

Построение минимальных вершинных расширений графа методом Рида-Фараджеева

И.А.К. Камил, М.Б. Абросимов, А.А. Лобов

Аннотация — Для исследования полной отказоустойчивости при отказе элементов дискретных систем в 1976 году Джон Хейз предложил модель, основанную на графах. Граф G^* называется вершинным k -расширением графа G , если после удаления любых k вершин из графа G^* граф G вкладывается в получившийся граф. Вершинное k -расширение графа G называется минимальным, если оно имеет наименьшее число вершин и рёбер среди всех вершинных k -расширений графа G . Позднее совместно с Фрэнком Харари модель была распространена и на отказы связей между элементами. Задача проверки вершинного k -расширения графа является NP-полной. Для некоторых видов графов удалось найти аналитическое решение задачи построения минимальных вершинных k -расширений: цепи, циклы, звёзды и др. В общем случае граф может иметь много неизоморфных минимальных вершинных k -расширений. Переборный алгоритм построения всех неизоморфных минимальных вершинных k -расширений заданного графа можно использовать только для графов с небольшим числом вершин. В некоторых случаях можно аналитически установить вектор степеней минимального вершинного k -расширения, что позволяет осуществить поиск намного быстрее. В работе предлагается алгоритм построения всех неизоморфных минимальных вершинных k -расширений заданного графа без проверки на изоморфизм методом канонических представителей в форме Рида-Фараджеева. Рассматривается программная реализация этого алгоритма в 4 модификациях и анализируются результаты их работы.

Ключевые слова — отказоустойчивость, расширение графа, исключение изоморфизма, каноническая форма.

I. ВВЕДЕНИЕ

В 1976 г. John P. Hayes предложил основанную на графах модель для исследования отказоустойчивости дискретных систем [1]. Технической системе сопоставляется граф. Элементам системы соответствуют вершины графа, а связям между

элементами – рёбра или дуги графа. Под отказом элемента системы понимается удаление из графа системы соответствующей вершины вместе со всеми её рёбрами. Позднее совместно с Frank Harary модель была распространена на отказы связей [2]. Формализацией отказоустойчивой реализации системы является расширение графа [3]. Напомним, что неориентированным графом (далее просто графом) называется пара $G = (V, \alpha)$, где α – симметричное и антирефлексивное отношение на множестве вершин V , называемое отношением смежности. Если $(u, v) \in \alpha$, то говорят, что вершины u и v смежны, и эти вершины соединены ребром (u, v) . При этом (u, v) и (v, u) – это одно и то же ребро, которое обозначают $\{u, v\}$. Говорят, что ребро $\{u, v\}$ инцидентно каждой из вершин u и v , и эти вершины называются концевыми вершинами или концами ребра $\{u, v\}$. Степенью вершины u в графе G будем называть количество вершин в G , смежных с u , и обозначать через $d(u)$. Основные определения даются по работе [4].

Граф $G^* = (V^*, \alpha^*)$ называется вершинным k -расширением (k – натуральное) графа $G = (V, \alpha)$ если граф G вкладывается в каждый подграф графа G^* , получающийся удалением любых его k вершин.

Граф $G^* = (V^*, \alpha^*)$ называется минимальным вершинным k -расширением (МВ- k -Р) n -вершинного графа $G = (V, \alpha)$, если выполняются следующие условия:

1. граф G^* является вершинным k -расширением графа G ;
2. граф G^* содержит $n + k$ вершин;
3. α^* имеет минимальную мощность среди всех графов, удовлетворяющих условиям 1 и 2.

В работе [1] были предложены схемы построения минимальных вершинных 1-расширений для цепей и циклов. В работе [5] было доказано, что задача построения вершинных k -расширений является вычислительно сложной. В данной работе будет рассмотрен переборный алгоритм построения всех минимальных вершинных k -расширений для заданного графа, а также алгоритмы построения всех неизоморфных минимальных вершинных k -расширений без проверки на изоморфизм. Мы будем в основном рассматривать случай $k = 1$, хотя все описанные методы работают и для общего случая. Напомним, что два графа $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$ называются изоморфными, если можно установить взаимно однозначное

Статья получена 13 февраля 2020.

И. А. К. Камил, аспирант, ФГБОУ ВО СГУ имени Н.Г.Чернышевского, Саратов, Россия (e-mail: kamil.iahab@mail.ru).

А. А. Лобов, аспирант, ФГБОУ ВО СГУ имени Н.Г.Чернышевского, Саратов, Россия (e-mail: aisaneikai@mail.ru).

М. Б. Абросимов, д.ф.-м.н., зав. кафедрой, ФГБОУ ВО СГУ имени Н.Г.Чернышевского, Саратов, Россия (e-mail: mic@ Rambler.ru).

Работа выполнена при поддержке Минобрнауки России в рамках выполнения государственного задания (проект № FSRR-2020-0006)

соответствие $\phi: V_1 \rightarrow V_2$, сохраняющее отношение смежности: (u, v) – ребро $G_1 \leftrightarrow (\phi(u), \phi(v))$ – ребро G_2 .

II. ПОСТРОЕНИЕ МИНИМАЛЬНЫХ ВЕРШИННЫХ РАСШИРЕНИЙ ГРАФА

Для построения всех неизоморфных минимальных вершинных k -расширений в работе [3] был предложен общий переборный алгоритм. Непосредственно его можно использовать для вычисления расширений графов с небольшим количеством вершин [7].

Алгоритм 1. Построение всех минимальных вершинных k -расширений графа.

1. $m := 0$, список $E := \emptyset$.
2. Перечисляем все графы, получающиеся из G добавлением k вершин и m рёбер. Добавляем в список E графы, которые являются k -расширениями графа G и не изоморфны ни одному графу в списке E .
3. Если на шаге 2 не было найдено графов, то $m := m + 1$, и переходим на шаг 2.

В алгоритме 1 можно выделить несколько подзадач. Рассмотрим каждую из них.

1. Перечисление графов, содержащих заданный граф, с m дополнительными рёбрами.
2. Проверка на k -расширение.
3. Исключение изоморфных копий.

Дополнительно можно выделить еще одну задачу:

4. Организация параллельных вычислений.

Напомним, что каждому n -вершинному графу $G = (V, \alpha)$ можно взаимно однозначно сопоставить матрицу смежности A размерности $n \times n$: элемент A_{ij} матрицы A равен 1, если $\{v_i, v_j\} \in \alpha$, в противном случае $A_{ij} = 0$.

III. ПЕРЕЧИСЛЕНИЕ ГРАФОВ С ЗАДАННЫМ ЧИСЛОМ РЕБЕР

Обозначим через $\bar{\alpha} = \{\bar{e}_1, \dots, \bar{e}_t\}$ множество рёбер, которые отсутствуют в графе $G = (V, \alpha)$. Каждому подмножеству $w = \{\bar{e}_{i_1}, \dots, \bar{e}_{i_r}\} \subseteq \bar{\alpha}$ можно однозначно сопоставить два объекта: упорядоченный вектор индексов $\vec{w} = (i_{j_1}, \dots, i_{j_k})$ входящих в него рёбер, где $i_{j_1} \leq \dots \leq i_{j_k}$, и граф $G_w = G + \{\bar{e}_{i_1}, \dots, \bar{e}_{i_r}\}$. Вектора индексов можно организовать в дерево префиксов. Данному дереву будет соответствовать дерево T_G . Все графы, которые можно получить добавлением к G m рёбер, будут располагаться на уровне m данного дерева, поэтому можно свести первую подзадачу к перечислению узлов дерева T_G на уровне m .

IV. ПРОВЕРКА НА РАСШИРЕНИЕ

Вторая подзадача является достаточно сложной. Для проверки на расширение по определению необходимо многократно проверить вложение. Задача проверки вложения является вычислительно сложной, так как это NP-полная задача, поэтому необходимо уменьшить число таких проверок. Это можно сделать первоначальной проверкой нижеперечисленных лемм [3, 6].

Лемма 1. Если минимальная степень вершины графа G есть $d > 0$, то его минимальное вершинное k -расширение

G^* не содержит вершин степени ниже $d + k$.

Лемма 2. Пусть наибольшая из степеней вершин графа G есть s , и в точности m вершин имеют такую степень, тогда минимальное вершинное k -расширение графа G содержит, по крайней мере, $k + m$ вершин степенью не ниже s .

Лемма 3. Если граф G^* является вершинным k -расширением графа G и в графе G^* на m рёбер больше, чем в графе G , то для каждого набора (v_1, \dots, v_k) из k вершин графа G^* выполняется следующее неравенство: $d(v_1) + \dots + d(v_k) - r \leq m$, где r – количество рёбер в подграфе графа G^* , состоящем из вершин $\{v_1, \dots, v_k\}$.

V. ИСКЛЮЧЕНИЕ ИЗОМОРФНЫХ КОПИЙ

Простым решением является поддержание списка полученных неизоморфных графов. Каждый новый граф проверяется на изоморфизм с уже содержащимися в списке графами. При таком подходе происходит большое количество проверок на изоморфизм. Данный недостаток можно исправить следующим образом: в списке хранятся полные инварианты уже построенных графов, поэтому для каждого графа достаточно один раз вычислить полный инвариант и дальше проверять его на равенство с уже имеющимися инвариантами. Таким инвариантом может быть максимальный матричный код или его аналоги. Подобная схема уменьшает количество решений вычислительно сложных задач, так как проверка на равенство выполняется за полиномиальное время.

Другим способом является применение техник исключения изоморфных копий (isomorphism rejection), хороший обзор которых можно найти в работе [8]. В [9] было предложено использовать метод канонических представителей, а также была предложена модель кода произвольного неориентированного графа $C_G(H)$. Данный код получается выписыванием элементов матрицы M_H ниже главной диагонали по строкам. Выписывание происходит в два этапа: сначала выписываются те элементы M_H^{ij} , где $M_G^{ij} = 1$, после те, для которых $M_G^{ij} = 0$. Максимальный из всех кодов изоморфных графов и соответствующий ему граф будем называть *каноническим относительно G* либо просто *каноническим*. Для канонических кодов справедлива следующая теорема.

Теорема 1. Пусть граф H отличается от графа G на $m > 0$ рёбер, и код графа H $C_G(H) = x_1 x_2 \dots x_k \underline{1} 0 \dots 0$ является каноническим, тогда код $c' = x_1 x_2 \dots x_k \underline{0} 0 \dots 0$, получающийся из $C_G(H)$ заменой самой правой единицы на ноль, также будет каноническим.

Доказательство. Предположим, что утверждение не является верным, и код $c' = x_1 x_2 \dots x_k \underline{0} 0 \dots 0$ не является каноническим. Обозначим граф, соответствующий коду c' , через W . Тогда существует граф W' , изоморфный графу W , код которого $C_G(W')$ является каноническим, и $C_G(W') > C_G(W)$. Обозначим через ϕ изоморфизм графа W' на граф W . Код графа H из условия получается из кода графа W добавлением одной единицы, то есть граф H получается из графа W добавлением одного ребра. Пусть, для определенности, это будет ребро $\{u, v\}$. Построим граф H' , добавив к графу W' ребро $\{\phi(u), \phi(v)\}$. Очевидно, что ϕ будет изоморфизмом графа H на граф H' , но код графа H' будет меньше кода графа

H , а это противоречит тому, что код графа H является каноническим.

Данная теорема означает, что дерево T_G содержит поддереву канонических представителей T_G^c . Это даёт возможность уменьшить количество проверок каноничности путём исключения набора узлов уровня m за счёт проверки каноничности их предка на более ранних уровнях дерева. Такое улучшение метода канонических представителей называется методом Рида-Фараджева.

Метод Рида-Фараджева состоит в уменьшении перебора объектов путём исключения их на раннем этапе.

В некоторых случаях можно определить вектор степеней, который имеют расширения графа. Это можно сделать, например, для циклов. Тогда можно использовать алгоритм 2.

Алгоритм 2. Построение всех минимальных вершинных k -расширений графа G при известном векторе степеней.

1. Строим все неизоморфные реализации заданного вектора степеней.
2. Выбираем среди построенных графов вершинные k -расширения графа G .

Для построения всех неизоморфных реализаций заданного вектора степеней можно использовать специальные генераторы [10, 11]. В частных случаях, когда расширение задаётся регулярным графом, можно использовать генератор genreg [12], а для кубических графов – minibaum [13] или snarkhunter [14].

VI. ОРГАНИЗАЦИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Рассмотрим два способа организации параллельных вычислений.

Первый способ может использоваться при построении МВ- k -Р достаточно большого количества графов, которые можно распределить заранее по процессам, которые будут их обрабатывать. При использовании такого подхода в программе не нужно реализовывать никаких дополнительных действий для синхронизации и распределения данных, достаточно просто назначить входные данные вычислительным узлам, однако при этом у разных процессов будет затрачено различное время, так как графы считаются различное время. Для того, чтобы избавиться от этого недостатка, можно распределять входные данные по процессам динамически, тогда общее время подсчёта уменьшится, так как процессы не будут простаивать, пока все входные данные не распределены.

Второй способ – это распределение графов с m дополнительными рёбрами и k дополнительными вершинами. Пусть необходимо построить МВ- k -Р графа G , тогда мы должны построить все возможные варианты наборов из m дополнительных рёбер для графа $G \cup O_k$: эти наборы можно распределить по процессам. Это можно сделать следующим способом: распределить узлы дерева $T_{G \cup O_k}$ или $T_G^c \cup O_k$ на уровне $x \leq m$ между процессами (дерево выбирается в зависимости от используемого метода). После каждый процесс

проверяет только те графы на уровне m , которые являются потомками распределённых ему узлов.

VII. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

На языке C++ с использованием MPI были реализованы 4 модификации алгоритма 1 в виде отдельных программ.

В первой модификации сначала проверяются леммы 1-3, после происходит проверка графа на расширение, а затем происходит проверка на изоморфизм путём вычисления и сравнения максимального кода. В список заносятся максимальные коды канонических графов.

Вторая модификация аналогична первой, но для исключения изоморфных графов используется проверка на каноничность.

В третьей модификации сначала проверяется выполнение лемм 1-3, затем каноничность, а в конце граф проверяется на расширение по определению.

Четвёртая модификация является обходом в ширину дерева $T_{G \cup O_k}^c$.

Для первой, второй и третьей программ был реализован перебор графов, удовлетворяющих леммам 1 и 3 (лемма 2 проверяется отдельно), а также исключая некоторые неканонические графы. При таком переборе остаётся хотя бы один представитель каждого класса изоморфизма (канонический), поэтому алгоритм остаётся корректным.

VIII. РЕЗУЛЬТАТЫ ВЫЧИСЛЕНИЙ И СРАВНЕНИЕ АЛГОРИТМОВ

В данном разделе представлены результаты вычислительного эксперимента для некоторых графов. Было замечено, что скорость проверки на расширение и каноничность существенно зависит также от кода графа. Удалось найти особый вид кода графа, при котором скорость построения расширения максимальна. Такой код получается упорядочиванием вершин графа так, чтобы вектор количества единиц в каждом ряду был лексикографически максимален, и среди таких графов код, составленный из тех же элементов, был также максимален.

В таблице 1 приведено время работы каждой из программ, которые были затрачены для подсчёта МВ-1-Р 9-вершинных графов и МВ-2-Р 7-вершинных графов, а в таблице 2 – аналогичные результаты для циклов. Также в таблице 2 показано сравнение со временем, затраченным на циклы при использовании алгоритма 2 [15]. Необходимо отметить, что алгоритм 2 показывает существенно лучшие результаты для циклов, однако его можно применять только в тех случаях, когда вектор степеней минимального вершинного расширения известен заранее. Для произвольного графа алгоритм 2 в общем случае применить нельзя.

МВ-2-Р 8-вершинных графов получилось посчитать только четвёртым методом, и на это ушло 28 минут.

Вычисления проводились на кластере Поволжского Регионального Центра Новых Информационных Технологий [16].

БИБЛИОГРАФИЯ

Таблица 1: Время выполнения экспериментов

	Модификация алгоритма			
	1	2	3	4
МВ-1-Р 9-вершинных графов	44ч.20м.	42ч.49м.	81ч.4м.	> 100ч
МВ-2-Р 7-вершинных графов	44м.1с.	48м.19с.	8м.19с.	3с.

На кластере использовалось 40 ядер, и время работы программы было ограничено 100 часами.

Из таблицы видно, что МВ-1-Р лучше всего считать второй программой, МВ-2-Р – четвёртой, а циклы – третьей или алгоритмом 2.

Количество найденных МВ-1-Р расширений графов с количеством вершин от 1 до 9 было занесено в открытую базу данных [17].

Таблица 2: Время вычисления циклов с n вершинами

n	Время вычисления, секунды				
	Модификация алгоритма 1				Алгоритм 2
	1	2	3	4	
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	2	0	0	0	0
11	8	0	0	0	0
12	215	0	5	2	0
13	1141	0	3	1	0
14	44008	3	526	117	0
15	-	4	374	23	0
16	-	85	88307	4919	5
17	-	112	82988	894	2
18	-	3741	-	272007	110
19	-	5513	-	-	30
20	-	180446	-	-	2400
21	-	287765	-	-	222
22	-	-	-	-	23154

IX. ЗАКЛЮЧЕНИЕ

Описанные в данной статье способы применения техники исключения изоморфизма в задаче построения вершинных расширений графа были реализованы в виде отдельных программ. Был проведён вычислительный эксперимент, который показал, что разные модификации алгоритма эффективны в разных ситуациях.

- [1] Hayes J. P. A graph model for fault-tolerant computing system // IEEE Transactions on Computers, 1976. Vol. C-25, No 9. P. 875–884. DOI: 10.1109/TC.1976.1674712
- [2] Harary F., Hayes J. P. Edge fault tolerance in graphs // Networks. 1993. Vol.23. P.135–142. DOI: 10.1002/net.3230230207
- [3] Абросимов М.Б. Графовые модели отказоустойчивости. Саратов : Издательство Саратовского университета, 2012, 192 с.
- [4] Богомолов А.М., Салий В.Н. Алгебраические основы теории дискретных систем. М.: Наука 1997.
- [5] Абросимов М.Б. О сложности векторных задач, связанных с расширениями графов // Матем. заметки. 2010. №5 (88). С.643-650. DOI: 10.4213/mzm8403
- [6] Абросимов М.Б. Минимальные расширения графов // Новые информационные технологии в исследовании дискретных структур. Томск: 2000. С. 59–64.
- [7] Абросимов М.Б. Минимальные расширения 4-,5-,6- и 7-вершинных графов. Саратов гос. ун-т. - Саратов, 2000. – 26с.; Деп. в ВИНТИ 06.09.2000, №2352-B00.
- [8] Brinkmann G. Isomorphism rejection in structure generation programs // Discrete Mathematical Chemistry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 2000. Vol. 51. P. 25–38. DOI: 10.1090/dimacs/051/03
- [9] Абросимов М. Б., Камил И. А. К., Лобов А. А. Построение всех неизоморфных минимальных вершинных расширений графа методом канонических представителей // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2019. Т. 19, вып. 4. С. 479–486. DOI: <https://doi.org/10.18500/1816-9791-2019-19-4-479-486>.
- [10] Grund R. Konstruktion schlichter Graphen mit gegebener Gradpartition // Bayreuther Mathematische Schriften. 1993. Vol.44. P.73–104.
- [11] Сухов С.А. DSR Generator // Свид. о гос. регистрации программы для ЭВМ № 2016610073. Зарегистрирована в Реестре программ для ЭВМ 11 января 2016 г.
- [12] Meringer M. Fast generation of regular graphs and construction of cages // Journal of Graph Theory. 1999. Vol.30. P.137–146.
- [13] Brinkmann G. Fast generation of cubic graphs // Journal of Graph Theory. 1996. Vol.23(2). P.139–149. [https://doi.org/10.1002/\(SICI\)1097-0118\(199610\)23:2<139::AID-JGT5>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0118(199610)23:2<139::AID-JGT5>3.0.CO;2-U)
- [14] Brinkmann G., Goedgebeur J., McKay B. D. Generation of cubic graphs // Discrete Mathematics and Theoretical Computer Science. 2011. Vol.13(2). P.69–80.
- [15] Абросимов М.Б., Бринкман Г., Сухов С.А. О количестве минимальных 1-расширений циклов с числом вершин до 26 и 28 // Компьютерные науки и информационные технологии : Материалы Междунар. науч. конф. – Саратов : Издат. центр «Наука», 2016. С. 9-11.
- [16] Официальный сайт Поволжского Регионального Центра Новых Информационных Технологий // URL:<http://prcnit.sgu.ru>
- [17] Энциклопедия: Мир графов // URL:<http://graphworld.ru>

Construction of minimum vertex extensions of a graph by the Read-Faradzhev method

I.A.K. Kamil, M.B. Abrosimov, A.A. Lobov

Annotation – In 1976 John Hayes proposed a graph-based model to investigate node fault tolerance of discrete systems. A graph G^* is vertex k -extension of a graph G if after removing any k vertices from G^* result graph contains G . A vertex k -extension of the graph G is called minimal if it has the least number of vertices and edges among all vertex k -extensions of G . Later together with Frank Harary, the model was extended to failures of connections between elements. The problem of verifying a vertex k -extension of a graph is NP-complete. For some types of graphs, it was possible to find an analytical solution to the problem of constructing minimal vertex k -extensions: paths, cycles, stars, etc. In general, a graph can have many nonisomorphic minimal vertex k -extensions. The exhaustive algorithm for constructing all nonisomorphic minimal vertex k -extensions of a given graph can be used only for graphs with a small number of vertices. In some cases, it is possible to find the degree vector of the minimum vertex k -extension, which allows to search much faster. In this paper, we propose an algorithm for constructing all nonisomorphic minimal vertex k -extensions of a given graph with isomorphism rejection by the method of canonical representatives in the Read-Faradzhev form. The implementation of this algorithm in 4 modifications is considered and the results of their work are analyzed.

Key words — fault tolerance, graph extension, isomorphism rejection, canonical form.

REFERENCES

- [1] Hayes J. P. A graph model for fault-tolerant computing system // IEEE Transactions on Computers, 1976. Vol. C-25, No 9. P. 875–884. DOI: 10.1109/TC.1976.1674712
- [2] Harary F., Hayes J. P. Edge fault tolerance in graphs // Networks. 1993. Vol.23. P.135–142. DOI: 10.1002/net.3230230207
- [3] Abrosimov M.B. Grafovye modeli otkazoustoichivosti. Saratov : Izdatel'stvo Saratovskogo universiteta, 2012, 192 p.
- [4] Bogomolov A.M., Salii V.N. Algebraicheskie osnovy teorii diskretnykh sistem. M.: Nauka 1997.
- [5] Abrosimov M.B. On the complexity of some problems related to graph extensions. Math. Notes, 2010. Vol. 88, no. 5, pp. 619–625. DOI: 10.1134/S0001434610110015
- [6] Abrosimov M.B. Minimal'nye rasshireniia grafov // Noveye informatsionnye tekhnologii v issledovanii diskretnykh struktur. Tomsk: 2000. pp. 59–64.
- [7] Abrosimov M.B. Minimal'nye rasshireniia 4-,5-,6- i 7-vershinnykh grafov. Saratov gos. un-t. - Saratov, 2000. – 26 p.; Dep. v VINITI 06.09.2000, №2352-V00.
- [8] Brinkmann G. Isomorphism rejection in structure generation programs // Discrete Mathematical Chemistry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 2000. Vol. 51. P. 25–38. DOI: 10.1090/dimacs/051/03
- [9] Abrosimov M.B., Kamil I.A.K., Lobov A.A. Postroenie vseh neizomorfnykh minimal'nykh vershinnykh rasshirenii grafa metodom kanonicheskikh predstavitelei // Izv. Sarat. un-ta. Nov. ser. Ser. Matematika. Mekhanika. Informatika. 2019. Vol. 19, no. 4. pp. 479–486. DOI: <https://doi.org/10.18500/1816-9791-2019-19-4-479-486>.
- [10] Grund R. Konstruktion schlichter Graphen mit gegebener Gradpartition // Bayreuther Mathematische Schriften. 1993. Vol.44. P.73–104.
- [11] Sukhov S.A. DSR Generator // Svid. o gos. registratsii programmy dlia EVM № 2016610073. Zaregistrovana v Reestre programm dlia EVM 11.01.2016.
- [12] Meringer M. Fast generation of regular graphs and construction of cages // Journal of Graph Theory. 1999. Vol.30. P.137–146.
- [13] Brinkmann G. Fast generation of cubic graphs // Journal of Graph Theory. 1996. Vol.23(2). P.139–149. [https://doi.org/10.1002/\(SICI\)1097-0118\(199610\)23:2<139::AID-JGT5>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0118(199610)23:2<139::AID-JGT5>3.0.CO;2-U)
- [14] Brinkmann G., Goedgebeur J., McKay B. D. Generation of cubic graphs // Discrete Mathematics and Theoretical Computer Science. 2011. Vol.13(2). P.69–80.
- [15] Abrosimov M.B., Brinkmann G., Sukhov S.A. O kolichestve minimal'nykh 1-rasshirenii tsiklov s chislom vershin do 26 i 28 // Komp'iuternye nauki i informatsionnye tekhnologii : Materialy Mezhdunar. nauch. kof. – Saratov : Izdat. tsentr «Nauka», 2016. pp. 9–11.
- [16] Volga Regional Center for New Information Technologies. Available at: <http://prcnit.sgu.ru>
- [17] Encyclopedia: World of Graphs. Available at: <http://graphworld.ru>