Санкт-Петербургский государственный университет

Кафедра Системного Программирования

Киряновский Илья Леонидович

Рандомизированные алгоритмы кластеризации на графах

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., профессор Граничин О. Н.

Рецензент:
к. ф.-м. н. Иванский Ю. В.

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Department of Software Engineering

Ilia Kirianovskii

Randomized Graph Clustering Algorithms

Graduate Qualification Work

Scientific supervisor:
professor Oleg Granichin

Reviewer:
PhD Yuri Ivanskiy

Saint-Petersburg
2019

Оглавление

Введение

Многие сложные системы, вознакающие в естественных, социальных, или технических науках, могут быть естественно представленны в виде сети или *графа*, где узлы представляют элементарные единицы системы, а дуги описывают отношения или *связь* между ними. Примерами таких систем являются World Wide Web (WWW) и Интернет, сети цитирования, научного сотрудничества (англ. collaboration network), экологические сети, пищевые сети, сотовые и молекулярные группы, континентальные энергосистемы, транспортные сети, и прочее [3, 58, 57]. Типичным подходом для анализа таких систем, представленных в виде графа, является выделение так называемых *сообществ* или *кластеров*.

Кластером называют такое подмножество вершин, которые более "тесно" связаны друг с другом, чем с другими вершинами из остальной сети. Как показано в [33, 53, 38] и других, многие примеры сетей из реальной жизни имеют скрытую внутреннюю структуру, соответствующую её естественной природе, и отражающую особенности её внутрисетевого взимодействия. Знание о таких структурах позволяет лучше объяснять различные явления присущие рассматриваемой системе, прогнозировать поведение социальных групп [24], анализировать динамику распространения компьютерных вирусов [4], выявлять общие метаболические пути у разных видов хозяев паразитов [53], находить тематически связанные страницы в Интернете [46, 78], и др.

Алгоритмы кластеризации на графах, также *поиска сообществ*, достаточно давно получили широкое распространение. Одной из первых работ по этой теме была работа С. А. Риса (S. A. Rice) 1927 года по обнаружению групп в органах государственной власти [71]. Впоследствии были предложены разные методы кластеризации (Керниган и Лин (Kernighan and Lin) [42], Суарис и Кедем (Suaris and Kedem) [82], Барнес (Barnes) [6], Форд и Фулкерсон (Ford and Fulkerson) [28], Флейк и др. (Flake et al.) [23, 78], Посен (Pothen) [65], Боллобас (Bollobas) [9], Хасти и др. (Hastie et al) [41], МакКвин (MacQueen) [49], Ратиган и

др. (Rattigan et al.) [68], Шенкер и др. (Schenker et al.) [36], Бездек (Bezdek) [7], Dunn [22], Фидлер (Fiedler) [27], Ши и Малик (Shi and Malik) [79, 80]), в том числе особенную популярность получили алгоритмы, основанные на оптимизации некоторой целевой функции качества [26, 15, 20, 90, 70, 56, 59, 66, 76, 85, 88, 39, 51, 52, 21, 48, 57, 74, 87].

С практической точки зрения, необходимость понимания внутренней структуры графа и его разбиения на кластеры становится всё более актуальной по мере увеличения количества обрабатываемых данных. В то же время, тенденция на решение задач и анализ данных в реальном времени (реклама, поиск преступников, рекомендательные системы, энергетические системы, и пр. [25, 37, 54, 16, 18, 77]) требует увеличения скорости работы таких алгоритмов. Однако, в общем виде задача кластеризации относится к классу трудоемких переборных задач, сложность которых повышается при увеличении количества узлов и связей между ними. Это приводит к необходимости применения новых подходов и создания новых, более эффективных, методов кластеризации.

Большое распространение получил иерархический алгоритм кластеризации *Louvain Method*, предложенный Блондель и др. (Blondel et al.) в 2008 году [26], который способен обработать большой объем разреженных данных за почти линейное время [50]. В работах [86, 19, 17, 29, 73] были рассмотрены улучшенные версии и модификации этого алгоритма. Тем не менее, как было отмечено например в [26, 17], для графов содержащих миллиарды связей, такая обработка требует большого количества временных или материальных затрат. Учитывая скорость развития сложных систем и стремительное увеличение количества данных, все еще актуальной является разработка новых алгоритмов, которые должны работать быстрее предыдущих, или не уступать им по качеству, что актуализирует направление выпускной квалификационной работы.

Рандомизированные алгоритмы часто применяются для решения сложных, трудоемких задач, и обладают существенными преимуществами по сравнению с детерминированными методами. В частности, в

работе О.Н. Граничина и др. [35] показано, что значительное ускорение многих алгоритмов обработки данных может быть достигнуто с помощью рандомизации. Такие методы продолжают активно развиваться и рассмотрены в работах Д. Калафиоре (G. Calafiore) и Б. Т. Поляка [11], О.Н. Граничина и Б.Т. Поляка [92], Р. Темпо и др. (R. Tempo et al.) [83], О.Н. Граничина и др. [35]. Важно отметить, что в случае использования таких алгоритмов для трудоемких переборных задач, хороший результат может быть получен с определенной вероятностью за ограниченное время.

Цель работы. Исследование и разработка алгоритмов кластеризации графов, работоспособных на больших объёмах данных.

Для достижения этой цели были поставлены и решены следующие задачи:

1. исследовать возможность оптимизации *state-of-the-art* Louvain метода с помощью рандомизированного подхода;

2. исследовать возможность использования алгоритма стохастической апроксимации для адаптивного выбора входных параметров у методов CGGC and RG;

3. провести апробацию новых разработанных и модифицированных алгоритмов на тестовых наборах данных.

Методы исследования. В выпускной работе используются методы теории оценивания, оптимизации, управления, графов, вероятностей и математической статистики; применяются стохастическая апроксимация, рандомизированные алгоритмы; используются методы кластеризации, основанные на максимизации функции качества.

Основные результаты. В ходе выполнения работы были получены следующие научные результаты:

1. предложена рандомизированная модификация Louvain метода, основанная на случайном выборе и проверке заданного количества соседних узлов;

2. предложены адаптивные модификации Core Groups Graph Cluster и Randomized Greedy методов, основанные на алгоритме стохастической апроксимации;

3. произведена апробация полученных алгоритмов, и сделано их сравнение со стандартными методами.

Научная новизна. Все основные научные результаты выпускной работы являются новыми.

Теоретическая ценность и практическая значимость. Теоретическая ценность результатов заключается в разработке рандомизированной модификации Louvain метода, основанного на случайном выборе ограниченного количества соседних узлов, который работает значительно быстрее стандартного подхода, и лишь с незначительной погрешностью; в предложении и разработке адаптивных модификаций Core Groups Graph Cluster и Randomized Greedy методов, в основе которых лежит алгоритм стохастической апроксимации SPSA, испытания для которых так же показали их эфективность.

Предложенные методы могут использоваться во многих областях науки для анализа и кластеризации различных больших графов, состоящих даже из миллиардов узлов, и имеющих совершенно разные внутренние особенности и структуры.

Апробация работы. Результаты выпускной работы докладывались на семинарах кафедр системного программирования и теоретической кибернетики математико-механического факультета СПбГУ, на конференции IFAC Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON‘15) (June 24 – 26, 2015, Saint Petersburg, Russia), на Восьмой традиционной всероссийской молодежной летней школе “Управление, информация и оптимизация” (пос. Репино, г. Санкт-Петербург, Россия, 14–19 июня, 2016), на конференциях 12th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP‘16) (June 29 – July 1, 2016, Eindhoven, The Netherlands), 55th IEEE Conference on Decision and Control (CDC‘17) (December 12 – 15, 2017, Las Vegas, USA), International Symposium of New Techniques in

Medical Diagnosis and Treatment (June 1–3, 2017, Wuhan, China).

Результаты выпускной работы были использованы в работах по грантам РФФИ 14-08-01015 “Адаптивное управление в стохастических сетях с запаздыванием и потерей данных”, РФФИ 16-07-00890 “Рандомизированные алгоритмы в автоматическом управлении и при извлечении знаний”.

Публикация результатов. Результаты, полученные в выпускной работе, нашли отражение в трёх научных работах [44, 45, 67], все из которых опубликованы в изданиях, индексируемых в базе данных Scopus. Две работы [45, 67] содержат основные результаты работы и опубликованы в периодических изданиях.

Структура и объем выпускной работы. Выпускная работа состоит из введения, трёх глав, заключения, списка литературы, включающего 93 источника. Текст занимает 44 страницы, содержит 5 рисунков и 7 таблиц.

Содержание работы.

Во **введении** обосновывается актуальность выпускной квалификационной работы, формулируется цель, ставятся задачи исследования и кратко излагаются результаты.

В **первой главе** приводится обзор литературы по теме исследования, в частности перечисляются основные методы кластеризации на графах, и выделяются их основные группы.

В разделе 1.1 описывается задача оценки качества кластеризации, приводятся основные понятия и обозначения, вводится функция качества модулярность, а также излагаются ее свойства, преимущества, и недостатки.

В разделе 1.2 формулируется задача кластеризации в терминах максимизации модулярности, а также приводятся ее постановки в виде задачи целочисленного линейного программирования и выпуклой задачи.

Разделе 1.3 содержит описания нескольких известных методов кластеризации, используемых в работе.

Во **второй главе** дан краткий обзор развития методов стохастической апроксимации, предложены рандомизированные и адаптивные

модификации рассматриваемых алгоритмов.

В разделе 2.1 представлена рандомизированная модификация Louvain метода на основе похода, применённого в Randomized Greedy алгоритме.

В разделе 2.2 приводится выпуклая функция качества, обосновывается ее применение в алгоритме стохастической апроксимации, описываются адаптивные модификации алгоритмов Randomized Greedy и Core Groups Graph Cluster.

В **третьей главе** описывается методика тестирования и приводятся результаты сравнения полученных модификаций с оригинальными алгоритмами.

В **заключении** формулируются основные результаты выпускной работы.

1. Методы кластеризации на графах

Алгоритмы кластеризации предназначены для выделения подгрупп, и в некоторых случаях, определения их иерархической структуры внутри сети. Часто такого рода информация может быть извлечена напрямую из самого графа, учитывая особенности отношения узлов между собой. В современной литературе можно выделить следующие группы таких алгоритмов: алгоритмы, основанные на поиске и удалении межсообщественных связей (Girvan and Newman, 2002; Radicchi et al., 2004); алгоритмы спектрального разбиения, основанные на вычислении собственных векторов матриц графа (Donetti and Munoz, 2004; Jin, 2015); статистические методы (Guimera and Sales-Pardo, 2009; Karrer and Newman, 2011; Peixoto, 2014); алгоритмы, основанные на оптимизации некоторой функции качества (Newman, 2004; Huang et al., 2011; Lancichinetti et al., 2011); и динамические методы (Rosvall and Bergstrom, 2008; Pons and Latapy, 2005). С более полным обзором методов кластеризации можно ознакомиться в работах Фортунато 2010 года [30], и Фортунато и Хрика 2016 года [32].

Кроме того, выделяется отдельная группа алгоритмов, предназначенная для поиска пересекающихся сообществ. Такие сообщества допускают, что узлы могут одновременно принадлежать нескольким кластерам. Тем не менее, в этой работе будут рассматриваться только алгоритмы поиска не пересекающихся сообществ.

1.1. Оценка качества кластеризации. Модулярность.

Одной из важных проблем кластеризации являлась разработка способов оценки качества получившегося разбиения. Было видно, что существующие алгоритмы справляются с задачей восстановления разбиения на искуственно синтезированных данных, или данных, где структура сообщества была заранее известна. Но большое практическое применение этой задачи подразумевало, что она будет использоваться на графах, где структура заранее не задана, или даже вовсе отсутствует. Таким образом, качество работы алгоритмов можно было бы оценивать

не только на известных заранее размеченных данных, но и на реальных примерах из жизни.

Пусть $G = (V, E)$ — граф, в котором $V \neq \emptyset$ обозначает множество вершин, а $E \neq \emptyset$ — множество ребер. Положим, что n и m — это количество элементов множеств V и E соответственно.

Обозначим за $C_i,\ i = 1, \ldots, k$ такое разбиение множества узлов V, что:

$$\bigcup_{i=1}^{k} C_i = V \qquad \forall i, j \in \{1, \ldots, k\},\ i \neq j \quad . \tag{1}$$

Такое разбиение графа называется кластерами или сообществами, а процесс их поиска называется задачей кластеризации, или обнаружения сообществ графа. Сообщества C_i и C_j будем называть соседними друг для друга, если между ними существует хотя бы одно ребро.

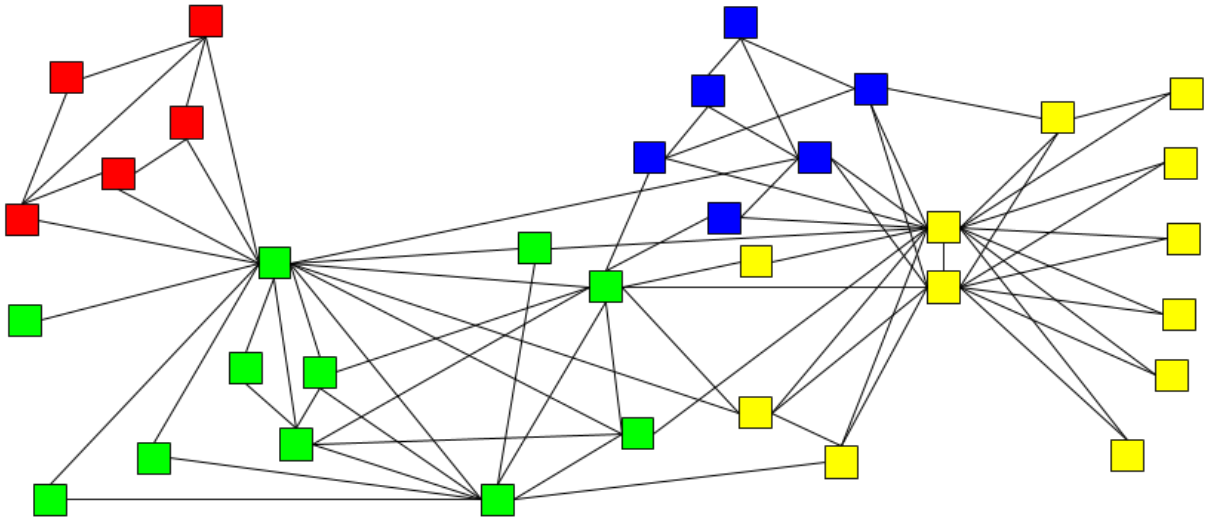
Для решения этой задачи были предложены разные методы оценки кластеров, но наибольшую популярность получил метод вычисления целевой функции *модулярность* (англ. *modularity*), предложенный Ньюманом и Гриваном в 2004 году. Ньюман и Гриван положили, что графы имеют строгую структуру сообществ если вершины внутри кластера более тесно связаны между собой чем с вершинами из остальной сети (см. Рис. 1).

Рассмотрим нормализированную матрицу смежности $e_{ij},\ i, j = 1, \ldots, k$, в которой e_{ij} это отношение количества ребер между кластерами C_i и C_j, к общему количеству ребер в графе. И пусть $a_i = \sum_{j \in 1,\ldots,k} e_{ij}$ — отношение количества ребер, связанных с кластером C_i к общему количеству ребер в графе. Тогда, функция модулярность Q будет вычисляется следующим образом:

$$Q(G, C) = \sum_{i \in 1,\ldots,k} \left(e_{ii} - a_i^2\right) \quad . \tag{2}$$

Как отмечается в [64], модулярность представляется в виде разницы между фактической плотностью ребер внутри кластера и их ожидаемой плотностью в произвольном графе с такой же степенью вершин

Рис. 1: Сеть друзей из 34 членов клуба карате в университете США в 1970-х годах, известная как *Zachary's karate club* [89]. Цветами показано разбиение на кластеры, полученное с помощью максимизации модулярности.



(англ. random null network). Значение этой функции лежит в интервале от -0.5 до 1 (больше — лучше) [60], и измеряет качество кластеризации в терминах того, что ребра более плотно лежат внутри какого-то сообщества чем между ними. Таким образом, если количество ребер лежащих внутри кластера не лучше чем в произвольном графе, то значение этой функции будет близко к 0, а на графах с ярко выраженной структурой сообществ оно будет приближаться к 1.

Одним из приемуществ этой функции является то, что ее можно использовать для автоматического определения количества кластеров в графе. Поэтому функция быстро приобрела общее признание и популярность, и активно дальше исследовалась. Например в работах (Barber, 2007; MacMahon and Garlaschelli, 2013; Traag and Bruggeman, 2009) были рассмотрены способы выбора других произвольных графов.

Кроме того, появилась новая категория алгоритмов кластеризации данных, основанная на идеи поиска кластеров, дающих максимальное значение модулярности: (Blondel et al., 2008; Guimera et al., 2004; Clauset et al., 2004; Newman, 2006; Danon et al., 2006; Pujol et al., 2006; Wakita and Tsurumi, 2007; Arenas et al., 2008) .

Также, модулярность была сформулирована для случая взвешенного графа. Пусть A — это матрица смежности, в которой A_{ij} обозначает вес ребра между вершинами i и j, или равно 0 если такого ребра нет. И пусть $d_i = \sum_{j \in 1,\ldots,n} A_{ij}$ — сумма всех ребер узла i. Тогда модулярность может быть вычислена следующим образом [55]:

$$Q = \frac{1}{2m} \sum_{i,j \in 1,\ldots,n} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j) \quad , \tag{3}$$

где $m = \frac{1}{2} \sum_{i,j \in 1,\ldots,n} A_{ij}$, C_i обозначает кластер узла i, и $\delta(u, v)$ — это символ Кронекера.

Однако, у этого подхода тоже есть минусы: модулярность плохо определяет группы маленького размера относительно размера всей сети, даже если эти группы хорошо определены. Это стало известно как *resolution limit problem* [31]. В дальнейшем для решения этой проблемы разрабатывались новые техники, основанные на модулярности: [31, 75, 69, 5, 40]. А также, были предложены новые функции качества: *Weighted Modularity* [40], Modularity with Split Penalty [14], Modularity Density [13] и др.

1.2. Постановка задачи кластеризации

Пусть для решения задачи кластеризации используется функция модулярность. Тогда такую задачу кластеризации можно сформулировать следующим образом:

$$\underset{C_i \in \mathfrak{C}}{\text{maximize}} \quad Q(G, C_i) \quad , \tag{4}$$

где $\mathfrak{C}$ — это множество всевозможных разбиений графа G на кластеры.

Однако, эта задача относится к классу NP-трудных задач [60].

Также Брендес и др. (Brandes et al.) [60], одновременно с Агарвал и Кемпе (G. Agarwal and D. Kempe) [2] показали, что эта задача может быть переформулирована в виде задачи целочисленного линейного

программирования:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2m}\sum_{u,v} m_{u,v}(1 - x_{u,v}) \\ \text{subject to} \quad & x_{u,w} <= x_{u,v} + x_{v,w} \quad \forall u, v, w \in V \quad , \\ & x_{u,v} \in \{0, 1\} \quad , \end{aligned} \tag{5}$$

где $m_{u,v} := A_{u,v} - \frac{deg(u)deg(v)}{2}$, и $x_{u,v}$ равно 0 если u и v относятся к одному кластеру, иначе 1.

Кроме того, в [2] была сформулирована выпуклая задача максимизации модулярности, но только для случая с двумя сообществами:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{4m}\sum_{u,v} m_{u,v}(1 + y_u y_v) \\ \text{subject to} \quad & y_v^2 = 1 \quad \forall v \in V \quad , \end{aligned} \tag{6}$$

где y_v принимает значение -1 если узел v принадлежит первому сообществу, или 1 если второму.

В последствии, в [12] было предложено обобщение выпуклой формулировки для случая с произвольным заданным количеством кластеров. Несмотря на существенное ограничение на заданное количество искомых сообществ, такая формулировка позволяет подойти к решению этой задачи с другой стороны [84].

1.3. Иерархические и рандомизированные алгоритмы

В задаче поиска сообществ особое внимание и наибольшее распространение получили иерархические методы, основанные на максимизации модулярности.

Louvain Method

В 2008 году Блондель и др. (Blondel et al.) предложили один из наиболее известных эффективных алгоритмов максимизации модуляр-

ности, так называемый Louvain метод [26].

Основная идея метода заключается в выполнении и повторении следующих этапов:

1. локальная максимизация модулярности путем перемещения узлов в соседнии кластеры;

2. агрегация всех узлов из одинаковых кластеров, и получение нового графа, в котором вершины — это полученные на предыдущем шаге кластеры.

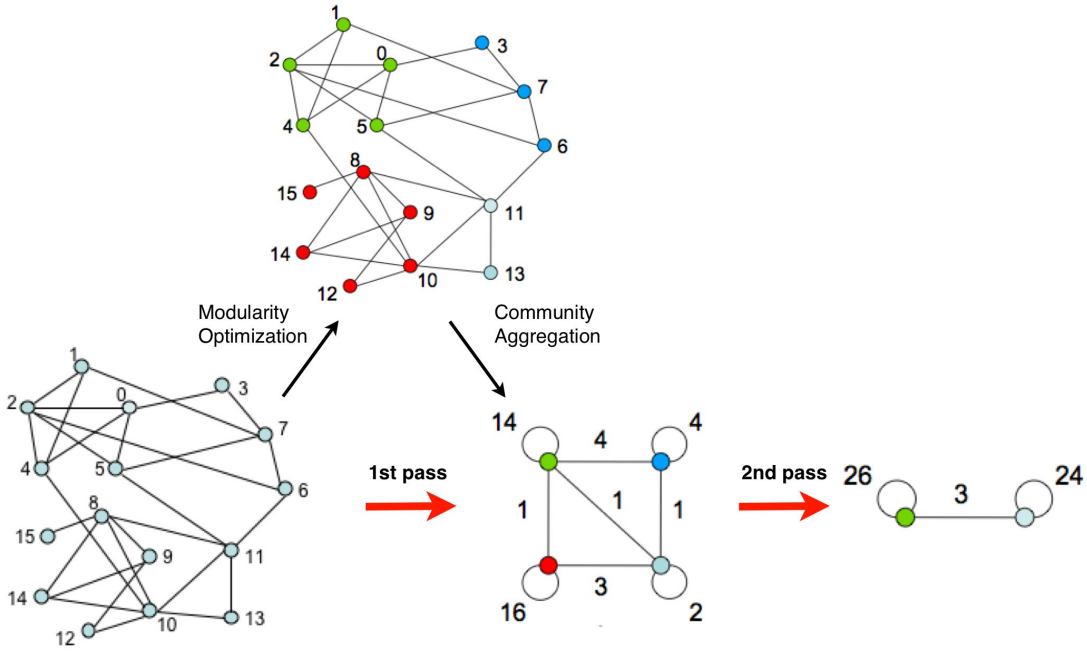
Алгоритм переходит ко второму этапу когда достигнут локальный максимум модулярности. Изменение моделярности, происходящее после перемещения изолированного узлы i в сообщество C_j вычисляется по формуле:

$$\begin{aligned}\Delta Q = &\left[\frac{\sum_{C_j,C_j}+\sum_{i,C_j}}{2m} - \left(\frac{\sum_{C_j}+\sum_i}{2m}\right)^2\right] \\ &- \left[\frac{\sum_{C_j,C_j}}{2m} - \left(\frac{\sum_{C_j}}{2m}\right)^2 - \left(\frac{\sum_i}{2m}\right)^2\right] \\ &= \frac{\sum_{i,C_j}}{2m} - \frac{\sum_{C_j}\sum_i}{2m^2} \quad , \quad (7)\end{aligned}$$

где $\sum_{C_j,C_j}$ — сумма весов всех ребер внутри C_j, $\sum_{C_j}$ — сумма весов всех ребер, связанных с узлами в C_j, $\sum_i$ — сумма весов всех ребер, связанных с узлом i, $\sum_{i,C_j}$ — сумма весов всех ребер связывающих i и узлы из C_j, и m — сумма всех весов всех ребер в сети.

Изначально каждый узел в сети помещается в свое собственное сообщество. Затем, (1) для каждого узла i берутся соседнии сообщества C_j и вычисляется изменение модулярности $\Delta Q_{i,C_j}$ при перемещении узла i в сообщество C_j. Если $\max_{C_j} \Delta Q_{i,C_j} > 0$, тогда узел i фактически перемещается в то сообщество, на котором будет получен максимальный прирост. После этого, (2) строится новый граф, узлами которого являются сообщества $C_j,\ \forall j$, и алгоритм возвращается к первому этапу.

Рис. 2: Визуализация этапов Louvain метода. Каждая итерация состоит из двух этапов. Первый этап: улучшение модулярности с помощью перемещения узлов в соседнии сообщества. Второй этап: агрегация найденых сообществ и получение новой сети. Этот рисунок изначально размещен в [26].



Эта процедура повторяется до тех пор, пока она дает прирост модулярности.

Для лучшего понимания, эти шаги изображены на Рис. 2.

Core Groups Graph Cluster

В 2012 году Овельгённе и Гейр-Шульц (Ovelgoenne, Geyer-Schulz) предложили метод Core Groups Graph Cluster (CGGC) [62]. Метод заключается в создании первоначального "качественного" разбиения на кластеры, и последующим разбиением этого графа с помощью некоторого финального алгоритма. Для создания изначального разбиения предполагается использовать нескольких заранее выбранных начальных алгоритмов, а затем вычислить из них разбиение равное их максимальному перекрытию (см. Алг. 1). Это означает, что та пара узлов, которая находится в одном сообществе во всех начальных разбиениях, также должна находится в одном сообществе в итоговом разбиении.

Algorithm 1 Core Groups Graph Cluster

Input: Граф G, набор начальных алгоритмов, финальный алгоритм;
Output: Разбиение графа G на кластеры C;

1: $S = \emptyset$;
2: **for** для каждого начального алгоритма **do**
3: создать разбиение на сообщества для графа G, и сохранить его в S;
4: **end for**
5: создать промежуточное разбиение на кластеры $\widetilde{P}$, основанное на разбиениях S;
6: создать разбиение на кластеры C, применив финальный алгоритм к $\widetilde{P}$;

Описанный алгоритм позволяет преодолеть сложность построения начального разбиения, неправильный выбор которого может в значительной мере отразиться на конечном результате. Такой подход показал хорошие результаты и выиграл 10th DIMACS Implementation Challenge [1] в 2012 году.

Randomized Greedy

В 2010 году Овельгённе и Гейр-Шульц (Ovelgoenne, Geyer-Schulz) предложили рандомизированную модификацию известного "жадного" NG [58] алгоритма, названную Randomized Greedy [61]. Этот алгоритм разбивает граф на n частей, а затем на каждой итерации он выбирает k произвольных сообществ с их соседями, и объединяет пару которая даст наибольший прирост модулярности, если такая есть. Результатом алгоритма будет разбиение на сообщества, которое имеет наибольшую глобальную модулярность.

Такой алгоритм, с входным параметром k далее будем обозначать RG_k.

Особенностью данного алгоритма является то, что требуется вручную задавать параметер k для каждого графа. Однако, нет оптимального значения такого параметра, которое бы позволило нам одинаково хорошо работать на каждом графе.

2. Рандомизированный и адаптивный подходы в решении задачи оптимизации модулярности

Стохастическая апроксимация была введена Роббинсом и Монро (Robbins and Monro) [72] и получила дальнейшее развитие в решении ряда оптимизационных задач Киефиром и Вольфовицем (Kiefer and Wolfowitz, KW) [43]. В 1954 году Блюм (Blum) расширил ее для многомерного случая [8]. В случае m-мерного пространства стандартная KW-процедура, основанная на конечно-разностной апроксимации вектора градиента функции, использует $2m$ измерений на каждой итерации, по 2 измерения на каждую координату вектора градиента. В 80–90-х годах была рассмотрена рандомизированная версия алгоритма стохастической апроксимации, требующая всего 1 либо 2 измерения на итерацию. Этот алгоритм был предложен Граничиным в 1989 [91], Поляком и Цибаковым в 1990 [93], и затем Спалом (Spall) в 1992 [81], и получил название — *одновременно возмущаемой стохастической апроксимацией* (англ. Simultaneous Perturbation Stochastic Approximation, SPSA).

Стохастическая апроксимация показала свою эффективность в проблемах минимизации стационарного функцинала. В [47, 10, 34] похожие алгоритмы с постоянным размером шага были применены для функционалов изменяющихся во времени.

Алгоритм одновременно возмущаемой стохастической апроксимацией с постоянным шагом описан в алгоритме 2.

Поскольку алгоритм SPSA хорошо подходит для создания адаптивных модификаций алгоритмов, зависящих от входных параметров, далее будет рассмотрена возможность применения дискретной версии алгоритма SPSA с постоянным размером шага к RG и CGGC для автоматического выбора оптимальных параметров.

Algorithm 2 SPSA с постоянным шагом

Input: функция f, начальное приближение $\hat{\theta}_0 \in \mathbb{R}^m$, возмущение $d \in \mathbb{R} \setminus \{0\}$, размер шага $\alpha \in \mathbb{R}^m$, и $\varepsilon > 0$;
Output: $\hat{\theta}_n$;

1: $n = 0$;
2: **repeat**
3: $n = n + 1$;
4: выбираем $\Delta_n \in \mathbb{R}^m$ такое, что $\Delta_{n_i} = \pm 1$ и $\Delta_{n_i} \sim \mathrm{B}\left(1, \frac{1}{2}\right)$;
5: $\theta_n^- = \hat{\theta}_{n-1} - d\Delta_n$ и $\theta_n^+ = \hat{\theta}_{n-1} + d\Delta_n$;
6: $y_n^- = f(\theta_n^-)$ и $y_n^+ = f(\theta_n^+)$;
7: $\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha\Delta_n \frac{y_n^+ - y_n^-}{2d}$;
8: **until** $|\hat{\theta}_n - \hat{\theta}_{n-1}| < \varepsilon$

2.1. Рандомизированная модификация Louvain

Louvain алгоритм эффективно справляется с большим количеством данных, но обработка графа с миллиардами связей все равно занимает довольно много времени [26].

Рассмотрим следующую идею модификации алгоритма, примененную в Randomized Greedy [63]. Пусть на каждой итерации алгоритм будет учитывать только произвольное множество соседей, что позволит снизить количество рассматриваемых вариантов (см. Алг. 3).

Этот подход уменьшает время вычислений, особенно для больших сетей, а также получает высокое значение функции модулярности.

Далее, в разделе 3.1 будет проведено сравнение оригинального алгоритма с его предложенной рандомизированной модификацией. Также будет представлена зависимость времени вычисления и полученной модулярности от количества рассматриваемых соседей.

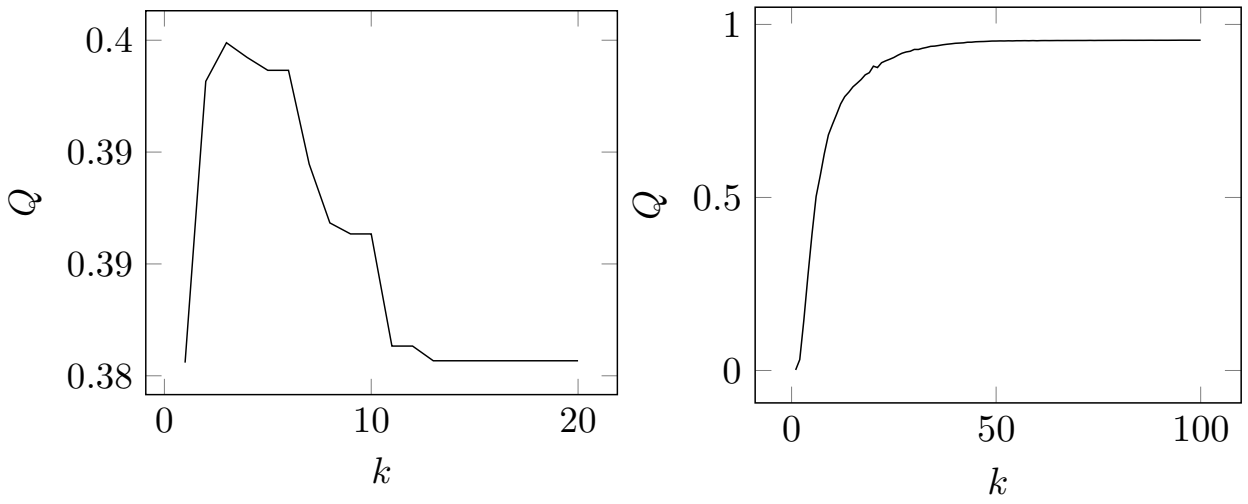
2.2. Адаптивные рандомизированные модификации Randomized Greedy и Core Groups Graph Cluster

Алгоритмы Randomized Greedy и Core Groups Graph Cluster зависят от различных параметров: RG_k зависит от параметра k, $CGGC$ зависит от начальных и финального алгоритма.

Algorithm 3 Рандомизированная модификация Louvain

Input: $G = (V, E)$;
Output: разбиение графа G;

$k = 0$, $G^0 = G$;
loop
    построить простое разбиение C^k для графа G^k такое что $C_i^k = \{i\}$;
    repeat ▷ Этап 1
        for узел $i \in G^k$ **do**
            удалить узел i из его сообщества C_i^k;
            выбрать C_N — произвольное множество соседних сообществ узла i;
            $C_j^k = \text{argmax}_{C_{j'}^k \in C_N} \Delta Q(i, C_{j'}^k)$;
            if $\Delta Q(C_j^k, i) > 0$ **then**
                добавить узел i в сообщество C_j^k;
            else
                вернуть узел i в сообщество C_i^k;
            end if
        end for
    until есть улучшения
    создать новый граф G^{k+1}, узлы которого — это сообщества C^k; ▷ Этап 2
    if $G^{k+1} = G^k$ **then**
        сделать разбиение C_{final} графа G;
        return C_{final};
    end if
    $k = k + 1$;
end loop



(a) *karate* (b) *netscience*

Рис. 3: Зависимость модулярности Q и параметра k на различных графах. Учитывая что $k \in \mathbb{N}$, для наглядности сделаем график непрерывным.

Как известно, SPSA хорошо подходит для создания гибких адаптивных алгоритмов, которые способны адаптироваться к различным входных параметрам. Далее будет рассмотрена возможность применения алгоритма SPSA с константной величиной шага к алгоритмам Randomized Greedy и Core Groups Graph Cluster для автоматического выбора набора оптимальных параметров.

2.2.1. Построение функции качества

В зависимости от входного графа, среднее значение модулярности финального разбиения у RG_k будет наибольшим на маленьких k (см. Рис. 3a), или оно будет расти по мере увеличения k (см. Рис. 3b).

Поскольку итерация RG_k имеет алгоритмическую сложность $O(k)$, рассмотрим следующую функцию качества:

$$F(Q, k) = -\ln Q + \beta \ln k \quad , \tag{8}$$

где $\beta \geq 0$ может быть рассмотрена как $\beta = \frac{\ln \gamma}{\ln 2}$, в которой γ представляет необходимое увеличение значения Q, в случае если k будет увеличен в 2 раза.

Использование алгоритма SPSA обосновано для выпуклых функций качества. Основываясь на значених функции для $k \in \mathbb{N}$, она может быть расширена на всем интервале до непрерывной функции, используя линейные функции между натуральными числами. На основе проведенных экспериментов (см. Рис. 4), можно сказать что функция качества $F(Q, k)$ — выпуклая.

Как можно видеть на Рис. 4b, в основном для получения больших значений у функции качества нужно использовать большую β. Однако, иногда этого также можно достигнуть с маленькими β.

2.2.2. Адаптивная модификация Randomized Greedy

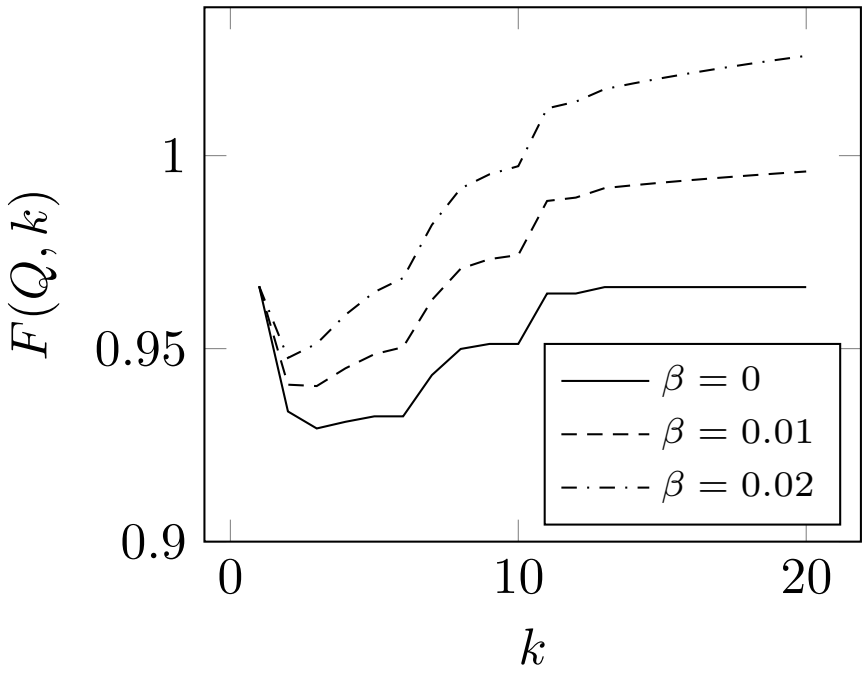
Для применения метода SPSA, необходимо разделить алгоритм RG на шаги длиной σ итераций. На каждом шаге, будет использоваться фиксированное k и каждые 2 шага будут выбираться 2 новых значения параметра k, основываясь на лучшем значении для k на предыдущей итерации.

Кроме того, пусть функция качества использует средний прирост модулярности за последнии σ шагов. Описанный алгоритм формально может быть записан следующим образом: см. Алг. 4.

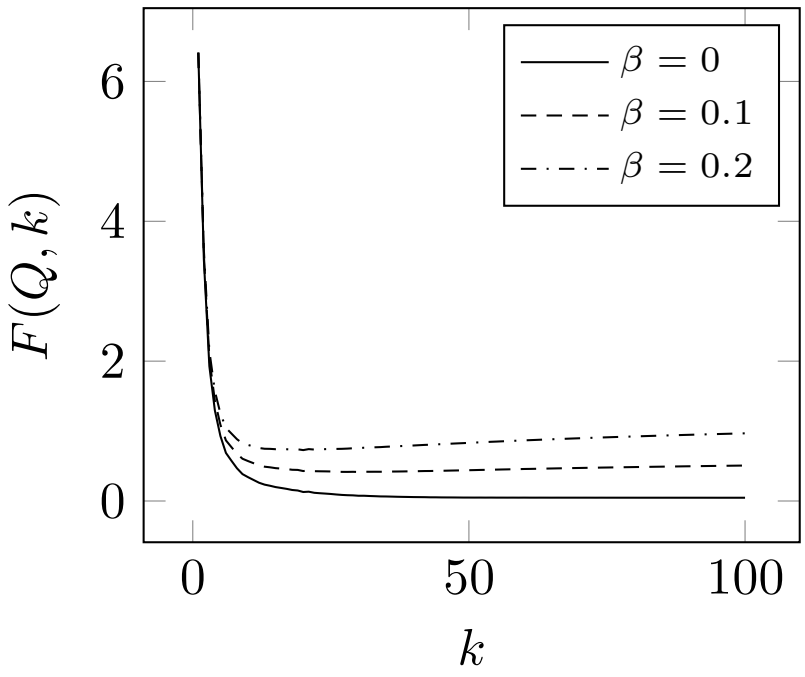
В отличие от Randomized Greedy, адаптивная модификация Randomized Greedy имеет 5 независимых параметров. Однако, согласно результатам тестирования на графах *cond-mat-2003*, *caidaRouterLevel* и *cnr-2000*, параметры α и σ почти не имеют влияния на финальное разбиение, в отличие от параметров d и $\hat{k}_0$ которые сильно на него влияют. Тем не менее, по результатам тестирования был подобран следующий набор параметров, хорошие результаты: $\alpha = 10$, $\sigma = 1000$, $d = 5$, $\hat{k}_0 = 8$

Часто, использование маленького β дает разбиение лучше, чем при $\beta = 0$ (см. Рис. 5).

В разделе 3.2 проведена апробация этого алгоритма.



(a) *karate*



(b) *netscience*

Рис. 4: Функция качества с $\beta = 0$, $\beta = 0.1$, и $\beta = 0.2$ для различных графов.

Algorithm 4 Адаптивная модификация Randomized Greedy

Input: $G = (V, E)$, начальное приближение $\hat{k}_0 \in \mathbb{N}$, возмущение $d \in \mathbb{N}$, разбер шага $\alpha \geq 0$, коэффициент значимости времени работы $\beta \geq 0$, количество итераций на шаге $\sigma \in \mathbb{N}$;

Output: разбиение графа G;

1: $n = 0$, разбить G на n сообществ;
2: **repeat**
3: $n = n + 1$;
4: $k_n^- = \max\{\hat{k}_{n-1} - d, 1\}$ и $k_n^+ = \hat{k}_{n-1} + d$;
5: вычислить средний прирост модулярности Q_n^- за следующие σ итераций: взять k_n^- случайных сообществ с их соседями и объеденить пару, которая даст наибольший прирост модулярности, и повторить итерацию;
6: вычислить среднее значение увеличения модулярности Q_n^+ за следующие σ итераций для k_n^+;
7: $y_n^- = -\ln Q_n^- + \beta \ln k_n^-$ и $y_n^+ = -\ln Q_n^+ + \beta \ln k_n^+$;
8: $\hat{k}_n = \max\left\{1, \left\lfloor \hat{k}_{n-1} - \alpha \frac{y_n^+ - y_n^-}{k_n^+ - k_n^-} \right\rfloor\right\}$;
9: **until** пока есть сообщества, которые могут быть объеденены
10: **return** разбиение на сообщества с наибольшим значением функции модулярности;

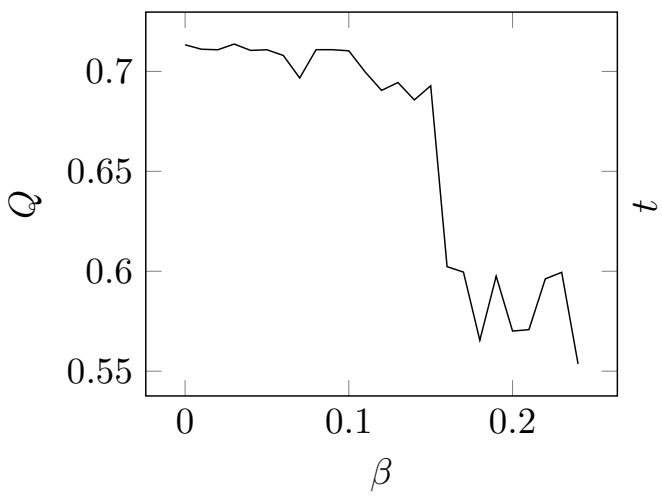
2.2.3. Адаптивная модификация Core Groups Graph Cluster

Для создания адаптивного алгоритма, способного работать на графах любых размеров, рассмотрим возможность использования $SPSA$ для выбора начальных методов. Кроме того, в целях улучшения финального разбиения, пусть промежуточное разбиение строится только по нескольким лучшим, а не по всем.

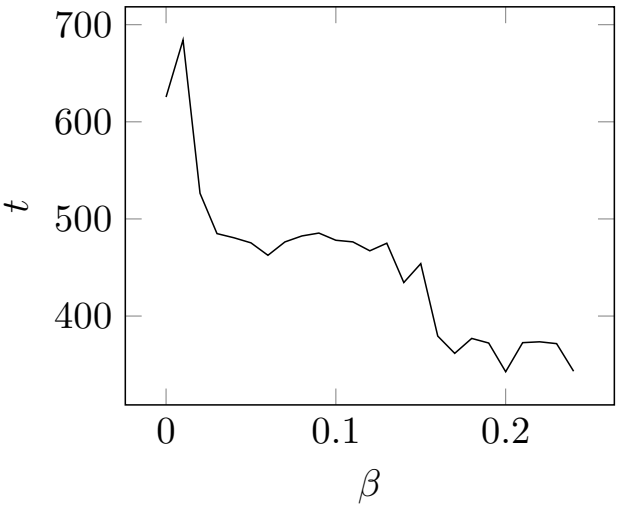
Описанный алгоритм ACGGC (см. Алг. 5) имеет много параметров, однако при их следующих значениях получаются хорошие результаты для всех расмотренных графов (см. Табл. 1 и Табл. 2): $d = 2$, $\alpha = 1000$, $l = 6$, $\hat{k}_0 = 5$, $k_{max} = 50$, $r = 0.05$.

Уменьшение значения параметра k_{max} сокращает время работы алгоритма, и в дополнение часто увеличивает полученную модулярность.

В разделе 3.3 приведено сравнение алгоритма Core Groups Graph Cluster с предложенной в работе адаптивной модификацией.



(a) Зависимость между Q и β.



(b) Зависимость между t и β.

Рис. 5: Зависимость модулярности Q и времени работы ARG алгоритма t и коэффициента β для графа *cond-mat-2003*.

Algorithm 5 Адаптивная модификация Core Groups Graph Clustering

Input: $G = (V, E)$, начальное приближение $\hat{k}_0 \in \mathbb{N}$, возмущение $d \in \mathbb{N}$, размер шага $\alpha \geq 0$, количество шагов $l \in \mathbb{N}$, $k_{max} \in \mathbb{N}_{\geq 2}$, и $r \in (0, 1]$;

Output: разбиение графа G;

1: $n = 0$, $S = \emptyset$;
2: **repeat**
3: $n = n + 1$;
4: $k_n^- = \max\{1, \hat{k}_{n-1} - d\}$, $k_n^+ = \min\{k_{max}, \hat{k}_{n-1} + d\}$;
5: создать разбиение P_n^- используя $RG_{k_n^-}$ и добавить его в S, модулярность которого Q_n^-;
6: создать разбиение P_n^+ используя $RG_{k_n^+}$ и добавить его в S, модулярность которого Q_n^+;
7: $y_n^- = -\ln Q_n^-$ и $y_n^+ = -\ln Q_n^+$ и пересчитать Q_{best};
8: $\hat{k}_n = \max\left\{1,\ \min\left\{k_{max},\ \left\lfloor \hat{k}_{n-1} - \alpha \frac{y_n^+ - y_n^-}{k_n^+ - k_n^-} \right\rfloor \right\}\right\}$
9: **until** $n = l$
10: $\widetilde{S} = \{S_i \in S$ такое, что $Q(G, S_i) \geq (1 - r)Q_{best}\}$;
11: создать промежуточное разбиение $\widetilde{P}$ на основе $\widetilde{S}$;
12: применить финальный алгоритм к промежуточному разбиению $\widetilde{P}$;

Таблица 1: Среднее значение модулярности для ACGGC с разными k_{max} на разных графах.

k_{max}	$+\infty$	50	20	10	6
polbooks	0.527237	0.527237	0.527237	0.527082	0.526985
adjnoun	0.299720	0.299690	0.299859	0.300141	0.299676
football	0.603324	0.603324	0.604184	0.604266	0.604266
jazz	0.444739	0.444739	0.444739	0.444739	0.444739
celegans	0.439770	0.439368	0.439750	0.439460	0.439431
email	0.573470	0.573416	0.573652	0.573756	0.573513
netscience	0.953033	0.908130	0.842085	0.793289	0.768572
cond-mat-2003	0.737611	0.743572	0.749595	0.749894	0.739200

Таблица 2: Среднее время работы алгоритмов $ACGGC$ с разными k_{max} на различных графах.

k_{max}	$+\infty$	50	20	10	6
polbooks	5.029	4.976	4.615	4.207	4.087
adjnoun	6.115	6.099	5.481	4.952	4.744
football	7.179	7.155	6.377	5.820	5.584
jazz	23.66	23.25	20.92	19.12	18.59
celegans	23.85	23.49	22.48	20.92	20.01
email	70.06	72.89	68.34	63.85	62.97
netscience	477.97	85.40	46.08	38.26	30.89
cond-mat-2003	41,950	9,596	6,075	5,092	4,166

3. Имитационное моделирование

Тестирование производилось на компьютере с ОС Ubuntu 15.10, Intel Core i5-5200U ЦПУ (2.20ГГц) и 16Гб ОЗУ.

Для измерения качества и времени вычисления этих алгоритмов были использованы тестовые графы, предложенные в 10th DIMACS Implementation Challenge - Graph Partitioning and Graph Clustering (см. [1]):

- karate.graph — Zachary's karate club: сеть друзей из 34 членов клуба карате университета США в 1970-х годах ($n = 34$, $m = 78$);
- as-22july06.graph — снимок структуры сети Интернет на уровне автономных систем, полученный из BGP таблиц Университета Орегона Route Views Project ($n = 22963$, $m = 48436$);
- cnr-2000.graph — снимок итальянских CNR доменов, полученных в результате очень маленького обхода ($n = 325557$, $m = 2738969$);
- eu-2005.graph — снимок .eu доменов, полученных в результате маленького обхода ($n = 862664$, $m = 16138468$);
- in-2004.graph — снимок .in доменов, полученных в результате обхода для Технологического Университета Нагаока ($n = 1382908$, $m = 13591473$);
- road_central.graph — граф дорог ($n = 14081816$, $m = 16933413$);
- uk-2002.graph — снимок .uk доменов, полученных в результате обхода компанией UbiCrawler ($n = 18520486$, $m = 261787258$);
- road_usa.graph — граф дорог США ($n = 23947347$, $m = 28854312$);
- uk-2007-05.graph — снимок .uk доменов, полученных в результате объединения 12 месячных наблюдений собранных для DELIS проекта ($n = 105896555$, $m = 3301876564$).

Таблица 3: Среднее время работы Louvain метода, Randomized Greedy алгоритма, и рассматриваемой модификации RL, в секундах. OOM означает недостаточно памяти.

	$Louvain$	$RG_{k=9}$	$RL_{k=75\%}$	$RL_{k=50\%}$	$RL_{k=25\%}$
karate	0.000	0.000	0.000	0.000	0.000
as-22july06	0.036	0.072	0.028	0.036	0.044
cnr-2000	4.594	4.040	0.792	0.768	0.932
eu-2005	14.106	24.676	3.844	4.448	4.748
in-2004	26.646	21.396	3.156	3.516	4.368
road_central	123.028	114.824	36.612	36.016	44.392
uk-2002	433.468	OOM	70.976	71.256	70.880
road_usa	183.516	196.072	53.036	48.552	49.852
uk-2007-05	OOM	OOM	OOM	OOM	OOM

3.1. Рандомизированная модификация Louvain

Ниже будет показано сравнение Louvain метода, Randomized Greedy с параметром $k = 9$, и рандомизированной модификации Louvain метода с различными значениями параметра k.

Для предложенного алгоритма рассмотрим следующие случаи, зависящие от максимального количества рассматриваемых соседей:

- $k = 75\%$ — количество рассматриваемых соседей на каждой итерации равно 75% от общего количества соседей (далее — $RL_{k=75\%}$);
- $k = 50\%$ — количество рассматриваемых соседей на каждой итерации равно 50% от общего количества соседей (далее — $RL_{k=50\%}$);
- $k = 25\%$ — количество рассматриваемых соседей на каждой итерации равно 25% от общего количества соседей (далее — $RL_{k=25\%}$).

Табл. 3 содержит среднее время работы для различных алгоритмов на тестовых графах.

В Табл. 4 показано среднее значение функции модулярности, полученное для различных алгоритмов на тестовых графах.

Как видно на предыдущих таблицах, предложенная рандомизированная модификация Louvain метода работает быстрее оригинального алгоритма, и Randomized Greedy алгоритма. Результат, полученный

Таблица 4: Среднее значение модулярности для Louvain метода, Randomized Greedy алгоритма, и рассматриваемой модификации.

	Louvain	$RG_{k=9}$	$RL_{k=75\%}$	$RL_{k=50\%}$	$RL_{k=25\%}$
karate	0.41452	0.39423	0.35528	0.36037	-0.04980
as-22july06	0.66230	0.64839	0.61879	0.59751	0.48388
cnr-2000	0.91276	0.91051	0.91073	0.90602	0.88533
eu-2005	0.93822	0.92746	0.92280	0.89709	0.85685
in-2004	0.98020	0.96735	0.97707	0.97012	0.93383
road_central	0.99738	0.99723	0.99509	0.99205	0.98569
uk-2002	0.98973	OOM	0.94453	0.93721	0.94389
road_usa	0.99804	0.99791	0.99623	0.99370	0.99382
uk-2007-05	OOM	OOM	OOM	OOM	OOM

предложенным алгоритмом сравним с результатами других алгоритмов. Таким образом, пренебрегая небольшим ухудшением качества кластеризации, описанный способ может быть использован для обработки большого объема данных за короткое время.

3.2. Адаптивная модификация Randomized Greedy

Для анализа полученных результатов также будем использовать набор тестовых графов с 10th DIMACS Implementation Challenge, описанный в предыдущем разделе. Качество полученного разбиения будет вычисляться значением функции модулярности Q (см. (2)).

Кроме того, для оценки качества алгоритмов был сгенерирован граф *auto40* ($n = 40000$) из 40 сообществ, с вероятностью 0.1 существования ребра между узлами одного сообщества, и вероятностью 10^{-4} существования ребра между узлами из разных сообществ.

В таблицах 5 и 6, RG_k и ARG сравниваются по средней модулярности и скорости работы для различных параметров k:

- $k = 1$ — минимальное значени k;
- $k = 3$ — значение, на котором часто получаются хорошие результаты;
- $k = 10$ — значение, на котором результат получается стабильным,

Таблица 5: Среднее значение модулярности для RG_k и ARG на разных графах.

	RG_1	RG_3	RG_{10}	RG_{50}	ARG
as-22july06	0.65281	0.64658	0.64024	0.63479	0.64264
cond-mat-2003	0.00012	0.19727	0.70738	0.69403	0.71193
auto40	0.78944	0.79988	0.80417	0.80273	0.80174
caidaRouterLevel	0.01938	0.81101	0.79883	0.79300	0.80216
cnr-2000	0.90237	0.91192	0.91144	0.90997	0.91039
eu-2005	0.92765	0.92559	0.91780	0.90416	0.91048
in-2004	0.00026	0.97836	0.97185	0.97596	0.97616

Таблица 6: Среднее время работы алгоритмов RG_k и ARG на различных графах, в миллисекундах.

	RG_1	RG_3	RG_{10}	RG_{50}	ARG
as-22july06	177	189	231	464	238
cond-mat-2003	58	184	463	931	474
auto40	4,652	4,591	6,017	12,558	6,479
caidaRouterLevel	852	9,114	10,244	15,217	11,514
cnr-2000	26,083	26,056	27,137	33,592	29,054
eu-2005	202,188	200,686	207,689	246,170	225,748
in-2004	9,208	487,953	553,196	607,408	617,345

т.е. лучшие относительные значения в среднем;

- $k = 50$ — большое значение k.

В качестве входных параметров ARG были использованы следующие параметры: $\alpha = 10$, $\sigma = 1000$, $d = 5$, $\hat{k}_0 = 8$, $\beta = 0.05$.

В большинстве случаев, RG_k с разными значениями k имеют более хорошие результаты чем ARG, но с другой стороны, ARG имеет более стабильный результат, аналогично RG_{10}, и более высокое среднее значение модулярности чем у RG_{10}.

3.3. Адаптивная модификация Core Groups Graph Cluster

В таблице 7 содержатся результаты тестирования 5 различных модификаций CGGC и ACGGC с их средним значением модулярности:

Таблица 7: Среднее значение модулярности для ACGGC и CGGC на разных графах. Примечание: [1]*pgpGiantCompo*, [2]*as-22july06*, [3]*cond-mat-2003*, [4]*caidaRouterLevel*.

	$ACGGC^I$	$ACGGC^{II}$	$CGGC_{10}^{10}$	$CGGC_3^{10}$	$CGGC_{10}^3$
karate	0.417242	0.417406	0.415598	0.396532	0.405243
dolphins	0.524109	0.523338	0.521399	0.523338	0.522428
chesapeake	0.262439	0.262439	0.262439	0.262439	0.262370
adjnoun	0.299704	0.299197	0.295015	0.292703	0.290638
polbooks	0.527237	0.527237	0.527237	0.526938	0.526784
football	0.603324	0.604266	0.604266	0.599537	0.599026
celegans	0.439604	0.438584	0.435819	0.436066	0.432261
jazz	0.444739	0.444848	0.444871	0.444206	0.444206
netscience	0.907229	0.835267	0.724015	0.708812	0.331957
email	0.573333	0.573409	0.571018	0.572667	0.567423
polblogs	0.424107	0.423208	0.422901	0.421361	0.390395
pgpGiant[1]	0.883115	0.883085	0.882237	0.882532	0.880340
as-22jul[2]	0.671249	0.670677	0.666766	0.669847	0.665260
cond-mat[3]	0.744533	0.750367	0.751109	0.708775	0.413719
caidaRou[4]	0.846312	0.855651	0.851622	0.858955	0.843835
cnr-2000	0.912762	0.912783	0.912500	0.912777	0.912496
eu-2005	0.938292	0.936984	0.935510	0.936515	0.936420
in-2004	0.979844	0.979771	0.979883		

- $ACGGC^I$ — $ACGGC$ с $d = 2$, $\alpha = 1000$, $l = 6$, $\hat{k}_0 = 5$, $k_{max} = 50$, $r = 0.05$, и финальным алгоритмом RG_{10};
- $ACGGC^{II}$ — $ACGGC$ с $d = 2$, $\alpha = 1000$, $l = 8$, $\hat{k}_0 = 5$, $k_{max} = 20$, $r = 0.05$, и финальным алгоритмом RG_{10};
- $CGGC_{10}^{10}$ — $CGGC$ с начальным алгоритмом RG_{10}, финальным алгоритмом RG_{10}, и $s = 16$;
- $CGGC_3^{10}$ — $CGGC$ с начальным алгоритмом RG_3, финальным алгоритмом RG_{10}, и $s = 16$;
- $CGGC_{10}^3$ — $CGGC$ с начальным алгоритмом RG_{10}, финальным алгоритмом RG_3, и $s = 16$.

где лучшее значение, и второе по величине значение.

В таблице 7 показано, что ACGGC обычно работает лучше чем CGGC. Во время тестирования был выбрано следующее значение параметра $s = 16$, так как CGGC с большим s обеспечивает более высокое значение модулярности [62].

Заключение

Основные научные результаты выпускной работы, полученные в рамках выполнения поставленных задач:

1. предложена рандомизированная модификация Louvain метода, основанная на случайном выборе и проверке заданного количества соседних узлов (Алгоритм 3, Раздел 2.1);

2. предложены адаптивные модификации Core Groups Graph Cluster и Randomized Greedy методов, основанные на алгоритме стохастической апроксимации SPSA (Алгоритм 4 и Алгоритм 5, Раздел 2.2);

3. произведена апробация полученных алгоритмов, и сделано их сравнение со стандартными методами.

Список литературы

[1] 10th DIMACS Implementation Challenge. — https://www.cc.gatech.edu/dimacs10/archive/clustering.shtml/.

[2] Agarwal Gaurav, Kempe David. Modularity-maximizing graph communities via mathematical programming // The European Physical Journal B. — 2008. — Vol. 66, no. 3. — P. 409–418.

[3] Albert R., Barabási A.-L. Statistical mechanics of complex networks // Reviews of Modern Physics. — 2002. — Vol. 74, no. 1. — P. 47–97.

[4] Algorithms and Applications for Community Detection in Weighted Networks / Z. Lu, X. Sun, Y. Wen et al. // IEEE Transactions on Parallel and Distributed Systems. — 2015. — Vol. 26, no. 11. — P. 2916–2926.

[5] Arenas Alex, Fernandez Alberto, Gomez Sergio. Analysis of the structure of complex networks at different resolution levels // New journal of physics. — 2008. — Vol. 10, no. 5. — P. 053039.

[6] Barnes Earl R. An algorithm for partitioning the nodes of a graph // SIAM Journal on Algebraic Discrete Methods. — 1982. — Vol. 3, no. 4. — P. 541–550.

[7] Bezdek James C. Pattern recognition with fuzzy objective function algorithms. — Kluwer Academic Publishers, Norwell, USA, 1981.

[8] Blum Julius R. Multidimensional stochastic approximation methods // The Annals of Mathematical Statistics. — 1954. — P. 737–744.

[9] Bollobás Béla. Random graphs // Modern graph theory. — Springer, 1998. — P. 215–252.

[10] Borkar Vivek S. Stochastic Approximation: a Dynamical Systems Viewpoint. — Cambridge University Press Cambridge, 2008.

[11] Calafiore Giuseppe, Polyak Boris T. Stochastic algorithms for exact and approximate feasibility of robust LMIs // IEEE Transactions on Automatic Control. — 2001. — Vol. 46, no. 11. — P. 1755–1759.

[12] Chan Yun Kwan, Yeung Dit-Yan. A convex formulation of modularity maximization for community detection // Twenty-Second International Joint Conference on Artificial Intelligence. — 2011.

[13] Chen Mingming, Nguyen Tommy, Szymanski Boleslaw K. On measuring the quality of a network community structure // 2013 International Conference on Social Computing / IEEE. — 2013. — P. 122–127.

[14] Chen Mingming, Nguyen Tommy, Szymanski Boleslaw K. A new metric for quality of network community structure // arXiv preprint arXiv:1507.04308. — 2015.

[15] Clauset A., Newman M.E.J., Moore C. Finding community structure in very large networks // Physical Review E. — 2004. — Vol. 70, no. 6. — P. 066111.

[16] Complex networks theory for analyzing metabolic networks / Jing Zhao, Hong Yu, Jianhua Luo et al. // Chinese Science Bulletin. — 2006. — Vol. 51, no. 13. — P. 1529–1537.

[17] Cordeiro Mário, Sarmento Rui Portocarrero, Gama João. Dynamic community detection in evolving networks using locality modularity optimization // Social Network Analysis and Mining. — 2016. — Vol. 6, no. 1. — P. 15.

[18] Criminal behavior analysis based on Complex Networks theory / Hong Wang, Zhao-wen Wang, Jian-bo Li, Qiu-hong Wei // 2009 IEEE International Symposium on IT in Medicine & Education / IEEE. — Vol. 1. — 2009. — P. 951–955.

[19] Cruz Juan David, Bothorel Cécile, Poulet François. Entropy based community detection in augmented social networks // 2011

International Conference on computational aspects of social networks (CASoN) / IEEE. — 2011. — P. 163–168.

[20] Danon Leon, Díaz-Guilera Albert, Arenas Alex. The effect of size heterogeneity on community identification in complex networks // Journal of Statistical Mechanics: Theory and Experiment. — 2006. — Vol. 2006, no. 11. — P. P11010.

[21] Duch J., Arenas A. Community detection in complex networks using Extremal Optimization // Physical Review E. — 2005. — Vol. 72, no. 2. — P. 027104.

[22] Dunn Joseph C. Well-separated clusters and optimal fuzzy partitions // Journal of cybernetics. — 1974. — Vol. 4, no. 1. — P. 95–104.

[23] Efficient identification of web communities / Gary William Flake, Steve Lawrence, C Lee Giles et al. // KDD. — Vol. 2000. — 2000. — P. 150–160.

[24] Enhancing sentiment analysis on twitter using community detection / William Deitrick, Benjamin Valyou, Wes Jones et al. // Communications and Network. — 2013. — Vol. 5, no. 03. — P. 192.

[25] Faloutsos Michalis, Faloutsos Petros, Faloutsos Christos. On power-law relationships of the internet topology // ACM SIGCOMM computer communication review / ACM. — Vol. 29. — 1999. — P. 251–262.

[26] Fast unfolding of communities in large networks / V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre // Journal of Statistical Mechanics: Theory and Experiment. — 2008. — Vol. 2008, no. 10. — P. 10008.

[27] Fiedler Miroslav. Algebraic connectivity of graphs // Czechoslovak mathematical journal. — 1973. — Vol. 23, no. 2. — P. 298–305.

[28] Ford Jr Lester Randolph, Fulkerson Delbert Ray. Solving the transportation problem // Management Science. — 1956. — Vol. 3, no. 1. — P. 24–32.

[29] Forster Richard. Louvain community detection with parallel heuristics on GPUs // 2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES) / IEEE. — 2016. — P. 227–232.

[30] Fortunato S. Community detection in graphs // Physics Reports. — 2010. — Vol. 486, no. 3-5. — P. 75–174.

[31] Fortunato Santo, Barthelemy Marc. Resolution limit in community detection // Proceedings of the National Academy of Sciences. — 2007. — Vol. 104, no. 1. — P. 36–41.

[32] Fortunato Santo, Hric Darko. Community detection in networks: A user guide // Physics reports. — 2016. — Vol. 659. — P. 1–44.

[33] Girvan M., Newman M.E.J. Community structure in social and biological networks // Proceedings of the National Academy of Sciences. — 2002. — Vol. 99. — P. 7821–7826.

[34] Granichin Oleg, Amelina Natalia. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances // IEEE Transactions on Automatic Control. — 2015. — Vol. 60, no. 6. — P. 1653–1658.

[35] Granichin O., Volkovich Z.V., Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. — Springer-Verlag Berlin Heidelberg, 2015. — Vol. 67 of Intelligent Systems Reference Library. — ISBN: 978-3-642-54785-0.

[36] Graph representations for web document clustering / Adam Schenker, Mark Last, Horst Bunke, Abraham Kandel // Iberian Conference on Pattern Recognition and Image Analysis / Springer. — 2003. — P. 935–942.

[37] Graph structure in the web / Andrei Broder, Ravi Kumar, Farzin Maghoul et al. // Computer networks. — 2000. — Vol. 33, no. 1-6. — P. 309–320.

[38] Grilli Jacopo, Rogers Tim, Allesina Stefano. Modularity and stability in ecological communities // Nature communications. — 2016. — Vol. 7. — P. 12031.

[39] Guimera Roger, Sales-Pardo Marta, Amaral Luís A Nunes. Modularity from fluctuations in random graphs and complex networks // Physical Review E. — 2004. — Vol. 70, no. 2. — P. 025101.

[40] Haq Nandinee Fariah, Moradi Mehdi, Wang Z Jane. Community structure detection from networks with weighted modularity // Pattern Recognition Letters. — 2019. — Vol. 122. — P. 14–22.

[41] Hastie Trevor, Tibshirani Robert, Friedman Jerome. The elements of statistical learning. — Springer, Berlin, Germany, 2001. — ISBN: 0387952845.

[42] Kernighan Brian W, Lin Shen. An efficient heuristic procedure for partitioning graphs // Bell system technical journal. — 1970. — Vol. 49, no. 2. — P. 291–307.

[43] Kiefer Jack, Wolfowitz Jacob. Stochastic estimation of the maximum of a regression function // The Annals of Mathematical Statistics. — 1952. — Vol. 23, no. 3. — P. 462–466.

[44] Kirianovskii Ilia. The Shortest Path Construction Method between Nodes in a Stochastic Network // IFAC-PapersOnLine. — 2015. — Vol. 48, no. 11. — P. 1086–1089.

[45] Kirianovskii Ilia, Granichin Oleg, Proskurnikov Anton. A new randomized algorithm for community detection in large networks // IFAC-PapersOnLine. — 2016. — Vol. 49, no. 13. — P. 31–35.

[46] Kleinberg J., Lawrence S. The Structure of the Web // Science. — 2001. — Vol. 294, no. 5548. — P. 1849–1850.

[47] Kushner Harold, Yin G George. Stochastic Approximation and Recursive Algorithms and Applications. — Springer Science & Business Media, 2003. — Vol. 35.

[48] Lehmann Sune, Hansen Lars Kai. Deterministic modularity optimization // The European Physical Journal B. — 2007. — Vol. 60, no. 1. — P. 83–88.

[49] MacQueen James. Some methods for classification and analysis of multivariate observations // Proceedings of the fifth Berkeley symposium on mathematical statistics and probability / Oakland, CA, USA. — Vol. 1. — 1967. — P. 281–297.

[50] Martelot E.L., Hankin C. Fast Multi-Scale Detection of Relevant Communities in Large-Scale Networks // Computer Journal. — 2013. — Vol. 56, no. 9. — P. 1136.

[51] Massen Claire P, Doye Jonathan PK. Identifying communities within energy landscapes // Physical Review E. — 2005. — Vol. 71, no. 4. — P. 046101.

[52] Medus Andres, Acuña Guillermo, Dorso Claudio Oscar. Detection of community structures in networks via global optimization // Physica A: Statistical Mechanics and its Applications. — 2005. — Vol. 358, no. 2-4. — P. 593–604.

[53] Modularity and predicted functions of the global sponge-microbiome network / Miguel Lurgi, Torsten Thomas, Bernd Wemheuer et al. // Nature communications. — 2019. — Vol. 10.

[54] Moore Cristopher, Newman Mark EJ. Epidemics and percolation in small-world networks // Physical Review E. — 2000. — Vol. 61, no. 5. — P. 5678.

[55] Newman M.E.J. Analysis of weighted networks // Physical Review E. — 2004. — Vol. 70, no. 5. — P. 056131.

[56] Newman M.E.J. Fast algorithm for detecting community structure in networks // Physical Review E. — 2004. — Vol. 69, no. 6. — P. 066133.

[57] Newman M.E.J. Modularity and community structure in networks // Proceedings of the National Academy of Sciences. — 2006. — Vol. 103, no. 23. — P. 8577–8582.

[58] Newman M.E.J., Girvan M. Finding and evaluating community structure in networks // Physical Review E. — 2004. — Vol. 69, no. 2. — P. 026113.

[59] Noack Andreas, Rotta Randolf. Multi-level algorithms for modularity clustering // International Symposium on Experimental Algorithms / Springer. — 2009. — P. 257–268.

[60] On modularity clustering / Ulrik Brandes, Daniel Delling, Marco Gaertler et al. // IEEE transactions on knowledge and data engineering. — 2007. — Vol. 20, no. 2. — P. 172–188.

[61] Ovelgonne Michael, Geyer-Schulz Andreas. Cluster cores and modularity maximization // 2010 IEEE International Conference on Data Mining Workshops / IEEE. — 2010. — P. 1204–1213.

[62] Ovelgönne Michael, Geyer-Schulz Andreas. An ensemble learning strategy for graph clustering. // Graph Partitioning and Graph Clustering. — 2012. — Vol. 588. — P. 187.

[63] Ovelgönne M., Geyer-Schulz A., Stein M. Randomized Greedy Modularity Optimization for Group Detection in Huge Social Networks // Proceedings of the fourth SNA-KDD Workshop, KDD 2010, July. — Vol. 25. — 2010. — P. 1–9.

[64] Paul Subhadeep, Chen Yuguo. Null models and modularity based community detection in multi-layer networks // arXiv preprint arXiv:1608.00623. — 2016.

[65] Pothen Alex. Graph partitioning algorithms with applications to scientific computing // Parallel Numerical Algorithms. — Springer, 1997. — P. 323–368.

[66] Pujol Josep M, Béjar Javier, Delgado Jordi. Clustering algorithm for determining community structure in large networks // Physical Review E. — 2006. — Vol. 74, no. 1. — P. 016107.

[67] Randomized algorithms with adaptive tuning of parameters for detecting communities in graphs / Natalia Amelina, Oleg Granichin, Olga Granichina et al. // 2016 IEEE 55th Conference on Decision and Control (CDC) / IEEE. — 2016. — P. 6222–6227.

[68] Rattigan Matthew J, Maier Marc, Jensen David. Graph clustering with network structure indices // Proceedings of the 24th international conference on Machine learning / ACM. — 2007. — P. 783–790.

[69] Reichardt Jörg, Bornholdt Stefan. Statistical mechanics of community detection // Physical Review E. — 2006. — Vol. 74, no. 1. — P. 016110.

[70] Revealing network communities through modularity maximization by a contraction–dilation method / Juan Mei, Sheng He, Guiyang Shi et al. // New Journal of Physics. — 2009. — Vol. 11, no. 4. — P. 043025.

[71] Rice S.A. The Identification of Blocs in Small Political Bodies // American Political Science Review. — 1927. — Vol. 21, no. 03. — P. 619–627.

[72] Robbins Herbert, Monro Sutton. A stochastic approximation method // The annals of mathematical statistics. — 1951. — P. 400–407.

[73] Rosvall Martin, Axelsson Daniel, Bergstrom Carl T. The map equation // The European Physical Journal Special Topics. — 2009. — Vol. 178, no. 1. — P. 13–23.

[74] Ruan Jianhua, Zhang Weixiong. An efficient spectral algorithm for network community discovery and its applications to biological and social networks // Seventh IEEE International Conference on Data Mining (ICDM 2007) / IEEE. — 2007. — P. 643–648.

[75] Ruan Jianhua, Zhang Weixiong. Identifying network communities with a high resolution // Physical Review E. — 2008. — Vol. 77, no. 1. — P. 016104.

[76] Schuetz Philipp, Caflisch Amedeo. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement // Physical Review E. — 2008. — Vol. 77, no. 4. — P. 046112.

[77] Scott John. Social Network Analysis // Sage. — 2012.

[78] Self-Organization and Identification of Web Communities / G.W. Flake, S. Lawrence, C.L. Giles, F.M. Coetzee // Computer. — 2002. — Vol. 35, no. 3. — P. 66–71.

[79] Shi Jianbo, Malik Jitendra. Motion segmentation and tracking using normalized cuts. — University of California, Berkeley, Computer Science Division, 1997.

[80] Shi Jianbo, Malik Jitendra. Normalized cuts and image segmentation // Departmental Papers (CIS). — 2000. — P. 107.

[81] Spall James C et al. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation // IEEE transactions on automatic control. — 1992. — Vol. 37, no. 3. — P. 332–341.

[82] Suaris Peter R, Kedem Gershon. An algorithm for quadrisection and its application to standard cell placement // IEEE Transactions on Circuits and Systems. — 1988. — Vol. 35, no. 3. — P. 294–303.

[83] Tempo Roberto, Calafiore Giuseppe, Dabbene Fabrizio. Randomized algorithms for analysis and control of uncertain systems: with applications. — Springer Science & Business Media, 2012.

[84] Vazirani Vijay V. Approximation algorithms. — Springer, 2001.

[85] Wakita K., Tsurumi T. Finding Community Structure in Mega-scale Social Networks // Proceedings of the 16th international conference on World Wide Web. — No. 153. — 2007.

[86] Waltman Ludo, Van Eck Nees Jan. A smart local moving algorithm for large-scale modularity-based community detection // The European Physical Journal B. — 2013. — Vol. 86, no. 11. — P. 471.

[87] White Scott, Smyth Padhraic. A spectral clustering approach to finding communities in graphs // Proceedings of the 2005 SIAM international conference on data mining / SIAM. — 2005. — P. 274–285.

[88] Xiang Biao, Chen En-Hong, Zhou Tao. Finding community structure based on subgraph similarity // Complex Networks. — Springer, 2009. — P. 73–81.

[89] Zachary W.W. An Information Flow Model for Conflict and Fission in Small Groups // Journal of Anthropological Research. — 1977. — Vol. 33, no. 4. — P. 452–473.

[90] An algorithm for detecting community structure of social networks based on prior knowledge and modularity / Haifeng Du, Marcus W Feldman, Shuzhuo Li, Xiaoyi Jin // Complexity. — 2007. — Vol. 12, no. 3. — P. 53–60.

[91] Граничин Олег Николаевич. Об одной стохастической рекуррентной процедуре при зависимых помехах в наблюдении, использующей на входе пробные возмущения // Вестник Ленинградского университета. Серия 1: Математика, механика, астрономия. — 1989. — no. 1. — P. 19–21.

[92] Граничин Олег Николаевич, Поляк Борис Теодорович. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. — Федеральное государственное унитарное предприятие Академический научно ..., 2003.

[93] Поляк Борис Теодорович, Цыбаков Александр Борисович. Оптимальные порядки точности поисковых алгоритмов стохастической оптимизации // Проблемы передачи информации. — 1990. — Vol. 26, no. 2. — P. 45–53.