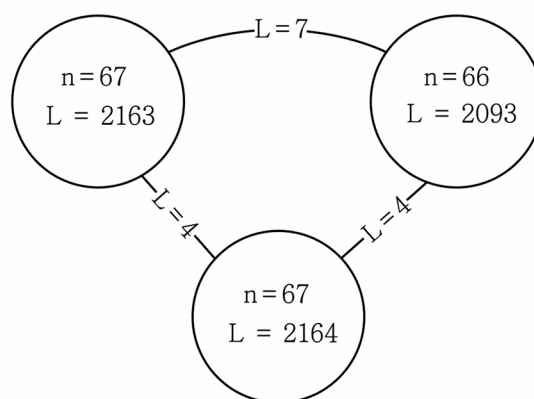
Пример 2



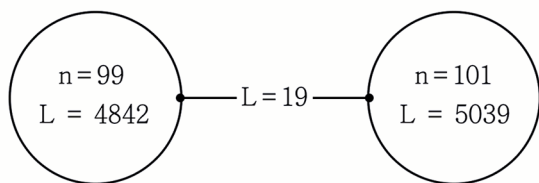
Пример 3



Пример 4 SBM



Пример 5 SBM



Пример 6

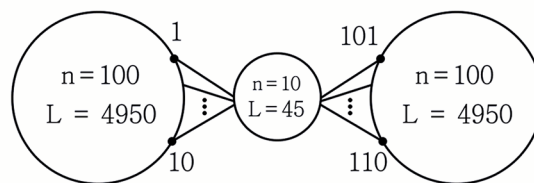
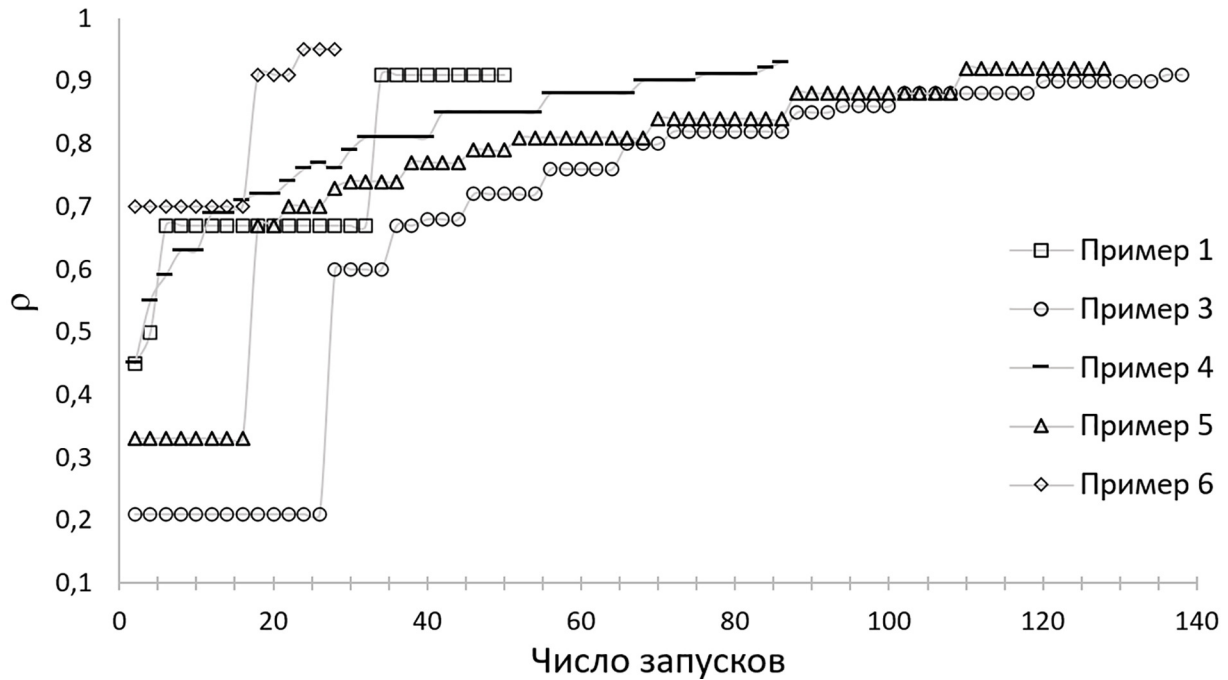


Рис. 1: Модельные графы для сравнительного анализа *Attendance* и *Betweenness*. Для каждого “сообщества” указано количество вершин и количество ребер, из которых оно состоит. Числом на ребре между “сообществами” указано количество ребер между вершинами этих сообществ. Для некоторых вершин указаны их номера.

Производилось сравнение именно с половиной числа искомых мостиковых вершин, так как для того, чтобы граф разделился на компоненты, необходимо удалить мостиковые ребра, а для удаления ребра необходимо удалить хотя бы одну из вершин, которые соединяет это ребро. Может случиться, что удаление одной мостиковой вершины приводит к удалению нескольких мостиковых ребер, но в таком случае по-прежнему нужно удалить не более половины мостиковых вершин.

Рис. 2: $\rho(Npath)$ для модельных графов.

Также отметим, что среди выделенных вершин оказывалось только лишь подмножество искомым мостиковых вершин, то есть немостиковые вершины среди выделенных не оказывались. Это можно объяснить тем, что, если в исследуемом графе есть сообщества, в предложенном алгоритме существенную роль играют траектории, прошедшие через мостиковые вершины, а траектории, у которых стартовые и терминальные вершины принадлежат одному сообществу, вносят значительно меньший вклад в величины Attendance.

Для сгенерированных графов было получено, что для графов с двумя сообществами оказывается достаточным число $Npath$, равное примерно 4% от N , для графов с тремя сообществами – примерно 5% от N , а в случае четырех сообществ – примерно 1% от N .

Этот результат демонстрирует преимущество в эффективности работы предложенного алгоритма в сравнении с алгоритмом [27] (что является одной итерацией пересчета Betweenness алгоритма Гирвана–Ньюмана), так как одна итерация алгоритма Гирвана–Ньюмана для расчета величины Betweenness предполагает построение всех кратчайших путей между каждой парой вершин A и B графа. Для этого последовательно рассматривается каждая из вершин графа от 1 до N и для каждой выбранной вершины рассматриваются все кратчайшие пути от нее до каждой из оставшихся вершин графа.

То есть процесс построения кратчайших путей от выбранной вершины до каждой из оставшихся вершин необходимо повторить N раз. В то время как, во-первых, в предложенном алгоритме достаточно рассматривать лишь некоторые кратчайшие пути между вершинами, а во-вторых, процесс построения кратчайших путей необходимо повторить существенно меньшее число раз.

Вернемся к вопросу о том, что в предыдущих рассуждениях рассматривался вариант алгоритма без пересчетов, схожий с более ранней версией алгоритма Гирвана–Ньюмана [27]. В ней автором было предложено удалять сразу несколько значимых вершин для разделения графа на сообщества. В более поздней работе [26] авторы указывали на лучшие результаты разделения графов, если после расчета *Betweenness* удалять первую (или одну случайно выбранную) значимую вершину, а затем пересчитывать *Betweenness* для оставшихся вершин, что нужно повторять несколько раз. Предлагаемый нами алгоритм, использующий центральность *Attendance*, можно модернизировать аналогичным образом без потери эффективности на каждом шаге. Уточнение числа повторений является предметом дальнейших исследований.

4. *Заключение.* В этой работе был предложен алгоритм поиска сообществ, основанный на расчете новой центральности вершин, названной “Посещаемость” (или *Attendance*). Предложенный алгоритм совмещает идеи применения невозвратных случайных блужданий, построения минимального остовного дерева (MST) парно, ранжирования вершин по величине центральности для выделения мостиковых вершин (мостиковыми ребрами названы ребра, соединяющие вершины разных сообществ, а соответствующие им вершины – мостиковыми вершинами).

Величина в предложенном алгоритме имеет некоторые сходства с имеющей широкую известность величиной *Betweenness*, однако принципиально иной подход в подсчете кратчайших путей между вершинами на графе (как было показано на графах, сгенерированных методом SBM) приводит к более высокой эффективности работы алгоритма в сравнении с алгоритмом, используемым для вычисления *Betweenness* [26]. Это можно объяснить тем, что в *Betweenness* вносят вклад кратчайшие пути между всеми парами вершин, тогда как в предложенном подходе достаточно рассмотрения лишь некоторого подмножества кратчайших траекторий, построение которых организовано методом MST.

Предложенный алгоритм может быть использован для решения ряда задач статистической физики, химической кинетики и междисциплинарных областей.

Авторы выражают благодарность Леонидову А. В. и Васильевой Е. Е. за продуктивное обсуждение данной работы. А также выражаем благодарность рецензентам за ценные замечания.

Л И Т Е Р А Т У Р А

- [1] M. E. J. Newman, Physical review E **64**(1), 016132 (2001).
- [2] M. E. J. Newman, SIAM review **45**(2), 167 (2003).
- [3] R. Albert, A. L. Barabási, Reviews of modern physics **74**(1), 47 (2002).
- [4] S. H. Strogatz, Nature **410**(6825), 268 (2001).
- [5] S. Boccaletti et al., Physics reports **424**(4-5), 175 (2006).
- [6] M. Rosvall, C. T. Bergstrom, Proceedings of the national academy of sciences **104**(18), 7327 (2007).
- [7] J. Rissanen, Automatica **14**(5), 465 (1978).
- [8] A. Barron, J. Rissanen, B. Yu, IEEE transactions on information theory **44**(6), 2743 (1998).
- [9] Jin, Di and Yu, Zhizhi and Jiao, et al., IEEE Transactions on Knowledge and Data Engineering **35**(2), 1149 (2021).
- [10] M. Rosvall, C. T. Bergstrom, Proceedings of the national academy of sciences **105**(4), 1118 (2008).
- [11] Jelena Smiljanić, Christopher Blöcker, Anton Holmgren, et al., Community detection with the map equation and infomap: Theory and applications //arXiv preprint arXiv:2311.04036. – 2023.
- [12] S. Fortunato, Physics reports **486**(3-5), 75 (2010).
- [13] M. E. J. Newman, Proceedings of the Royal Society A **479**(2270), 20220774 (2023).
- [14] J. P. Bagrow, E. M. Bollt, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **72**(4), 046108 (2005).
- [15] L. F. Costa, Hub-based community finding //arXiv preprint cond-mat/0405022. – 2004.
- [16] U. N. Raghavan, R. Albert, S. Kumara, Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **76**(3), 036106 (2007).
- [17] G. de Guzzi Bagnato, J. R. F. Ronqui, G. Travieso, Physica A: Statistical Mechanics and its Applications **505**, 1046 (2018).
- [18] Víctor M. López Millán, Vicent Cholvi, Luis López, Antonio Fernández Anta, Networks **60**(2), 71 (2012).

- [19] P. W. Holland, K. B. Laskey, S. Leinhardt, Social networks **5**(2), 109 (1983).
- [20] L. Danon, A. Diaz-Guilera, A. Arenas, Journal of Statistical Mechanics: Theory and Experiment **2006**(11), P11010 (2006).
- [21] R. Guimera et al., Physical review E **68**(6), 065103 (2003).
- [22] P. Basuchowdhuri, S. Anand, D. R. Srivastava, et al., Advanced Computing, Networking and Informatics-Volume 2: Wireless Networks and Security Proceedings of the Second International Conference on Advanced Computing, Networking and Informatics (ICACNI-2014) (Springer International Publishing, 2014).
- [23] K. Asmi, D. Lotfi, M. El Marraki, 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM) (IEEE, 2016).
- [24] Wu Jianshe, Li Xiaoxiao, Jiao Licheng, et al., Physica A: Statistical Mechanics and its Applications **392**(9), 2265 (2013).
- [25] R. K. Behera, S. K. Rath, M. Jena, Procedia Computer Science **93**, 1070 (2016).
- [26] M. E. J. Newman, M. Girvan, Physical review E **69**(2), 026113 (2004).
- [27] M. E. J. Newman, Physical review E **64**(1), 016132 (2001).
- [28] <https://www.cs.cornell.edu/courses/cs685/2002fa/data/gr0.California>.
- [29] W. W. Zachary, Journal of anthropological research **33**(4), 452 (1977).
- [30] M. Girvan, M. E. J. Newman, Proceedings of the national academy of sciences **99**(12), 7821 (2002).
- [31] D. E. Knuth, The Stanford GraphBase: a platform for combinatorial computing (New York, AcM Press, 1993). – Vol. 1.
- [32] <https://github.com/gonum/gonum/tree/master/graph>.

Поступила в редакцию 2 апреля 2025 г.

После доработки 25 апреля 2025 г.

Принята к публикации 28 апреля 2025 г.