Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет экономических наук

Научная статья на тему

«Анализ знаковых графов»

Выполнил:
Студент группы БСТ201
Комогорцев Андрей Иванович
Ф.И.О.

Руководитель: Егорова Людмила Геннадьевна
Доцент, кандидат физико-математических наук
степень, звание, должность Ф.И.О.

Москва 2022

Содержание

Введение

В данной работе будут рассмотрены методы анализа систем с сетевой структурой, способы выделения ключевой информации, реализация алгоритмов на языке программирования Python версии 3.9 и их применение на реальных данных.

Применение знаковых графов и их востребованность растет с каждым годом. Графы удобны для хранения больших данных, используются для построения рекомендательных систем и дополнения ML-моделей (Machine Learning) неочевидными признаками, сложно отсиживаемых с помощью табличных данных. Анализ сетевых структур или SNA (Social Network Analysis) стал гораздо популярнее и более востребованным с появлением таких гигантов среди социальных сетей, как Facebook, Twitter и др.

Хотя анализ сетевых структур имеет очень много спецификаций, каждая из которых заточена под свою область исследования, в этой работе мы коснемся только одного из методов выделения информации из сетей, имеющих социальный характер, а именно: балансировка и выделение кластеров знакового графа. Этот раздел весьма полезен для выделения сообществ или групп, имеющих высокую согласованность в отношениях к определённым объектам или другим сообществам. Более подробно, детально и чётко все задачи будут поставлены в основной части работы.

Основная часть

Введение основных терминологий и условностей

Для начала стоит ответить на фундаментальный вопрос исследуемого объекта: что такое знаковый граф? Знаковый граф $G = (V, E, W)$ является комбинацией двух множеств и матрицы, а именно: множества вершин (узлов) $V = \{v_i : i = \overline{1,n}\}$, множества ребер $E = \{(v_i, v_j) \in V \times V | i \neq j\}$ и матрицы смежности $W = [w_{ij}]_{n \times n} : w_{ij} = sign(v_i, v_j)$. Множество ребер, в свою очередь, можно разбить на два подмножества $E = E^+ \cup E^-$, где E^+ является множеством ребер, имеющих положительную связь, а E^- является множеством ребер, имеющих отрицательную связь. Очевидно, что если $w_{ij} = 1$, то $(v_i, v_j) \in E^+$, или же если $w_{ij} = -1$, то $(v_i, v_j) \in E^-$.

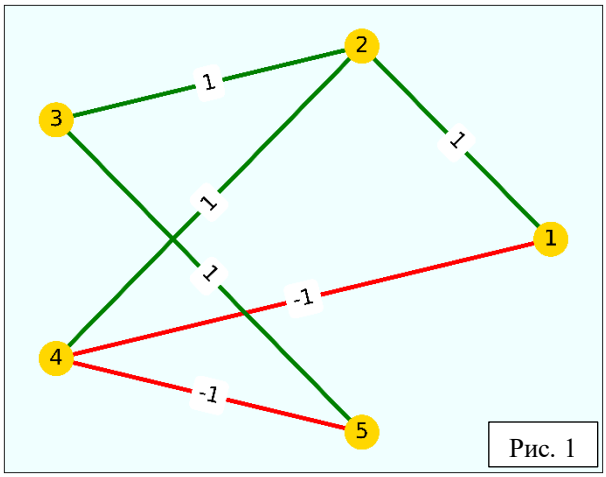

Рис. 1

Другим словами, знаковый граф – это когнитивная математическая модель, позволяющая определить вид связи (положительная и отрицательная) между сущностями. Именно наличие двух видов связи является концептуальным отличием от неподписанного графа, что позволяет моделировать сети с более сложной структурой. Стоит также упомянуть, что графы бывают ориентированными (направленными) и неориентированными. В этой работе будут рассматриваться только неориентированные графы. То есть связи между объектами являются взаимными, и следовательно матрица связей W будет симметричной, так как $w_{ij} = w_{ji}$. Эта условность является упрощением, так как в действительности связи могут быть далеко не взаимными. Чтобы сделать интерпретация знакового графа более наглядной и легко воспринимаемой, можно обусловиться, что множеством вершин являются люди, я связи между ними показывают, дружат они или нет. Пример знакового графа можно наблюдать на рисунке 1.

После определения основных объектов для анализа знаковых графом, можно задаться следующим вопросом: возможно ли разбить знаковый граф на две группы так, чтобы внутри каждой группы любая пара вершин не имела неотрицательных связей, а также любая пара вершин, находящихся не в одной группе, не имела положительных связей? Если же знаковый граф удается так разбить, то его называют **сбалансированным**.

Для определения сбалансированности нужно ввести еще несколько понятий, а именно: знак цикла и знак цепи в знаковом графе. Знаком цикла\цепи в знаковом графе называется произведение знаков (-1, если связь отрицательная и 1, если связь

положительная) всех ребер, входящих в этот цикл\эту цепь. Стоит отметить, что цикл и цепь должны быть простыми, или же для цикла: каждая вершина входит в ребра, используемые в произведении для нахождения знака цикла, ровно 2 раза; для цепи: вершины, между которыми строится цепь, и все остальные вершины цепи входят в ребра, используемые в произведении для нахождения знака цепи, ровно 1 и 2 раза соответственно. Это условие необходимо, чтобы в один цикл\цепь не входили другие цепи, так как они способны поменять знак.

Теорема 1: Знаковый граф сбалансирован если, и только если для любой пары вершин любая цепь (опять же простая), соединяющая их, имеет один и тот же знак.

Теорема 2: Знаковый граф сбалансирован если, и только если любой цикл (опять же простой) имеет положительный знак.

Однако для этих теоремы существует исключение, а именно: если граф имеет структуру дерева (связный граф, без циклов), то и не будет существовать ни одного цикла, для определения сбалансированности, а также любая пара вершин будет иметь только одну цепь (иначе был бы цикл, включающий эти цепи). Поэтому следует помнить, что знаковый граф, имеющий такую структуру, всегда сбалансирован. Докажем это более строго в следующем абзаце.

Доказательство: возьмем произвольную вершину A знакового графа $G = (V, E, W)$, имеющего структуру дерева, и разобьём оставшееся множество вершин на два подмножества X и Y по следующему правилу: $B \in X$, если и только если знак цепи от A до B имеет положительный знак. В противном же случае вершина B принадлежит множеству Y. Так как цепь, связывающая любую пар вершин, единственна (иначе был бы цикл), то данное разбиение будет единственным. Заметим, что знак цепи между любой парой вершин B и C можно задать, как произведение знаков цепей от A до B и от A до C. Если вершина A принадлежит цепи, соединяющей B и C, то можно сказать, что: $sign(B - C) = sign(B - A) * sign(A - C) = sign(A - B) * sign(A - C)$. Если же вершина A не входит в цепь $B - C$, при этом одна из цепей $A - B$ и $A - C$ является под цепью другой, то, не умаляя общности, можно сказать, что $A - B$ является под цепью $A - C$, а следовательно: $sign(A - B) * sign(A - C) = sign(A - B) * sign((A - B) + (B - C)) = sign(A - B) * sign(A - B) * sign(B - C) = (sign(A - B))^2 * sign(B - C) = 1 * sign(B - C) = sign(B - C)$. Если же вершина A не входит в цепь $B - C$, а также ни одна из цепей $A - B$ и $A - C$ не является под цепью другой, то существует такая вершина D, что $B - C = (B - D) + (D - C)$. Тогда, можно сказать, что: $sign(A - B) * sign(A - C) = sign\big((A - D) + (D - B)\big) * sign\big((A - D) + (D - C)\big) = \big(sign(A - D)\big)^2 * sign(D - B) * sign(D -$

$C) = 1 * sign\big((D - B) + (D - C)\big) = sign(B - C)$. Таким образом, знак цепи $B - C$ можно определить, как произведение знаков цепей $A - B$ и $A - C$. Пусть, изначальная вершина A и множество вершин, принадлежащих X, относятся к первой группе, а множество вершин, принадлежащих Y, относится ко второй группе. Тогда, основываясь на полученных результатах, можно сказать, что полученное разбиение на группы единственно и не зависит от выбора первоначальной вершины. Действительно, цепь для любой пары вершин, находящихся в одной группе, будет иметь положительный знак, так как произведение знаков цепей от A до этих вершин будет равно $t * t = t^2 = 1$, где $t = -1$, если вершины находятся во второй группе, и $t = 1$, если вершины находят в первой группе. Если же вершины находятся в разных группах, то цепь, соединяющая их, будет иметь отрицательный знак, так как произведение знаков цепей от A до этих вершин будет равно $t * (-t) = -t^2 = -1$, где $t = -1$, если первая вершина находится во второй группе, и $t = 1$, если первая вершина находится в первой группе. Тогда, между группами будут отсутствовать положительные ребра, иначе бы существовала цепь, соединяющая вершины из разных групп и имеющая положительный знак, а также внутри каждой группы будут отсутствовать отрицательные ребра, иначе бы существовала цепь, соединяющая вершины из одной группы и имеющая отрицательный знак. Последние утверждение как раз соответствует определению сбалансированного знакового графа. Стоит подчеркнуть, что везде в доказательстве под цепью подразумевалась простая цепь. ■

Существуют также несвязанные графы, тогда для определения сбалансированности следует рассматривать каждую компоненту связанности этого графа. Если же каждая компонента связанности этого графа сбалансирована, то и сам граф сбалансирован. Существуют и знаковые графы, не имеющие отрицательных связей. Такие графы называются **абсолютно сбалансированными**, так как все множество вершин можно положить в одну группу, а вторая группа будет пустой.

Довольно маленько множество графов является сбалансированным. Поэтому, для подсчета меры сбалансированности был предложен следующий подход: $\beta(G) = \frac{p}{t} \in [0; 1]$, где $p -$ количество положительных циклов, и $t -$ количество всех циклов, называется **относительной мерой сбалансированности**. Если же граф сбалансирован, то данный коэффициент равен 1. Этот способ подсчета меры сбалансированности имеет и недостатки, такие как: не помогает понять, какие именно вершины и ребра генерируют несогласованность графа, а также не позволяет сравнивать сбалансированность графов, на основе значений этого коэффициента (сравнение является необъективным).

Следующим этап исследования было предложено увеличение количества групп, на которые мы разбиваем множество вершин. Предположим, что граф состоит из трех вершин A, B и C и трех отрицательных ребер. Очевидно, что такой граф не является сбалансированным, но он идеально разбивается на три группы. Еще интуитивно можно понять, что во многих социальных структурах далеко не две крайности, и не удастся разбить множество людей на две согласованные группы. Однако, относительной согласованности можно достичь, если увеличить количество групп.

Если знаковый граф удается разбить на более чем две группы так, чтобы внутри каждой группы были только положительные ребра, а все ребра, соединяющие вершины из разных групп, имели отрицательный знак, то такой граф называется **относительно сбалансированным** или же **группируемым**. **Теорема 1** также верна и для группируемых знаковых графов, однако вторая теорема, говорящая, что знак любого цикл должен быть положительным, в данном случае не всегда верна.

Теорема 3: Знаковый граф группируем тогда, и только тогда, когда он не содержит циклов, включающих только одно отрицательное ребро.

Продолжая тему оценки меры согласованности знакового графа, было непонятно, что делать с группируемыми графами, и какую меру вводить для них. Было предложено в качестве меры группируемости знакового графа, брать ошибку разбиения, или же количество положительных\отрицательных ребер между группами\внутри групп. Однако, значение ошибки разбиения зависит от того, какое разбиение мы выберем, а также от нашей чувствительности к ошибкам разного рода. Пусть $P(C, G, \alpha) = \alpha \cdot P_1 + (1 - \alpha) \cdot P_2$, где $P_1 -$ количество отрицательных связей внутри кластеров (ошибка первого рода), $P_2 -$ количество положительных связей между кластерами (ошибка второго рода), $\alpha \in [0; 1] -$ чувствительность к отрицательным связям внутри кластеров (чем больше значение α, тем сильнее и острее мы реагируем на наличие отрицательных ребер внутри группы), $C = \{c_i : i = \overline{1, k}\} -$ разбиение множества вершин, на кластеры, где $c_i -$ множество вершин, входящих в i-ый кластер.

Следующим продвижением в развитии анализа знаковых графов было расширение меры отношений между вершинами от множества $\{-1, 0, 1\}$ до отрезка $[-R; R]$, где $R -$ максимально возможное отношение между вершинами. Данное допущение позволяет ранжировать отношения между вершинами или понять какая пара объектов дружит или враждует между собой сильнее, чем другая. Безусловно, это допущение приводит к изменениям матрицы смежностей $W = \left[w_{ij}\right]_{n \times n} : w_{ij} \in [-R; R]$, а также подсчета меры сгруппированности знакового графа $P(C, G, \alpha) = \alpha \cdot P_1 + (1 - \alpha) \cdot P_2$, где $P_1 -$ сумма весов

ребер, имеющих отрицательный знак внутри кластеров и взятых по модулю, P_2 – сумма весов ребер, имеющих положительный знак между кластерами (так как они положительны, то и необходимость в модуле отсутствует).

Следующим этапом исследования было нахождение методов выделения кластеров из множества вершин знакового графа. Положительность связей не гарантирует определения этих вершин в один кластер. То же касается и отрицательных связей. Это делает разбиение более неочевидным и требует разработки определенных алгоритмов для поиска согласованных групп.

Алгоритмы кластеризации знаковых графов

Первым методом поиска кластеров в знаковом графе будет алгоритм под названием ***S-method***. Он имеет следующую структуру: строится множество $S = \{s_i : i = \overline{1,n}\}$, где s_i – множество вершин, с которыми согласована (имеет положительные ребра) i-ая вершина (предполагается, что каждая вершина согласована сама с собой). Множество $C = \{\}$, изначально являющееся пустым, будет хранить в себе кластеры. Далее строится i-ый кластер c_i по следующему правилу: $v_j \in c_i$, если $s_j \in s_{max,i}$, где $s_{max,i}$ – максимальное по мощности множество среди всех s_i, входящих в множество S (под мощностью множества подразумевается количество элементов, входящих в это множество), а также прародитель i-ого кластера. После построение кластера $c_i = \{v_{t_j,i} : j = \overline{1,r_i}\}$, где r_i – количество вершин, входящих в кластер c_i, его добавляют в множество C, а все множества s_i с индексами $t_j : j = \overline{1,r_i}$ удаляются из множества S. Данный алгоритм продолжается до тех пора, пока множество S не станет пустым.

Рассмотрим предложенный алгоритм на конкретном примере (рис. 2). Тогда $S = \{s_1 = \{1,4,5\}, s_2 = \{2,3,4\}, s_3 = \{2,3\}, s_4 = \{1,2,4,5\}, s_5 = \{1,4,5\}\}$. Согласно *S-method* первым прародителем для первого кластера будет вершина v_4, так как множество s_4 имеет максимальную мощность. В первый кластер войдут следующие вершины: $c_1 = \{v_1, v_4, v_5\}$, так как $s_1 = \{1,4,5\} \in s_4 = \{1,2,4,5\}$, $s_4 = \{1,2,4,5\} \in s_4 = \{1,2,4,5\}$ и $s_5 = \{1,4,5\} \in s_4 = \{1,2,4,5\}$, а после добавления первого кластера c_1 в множество C, множество S примет вид: $S = \{s_2 = \{2,3,4\}, s_3 = \{2,3\}\}$. Следующим прародителем для второго кластера станет вершина v_2, так как s_2 имеет максимальную мощность. Во второй

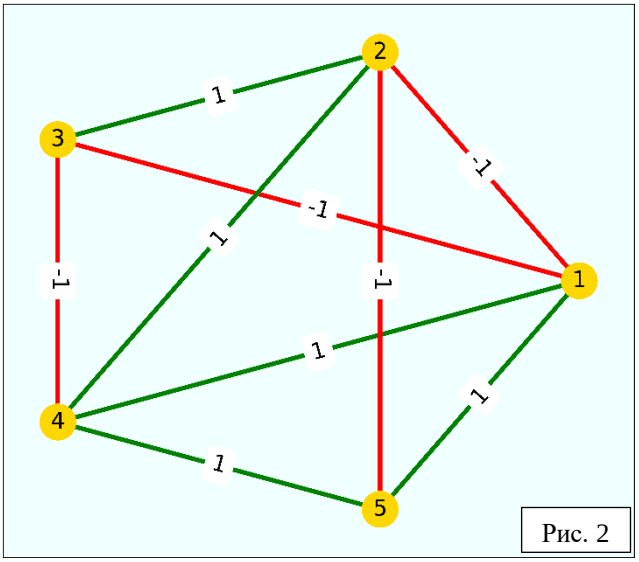

Рис. 2

кластер войдут все оставшиеся вершины: $c_2 = \{v_2, v_3\}$, так как $s_2 = \{2,3,4\} \in s_2 = \{2,3,4\}$ и $s_3 = \{2,3\} \in s_2 = \{2,3,4\}$ и множество S станет пустым. Алгоритм окончен. Полученное разбиение: $C_{S-method} = \{c_1 = \{v_1, v_4, v_5\}, c_2 = \{v_2, v_3\}\}$. Можно оценить полученное разбиение с помощью меры сгруппированности $P(C = C_{S-method}, G, \alpha = 0{,}5) = 0{,}5 \cdot 1 = 0{,}5 -$ что является довольно хорошим показателем.

Следующим методом поиска кластеров в знаковом графе будет алгоритмы под названием $\boldsymbol{Min\text{-}Error}$. Исходя из названия, можно понять, что данный алгоритм минимизирует ошибку разбиения или же $P(C, G, \alpha)$. Первоначально стоит разбить множество вершин случайным образом на кластеры. То есть в качестве гиперпараметра будет выступать количество кластеров k. Далее, перебирая вершины, мы перекладываем их в новые кластеры, если при этом перемещении ошибка уменьшается, или же: v_l перекладывается из c_i в c_j, если $P(C^*, G, \alpha) < P(C, G, \alpha)$, где $C -$ разбиение на кластеры, где $v_l \in c_i$, $C^* -$ новое разбиение на кластеры, где $v_l \in c_j$. Данный алгоритм перемещения вершин из одних кластеров в другие повторяется до тех пор, пока разбиение не станет устойчивым, то есть перемещение любой вершины в любой другой кластер не приведет к уменьшению ошибки разбиения $P(C, G, \alpha)$ (сходимость очевидна, так как бесконечно ошибка уменьшатся не может, она ограничена снизу нулем). Стоит отметить, что полученное разбиение не является абсолютно оптимальным, так как в действительности мы могли найти лишь локальный минимум. Поэтому этот алгоритм нужно повторить l раз, и среди полученных разбиений, выбрать наилучшее. Как упоминалось выше, данный алгоритм в отличие от предыдущего имеет гиперпараметры, а именно: $k -$ количество кластеров в первоначальном случайном разбиении, $l -$ число раз повторения алгоритма, и кончено же $\alpha -$ чувствительность к ошибкам первого рода (отрицательным ребрам внутри групп). Так как данный алгоритм опирается непосредственно на ошибку разбиения, то и α является гиперпараметром.

Третьим методом для нахождения кластеров в знаковом графе будет алгоритм под название $\boldsymbol{FCSG}$ (Fast Clustering Signed Graph). В сравнении двумя вышеупомянутыми алгоритмами, этот метод требует введения новых сущностей и объектов, а также расширение теоретической базы. Согласно этому алгоритму, в первую очередь мы применяем метод $\boldsymbol{RWG}$ (Random Walk Gap) для изменения матрицы смежности. Концептуальной идей метода RWG (или же разность случайных блужданий по графу) является выявление положительных ребер, связывающих вершины из разных кластеров, и уменьшение их веса. Введем несколько понятий, для определения типа ребра.

Положительное ребро (v_i, v_j) знакового графа G называется **внутренним**, если v_i и v_j принадлежат одному кластеру, а также ни одна из вершин v_i или v_j не связана (положительным ребром) ни с одной вершиной из любого другого кластера.

Положительное ребро (v_i, v_j) знакового графа G называется **крайним**, если v_i и v_j принадлежат одному кластеру, и хотя бы одна из вершин v_i или v_j связана (положительным ребром) хотя бы с одной вершиной из любого другого кластера.

Положительное ребро (v_i, v_j) знакового графа G называется **внешним**, если v_i и v_j принадлежат разным кластерам.

Рассмотрим два графа $G' = (V, E', W')$ и $G'' = (V, E'', W'')$ построенных на первоначальном графе $G = (V, E, W)$ по следующим правилам: $E' = E^+$ (под множеством ребер рассматриваются только положительные ребра), $E'' = E^+ + |E^-|$ (под множеством ребер рассматриваются все ребра, веса которых взяты по модулю) и соответствующие им новые матрицы смежности W' и W''. Далее, на полученных матрицах смежностей, строятся матрицы вероятностей перехода θ' и θ'' по следующему правилу:

$$\theta' = \left[\theta'_{ij}\right]_{n\times n} : \theta'_{ij} = \frac{w'_{ij}}{\sum_{j=1}^{n} w'_{ij}};\ \theta'' = \left[\theta''_{ij}\right]_{n\times n} : \theta''_{ij} = \frac{w''_{ij}}{\sum_{j=1}^{n} w''_{ij}}$$

Ячейки θ'_{ij} и θ''_{ij} полученных матриц можно интерпретировать, как вероятность перехода из вершины v_i в вершину v_j в графах G' и G'' соответственно. Далее построим матрицы $H'^{(k)}$ и $H''^{(k)}$ по следующему принципу:

$$H'^{(k)} = \left[h'^{(k)}_{ij}\right]_{n\times n} = \theta'_{ij} \times \ldots \times \theta'_{ij} \text{ и } H''^{(k)} = \left[h''^{(k)}_{ij}\right]_{n\times n} = \theta''_{ij} \times \ldots \times \theta''_{ij}$$

, где $\times -$ оператор умножения матриц, а также количество умножения равно степени k. После, строятся матрицы

$$H^{G'} = \left[h^{G'}_{ij}\right]_{n\times n} = \sum_{k=1}^{L} H'^{(k)} = \left[\sum_{k=1}^{L} h'^{(k)}_{ij}\right]_{n\times n}$$

и

$$H^{G''} = \left[h^{G''}_{ij}\right]_{n\times n} = \sum_{k=1}^{L} H''^{(k)} = \left[\sum_{k=1}^{L} h''^{(k)}_{ij}\right]_{n\times n}$$

, ячейки которых $h^{G'}_{ij}$ и $h^{G''}_{ij}$ можно интерпретировать, как вероятность перехода из вершины v_i в вершину v_j в пределах L шагов. Число шагов или же L мы определяем сами, так что этот переменная является гиперпараметром. Чтобы выяснить разницу между вероятностями перехода из вершины v_i в вершину v_j в случае, когда граф строится только

на положительных ребрах и в случае, когда граф строиться на всех ребрах, веса которых взяты по модулю, построим новую матрицу

$$D = \left[d_{ij}\right]_{n\times n} = \left(H^{G''} - H^{G'}\right)/H^{G'} = \left[\left(h_{ij}^{G''} - h_{ij}^{G'}\right)/h_{ij}^{G'}\right]_{n\times n}$$

, где $/-$ оператор поэлементного деления матриц. Каждая ячейка d_{ij} показывает, во сколько раз изменилась вероятность перехода из вершины v_i в вершину v_j в пределах L шагов, если рассматривать графы G' и G'' (если $h_{ij}^{G'}$ равно нулю, то в таких случая берётся значение очень близкое к нулю, например: $0{,}0001$).

Если рассмотреть множество положительных ребер в терминах их принадлежности к определенному типу (внутренние, крайние, внешние), то становится очевидно (с теоретической точки зрения), что для внешних ребер вероятность перехода увеличится на большую величину, так как между кластерами, включающих в себя вершины v_i и v_j, для которых строится это ребро, находится много отрицательных ребер. Для крайних ребер также будет наблюдаться увеличение вероятности, хотя и не на такую большую величину, как для внешних ребер. Внутренние же ребре почти не изменят своих вероятностей, так как они соединяют вершины, не имеющие положительных связей с вершинами из других кластеров. Немного преобразуя матрицу D, а также нормируя и усредняя значения её ячеек, получим матрицу весов для отношения в матрице смежностей и саму матрицу смежностей:

$$D^* = \left[d_{ij}^*\right]_{n\times n} : d_{ij}^* = \begin{cases} d_{ij}, & \text{если } d_{ij} > 0 \\ 0, & \text{иначе} \end{cases};$$

$$H = \left[h_{ij}\right]_{n\times n} = exp\left(-0.5 \times \left(D^* + D^{*T}\right)\right) = \left[e^{-\frac{d_{ij}^* + d_{ji}^*}{2}}\right]_{n\times n};$$

$$W^* = \left[w_{ij}^*\right]_{n\times n} : w_{ij}^* = \begin{cases} w_{ij} \cdot h_{ij}, & \text{если } w_{ij} > 0 \\ w_{ij}, & \text{иначе} \end{cases}$$

, где $D^{*T} -$ транспонированная матрица D^*, $exp(X) -$ поэлементное экспонирование матрицы X. Как и упоминалось выше, данный метод разности случайного блуждания выделяет положительные ребра, соединяющие вершины, находящиеся в разных кластерах, и уменьшает значение связей для них. Более интуитивное объяснение было упомянуто выше в терминах типов ребер (внешние, крайние, внутренние), где «ненастоящая» положительная связь отслеживается с помощью увеличения вероятности перехода по ребру, в случае, когда берется граф без отрицательных ребер и в случае, когда значения всех ребер берутся по модулю.

После применения метода *RWG* и получения новой матрицы смежностей W^*, можно переходить, непосредственно, к кластеризации знакового графа. В новом графе G^* берется положительное ребро с максимальным значением связи, пусть это будут вершины v_i и v_j.

Эти вершины объединяются в одну, а всевозможные связи вида (v_i, v_t) и (v_j, v_t) складывается $(w_{it}^* + w_{jt}^*)$, и следовательно, получается новая матрица смежностей: $W^* \rightarrow W_1^*$, где $W_1^* = \left[w^*{}_{1_{ij}}\right]_{(n-1)\times(n-1)}$. Если же ребер, имеющих максимальную связанность несколько, то выбирается любая из них. Данная операция слияния повторяется до тех пор, пока не останутся только отрицательные ребра.

Рассмотрим предложенный метод на конкретно примере (рис. 3). Изначальная матрица смежностей имеет следующий вид:

$$W = \begin{pmatrix} 0 & 5 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 5 & 0 & 5 & 0 & 0 & 3{,}2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 1 & 0 & 0 & -5 & 0 & -6 \\ 2 & 0 & 1 & 0 & 3 & -4 & 0 & 0 & 0 \\ 3 & 0 & 0 & 3 & 0 & -5 & 0 & 0 & 0 \\ 0 & 3{,}2 & 0 & -4 & -5 & 0 & 3 & 3 & 3 \\ 0 & 0 & -5 & 0 & 0 & 3 & 0 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 2 \\ 0 & 0 & -6 & 0 & 0 & 3 & 3 & 2 & 0 \end{pmatrix}$$

Если же применить *RWG*, то матрица смежностей преобразуется в новую:

$$W^* = \begin{pmatrix} 0 & 5 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 5 & 0 & 4{,}7042 & 0 & 0 & 2{,}7335 & 0 & 0 & 0 \\ 0 & 4{,}7042 & 0 & 1 & 0 & 0 & -5 & 0 & -6 \\ 2 & 0 & 1 & 0 & 3 & -4 & 0 & 0 & 0 \\ 3 & 0 & 0 & 3 & 0 & -5 & 0 & 0 & 0 \\ 0 & 2{,}7335 & 0 & -4 & -5 & 0 & 3 & 3 & 3 \\ 0 & 0 & -5 & 0 & 0 & 3 & 0 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 2 \\ 0 & 0 & -6 & 0 & 0 & 3 & 3 & 2 & 0 \end{pmatrix}$$

Если же попытаться кластеризировать граф W без применения метода разности случайного блуждания (*RWG*), то в результате получиться следующее разбиение: $C_{Fusion} = \{c_1 = \{1,2,3,6\}, c_2 = \{4,5\}, c_3 = \{7,8,9\}\}$, для которого $P(C = C_{Fusion}, G, \alpha = 0{,}5) = 7{,}5$. Для нового графа G^* получается новое разбиение, всего с двумя кластерами: $C_{FCSG} = \{c_1 = \{1,2,3,4,5\}, c_2 = \{6,7,8,9\}\}$, для которого $P(C = C_{FCSG}, G, \alpha = 0{,}5) = 1{,}6$ – что гораздо меньше чем для разбиения C_{Fusion} без применения метода *RWG*.

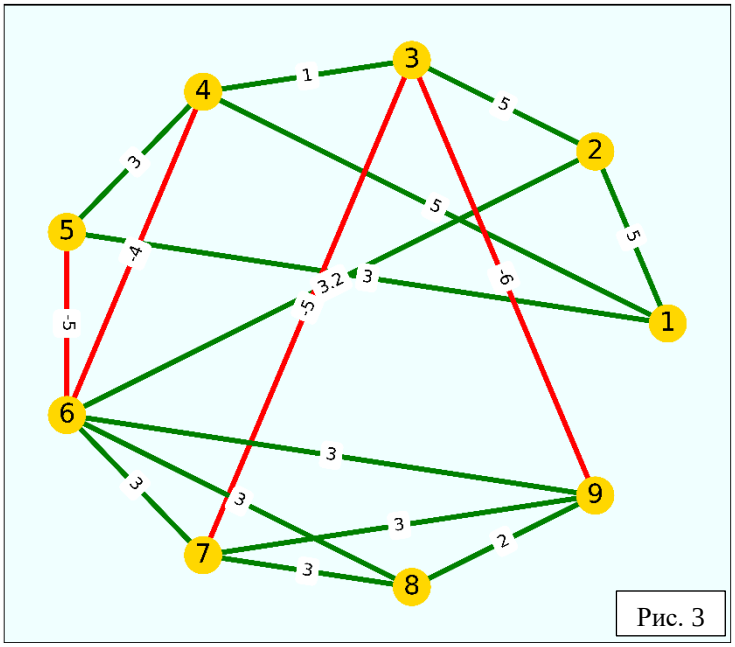

Рис. 3

Особенности приведенных алгоритмов

Первый из методов кластеризации знаковых графов под названием $S\text{-}method$ является самым упрощенным, статичным и быстрым. Он не требует определения каких-либо гиперпараметров, что является огромным плюсом. Однако, у данного алгоритма есть и огромные недостатки, такие как: он не видит разницы между отрицательными связями и отсутствием связей, для него они определены как одно множество и обозначаются как 0. Можно представить случай, когда вершина v_1 имеет положительные ребра с девятью другими вершинами $\{v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$, при этом каждая из этих вершин «не дружит» ни с кем, кроме как с v_1. Согласно этому методу, несмотря на большое количество отрицательных связей внутри группы, все вершины $v_i: i = \overline{1,10}$ войдут в один кластер, где прародителем будет вершина v_1. Также, данный метод не распознает разные веса отношений, то есть ребра со связями 100 и $0{,}01$ инициализируются, как просто положительны, однако очевидно, что между ними присутствует существенная разница. И конечно же, если в какой-то паре вершин v_i и v_j, каждый будет иметь такого друга с кем не дружит второй, то эта пара вершин не войдет в один кластер, на смотря на количество общих друзей этих вершин, так как ни одно из множеств s_i и s_j не является подмножеством другого. Рассмотрим случай, когда: $s_1 = \{v_i: i = \overline{1,20}\} \cup \{v_{21}\}$ и $s_2 = \{v_j: j = \overline{1,20}\} \cup \{v_{22}\}$. Согласно данному методу, вершины v_1 и v_2 не войдут в один кластер несмотря на то, что они имеют целых 18 общих друзей и всего по 1 необщему другу. Все эти проблемы в совокупность с большой чувствительностью группировки приводят к результатам, где множество вершин, состоящее из 4000 вершин, разбивается на $1500 - 2000$ кластеров, что очень много и выглядит нереалистичным.

Следующий по очередности изложения метод под название $Min\text{-}Error$, в отличие от предыдущего метода, является довольно медленным и не статичным, из-за случайности разбиения на первом этапе. Также, данный алгоритм требует определения большого числа гиперпараметров, таких как: $k -$ число кластеров, $l -$ количество повторений алгоритма, $\alpha -$ чувствительность к вершинам внутри кластеров. Последний параметр относительно легко определить, однако с другими все не так однозначно. Можно предположить, что количество повторений l функционально зависит от количества всевозможных начальных случайных разбиений, однако количество последних растет с около экспоненциальной скоростью:

$$\frac{n! \cdot C_{n-1}^{k-1}}{\left(\left\lceil \frac{n}{k} \right\rceil !\right)^k}$$

$-$ нестрого оцененное количество первоначальных разбиений, где $n!-$ количество способов перемещать вершины в любом порядке на прямой, $C_{n-1}^{k-1}-$ количество способов выбрать $k-1$ перегородку между n вершинами для k кластеров, $\left(\left\lceil\frac{n}{k}\right\rceil!\right)^k-$ среднее количество перестановок внутри кластеров, $\left\lceil\frac{n}{k}\right\rceil-$ округление вверх. Также стоит упомянуть, что в ходе каждого из l повторений этого алгоритма каждая вершина перемещается в каждый другой кластер или же совершается n^{k-1} операций неопределенное количество раз r до тех пор, пока разбиение не станет устойчивым. Эти два фактора в совокупности дают очень медленную скорость данного алгоритма и не пригодность его использования для больших знаковых графов, имеющих тысячи вершин и сотни тысяч ребер. Что касается минимальности полученного разбиение, то в действительности, каждый раз мы падаем в локальный минимум, и для нахождения относительно глобального минимума требуется большое количество повторений l.

И в качестве последнего из предложенных алгоритмов рассмотрим метод под названием $FCSG$. В отличие метода, минимизирующего ошибку разбиения, этот алгоритм очень быстрый. Также, он требует инициализации только одного гиперпараметра $L-$ длина пути в случайном блуждании на графе. Можно предположить, что длина пути как-то функционально зависит от количества вершин и количества положительных и отрицательных ребер, однако эта зависимость не является очевидно и тривиальной. Поэтому, следует перебрать небольшой спектр возможных значений L и посмотреть, изменяются ли полученные кластеры или же разница состоит только в очередности соединения вершин. Стоит отметить, что очень большое значение L помимо того, что сильно замедлит алгоритм (за счет возведения матриц смежностей в большую степень), так еще и значения новой матрицы потеряют интерпретируемость. Также, следует упомянуть об эксперименте в маленьком мире, где утверждается, что любую пару людей в среднем разделяет всего $5-6$ рукопожатий, однако не следует это воспринимать рекомендуемое количество шагов, ведь граф может быть коридорного типа и иметь «вытянутую» структуру.

Последний метод способен генерировать очень неплохие результаты, однако не наилучшие (метод $Min\text{-}Error$ всегда получает результаты не ниже, чем $FCSG$, а зачастую и выше). Можно попытаться решить этот вопрос создав композицию этих двух методов, а именно: вместо того, чтобы в самом начале разбивать вершины случайным образом на кластеры, используем результаты алгоритма $FCSG$ как нулевое разбиение. Тогда, наша близость к относительно глобальному минимуму будет высокой, и останется перетащить не так много вершин, как при случайном разбиении. Также, этот метод будет статичным,

так как в нем отсутствует случайность, а следовательно, пропадает надобность в определении гиперпараметра l, отвечающего за количество повторений алгоритма $Min\text{-}Error$. И наконец, количество кластеров будет определено автоматически по результатам выполнения алгоритма $FCSG$. Однако, у данного метода есть и недостатки, такие как: количество кластеров после применения алгоритма $FCSG$ зачастую существенно (в рамках алгоритма $Min\text{-}Error$), и даже наличие 10 групп способно весомо увеличить время выполнения кластеризации (не стоит забывать, как зависит количество перестановок каждой вершины в каждый кластер зависит от количества вершин и количества кластеров: n^{k-1}). Суммируя, можно сказать, что этот метод немного расширяет допустимое множество значений числа кластеров, так как нулевое разбиение гораздо ближе находится к относительно глобальному минимуму, и следовательно, потребуется меньше вершин переложить в другие группы, а также является статичным (отсутствует случайность), что не требует повторения алгоритма, до достижения лучшего результата. Назовем этот метод ***Composion of FCSG and M-E algorithms***.

Применение приведенных алгоритмов на реальных данных

Для того, чтобы сравнить вышеупомянутые алгоритмы поиска кластеров в знаковом графе и подчеркнуть их особенности, в данном разделе будет рассмотрен знаковый граф, включающий в себя сетевую структуру взаимоотношений героев, принимавших участие в одной из частей серии фильмов «Гарри Потер». Всего в графе 65 вершин и 513 ребер. Dataset находится в открытом доступе. Согласно данным, полученный граф является ориентированным, так как матрица смежности не является симметричной. Для этого мы немного преобразуем матрицу отношений по следующему правилу:

$$W \to \dot{W} = \left[\dot{w}_{ij}\right]_{n\times n} : \dot{w}_{ij} = \begin{cases} \dfrac{w_{ij} + w_{ji}}{2}, & \text{если } w_{ij} > 0 \text{ и } w_{ji} > 0 \\ \min(w_{ij}, w_{ji}), & \text{иначе} \end{cases}$$

, которое имеет следующее подкрепление: если кто-то из пары персонажей не дружит с другим, то и сама пара является недружелюбной; если каждый из пары персонажей имеет с другим неотрицательную связь (пара не враждует между собой), то отношение пары определяется как среднее отношений каждого из персонажей к другому. Данное преобразование сделает матрицу смежностей симметричной, а следовательно, граф будет неориентированным и можно будет применять рассмотренные методы кластеризации. Также, в качестве ошибки $P(C, G, \alpha)$ использовалась другая формула: $P^*(C, G, \alpha) = 2 \cdot (\alpha \cdot$

$P_1 + (1 - \alpha) \cdot P_2$). Это было сделано с целью упрощения подсчета ошибки на матрице (в матрице смежностей каждое ребро упоминается дважды: w_{ij} и w_{ji}).

Стоит упомянуть, что все алгоритмы были реализованы на языке Python версии 3.9, который, в свою очередь, считается довольно медленным, и не утверждается, что их реализация является оптимальной с точки зрения времени выполнения каждого из методов.

На рисунке 4 представлен результат работы алгоритма $S\text{-}method$. Как можно заметить, число кластеров является довольно большим (27), что говорит о нестандартности разбиения. Также можно заметить, что между кластерами наблюдается большое количество положительных ребер, что сгенерировало большую часть ошибки разбиения. Многие кластеры могли быть объединены между собой, при этом данное перемещение поспособствовало бы сильному уменьшению ошибки разбиения, однако $S\text{-}method$, как уже упоминалось выше, является очень чувствительным к отношениям и малейшее несоответствие в множестве вершин s_i, приводит к определению персонажей, возможно схожих по отношению к другим, в разные группы.

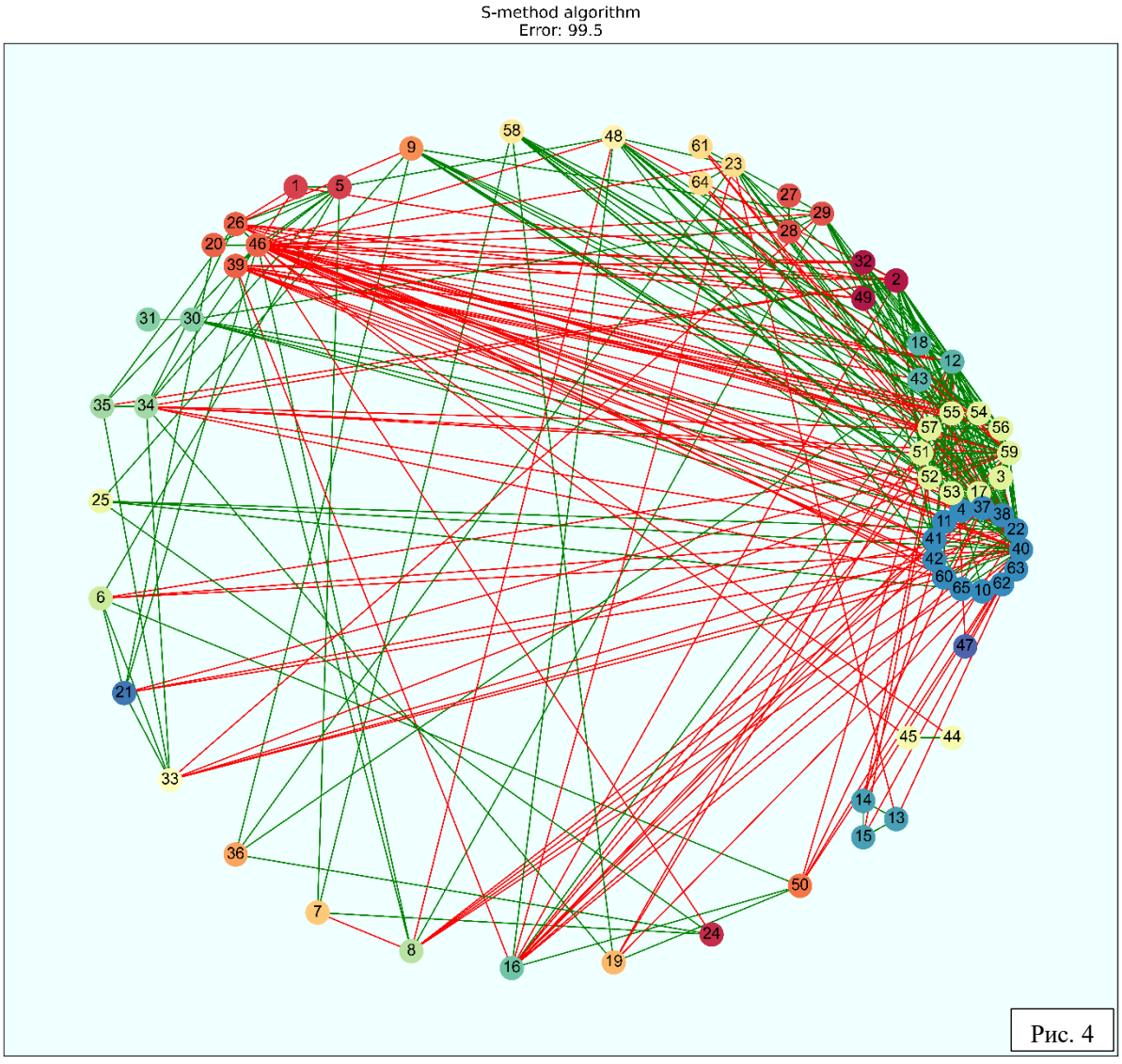

Рис. 4

Следующим методом кластеризации был $Min\text{-}Error$. Результаты применения алгоритма можно наблюдать на рисунке 5. Как можно заметить, четко выделяются 3 кластера, между которыми очень мало положительных связей, а внутри групп доминируют ребра, имеющие положительный знак. Полученное разбиение является относительно минимальным. По результатам более тысячи повторений этого алгоритма, результата меньше, чем 9,5 не наблюдалось. В качестве гиперпараметров было указано, что $k = 3 -$ количество кластеров и $l = 20 -$ число повторений алгоритма. Параметр α, как и для всех других методов, брался по умолчанию как $0{,}5$. Стоит отметить, что полученное разбиение не единственно, и подобной ошибки можно достичь для следующего числа кластеров: $3, 5, 7$ и 10. Список не является исчерпывающим, и допускает наличия разбиения на кластеры при большем числе групп.

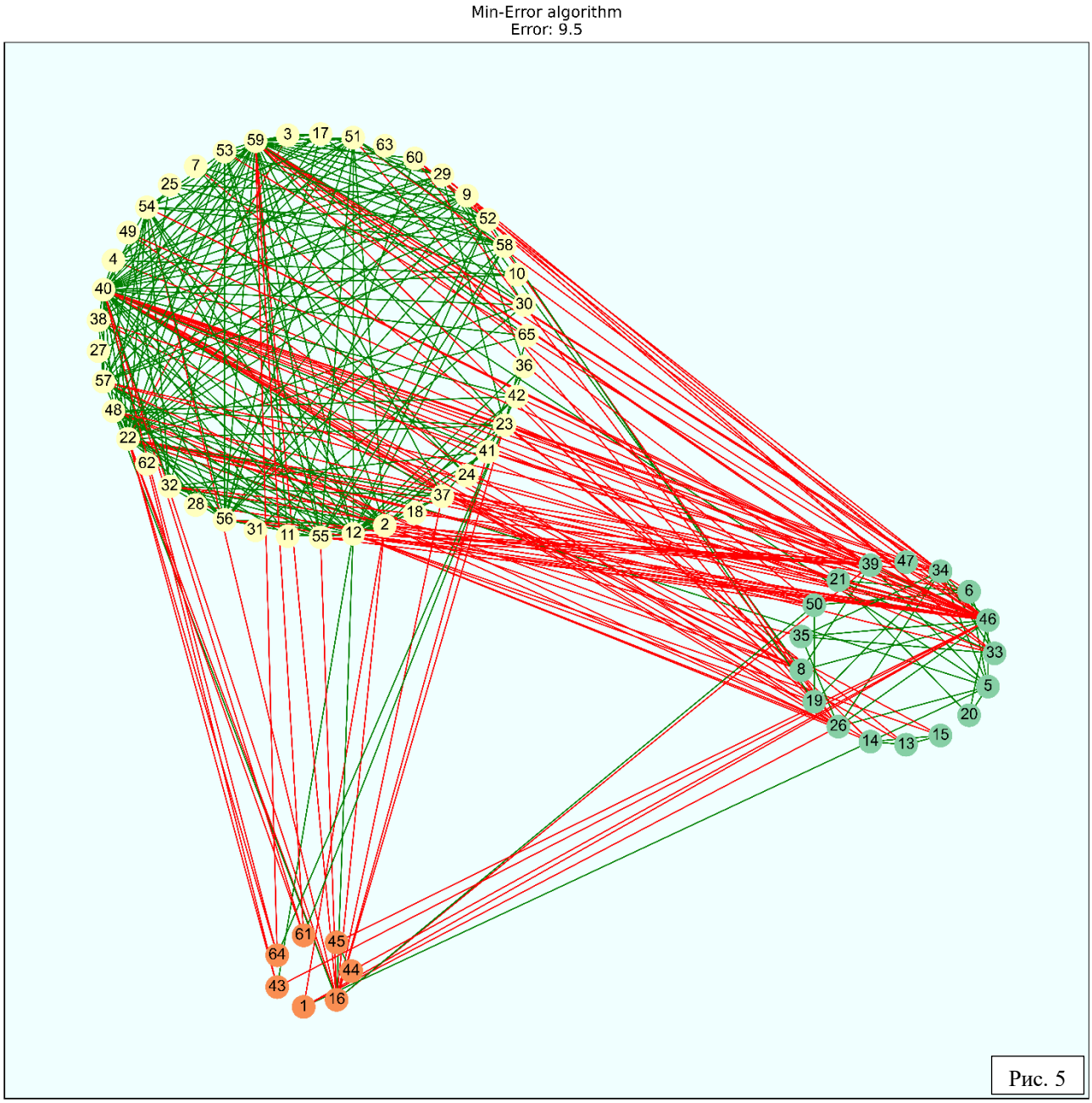

Рис. 5

И наконец, на рисунке 6 изображен результат кластеризации алгоритма $FCSG$. Как можно заметить, ошибка разбиения довольно маленькая и близка к относительному глобальному минимуму, полученному по методу $Min\text{-}Error$. Число кластеров больше, чем для алгоритма, минимизирующего ошибку разбиения, одна не настолько большое, как у метода $S\text{-}method$. Внутри самых массивных кластеров доминируют положительные ребра, что говорит о высокой согласованности вершин внутри групп. Стоит также заметить, что присутствуют единичные вершины, которые можно было бы объединить в один кластер без увеличения ошибки, но данный алгоритм не производит слияние вершин, если они не связаны положительно. В действительности же нейтральная связь никак не противоречит хорошему разбиению и согласованности внутри групп.

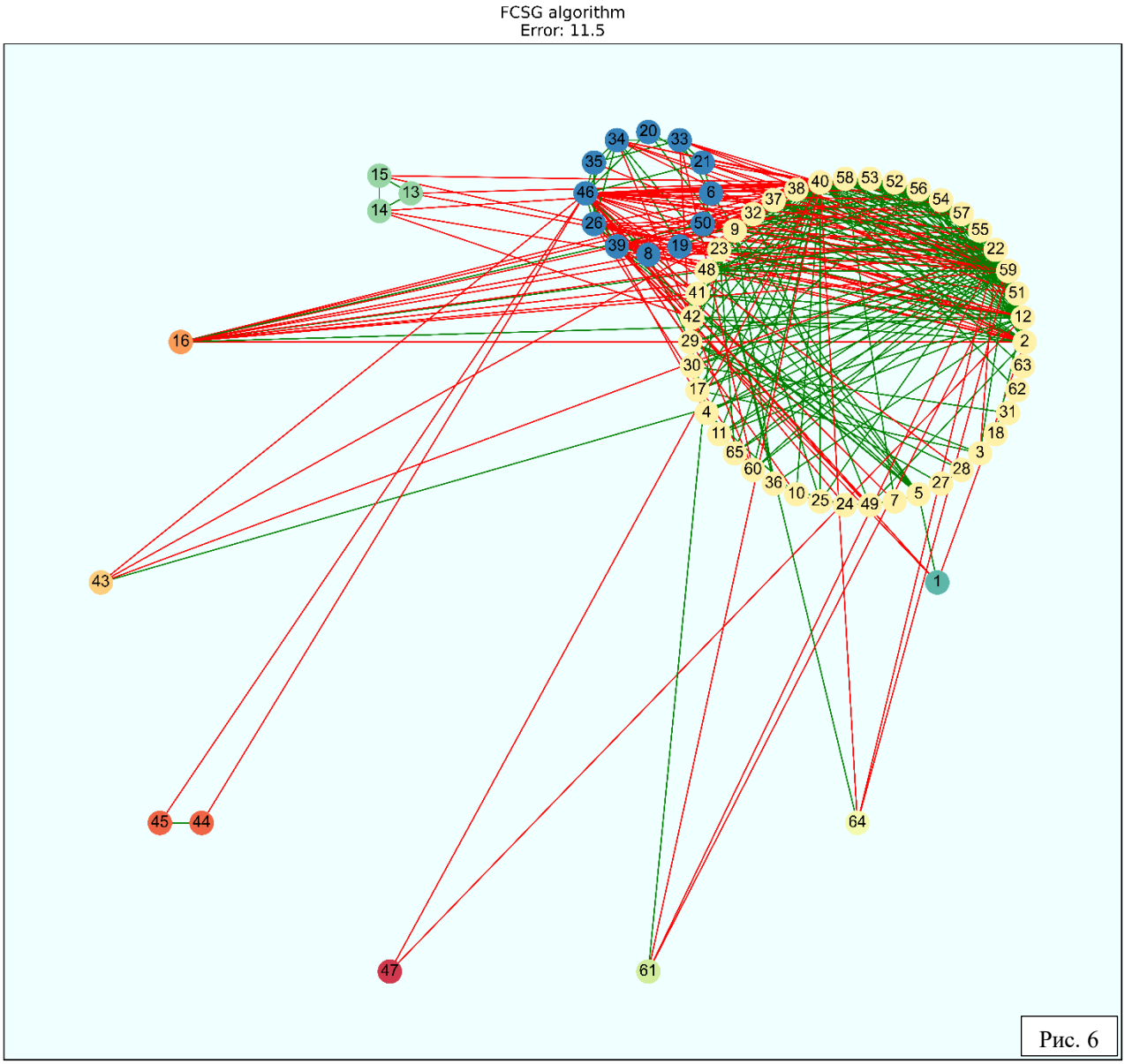

Рис. 6

По результатам применение метода $Composion\ of\ FCSG\ and\ M\text{-}E\ algorithms$ было получено разбиение для 10 кластеров. Само разбиение не требует визуализации, так как оно почти никак не отличается от разбиения, проиллюстрированного на рис. 6 для метода $FCSG$ (всего 2 вершины переместились из одного кластера в другой). Ключевым отличием применения данного метода от алгоритма $FCSG$ является то, что он достиг относительно глобального минимума (9,5) и за относительно неплохое время.

Чтобы более многогранно сравнить полученные разбиения, а также установить зависимость ошибки разбиения для каждого кластера от времени выполнения, можно обратиться к рисунку 7, где представлена основная информация. В момент времени $t = 0$ используется разбиение, где каждая вершина состоит в кластере, состоящим только из нее самой. Под этим разбиением и предполагается нулевая ошибка разбиения.

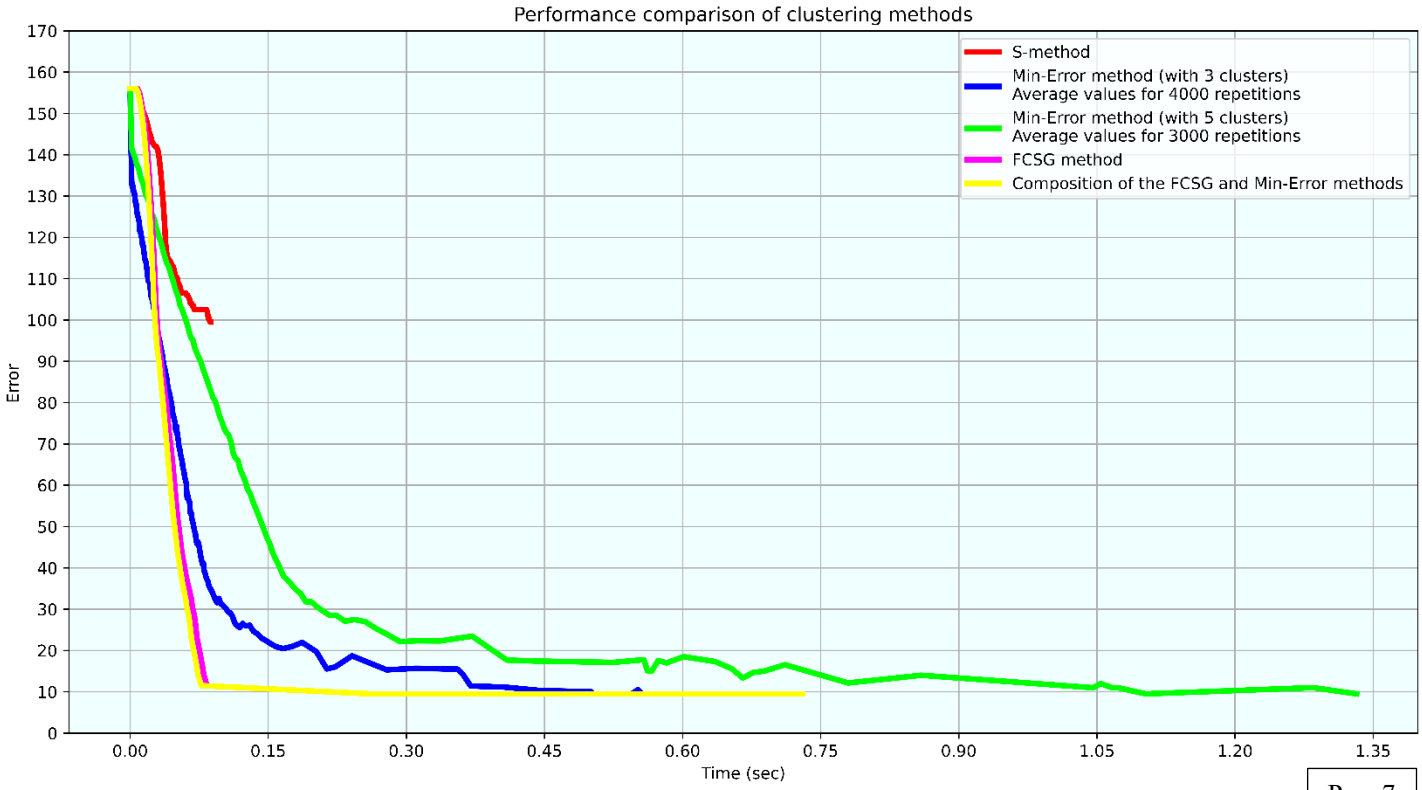

Рис. 7

Как можно заметить по графикам, самым быстрым с точки зрения выполнения алгоритма является метод $FCSG$, также для него скорость убывания ошибки разбиения самая большая. Метод $Composion\ of\ FCSG\ and\ M\text{-}E\ algorithms$ полностью совпадает с методом $FCSG$ в скорости убывания ошибки на первом участке времени, что довольно очевидно ввиду структуры самого метода. После достижения ошибки в 11,5 скорость сильно замедляется и алгоритм начинает перебирать все вершины, чтобы перетащить каждую в нужный кластер. Так как количество кластеров относительно большое (в рамках

минимизации ошибки разбиение), то алгоритм 99% времени перебирает «плохие» варианты, когда какое-то множество вершин не требует никаких перемещений, и очень редко натыкается на вершин, перемещение которые приведет к уменьшению ошибки.

Разбиение с использованием метода $S\text{-}method$ имеет хорошую скорость на протяжении почти всего времени выполнения кластеризации, однако выполнение алгоритма быстро заканчивается и значение ошибки доходит до 99,5, так и не сумев приблизиться к показателям других методов.

Синий и зеленый график показывают среднюю зависимости ошибки от времени выполнения алгоритма $Min\text{-}Error$ для 3 и 5 кластеров, а также 4000 и 3000 тысяч повторений соответственно. Как можно заметить, в самом начале значение ошибки уменьшается на $15 - 20$ пунктов, что показывает среднее изменение ошибки при случайном разбиении. Далее кривые принимают форму прямой, это можно объяснить тем, что на начальном этапе ошибки разбиения распределены по вершинам равномерно. Высокую же скорость, на начальном этапе можно объяснить тем, что очень много вершин находят не в «свои» кластеры, и следовательно, большая часть вершин перетаскиваются из одной группы в другую в рамках одного цикла. Со временем остаются только «проблемные» вершины, которые приходится перетаскивать по одной и проверять все остальные на необходимость перемещения в другой кластер заново. Так как все больше вершин стоят на «своих местах», то и скорость уменьшения ошибки замедляется. Это можно заметить при достижении в примерно 31 и 38 пунктов для синей и зеленой кривой соответственно. Стоит напомнить, что это усредненные значения, поэтому выполнение этого метода на протяжении 0,6 секунд не гарантирует достижение относительно глобального минимума. По результатам 10000 повторений метода $Min\text{-}Error$ для $3 \mid 5$ кластеров были получены следующие значения соответственно: среднее время выполнение одного повторения $-0{,}2596 \mid 0{,}5077$ секунды, средняя ошибка разбиения $-13{,}268 \mid 15{,}0689$, доля разбиений, достигших относительно глобального минимума $(9{,}5) \; -27{,}48\,\% \mid 7{,}62\,\%$, при этом $95{,}33\,\% \mid 91{,}55\,\%$ полученных разбиений не превышают ошибки в $19 \mid 19{,}5$ пунктов. Полученные результаты говорят нам о том, что в среднем для достижения относительного глобального минимума в разбиение на 3 и 5 кластеров требуется 4 и 13 повторений соответственно. Также, стоит упомянуть, что скорость этого метода очень сильно зависит от количества вершин и количества кластеров, что делает рассматриваемые данные удобными и пригодными к кластеризации с помощью данного метода. В более сложных и больших структурах знаковых графов применение данного метода может быть затруднительным.

Заключение

В данной работе были рассмотрены основные теоретические вопросы, касаемо определения сбалансированности и группируемости знакового графа. Были рассмотрены и реализованы 3 алгоритма выделения кластеров для графов с сетевой структурой, а также предложен новый метод, являющий комбинацией двух приведенных. Помимо вышеупомянутых методов кластеризации знакового графа, также существует еще большое количество алгоритмов выделения кластеров, и каждый из них имеет свою спецификация и свои особенности. Чтобы более комплексно и оптимально подойти к выделению групп, всегда будет верным решением рассмотреть сразу несколько алгоритмов, чтобы добиться наилучшего баланса в ошибки разбиения и скорости выполнения.

Безусловно, с каждым годом популярность и востребованность сетевых структур растет, и следовательно, приложение знаковых графов будет проникать все в большее число направлений. Знаковые графы находят применение в таких сферах, как поиск и отслеживание мошенников среди пользователей сети, разработка рекомендательных систем для пользователей с одинаковыми интересами, предсказание отношений между пользователями, построение ML моделей для разных объектов исследования и многие другие.

Список использованной литературы

1) Алескеров Ф. Т., Фабина Э. Л., Шварц Д. А. – «Бинарные отношения, графы и коллективные решения» - 2012 г.
2) Дорогов А. Ю. – «Математические основы методов оптимальной частичной балансировки знаковых графов» - 2007 г.
3) Коровин Д. И., Мазутский Н. М. – «Применение теории сбалансированности когнитивных графов для решения задач управления экономическими процессами»
4) Леденева Т. М., Погосян К. С. – «Спектральный подход к анализу экспертной группы» - 2018 г.
5) Погосян К. С. – «Определение сбалансированности знакового графа экспертной группы» - 2015 г.
6) Семенова Д., Ибрагимова Э. – «Распознавание k - кластеризуемости знаковых графов» - 2021 г.
7) Jualin Hua, Jian Yu, Miin-Shen Yang – «Fast clustering for signed graphs based on random walk gap»
8) Mrvar A. – «Balanced and partitionable signed graph»