



ИНФОРМАТИКА

УДК 519.17

Построение минимальных рёберных расширений графа без проверки на изоморфизм

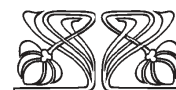
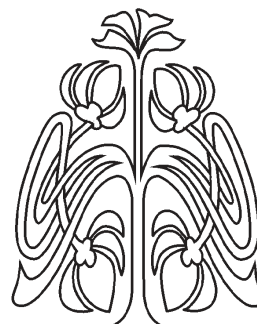
М. Б. Абросимов, Х. Х. К. Судани, А. А. Лобов

Абросимов Михаил Борисович, доктор физико-математических наук, заведующий кафедрой теоретических основ компьютерной безопасности и криптографии, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, Россия, 410012, г. Саратов, ул. Астраханская, д. 83, mic@rambler.ru

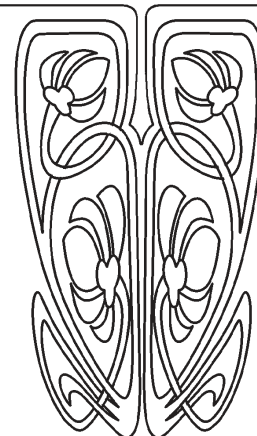
Судани Хайдер Хуссейн Карим, Министерство науки и технологий Ирака, Багдад, Ирак; аспирант кафедры теоретических основ компьютерной безопасности и криптографии, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, Россия, 410012, г. Саратов, ул. Астраханская, д. 83, hayder.1977@mail.ru

Лобов Александр Андреевич, аспирант кафедры теоретических основ компьютерной безопасности и криптографии, Саратовский национальный исследовательский государственный университет имени Н. Г. Чернышевского, Россия, 410012, г. Саратов, ул. Астраханская, д. 83, aisaneka@mail.ru

В 1993 г. Frank Harary и John P. Hayes предложили основанную на графах модель для исследования отказов связей элементов дискретных систем. Технической системе сопоставляется граф. Элементам системы соответствуют вершины графа, а связям между элементами — рёбра или дуги графа. Под отказом связи между элементами системы понимается удаление из графа системы соответствующего ребра (или дуги). Формализацией отказоустойчивой реализации системы является расширение графа. Граф G^* называется рёберным k -расширением графа G , если после удаления любых k рёбер из графа G^* граф G вкладывается в получившийся граф. Рёберное k -расширение n -вершинного графа G называется минимальным, если оно имеет n вершин и минимальное число рёбер среди всех рёберных k -расширений графа G с n вершинами. В работе предлагается алгоритм построения всех неизоморфных минимальных рёберных k -расширений заданного графа без проверки на изоморфизм методами канонических представителей и Рида – Фараджева.



НАУЧНЫЙ
ОТДЕЛ





Ключевые слова: отказоустойчивость, отказы связей, расширение графа, изоморфизм, канонический код, метод канонических представителей, метод Риды – Фараджева.

Поступила в редакцию: 20.10.2019 / Принята: 02.12.2019 / Опубликовано: 02.03.2020

Статья опубликована на условиях лицензии Creative Commons Attribution License (CC-BY 4.0)

DOI: <https://doi.org/10.18500/1816-9791-2020-20-1-105-115>

ВВЕДЕНИЕ

Под полной отказоустойчивостью технической системы понимается возможность системы продолжать работать при единичном или множественных отказах без существенной потери функциональных свойств. Отказы могут возникать в элементах системы или в связях между элементами. В 1976 г. John P. Hayes [1] предложил теоретическую модель для исследования полной отказоустойчивости дискретных систем с отказами элементов на языке теории графов. Позднее совместно с Frank Nagay модель была распространена на отказы связей между элементами [2]. Технической системе сопоставляется граф. Элементом системы соответствуют вершины графа, а связям между элементами — рёбра или дуги графа. Соответственно, граф системы будет неориентированный или ориентированный. В данной работе мы будем рассматривать неориентированные графы, хотя предлагаемые методы могут быть легко перенесены и на случай ориентированных графов. Под отказом связи между двумя элементами системы понимается удаление из графа системы соответствующего ребра. Формализацией отказоустойчивой реализации системы является расширение графа [3]. Основные необходимые определения по теории графов, которые используются далее, можно найти в работах [3, 4].

Граф $G^* = (V^*, \alpha^*)$ называется рёберным k -расширением (k – натуральное) графа $G = (V, \alpha)$, если граф G вкладывается в каждый граф, получающийся из графа G^* удалением любых его k рёбер. Граф $G^* = (V^*, \alpha^*)$ с n вершинами называется минимальным рёберным k -расширением (MP- k -P) n -вершинного графа $G = (V, \alpha)$, если граф G^* является рёберным k -расширением графа G и имеет минимально возможное число рёбер среди всех n -вершинных рёберных k -расширений графа G .

В работе [1] были предложены схемы построения MP-1-P для цепей и циклов. В работе [5] было доказано, что задача построения рёберных k -расширений является вычислительно сложной. Граф в общем случае может иметь несколько неизоморфных MP- k -P. В работе [6] был предложен алгоритм построения всех неизоморфных минимальных вершинных k -расширений методом канонических представителей. В этой статье будет рассмотрена задача построения всех неизоморфных минимальных рёберных k -расширений заданного графа. Будет рассмотрен переборный алгоритм построения всех MP- k -P, а также алгоритмы построения всех неизоморфных MP- k -P без проверки на изоморфизм. Напомним, что два графа, $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$, называются *изоморфными*, если можно установить взаимно однозначное соответствие $\phi : V_1 \rightarrow V_2$, сохраняющее отношение смежности: $(u, v) \in \alpha_1 \leftrightarrow (\phi(u), \phi(v)) \in \alpha_2, \forall u, v \in V_1$. Инвариантом графа G называется его характеристика, одинаковая для всех изоморфных G графов. Примеры простейших инвариантов — порядок графа n (число вершин) или размер графа t (число рёбер). Инвариант называется полным, если его значения различаются для неизоморфных графов. Полный инвариант может использоваться для решения задачи изоморфизма графов, поэтому его вычисление не может быть эффективнее решения задачи об изоморфизме. На данный момент неизвестны эффективно вычисляемые полные инварианты графа. Один из полных инвариантов будет далее рассмотрен в работе.



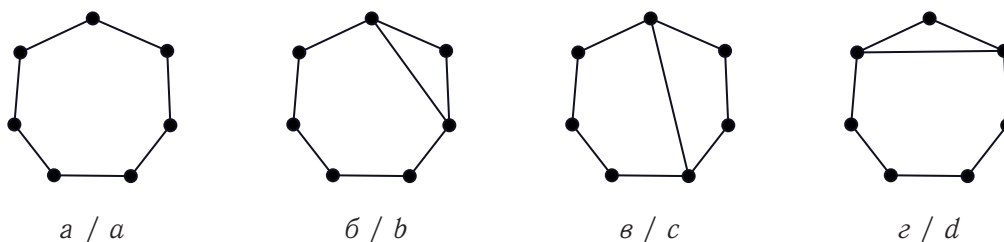
1. ПОСТРОЕНИЕ РЁБЕРНЫХ РАСШИРЕНИЙ ГРАФА

Для построения всех неизоморфных МР- k -Р в работе [3] был предложен общий переборный алгоритм.

Алгоритм 1. Построение всех МР- k -Р графа G .

1. $m := 0$.
2. Строим все графы, получающиеся из графа G добавлением m дополнительных рёбер.
3. Если на шаге 2 не построено ни одного графа, то завершить алгоритм, расширений не существует.
4. Выбираем среди построенных на шаге 2 графов рёберные k -расширения графа G .
5. Если на шаге 4 не было найдено графов, то присваиваем $m := m+1$ и переходим на шаг 2.
6. Среди графов, выбранных на шаге 4, оставляем по одному представителю от классов изоморфных графов. Полученные графы являются МР- k -Р графа G .

На практике шаги 2–4 могут быть совмещены. Рассмотрим идею работы алгоритма на примере построения минимального рёберного 1-расширения 7-вершинного цикла C_7 , изображенного на рисунке, а.



Цикл C_7 и графы, отличающиеся от него на 1 ребро
The cycle C_7 and graphs differing by 1 edge from it

На шаге 2 алгоритма необходимо добавить к циклу C_7 m рёбер, где $m = 1, 2, \dots$. Добавить $m = 1$ ребро к графу C_7 означает заменить один из нулей, расположенных в матрице смежности выше или ниже главной диагонали, на единицу. При непосредственном переборе всех вариантов это можно сделать C_{14}^m , т. е. 14 способами. Легко заметить, что на самом деле получится только 2 неизоморфных графа (на рис. — это графы б и г), все остальные будут изоморфны одному из них (на рис. графы б и в изоморфны). При $m = 2$ число способов будет 91, а неизоморфных графов — 10, при $m = 3$ соответственно 364 и 30. Далее каждый получившийся граф проверяется на шаге 4 — является ли он рёберным 1-расширением заданного графа. В общем случае для этого требуется проверить, что исходный граф допускает вложение в каждый граф, получающийся из очередного кандидата удалением одного ребра. Если хотя бы для одного графа не существует вложения графа C_7 , то исходный граф не является рёберным 1-расширением. Для того чтобы проверить, что граф не является МР- k -Р, можно использовать следующую лемму [3, 7].

Лемма 1. Если минимальная степень вершины графа G есть $d > 0$, то его МР- k -Р G^* не содержит вершин степени ниже $d + k$.

С помощью этой леммы для цикла C_7 можно установить, что в МР-1-Р не может быть вершин степени ниже 3. Как следствие, число дополнительных рёбер будет не меньше 4. В работе [1] доказывается, что МР-1-Р цикла имеет вектор степеней $(4, 3, \dots, 3)$ или $(3, \dots, 3)$ в зависимости от чётности числа вершин цикла. Число



вариантов при $m = 4$ составит 1001, из которых неизоморфных будет только 58, а неизоморфных графов, удовлетворяющих лемме 1 (т. е. графов с вектором степеней $(4, 3^6)$), всего 4. У цикла C_7 всего 2 неизоморфных МР-1-Р. Алгоритм 1 можно использовать для построения МР- k -Р графов с небольшим числом вершин [8].

Наметим основной путь для оптимизации алгоритма 1: на шаге 3 нужно оставлять как можно меньше кандидатов для последующей проверки на расширение на шаге 4. Далее в статье будет рассмотрено решение этой задачи. Для того чтобы снизить количество поисков вложений, при проверке графа на расширение можно использовать следующую лемму. Будем говорить, что у ребра $e = \{u, v\}$ графа H нет прообраза в графе G , если существует вложение ϕ графа G в граф H такое, что в графе G нет ребра между прообразами вершин u и v , т. е. между $\phi^{-1}(u)$ и $\phi^{-1}(v)$.

Лемма 2. Если ϕ — вложение графа G в граф H и рёбра e_1, e_2, \dots, e_m графа H не имеют прообразов, то ϕ будет являться вложением графа G в каждый граф, полученный удалением любых k рёбер графа H из данного набора, где $0 \leq k \leq m$.

У алгоритма 1 можно выделить несколько недостатков, связанных с избыточным перебором. Один из них состоит в следующем: если на шаге 2 могут появляться изоморфные графы, то необходимо хранить все построенные расширения, чтобы на шаге 6 исключить изоморфные копии. Если на шаге 2 строить только неизоморфные графы, то необходимость хранения всех построенных расширений исчезнет, а также уменьшится число графов, которые нужно будет проверять на расширение. Методы генерации без проверки на изоморфизм достаточно интенсивно исследуются для построения графов различного вида. Хороший обзор можно найти в работе [9]. Далее в статье будет описан метод канонических представителей для поставленной задачи построения МР- k -Р и рассмотрены его оптимизационные варианты.

2. МЕТОД КАНОНИЧЕСКИХ ПРЕДСТАВИТЕЛЕЙ

Идея метода канонических представителей состоит в том, что выбирается некоторое правило, по которому из всех изоморфных графов один объявляется каноническим. В общем виде метод можно описать следующим образом [9]:

- 1) определяется способ кодирования графов;
- 2) среди всех кодов изоморфных графов выбирается канонический код (представитель);
- 3) порождаются все возможные коды графов;
- 4) порождённый граф принимается, если его код канонический, в противном случае исключается.

С использованием метода канонических представителей алгоритм 1 можно преобразовать в алгоритм 2.

Алгоритм 2. Построение всех МР- k -Р графа G без проверки на изоморфизм.

1. $m := 0$.
2. $m := m + 1$.
3. Строим все неизоморфные графы, получающиеся из графа G добавлением m рёбер.
4. Если на шаге 3 не построено ни одного графа, то завершить алгоритм, расширений не существует.
5. Выбираем среди построенных графов рёберные k -расширения графа G .
6. Если на шаге 5 не было найдено графов, то переходим на шаг 2.
7. Полученные на шаге 5 графы являются МР- k -Р графа G .



В алгоритме 2 при использовании метода канонических представителей не нужно проверять графы на изоморфизм. Также изменён порядок проверки: сначала находим все неизоморфные графы, а после проверяем, являются ли они рёберными k -расширениями или нет. Для правильности предложенного алгоритма необходимо, чтобы все канонические представители каждого класса изоморфных графов были построены на шаге 3. Для использования метода канонических представителей самым важным является выбор метода кодирования графа. Предлагается взять код, основанный на матрице смежности графа. Напомним, что для простых неориентированных графов матрица смежности симметрична относительно главной диагонали, а на главной диагонали расположены нули. Таким образом, матрица смежности полностью определяется своей частью, расположенной выше или ниже главной диагонали.

Определим способ кодирования графа. Рассмотрим два n -вершинных графа, $G = (Z_n, \alpha)$ и $H = (Z_n, \beta)$, где $Z_n = \{0, \dots, n-1\}$ и $\alpha, \beta \subseteq Z_n \times Z_n$. Для каждого графа составим матрицы смежности. Через G обозначим граф, для которого требуется найти МР- k -Р. Через H обозначим граф, для которого будем строить код. Число вершин в графах G и H равно n . Определим код $C_G(H)$ графа H следующим образом: будем дважды просматривать элементы матрицы смежности графа G , находящиеся ниже главной диагонали, по строкам слева направо и сверху вниз, выписывая соответствующие элементы матрицы смежности графа H по следующим правилам:

1) в первый раз выписываем элемент матрицы смежности графа H , если в матрице смежности графа G стоит 1;

2) во второй раз выписываем элемент матрицы смежности графа H , если в матрице смежности графа G стоит 0.

По построению очевидно, что код n -вершинного графа с m рёбрами будет иметь длину C_n^2 и содержать ровно m единиц. Будем называть граф H каноническим относительно G (либо просто каноническим) и его код каноническим, если среди всех графов, изоморфных H , код графа H является лексикографически наибольшим: $\{\forall R \cong H, R \neq H : C_G(R) < C_G(H)\}$. Справедливы следующие утверждения:

Утверждение 1. Если G является частью графа H , то $C_G(G) \leq C_G(H)$, иначе $C_G(G) > C_G(H)$.

Утверждение 2. Граф G вкладывается в граф H тогда и только тогда, когда существует $W \cong H$, такой, что $C_G(W) \geq C_G(G)$. Это означает, что если граф G вкладывается в граф H , то существует изоморфный ему канонический граф W : $C_G(W) \geq C_G(G)$.

Это значит, что канонического представителя класса изоморфизма каждого графа, в который вкладывается граф G , можно получить добавлением рёбер в граф G . Таким образом, алгоритм 2 является корректным.

Алгоритм 3. Построение всех неизоморфных графов, отличающихся от заданного графа G на m дополнительных рёбер, методом канонических представителей.

1. Начальным выбирается код $c = 1^p 0^{C_n^2 - p}$, список $R = \emptyset$.

2. Перебираем все варианты размещения m единиц в хвосте из 0 кода c . Если очередной получившийся код c' является каноническим, то добавляем c' в R .

Обычно при реализации метода канонических представителей нет необходимости перебирать всех возможных представителей. Если на некотором шаге известно, что последующие итерации не могут привести к каноническому представителю, то



можно произвести усечение перебора. Такая оптимизация называется методом Рида – Фараджера [9]. В нашем примере построение графов можно рассматривать как перебор кодов, которые начинаются с кода графа G и отличаются от него добавлением m единиц в тех позициях, где стоят 0. Можно не рассматривать те графы, которые заведомо не могут быть расширениями графа G , в том числе те, которые не удовлетворяют леммам. Докажем вспомогательное утверждение.

Теорема 1. Пусть граф H отличается от графа G на $m > 0$ рёбер и код графа H $C_G(H) = x_1x_2\dots x_k\underline{1}0\dots 0$ является каноническим. Тогда код $c' = x_1x_2\dots x_k\underline{0}0\dots 0$, получающийся из $C_G(H)$ заменой самой правой единицы на ноль, также будет каноническим.

Доказательство. Предположим, что утверждение неверно и код

$$c' = x_1x_2\dots x_k\underline{0}0\dots 0$$

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, т. е. 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. \square

Теорема 1 означает, что если на некотором шаге был получен код c , который не является каноническим, то нет необходимости перебирать коды, которые отличаются от полученного добавлением единиц в позициях, следующих за последней единицей кода c . Все эти коды по теореме 1 не будут каноническими. Это является основой для использования метода Рида – Фараджера.

Зафиксируем n -вершинный граф G , относительно которого и будем строить код графов. Построим дерево T_n всех n -вершинных графов следующим образом: пусть код графа равен $a_0a_1\dots a_k\underline{1}0\dots 0$, тогда его родительским узлом будет являться граф с кодом $a_0a_1\dots a_k\underline{0}0\dots 0$, т. е. код графа родительского узла получается заменой крайней правой 1 на 0. В таком дереве код родительского элемента всегда меньше, чем код любого его дочернего элемента.

По теореме 1 получим, что все n -вершинные канонические графы также образуют дерево T_n^C , которое будет являться поддеревом T_n . Из утверждения 1 получим, что поддерево T_G дерева T_n с корнем G — это все графы, которые могут быть получены из G добавлением рёбер, т. е. те графы, которые рассматриваются в алгоритме 1. По утверждению 2 и теореме 1, дерево T_G содержит поддерево T_G^C , которое состоит из всех канонических представителей графов, в которые граф G вкладывается. В алгоритме 4 представлен способ получения дочерних элементов произвольного узла H дерева T_G^C . Обозначим через $l(c)$ длину кода c , а через $c[i]$ — элемент кода в позиции i .

Алгоритм 4. Получение дочерних элементов узла H дерева T_G^C .

1. $c := C_G(H)$, $W = \emptyset$.
2. Вычислить i — номер крайней правой единицы в коде c .
3. $i := i + 1$.
4. Если $i = l(c)$, то результат W , завершить алгоритм.
5. $c[i] := 1$.



6. Если c — канонический код, то добавить соответствующий коду c граф в W .
7. $c[i] := 0$.
8. Перейти на шаг 3.

Далее с использованием алгоритма 4 можно реализовать метод Рида – Фараджева для построения МР- k -Р заданного графа.

Алгоритм 5. Построение всех МР- k -Р графа G обходом дерева.

1. $m := 0$, коды графов строятся на основе графа G .
2. $m := m + 1$.
3. Составить S — множество всех элементов дерева T_G^C на расстоянии m от G .
4. Убрать из S все графы, не являющиеся k -расширениями G по условию леммы 1.
5. Убрать из S все графы, не являющиеся k -расширениями G на основе проверки по определению.
6. Если в S остались графы, то они являются МР- k -Р, завершить алгоритм, иначе перейти на шаг 2.

В алгоритме 5 не сказано, каким именно методом необходимо составлять множество элементов дерева. Для этой цели можно использовать как обход в глубину, так и обход в ширину. У каждого из методов есть свои достоинства и недостатки. Если каждый раз использовать обход в глубину, начиная с корня, то некоторые узлы будут проверяться на каноничность несколько раз, что увеличит количество проверок каноничности. Если использовать обход в ширину, то необходимо хранить последний построенный уровень дерева T_G^C целиком, что увеличит расходы памяти до $O(2^{\frac{n(n-1)}{2}})$, но устранил лишние проверки. Для реализации алгоритма в программе был выбран обход в ширину.

Возможен также другой вариант использования метода Рида – Фараджева. Разместим проверку выполнения леммы 1 до проверки каноничности графа. Проверка леммы 1 выполняется за полиномиальное время — быстрее, чем проверка каноничности и проверка на расширение по определению. Данный метод описан в алгоритме 6.

Алгоритм 6. Построение всех МР- k -Р графа G .

1. $m := 0$, коды графов строятся на основе графа G .
2. $m := m + 1$.
3. Построить множество графов S , полученных из G добавлением m рёбер.
4. Удалить из S все графы, которые не являются рёберными k -расширениями графа G исходя из леммы 1.
5. Удалить из S все графы, которые не являются каноническими.
6. Удалить из S все графы, которые не являются расширениями графа G .
7. Если S пусто, то перейти на шаг 2, иначе S — множество всех неизоморфных МР- k -Р графа G .

Шаги 3–6 алгоритма 6 можно объединить и не хранить в памяти все графы, для чего необходимо иметь возможность их перечислять. При большом количестве вершин и малом количестве рёбер в графе G перебор всех графов осуществляется долго. Для ускорения можно осуществлять перебор с отсечением графов, которые не являются расширениями по лемме 1, а также заведомо не являются каноническими. Для этого составляется множество пар следующего вида: $(t, \{e_j^1, \dots, e_j^p\})$, это означает, что при переборе необходимо добавить как минимум t рёбер из множества $\{e_j^1, \dots, e_j^p\}$. Пусть требуется найти МР- k -Р n -вершинного графа G , тогда множество пар будет составляться так.

1. Если в G минимальная степень $d > 0$, то для каждой вершины u графа H со



степенью $d(u) < d + k$ составляются следующие пары:

$$(w = d + k \checkmark d(u), W = u, v_1^u, \dots, u, v_{n+k-d(v)-1}^u),$$

где v_i^u — не смежные с v вершины. Данные пары нужны для отсечения всех графов, не удовлетворяющих лемме 1.

2. Пара $(1, W)$, где W — множество тех рёбер, при добавлении одного из которых в H получается канонический граф. Данная пара отсекает некоторое количество неканонических графов по условию из теоремы 1.

Отметим, что данный перебор может использоваться и в алгоритме 1. Так как канонический представитель каждого класса не может быть отсечён при переборе на основании выбора элемента из второй группы, то для каждого класса изоморфизма, представители которого будут являться расширениями, при переборе будет присутствовать хотя бы один элемент из каждой группы изоморфных графов.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ, ОЦЕНКА ЭФФЕКТИВНОСТИ И РЕЗУЛЬТАТЫ

На языке C++ с использованием MPI были реализованы алгоритмы 1, 5 и 6. На вход программы графы подавались в формате graph6, который представляет собой закодированную специальным образом часть матрицы смежности графа, расположенную выше главной диагонали [10]. Этот формат является основным в пакете nauty [11], который использовался для генерации графов. Было замечено, что результаты существенно зависят от нумерации вершин графа. Для графа с лексикографически наибольшим значением кода graph6 среди всех изоморфных ему графов время подсчёта MP- k -P обычно существенно меньше, чем у изоморфного ему графа, сгенерированного программой geng из пакета nauty. Так, например, при использовании алгоритма 6 для 12-вершинного цикла время построения MP-1-P отличалось примерно в 228 раз, а для 10-вершинного — примерно в 30 раз. Время подсчёта MP-1-P 9-вершинных графов алгоритмом 6 уменьшилось приблизительно в 72 раза. Этот вопрос требует дополнительного исследования и выходит за рамки данной работы. Во всех описываемых далее экспериментах использовались коды графов, сгенерированные программой geng в режиме по умолчанию.

На практике в алгоритмах 2, 5 и 6 не нужно хранить все графы, полученные добавлением k рёбер, и канонические графы в памяти. Достаточно иметь возможность перечислять графы и выбирать из них канонические. В таком случае достаточно $O(n^2)$ памяти для реализации данного алгоритма (хранение матрицы смежности графа). Вычислительная сложность каждого из описанных алгоритмов экспоненциальна.

В табл. 1 и 2 показано сравнение времени вычисления MP-1-P и MP-2-P графов с заданным количеством вершин. Для построения всех графов с заданным числом вершин использовался генератор geng из пакета nauty [11]. Вычисления выполнялись на кластере Поволжского регионального центра новых информационных технологий (ПРЦ НИТ) [12]. В вычислениях было задействовано 40 ядер. Полученные в рамках вычислительного эксперимента результаты были добавлены в энциклопедию «Мир графов» [13]. В реализациях были учтены приведённые ранее леммы.



Таблица 1 / Table 1

Время подсчёта МР-1-Р всех графов с заданным числом вершин n
 Counting time МР-1-Р of all graphs with a given number of vertices n

n	Всего графов Total Counts	Алгоритм 1 Algorithm 1	Алгоритм 5 Algorithm 5	Алгоритм 6 Algorithm 6
1–7	1249	< 1 с.	< 1 с.	< 1 с.
8	12 346	13 с.	6 с.	6 с.
9	274 668	1 ч. 7 м.	1 ч. 3 м.	59 м.

Таблица 2 / Table 2

Время подсчёта МР-2-Р всех графов с заданным числом вершин n
 Counting time МР-2-Р of all graphs with a given number of vertices n

n	Всего графов Total Counts	Алгоритм 1 Algorithm 1	Алгоритм 5 Algorithm 5	Алгоритм 6 Algorithm 6
1–7	1249	< 1 с.	< 1 с.	< 1 с.
8	12 346	2 м. 38 с.	17 с.	16 с.
9	274 668	1 д. 12 ч.	3 ч. 38 м.	3 ч. 9 м.

Можно отметить прирост скорости построения МР- k -Р для алгоритма 2 по сравнению с алгоритмом 1. Заметим, что лемма 2 даёт небольшую прибавку в скорости к алгоритмам 5 и 6 и достаточно существенную к алгоритму 1.

Для оценки вычислительной сложности каждой из частей программы были проведены замеры времени выполнения некоторых особо важных её частей. Основное время в среднем ожидаемо занимает проверка графа на расширение или каноничность. Проверка каноничности происходит быстрее, чем проверка на расширение, поэтому алгоритм 1, в котором основным действием является проверка на расширение, уступает остальным алгоритмам. Особенно это заметно при подсчёте МР-2-Р, так как задача проверки каноничности не усложняется, а в задаче проверки графа на расширение увеличивается количество поисков вложения. Следует отметить, что использование леммы 2 существенно улучшает время построения МР-2-Р графа по алгоритму 1. Например, время построения МР-2-Р всех 9-вершинных графов сокращается более чем вдвое: с 3 дней 8 часов до 1 дня 12 часов. Для остальных алгоритмов лемма 2 не даёт такого результата.

ЗАКЛЮЧЕНИЕ

В работе были описаны алгоритмы построения всех минимальных рёберных k -расширений для заданного графа с помощью методов канонических представителей и Рида – Фараджева. Алгоритмы были реализованы на языке С++ с использованием технологии МРІ и протестированы на кластере Поволжского регионального центра новых информационных технологий. Сравнения данных алгоритмов с обычным алгоритмом и друг с другом показали преимущества и недостатки каждого из них.

Библиографический список

1. Hayes J. P. A graph model for fault-tolerant computing system // IEEE Trans. Computers. 1976. Vol. C-25, № 9. P. 875–884. DOI: <https://doi.org/10.1109/TC.1976.1674712>
2. Harary F., Hayes J. P. Edge fault tolerance in graphs // Networks. 1993. Vol. 23. P. 135–142. DOI: <https://doi.org/10.1002/net.3230230207>



3. *Абросимов М. Б.* Графовые модели отказоустойчивости. Саратов : Изд-во Саратов. ун-та, 2012. 192 с.
4. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М. : Наука, 1997. 368 с.
5. *Абросимов М. Б.* О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. Т. 88, вып. 5. С. 643–650. DOI: <https://doi.org/10.4213/mzm8403>
6. *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>
7. *Абросимов М. Б.* Минимальные расширения графов // Новые информационные технологии в исследовании дискретных структур. Томск : ИД Том. гос. ун-та, 2000. С. 59–64.
8. *Абросимов М. Б.* Минимальные расширения 4-, 5-, 6- и 7-вершинных графов. Саратов. гос. ун-т. Саратов, 2000. 26 с. ; Деп. в ВИНТИ 06.09.2000, № 2352-В00.
9. *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: <https://doi.org/10.1090/dimacs/051/03>
10. *McKay B. D.* Graph formats. URL: <http://users.cecs.anu.edu.au/bdm/data/formats.html> (дата обращения: 01.05.2019).
11. *McKay B. D., Piperno A.* Practical Graph Isomorphism, II // Journal of Symbolic Computation. 2014. Vol. 60. P. 94–112. DOI: <https://doi.org/10.1016/j.jsc.2013.09.003>
12. Поволжский региональный центр новых информационных технологий : [сайт]. URL: <http://prcnit.sgu.ru> (дата обращения: 01.05.2019).
13. Мир графов. URL: <http://graphworld.ru> (дата обращения: 01.05.2019).

Образец для цитирования:

Абросимов М. Б., Судани Х. Х. К., Лобов А. А. Построение минимальных рёберных расширений графа без проверки на изоморфизм // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2020. Т. 20, вып. 1. С. 105–115. DOI: <https://doi.org/10.18500/1816-9791-2020-20-1-105-115>

Construction of All Minimal Edge Extensions of the Graph with Isomorphism Rejection

M. B. Abrosimov, H. H. K. Sudani, A. A. Lobov

Mikhail B. Abrosimov, <https://orcid.org/0000-0002-4473-8790>, Saratov State University, 83 Astrakhanskaya St., Saratov 410012, Russia, mic@rambler.ru

Hayder H. K. Sudani, <https://orcid.org/0000-0002-9502-2952>, Ministry of Science and Technology of Iraq, Baghdad, Iraq; Saratov State University, 83 Astrakhanskaya St., Saratov 410012, Russia, hayder.1977@mail.ru

Alexander A. Lobov, <https://orcid.org/0000-0003-3422-3811>, Saratov State University, 83 Astrakhanskaya St., Saratov 410012, Russia, aisaneka@mail.ru

In 1993 Frank Harary and John P. Hayes proposed a graph model for investigating edge fault tolerance of discrete systems. The technical system is mapped to a graph. The elements of the system correspond to the vertices of the graph, and links between the elements correspond to edges or arcs of the graph. Failure of a system element refers to the removal of the corresponding vertex from the system graph along with all its edges. The formalization of a fault-tolerant system



implementation is the extension of the graph. The graph G^* is called the edge k -extension of the graph G if, after removing any k edges from the graph G^* result graph contains the graph G . The edge k -extension of a graph G is called minimal if it has the least number of vertices and edges among all edge k -extensions of a graph G . An algorithm for constructing all nonisomorphic minimal edge k -extensions of a given graph using methods of canonical representatives and Read–Faradjev are proposed.

Keywords: fault tolerance, edge fault tolerance, graph extension, isomorphism rejection, canonical form, method of generating canonical representatives, Read-Faradjev-type orderly algorithm.

Received: 20.10.2019 / Accepted: 02.12.2019 / Published: 02.03.2020

This is an open access article distributed under the terms of Creative Commons Attribution License (CC-BY 4.0)

References

1. Hayes J. P. A Graph Model for Fault-tolerant Computing System. *IEEE Trans. Computers*, 1976, vol. C-25, no. 9, pp. 875–884. DOI: <https://doi.org/10.1109/TC.1976.1674712>
2. Harary F., Hayes J. P. Edge Fault Tolerance in Graphs. *Networks*, 1993, vol. 23, pp. 135–142. DOI: <https://doi.org/10.1002/net.3230230207>
3. Abrosimov M. B. *Grafovyye modeli otkazoustoichivosti* [Fault tolerance graph models]. Saratov, Izdatel'stvo Saratovskogo universiteta, 2012. 192 p. (in Russian).
4. Bogomolov A. M., Salii V. N. *Algebraicheskie osnovy teorii diskretnykh sistem* [Algebraic foundations of the theory of discrete systems]. Moscow, Nauka, 1997. 384 p. (in Russian).
5. Abrosimov M. B. On the Complexity of Some Problems Related to Graph Extensions. *Math. Notes*, 2010, vol. 88, iss. 5, pp. 619–625. DOI: <https://doi.org/10.1134/S0001434610110015>
6. Abrosimov M. B., Kamil I. A. K., Lobov A. A. Construction of All Nonisomorphic Minimal Vertex Extensions of the Graph by the Method of Canonical Representatives. *Izv. Saratov Univ. (N. S.), Ser. Math. Mech. Inform.*, 2019, vol. 19, iss. 4, pp. 479–486 (in Russian). DOI: <https://doi.org/10.18500/1816-9791-2019-19-4-479-486>
7. Abrosimov M. B. Minimal graph extensions. In: *Novyye informatsionnyye tekhnologii v issledovanii diskretnykh struktur* [New information technologies in the study of discrete structures]. Tomsk, Izdatel'skii dom Tomskogo gosudarstvennogo universiteta, 2000, pp. 59–64 (in Russian).
8. Abrosimov M. B. *Minimal'nye rasshireniia 4-, 5-, 6- i 7-vershinnykh grafov* [Minimal extension of graphs with 4, 5, 6 and 7 vertices]. Saratov State University. Saratov, 2000, 26 p.; VINITI 06.09.2000, no. 2352-B00 (in Russian).
9. Brinkmann G. Isomorphism rejection in structure generation programs. *Discrete Mathematical Chemistry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2000, vol. 51, pp. 25–38. DOI: <https://doi.org/10.1090/dimacs/051/03>
10. McKay B. D. *Graph formats*. Available at: <http://users.cecs.anu.edu.au/bdm/data/formats.html> (accessed 1 May 2019).
11. McKay B. D., Piperno A. Practical Graph Isomorphism, II. *Journal of Symbolic Computation*, 2014, vol. 60, pp. 94–112. DOI: <https://doi.org/10.1016/j.jsc.2013.09.003>
12. Volga Regional Center for New Information Technologies. Site. Available at: <http://prenit.sgu.ru> (accessed 1 May 2019) (in Russian).
13. *Mir grafov* (Graph World). Available at: <http://graphworld.ru> (accessed 1 May 2019) (in Russian).

Cite this article as:

Abrosimov M. B., Sudani H. H. K., Lobov A. A. Construction of All Minimal Edge Extensions of the Graph with Isomorphism Rejection. *Izv. Saratov Univ. (N. S.), Ser. Math. Mech. Inform.*, 2020, vol. 20, iss. 1, pp. 105–115 (in Russian). DOI: <https://doi.org/10.18500/1816-9791-2020-20-1-105-115>
