УДК 519.17+157 DOI 10.17223/20710410/64/9

ПОДХОД К АНАЛИЗУ И ПОСТРОЕНИЮ АЛГОРИТМОВ РЕШЕНИЯ ОДНОЙ ЗАДАЧИ КЛАСТЕРИЗАЦИИ НА ЗНАКОВЫХ ГРАФАХ[1]

А. А. Солдатенко, Д. В. Семенова, Э. И. Ибрагимова

Сибирский федеральный университет, г. Красноярск, Россия

E-mail: ASoldatenko@sfu-kras.ru, DVSemenova@sfu-kras.ru, IbragimovaEI@mail.ru

Рассматривается NP-трудная оптимизационная задача корреляционной кластеризации для неориентированных и невзвешенных знаковых графов без кратных рёбер и петель, где функционал ошибки представляет собой линейную комбинацию межкластерной и внутрикластерной ошибок. Предложен системный подход построения и анализа алгоритмов, основанных на структуре графа, для решения этой задачи. Подход представлен в виде общей схемы, состоящей из шести взаимосвязанных блоков, отражающих основные этапы решения задачи корреляционной кластеризации. С использованием данной схемы проанализированы шесть существующих алгоритмов. Согласно общей схеме построен новый алгоритм `CarVeR`, который является модификацией алгоритма $\texttt{SGClust}_\alpha$ с помощью потенциальных функций. Топология общей схемы открывает возможности для анализа и доказательства вычислительной сложности алгоритмов, что продемонстрировано в теореме о вычислительной сложности алгоритма `CarVeR`. Представлены вычислительные эксперименты на синтетических данных для сравнения пяти алгоритмов. Результаты экспериментов показали конкурентную способность алгоритма `CarVeR` как по времени выполнения, так и по минимизации значения функционала ошибки.

Ключевые слова: *знаковый граф, корреляционная кластеризация, систематизация алгоритмов, потенциальные функции.*

APPROACH TO ANALYSIS AND CONSTRUCTION OF ALGORITHMS FOR SOLVING ONE CLUSTERING PROBLEM ON SIGNED GRAPHS

A. A. Soldatenko, D. V. Semenova, E. I. Ibragimova

Siberian Federal University, Krasnoyarsk, Russia

We consider the NP-hard correlation clustering problem for undirected and unweighted signed graphs without multiple edges and loops, where the error functional is a linear combination of intercluster and intracluster errors. In this paper, we propose a systematic approach for constructing and analyzing graph structure based algorithms to solve this problem. The approach is presented in the form of a general scheme consisting of six interrelated blocks reflecting the main stages of solving the correlation clustering problem. Six existing algorithms have been analyzed using this scheme. According to the general scheme, a new algorithm `CarVeR` has been constructed, which is a modification of the $\texttt{SGClust}_\alpha$ algorithm using potential functions. The topology

[1]Работа поддержана Красноярским математическим центром, финансируемым Минобрнауки РФ (Соглашение № 075-02-2024-1429).

of the general scheme opens up the possibility of analyzing and proving the computational complexity of the algorithms, which is demonstrated in the computational complexity theorem of the CarVeR algorithm. This paper presents computational experiments on synthetic data to compare five algorithms. The experimental results show the competitive ability of the CarVeR algorithm both in terms of execution time and minimization of the value of the error functional.

Keywords: *signed graph, correlation clustering, algorithm systematization, potential functions.*

Введение

На протяжении десятилетий исследователи активно изучают задачу корреляционной кластеризации и предлагают различные методы ее решения. В западной литературе данная задача носит название Correlation Clustering problem. Исторический обзор по данной проблеме приведён в [1], а достаточно подробный обзор существующих методов решения представлен в [2].

Алгоритмы решения задачи корреляционной кластеризации можно условно разделить на три группы [2]. Первая группа алгоритмов учитывает структуру графа [2, 3]. Для второй группы характерно представление задачи корреляционной кластеризации как задачи математического программирования (например, линейного, целочисленного линейного, полуопределённого и др.) и использование соответствующих методов и алгоритмов для решения задачи [2, 4–8]. Третья группа основана на различных матричных представлениях графа, что позволяет применять в алгоритмах аппарат матричной алгебры [2, 9]. Далее внимание акцентировано на первой группе алгоритмов.

Структура работы следующая. В п. 1 приведены необходимые для дальнейшего изложения определения и формулировка задачи корреляционной кластеризации знаковых графов. В п. 2 исследованы популярные алгоритмы её решения, основанные на структуре графа, и предложена общая схема построения и анализа таких алгоритмов. В п. 3 исследуется новый алгоритм CaRVeR. Результаты вычислительных экспериментов по сравнению алгоритма CaRVeR с некоторыми известными алгоритмами представлены в п. 4.

1. Постановка задачи

1.1. Знаковый граф

В работе исследуются знаковые графы вида $\Sigma = (G, \sigma)$, где $G = (V, E)$ является неориентированным невзвешенным графом без кратных рёбер и петель с множеством вершин V, $|V| = n \geqslant 2$, и множеством рёбер E, $|E| = m \geqslant 1$. В графе G каждое ребро однозначно представляется неупорядоченной парой $e = (u, v)$, где $e \in E$, $u, v \in V$. В этом случае говорят, что ребро e инцидентно вершинам u и v, а вершины u и v смежны. Обозначим множество вершин, смежных с v, как $\Gamma(v) = \{u\colon (v, u) \in E\}$. Под степенью вершины v понимается число рёбер, инцидентных ей. Очевидно, что $\delta(v) = |\Gamma(v)|$; степенью графа будем считать $\Delta = \max\limits_{v \in V} \delta(v)$. На рёбрах $(u, v) \in E$ графа G задана функция знака $\sigma : E \to \{+, -\}$, которая порождает разбиение множества рёбер графа $E = E^+ \cup E^-$, где E^+ — множество положительных, E^- — множество отрицательных рёбер. Для ребра $e = (u, v)$ функция знака представима в виде

$$\sigma(u, v) = \operatorname{sign}\Big(\big[(u, v) \in E^+\big] - \big[(u, v) \in E^-\big]\Big),$$

где $[\cdot]$ — скобки Айверсона [10].

Знаковый граф называется k-сбалансированным, если множество его вершин можно разбить на k попарно непересекающихся непустых подмножеств так, что все положительные рёбра находятся внутри, а отрицательные — между подмножествами [11].

1.2. Задача корреляционной кластеризации

Обозначим систему множеств, образующих разбиение множества вершин V на k подмножеств, как

$$\mathcal{C} = \left\{ C_i \subseteq V : \bigcup_{i=1}^{k} C_i = V,\ C_i \cap C_j = \varnothing, i \neq j;\ i = 1, \ldots, k \right\}. \tag{1}$$

Известно, что для произвольного знакового графа свойство k-сбалансированности может не выполняться. В этом случае интересен поиск такого разбиения множества вершин графа, для которого число отрицательных рёбер внутри подмножеств и число положительных рёбер между подмножествами будут минимальны. Данная задача рассматривается как задача кластеризации графа со специальным видом функционала ошибки. Элементы разбиения $C_i \in \mathcal{C}$ будем называть кластерами.

Под положительной ошибкой $P(\mathcal{C})$ разбиения (1) будем понимать число положительных рёбер между подмножествами $C_1, \ldots, C_k$. Заметим, что $P(\mathcal{C})$ — это межкластерная ошибка, вычисляемая по формуле

$$P(\mathcal{C}) = \sum_{i=1}^{k} \sum_{u \in C_i} \sum_{v \in V \setminus C_i} [(u, v) \in E^+]\,. \tag{2}$$

Под отрицательной ошибкой $N(\mathcal{C})$ будем понимать число отрицательных рёбер внутри подмножеств для разбиения (1). Отрицательная ошибка — это внутрикластерная ошибка, вычисляемая по формуле

$$N(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\{u,v\} \subseteq C_i} [(u, v) \in E^-]\,. \tag{3}$$

В [12] авторы предлагают представлять суммарную ошибку в виде выпуклой комбинации положительной и отрицательной ошибок, зависящей от параметра $\alpha \in [0, 1]$:

$$Q_\alpha(\mathcal{C}) = \alpha N(\mathcal{C}) + (1 - \alpha) P(\mathcal{C}). \tag{4}$$

Заметим, что функционал ошибки (4) всегда удовлетворяет неравенству

$$0 \leqslant Q_\alpha(\mathcal{C}) \leqslant \alpha |E^-| + (1 - \alpha) |E^+|.$$

Задачу кластеризации знакового графа будем рассматривать в следующей постановке [13, 14].

Correlation Clustering problem (задача CC)

Условие: задан знаковый граф $\Sigma = (G, \sigma)$, где $G = (V, E)$ — неориентированный граф; $n = |V| \geqslant 2$; $m = |E| \geqslant 1$.

Вопрос: для заданного $\alpha \in [0, 1]$ требуется найти разбиение $\mathcal{C}$ множества вершин V знакового графа Σ с минимальной суммарной ошибкой $Q_\alpha(\mathcal{C})$.

В работе [13] показано, что задача корреляционной кластеризации знаковых графов с функционалом ошибки в виде (4) при $\alpha = 0{,}5$ в распознавательной форме является NP-полной.

Решением задачи является множество кластеров $\mathcal{C}^*$, доставляющих минимум функционалу ошибки (4):

$$\mathcal{C}^* = \arg\min_{\mathcal{C}\in\Phi}\big[\alpha N(\mathcal{C}) + (1-\alpha)P(\mathcal{C})\big], \tag{5}$$

где $\Phi = \bigcup\limits_{k=1}^{n} \Phi_k$ — множество всех возможных разбиений V; Φ_k — множество разбиений на k подмножеств. Мощность пространства решений Φ равна числу Белла B_n. Следует отметить, что решение (5) может быть не единственным.

При $\alpha = 0$ и 1 данная задача вырождается в полиномиально разрешимые случаи минимизации межкластерной (2) и внутрикластерной (3) ошибок соответственно.

Одна из стратегий поиска нетривиального решения задачи корреляционной кластеризации знаковых графов для $\alpha = 0$ формулируется следующим образом. Множество кластеров $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ формируется из исходного знакового графа Σ путём нахождения компонент связности $C_1, C_2, \ldots, C_k$ порождённого графа $\Sigma^+ = (V, E^+)$. Полученное разбиение имеет ошибку $Q_0(\mathcal{C}) = 0$. Такая стратегия может быть реализована алгоритмами поиска в глубину или ширину, временна́я сложность которых составляет $\mathcal{O}(n + |E^+|)$ [15]. Для поиска нетривиального решения с параметром $\alpha = 1$ можно применить следующую стратегию. Изначально полагается, что все вершины находятся в одном кластере. Далее на каждом шаге вершина, инцидентная наибольшему числу отрицательных рёбер, выделяется в отдельный кластер. В результате для любого ребра $e = (u, v) \in E^-$ выполняется, что $u \in C_i$ и $v \in C_j$, где $i \neq j$. Данная процедура приводит к разбиению $\mathcal{C}$ с ошибкой $Q_1(\mathcal{C}) = 0$. Такая стратегия выполнима за время, не превышающее $\mathcal{O}(n^2 + nm)$.

2. Общая схема алгоритмов, основанных на структуре графа

Исследование алгоритмов, основанных на структуре графа, выявило общую схему в организации вычислений для решения задачи корреляционной кластеризации, которая представлена на рис. 1.

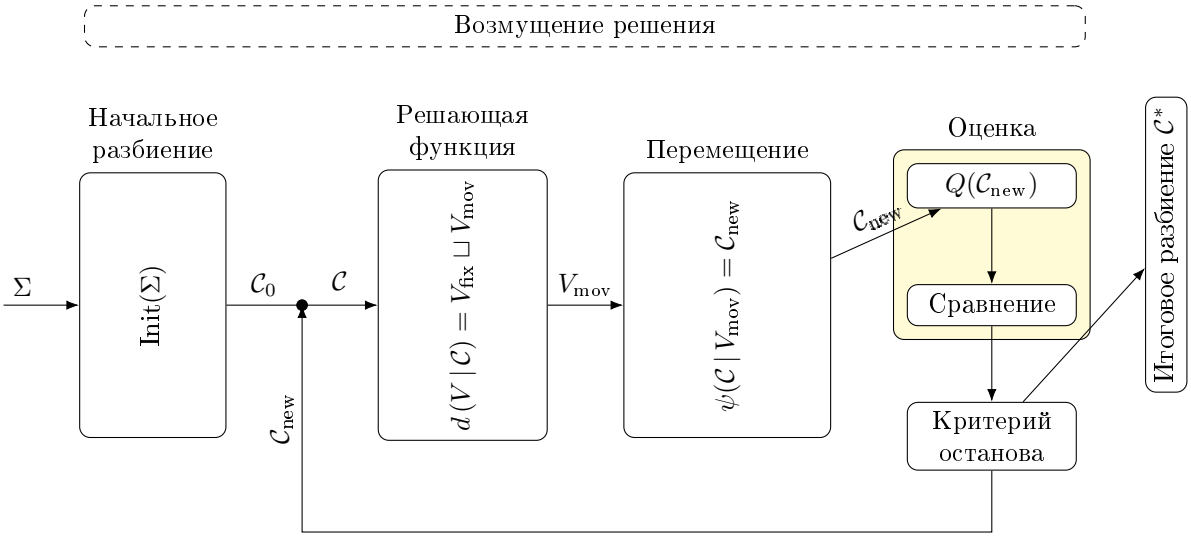


Рис. 1. Общая схема алгоритмов, основанных на структуре графа

Блок «Начальное разбиение» содержит функцию $\mathrm{Init}(\Sigma)$, осуществляющую первоначальное разбиение множества вершин V знакового графа Σ по некоторому правилу. Тривиальным случаем такого разбиения будем считать случайное разбиение V на

фиксированное или нефиксированное количество кластеров. Данный блок зачастую выполняется единожды при запуске алгоритма и формирует первоначальное разбиение $\mathcal{C}_0 \in \Phi$ вида (1).

Блок «Решающая функция» является подготовительным для блока «Перемещение». Решающая функция осуществляет разделение множества вершин V с учётом текущего разбиения $\mathcal{C}$ на два подмножества

$$d\,(V \,|\, \mathcal{C}) = V_{\text{fix}} \sqcup V_{\text{mov}}, \tag{6}$$

где V_{fix} — множество заблокированных для перемещения между кластерами вершин; V_{mov} — множество допустимых для перемещения вершин. Тривиальной будем считать решающую функцию, возвращающую $V_{\text{mov}} = V$.

Блок «Перемещение» содержит функцию $\psi(\mathcal{C} \,|\, V_{\text{mov}})$, которая отвечает за перемещение вершин из множества V_{mov} между кластерами текущего разбиения $\mathcal{C}$ и тем самым формирует новое разбиение $\mathcal{C}_{\text{new}} \in \Phi$. Вершины могут перемещаться не только между существующими кластерами, но и образовывать новые кластеры.

Блок «Оценка» состоит из двух этапов. На первом этапе вычисляется функционал ошибки $Q(\mathcal{C}_{\text{new}})$. На втором этапе проводится сравнение текущего разбиения с ранее найденными по значению функционала ошибки. Данный блок присутствует и в алгоритме, который находит последовательно или параллельно несколько разбиений.

Блок «Критерий останова» позволяет исключить перебор по всему множеству Φ. В качестве критерия останова могут выступать: время, число итераций, значение функционала ошибки, невязка функционала ошибки и т. п.

Блок «Возмущение решения» нацелен на выход из локального минимума функционала ошибки путём перемешивания вершин текущего разбиения, при этом способ перемешивания может определять основную идею алгоритма. Данный блок может следовать после любого другого блока общей схемы и повторяться многократно.

В табл. 1 известные алгоритмы решения задачи корреляционной кластеризации представлены в виде последовательностей блоков из схемы рис. 1. Символами «+»/«−» обозначается соответственно присутствие или отсутствие блока. Используются также следующие обозначения: alg — результат работы другого алгоритма; special — специальным образом; trivial — тривиальный случай; time — время; iter — число итераций; $|V|$ — просмотрены все вершины. В столбце «Перемещение»: i — алгоритм одномоментно перемещает i вершин и при этом может создавать новые кластеры; ① — алгоритму запрещено создавать новые кластеры в процессе перемещения одной вершины. Число вершин i может быть фиксированным в алгоритме либо являться его входным параметром, что обозначается как r.

Рассмотрим структуру алгоритмов из табл. 1 в соответствии со схемой рис. 1.

Алгоритм Relocation heuristic (RH) предложен в [14] и относится к классу эвристических алгоритмов. Количество кластеров разбиения является входным параметром алгоритма. Первоначальное разбиение $\mathcal{C}_0$ строится случайным образом, что соответствует тривиальному случаю блока «Начальное разбиение». Множество допустимых для перемещения вершин V_{mov} между кластерами совпадает со всем множеством вершин V, что соответствует тривиальному случаю для функции (6) в блоке «Решающая функция». Блоку «Перемещение» в табл. 1 соответствует символ ①, что означает перемещение между кластерами ровно одной вершины без образования новых кластеров. Для каждой вершины оцениваются все её возможные перемещения между кластерами. Реализуется перемещение, при котором ошибка будет наименьшей. Процесс повторяется до тех пор, пока не истечёт заданное время.

Таблица 1

Представление алгоритмов решения задачи корреляционной кластеризации по фазам схемы рис. 1

Алгоритм	Авторы	Начальное разбиение	Решающая функция	Перемещение	Критерий останова	Возмущение решения
Relocation heuristic (RH)	P. Doreian, A. Mrvar (1996)	trivial	trivial	①	time	–
Tabu search	M. J. Brusco, P. Doreian (2019)	alg	special	①	time	–
Variable neighborhood search	M. J. Brusco, P. Doreian (2019)	alg	trivial	①	time	+
KwikCluster	N. Ailon, M. Charikar, A. Newman (2008)	special	–	–	$\|V\|$	–
Iterated local search (ILS)	M. Levorato, L. Drummond, Y. Frota, R. Figueiredo (2015)	special	trivial	$1 \dots r$	iter, time	+
$\texttt{SGClust}_\alpha$	Э. И. Ибрагимова, Д. В. Семенова, А. А. Солдатенко (2023)	special	special	1	$\|V\|$	–

Метаэвристический алгоритм Tabu search предложен в [16]. Количество кластеров разбиения является входным параметром алгоритма. Начальное разбиение $\mathcal{C}_0$ является результатом работы алгоритма RH. Во избежании полного перебора авторами вводится решающая функция вида (6), где множество фиксированных вершин V_{fix} определяется списком *tabu*. Список *tabu* формируется из пар (v, it_v), где v — перемещённая на текущей итерации вершина, а it_v — число итераций, на которое вершина v помещается в список *tabu*. Все последующие блоки аналогичны алгоритму RH.

Метаэвристический алгоритм Variable neighborhood search предложен в [16]. Алгоритм представляет собой модификацию RH. Суть модификации состоит в добавлении блока «Возмущение решения» для разбиения, подающегося на вход блоку «Решающая функция», и в изменении блока «Оценка». Решение возмущается следующим образом. Для каждой вершины графа разыгрывается случайная бернуллиевская величина с заданным параметром вероятности успеха *ypert*. В случае наступления успеха данная вершина перемещается из текущего кластера в другой случайный кластер. Далее запускается алгоритм RH с начальным разбиением, соответствующим возмущённому решению. В блоке «Оценка» результат работы RH сравнивается с предыдущим разбиением. Если ошибка не уменьшилась, то вероятность перемещения *ypert* увеличивается на шаг *ystep*, заданный параметром алгоритма, а полученное разбиение забывается. В противном случае решение становится новым текущим разбиением.

Эвристический алгоритм KwikCluster предложен в [3]. Согласно схеме рис. 1, алгоритм содержит только блок «Начальное разбиение». Построение разбиения осуществляется следующим образом: случайно выбирается вершина из множества непросмотренных вершин; вершины, соединённые с ней положительным ребром, помещаются в тот же кластер и отмечаются как просмотренные. Процесс повторяется, пока не бу-

дут просмотрены все вершины. Алгоритм в ходе работы не выполняет дальнейшего перемещения вершин.

Метаэвристический алгоритм Iterated local search (ILS) предложен в [17]. Блок «Начальное разбиение» строится следующим образом. Вводится функция дисбаланса специального вида для ранжирования вершин графа, что позволяет сформировать упорядоченный список вершин. Далее строится α-срез списка вершин, из которого случайным образом выбирается вершина и размещается в кластере согласно функции дисбаланса. Процедура повторяется до тех пор, пока все вершины не будут размещены по заданному числу кластеров. Глубина среза α является входным параметром алгоритма. Блок «Возмущение решения» применяется к разбиению, передаваемому в блок «Решающая функция», и заключается в следующем. Выбирается случайная вершина из случайного кластера и перемещается в другой случайный кластер. Данная процедура выполняется t раз. Множество допустимых для перемещения вершин V_{mov} между кластерами совпадает с множеством вершин V, что соответствует тривиальному случаю для функции (6) в блоке «Решающая функция». В табл. 1 блок «Перемещение» содержит обозначение $1 \dots r$, что соответствует перемещению между кластерами от одной до r вершин с возможностью образования новых кластеров. Перемещения вершин перебираются до тех пор, пока не будет найдено первое улучшение функционала ошибки. Данное перемещение будет результатом блока. Блок «Оценка» сравнивает полученное разбиение с предыдущим. Если ошибка не уменьшилась, то число возмущений t увеличивается на единицу. Возмущения осуществляются до тех пор, пока t не достигнет значения соответствующего параметра алгоритма. Авторы предлагают запускать данный алгоритм многократно, согласно значению параметра iter, либо до истечения заданного времени работы time. В качестве итогового разбиения выбирается наилучшее в смысле функционала ошибки среди всех решений.

Эвристический алгоритм $\texttt{SGClust}_\alpha$ предложен в работах [18, 19]. В качестве первоначального разбиения выбираются компоненты связности в графе Σ^+, который получается из графа Σ путём удаления всех отрицательных рёбер. Решающая функция (6) зависит от внутрикластерной ошибки каждой вершины и обновляемого на каждой итерации закрытого списка *closeList*. Если внутрикластерная ошибка вершины отлична от нуля и вершина не содержится в списке *closeList*, то она помещается в множество V_{mov}. Блоку «Перемещение» в табл. 1 соответствует символ 1, что означает перемещение между кластерами ровно одной вершины с возможностью образования новых кластеров. Вершина с наибольшей внутрикластерной ошибкой выбирается из множества V_{mov}. Данная вершина поочередно присоединяется к каждому из кластеров текущего разбиения, включая пустой. Реализуется перемещение, обеспечивающее наименьшее значение функционала ошибки. Перемещённая вершина добавляется в закрытый список *closeList*. Процесс повторяется, пока не будут просмотрены все вершины из множества V либо на текущей итерации множество допустимых для перемещения вершин не станет пустым ($V_{\text{mov}} = \varnothing$).

3. Алгоритм CaRVeR с потенциальными функциями

Рассмотрим модификацию алгоритма $\texttt{SGClust}_\alpha$, именуемую CaRVeR (***Car****eful* ***Ve****rtex* ***R****elocator*), впервые изложенную в [20]. Суть модификации заключается в применении потенциальных функций в блоке «Решающая функция». Исследуем данный алгоритм согласно блокам схемы рис. 1.

3.1. Начальное разбиение

Алгоритм может работать с любым начальным разбиением, однако целесообразно строить некоторое разбиение, отличное от тривиального случая. В алгоритме используется следующий простой двухшаговый метод построения начального разбиения. На первом шаге строится граф Σ^+ путём удаления всех отрицательных рёбер в исходном графе Σ. Этот шаг требует времени $\mathcal{O}(m)$. На втором шаге алгоритмом поиска в ширину в графе Σ^+ выделяются компоненты связности. Множество вершин в компоненте связности графа Σ^+ является кластером в графе Σ. Множество $\mathcal{C}$ всех таких кластеров образует первоначальное разбиение $\mathcal{C}_0$. Этот шаг требует времени не более чем $\mathcal{O}(n+m)$ [15]. Количество кластеров определяется числом компонент связности $|\mathcal{C}_0| = k(\Sigma^+)$.

Из этих рассуждений вытекает следующая лемма:

Лемма 1. Сложность выполнения блока «Начальное разбиение» для алгоритма `CaRVeR` составляет $\mathcal{O}(n+m)$.

Отличительное свойство такого начального разбиения заключается в том, что оно обладает межкластерной ошибкой $P(\mathcal{C}) = 0$. Данная стратегия обоснована тем, что алгоритм `CaRVeR` в ходе работы уменьшает внутрикластерную ошибку без увеличения суммарной ошибки.

3.2. Решающая функция

Решающая функция (6) зависит от значения потенциальной функции для каждой вершины при текущем разбиении $\mathcal{C}$. Значение потенциальной функции вершины v, принадлежащей кластеру $C \in \mathcal{C}$, будем вычислять следующим образом:

$$\begin{aligned}\pi(v) = \alpha \left(\sum_{u \in \Gamma(v) \cap C} [(v,u) \in E^-] - \sum_{u \in \Gamma(v) \setminus C} [(v,u) \in E^-] \right) + \\ +(1-\alpha) \left(\sum_{u \in \Gamma(v) \setminus C} [(v,u) \in E^+] - \sum_{u \in \Gamma(v) \cap C} [(v,u) \in E^+] \right).\end{aligned} \tag{7}$$

В (7) первое слагаемое определяется как разность текущего вклада вершины v в отрицательную ошибку и числа корректных отрицательных рёбер. Второе слагаемое есть разность текущего вклада вершины v в межкластерную ошибку и числа корректных положительных рёбер. Следовательно, потенциальная функция (7) состоит из межкластерной и внутрикластерной ошибки вершины v и носит следующий смысл: «может ли при идеальных условиях вершина v иметь меньший вклад в ошибку, чем сейчас в кластере C?» Выбор вершины v и её перемещение в другой кластер однозначно увеличат ошибку пропорционально числу корректных положительных рёбер, инцидентных v, в текущей кластеризации, при этом некоторое число корректных отрицательных рёбер может сохраниться.

Таким образом, будем говорить, что вершина v не подлежит перемещению, т. е. $v \in V_{\text{fix}}$, если $\pi(v) < 0$ либо она была перемещена ранее, и $v \in V_{\text{mov}}$, если $\pi(v) \geqslant 0$. Аналогично алгоритму $\texttt{SGClust}_\alpha$, в множестве V_{mov} не могут содержаться вершины из закрытого списка *closeList*.

Для вычисления потенциальных функций $\pi(v)$ для всех вершин $v \in V$ требуется проверить кластер для каждой вершины из окрестности $\Gamma(v)$. Известно, что сумма степеней всех вершин в графе равна $2m$. Тогда справедлива следующая

Лемма 2. Сложность выполнения блока «Решающая функция» для алгоритма `CaRVeR` составляет $\mathcal{O}(m)$.

Отметим, что вид формулы (7) позволяет выполнять пересчёт значения для вершины v только при изменении кластера самой вершины v или смежной с ней вершины.

3.3. Перемещение

Данный блок описывается функцией $\psi(\mathcal{C} \mid V_{\text{mov}})$, которая выбирает единственную вершину $v \in V_{\text{mov}}$ с наибольшим значением потенциальной функции $\pi(v)$. Если таких вершин несколько, то из них выбирается случайная. Далее определяется кластер, в котором вершина v даст наименьший вклад в ошибку. Ошибку вершины v относительно кластера C_i в разбиении $\mathcal{C}$ будем определять по формуле

$$\tau(C_i, v) = \alpha \sum_{u \in \Gamma(v) \cap C_i} [(v,u) \in E^-] + (1-\alpha) \sum_{u \in \Gamma(v) \setminus C_i} [(v,u) \in E^+]\,. \tag{8}$$

Вершина v перемещается в кластер C_i, доставляющий минимум функции $\tau(C_i, v)$.

Вычисление формулы (8) требует не более чем Δ времени для каждого кластера C_i, где Δ — степень графа. Очевидно, что число кластеров не может превосходить числа вершин n. Из этих рассуждений вытекает следующая лемма:

Лемма 3. Сложность выполнения блока «Перемещение» для алгоритма `CaRVeR` составляет $\mathcal{O}(nm)$.

3.4. Оценка и критерий останова

Блок «Оценка» организован стандартным образом и содержит вычисление функционала ошибки (4) на последнем шаге алгоритма.

В алгоритме `CaRVeR` блок «Критерий останова» организован следующим образом. В конце каждой итерации алгоритма обновляется закрытый список *closeList*, в который помещается вершина, выбранная в блоке «Перемещение» на текущей итерации. Останов алгоритма происходит в том случае, если в блоке «Решающая функция» $V_{\text{mov}} = \varnothing$.

Фиксация вершины от дальнейших перемещений выполнима за константное время $\mathcal{O}(1)$, а критерий останова формирует основной цикл алгоритма с числом итераций, не превосходящим n.

Из лемм 1–3 и критерия останова выводится итоговая оценка сложности алгоритма `CaRVeR`.

Теорема 1. Временная сложность алгоритма `CaRVeR` составляет $\mathcal{O}(n^2\, m)$.

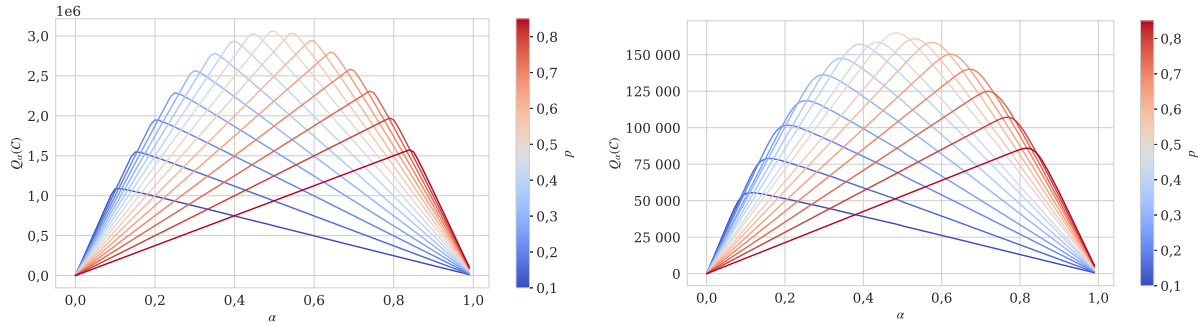
4. Вычислительные эксперименты

Опишем две серии экспериментов на синтетических данных.

Первая серия экспериментов предназначена для изучения поведения алгоритма `CaRVeR` в зависимости от параметра графа p и значения α функционала ошибки (4). Исследования проводились на графах с фиксированным числом вершин, равным 5000. Рассматривались два типа графов: полные графы и графы, смоделированные методом Ваксмена [21] с параметрами 0,15 и 0,4. Знаки рёбер для каждого графа формировались по схеме Бернулли с параметром p из набора $\{0{,}10; 0{,}15; \ldots; 0{,}85\}$. Параметр p обозначает вероятность положительного знака ребра. Значения параметра α функционала ошибки (4) выбирались из диапазона от 0 до 1 с шагом 0,01.

На рис. 2 представлено типичное поведение функционала ошибки в зависимости от значения параметра α для графов с разной долей положительных рёбер p. В каждом случае функционал ошибки $Q_\alpha(\mathcal{C})$ достигает своего максимального значения при $\alpha \approx p$. Рис. 2, *б* отражает поведение функционала ошибки для неполных графов, сгенерированных по методу Ваксмена. Аналогичное поведение характерно для полных

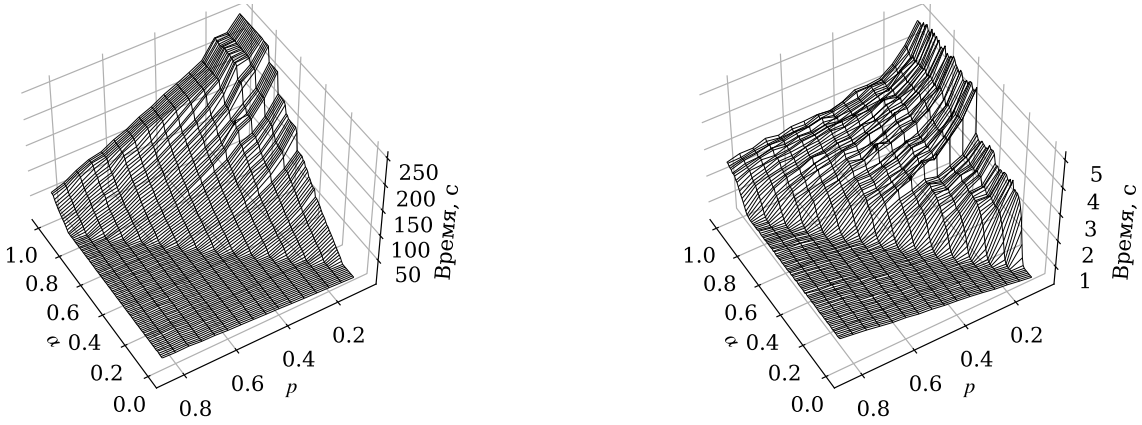
графов (рис. 2, *а*). Стоит отметить, что для неполных графов число рёбер в среднем составляет 708827, а для полных графов оно равно 12497500.



а *б*

Рис. 2. Поведение функционала ошибки алгоритма `CaRVeR` в зависимости от значения параметра α на графах с разной долей положительных рёбер p: *а*) для полных графов; *б*) для графов, сгенерированных по методу Ваксмена

С ростом параметра α трудоёмкость алгоритма `CaRVeR` возрастает, при этом высокая доля положительных рёбер p в графе замедляет рост затрачиваемого времени. На полных графах время выполнения алгоритма существенно возрастает, это следует из специфики блока «Решающая функция», поскольку требуется обновлять окрестность каждой вершины. Данные результаты представлены на рис. 3.

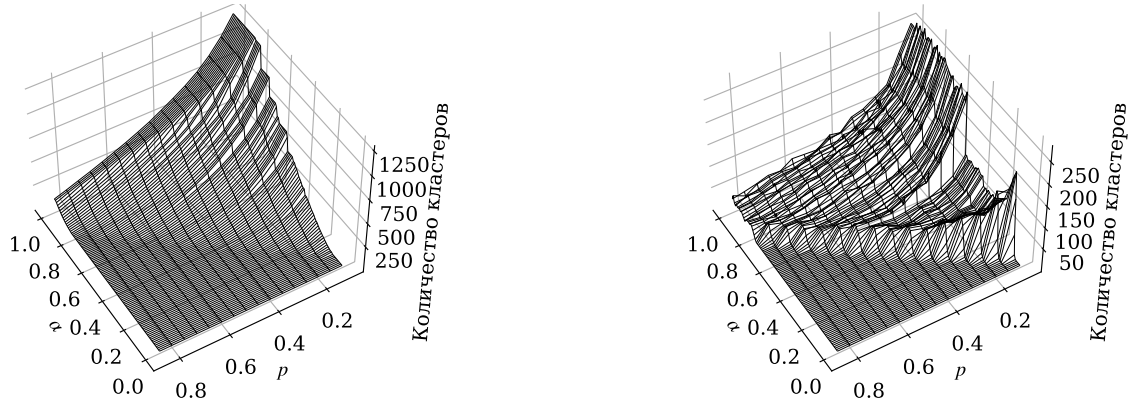


а *б*

Рис. 3. Время работы алгоритма `CaRVeR` при различных параметрах α на графах с разной долей положительных рёбер p: *а*) для полных графов; *б*) для графов, сгенерированных по методу Ваксмена

Число кластеров, выделяемых алгоритмом `CaRVeR`, представлено на рис. 4. Для полных графов число кластеров растёт с уменьшением доли положительных рёбер p (рис. 4, *а*), что естественно для стратегии, применяемой алгоритмом `CaRVeR`.

Вторая серия экспериментов представляет сравнение алгоритма `CaRVeR` с алгоритмами KwikCluster, $\texttt{SGClust}_\alpha$, RH, Tabu search. Алгоритмы сравнивались по времени работы и значению функционала ошибки $Q_\alpha(\mathcal{C})$ при $\alpha = 0{,}5$. Начальное разбиение для алгоритмов RH и Tabu search задавалось случайным образом, а число кластеров — таким же, что было получено алгоритмом `CaRVeR` для данного графа. Сравне-



а *б*

Рис. 4. Количество выделяемых кластеров алгоритмом `CaRVeR` при различных параметрах α на графах с разной долей положительных рёбер p: *а*) для полных графов; *б*) для графов, сгенерированных по методу Ваксмена

ние проводилось на графах с разной долей положительных рёбер $p = \{0{,}25; 0{,}5; 0{,}75\}$. Для каждой доли p были сгенерированы 100 графов методом Ваксмена с параметрами 0,15 и 0,4.

Результаты сравнения алгоритмов по времени представлены в табл. 2. Алгоритм `CaRVeR` ведёт себя достаточно стабильно по времени выполнения. Заметим, что представленные алгоритмы работают значительно быстрее на графах с большей долей положительных рёбер p.

Таблица 2

Сравнение алгоритмов по времени работы

p	Алгоритм	Среднее время, с	Минимальное время, с	Максимальное время, с	Среднеквадратичное отклонение
0,25	`CaRVeR`	0,016	0,015	0,018	0,001
	KwikCluster	0,001	0,001	0,001	0
	RH	30,667	26,145	36,995	2,039
	$\texttt{SGClust}_\alpha$	0,018	0,017	0,02	0,001
	Tabu search	71,322	60,041	79,819	5,312
0,5	`CaRVeR`	0,008	0,007	0,009	0,001
	KwikCluster	0,001	0,001	0,001	0
	RH	16,158	10,642	19,663	1,623
	$\texttt{SGClust}_\alpha$	0,016	0,014	0,017	0,001
	Tabu search	64,701	60,031	69,755	2,892
0,75	`CaRVeR`	0,004	0,004	0,005	0,001
	KwikCluster	0,001	0,001	0,001	0
	RH	1,636	0,405	3,221	0,587
	$\texttt{SGClust}_\alpha$	0,008	0,007	0,01	0,001
	Tabu search	60,294	60,0	60,981	0,242

На рис. 5 и 6 представлена скрипичная диаграмма значений функционала ошибки $Q_\alpha(\mathcal{C})$. Из рис. 5 видно, что на полных графах алгоритм $\texttt{SGClust}_\alpha$ и его модификация `CaRVeR` в среднем демонстрируют наилучший результат среди тестируемых алгоритмов. Аналогичный вывод справедлив и для графов, сгенерированных методом

Ваксмена (рис. 6). Следует отметить, что алгоритм KwikCluster демонстрирует удовлетворительные результаты по значению функционала ошибки на неполных графах и, учитывая его высокую скорость работы, может быть использован как алгоритм первоначального разбиения. Наибольший разброс в значениях функционала ошибки показывает алгоритм Tabu search. Алгоритмы $\texttt{SGClust}_\alpha$ и `CaRVeR` менее чувствительны к изменению доли положительных рёбер p, что видно из рис. 6.

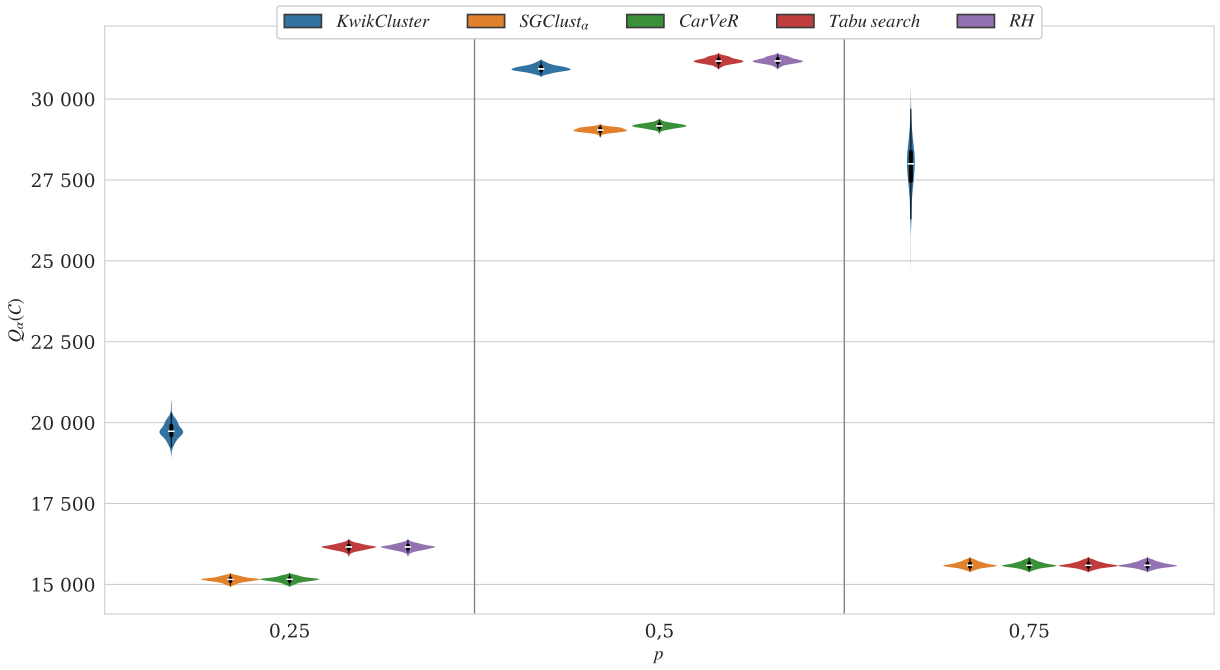


Рис. 5. Сравнение алгоритмов по функционалу ошибки $Q_\alpha(\mathcal{C})$ для полных графов

На рис. 7 и 8 отражены средние доли в процентах межкластерной $P(\mathcal{C})$ и внутрикластерной $N(\mathcal{C})$ ошибок в найденном решении, что в некотором смысле характеризует стратегию тестируемых алгоритмов. В алгоритме `CaRVeR`, в сравнении с алгоритмом $\texttt{SGClust}_\alpha$, происходит балансировка внутрикластерной и межкластерной ошибок, что обусловлено спецификой потенциальной функции (рис. 7 и 8). Для полных графов (рис. 7) алгоритм KwikCluster находит в среднем равные доли межкластерной и внутрикластерной ошибок, а для неполных (рис. 8) доля межкластерной ошибки значительно превосходит долю внутрикластерной ошибки.

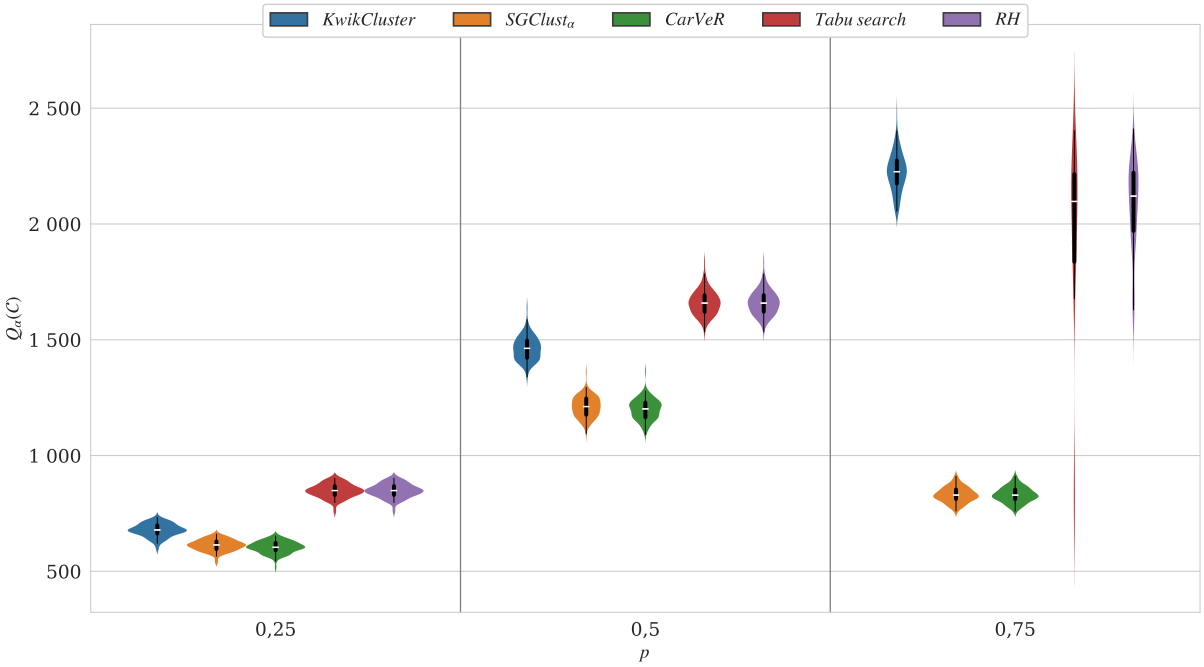


Рис. 6. Сравнение алгоритмов по функционалу ошибки $Q_\alpha(\mathcal{C})$ для графов, сгенерированных методом Ваксмена

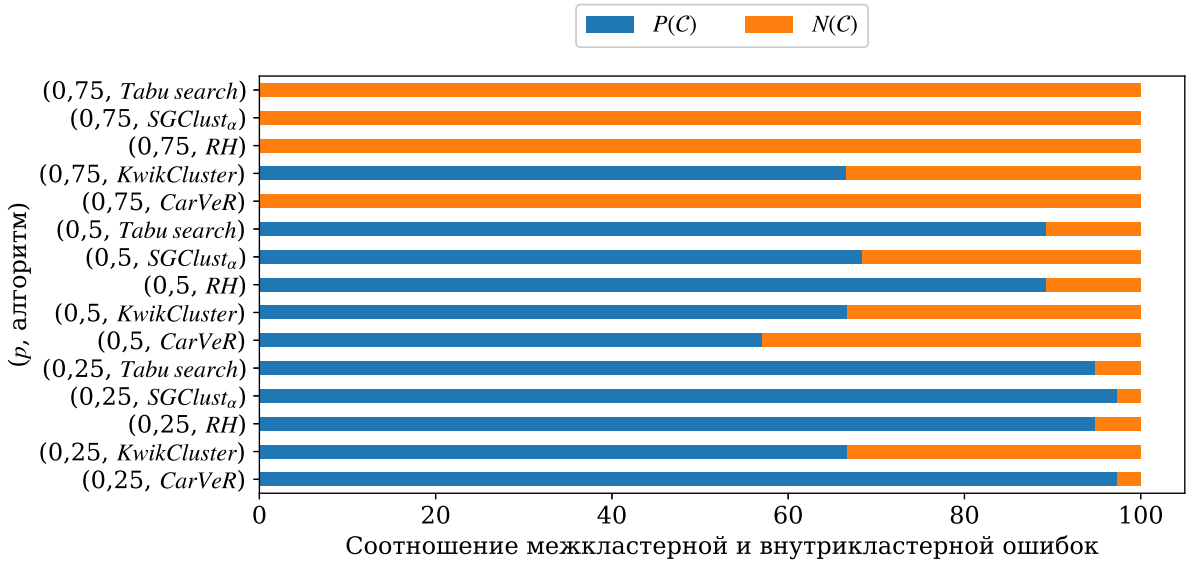


Рис. 7. Сравнение алгоритмов по долям p в процентах межкластерной $P(\mathcal{C})$ и внутрикластерной $N(\mathcal{C})$ ошибок для найденного решения на полных графах

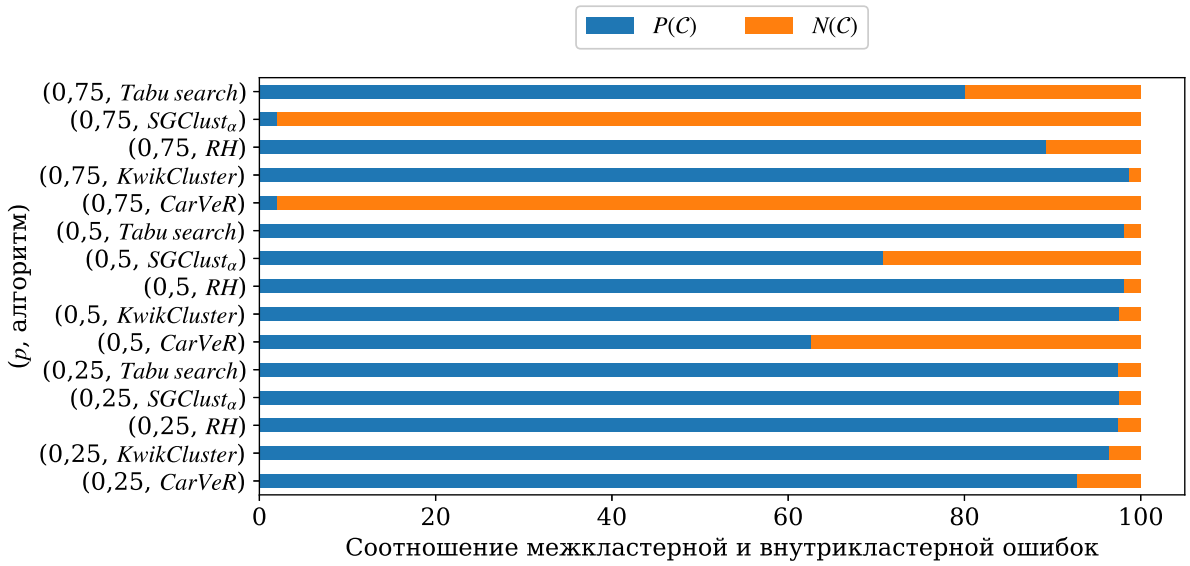


Рис. 8. Сравнение алгоритмов по долям p в процентах межкластерной $P(\mathcal{C})$ и внутрикластерной $N(\mathcal{C})$ ошибок для найденного решения на графах, сгенерированных методом Ваксмена

Заключение

Проведённый системный анализ совокупности алгоритмов решения задачи корреляционной кластеризации, основанных на структуре знакового графа, позволил выявить общую концептуальную схему конструирования и анализа таких алгоритмов. Компонентами схемы являются шесть основных блоков, характеризующих возможные этапы решения, что продемонстрировано на примере описания известных алгоритмов KwikCluster, $\texttt{SGClust}_\alpha$, Relocation heuristic, Tabu search, Variable neighborhood search, Iterated local search. Такой общий взгляд на структуру алгоритмов позволяет модифицировать существующие алгоритмы путём внесения изменений в один или несколько блоков схемы. Предложен новый алгоритм `CaRVeR`. Согласно общей схеме, он является улучшенной модификацией алгоритма $\texttt{SGClust}_\alpha$ в блоке «Решающая функция». Более того, топология общей схемы открывает возможности для анализа и доказательства вычислительной сложности алгоритмов, что представлено в теореме 1.

Серии вычислительных экспериментов на синтетических данных показали результативность алгоритма `CaRVeR` в сравнении с алгоритмами KwikCluster, $\texttt{SGClust}_\alpha$, Relocation heuristic, Tabu search как по времени работы, так и по значению функционала ошибки.

Перспективны дальнейшие исследования потенциальных функций на блоке «Решающая функция» общей схемы, а также формирование базового алгоритма, содержащего все блоки общей схемы, для осуществления различных модификаций любого из блоков.

ЛИТЕРАТУРА

1. *Il'ev V., Il'eva S., and Kononov A.* Short survey on graph correlation clustering with minimization criteria // LNCS. 2016. V. 9869. P. 25–36.
2. *Wahid D. F. and Hassini E.* A literature review on correlation clustering: Cross-disciplinary taxonomy with bibliometric analysis // Oper. Res. Forum. 2022. V. 3. Article 47.

3. *Ailon N., Charikar M., and Newman A.* Aggregating inconsistent information: Ranking and clustering // J. ACM. 2008. V. 55. Iss. 5. Article 23. P. 1–27.
4. *Demaine E. D. and Immorlica N.* Correlation clustering with partial information // LNCS. 2003. V. 2764. P. 1–13.
5. *Swamy C.* Correlation clustering: Maximizing agreements via semidefinite programming // Proc. SODA'04. New Orleans, Louisiana, 2004. P. 526–527.
6. *Bonizzoni P., Vedova D. G., Dondi R., and Jiang T.* Correlation clustering and consensus clustering // LNCS. 2005. V. 3827. P. 226—235.
7. *Figueiredo R. and Moura G.* Mixed integer programming formulations for clustering problems related to structural balance // Social Networks. 2013. No. 35. P. 639–651.
8. *Queiroga E., Subramanian A., Figueiredo R., and Frota Yu.* Integer programming formulations and efficient local search for relaxed correlation clustering // J. Glob. Optim. 2021. No. 81. P. 919–966.
9. *Doreian P. and Mrvar A.* Partitioning signed social networks // Soc. Networks. 2009. No. 31. P. 1–11.
10. *Graham R., L., Knuth D., E., and Patashnik O.* Concrete Mathematics: A Foundation for Computer Science. 2nd ed. Massachusetts, USA: Addison-Wesley, 1994. 657 p.
11. *Cartwright D. and Harary F.* Structural balance: A generalization of Heider's theory // Psychol. Rev. 1956. V. 63. No. 5. P. 227–293.
12. *Doreian P. and Mrvar A.* Structural Balance and Partitioning Signed Graphs. `http://mrvar2.fdv.si/pajek\/SignedNetworks/Bled94.pdf`. 1996.
13. *Bansal N., Blum A., and Chawla S.* Correlation clustering // Machine Learning. 2004. No. 56. P. 89–113.
14. *Doreian P. and Mrvar A.* A partitioning approach to structural balance // Soc. Networks. 1996. No. 18. P. 149–168.
15. *Kormen T. H., Leiserson C. E., Rivest R. L., and Stein C.* Introduction to Algorithms. 3rd ed. Cambridge: MIT Press, 2009. 1312 p.
16. *Brusco M. J. and Doreian P.* Partitioning signed networks using relocation heuristics, tabu search, and variable neighborhood search // Soc. Networks. 2019. No. 56. P. 70–80.
17. *Levorato M., Figueiredo R., Frota Yu., and Drummond L.* Evaluating balancing on social networks through the efficient solution of correlation clustering problems // EURO J. Comput. Optim. 2017. V. 5. P. 467–498.
18. *Ibragimova E., Semenova D., and Soldatenko A.* Comparison of two heuristic algorithms for correlation clustering problem solving // 5th Intern. Conf. PCI. Baku, Azerbaijan, 2023. P. 1–4.
19. *Soldatenko A., Semenova D., and Ibragimova E.* On heuristic algorithm with greedy strategy for the correlation clustering problem solution // LNCS. 2024. V. 14123. P. 462–477.
20. *Солдатенко А. А., Семенова Д. В., Ибрагимова Э. И.* Алгоритм с потенциальными функциями для задачи разбиения знаковых графов // Информационные технологии и математическое моделирование. Томск, 2023. С. 238–244.
21. *Waxman B. M.* Routing of multipoint connections // IEEE J. Selected Areas Commun. 1988. V. 6. No. 9. P. 1617–1622.

REFERENCES

1. *Il'ev V., Il'eva S., and Kononov A.* Short survey on graph correlation clustering with minimization criteria. LNCS, 2016, vol. 9869, pp. 25–36.
2. *Wahid D. F. and Hassini E.* A literature review on correlation clustering: Cross-disciplinary taxonomy with bibliometric analysis. Oper. Res. Forum, 2022, vol. 3, article 47.

3. *Ailon N., Charikar M., and Newman A.* Aggregating inconsistent information: Ranking and clustering. J. ACM, 2008, vol. 55, iss. 5, article 23, pp. 1–27.
4. *Demaine E. D. and Immorlica N.* Correlation clustering with partial information. LNCS, 2003, vol. 2764, pp. 1–13.
5. *Swamy C.* Correlation clustering: Maximizing agreements via semidefinite programming. Proc. SODA'04, New Orleans, Louisiana, 2004, pp. 526–527.
6. *Bonizzoni P., Vedova D. G., Dondi R., and Jiang T.* Correlation clustering and consensus clustering. LNCS, 2005, vol. 3827, pp. 226—235.
7. *Figueiredo R. and Moura G.* Mixed integer programming formulations for clustering problems related to structural balance. Soc. Networks, 2013, no. 35, pp. 639–651.
8. *Queiroga E., Subramanian A., Figueiredo R., and Frota Yu.* Integer programming formulations and efficient local search for relaxed correlation clustering. J. Glob. Optim., 2021, no. 81, pp. 919–966.
9. *Doreian P. and Mrvar A.* Partitioning signed social networks. Soc. Networks, 2009, no. 31, pp. 1–11.
10. *Graham R., L., Knuth D., E., and Patashnik O.* Concrete Mathematics: A Foundation for Computer Science. 2nd ed. Massachusetts, USA, Addison-Wesley, 1994. 657 p.
11. *Cartwright D. and Harary F.* Structural balance: A generalization of Heider's theory. Psychol. Rev., 1956, vol. 63, no. 5, pp. 227–293.
12. *Doreian P. and Mrvar A.* Structural Balance and Partitioning Signed Graphs. `http://mrvar2.fdv.si/pajek\/SignedNetworks/Bled94.pdf`, 1996.
13. *Bansal N., Blum A., and Chawla S.* Correlation clustering. Machine Learning, 2004, no. 56, pp. 89–113.
14. *Doreian P. and Mrvar A.* A partitioning approach to structural balance. Soc. Networks, 1996, no. 18, pp. 149–168.
15. *Kormen T. H., Leiserson C. E., Rivest R. L., and Stein C.* Introduction to Algorithms. 3rd ed. Cambridge, MIT Press, 2009. 1312 p.
16. *Brusco M. J. and Doreian P.* Partitioning signed networks using relocation heuristics, tabu search, and variable neighborhood search. Soc. Networks, 2019, no. 56, pp. 70–80.
17. *Levorato M., Figueiredo R., Frota Yu., and Drummond L.* Evaluating balancing on social networks through the efficient solution of correlation clustering problems. EURO J. Comput. Optim., 2017, vol. 5, pp. 467–498.
18. *Ibragimova E., Semenova D., and Soldatenko A.* Comparison of two heuristic algorithms for correlation clustering problem solving. 5th Intern. Conf. PCI, Baku, Azerbaijan, 2023, pp. 1–4.
19. *Soldatenko A., Semenova D., and Ibragimova E.* On heuristic algorithm with greedy strategy for the correlation clustering problem solution. LNCS, 2024, vol. 14123, pp. 462–477.
20. *Soldatenko A. A., Semenova D. V., Ibragimova E. I.* Algoritm s potentsial'nymi funktsiyami dlya zadachi razbieniya znakovykh grafov [Algorithm with potential functions for the problem of partitioning signed graphs]. Informatsionnye Tekhnologii i Matematicheskoe Modelirovanie, Tomsk, 2023, pp. 238–244. (in Russian)
21. *Waxman B. M.* Routing of multipoint connections IEEE J. Selected Areas Commun.. 1988, vol. 6, no. 9, pp. 1617–1622.