

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Факультет компьютерных наук и информационных технологий
Кафедра теоретических основ компьютерной безопасности и криптографии

На правах рукописи

Лобов Александр Андреевич

**Моделирование и анализ отказоустойчивости вычислительных
систем с топологией, основанной на гиперкубах**

Специальность 1.2.2 Математическое моделирование, численные методы и
комплексы программ

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:

д.ф.-м.н., проф. Абросимов Михаил Борисович

Саратов 2026

ОГЛАВЛЕНИЕ

Введение.....	4
Глава 1. Алгоритмы построения реализаций моделируемых графами отказоустойчивых систем и способы их оптимизации.....	28
1.1 Задача построения суперграфов.....	32
1.1.1 Метод канонического представителя и матричные коды графов....	33
1.1.2 Метод Рида-Фараджеева.....	36
1.1.3 Дерево максимальных матричных кодов.....	37
1.1.4 Дерево минимальных матричных кодов.....	39
1.1.5 Алгоритмы построения частей графа и его суперграфов.....	41
1.2 Перебор сочетаний с ограничениями.....	42
1.3 Построение Mk-ВОТР методом объединения изоморфных графов.....	48
1.4 Параллельные вычисления.....	54
1.5 Результаты вычислительных экспериментов.....	55
Глава 2. Моделирование вершинной отказоустойчивости гиперкубов.....	62
2.1 Многослойные графы.....	62
2.2 Вершинные отказоустойчивые реализации 4-слойных графов.....	67
2.3 Вершинные отказоустойчивые реализации гиперкуба как 4-слойного графа.....	77
2.4 Связь с результатами вычислительных экспериментов.....	85
Глава 3. Моделирование рёберной отказоустойчивости гиперкубов.....	86
3.1 Единственность минимальной 1-рёберной отказоустойчивой реализацией гиперкубов малой размерности.....	88
3.2 Единственность минимальной 1-рёберной отказоустойчивой реализации гиперкуба размерности более 4.....	95
3.3 Связь с полученными численными методами результатами по минимальным рёберным отказоустойчивым реализациям.....	106
Заключение.....	108

Приложение А. Свидетельство о государственной регистрации программы для ЭВМ GenSupg.....	120
Приложение Б. Свидетельство о государственной регистрации программы для ЭВМ FTConstructor.....	122
Приложение В. Свидетельство о государственной регистрации программы для ЭВМ extvu.....	124

ВВЕДЕНИЕ

Актуальность работы. В данной работе рассматриваются математические модели, методы и вычислительные алгоритмы, применимые для моделирования и анализа отказоустойчивости вычислительных систем с топологией, основанной на гиперкубах.

Все существующие на текущий момент алгоритмы являются комбинаторными и основаны на решении NP -полной задачи проверки вложения графа, также известной как задача «Изоморфизм подграфу». Известные методы используют комбинаторный перебор и требуют значительных вычислительных ресурсов, демонстрируют экспоненциальный рост трудоемкости при увеличении размерности топологии. В связи с этим особую актуальность приобретают методы, направленные на снижение вычислительной сложности путём исключения из перебора одинаковых или изоморфных копий. Обычно для решения таких задач используется метод исключения изоморфных копий на основе метода канонических представителей и метода Рида-Фараджева. Сами эти методы являются лишь общим описанием шагов и не содержат непосредственных инструкций, как их применять к любому математическому объекту, поэтому для каждого объекта необходимо их адаптировать, что само по себе является отдельной сложной задачей.

Степень разработанности. Первой работой, посвящённой отказоустойчивости вычислительных систем с заданной топологией, была статья Авижениса (A. Avižienis) 1975 года ([61]), в которой было описано два подхода повышения надёжности вычислительной системы.

Первый подход (**fault-intolerance**) направлен на снижение вероятности возникновения ошибок за счёт повышения надёжности каждого отдельного

компонента системы и защиты от внешних воздействий, что повышает надежность всей структуры.

Второй подход обеспечивает отказоустойчивость путём **введения избыточных структур**. В систему вводятся избыточные структуры таким образом, чтобы она могла остаться в работоспособном состоянии после возникновения ошибки. При этом система может продолжать работать с частичной деградацией функциональных возможностей (partial fault-tolerance) или без потери функциональных свойств.

Для анализа устойчивости систем к отказам элементов без утраты функциональности, в 1976 году Дж. П. Хейзом (J. P. Hayes) [73] была разработана математическая модель на основе графов, предполагающая введение избыточных структур. Она получила название k -fault tolerance (k -FT) или k -отказоустойчивость. В рамках такой модели вершинами графа обозначаются элементы системы, а связи между ними – рёбрами. Отказ трактуется как удаление вершины и инцидентных ей рёбер из графа. В этой работе рассматривались цветные графы, где каждая вершина имеет определенный цвет, часто кодируемый целыми числами. Вершины одного цвета объединяют элементы одного типа, например коммутаторы или вычислительные устройства. Например, один цвет может обозначать вычислительные модули, а другой – коммутационные устройства.

Было определено понятие оптимальной k -узловой отказоустойчивой реализации (optimal k -fault tolerant). К такой реализации относятся графы, обладающие минимальным избыточным количеством узлов и среди них – минимальным количеством ребер. Данный подход обусловлен стремлением снизить стоимость системы, поскольку затраты на узлы обычно значительно превышают затраты на каналы связи между ними.

В системе отказ может произойти и в соединении между узлами, поэтому позднее в [69] появилась аналогичная модель k -edge fault tolerance (k -EFT) или k -рёберная отказоустойчивость, а предложенная Хейзом модель

получила название *k*-node fault tolerance (*k*-NFT) или *k*-узловая отказоустойчивость [70]. В задаче *k*-узловой отказоустойчивости отказ связи трактуется как удаление соответствующего ребра из графа.

Необходимо уточнить, что в данной работе понимается под термином «граф». Также приведём ряд определений из теории графов, которые потребуются далее [9].

Орграф G определяется как упорядоченная пара $G=(V, \alpha)$, в которой V представляет собой конечное непустое множество вершин, а $\alpha \subseteq V \times V$ является отношением на этом множестве, именуемым отношением смежности. Дуга из вершины u в вершину v обозначается как элемент $(u, v) \in \alpha$.

Граф $G=(V, \alpha)$ считается неориентированным, если его отношение смежности обладает свойствами антирефлексивности $(\forall u \in V)((u, u) \notin \alpha)$ и симметричности $((\forall u, v \in V)((u, v) \in \alpha \Rightarrow (v, u) \in \alpha))$. В таком графе для любой дуги $(u, v) \in \alpha$ существует обратная дуга $(v, u) \in \alpha$, также являющаяся элементом отношения смежности. Это позволяет отождествить обе дуги с единым ребром $(u, v) \in \alpha$, а вершины u и v в этом случае называются смежными.

Неориентированный граф в данной работе называется просто *графом*.

Под *степенью* вершины u подразумевается число смежных с ней вершин. Если выписать степени всех вершин графа и упорядочить их по убыванию, полученная последовательность будет называться *вектором степеней* данного графа.

Граф, полученный удалением вершины v из графа $G=(V, \alpha)$ определяется следующим образом $G-v=(V \setminus \{v\}, \beta=\alpha \cap U \times U)$.

Под результатом добавления к графу $G=(V, \alpha)$ ребра $\{u, v\}$ будем подразумевать граф $G+\{u, v\}=(V, \alpha \cup \{\{u, v\}\})$. Аналогично определяется операция удаления ребра $G-\{u, v\}=(V, \alpha \setminus \{\{u, v\}\})$.

Говорят, что граф $G=(V, \alpha)$ вкладывается в граф $H=(U, \beta)$, если можно найти инъективное отображение $\varphi: V \rightarrow U$, для которого выполняется импликация: $(\forall v_1, v_2 \in V) ((v_1, v_2) \in \alpha \Rightarrow (\varphi(v_1), \varphi(v_2)) \in \beta)$. Само отображение φ в этом случае носит название вложения графа G в граф H .

М. Б. Абросимовым для определения отказоустойчивых моделей графов были использованы понятия *рёберных* и *вершинных k -расширений*. Позднее задача построения расширений была обобщена до задачи построения оптимальной реконструкции в [50].

Для данных задач существует несколько определений для данных объектов, например во многих работах используются термины вершинных и рёберных k -расширений (k -ВОТР и k -РОТР). В данной работе будут использоваться термины, которые более точно отражают исходную суть задачи.

Пусть дан n вершинный граф $G=(V, \alpha)$.

k -вершинной отказоустойчивой реализацией (k -ВОТР) для n -вершинного графа G будем называть такой граф G^* , который удовлетворяет условию: $(\forall v_1, \dots, v_k \in V)$ граф G вкладывается в $G^* - v_1 - \dots - v_k$. Если при этом G^* содержит на k вершин больше, чем исходный граф G , и обладает минимально возможным числом рёбер среди всех k -ВОТР графа G с аналогичным количеством вершин, то он называется минимальной k -ВОТР (Mk -ВОТР).

k -рёберной отказоустойчивой реализацией (k -РОТР) n -вершинного графа $G=(V, \alpha)$ называется граф G^* , для которого справедливо: $(\forall \{v_{i_1}, v_{j_1}\}, \dots, \{v_{i_k}, v_{j_k}\} \in \alpha)$ граф G вкладывается в $G^* - \{v_{i_1}, v_{j_1}\} - \{v_{i_k}, v_{j_k}\}$. В случае, когда число вершин в G^* совпадает с n , а количество рёбер является минимальным среди всех k -ВОТР графа G с тем же числом вершин, G^* называют минимальной k -РОТР.

Минимальные отказоустойчивые реализации являются оптимальными в терминах k -отказоустойчивости Хейза.

Частью графа $G=(V, \alpha)$ называется граф $H=(U, \beta)$ такой, что $U \subseteq V \wedge \alpha \subseteq \beta$. Если $H \neq G$, то такая часть называется *собственной*.

Граф G^* называется *неприводимой k -вершинной отказоустойчивой реализацией* (сокращённо Нк-ВОТР), если никакая его собственная часть не является k -ВОТР графа.

Граф G^* называется *неприводимой k -рёберной отказоустойчивой реализацией* (сокращённо Нк-РОТР), если никакая его собственная часть не является k -РОТР графа.

Если описывать это менее формально, то при удалении любого ребра из неприводимой отказоустойчивой реализации графа G , получится граф, который не является отказоустойчивой реализацией графа G .

Любая минимальная отказоустойчивая реализация графа всегда является неприводимой. В противном случае нашлась бы иная отказоустойчивая реализация для этого же графа с меньшим количеством рёбер, что противоречило бы свойству минимальности. Обратное утверждение неверно. Неприводимая отказоустойчивая реализация не всегда будет минимальной.

Не у всех графов существует минимальная k -рёберная отказоустойчивая реализация. В задаче построения отказоустойчивых реализаций обычно рассматривается конкретный класс графов, например орграфы, неориентированные графы или мультиграфы, и поиск отказоустойчивых реализаций происходит среди графов того же класса.

Рассмотрим полный граф $K_n=(V, V \times V), |V|=n$, ($A \times B = \{(a, b) \mid a \in A, b \in B\}$ — декартово произведение множеств A и B). Для такого графа не существует минимальной k -рёберной отказоустойчивой реализации. Причина в том, что он уже содержит максимально возможное

количество рёбер для данного числа вершин, и добавить новое ребро в него нельзя. При этом по определению количество вершин в минимальной k -рёберной отказоустойчивой реализации должно быть таким же, как в исходном графе.

У каждого графа G есть тривиальная k -вершинная отказоустойчивая реализация (Tk-ВОТР), которую можно получить путём соединения данного графа с полным k -вершинным графом ($G+K_k$, где знаком $+$ обозначают операцию соединения графов).

Соединением графов $G=(V, \alpha)$ и $H=(U, \beta)$ называется граф $G+H=(V \cup U, \alpha \cup \beta \cup (V \times U) \cup (U \times V))$. При этом предполагается, что множества вершин U и V не имеют общих элементов.

Из-за того, что у каждого графа есть тривиальная k -вершинная отказоустойчивая реализация, то минимальная k -вершинная отказоустойчивая реализация существует всегда.

Под n -вершинным циклом понимается граф, который задаётся следующим образом: $C_n=(V, \alpha)$, где $V=\{1, \dots, n\}$, а множество рёбер имеет вид $\alpha=\{\{1, 2\}, \{2, 3\}, \dots, \{n-1, n\}, \{1, n\}\}$.

На рисунке 1 приведён пример каждого из отказоустойчивых реализаций, указанных ранее, для цикла C_8 для k равного 1. Под пунктами а, б, в и г изображены:

- а. Цикл C_8 .
- б. М1-ВОТР цикла C_8
- в. М1-ВОТР цикла C_8 и
- г. Одна из Н1-ВОТР цикла C_8 .

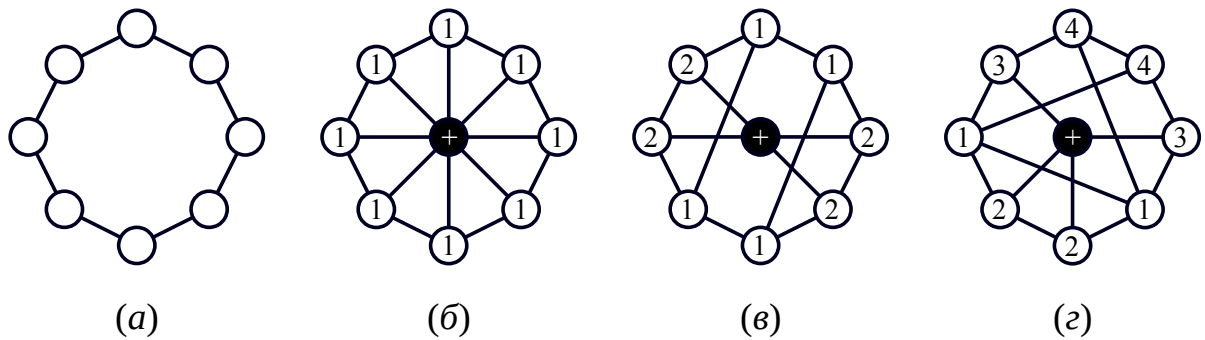


Рисунок 1 – Цикл C_8 (a), его T1-BOtP (б), M1-BOtP (в) и H1-BOtP (г)

Рассмотрим граф $G=(V, \alpha)$. Биективное отображение $\varphi: V \rightarrow V$, удовлетворяющее условию $(\forall \{u, v\} \in V \times V) \{u, v\} \in \alpha \leftrightarrow \{\varphi(u), \varphi(v)\} \in \alpha$, называется автоморфизмом графа G . Если существует автоморфизм φ , переводящий вершину u в v (то есть $\varphi(u)=v$), то вершины u и v называются подобными. Множество всех автоморфизмов графа G принято обозначать как $Aut(G)$.

На рисунке 1 одинаковыми числами помечены подобные вершины.

Покажем, на рисунке 1 изображены 1-BOtP 8-вершинного цикла. Проверку будем осуществлять по определению. Графы, которые получены удалением представителя одного класса подобных вершин, будут изоморфными, поэтому можно рассматривать только один из них.

Напомним определение изоморфизма. Графы $G=(V, \alpha)$ и $H=(U, \beta)$ называются изоморфными, если можно найти такую биекцию $\varphi: V \rightarrow U$, для которой выполняется условие: $(\forall \{u, v\} \in V \times V) \{u, v\} \in \alpha \leftrightarrow \{\varphi(u), \varphi(v)\} \in \beta$. Само отображение φ в этом случае носит название изоморфизма. Поскольку отношение изоморфизма является отношением эквивалентности, оно разбивает всё множество графов на непересекающиеся подмножества, будем называть их классами изоморфизма.

На рисунке 2 разобраны все различные случаи.

В пункте a приведён пример удаления дополнительной вершины. Поскольку при этом ни одно исходное ребро графа не удаляется, вложение

существует всегда — в качестве него можно взять тождественное отображение.

Под пунктами a и $б$ показано вложение C_8 в графы, полученные из Т1-ВОтР удалением по одной из подобных вершин, для М1-ВОтР — под пунктами $в$ и $г$, и для Н1-ВОтР — под оставшимися пунктами $д$, $е$, $ж$ и $з$.

Под n -вершинной цепью P_n понимается граф $P_n = (V_n, \alpha_n)$, где $V_n = \{1, \dots, n\}$ и $\alpha = \{\{1, 2\}, \{2, 3\}, \dots, \{n-1, n\}\}$.

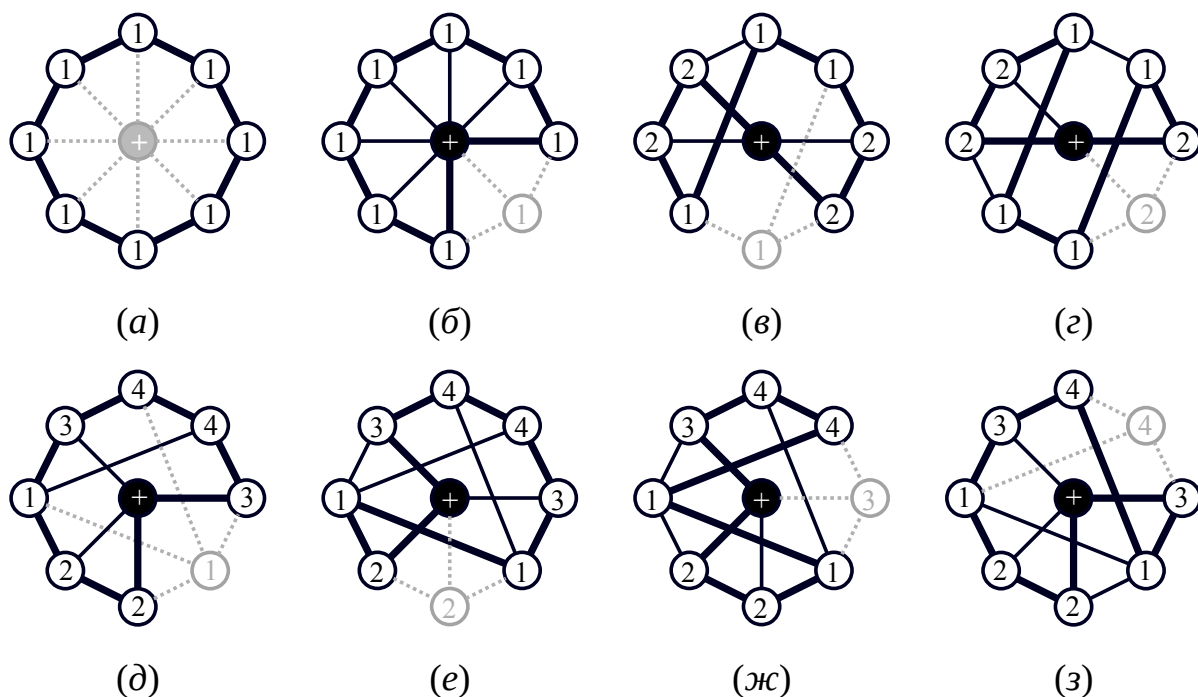


Рисунок 2 – Вложение C_8 в графы, полученные удалением одной вершины из его тривиальной (a , $б$), минимальной ($в$, $г$) и неприводимой ($д$, $е$, $ж$, $з$) отказоустойчивой реализации, показанных на рисунке 1

Для цепей $P_n, n \geq 3$ результат был представлен в первых же работах: в [73] описаны Mk -ВОтР, затем в [69] аналогичным образом описываются Mk -РОтР цепи. Для случая $k=1$ единственными М1-РОтР и М1-ВОтР цепи являются циклы C_n и C_{n+1} соответственно. Для случая $k>1$ Mk -РОтР цепи

P_n являются $M(k-1)$ -РОТР цикла C_n , а Mk -ВОТР цепи – $M(k-1)$ -ВОТР цикла C_{n+1} .

Результаты по целым классам графов обычно получают аналитическим путём. Основной проблемой в таком случае является выдвижение предположений о структуре искомых отказоустойчивых реализаций. Для решения данной проблемы нужны несколько примеров отказоустойчивых реализаций, которые можно получить в результате проведения вычислительного эксперимента, в ходе которого происходит построение Mk -ВОТР или Mk -РОТР для некоторых представителей исследуемого класса графов.

В ряде работ исследуются вычислительные алгоритмы и методы решения задачи поиска Mk -ВОТР и Mk -РОТР графа. Для этого используются различные подходы, но все они заключаются в построении множества графов, которое будет содержать все искомые объекты, и в выборе из этого множества k -ВОТР и k -РОТР графа с минимальным числом рёбер.

Следует также отметить: если граф G является k -ВОТР или k -РОТР графа H , то всякий изоморфный ему граф также будет k -ВОТР или k -РОТР графа, изоморфного H . Поэтому рассматривать все изоморфные графы не имеет смысла. Достаточно проверить одного представителя класса изоморфизма (это же справедливо для Mk -ВОТР, Mk -РОТР, Nk -ВОТР и Nk -РОТР). Для решения этой задачи при построении графов обычно используются результаты работ И. А. Фараджева ([67]), Рида (R. C. Read) ([82], [65], [66]), Маккея (B. McKay) ([74], [75], [76]) и других. На их основе было реализовано множество эффективных генераторов графов, например, программа genreg [77].

Рассмотрим примеры работ, в которых исследуется и применяется алгоритмический подход к построению отказоустойчивых реализаций графов.

В работе [55] данные по M1-ВОТР и M1-РОТР были получены с помощью генератора [54], который строит все попарно неизоморфные графы с заданным вектором степеней. В этой работе рассматриваются циклы, для которых все M1-ВОТР и M1-РОТР имеют одинаковый вектор степеней. В общем случае необходимо определить вектора степеней, которые могут иметь минимальные отказоустойчивые реализации, найти все возможные графы с заданными векторами степеней и проверить их на соответствие отказоустойчивой реализации по определению (кратко – проверить на отказоустойчивость).

В работах [53] и [13] использовались алгоритмы, в основе которых лежит способ, описанный в работе [8]. Данный способ основан на переборе суперграфов в порядке увеличения количества рёбер. Перебираются все графы с заданным числом дополнительных рёбер и если среди них находится хотя бы одна отказоустойчивая реализация, то ответ найден, а если нет, происходит увеличение количества дополнительных рёбер и перебор повторяется. Напомним, что *суперграфом* графа H называется граф G , когда H является частью G .

В [53] и [13] для решения задачи построения отказоустойчивых реализаций был применён метод исключения изоморфных копий, который получилось использовать при особом способе кодирования графа. Данный способ будет рассмотрен в этой работе.

С помощью программного комплекса [25] были построены все M1-ВОТР для графов с числом вершин до 9 включительно [12], все M1-РОТР для графов с количеством вершин до 10 включительно [52], а также все M2-РОТР, M3-РОТР и M4-РОТР для графов с количеством вершин, не превышающим 9.

В некоторых работах, объектом исследования являются отказоустойчивые реализации цветных графов и методов их алгоритмического построения.

Вершинной раскраской графа $G=(V, \alpha)$ в k цветов C называется сюръективное отображение $c:V \rightarrow C$. Если в этом определении заменить множество вершин V на множество рёбер α , мы получим определение рёберной раскраски. Граф, для которого задана такая раскраска, именуется цветным. В качестве множества цветов C обычно выступает некоторое множество мощности k , например, $C=\{0, \dots, k-1\}$.

С самого момента введения графовой модели отказоустойчивости авторы рассматривали цветные графы в качестве объекта исследования. Например, в работе [73], в которой была введена данная модель, рассматривались 1-ВОТР 3-х уровневых утолщённых деревьев (вершины, расположенные на одинаковом уровне, окрашены в один цвет, а те, которые находятся на разных уровнях, – в разные).

В работах [47], [48], [46], [49] и [81] изучались отказоустойчивые реализации цветных звёзд и полных графов, а также алгоритмы их построения. Результаты данных работ были получены путём обобщения полученных путём вычислительного эксперимента отказоустойчивых реализаций с помощью программы построения отказоустойчивых реализаций цветных графов [44], которая опирается на алгоритм генерации цветных графов, использующий метод исключения изоморфных копий графов без проверки на изоморфизм [45].

Графы, для которых можно построить отказоустойчивые реализации программным способом с помощью рассмотренных ранее алгоритмов и программ, имеют небольшое число вершин, например, не были построены все М1-ВОТР и М1-РОТР 30-ти вершинного цикла на данный момент. Теоретические результаты часто охватывают отказоустойчивые реализации графов с большим числом вершин, например, представителей целого класса графов.

Существует ряд исследований, посвященных графам специального вида: в работе [17] рассматриваются функциональные графы, в [6], [7], [18],

[19], [20] и [22] – деревья специального вида, в [21] – объединения полных графов, а [11] посвящена гусеницам (подкласс деревьев).

Если в n -вершинном графе есть вершина степени $n-1$, то он называется *предполным*. В [1] были описаны Mk -ВОТР предполных графов. Также для некоторых предполных графов были результаты и о Mk -РОТР ([2] и [4]).

Для полного графа K_n существует единственное с точностью до изоморфизма Mk -ВОТР – его Tk -ВОТР $K_n + K_k = K_{n+k}$.

Орграф $G=(V, \alpha)$ называется *ориентацией* графа $H=(V, \beta)$, если $(\forall u, v \in V)(\{u, v\} \in \beta \Leftrightarrow ((u, v) \in \alpha \wedge (v, u) \notin \alpha \vee (v, u) \in \alpha \wedge (u, v) \notin \alpha))$.

Менее формально, ориентацией графа H является орграф G , который может быть получен заменой каждого его ребра $\{u, v\}$ на одну из дуг (u, v) или (v, u) .

Изучались также и отказоустойчивые реализации орграфов. В [42] были получены точные оценки количества рёбер в отказоустойчивых реализациях ориентаций полных графов, в [43] – ориентациях циклов и цепей. Отказоустойчивые реализации ориентированных объединений графов изучались в [10].

Поскольку сама задача связана с построением отказоустойчивой вычислительной системы, объектом исследования, как правило, служат графы, описывающие её топологию. Распространёнными объектами исследования являются циклы, многомерные решётки, торы и гиперкубы.

Достаточно работ было посвящено изучению отказоустойчивых реализаций циклов. В [73] было описано Mk -ВОТР цикла и способ их построения, а в [69] описаны некоторые Mk -РОТР цикла. В [85] также изучались отказоустойчивые реализации цикла, так как цикл является основой для сети Token Ring. В целом, вопрос количества попарно неизоморфных Mk -ВОТР и Mk -РОТР цикла не решён: имеющиеся результаты

показывают, что количество таких отказоустойчивых реализаций достаточно велико и быстро растёт с увеличением количества вершин в цикле [55].

Декартовым произведением графов $G=(V, \alpha)$ и $H=(U, \beta)$ называется граф $G \times H=(W, \gamma)$, где

- $W=V \times U=\{(u, v) \mid u \in U \wedge v \in V\}$.
- $\gamma=\left\{\left((v_1, u_1), (v_2, u_2)\right) \mid u_1, u_2 \in U \wedge v_1, v_2 \in V \wedge \right.$
 $\left. \wedge (u_1=u_2 \wedge (v_1, v_2) \in \alpha \vee v_1=v_2 \wedge (u_1, u_2) \in \beta)\right\}$.

Граф $T_{n_1, \dots, n_d}=C_{n_1} \times \dots \times C_{n_d}$ называется d -мерным тором. В свою очередь $M_{n_1, \dots, n_d}=P_{n_1} \times \dots \times P_{n_d}$ называется d -мерной решёткой.

При проектировании коммуникационных сетей суперкомпьютеров широко применяются топологии d -мерных решёток и торов. Так, 3-мерный тор лежит в основе Cray Gemini [60] и Extoll [79]. Сеть «Ангара» [51] построена на топологии 4-мерного тора. Сети суперкомпьютеров IBM BlueGene/Q [71] используют топологию 5-мерного тора; сети Tofu [56], Tofu2 [57], TofuD [58] компании Fujitsu используют топологию 6-мерного тора. Отказоустойчивость является одной из задач при построении таких устройств. Так, например, в суперкомпьютере Fugaku [59], при отказе, сеть продолжает функционировать как решётка той же размерности.

Изначально было выдвинуто предположение о том, что d -мерный тор $T_{n_1, \dots, n_d}$ является M1-РОТР d -мерной решётки $M_{n_1, \dots, n_d}$ [69], однако в [80] показано, что для решёток существует 1-РОТР с меньшим числом дополнительных рёбер и дана схема его построения.

Существуют работы, посвящённые особому классу графов, которые называются циркулянтными. Под *циркулянтным графом* подразумевается граф вида $C_N(S)=\left(\{0, 1, \dots, N-1\}, \{\{u, v\} \mid (\exists i \in S) | u-v| \equiv i \pmod{N}\}\right)$. Вершины циркулянтного графа всегда имеют одинаковую степень, то есть все циркулянтные графы являются *регулярными*.

В [83] описаны k -ВОТР для произвольного циркулянтного графа.

В [64] для d -мерных решёток была представлена схема построения k -ВОТР, которая является циркулянтным графом $C_{2^d+k}(S)$, где

$$S = \bigcup_{i \in \{1, 2, 2^2, \dots, 2^{d-1}\}} \{i, i+1, \dots, i + \lfloor \frac{k}{2} \rfloor\}.$$

Также в данной работе рассматривались отказоустойчивые реализации 1-ВОТР 2-мерных торов.

Аналогичный способ построения 1-ВОТР используется в [14] и [15]. Для произвольного n -вершинного графа $G = (\{v_1, \dots, v_n\}, \alpha)$ составляется объединении графов $G^* = G^0 \cup \dots \cup G^n$. В работе считается, что $G^i = \varphi_i(G)$, где

$$\varphi_i(v_j) = \begin{cases} v_j, & \text{если } j > i \\ v_{j-1}, & \text{иначе} \end{cases}.$$

Отображение φ_i будет являться вложением в случае отказа вершины v_i .

Под *действием* (или *применением*) инъективного отображения $\varphi: V \rightarrow U$ к графу $G = (V, \alpha)$ будем понимать операцию, результатом которой является новый граф $\varphi(G) = H = (\text{Im}_\varphi, \beta)$. Здесь Im_φ обозначает множество образов, а множество рёбер β задаётся правилом: $\beta = \{(\varphi(v_1), \varphi(v_2)) \mid (v_1, v_2) \in \alpha\}$. Очевидно, что полученный таким образом граф H будет изоморфен исходному графу G .

В этих статьях рассматривались различные классы графов, в частности к решётки и торы. Следует заметить, что полученные в них отказоустойчивые реализации не являются минимальными.

Под N -мерным *гиперкубом* или, упрощённо, N -кубом понимается граф $Q_N = (V = \{0, 1\}^N, \alpha = \{(u, v) \mid h(u, v) = 1\})$. В этом определении под h подразумевается расстояние Хэмминга то есть количество позиций, в которых различаются два вектора одинаковой длины. Таким образом, вершинами графа являются все возможные двоичные векторы длины N , и две

вершины соединяются ребром тогда и только тогда, когда их метки отличаются ровно в одном разряде.

Топология гиперкуба исторически играла ключевую роль в архитектуре высокопроизводительных вычислительных систем. В период с 1997 по 2003 год как минимум 10 из 500 самых мощных суперкомпьютеров мира базировались на этой топологии [78]. В частности, её применяли в решениях компаний nCUBE [72] и Intel (iPSC/860) [86]. Однако с увеличением количества узлов в современных суперкомпьютерах наметился переход к производным структурам: обобщённому гиперкубу и расширенному обобщённому гиперкубу [16], который строится на основе квазиполных графов.

Топология гиперкуба применяется по сей день. В списке 500 мощнейших суперкомпьютеров присутствует суперкомпьютер EAGLE, запущенный в 2018 году, который был построен на основе топологии 8-куба. По состоянию на июль 2026 года он занимает 7-е место [78].

Альтернативный способ описания N -куба выглядит следующим образом: $Q_N = Q_{N-1} \times P_2$. Он состоит из двух $(N-1)$ -кубов, с соединёнными соответствующими вершинами. Это означает, что N -куб можно представить как два экземпляра $(N-1)$ -куба, у которых соответствующие вершины соединены между собой рёбрами.

В работе [83] была представлена k -ВОТР для циркулянтного графа, применяемая и для N -куба. Далее в [64] данная схема была улучшена, но при этом в полученных по схеме отказоустойчивых реализаций для гиперкуба дополнительных рёбер больше, чем в Tk -ВОТР.

В [69] была описана схема построения M1-РОТР N -куба для произвольного N , которая совпала с появившейся позднее схемой построения 1-РОТР многомерных решёток [80].

Можно выделить две основные проблемы, связанные с исследованием отказоустойчивости вычислительных систем:

Во-первых, это аналитическое описание отказоустойчивых реализаций для вычислительных систем, моделируемых различными семействами графов.

Во-вторых, это разработка вычислительных алгоритмов, которые позволяют находить отказоустойчивые реализации для конкретных топологий вычислительных систем, и их комбинаторная сложность.

Объектом исследования являются вычислительные системы с топологией, основанной на гиперкубах.

Предметом исследования являются математические модели вершинной и рёберной отказоустойчивых реализаций гиперкубов и вычислительные алгоритмы их построения.

Целью данной работы является исследование математических моделей отказоустойчивости гиперкубов. Особый интерес представляют следующие задачи:

1. Разработать новую математическую модель представления графа, которая позволит использовать методы канонических представителей и Рида-Фараджева, обеспечивающие эффективное исключение изоморфных копий при конструировании суперграфов и частей графа – подзадач, возникающих при решении задачи построения отказоустойчивых реализаций графов.
2. Разработать вычислительный алгоритм построения минимальных k -вершинных отказоустойчивых реализаций, который позволит существенно сократить комбинаторную сложность и решить задачу построения всех минимальных 1-вершинных отказоустойчивых реализаций для 16-вершинного гиперкуба.
3. Разработать оптимизированные модели 1- и 2-вершинных отказоустойчивых реализаций гиперкубов с уменьшенным количеством рёбер.

4. Найти исчерпывающее решение задачи построения устойчивой к отказу одного ребра модели системы на основе топологии гиперкуба и доказать теорему о полноте полученного решения.

Методология и методы исследования. В работе используются методы теории алгоритмов, комбинаторной оптимизации, дискретной математики и теории графов.

Положения, выносимые на защиту:

1. Разработана новая математическая модель представления графа, которая позволяет использовать метод канонических представителей и метод Рида-Фараджева для исключения изоморфных копий при построении суперграфов и частей графа.
2. Разработан и обоснован вычислительный алгоритм построения минимальных k -вершинных отказоустойчивых реализаций, основанный на методе объединения изоморфных графов, который был реализован в программе, написанной на языке C++, и позволил впервые решить задачу построения минимальных 1-вершинных отказоустойчивых реализаций 16-вершинного гиперкуба в ходе вычислительного эксперимента.
3. Разработаны оптимизированные модели построения 1- и 2-вершинных отказоустойчивых реализаций гиперкубов, в которых количество рёбер на 1 и 2 меньше, чем в тривиальных 1- и 2-вершинных отказоустойчивых реализаций соответственно, которые применимы и ко всем двудольным графам, отличным от полных двудольных.
4. Найдено исчерпывающее решение задачи построения оптимальной устойчивой к отказу одного ребра модели системы, основанной на топологии гиперкуба. Сформулирована и доказана теорема о полноте найденного решения, подтверждающая её завершённость.

Научная новизна. Все выносимые на защиту результаты являются новыми:

1. Разработанная математическая модель представления графа позволила применять методы канонических представителей и Рида-Фараджера для решения задачи построения суперграфов и частей графа, что было невозможно при использовании традиционных методов кодирования графа. Реализованная программа «GenSupg» является первой, использующей данный подход к решению этой задачи.
2. Разработан новый вычислительный алгоритм построения минимальных k -вершинных отказоустойчивых реализаций, основанный на методе объединения изоморфных графов. Данный алгоритм был реализован в программе «extvu» и позволил снизить комбинаторную сложность задачи так, что впервые удалось построить все минимальные 1-вершинные отказоустойчивые реализации 16-вершинного гиперкуба в ходе вычислительного эксперимента.
3. Полученные модели построения устойчивых к отказу 1 и 2 узлов систем, основанных на топологии гиперкуба, превосходят предыдущие результаты в плане уменьшенного количества избыточных рёбер и сниженной максимальной степени вершины.
4. Найдено исчерпывающее решение задачи построения устойчивой к отказу одного ребра системы, основанной на топологии гиперкуба. Данный результат сформулирован в виде теоремы и аналитически доказан.

Теоретическая и практическая значимость. Результаты имеют как практический, так и теоретический характер.

Теоретическая значимость состоит в следующем:

1. Разработанный метод кодирования графов позволяет применять методы канонических представителей и Рида-Фараджера для задачи построения частей графа и суперграфов, что невозможно было при иных способах кодирования.

2. Разработан алгоритм, основанный на новом методе объединения изоморфных копий графа, который исключает проверку, основанную на NP-полной задаче. Данный алгоритм показал свою эффективность на 16-вершинном гиперкубе, который ранее не был просчитан ни одним другим алгоритмом.
3. Сформулированы теоремы, позволяющие строить оптимизированные устойчивые к отказу узлов систем, базирующиеся как на топологии гиперкубов, так и на любых других топологиях, представимые двудольными графами, отличными от полных двудольных графов.
4. В работе сформулированы и доказаны теоремы, которые полностью покрывают все возможные минимальные 1-рёберные отказоустойчивые реализации, что полностью закрывает данную задачу.

Практическая значимость состоит в следующем:

1. Предложенный способ кодирования графа позволяет избежать хранения в памяти всех предшествующих решений, что позволяет радикально сократить используемую оперативную память по сравнению с другими алгоритмами. В данной работе в программных реализациях на языке программирования C++ это позволило эффективно распараллеливать вычисления как на основе многопоточности, так и на основе передачи сообщений (технология параллельного программирования MPI). Следовательно, все предложенные алгоритмы будут эффективны как на достаточно мощных современных симметричных мультипроцессорных системах с общей памятью, так и на высокопроизводительных кластерных системах с распределенной памятью.
2. Реализованные приложения «extvu», «GenSupg» и «FTConstructor» позволяют строить графовые модели оптимальных отказоустойчивых реализаций систем, в том числе и гиперкубов.

Степень достоверности и апробация работы. Полученные результаты докладывались на следующих конференциях:

1. VIII Международная научная конференция «Компьютерные науки и информационные технологии» памяти А. М. Богомолова, СГУ, Саратов, 2018.
2. Международная конференция «Ломоносов-2018», МГУ, Москва, 2018.
3. 17-я международная конференция «Сибирская научная школа-семинар «Компьютерная безопасность и криптография»» Sibecrypt'18, Томск, 2018.
4. XIII Международный семинар «Дискретная математика и её приложения» имени академика О.Б. Лупанова, МГУ, Москва, 2019.
5. 18-я международная конференция «Сибирская научная школа-семинар «Компьютерная безопасность и криптография»» Sibecrypt'19, Томск, 2019.
6. Международная конференция «Ломоносов-2021», МГУ, Москва, 2021.
7. IX Международная научная конференция «Компьютерные науки и информационные технологии» памяти А. М. Богомолова, СГУ, Саратов, 2021.
8. 21-я международная конференция «Сибирская научная школа-семинар „Компьютерная безопасность и криптография“» имени Г. П. Агibalова Sibecrypt'22, Красноярск, 2022.

Публикации. По теме диссертации было опубликовано 19 работ, из которых 6 опубликованы в изданиях из списка ВАК, рекомендованных для специальности 1.2.2 «Математическое моделирование численные методы и комплексы программ» (категории К1, К2, К3 указаны по списку ВАК, уровни У1, У2, У3, У4 указаны по списку Российского центра научной информации, квартили Q1, Q2, Q3, Q4 указаны согласно сайту SCImago):

- В ВАК, Web of Science и Scopus входит 3 работы: [30] (К1, У1, Q3 по WoS/Scopus), [31] (К1, У1, Q3) и [37] (К1, У1, Q3).

- Помимо этого в ВАК входит 3 работы: [33] (К1, У4), [39] (К1, У4), [40] (К2).
- Только в РИНЦ входят 7 работ: [26], [27], [29], [32], [34], [36], [38].
- Не входят в РИНЦ 3 работы: [28], [35], [41].
- 3 свидетельства о регистрации программы для ЭВМ: [23], [24] и [25].

Зарегистрированные программы для ЭВМ использовались для проведения вычислительных экспериментов, позволивших получить отказоустойчивые модели гиперкубов небольших размеров, которые позволили выдвинуть предположения о способе построения моделей гиперкубов больших размерностей, а также для проверки неприводимости рассматриваемых отказоустойчивых реализаций и нахождения частей, которые также являются отказоустойчивыми реализациями (операция приведения отказоустойчивых реализаций).

Для всего этого были использованы разработанные программы, а именно программа для построения всех попарно неизоморфных суперграфов GenSupg [24] (свидетельство приведено в приложении А), программный комплекс построения отказоустойчивых реализаций графов методом исключения изоморфных копий FTConstructor [25] (свидетельство приведено в приложении Б) для построения всех M_k -ВОТР и M_k -РОТР заданного графа с использованием техник исключения изоморфных копий и программа «extvu» [23] (свидетельство приведено в приложении В) для построения M_1 -ВОТР 16-вершинного гиперкуба разработанным методом объединения изоморфных копий.

Личный вклад. Все выносимые на защиту результаты получены соискателем лично.

Научный руководитель М. Б. Абросимов поставил тему диссертационной работы и осуществлял проверку научные статьи.

Личный вклад автора в основные результаты, опубликованные в работах, заключаются в следующем:

В статье [30] («Построение всех неизоморфных минимальных вершинных отказоустойчивых реализаций графа методом канонических представителей») автором предложена математическая модель представления графа, позволяющая использовать методы исключения изоморфных копий без проведения проверки на изоморфизм для задачи генерации суперграфов.

В статье [31] («Построение минимальных рёберных отказоустойчивых реализаций графа без проверки на изоморфизм») автором описан метод построения суперграфов, исключающий проверку на изоморфизм как подзадачу построения минимальных рёберных отказоустойчивых реализаций.

В статье [32] («Построение всех неизоморфных суперграфов без проверки на изоморфизм») автор обобщил предыдущие результаты, предложив единый алгоритмический подход к построению суперграфов без проверки на изоморфизм.

В статье [33] («Построение минимальных вершинных отказоустойчивых реализаций графа методом Рида-Фараджеева») автором описана система кодирования графов и структура дерева кодов. Это позволило внедрить технику исключения изоморфных копий по методу Рида-Фараджеева в процесс построения суперграфов.

Таким образом, личный научный вклад автора в данных четырёх статьях заключается в разработке способов кодирования графов, обеспечивающих эффективное построение суперграфов, а также в решении сопутствующих задач их генерации.

Остальные работы, в которых только один автор или в качестве соавтора указан только научный руководитель, были написаны соискателем полностью лично.

В программном комплексе «FTConstructor» автор занимался разработкой способа кодирования графа, связанного с проверкой графа на каноничность, а также вспомогательными функциями.

Программы «GenSupg» и «extvu» были полностью разработаны соискателем на языках С и С++ соответственно.

Объем и структура работы. Работа состоит из введения, трёх глав, заключения, списка литературы и трёх приложений.

В первой главе «Алгоритмы построения реализаций моделируемых графами отказоустойчивых систем и способы их оптимизации» описываются численные методы решения задачи и возникающие при их использовании подзадачи, приводятся и обосновываются алгоритмы, используемые для решения задач:

1. Приведены способы кодирования графа, позволяющие использовать техники исключения изоморфных копий для построения суперграфов и частей заданного графа.
2. Описаны и обоснованы алгоритмы построения всех Mk -РОТР и Mk -ВОТР заданного графа, в основе которых лежит генерация суперграфов и частей без проверки на изоморфизм.
3. Разработан новый алгоритм построения Mk -ВОТР, основанный на ранее не использованном методе объединения изоморфных графов, позволяющий исключить решение вычислительно сложной задачи проверки вложения графа. Приводятся способы комбинаторной оптимизации данного алгоритма на основе автоморфизмов графа.

Во второй главе «Моделирование вершинной отказоустойчивости гиперкубов» указываются теоретические результаты, полученные при поиске схем построения вершинных отказоустойчивых реализаций гиперкубов:

1. Описаны устойчивые к отказу одного и двух узлов модели систем, основанных на топологии гиперкуба, которые могут применяться и для более широкого класса систем, основанных на топологии двудольных графов, отличных от полных двудольных.

2. Посчитано количество различных отказоустойчивых систем, основанных на топологии гиперкубов, которые можно получить с помощью данных моделей. Также исследованы их свойства.

В третьей главе «Моделирование рёберной отказоустойчивости гиперкубов» представлен и обоснован результат, описывающий все возможные устойчивые к отказу одного ребра модели гиперкуба, который полностью закрывает данную задачу.

В приложениях приводятся три свидетельства о регистрации разработанных программ для ЭВМ.

Объём работы – 125 страниц, 86 использованных источников, 30 рисунков и 1 таблица.

Благодарности. Автор выражает глубокую благодарность своему научному руководителю доктору физико-математических наук Михаилу Борисовичу Абросимову за постановку темы и задач исследования, за многолетнюю плодотворную работу, всестороннюю помощь и поддержку в вопросе подготовки диссертации.

Глава 1. Алгоритмы построения реализаций моделируемых графами отказоустойчивых систем и способы их оптимизации

Существуют универсальные алгоритмы построения вершинных и рёберных отказоустойчивых реализаций. Все данные алгоритмы основаны на переборе графов из заданного множества.

Как ранее было сказано, если граф G^* является k -ВОТР графа G , то все изоморфные ему графы также будут являться k -ВОТР графа G . Это верно и для k -РОТР. Учитывая это, при построении всех отказоустойчивых реализаций достаточно рассматривать только попарно неизоморфные графы. Эта особенность применяется во всех алгоритмах.

Перед описанием алгоритмов следует отметить, что если граф G^* является k -ВОТР графа G , то все изоморфные G^* графы также будут являться k -ВОТР графа G . Верно и противоположное. Аналогичная ситуация и с k -РОТР. Следовательно, при построении всех отказоустойчивых реализаций достаточно рассматривать только попарно неизоморфные графы, что учитывают все алгоритмы.

В [8] описано несколько алгоритмов построения вершинных и рёберных отказоустойчивых реализаций произвольного графа. Первый вид алгоритмов основан на переборе суперграфов заданного графа и поиске среди них отказоустойчивых реализаций. Графы рассматриваются в порядке увеличения числа рёбер в них. Первые найденные отказоустойчивые реализации будут минимальными, так как никакие графы с меньшим числом рёбер отказоустойчивыми реализациями нужного типа не являются. Далее нужно перебрать все суперграфы с заданным числом рёбер и среди них будут находиться все неизоморфные отказоустойчивые реализации графов. Данный способ построения опирается на то, что среди суперграфов будут представители каждого класса изоморфизма минимальных отказоустойчивых

реализаций заданного типа данного графа. Это утверждение интуитивно понятно, но можно это доказать и более строго. Доказательство описано в лемме 1.

Лемма 1. Пусть $\phi: V \rightarrow U$ является вложением графа $G=(V, \alpha)$ в $H=(U, \beta)$ и $V \subseteq U$. Тогда можно построить граф $H_1=(U_1, \beta_1)$, изоморфный H , такой, что G будет являться его частью.

Доказательство. Поскольку $\phi: V \rightarrow U$ – вложение, выполняется условие $(\forall x, y \in V)((x, y) \in \alpha \rightarrow (\phi(x), \phi(y)) \in \beta)$. В силу инъективности отображения ϕ , образы различных вершин не совпадают, что позволяет определить обратное отображение $\phi^{-1}: \text{Im}_\phi \rightarrow V$ как $\phi^{-1}(y) = x \Leftrightarrow \phi(x) = y$. Обозначим через $\{q_1, \dots, q_{|U|-|V|}\}$ элементы множества $U \setminus V$, а через $\{y_1, \dots, y_{|U|-|V|}\}$ элементы множества $U \setminus \text{Im}_\phi$. Определим преобразование $\psi: U \rightarrow U$ следующим образом:

$$\psi(x) = \begin{cases} \phi^{-1}(x), & \text{если } x \in \text{Im}_\phi \\ q_i, & \text{если } x = y_i \end{cases}$$

Рассмотрим граф $\psi(H) = H_1 = (U, \beta_1)$. Покажем, что все рёбра графа G будут являться рёбрами графа H_1 :

$$\begin{aligned} \{x, y\} \in \alpha &\Rightarrow x, y \in V \Rightarrow \phi(x), \phi(y) \in \text{Im}_\phi \Rightarrow \psi(\phi(x)) = \phi^{-1}(\phi(x)) = x \wedge \\ &\wedge \psi(\phi(y)) = \phi^{-1}(\phi(y)) = y \Rightarrow \{\psi(\phi(x)), \psi(\phi(y))\} = \{x, y\}. \end{aligned}$$

Это означает, что граф G является частью графа H_1 по определению. Так как H_1 изоморфен H , в который вкладывается G , значит лемма верна, что и требовалось доказать.

■

Следствие. Так как определения k -ВОТР и k -РОТР основаны на вложении, аналогичное утверждение справедливо и для отказоустойчивых реализаций. Если граф G^* является отказоустойчивой реализацией графа G ,

то существует изоморфный G^* граф H , который является суперграфом графа G . Отсюда и следует корректность алгоритма, основанного на переборе суперграфов.

■

В [8] описаны отдельные алгоритмы для задач построения Mk -РОТР и Mk -ВОТР. Данные алгоритмы очень похожи друг на друга структурно, поэтому данный подход можно обобщить в качестве алгоритма 1.

Алгоритм 1 – Базовый алгоритм построения Mk -ВОТР и Mk -РОТР

Вход: граф $G=(V, \alpha)$, k , m_{min} , тип отказоустойчивой реализации (вершинная или рёберная).

Выход: все неизоморфные Mk -ВОТР графа G .

1. Если нужно построить Mk -ВОТР то:

$$1.1. \quad G_T = G \cup O_k.$$

$$1.2. \quad m_{max} = |\alpha| + k|V| + \left(\frac{k(k-1)}{2}\right).$$

Иначе:

$$1.3. \quad G_T = G.$$

$$1.4. \quad m_{max} = \frac{(|V|-1)|V|}{2}.$$

2. $m = m_{min}$.

3. Если $|\alpha| + m > m_{max}$, то результат $\emptyset$, завершить алгоритм.

4. $E = \emptyset$.

5. Для каждого графа H , полученного добавлением к G_T m рёбер:

5.1. Если H является отказоустойчивой реализацией графа G нужного типа и $\nexists H_1 \in E : H_1 \cong H$ выполнить $E = E \cup \{H\}$.

6. Если $E \neq \emptyset$ то результат E , завершить алгоритм, в противном случае перейти к шагу 7.

7. $m = m + 1$.

8. Перейти к шагу 3.

Значения m_{max} на шаге 1 – это максимальное число рёбер, до которого нужно дойти. Если не было встречено ни одной отказоустойчивой реализации, то это означает, что минимальных отказоустойчивых реализаций просто не существует, что возможно для Mk -РОТР. Например для полных графов не существует неориентированного графа с числом рёбер, больше, чем у полного и с тем же количеством вершин. Значение m_{max} для задачи Mk -РОТР определяется как максимальное число рёбер для неориентированного графа с числом вершин как у исходного графа. Для Mk -ВОТР всегда существует тривиальное k -ВОТР, получающееся соединением его с полным k -вершинным графом, поэтому m_{max} равен количеству рёбер в такой отказоустойчивой реализации. В полном k -вершинном графе $\frac{k(k-1)}{2}$ рёбер и соединение каждой вершины G с данным полным графом даёт $k|V|$ рёбер дополнительно.

Если известна нижняя граница количества дополнительных рёбер в Mk -ВОТР, то можно начинать алгоритм не с 1, а с данной границы m_{min} . Получить нижнюю границу можно с помощью использования необходимых условий из лемм 2 и 3 (данные леммы описаны в [8]).

Лемма 2. Пусть $d > 0$ – минимальная степень вершины в графе G . Тогда в его Mk -ВОТР не может быть вершин, степень которых меньше $d + k$.

Лемма 3. Пусть $d > 0$ – минимальная степень вершины в графе G . Тогда в его Mk -РОТР не может быть вершин, степень которых меньше $d + k$.

Например, если взять $M1$ -РОТР 3-куба, то его вектор степеней равен (3^8) , а по леммам 2 и 3 в его $M1$ -ВОТР и $M1$ -РОТР не должно быть вершин со степенью меньше $d_{min} + k$, в данном случае – это 4. А значит, у $M1$ -ВОТР и $M1$ -РОТР степени вершин будут покомпонентно не меньше, чем (4^9) и (4^8) соответственно. По вектору степеней можно определить количество рёбер в

графе по лемме о рукопожатии [9] как сумму элементов вектора, делённую на 2. В первом случае – это $\frac{4 \cdot 9}{2} = 18$, во втором – $\frac{4 \cdot 8}{2} = 16$, следовательно, m_{min} будет равен 6 и 4 соответственно, так как у 3-куба 12 рёбер. Если при делении получается число не целое, то результат деления нужно округлять вверх, так как это именно нижняя граница.

Пункт 4 в алгоритме является ключевым. В нём решаются все основные задачи:

1. Построение суперграфов.
2. Проверка рассматриваемого графа на отказоустойчивость.
3. Исключение изоморфных копий.

Согласно [5] и [3] задача проверки графа на соответствие k -вершинной и рёберной отказоустойчивой реализации является NP -полной. Следовательно, данный алгоритм поиска минимальных отказоустойчивых реализаций, в основе которого лежит решение данной задачи, в общем случае является вычислительно сложным.

Рассмотрим все эти подзадачи по отдельности.

1.1 Задача построения суперграфов

Данная задача является основной в базовом алгоритме. Суперграфы можно перебирать с помощью матрицы смежности или матричного кода графа. Достаточно определить позиции, на которых стоит 0 – это множество рёбер, которые можно добавить в граф, пусть это будет множество α . Далее все суперграфы с заданным числом дополнительных рёбер m можно получить с помощью алгоритма перебора сочетаний из m по $|\alpha|$. Их количество равно $C_{|\alpha|}^m$. Однако, среди них будет встречаться много изоморфных, среди которых впоследствии нужно будет оставить лишь один. Можно данную задачу решить на этапе перебора на основе техники

исключения изоморфных копий, общая концепция которой хорошо описана в [63].

Впервые данный способ построения суперграфов был реализован в программе Gensupg [24]. В ней были реализованы методы канонического представителя и Рида-Фараджева ([63]), а в основе лежал особый метод кодирования графа. После этого данные результаты были описаны в совместной публикации с Камилем И. А. К. и Судани Х. Х. К. ([32]), которые занимались алгоритмами построения минимальных вершинных и рёберных отказоустойчивых реализаций графов без проверки на изоморфизм и были использованы в реализации программного комплекса «FTConstructor» [25].

Далее будут описаны основные моменты, связанные с генерацией суперграфов с помощью техник исключения изоморфных копий.

1.1.1 Метод канонического представителя и матричные коды графов

Метод канонических представителей является базовым способом решения задачи исключения изоморфных копий без проверки на изоморфизм. Суть этого метода заключается в следующем: во-первых, выбирается способ кодирования объектов, а во-вторых, среди кодов изоморфных объектов выбирается один в качестве канонического. Граф, имеющий данный код называется каноническим представителем.

Приведём один пример. Для графа способом кодирования может являться матричный код графа, формируемый путем поэлементного выписывания матрицы смежности по строкам. Для неориентированных графов обычно рассматриваются только элементы ниже или выше главной диагонали. В качестве канонического кода выбирается минимальный (миникод) или максимальный матричный код (максикод) среди изоморфных графов.

При таком способе кодирования канонический представитель может не быть суперграфом, поэтому для построения суперграфов необходим другой способ кодирования. На рисунке 3 изображён пример такой ситуации. Под буквой *a* нарисован граф *G*. Далее, под буквой *б* показан его суперграф *H* с максимальным кодом среди изоморфных ему суперграфов *G*. Он не является каноническим представителем, так как существует другой изоморфный ему граф с большим кодом. Этот граф изображён под пунктом *в*. Под изображением каждого из графов выписан его код.

Данный пример показывает, что такой способ кодирования графов не подходит для решения задачи построения суперграфов методом канонического представителя, поэтому нужен другой метод кодирования.

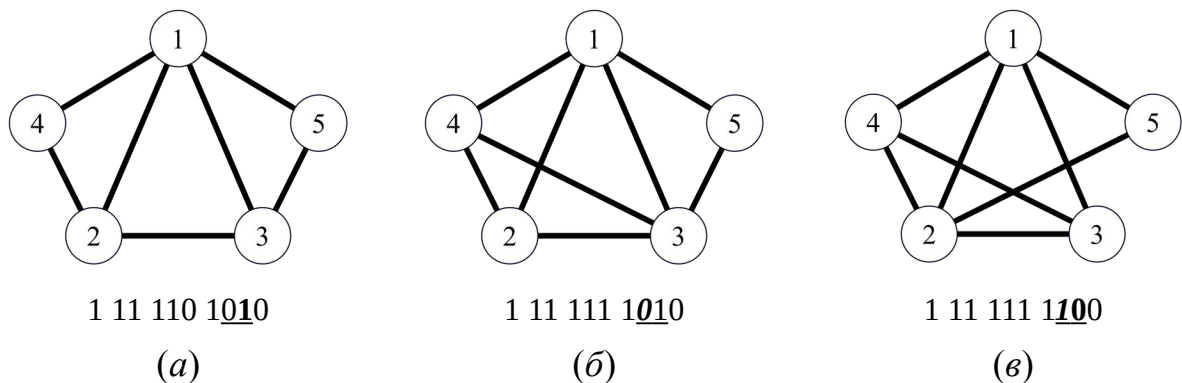


Рисунок 3 – Пример суперграфа (б) графа *G* (a) с максимальным кодом среди суперграфов *G*, и изоморфный ему граф (в), не являющийся суперграфом графа *G*, с большим чем у (б) кодом

Изменим порядок выписывания элементов матрицы смежности. Для графа *G* определим построение кода $C^G(H)$ следующим образом: выписываются элементы матрицы смежности графа *H* по следующим правилам:

- Элементы матрицы смежности выписываются в 2 этапа.
- Рассматриваются только те элементы матрицы смежности, которые находятся ниже главной диагонали.

- Элементы матрицы смежности выписываются построчно, строки выписываются вверху вниз, а элементы слева направо.
- На первом этапе элемент выписывается, если на аналогичной позиции в матрице смежности графа G стоит единица.
- На втором этапе элемент выписывается, если на аналогичной позиции в матрице смежности графа G стоит ноль.

Примечание. В данной работе рассматриваются только неориентированные графы с симметричной матрицей смежности, но если работа ведётся с орграфами, то можно выписывать также симметричные элементы и элементы с главной диагонали. Тогда описанный метод генерации суперграфов также будет работать и для орграфов.

Рассмотрим заложенный в данный способ кодирования принцип. Код графа G будет равен $C^G(G) = 1...10...0$. Если в графе H нет хотя бы одного ребра, которое есть в G , то вместо одной из начальных единиц будет стоять ноль, из-за чего $C^G(H) < C^G(G)$. Если же все рёбра графа G присутствуют в H , то он по определению будет являться суперграфом графа G , в то же время его код будет не меньше, чем код G , то есть $C^G(S) \geq C^G(G)$.

Это означает, что код любого суперграфа графа G больше, чем код любого графа, который таковым не является, из чего следует то, что максимальным кодом среди изоморфных графов, в которые вкладывается G , будет обладать один из его суперграфов. Следовательно, если выбрать в качестве канонического представителя класса изоморфизма граф с максимальным кодом M_G , то канонический представитель отказоустойчивых реализаций будет находиться именно среди суперграфов. Значит, метод канонического представителя для решения задачи исключения изоморфных копий среди графов в алгоритме 1 использовать с данным способом кодирования можно.

Ещё одной задачей, которую можно решить с помощью канонических представителей, является построение частей графа. Она может пригодиться при проверке графа на рёберную отказоустойчивую реализацию и неприводимость. В обоих случаях необходимо удалять рёбра из графа и проверять вложение, а удаление рёбер по сути является перебором частей графа.

Для решения этой задачи рассмотрим второй способ построения матричного кода графа, аналогичный рассмотренному ранее, но с другим порядком выписывания:

- На первом этапе элементы выписываются, если на аналогичной позиции в матрице смежности графа G стоит ноль.
- На втором этапе элемент выписывается, если на аналогичной позиции в матрице смежности графа G стоит единица.

Такой код графа G обозначим как D^G . При такой организации построения кода получим, что $D^G(G)=0...01...1$. Если граф H не является частью G , то вместо какого-либо нуля в коде графа G должна стоять единица, поэтому $D^G(H) > D^G(G)$. Если же граф S является частью G , то отличаться их коды будут в позициях, где у кода G стоят единицы, поэтому $D^G(S) \leq D^G(G)$. Получается, что коды всех частей G меньше кода G . Это значит, что коды графов, вкладывающихся в G , но не являющихся его частью, не минимальные по лемме 1.

1.1.2 Метод Рида-Фараджеева

Метод Рида–Фараджеева представляет собой оптимизированную версию метода канонического представителя. Его ключевая идея заключается в поэтапном построении кода объекта. Процесс начинается с формирования начального приближения, которое затем многократно уточняется до

получения полного кода искомого объекта. Важным условием является возможность на каждом этапе определить, будет ли текущий промежуточный код приводить к неканоническому результату. Это позволяет на раннем этапе прервать генерацию неперспективных ветвей (применение метода ветвей и границ). При этом одно промежуточное приближение может быть общим для нескольких различных конечных объектов.

С практической точки зрения метод требует организации всех кодов в виде дерева, так что канонические представители каждого класса изоморфизма образуют его поддереву. Ключевое свойство метода заключается в том, что если промежуточный код не является каноническим, то и все его потомки в дереве также не будут каноническими.

Рассмотрим примеры.

1.1.3 Дерево максимальных матричных кодов

Пусть код графа G – двоичное число, полученное путём выписывания элементов матрицы смежности в произвольном порядке. Все перечисленные ранее коды попадают под такое определение.

Рассмотрим следующий способ организации дерева:

1. Корень дерева – элемент с кодом $0\dots 0$
2. Родителем элемента с кодом $x_1\dots x_{k-1}10\dots 0$ будет элемент с кодом $x_1\dots x_{k-1}00\dots 0$.

При подобном построении дерева у него будет поддерево, образованное максимальными кодами. Покажем это.

Теорема 1 (О поддереве максимальных кодов). Пусть двоичная последовательность $x_1\dots x_{k-1}10\dots 0$ представляет собой максимальный матричный код. В таком случае код $x_1\dots x_{k-1}00\dots 0$, соответствующий

родительскому элементу в дереве, также будет являться максимальным матричным кодом.

Доказательство. Пусть условие теоремы не выполняется, и $c_1 = x_1 \dots x_{k-1} \mathbf{1} 0 \dots 0$ является максимальным кодом среди графов, принадлежащих классу изоморфизма, а $c_2 = x_1 \dots x_{k-1} \mathbf{0} 0 \dots 0$ не является. Это означает, что существует код изоморфного графа $c_2^\phi = y_1 \dots y_{k-1} y_k \dots y_n > x_1 \dots x_{k-1} \mathbf{0} 0 \dots 0 = c_2$. Обозначим изоморфизм из графа с кодом c_2 в граф с кодом c_2^ϕ как ϕ .

Раз $c_2^\phi = y_1 \dots y_{k-1} y_k \dots y_n > x_1 \dots x_{k-1} \mathbf{0} 0 \dots 0 = c_2$, то должна быть позиция, где в c_2^ϕ стоит 1, а в c_2 – 0. Обозначим её номер как t . Так как графы с данными кодами изоморфны, то количество единиц в них одинаково. Все элементы на позициях с номерами, меньшими чем t , будут равны. У $c_2 = x_1 \dots x_{k-1} \mathbf{0} 0 \dots 0$ все единицы стоят на первых $k-1$ позициях. Если $t \geq k$, то первые $k-1$ элемента кодов c_2^ϕ и c_2 совпадают, а значит, все единицы кода c_2 должны быть среди первых $k-1$. Но одна единица должна также быть и на позиции $t \geq k$, что невозможно ввиду равенства количества единиц в кодах. Получили противоречие. Значит, $t < k$, поэтому c_2^ϕ можно записать как $c_2^\phi = x_1 \dots x_{t-1} \mathbf{1} y_{t+1} \dots y_{k-1} y_k \dots y_n > x_1 \dots x_{t-1} \mathbf{0} x_{t+1} \dots x_{k-1} \mathbf{0} 0 \dots 0 = c_2$. Отметим, что $x_t = 0$.

Так как коды c_1 и c_2 отличны только в позиции k , то $c_1 = x_1 \dots x_{t-1} \mathbf{0} x_{t+1} \dots x_{k-1} \mathbf{1} 0 \dots 0$. Применим ϕ к графу с кодом c_1 и получим граф с кодом c_1^ϕ , который отличается от c_2^ϕ только в одной позиции s , в которую переместится единица с позиции k кода c_1 .

Если $s > t$, то вместо одной из $y_{t+1}, \dots, y_n$, которая равна 0, стоит 1, значит, $c_1^\phi > c_2^\phi = x_1 \dots x_{t-1} \mathbf{1} y_{t+1} \dots y_{k-1} y_k \dots y_n > x_1 \dots x_{t-1} \mathbf{0} x_{t+1} \dots x_{k-1} \mathbf{1} 0 \dots 0 = c_1$, что противоречит максимальнойности c_1 . Новая единица на позиции t появиться не

может, так как в c_2^ϕ на этом месте уже стоит единица. Следовательно, $s < t \leq k-1$.

Если $s < t \leq k-1$, то в коде c_1 у элемента на позиции с номером s стоит 0, так как первые $k-1$ элементы кодов c_1 и c_2^ϕ равны, а отличие c_2^ϕ и c_1^ϕ заключается только в том, что на позиции s в c_1^ϕ стоит единица. Значит, $c_1^\phi = x_1 \dots x_{s-1} \mathbf{1} x_{s+1} \dots x_{t-1} \mathbf{1} y_{t+1} \dots y_n > x_1 \dots x_{s-1} \mathbf{0} x_{s+1} \dots x_{k-1} \mathbf{1} 0 \dots 0 = c_1$.

Таким образом, если код c_2 не максимальный, то в любом случае $c_1^\phi > c_1$, а значит, c_1 тоже не максимальный, что противоречит предположению.

■

Теорема 1 показывает, что поддерево максимальных матричных кодов представителей существует для любого порядка выписывания элементов матрицы смежности, так как элементы кода были задействованы по номерам, получаемым в порядке выписывания.

1.1.4 Дерево минимальных матричных кодов

Можно также построить дерево минимальных матричных кодов. Пусть код графа – двоичное число, полученное путём выписывания элементов матрицы смежности в произвольном порядке. Все перечисленные ранее коды попадают под такое определение.

Рассмотрим второй способ организации дерева графов:

1. Корень дерева – элемент с кодом $1 \dots 1$.
2. Родителем элемента с кодом $x_1 \dots x_{k-1} \mathbf{0} 1 \dots 1$ будет элемент с кодом $x_1 \dots x_{k-1} \mathbf{1} 1 \dots 1$.

Следует отметить, что данное дерево можно получить из дерева максимальных матричных кодов, заменив код каждого графа на код его дополнения (дополнением графа $G=(V, \alpha)$ называется граф

$\bar{G}=(V,(V\times V)\setminus\alpha\setminus\{(v,v)|v\in V\}))$. Для этого достаточно показать, что поразрядная инверсия максимального кода графа является минимальным кодом его дополнения (теорема 2).

Теорема 2 (Связь максимального и минимального матричных кодов графа и его дополнения). Если выполнить побитовую инверсию максимального матричного кода графа G , полученная последовательность будет в точности равна минимальному матричному коду дополнения графа G .

Доказательство. Предположим, что последовательность $x_1 \dots x_{t-1} x_t \dots x_n$ является максимальным матричным кодом графа $\setminus(G)$. Тогда код дополнения $\bar{G}$ примет вид $\bar{x}_1 \dots \bar{x}_t \bar{x}_{t+1} \dots \bar{x}_n$. Допустим, что этот код не является минимальным. Следовательно, найдётся изоморфный ему граф G^* , код которого $\bar{x}_1 \dots \bar{x}_t y_{t+1} \dots y_n$ будет меньше. Пусть первые t элементов кодов совпадают (t может принимать значение 0), а первое различие возникает на позиции $t+1$.

Так как $\bar{x}_1, \dots, \bar{x}_t, y_{t+1}, \dots, y_n$ меньше, чем $\bar{x}_1, \dots, \bar{x}_t, \bar{x}_{t+1}, \dots, \bar{x}_n$, то $y_{t+1} < \bar{x}_{t+1}$. Рассмотрим дополнение $\bar{G}^*$ графа G^* . Код его будет равен $x_1, \dots, x_t, \bar{y}_{t+1}, \dots, \bar{y}_n$. Так как $y_{t+1} < \bar{x}_{t+1}$, то $\bar{y}_{t+1} > x_{t+1}$, а значит, $x_1 \dots x_{t-1} x_t \dots x_n < x_1 \dots x_t \bar{y}_{t+1} \dots \bar{y}_n$. Так как $\bar{G}$ и G^* изоморфны, то их дополнения G и $\bar{G}^*$ тоже изоморфны, а так как код $\bar{G}^*$ больше, чем код G , который максимальный по условию, получили противоречие, поэтому $\overline{C_{max}(G)} = C_{min}(\bar{G})$. ■

Следствие. Функцию построения минимального матричного кода можно получить через функцию построения максимального матричного кода, побитовой инверсии и дополнения графа: $C_{min}(G) = \overline{C_{max}(\bar{G})}$. Аналогично с максимальным матричным кодом: $C_{max}(G) = \overline{C_{min}(\bar{G})}$.

1.1.5 Алгоритмы построения частей графа и его суперграфов

Опираясь на деревья, которые были описаны выше, можно предложить следующие алгоритмы построения суперграфов (алгоритм 2) и частей графа (алгоритм 3).

Алгоритм 2 – Построение суперграфов

Вход: граф G , число добавляемых рёбер m .

Выход: множество суперграфов графа G , содержащих m дополнительных рёбер.

1. Инициализировать $C = \emptyset$, $N = \{G\}$, $t = 0$.
 2. Если $t = m$, то завершить работу алгоритма и выдать в качестве результата множество N .
 3. Инициализировать $C = N$, $N = \emptyset$.
 4. Для каждого $H \in C$ выполнить:
 - 4.1. $c = x_1 \dots x_{k-1} 1 0 0 \dots 0 0$ – код графа H .
 - 4.2. $N = N \cup \{x_1 \dots x_{k-1} 1 1 0 \dots 0 0, \dots, x_1 \dots x_{k-1} 1 0 0 \dots 0 1\} \cap C_{max}^G$.
 5. Выполнить $t = t + 1$ и перейти к шагу 2.
-

Алгоритм 3 – Построение частей графа

Вход: граф G , количество удалённых рёбер m .

Выход: множество попарно неизоморфных частей графа G , из которого удалено m рёбрами.

1. Инициализировать $C = \emptyset$, $N = \{G\}$, $t = 0$.
 2. Если $t = m$, то завершить работу алгоритма и выдать в качестве результата множество N .
 3. Инициализировать $C = N$, $N = \emptyset$.
 4. Для каждого $H \in C$ выполнить:
 - 4.1. $x_1 \dots x_{k-1} 0 1 1 \dots 1 1$ – код графа H .
 - 4.2. $N = N \cup \{x_1 \dots x_{k-1} 0 0 1 \dots 1 1, \dots, x_1 \dots x_{k-1} 0 1 1 \dots 1 0\} \cap D_{min}^G$.
 5. Выполнить $t = t + 1$ и перейти к шагу 2.
-

Следует обратить внимание на структурное сходство алгоритмов 2 и 3. Оно объясняется тем, что оба алгоритма по сути реализуют одну и ту же процедуру – обход поддерева. Ключевое различие между ними заключается в том, какой именно матричный код используется при построении.

На деле можно использовать несколько другой подход. Как ранее было отмечено в теореме 2, максимальный код графа является минимальным кодом его дополнения, поэтому задачу построения частей графа G можно решить по-другому – дополнения суперграфов дополнения G будут являться частями G . При этом, так как если G и H изоморфны тогда и только тогда, когда их дополнения тоже изоморфны (их изоморфизмы будут одинаковыми), то задача будет решена полностью.

1.2 Перебор сочетаний с ограничениями

Леммы 2 и 3 позволяют установить минимальную степень вершины в Mk -РОТР и Mk -ВОТР $H=(W, \beta)$ графа $G=(V, \alpha)$, при условии, если в G нет висячих вершин. Если отказоустойчивая реализация построена путём добавления вершин и рёбер, то $V \subseteq W \wedge \alpha \subseteq \beta$. Рассмотрим дополнительные рёбра $\gamma = \beta \setminus \alpha$. По лемме 3 если в графе G минимальная степень вершин $d_{min} > 0$, то в его Mk -РОТР и Mk -ВОТР, не должно быть вершин степени $d < d_{min} + k$, поэтому в γ будет присутствовать не менее $d_{min} + k - d(u)$ рёбер, инцидентных $u \in W$. Получили необходимое условие соответствия определению Mk -РОТР и Mk -ВОТР.

Вследствие этого для каждой вершины u можно сформировать множество $\eta_u = \{\{u, v\} \in W \times W \mid \{u, v\} \notin \alpha\}$ и далее при рассмотрении графа $H=(W, \beta)$ проверять, что $|\eta_u \cap \beta| \geq d_{min} + k - d(u)$. Если это условие не выполняется, то граф не является ни Mk -ВОТР, ни Mk -РОТР.

Очевидно, что данное необходимое условие будет давать отрицательный результат тем чаще, чем больше вершин будут иметь степень,

меньше минимально возможной, поэтому в данном случае наиболее эффективно будет применение его по отношению к регулярным графам, в том числе и к гиперкубам.

Однако, можно ещё более увеличить эффективность путём построения только таких графов, которые изначально будут соответствовать данному условию. Это можно сделать с помощью алгоритма 4.

Алгоритм 4 – Построение сочетаний с ограничениями

Вход:

1. Множество элементов $A = \{a_1, \dots, a_n\}, a_1 < \dots < a_n$.
2. m – необходимая длина выходного вектора.
3. Вектор упорядоченных пар $((A_1, m_1), \dots, (A_k, m_k))$ – множества и необходимое количество элементов из них в выходном векторе, $A_k = \{a_1, \dots, a_n\}, m_k = m, \forall i \in \{1, \dots, k\} A_i \subseteq \{a_1, \dots, a_n\} \wedge m_i \geq 0$.

Выход: множество R всех возможных множеств $B = \{b_1, \dots, b_m\} \subseteq \{a_1, \dots, a_n\}$ таких, что $(\forall i \in \{1, \dots, k\}) (|A_i \cap \{b_1, \dots, b_m\}| \geq n_i)$.

- 1 Если $m_1 > m$, то ни одного графа построить невозможно, результат $\emptyset$.
- 2 $R = \emptyset$.
- 3 Для каждого возможного подмножества множества A_1 из m_1 элементов $B_1 \in \{\{b_1, \dots, b_{m_1}\} \subseteq A_1\}$ выполнить:
 - 3.1 Если $k = 1$ то $S = \{B_1\}$.
 - 3.2 Иначе:
 - 3.2.1 $(\forall i \in \{1, \dots, k\}) m'_i = \max(m_i - |A_i \cap B_1|, 0)$ – количество элементов, которые остаётся добавить из группы A_i с учётом, что $B_1 \subseteq B$.
 - 3.2.2 $b_{\max} = \max_{b \in B_1} b$ – наибольший из используемых элементов множества B_1 .
 - 3.2.3 $D = \{x \in B_1 | x \leq b_{\max}\}$ – множество запрещённых для дальнейшего использования элементов.
 - 3.2.4 $\forall i \in \{1, \dots, k\} A'_i = A_i \setminus D$ – множества A_i без запрещённых элементов.
 - 3.2.5 $A' = A \setminus D$ – множество A без запрещённых элементов.

3.2.6 $R' = \left\{ B' = \{b_{m_1+1}, \dots, b_m\} \subseteq A' \mid (\forall i \in \{2, \dots, k\}) |A'_i \cap B'| \geq m'_i \right\}$ – рекурсивно

сформировать множество R' всех таких множеств B' , которые будут содержать не менее n'_i элементов из множества A'_i для $i \in \{2, \dots, k\}$.

3.2.7 $S = \{B = B_1 \cup B' \mid B' \in R'\}$ – множество, содержащее подмножества A , полученные объединением каждого элемента B' из R' и B_1 . По построению $(\forall B \in S)(\forall i \in \{1, \dots, k\}) (|A_i \cap B| \geq m_i)$.

3.3 $R = R \cup S$ – добавить построенные множества в результат.

Для того, чтобы показать корректность алгоритма, докажем лемму 4.

Лемма 4. Применение алгоритма 4 позволяет построить все сочетания, удовлетворяющие заданному условию, причём каждый из них будет получен ровно один раз.

Доказательство. В доказательстве приняты те же обозначения, что и в алгоритме 4.

С одной стороны каждое множество может быть построено единственным способом данным алгоритмом. Пусть было построено 2 одинаковых множества $B = B^*$ двумя разными путями $B_1, \dots, B_k$ и $B_1^*, \dots, B_k^*$.

В алгоритме рекурсивно выбираются подмножества $B_i \subseteq A_i$. То, из каких элементов может состоять B_i , зависит только от $B_1, \dots, B_{i-1}$. Это происходит на шаге 3 при переборе всех различных подмножеств, поэтому случай разных построений, когда $(\forall i \in \{1, \dots, k\})(B_i^* = B_i)$ невозможен. Следовательно, если одно и то же множество было построено двумя различными путями, то $(\exists t)(B_t^* \neq B_t \wedge (\forall i \in \{1, \dots, t-1\})(B_i = B_i^*))$. Выберем самый малый элемент b из множества элементов, которые присутствуют только в одном из множеств B_t или B_t^* .

Пусть без ограничения общности $b \in B_t$ (в противном случае их можно переобозначить). Так как $B_1 \cup \dots \cup B_k = B_1^* \cup \dots \cup B_k^*$ и $(\forall i \in \{1, \dots, t-1\})(b \notin B_i^* = B_i) \wedge b \notin B_t^*$, то $(\exists s \in \{t+1, \dots, k\})(b \in B_s)$.

Количество элементов в B_t и B_t^* равно, так как зависит оно только от предыдущих выбранных множеств, а $B_1 = B_1^*, \dots, B_{t-1} = B_{t-1}^*$, поэтому существует как минимум 2 элемента b и b^* в B_t и B_t^* , которые не являются общими, но поскольку b – минимальный из них, то $b < b^* \leq \max_{e \in B_t^*} e$,

поэтому он был выбран запрещённым элементом на шаге 3.2.3 и на шаге 3.2.4 был удалён из списка элементов, которые будут использованы на более глубоких шагах рекурсии, поэтому $(\forall i \in \{t+1, \dots, k\})(b \notin B_i^*)$, следовательно, $(\forall i \in \{1, \dots, k\})(b \notin B_i^*)$, а это значит, что во втором множестве B^* нет элемента b , который есть в B , что противоречит их равенству.

С другой стороны каждое удовлетворяющее условию $(\forall i \in \{1, \dots, k\})(|A_i \cap \{b_1, \dots, b_m\}| \geq n_i)$ множество вершин может быть построено. Пусть дано множество B и $B_1^* = B \cap A_1, \dots, B_k^* = B \cap A_k$ такие, что $|B_1^*| \geq m_1, \dots, |B_k^*| \geq m_k$.

Рассмотрим произвольное множество $B = \{b_1, \dots, b_m\} \subseteq \{a_1, \dots, a_n\}$ таких, что $(\forall i \in \{1, \dots, k\})(|A_i \cap \{b_1, \dots, b_m\}| \geq n_i)$ и укажем для него цепочку построения через множества $B_i \subseteq A_i$ такие, что $|A_i \cap (B_1 \cup B_2 \cup \dots \cup B_i)| \geq m_i$.

Пусть B_i состоит из m_i' минимальных элементов из $B \cap A_i'$, где

- $m_i' = \max(m_i - |A_i \cap (B_1 \cup \dots \cup B_{i-1})|, 0)$ – количество элементов из A_i' , которых не хватает в $B_1 \cup \dots \cup B_{i-1}$ чтобы удовлетворять условию $|B_1 \cup \dots \cup B_{i-1}| = m_i$ (при этом $m_1' = m_1$);

- $A'_i = A_i \setminus (B_1 \cup \dots \cup B_{i-1})$ – элементы из A_i , которые не состоят в $B_1, \dots, B_{i-1}$ (при этом $A'_1 = A_1$).

На каждом шаге m'_i элементов из A'_i найдётся, так как в противном случае в B было бы меньше, чем m_i элементов, что противоречит определению B , то есть такой путь найдётся.

В алгоритме на i -той глубине рекурсии будет выбрано в том числе и множество B_i , состоящее из m'_i элементов из A'_i . Покажем это индукцией по i :

$i=1$: Алгоритм работает с множествами A_i и значениями m_i , которые пришли алгоритму на вход. На шаге 3 перебираются все возможные m_1 -элементные подмножества множества A_1 , а значит, при переборе появится и множество B_1 , так как $m'_1 = m_1$ и $A'_1 = A_1$.

$i>1$: Пусть ранее алгоритмом были выбраны множества $B_1, \dots, B_{i-1}$, покажем, что B_i будет выбрана на i глубине рекурсии.

На шаге 3.2.1 пересчитывается количество элементов m_j из A_j , которых не хватает для выполнения условий, наложенных на количество элементов подмножеств в выходных множествах, то есть m_1 алгоритма на i -той глубине рекурсии будет совпадать с m'_i .

На шаге 3.2.2 формируется множество D , элементы которого будут исключены из дальнейшего рассмотрения алгоритмом. Оно состоит из тех элементов, которые меньше максимального элемента из множества B_1 на глубине рекурсии j алгоритма, то есть из B_j . То есть множество B_1 алгоритма может быть меньше, чем $B_i = A_i \setminus (B_1 \cup \dots \cup B_{i-1})$ и отличается только в тех элементах, которые были меньше максимального элемента из какого-либо B_j ($j < i$), при этом в B_j не попавший. Если бы в B_i появился такой элемент, то это противоречило бы выбору B_j из B , так как именно он

попал бы в B_j вместо максимального из B_j . Остальные элементы из A_i могут находиться в B_i , то есть множество A_1 алгоритма содержит все элементы, из которых может состоять B_i . Так как на шаге 3 перебираются все подмножества, то B_i также будет при переборе.

В итоге получается, что каждое множество будет построено алгоритмом, при этом только одним способом.

■

Чтобы применять данный алгоритм для построения Mt-POtP, необходимо занумеровать рёбра, которые можно добавить в граф $G=(V, \alpha)$, где $V=\{v_1, \dots, v_n\}$. Так получаются элементы $A=\{a_1, \dots, a_p\}=(V \times V) \setminus \alpha$. Далее, для того, чтобы выходной граф соответствовал условиям леммы 3, нужно найти минимальную степень d_{min} вершины графа. Для каждой вершины сформировать множества $A_i=\{\{v, u\} \in A \mid v=v_i\}$ и $m_i=\max(0, (d_{min}+t)-d(v_i))$ – количество рёбер, которые нужно добавить к вершине v_i для того, чтобы её степень была не меньше, чем $d_{min}+t$. В данном случае $k=n+1$. Последнее множество $A_k=A$, $m_k=m$ – текущее рассматриваемое количество дополнительных рёбер. Данное ограничение нужно для того, чтобы в выходном векторе было не меньше m элементов. Также пункт 1 контролирует то, что в выходном векторе будет не более m элементов.

Для решения задачи Mt-VOtP можно по аналогии, используя лемму 2, получить $A_1, \dots, A_n, A_{n+1}, \dots, A_{n+t}, A_{n+t+1}$ и $m_1, \dots, m_n, m_{n+1}, \dots, m_{n+t}, m_{n+t+1}$ чисел, только рассматривая $A=(V^* \times V^*) \setminus \alpha$, где $V^*=V \cup O_t$ при построении. Также можно задействовать n -вершинные подграфы графа $G^*=(V^*, \alpha)$ (графы вида $H-v_1-\dots-v_r$ для некоторых вершин $v_1, \dots, v_r$ будем называть подграфами). Для каждого из таких подграфов $H=G^*-v_{i_1}-\dots-v_{i_t}=(U_H, \alpha_H)$

можно сформировать $A_H = (U_H \times U_H) \setminus \alpha_H$ и $m_H = \max(0, |\alpha| - |\alpha_H|)$. Если группа (A_H, m_H) будет участвовать в переборе, то в каждом n -вершинном подграфе графа G^* не меньше рёбер, чем в G .

1.3 Построение Mk-ВОТР методом объединения изоморфных графов

Для построения Mk-ВОТР можно использовать другой подход, основанный на объединении графов, изоморфных G . Пусть G_k^* является Mk-ВОТР графа G , тогда при удалении произвольных k вершин из G_k^* получится граф $G_{v_{i_1}, \dots, v_{i_k}} = G_k^* - v_{i_1} - \dots - v_{i_k}$, в который будет вкладываться G . Пусть само вложение будет $\psi_{v_{i_1}, \dots, v_{i_k}}$, тогда $\psi_{v_{i_1}, \dots, v_{i_k}}(G) \subseteq G_{v_{i_1}, \dots, v_{i_k}} \subseteq G_k^*$. Объединяя такие графы получим $\bigcup_{u_1, u_2, \dots, u_k \in V_k} \psi_{u_1, \dots, u_k}(G) \subseteq G_k^*$. Ввиду минимальности G_k^* получим $\bigcup_{u_1, u_2, \dots, u_k \in V_k} \psi_{u_1, \dots, u_k}(G) = G_k^*$. Если это не так, то будет существовать ребро e , которое не входит ни в один из графов, участвующих в объединении, а значит, $G_k^* - e$ будет являться отказоустойчивой реализацией того же типа с меньшим числом рёбер, что противоречит минимальности G_k^* .

Используя подход с объединениями изоморфных графов можно избавиться от необходимости проверять полученные графы на соответствие Mk-ВОТР по определению. Таким образом, остаётся задача исключения изоморфных копий.

Общий алгоритм подхода:

Алгоритм 5 – Построение Mk-ВОТР методом объединения изоморфных графов

Вход:

1. $G=(V, \alpha)$ со множеством вершин $V=\{1,2, \dots, n\}$.
 2. k – положительное целое число.
 3. m_{start} – оценка сверху количества дополнительных рёбер в отказоустойчивой реализации.
-

1. $V^*=\{1, \dots, n-1, n, \dots, n+k\}$ – множество вершин Mk-ВОТР.
 2. $\{V_1, \dots, V_{C_{n+k}^n}\}:=\{X\subseteq V^* | |X|=n\}$ – все n -элементные подмножества множества V^* .
 3. $m:=m_{start}$.
 4. $E=\emptyset$.
 5. Для каждого различного набора графов $G_1=(V_1, \alpha_1), G_2=(V_2, \alpha_2), \dots, G_{C_{n+k}^n}=(V_{C_{n+k}^n}, \alpha_{C_{n+k}^n})$ таких, что $G_1\cong G, \dots, G_{C_{n+k}^n}\cong G$:
 1. $H=G_1\cup\dots\cup G_{C_{n+k}^n}$.
 2. Если количество рёбер в H меньше m , то:
 1. m = количество рёбер в H .
 2. $E=\emptyset$.
 3. Если количество рёбер в H равно m и $\nexists(F\in E)(F\cong H)$, то:
 $E=E\cup\{H\}$.
 6. Результатом является множество E , которое содержит все Mk-ВОТР графы с n вершинами.
-

В пункте 2 строятся n элементные подмножества $n+k$ элементного множества. Данное действие можно выполнить, перебрав все возможные сочетания из $n+k$ по n . Алгоритм для такого перебора широко известен.

Пятый пункт алгоритма описывает основные шаги, которые занимают большую часть его работы:

1. Перебор всех возможных сочетаний изоморфных графов, у каждого из которых заранее известно множество вершин.

2. Построение объединений найденных наборов графов.
3. Выбор из полученных графов тех, у которых количество рёбер минимально.
4. Отсечение изоморфных копий графа в списке.

Опишем первый пункт подробнее. Изоморфные графы можно получить путём применения изоморфного отображения ϕ , которое по сути является перестановкой имён вершин. Пусть даны граф $G=(V, \alpha)$ и биекция двух n -элементных множеств $\phi: V \rightarrow U$, тогда изоморфный граф можно получить следующим образом: $\phi(G)=(U, \{(\phi(u), \phi(v)) \mid (u, v) \in \alpha\})$. При этом $\phi = \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ u_1 & u_2 & \dots & u_n \end{pmatrix}$, поэтому все такие отображения можно перебрать, зафиксировав верхнюю строку и переставляя элементы нижней или наоборот, то есть данный перебор сводится к перебору перестановок элементов одного из множеств.

Среди различных изоморфных преобразований могут встречаться такие, что будут давать один и тот же результат. Пусть даны два изоморфизма $\phi, \psi: \phi \neq \psi$, будем называть их *равнозначными*, если $\phi(G) = \psi(G)$.

Пусть $a \in \text{Aut}(G)$ является автоморфизмом графа G , тогда для любого изоморфизма $\phi: V \rightarrow U$ преобразование $a \circ \phi$ равнозначно ему, так как $(a \circ \phi)(G) = \phi(a(G)) = \phi(G)$.

Для уменьшения перебора необходимо из всех равнозначных преобразований оставить одно. Решим задачу методом канонических представителей [63]. В качестве канонического представителя класса равнозначных перестановок выберем лексикографически наименьшее из них. Порядок будем определять нижней частью перестановки при упорядоченной верхней части по возрастанию.

Если известно множество всех автоморфизмов $\text{Aut}(G)$ и изоморфизм ϕ , то множество равнозначных преобразований можно получить домножением

слева на все возможные автоморфизмы графа G : $H = \{a \circ \phi \mid a \in \text{Aut}(G)\}$.
 Других подобных преобразований нет, так как для любого ψ : $\psi(G) = F$
 $F = \psi(G) = \psi(\text{id}(G))$, то есть $\psi = \text{id} \circ \psi$, а $\text{id} \in \text{Aut}(G)$, поэтому $\psi \in H$.

Рассмотрим пример действия автоморфизма $a = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 4 & 2 & 6 & 5 \end{pmatrix}$ на произвольный изоморфизм $\phi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{pmatrix}$ слева:

$a \circ \phi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ x_1 & x_3 & x_4 & x_2 & x_6 & x_5 \end{pmatrix}$. Произошла фиксированная перестановка элементов нижней части изоморфизма ϕ , задаваемая автоморфизмом. Сравним лексикографический порядок нижних частей изоморфизмов ϕ и $a \circ \phi$: $(x_1, x_2, x_3, x_4, x_5, x_6)$ и $(x_1, x_3, x_4, x_2, x_6, x_5)$. Так как изоморфизм является биективным преобразованием, то все образы различны, а значит, следует найти первый в последовательностях отличающийся элемент. В данном случае первый элемент в обеих последовательностях одинаков и равен x_1 , а второй – различен, поэтому нужно сравнить x_2 и x_3 . Если $x_2 > x_3$, то $a \circ \phi < \phi$, следовательно, изоморфизм ϕ не является лексикографически минимальным, следовательно, он не канонический.

Если рассмотреть в таком ключе все существующие автоморфизмы, то можно получить ряд условий вида $x_i < x_j$, выполнение которых будет определять канонического представителя. При этом $i < j$, так как это первый номер элемента, который отличается в последовательностях, а значит, элементы с меньшими номерами были равны и уже задействованы для элементов с меньшими номерами, поэтому сравнение может идти только элементами, номера которых больше i .

Отношение меньше является отношением порядка, поэтому при наличии правил $x_i < x_j$ и $x_j < x_k$ правило $x_i < x_k$ будет выполняться из-за

свойства транзитивности, поэтому такая проверка будет избыточна. Это даёт возможность уменьшить количество проверок.

Для проверки правила достаточно знать 2 элемента изоморфизма, поэтому сам изоморфизм можно строить постепенно. Для этого можно использовать обход префиксного дерева перестановок n -элементного множества образов изоморфизма. Такое дерево позволит постепенно строить изоморфизм и проверять правила сразу при появлении необходимых элементов. Если правило не выполнено для какого-то префикса изоморфизма, то оно не будет выполнено и для всех его потомков в дереве, так как в потомках данный префикс присутствует, поэтому такие узлы дерева можно не посещать. То есть происходит исключение из обхода всей ветви дерева.

Префиксное дерево в данном случае можно определить следующим образом:

1. Метка корня дерева – пустая последовательность.
2. Листы дерева помечены перестановками.
3. Узел с меткой $(p_1, \dots, p_{i-1}, p_i)$, где $i > 0$ будет дочерним по отношению к узлу с меткой $(p_1, \dots, p_{i-1})$.

Также при использовании такого дерева возможно постепенное построение объединения графов. При переходе к дочернему узлу в изоморфизме определится образ ещё одной вершины, поэтому появятся новые известные рёбра, которые точно будут присутствовать в объединении графов. Далее следует посчитать количество получившихся на данный момент рёбер. Если их стало больше, чем верхняя граница m в алгоритме, то ни один дочерний узел не содержит решение задачи.

Рассмотрим пример объединения графа G и изоморфного ему графа H с известной частью изоморфизма $\phi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 3 & ? & ? & ? \end{pmatrix}$. Неизвестная часть изоморфизма обозначена знаком вопроса. На рисунке 4 показано объединение данных графов. Сами графы представлены в виде матриц смежности.

G	1	2	3	4	5	6		H	1	3	4	5	6	7		$G \cup H$	1	2	3	4	5	6	7
1	0	1	1	0	0	0		1	?	?	?	?	?	?		1	?	1	1	?	?	?	?
2	1	0	0	1	1	1		3	?	0	1	?	0	?		2	1	?	?	1	1	1	?
3	1	0	0	1	0	0		4	?	1	0	?	1	?		3	1	?	0	1	?	0	?
4	0	1	1	0	0	1		5	?	?	?	?	?	?		4	?	1	1	0	?	1	?
5	0	1	0	0	0	0		6	?	0	1	?	0	?		5	?	1	?	?	?	?	?
6	0	1	0	1	0	0		7	?	?	?	?	?	?		6	?	1	0	1	?	0	?
																7	?	?	?	?	?	?	?

Рисунок 4 – Объединение графа G с изоморфным ему графом H при известных трёх элементах изоморфизма

Пусть в данном изоморфизме известно на одну вершину больше, для примера возьмём $\phi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 3 & 7 & ? & ? \end{pmatrix}$. На рисунке 5 показано объединение тех же графов с четырьмя известными элементами в изоморфизме.

G	1	2	3	4	5	6		H	1	3	4	5	6	7		$G \cup H$	1	2	3	4	5	6	7
1	0	1	1	0	0	0		1	?	?	?	?	?	?		1	?	1	1	?	?	?	?
2	1	0	0	1	1	1		3	?	0	1	?	0	1		2	1	0	0	1	1	1	0
3	1	0	0	1	0	0		4	?	1	0	?	1	0		3	1	0	0	1	?	0	1
4	0	1	1	0	0	1		5	?	?	?	?	?	?		4	?	1	1	0	?	1	0
5	0	1	0	0	0	0		6	?	0	1	?	0	1		5	?	1	?	?	?	?	?
6	0	1	0	1	0	0		7	?	1	0	?	1	0		6	?	1	0	1	?	0	1
																7	?	0	1	0	?	1	0

Рисунок 5 – Объединение графа G с изоморфным ему графом H при известных четырёх элементах изоморфизма

На рисунках 4 и 5 показано, какие элементы матрицы смежности объединения графов будут равны единице или нулю, а какие неизвестны. Так как вершина 2 не является вершиной графа H , все инцидентные ей рёбра в объединении графов сохраняются. Некоторые рёбра неизвестны.

При объединении графов $F_1 = G_1 \cup G_2 \cup \dots \cup G_N$ и $F_2 = H_1 \cup H_2 \cup \dots \cup H_N$, $\forall (i \in \{1, \dots, N\}) (G_i \cong G \wedge H_i \cong G)$ может получиться

так, что $F_1 \cong F_2$ или $F_1 = F_2$. Исключение одинаковых графов позволяет уменьшить количество полученных одинаковых отказоустойчивых реализаций. Для уменьшения количества изоморфных графов можно зафиксировать один из них, тогда для одного из множеств вершин не нужно будет перебирать изоморфные графы.

1.4 Параллельные вычисления

Большинство современной вычислительной техники имеет несколько вычислительных ядер, способных работать параллельно с единой системой памяти. Также есть кластеры и суперкомпьютеры. Задачи, рассматриваемые в данной работе являются переборными и к ним применимы стандартные способы разбиения задачи на подзадачи и решения их независимо. Далее будем предполагать, что распараллеливание происходит на уровне процессов в операционных системах.

Каждая из задач либо является обходом дерева, либо сводится к построению всех возможных векторов заданного вида $(a_1, a_2, \dots, a_n)$. В случае с векторами можно также построить префиксное дерево.

Далее на определённом уровне данного дерева происходит распределение ветвей между процессами, как это показано на рисунке 6 для 3 процессов.

Такой способ можно реализовать без любых коммуникаций между процессами. Для этого верхняя часть дерева обходится всеми процессами, а каждая оставшаяся ветвь обрабатывается закреплённым за ней процессом. При распределении можно заводить счётчик и номер процесса. При посещении вершины на уровне, на котором происходит распределение нагрузки, её дочерние элементы посещаются только если счётчик равен номеру процесса. Счётчик изменяется при каждом посещении вершины заданного уровня.

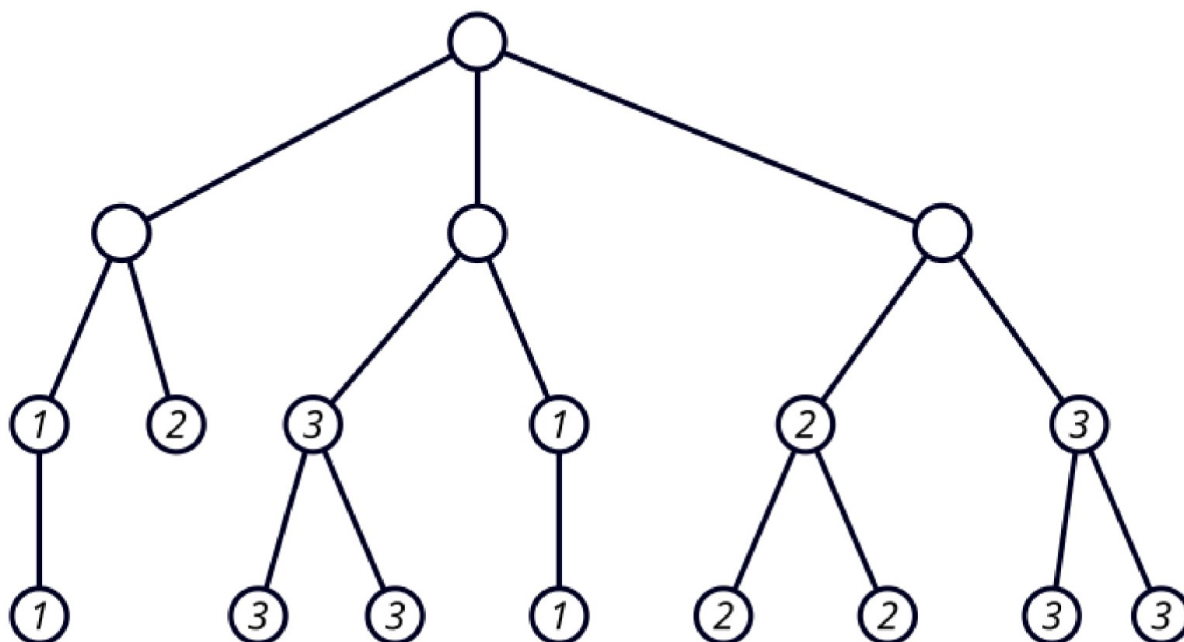


Рисунок 6 – Пример распределения ветвей при распараллеливании
вычислений

1.5 Результаты вычислительных экспериментов

В данной главе представлены результаты вычислительных экспериментов, выполненных с помощью программ «FTConstructor» и «extvu». Полученные данные позволили разработать модели построения отказоустойчивых реализаций гиперкубов и аналитически обосновать их корректность. Подробное описание моделей и теоретических выводов приводится в последующих главах.

С помощью реализаций алгоритмов 1 и 4 были построены все Mk-РОТР 3-куба и M1-РОТР 4-куба. M1-РОТР 16-вершинного и 8-вершинного гиперкуба были сгенерированы той же программой в [53], и приведены в данной работе для полноты картины.

Для того, чтобы описать результаты по полученным рёберным отказоустойчивым реализациям дадим следующее определение. Рассмотрим два натуральных числа $N, i: i \leq N$. Пусть $Q_N^i = \{V, \alpha\}$, где $V = \{0, 1\}^N$ и

$\alpha = \{\{u, v\} | u, v \in V \wedge h(u, v) = i\}$, а $h(u, v)$ – расстояние Хэмминга между метками вершин u и v . При этом графы Q_N^i и Q_N^j при $i \neq j$ не имеют общих рёбер. Обозначим через $Q_N^{i_1, \dots, i_k} = Q_N^{i_1} \cup Q_N^{i_2} \cup \dots \cup Q_N^{i_k}$. Данные графы будем называть Q -графами.

На рисунке 7 показаны единственные М1-РОТР (а), М2-РОТР (б), М3-РОТР (в) и М4-РОТР (г) графа Q_3 .

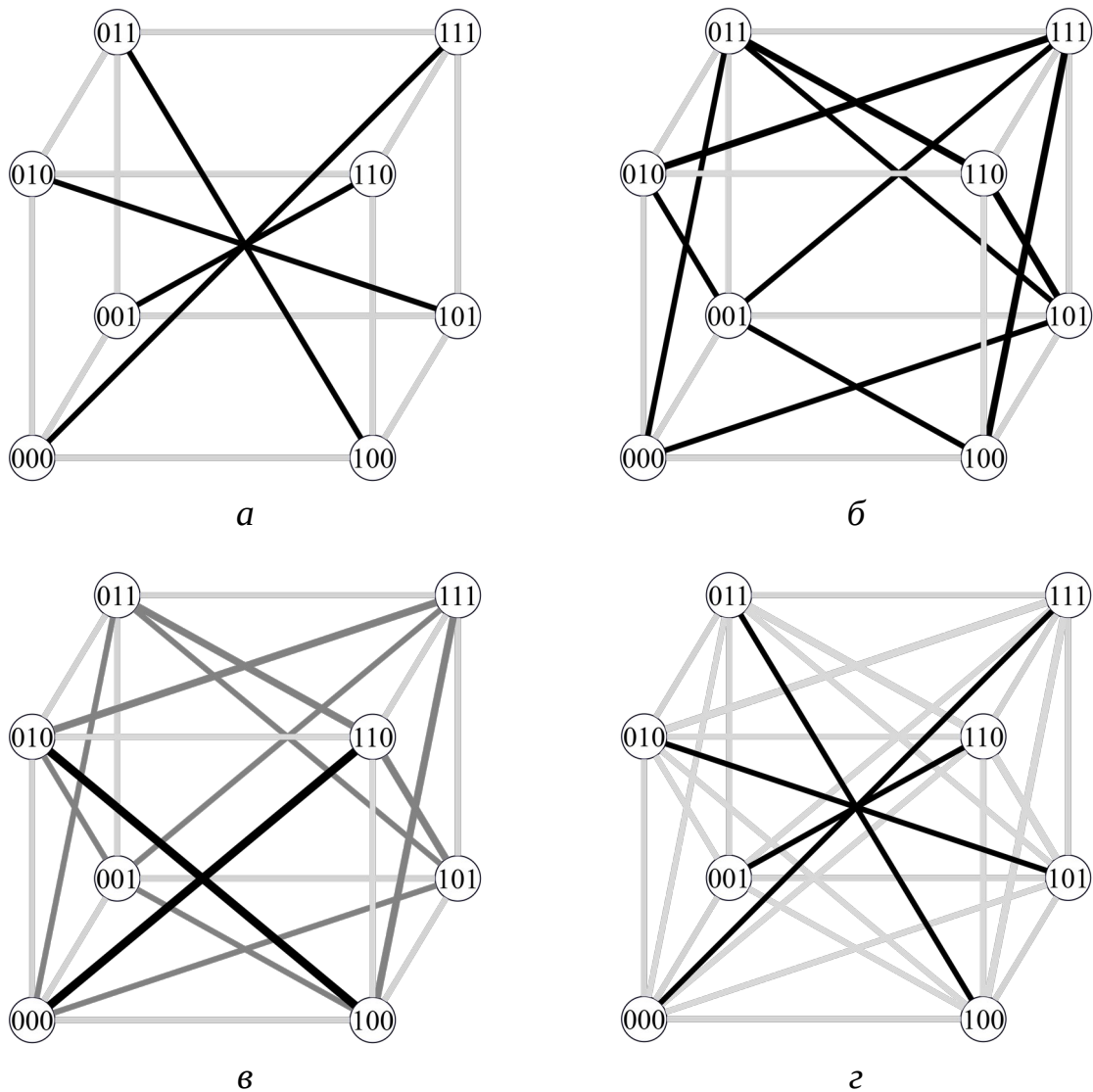


Рисунок 7 – М1-РОТР, М2-РОТР, М3-РОТР, М4-РОТР 3-куба

Стоит отметить, что:

1. $Q_3^{1,3}$ является М1-РОТР.

2. $Q_3^{1,2}$ является МЗ-РОТР.
3. М2-РОТР является частью $Q_3^{1,2}$, поэтому может быть построено с помощью приведения $Q_3^{1,2}$ как 2-РОТР.
4. $Q_3^{1,2,3}$ является М4-РОТР.

Если граф G^* является k -РОТР графа G , то G^* является t -РОТР для $t < k$, поэтому среди частей графа G^* можно найти Ht -РОТР и выбрать из них те, что имеют минимальное количество рёбер. Данную операцию и будем называть *приведением отказоустойчивой реализации*. Для её выполнения можно использовать перебор неизоморфных частей графа.

Схема получения всех Mk -РОТР 3-куба показана на рисунке 8: пунктирной стрелкой показан путь получения отказоустойчивых реализаций путём операции приведения, стрелкой с непрерывной линией – проверка на отказоустойчивость соответствующих Q -графов.

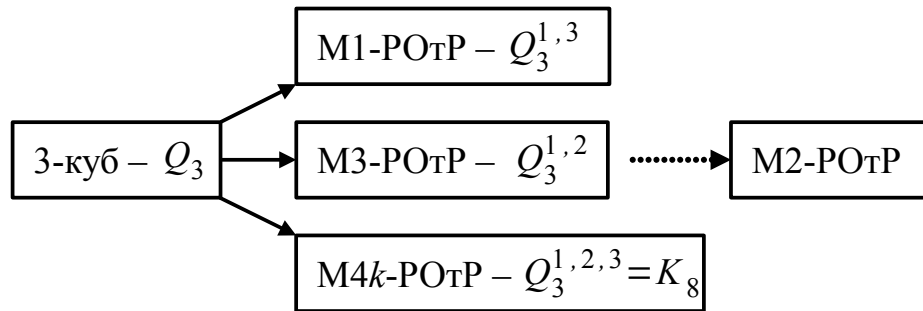


Рисунок 8 – Схема построения Mk -РОТР 3-куба

Все графы вида $Q_3^{1,i_1,\dots,i_t}$ являются Mk -РОТР графа Q_3 , при этом М2-РОТР является частью МЗ-РОТР. То есть по данной схеме можно построить все Mk -РОТР 3-куба. Отсюда возникает следующее логичное обобщающее предположение:

Предположение 1 (о регулярных Mk -РОТР гиперкубов). Графы вида $Q_N^{1,i_1,\dots,i_t}$ являются Mk -РОТР для $k = C_N^{i_1} + \dots + C_N^{i_t}$.

Для его проверки был проведён аналогичный вычислительный эксперимент для 4-куба, результаты можно видеть на рисунке 9. Были проверены графы:

1. $Q_4^{1,3}$ – M4-РОТР 4-куба.
2. $Q_4^{1,3,4}$ – не является M5-РОТР 4-куба.
3. $Q_4^{1,2}$ – M6-РОТР 4-куба.
4. $Q_4^{1,2,4}$ – M7-РОТР 4-куба.
5. $Q_4^{1,2,3}$ – M10-РОТР 4-куба.
6. $Q_4^{1,2,3,4} = K_{16}$ – M11-РОТР 4-куба.
7. $Q_4^{1,3}$ – M4-РОТР 4-куба.

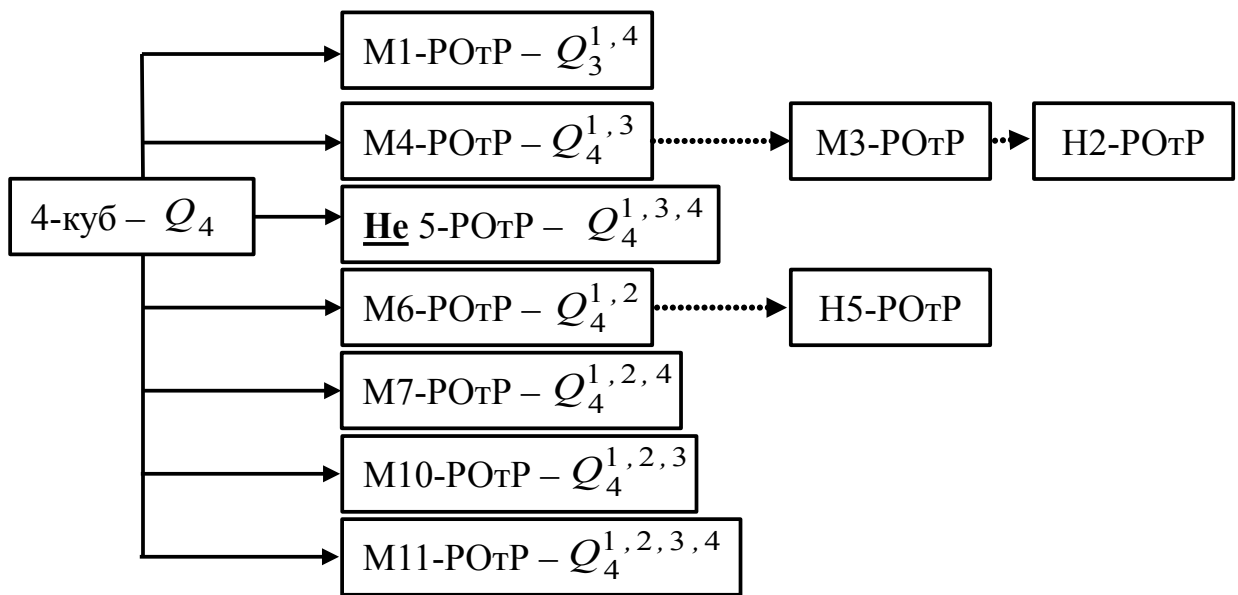


Рисунок 9 – Схема построения Mk-РОТР и Nk-РОТР 4-куба

Граф $Q_4^{1,3,4}$ не является M5-РОТР 4-куба. Это значит, что предположение о том, что все такие графы $Q_N^{1,i_1,\dots,i_t}$ являются Mk-РОТР гиперкуба для некоторого k не подтвердилось. Однако, хоть само предположение не является полностью верным, все проверенные графы, за

исключением одного, оказались минимальными отказоустойчивыми реализациями гиперкубов, и это значит, что необходимо данное предположение уточнять.

Так как граф $Q_4^{1,3,4}$ не является М5-РОТР, была предпринята попытка получить М5-РОТР путём приведения М6-РОТР $Q_4^{1,2}$ 4-куба. Были построены все части графа $Q_4^{1,2}$, которые являются 5-РОТР, среди которых было найдено одно Н5-РОТР с вектором степеней $(10^2, 9^{14})$. Среди всех найденных 5-РОТР у него оказалось минимальное число рёбер – на одно ребро больше, чем в $Q_4^{1,3,4}$. Является ли этот граф минимальным – отдельный сложный вопрос. Его матрица смежности приведена на рисунке 10.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
2	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1
3	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	1
4	0	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1
5	0	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0
6	0	1	1	1	1	0	1	1	0	0	1	1	0	1	0	0
7	0	0	0	1	1	1	0	1	1	1	1	1	1	0	0	0
8	1	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0
9	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1
11	1	1	0	0	1	1	1	1	0	0	0	1	1	0	1	0
12	1	1	0	1	0	1	1	0	1	0	1	0	1	1	0	1
13	1	1	0	1	1	0	1	0	0	1	1	1	0	0	0	1
14	1	1	1	0	0	1	0	1	1	0	0	1	0	0	1	1
15	1	1	1	0	1	0	0	1	0	1	1	0	0	1	0	1
16	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	0

Рисунок 10 – Матрица смежности Н5-РОТР 4-куба

Среди частей М4-РОТР $Q_4^{1,3}$ был найден регулярный граф степени 7, который является 3-РОТР 4-куба. Так как у 4-куба минимальная степень вершины равна 4, то у М3-РОТР по лемме 3 минимальная степень вершины равна 7, поэтому найденное 3-РОТР 4-куба является минимальным. Среди

частей данной отказоустойчивой реализации нашлось Н2-РОТР 4-куба с вектором степеней $(7^8, 6^8)$.

Данные графы показаны на рисунке 11. Под пунктом *а* показано Н2-РОТР, под *б* – М3-РОТР и под *в* – М4-РОТР, равное $K_{8,8}$ – полному двудольному 16-вершинному графу с равными долями.

Для построения частей заданного графа использовался алгоритм 2 для его дополнения.

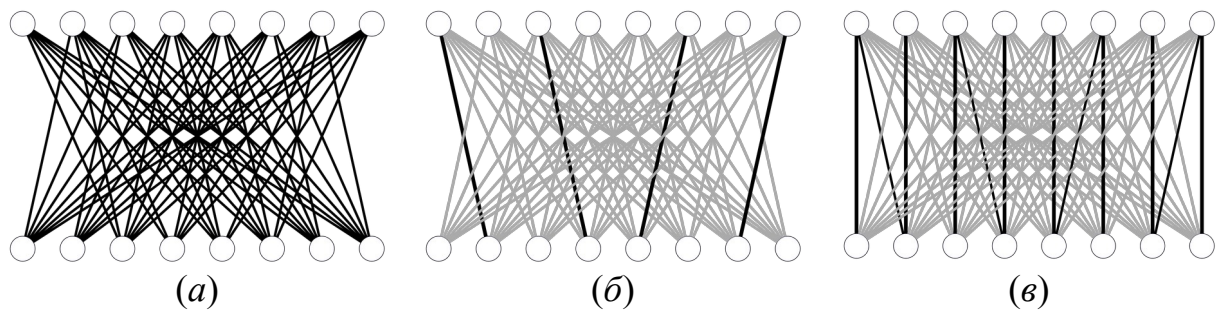


Рисунок 11 – М4-РОТР, М3-РОТР, Н2-РОТР 4-куба

Также с использованием алгоритмов 1 и 2 были построены все М1-ВОТР, М2-ВОТР и М3-ВОТР 3-куба. М1-ВОТР оказалось единственным и изоморфным тому графу, что может быть получен по схеме из теоремы 3. А вот различных неизоморфных М2-ВОТР оказалось 2, они изображены на рисунке 12.

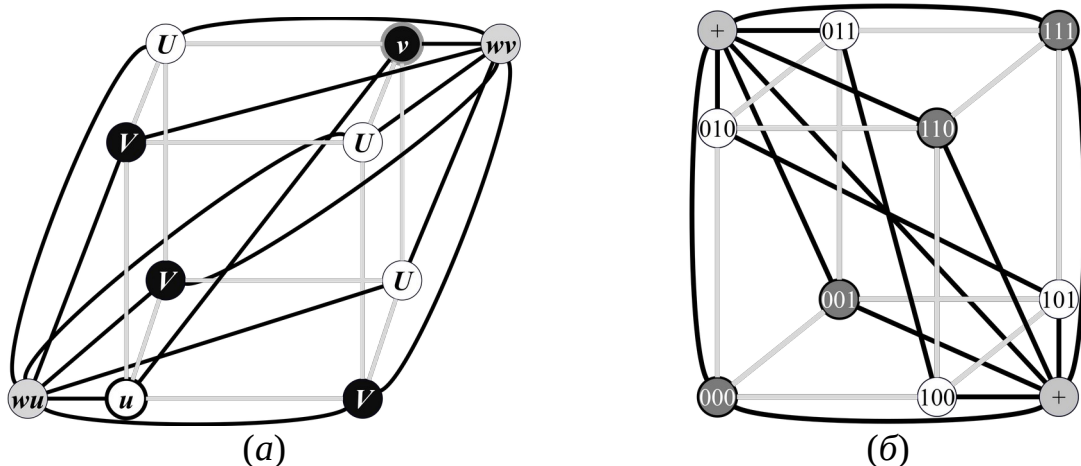


Рисунок 12 – М2-ВОТР 3-куба

На рисунке 13 показаны МЗ-ВОТР 3-куба. Подобные вершины на изображениях имеют одинаковые метки.

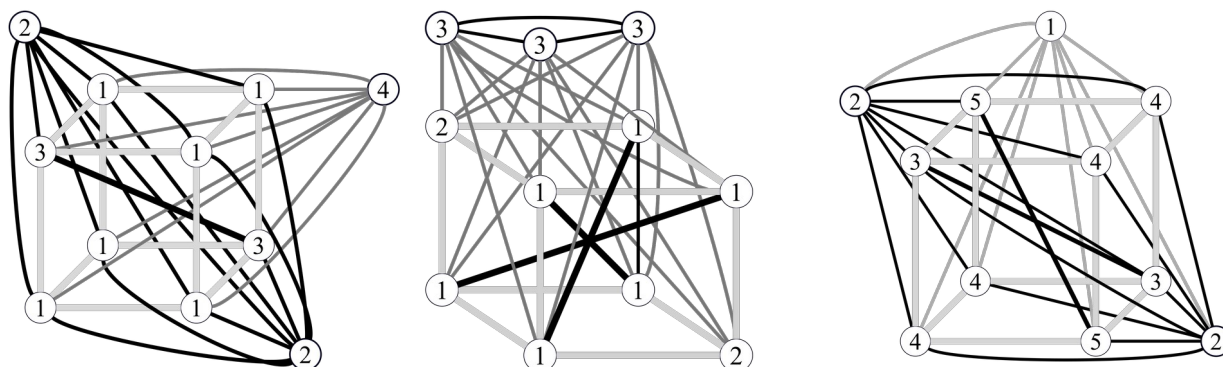


Рисунок 13 – МЗ-ВОТР 3-куба

С помощью реализации алгоритма 5 был проведён вычислительный эксперимент, который показал, что для 4-куба существует единственное с точностью до изоморфизма М1-ВОТР. Данный результат совместно с результатами по М1-ВОТР 3-куба позволил сделать предположение о структуре М1-ВОТР произвольного гиперкуба, которое в дальнейшем было подтверждено аналитически. Кроме того, данный результат охватил более широкий класс графов, что будет описано в следующей главе.

Глава 2. Моделирование вершинной отказоустойчивости гиперкубов

В данной главе будут представлены результаты, полученные путём обобщения итогов вычислительных экспериментов, которые касаются вершинных отказоустойчивых реализаций N -кубов при $N \geq 3$, а именно:

1. Представлены найденные схемы построения 1-ВОТР и 2-ВОТР гиперкубов, применимые ко всем двудольным графам, за исключением полных двудольных.
2. Доказано, что количество дополнительных рёбер в построенных по представленным схемам 1-ВОТР и 2-ВОТР меньше, чем количество рёбер в Т1-ВОТР на 1 и Т2-ВОТР на 2 соответственно, то есть они более оптимальны, чем известные схемы.
3. Найдено и аналитически доказано точное количество попарно неизоморфных отказоустойчивых реализаций для N -куба, которые могут быть построены с использованием данных схем.
4. Для случая 1-ВОТР доказана неприводимость.

Полученные схемы опираются на класс 4-слойных графов, который будет описан далее в этой главе.

2.1 Многослойные графы

Будем называть граф k -слойным, если его вершины можно раскрасить в k цветов, занумерованных от 0 до $k - 1$, соблюдая следующие условия: для любого ребра разность цветов инцидентных ему вершин по модулю равна единице, и каждый из k цветов использован хотя бы для одной вершины. Совокупность всех таких графов будем обозначать через L_k .

На рисунке 14 показан пример 4-слойного графа. Для большей наглядности метки вершин обозначают цвета.

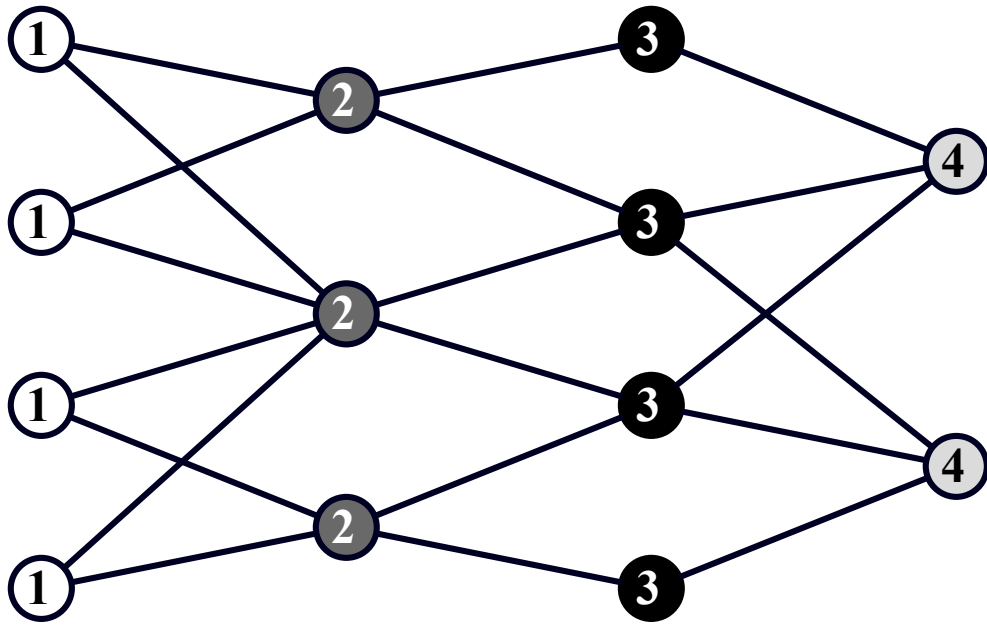


Рисунок 14 – Пример 4-слойного графа

Опишем множества k -слойных графов.

1-слойные графы – это множество, состоящее из вполне несвязных графов $O_n = (V, \emptyset)$, так как если в графе есть хотя бы одно ребро, то по определению вершины на его концах должны иметь разный цвет, а значит, будет как минимум 2 цвета.

По определению 2-слойные графы совпадают с двудольными графами. *Двудольным* называется граф, вершины которого можно разделить на две доли (два непересекающихся множества) так, чтобы каждое ребро соединяло вершины разных долей.

Лемма 5 (Об иерархии k -слойных графов). Для любого $k \geq 2$ справедливо включение $L_{k+1} \subseteq L_k$.

Доказательство. Возьмём произвольный граф $G \in L_{k+1}$ и его раскраску в $k+1$ цвет. Вершины, имеющие цвет $k+1$, по определению могут соседствовать только с вершинами цвета k . Вершины цвета $k-1$ также могут быть смежны с вершинами цвета k , поскольку $|(k+1)-k|=|(k-1)-k|=1$. Если теперь всем вершинам цвета $k+1$ присвоить цвет $k-1$, мы получим корректную раскраску, удовлетворяющую определению k -слойного графа,

значит $G \in L_k$. Поскольку граф G был выбран произвольно, включение $L_{k+1} \subseteq L_k$ доказано.

■

Как пример для иллюстрации леммы 5, 4-слойный граф, изображённый на рисунке 14, показан на рисунке 15 с раскраской 3-слойного графа.

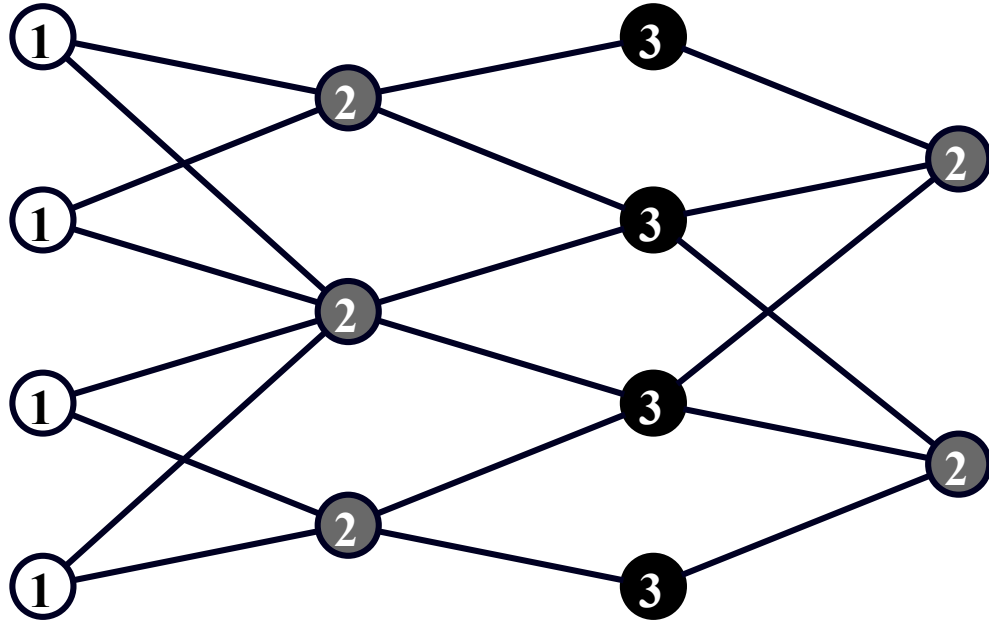


Рисунок 15 – Иллюстрация действия леммы 5 по перекрашиванию 4-слойного графа в 3-слойный

Определим представителей каждого из класса графов. Очевидно, что k -слойными являются только двудольные графы. Лемма 6 показывает, что допустимое значение k для двудольного графа зависит от *диаметра* графа – наибольшего расстояния между двумя вершинами.

Лемма 6 (Критерий k -слойного графа). Пусть граф G состоит из t компонент связности, диаметры которых равны $d_1, d_2, \dots, d_t$. Тогда G является k -слойным в том и только в том случае, когда G — двудольный граф и выполняется неравенство $k \leq \left(\sum_{i=1}^t d_i \right) + t$.

Доказательство. Начнём с рассмотрения связного графа. Выберем вершину u , эксцентриситет которой совпадает с диаметром графа. Присвоим

каждой вершине v цвет, равный расстоянию между вершинами v и u . В результате будет задействован $d+1$ цвет, поскольку существует вершина, удалённая от u на расстояние, равное диаметру d .

Рассмотрим произвольное ребро $\{v, w\}$. Так как w и v смежны, верны неравенства $d(u, v)+1 \geq d(u, w)$ и $d(u, w)+1 \geq d(u, v)$, откуда получаем $|d(u, v) - d(u, w)| \leq 1$. Если бы разность равнялась нулю, вершины v и w оказались бы в одной доле двудольного графа, что исключает наличие между ними ребра. Следовательно, $|d(u, v) - d(u, w)| = 1$.

Таким образом, была построена раскраска k -слойного графа, для $k=d+1$. С учётом леммы 5 можно построить раскраску и в меньшее количество цветов.

Допустим, что $k > d+1$. Тогда в графе найдётся вершина с цветом 0 и вершина с цветом $d+1$. Поскольку граф связный, между ними существует путь. Рёбра соединяют только вершины, цвета которых различаются на единицу, поэтому любой путь от вершины цвета 1 до вершины цвета $d+2$ должен содержать не менее $d+1$ рёбер, что превышает диаметр графа. Полученное противоречие доказывает, что $k \leq d+1$.

Рассмотрим случай несвязного графа.

Так как каждая компонента связности может рассматриваться как связный двудольный граф, к ней применимы результаты, полученные в первой части доказательства. Следовательно, каждая компонента может быть раскрашена в $k_i \leq d_i + 1$ цветов. Раскрасив каждую компоненту в разные цвета, получим $k \leq \sum_{i=1}^t (d_i + 1) = \left(\sum_{i=1}^t d_i \right) + t$ цветов.

■

Для задачи построения k -вершинных отказоустойчивых реализаций графов, в том числе и гиперкубов, важен класс 4-слойных графов. Описание представителей данного класса может быть получено с помощью следствия данной теоремы.

Следствие (Критерий 4-слойного графа). Двудольный граф $G=(V \cup U, \alpha)$, где U и V – доли графа, является 4-слойным тогда и только тогда, когда количество вершин в нём не меньше 4 и $(\exists u \in U)(\exists v \in V)\{u, v\} \notin \alpha$.

Доказательство. Достаточность. Если количество вершин меньше 4, то их невозможно раскрасить в 4 цвета. Если граф связный, то между u и v из разных долей нет ребра, поэтому путь между ними будет состоять минимум из трёх рёбер (три перехода между долями).

Если граф несвязный и количество компонент в нём равно 2, то либо одна из компонент является O_1 , тогда есть её диаметр 0, а в другой компоненте не менее 3-х вершин, при этом как минимум две из них в одной доле. Расстояние между вершинами одной доли не может быть меньше 2-х, то есть по формуле из леммы 6 получим, что диаметры равны 0 и 2, количество компонент равно 2, поэтому граф будет 4-слойным. Если в каждой компоненте минимум 2 вершины, то между ними есть ребро и диаметр не может быть меньше 1, следовательно, граф является 4-слойным.

Если количество компонент 3, то в графе из 4-х вершин будет хотя бы одна компонента с двумя вершинами, диаметр которой будет не меньше 1, а значит, по критерию из леммы 6 он будет 4-слойным.

Более 3-х компонент автоматически приводят к 4-слойности графа по той же формуле.

Необходимость следует из того, что вершины слоёв 1 и 4 не смежны друг с другом и принадлежат разным долям.

■

Таким образом, если граф является двудольным, количество вершин в нём не меньше 4, и в нём есть по крайней мере две несмежные вершины u и v из разных долей, то граф является 4-слойным. Под такое определение среди

двудольных графов не попадают полные двудольные графы $K_{n,m}=(V \cup U, \alpha)$, где V и U – доли, $|V|=n$, $|U|=m$, $\alpha=U \times V \cup V \times U$.

При большом количестве вершин n количество $|B_n|$ полных двудольных n -вершинных графов мало по сравнению с количеством n -вершинных двудольных графов $|b_n|$, что видно из таблицы 1. Таким образом, количество 4-слойных n -вершинных графов $|L_4^n|$ практически равно количеству n -вершинных двудольных графов.

Таблица 1 – Количество графов

n	2	3	4	5	6	7	8	9	10	11	12
$ b_n $	2	3	7	13	35	88	303	1119	5479	32303	251135
$ B_n $	1	1	2	2	3	3	4	4	5	5	6
$ L_4^n $	1	2	5	11	32	85	299	1115	5474	32298	251129

Данные по количеству двудольных графов взяты из последовательности A033995 онлайн энциклопедии целочисленных последовательностей [84].

Количество полных двудольных n -вершинных графов равно $\left\lfloor \frac{n}{2} \right\rfloor$ – количеству различных размеров долей у n -вершинного двудольного графа (различных двухэлементных множеств чисел $|\{\{a, b\} \mid a+b=n\}|$).

2.2 Вершинные отказоустойчивые реализации 4-слойных графов

Для 4-слойных графов можно предложить схему построения 1-ВОТР, в котором на 1 ребро меньше, чем в T1-ВОТР. Данная отказоустойчивая реализация описана теоремой 3.

Теорема 3 (О 1-вершинной отказоустойчивой реализации 4-слойного графа). Пусть дан двудольный граф $G=(W, \alpha)$ с n вершинами

($n \geq 4$), доли которого обозначены как U и V . Пусть в нём найдутся две несмежные вершины $u \in U$ и $v \in V$ (далее будем называть их опорными). Тогда граф $G^* = (W^*, \alpha^*)$, где $W^* = W \cup \{w\}$, а $\alpha^* = \alpha \cup \beta \cup \delta_u \cup \delta_v \cup \mu$, является 1-ВОТР графа G . При этом число добавленных рёбер равно $n - 1$, а дополнительные множества рёбер определяются следующим образом:

- $\beta = \{\{w, x\} | x \in N_v \cup N_u\}$;
- $\delta_u = \{\{u, x\} | x \in V \setminus N_u \setminus \{v\}\}$;
- $\delta_v = \{\{v, x\} | x \in U \setminus N_v \setminus \{u\}\}$;
- $\mu = \{\{u, v\}\}$.

Доказательство. Рассмотрим количество дополнительных рёбер в графе G^* . Множества β , δ_u , δ_v и μ не имеют попарных пересечений, поэтому:

$$\begin{aligned} |\beta \cup \delta_u \cup \delta_v \cup \mu| &= |\beta| + |\delta_u| + |\delta_v| + |\mu| = |N_u \cup N_v| + |V \setminus N_u \setminus \{v\}| + |U \setminus N_v \setminus \{u\}| + 1 = \\ &= |N_u| + |N_v| + |V| - |N_u| - 1 + |U| - |N_v| - 1 + 1 = |U| + |V| - 1 = n - 1. \end{aligned}$$

Заметим, что если вершины u и v и доли U и V поменять местами (переобозначить), то отказоустойчивая реализация останется тем же самым графом.

Рассмотрим несколько случаев удаления вершины:

Случай 1 – удалена вершина w . Покажем, что $g_w: g_w(x) = x$ является вложением G в $G^* - w = (W, (\alpha \cup \beta \cup \delta_u \cup \delta_v \cup \eta) \cap (W \times W)) = (W, \alpha \cup \delta_u \cup \delta_v \cup \eta)$. По построению $\{g_w(x), g_w(y)\} = \{x, y\} \in \alpha \subseteq \alpha^*$, $\alpha \subseteq W \times W$ и $w \notin W$, следовательно, все рёбра графа G присутствуют в $G^* - w$.

Случай 2 – удалена вершина v (для u симметрично).

$$g_v(x) = \begin{cases} w, & \text{если } x = v \\ x, & \text{иначе} \end{cases}$$

Рассмотрим $\{v, s\} \in \alpha$. По определению N_v , $s \in N_v$, а значит, $\{g_v(v), g_v(s)\} = \{w, s\} \in \beta$, при этом $w \neq v$ и $s \neq v$, следовательно, данное ребро присутствует в графе $G^* - v$.

Случай 3 – удалена вершина $r \in V \setminus \{v\}$ (для $r \in U \setminus \{u\}$ симметрично).

$$g_r(x) = \begin{cases} w, & \text{если } x = v \\ v, & \text{если } x = r \\ x, & \text{иначе} \end{cases}$$

Рассмотрим $\{r, s\} \in \alpha$. U и V – доли двудольного графа, поэтому если $r \in V \setminus \{v\}$, то $s \in U$. Значит, $\{g_r(r), g_r(s)\} = \{v, s\} \in \delta_v \cup \eta \cup \{\{x, y\} \mid \{v, x\} \in \alpha\}$, так как по построению вершина v смежна со всеми вершинами из V . Ребро $\{v, s\}$ существует в $G^* - r$, так как $v \neq r$ и $s \neq r$. Рассмотрение ребра $\{w, s\}$ аналогично пункту 2.

Таким образом, по определению G^* является 1-ВОТР графа G , и количество дополнительных рёбер в нём равно $n - 1$.

■

Степени вершин 1-ВОТР G^* графа G из теоремы 3 будут следующими:

1. $d(u_{G^*}) = |V|$ – вершина u_{G^*} смежна со всеми вершинами доли V .
2. $d(v_{G^*}) = |U|$ – вершина v_{G^*} смежна со всеми вершинами доли U .
3. $d(w) = d(u_G) + d(v_G)$ – вершина w смежна с каждой вершиной, с которой были смежны u и v в графе G .
4. $d(x_{G^*}) = d(x_G) + 1$ – к каждой другой вершине добавлено ровно одно ребро: если вершина была смежна с u_G или v_G , то дополнительное ребро ведёт к w , если нет, то вершины доли U соединяются с v , а из доли V – с u .

Построение данного 1-ВОТР похоже на построение T1-ВОТР, где рёбра, ведущие к дополнительной вершине, распределяются между u , v и w . Из-за

этого максимальная степень вершины отказоустойчивой реализации оказывается меньше.

Помимо схемы построения 1-ВОТР, для 4-слойного графа существует похожая схема построения 2-ВОТР, в котором на 2 ребра меньше, чем в Т2-ВОТР того же графа, и при этом максимальная степень вершины оказывается меньше. Теорема 4 описывает данную отказоустойчивую реализацию.

Теорема 4 (О 2-вершинной отказоустойчивой реализации 4-слойного графа). О 2-вершинной отказоустойчивой реализации 4-слойного графа). Пусть $G=(W, \alpha)$ – двудольный граф с n вершинами ($n > 3$), имеющий доли U и V , а $u \in U$ и $v \in V$ – две несмежные вершины. Тогда граф $G^*=(W^*, \alpha^*)$, где $W^*=W \cup \{w_u, w_v\}$, а $\alpha^*=\alpha \cup \beta_v \cup \beta_u \cup \delta_u \cup \delta_v \cup \eta$, будет являться 2-ВОТР графа G . Множества дополнительных рёбер определяются так:

- $\beta_v = \{\{w_v, x\} \mid x \in N_v \cup V\};$
- $\beta_u = \{\{w_u, x\} \mid x \in N_u \cup U\};$
- $\delta_u = \{\{u, x\} \vee x \in (V \setminus N_u \setminus \{v\})\};$
- $\delta_v = \{\{v, x\} \mid x \in U \setminus N_v \setminus \{u\}\};$
- $\eta = \{\{u, v\}\}.$

Доказательство. Подсчёт проводится по той же схеме, что и в теореме 3. Количество дополнительных рёбер:

$$\begin{aligned}
 & |N_v \cup V| + |N_u \cup U| + |V \setminus N_u \setminus \{v\}| + |U \setminus N_v \setminus \{u\}| + 1 = \\
 & = |N_v| + |V| + |N_u| + |U| + |V| - |N_u| - 1 + |U| - |N_v| - 1 + 1 = \\
 & = 2|V| + 2|U| - 1 = 2n - 1
 \end{aligned}$$

Рассмотрим различные возможные ситуации и укажем схемы построения вложения. Для начала отметим, что добавление рёбер в графе производится симметрично относительно троек (u, U, w_u) и (v, V, w_v) , поскольку при переименовании u в v и w_u в w_v , U в V и наоборот, будет

построен точно такой же граф, что и без переименования. Воспользуемся данной симметрией для уменьшения количества перебираемых случаев.

Случай 1 – граф $G^* - w_v$ является 1-ВОТР графа G , так как в него вкладывается 1-ВОТР $H^* = (W \cup \{w\}, \alpha^1 \cup \beta^1 \cup \delta_u^1 \cup \delta_v^1 \cup \eta^1)$ графа G , построенного по теореме 3. Покажем это:

1. $\{s, x\} \in \alpha^1 \cup \delta_u^1 \cup \delta_v^1 \cup \eta^1 \Rightarrow \{\varphi(s), \varphi(x)\} = \{s, x\} \in \alpha \cup \delta_u \cup \delta_v \cup \eta$ так как $\alpha^1 = \alpha$, $\delta_u^1 \subseteq \delta_u$, $\delta_v^1 \subseteq \delta_v$ и $\eta^1 = \eta$.
2. $\{w, x\} \in \beta^1 \Rightarrow x \in N_u \cup N_v \subseteq N_u \cup U \Rightarrow \{\varphi(w), \varphi(x)\} = \{w_u, x\} \in \beta_u$.

Это означает, что вложение при удалении вершины w_v и произвольной другой вершины существует. При удалении w_u ситуация аналогична.

Случай 2 – рассмотрим $H^* = (W \cup \{w\}, \alpha^1 \cup \beta^1 \cup \delta_u^1 \cup \delta_v^1 \cup \eta^1)$ – 1-ВОТР графа G из теоремы 3, и $G^* - v$ и покажем, что существует вложение φ графа H^* в G^* .

$$\varphi(x) = \begin{cases} w_v, & \text{если } x = w \\ w_u, & \text{если } x = v \\ x, & \text{иначе} \end{cases}$$

1. $\{s, x\} \in \alpha^1 \Rightarrow s = v$ (при $x = v$ аналогично) $\Rightarrow \{\varphi(v), \varphi(x)\} = \{w_u, x\} \in \beta_u$ или $s \neq v$ и $x \neq v \Rightarrow \{\varphi(s), \varphi(x)\} = \{s, x\} \in \alpha$;
2. $\{w, x\} \in \beta^1 \Rightarrow x \in N_u \cup N_v \Rightarrow \{\varphi(w), \varphi(x)\} = \{w_v, x\} \in \beta_v$;
3. $\{u, x\} \in \delta_u^1 \Rightarrow x \in V \setminus N_u \setminus \{v\} \Rightarrow \{\varphi(u), \varphi(x)\} = \{u, x\} \in \delta_u$;
4. $\{v, x\} \in \delta_v^1 \Rightarrow x \in U \setminus N_v \setminus \{u\} \subseteq U \Rightarrow \{\varphi(v), \varphi(x)\} = \{w_u, x\} \in \beta_u$;
5. $\{u, v\} = \{v, u\} \in \eta^1 \Rightarrow \{\varphi(v), \varphi(u)\} = \{w_u, u\} \in \beta_u$.

То есть в $G^* - v$ вкладывается 1-ВОТР графа G , а это значит, что $G^* - v$ также является 1-ВОТР графа G . Случай $G - u$ аналогичен.

Случай 3 – рассмотрим удаление вершин, принадлежащих одной доли:
 $r_1, r_2 \in V \setminus \{v\}$.

$$g_1(x) = \begin{cases} w_u, & \text{если } x = r_1 \\ v, & \text{если } x = r_2 \\ w_v, & \text{если } x = v \\ x, & \text{иначе} \end{cases}$$

Покажем, что отображение g_1 является вложением:

1. $\{r_1, s\} \in \alpha \Rightarrow s \in U \Rightarrow \{g_1(r_1), g_1(s)\} = \{w_u, s\} \in \beta_u \subseteq \alpha^*$;
2. $\{r_2, s\} \in \alpha \Rightarrow s \in U \Rightarrow \{g_1(r_2), g_1(s)\} = \{v, s\}$;
 - 2.1. $s = u \Rightarrow \{v, s\} = \{v, u\} \in \eta$;
 - 2.2. $s \in N_v \Rightarrow \{v, s\} \in \alpha \subseteq \alpha^*$;
 - 2.3. $s \in U \setminus N_v \setminus \{u\} \Rightarrow \{v, s\} \in \delta_v \subseteq \alpha^*$;
3. $\{v, s\} \in \alpha \Rightarrow s \in N_v \Rightarrow \{g_1(v), g_1(s)\} = \{w_v, s\} \in \beta_v \subseteq \alpha^*$;
4. $\{s_1, s_2\} \in \alpha \wedge \{s_1, s_2\} \cap \{v, r_1, r_2\} = \emptyset \Rightarrow \{g_1(s_1), g_1(s_2)\} = \{s_1, s_2\} \in \alpha \subseteq \alpha^*$.

По определению g_1 является вложением.

Случай $r_1, r_2 \in U \setminus \{u\}$ аналогичен.

Случай 4 – при удалении вершин из разных долей: $r_v \in V \setminus \{v\}$ и $r_u \in U \setminus \{u\}$.

$$g_2(x) = \begin{cases} v, & \text{если } x = r_v \\ u, & \text{если } x = r_u \\ w_u, & \text{если } x = v \\ w_v, & \text{если } x = u \\ x, & \text{иначе} \end{cases}$$

Рассмотрим вложение и рёбра графа G :

1. $\{u, s\} \in \alpha \Rightarrow s \in N_u \Rightarrow \{g_2(u), g_2(s)\} = \{w_v, s\} \in \beta_v \subseteq \alpha^*$;
2. $\{v, s\} \in \alpha \Rightarrow s \in N_v \Rightarrow \{g_2(v), g_2(s)\} = \{w_u, s\} \in \beta_u \subseteq \alpha^*$;

3. $\{r_u, s\} \in \alpha \Rightarrow s \in V$ (так как G – двудольный и $r_u \in U$):

$$3.1. \quad s = r_v \Rightarrow \{g_2(r_u), g_2(r_v)\} = \{u, v\} \in \eta \subseteq \alpha^*;$$

$$3.2. \quad s = v \Rightarrow \{g_2(r_u), g_2(v)\} = \{u, w_u\} = \{w_u, u\} \in \beta_u \subseteq \alpha^*;$$

$$3.3. \quad s \in N_u \Rightarrow \{g_2(r_u), g_2(s)\} = \{u, s\} \in \alpha \subseteq \alpha^*;$$

$$3.4. \quad s \in V \setminus N_u \setminus \{r_v, v\} \Rightarrow \{g_2(r_u), g_2(s)\} = \{u, s\} \in \delta_u \subseteq \alpha^*;$$

4. $\{r_v, s\} \in \alpha \Rightarrow s \in U$:

$$4.1. \quad s = r_u \Rightarrow \{g_2(r_v), g_2(r_u)\} = \{v, u\} = \{u, v\} \in \eta \subseteq \alpha^*;$$

$$4.2. \quad s = u \Rightarrow \{g_2(r_v), g_2(u)\} = \{v, w_v\} = \{w_v, v\} \in \beta_v \subseteq \alpha^*;$$

$$4.3. \quad s \in N_v \Rightarrow \{g_2(r_v), g_2(s)\} = \{v, s\} \in \alpha \subseteq \alpha^*;$$

$$4.4. \quad s \in U \setminus N_v \setminus \{r_u, u\} \Rightarrow \{g_2(r_v), g_2(s)\} = \{v, s\} \in \delta_v \subseteq \alpha^*;$$

$$5. \quad \{s_1, s_2\} \in \alpha \wedge \{s_1, s_2\} \cap \{u, v, r_u, r_v\} = \emptyset \Rightarrow \{g_2(s_1), g_2(s_2)\} = \{s_1, s_2\} \in \alpha \subseteq \alpha^*.$$

Таким образом, g_2 является вложением G в $G^* - r_u - r_v$.

Все случаи разобраны, поэтому по определению G^* является 2-ВОТР графа G . ■

Степени вершин 2-ВОТР G^* графа G из теоремы 4 будут следующими:

1. $d(u_{G^*}) = |V| + 1$ – вершина u_{G^*} смежна с w_u и со всеми вершинами из доли V .
2. $d(v_{G^*}) = |U| + 1$ – вершина v_{G^*} смежна с w_v и со всеми вершинами из доли U .
3. $d(w_u) = |U| + d(u_G)$ – вершина w_u смежна с каждой вершиной, с которой были смежны u и v в графе G , а также со всеми вершинами из U .
Вершины, смежные с v_G , находятся в доле U , поэтому степень v_G в формуле не учитываются.
4. $d(w_v) = |V| + d(v_G)$ – аналогично пункту 3.

5. $d(x_{G^*})=d(x_G)+2$ – к каждой другой вершине добавлено ровно два ребра:

5.1. если вершина была смежна с u_G или v_G , то дополнительные рёбра ведут к w_u и w_v .

5.2. если нет, то вершины доли U соединяются с v и w_u , а из доли V – с u и w_v .

Построение данного 2-ВОТР похоже на построение 1-ВОТР из теоремы 3 разделением вершин, но с отличиями. Основное – это то, что дополнительные вершины соединяются со всеми вершинами одной из долей. Это структурные изменения, которые не позволяют прогнозировать по аналогии, каким образом будут выглядеть схемы построения Mk -ВОТР для $k > 3$.

Рассмотрим пример построения данных 1- и 2-отказоустойчивые реализации. На рисунке 16 приведён пример таких графов. Под буквой *a* изображён 4-слойный граф G . Под пунктом *б* показана дополнительная вершина и разметка, применяющаяся при построении отказоустойчивой реализации. Под пунктом *в* изображена отказоустойчивая реализация, построенная по методу из теоремы 3 при данной разметке вершин. Под буквой *г* показаны 2-вершинные отказоустойчивые реализации, построенные по теореме 4.

На рисунке видно, что 1-ВОТР является частью 2-ВОТР, если разметка вершин исходного графа одинакова. Также из рисунка видно, что данная разметка не единственная. В графе много других вершин из разных долей, не соединённых ребром, поэтому количество отказоустойчивых реализаций, которые можно построить, больше, чем одна.

Описанные в теоремах 3 и 4 отказоустойчивые реализации могут являться минимальными. Также данные вершинные отказоустойчивые

реализации могут быть неприводимыми, а могут и не быть – это будет зависеть непосредственно от графа и требует проверки.

Для регулярных графов можно упростить проверку на неприводимость. Для доказательства понадобится лемма 2 (лемма 2.1.4 из [8]). Само доказательство приведено в леммах 7 и 8.

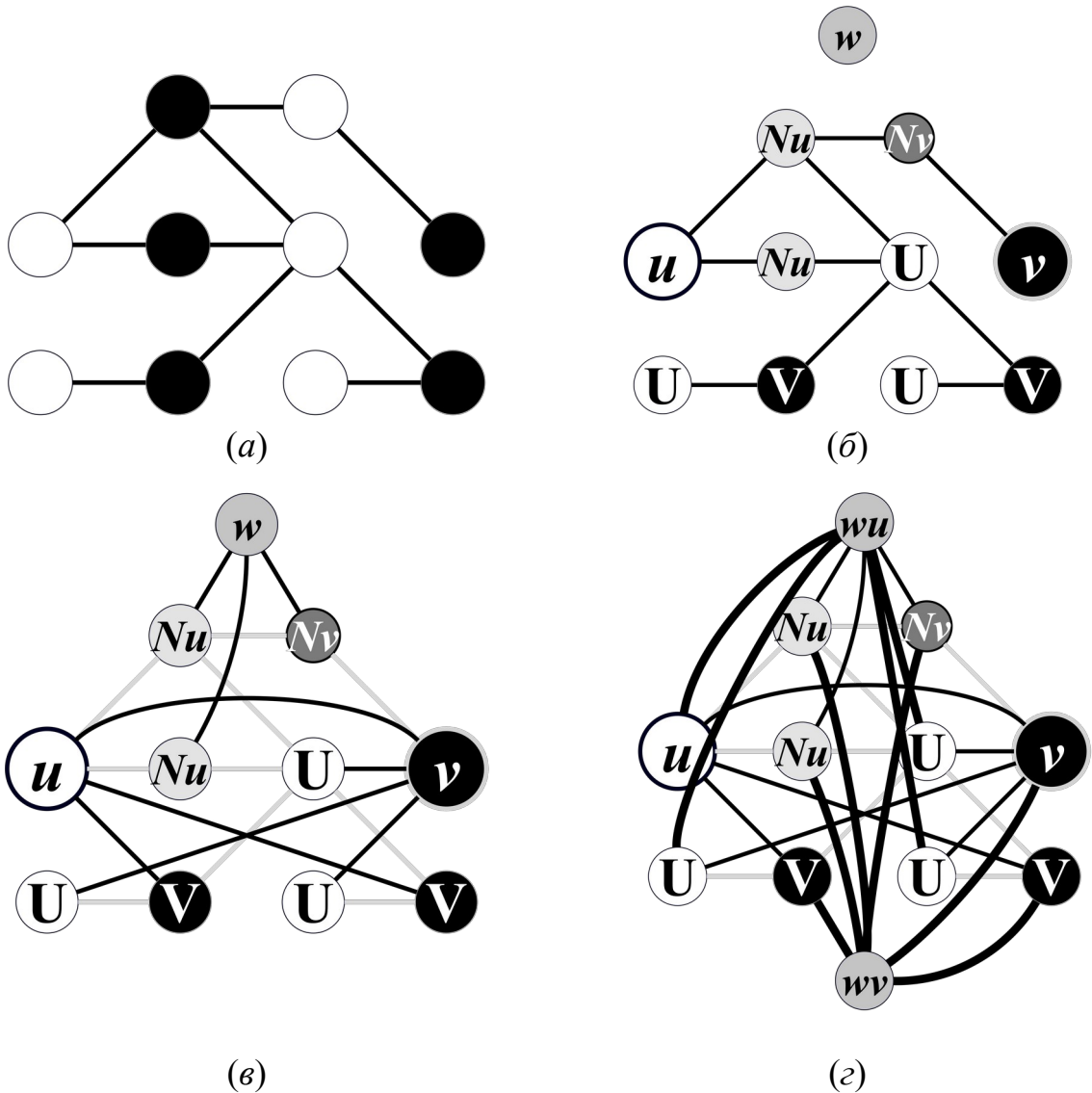


Рисунок 16 – Пример построения 1-ВОТР и 2-ВОТР 4-слойного графа

Лемма 7. Если G – регулярный 4-слойный граф степени $d > 0$, то G^* – 1-ВОТР графа G , построенное по схеме из теоремы 3, неприводимо тогда и только тогда, когда $G^* - \{u, v\}$ не является 1-ВОТР графа G .

Доказательство. По построению, в теореме 3 к каждой вершине графа G добавляется ребро $\{u, v\}$, а также к каждой вершине x , отличной от u, v и w , добавляется ребро $\{x, v\}$, $\{x, u\}$ или $\{x, w\}$, что увеличивает степень вершины x до $d+1$. Это означает, что если удалить ребро, отличное от $\{u, v\}$, из графа G^* , то это ребро будет соединять вершину степени $d+1$, образуется вершина степени d . Такой граф не может быть 1-ВОТР графа G , так как по лемме 2 минимальная степень вершины в Mk -ВОТР, а значит и в каждом k -ВОТР с тем же количеством вершин степень должна быть не меньше $d+1$. Отсюда следует, что если G^* без ребра $\{u, v\}$ является 1-ВОТР, то отказоустойчивая реализация G^* – приводимая, иначе – неприводимая. ■

Лемма 8. Если G – регулярный 4-слойный граф степени $d>0$, то G^* – 2-ВОТР графа G , построенное по схеме из теоремы 4, неприводимо тогда и только тогда, когда ни один из трёх графов $G^* - \{u, v\}$, $G^* - \{w_v, v\}$ и $G^* - \{w_u, u\}$ не является 2-ВОТР графа G .

Доказательство. Опираясь на доказательство леммы 1, получим, что среди дополнительных рёбер удаление произвольного ребра, отличного от $\{u, v\}$, $\{w_v, v\}$ и $\{w_u, u\}$, приводит к образованию вершины степени $d+1$, что меньше, чем $d+2$. Поэтому для определения неприводимости отказоустойчивая реализация остаётся проверить только те графы, которые получены удалением данных рёбер. ■

Найденные схемы построения 1-ВОТР и 2-ВОТР можно применить к различным известным классам графов. Например, к деревьям и многомерным решёткам, так как они являются двудольными графами, а также к некоторым торам $T_{n_1 \times n_2 \times \dots \times n_d}$, которые тоже являются двудольными при чётных значениях $n_1, \dots, n_d$. К 4-слойным графам относятся также и гиперкубы.

2.3 Вершинные отказоустойчивые реализации гиперкуба как 4-слойного графа

Теоремы 3 и 4, применимы к N -кубам при $N \geq 3$, поскольку для них можно построить $(N+1)$ -слойную раскраску: цвет i ($i \in \{0, 1, \dots, N\}$) присваивается вершинам, в метках которых содержится ровно i единиц. Эту раскраску мы будем использовать в дальнейшем.

Поскольку N -куб является двудольным [68] и симметричным [62] графом, обе его доли содержат равное число вершин. Все 2^N вершин N -куба имеют степень N . Следовательно, степени вершин 1-ВОТР распределяются так:

1. $d(u) = d(v) = \frac{2^N}{2} = 2^{N-1}$.
2. $d(w) = 2N$.
3. Степень остальных вершин равна $N+1$.

Каждый N -куб $Q_N = (V, \alpha)$ является дистанционно-транзитивным графом [62], то есть $(\forall u_1, u_2, v_1, v_2 \in V) d(u_1, u_2) = d(v_1, v_2) \Rightarrow \exists \Omega_{u_1, u_2, v_1, v_2} \in \text{Aut}(Q) : \Omega_{u_1, u_2, v_1, v_2}(u_1) = v_1 \wedge \Omega_{u_1, u_2, v_1, v_2}(u_2) = v_2$.

Построение 1-ВОТР для 3-куба и 4-куба изображено на рисунках 17 и 18 соответственно.

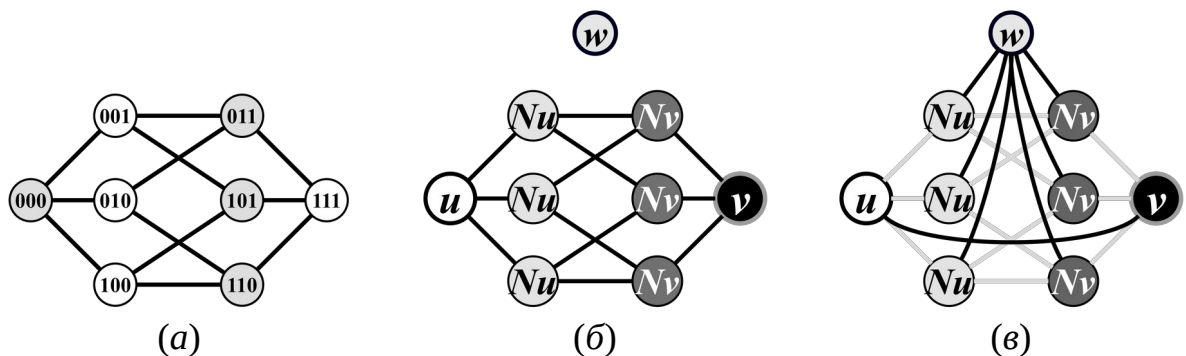


Рисунок 17 – Построение 1-ВОТР 3-куба

Под пунктом a изображён сам гиперкуб, цвет вершин означает принадлежность к 1-ой или 2-ой доле. Далее под пунктом b показана

разметка вершин, которая используется при построении 1-ВОТР, и под пунктом *в* показана сама отказоустойчивая реализация. Более тёмным цветом относительно других выделены дополнительные рёбра.

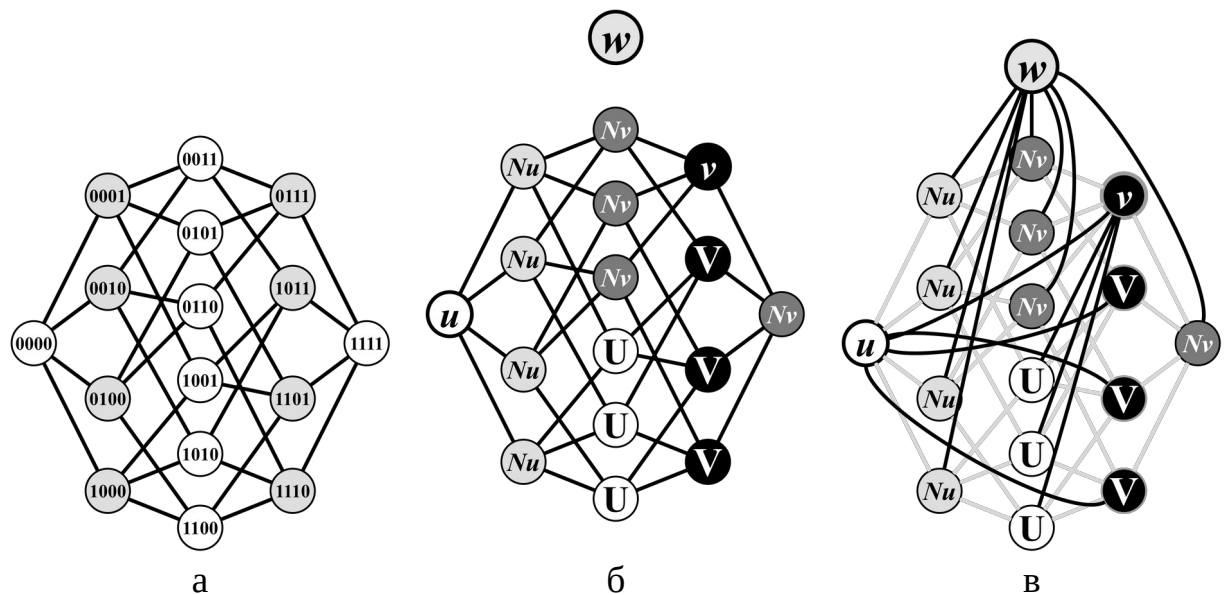


Рисунок 18 – Построение 1-ВОТР 4-куба

Других неизоморфных данным графам 1-ВОТР, которые можно построить по теореме 3 для 3-куба и 4-куба, не существует.

Для 5-куба ситуация другая. По тому же принципу можно построить 2 неизоморфные отказоустойчивые реализации, которые изображены на рисунке 19 под пунктами *а* и *б*.

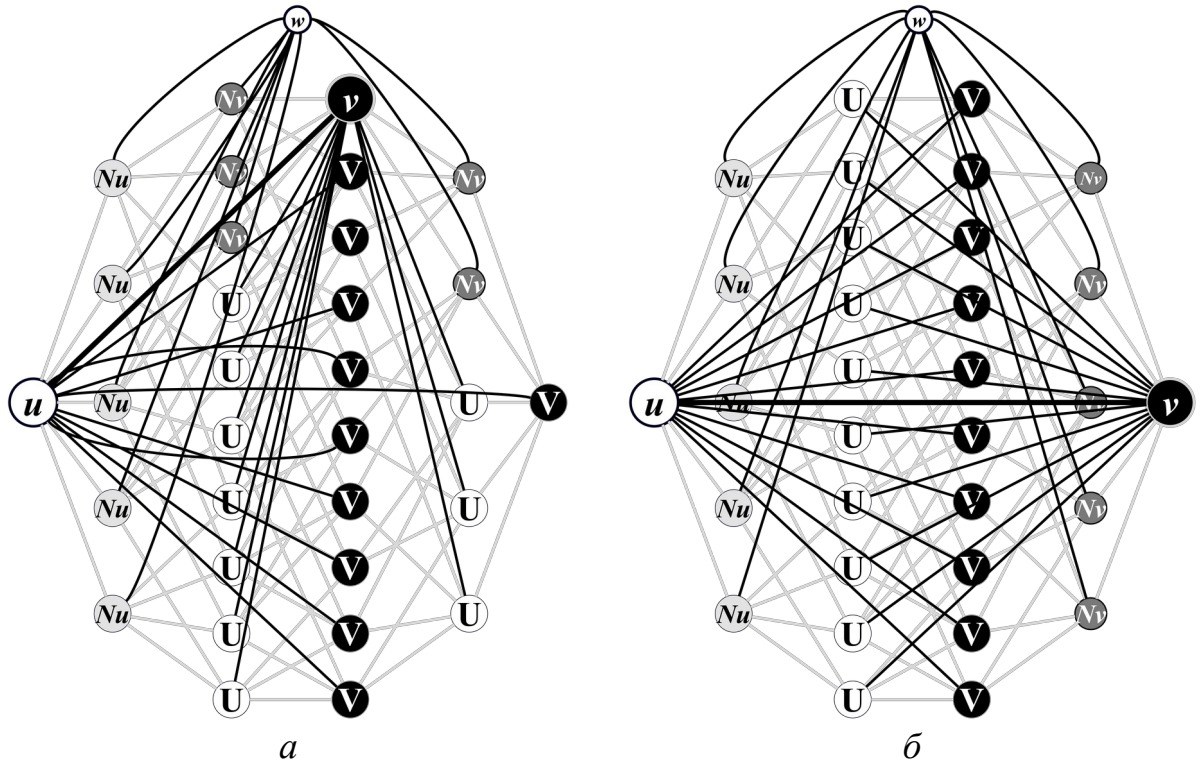


Рисунок 19 – 1-ВОТР 5-куба

Таким образом, возникает вопрос о количестве попарно неизоморфных 1-ВОТР и 2-ВОТР гиперкуба, которые можно построить с помощью теорем 3 и 4. Ответ на данный вопрос дают теоремы 5 и 6 соответственно.

Теорема 5 (О числе неизоморфных 1-ВОТР гиперкуба, которые можно получить по теореме 3). По теореме 3 можно построить равно $\left\lfloor \frac{N-1}{2} \right\rfloor$ попарно неизоморфных 1-ВОТР N -куба.

Доказательство. В силу дистанционной транзитивности гиперкуба G , если взять две пары вершин $\{u, v\}$ и $\{a, b\}$, находящиеся на одинаковом расстоянии друг от друга, в качестве ведущих, то построенные таким образом отказоустойчивые реализации окажутся изоморфными. Изоморфизм между ними задаётся отображением $\Omega_{u,v,a,b} \cup (w, w)$, где через $\Omega_{u,v,a,b}$ обозначается автоморфизм гиперкуба, такой, что $\Omega_{u,v,a,b}(u)=a$ и $\Omega_{u,v,a,b}(v)=b$.

Отсюда следует, что неизоморфными могут оказаться лишь те отказоустойчивые реализации гиперкуба с опорными вершинами u и v , расстояние между которыми принимает различные значения из ряда 3, 5, 7 и т. д. Покажем это. Выберем произвольные две отказоустойчивые реализации N -куба H_1 и H_2 . Расстояние между опорными вершинами в них равно d_1 и d_2 , при этом пусть $d_1 < d_2$. Ранее отмечалось, что степени вершин отказоустойчивых реализаций N -куба, равны:

1. $d(w) = 2N$.
2. $d(u) = d(v) = 2^{N-1}$.
3. Степень остальных вершин равна $N + 1$.

Проведём в G_1 путь длины d_1 , не проходящий дважды через одну вершину. Обозначим его как $(u, x_1, x_2, \dots, x_{d_1-2}, x_{d_1-1}, v)$, где $x_1 \in N_u, x_{d_1-1} \in N_v, d(x_k) = N + 1$. Вершины из множеств N_u и N_v отличаются от других, так как смежны с вершиной w уникальной степени $2N$.

Проверим возможность существования аналогичного пути в графе G_2 . В таком пути нет вершины w . Это значит, что рёбра из множества β не могут быть использованы. Вершины u и v находятся на концах, а $x_1 \in N_u, x_{d_1-1} \in N_v$ и они всегда есть в пути, так как $d_1 \geq 3$, поэтому рёбра из δ_u, δ_v и η не могут быть использованы. Поэтому использоваться могут только рёбра самого гиперкуба. Так как расстояние между вершинами u и v в самом гиперкубе отличается и $d_2 > d_1$, значит такой путь в графе отсутствует. Из-за того, что G_1 содержит простой путь длины d_1 между вершинами степени 2^{N-1} с заданными условиями, а в G_2 – нет, две данные полученные по теореме 3 отказоустойчивые реализации гиперкуба неизоморфны. И необходимым и достаточным условием для этого является различное расстояние между

опорными вершинами. Отсюда следует, что количество неизоморфных 1-ВОТР для N -куба, построенных по схеме из теоремы 3, равно $\left\lfloor \frac{N-1}{2} \right\rfloor$.

■

Теорема 6 (О числе неизоморфных 2-ВОТР гиперкуба, которые можно получить по теореме 4). Количество попарно неизоморфных 2-ВОТР N -куба, которые можно получить по теореме 4, равно $\left\lfloor \frac{N-1}{2} \right\rfloor$.

Доказательство. Рассуждения аналогичны доказательству теоремы 5. При построении пути в качестве промежуточных вершин не могут выступать u, v, w_u и w_v , поэтому могут использоваться только рёбра гиперкуба. Степени всех промежуточных вершин в этом случае равны $N+2$, а степени остальных от этого значения отличаются.

■

Вопрос о том, являются ли отказоустойчивые реализации, построенные по теоремам 3 и 4, минимальными, остаётся открытым, как и вопрос о количестве минимальных отказоустойчивых реализаций. Однако, можно однозначно утверждать, что построенные по теореме 3 1-ВОТР гиперкуба являются неприводимыми, что доказывается в следующей теореме.

Теорема 7 (О неприводимости 1-вершинной отказоустойчивой реализации гиперкуба). Всякая 1-ВОТР N -куба, построенная по теореме 3, является неприводимой.

Доказательство. Согласно лемме 7 1-ВОТР G^* N -куба Q_N является неприводимой в том и только в том случае, когда граф $G^* - \{u, v\}$ не является 1-ВОТР. Рассмотрим граф $H = G^* - \{u, v\} - r$, где $r \in N_u$.

Рассмотрим отдельно случай $N=3$. В графе G^* вершина u имеет степень 4. После удаления смежной с ней вершины r и ребра $\{u, v\}$ степень u

в графе H становится равной 2, что меньше степени вершин в 3-кубе (она равна 3). Поскольку число вершин в G и H совпадает, вложение G в H невозможно.

Перейдём к случаю $N > 3$. Сопоставим каждой вершине N -куба цвет, равный числу единиц в её метке, что соответствует раскраске $(N + 1)$ -слойного графа. Гиперкуб обладает свойством вершинной симметричности: если существует вложение, при котором вершина u переходит в v , то можно построить и вложение, переводящее любую другую вершину гиперкуба в v . Если же такого вложения нет, то его не существует и для остальных вершин, поскольку рассматривается вложение в граф с тем же числом вершин.

В N -кубе существует единственная вершина s_0 цвета 0. С ней смежны все вершины цвета 1, количество которых равно N . Обозначим их через $s_1, s_2, \dots, s_N$.

Каждая вершина цвета 1 содержит в своей метке ровно одну единицу, поэтому её можно однозначно идентифицировать по номеру позиции, в которой эта единица стоит.

Отображение, переводящее вершину в другую вершину, метка которой получается перестановкой битов исходной метки, является автоморфизмом гиперкуба. Действительно, одновременная перестановка компонент в двух двоичных метках не изменяет расстояние Хэмминга между ними. С помощью такого автоморфизма можно произвольно переставлять вершины цвета 1, поэтому при проверке вложения достаточно рассмотреть лишь один фиксированный порядок этих вершин.

Пусть вершина r не смежна ни с одной вершиной из N_v . В графе G^* вершина r имеет $N + 1$ инцидентное ребро, среди которых N рёбер гиперкуба и одно дополнительное ребро, ведущее к w . Следовательно, с вершинами из N_v она может быть смежна только посредством рёбер гиперкуба. Если

расстояние между u и v в гиперкубе превышает 3, то расстояние между вершинами, смежными с u и v , больше 1, и они не могут быть смежны в G^* . Если же это расстояние равно 3, метки u и v различаются в трёх позициях, и метка r должна совпадать с u во всех этих позициях, что возможно, поскольку $\setminus(N > 3 \setminus)$.

Назовём вершинами типа t образы вершин цвета t гиперкуба. В графе $H = G^* - \{u, v\} - r$. вершина r была смежна с u , w и $N - 1$ вершинами, степень которых равнялась $N + 1$. После удаления r в графе H их степень стала равна N . Обозначим эти вершины через $q_1, \dots, q_{N-1}$.

Предположим, что s_0 отображается в v , то есть v — вершина типа 0. Поскольку $r \in N_u \subseteq V$, множество $\{q_1, \dots, q_{N-1}\} \subset U$ целиком лежит в U , а значит, все эти вершины смежны с v . Их степени равны N , что совпадает со степенью вершин N -куба, поэтому все инцидентные им рёбра должны быть использованы при вложении. Следовательно, они являются вершинами типа 1, и в них отображаются $N - 1$ вершин $s_1, \dots, s_{N-1}$ цвета 1.

Порядок выбора неважен, это было указано ранее. Осталось выбрать последнюю вершину типа 1 графа H .

Поскольку все рёбра вершин $q_1, \dots, q_{N-1}$ должны быть использованы, только смежные с ними вершины будут иметь тип 2. Каждая из вершин q_i отличается от r ровно в одной позиции, поэтому их метки можно записать так:

$$r = (r_1, r_2, \dots, r_{N-2}, r_{N-1}, r_N);$$

$$q_1 = (r_1 \oplus 1, r_2, \dots, r_{N-2}, r_{N-1}, r_N);$$

...

$$q_{N-1} = (r_1, r_2, \dots, r_{N-2}, r_{N-1} \oplus 1, r_N).$$

Каждая пара вершин q_i, q_j имеет две общие смежные вершины: v типа 0 и $q_{i,j} = (r_1, \dots, r_{i-1}, r_i \oplus 1, r_{i+1}, \dots, r_{j-1}, r_j \oplus 1, r_{j+1}, \dots, r_N)$, которая может

быть вершиной типа 2. Каждая вершина $q_i \in \{q_1, \dots, q_{N-1}\}$ смежна с другой вершиной, имеющей метку $q_{i,N} = (r_1, \dots, r_{i-1}, r_i \oplus 1, r_{i+1}, \dots, r_{N-1}, r_N \oplus 1)$.

Тогда число смежных вершин будет равно числу вершин цвета 2:

$$C_{n-1}^2 + C_{n-1}^1 = \frac{(n-1)(n-2)}{2} + (n-1) = \frac{(n-1)(n-2)}{2} + \frac{2(n-1)}{2} = \frac{n(n-1)}{2} = C_n^2 \quad \text{а}$$

значит все они будут вершинами типа 2.

На примере 5-куба на рисунке 20 показаны определённые на текущий момент доказательства вершины типа 0 – белой вершиной с чёрной меткой, 1 – светло-серой вершиной с чёрной меткой и 2 – тёмно-серой вершиной с белой меткой. Пунктиром показаны удалённые рёбра.

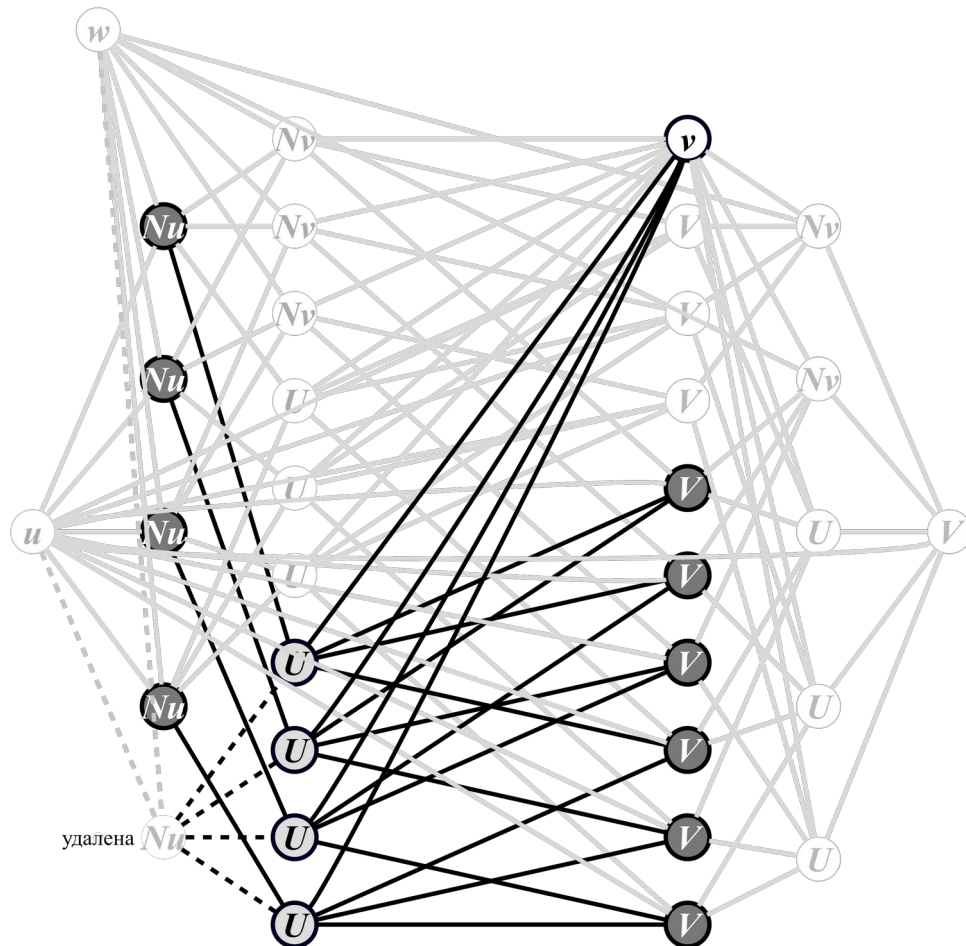


Рисунок 20 – Пример используемых вершин при доказательстве неприводимости 5-куба

Каждая вершина цвета 2 соединена с двумя другими вершинами цвета 1, причём для любой пары вершин цвета 1 должна существовать общая смежная вершина цвета 2. Следовательно, последняя вершина типа 1 обязана быть смежной с вершинами $q_{i,N}$, а также с вершиной v . Поскольку по построению вершина w не смежна с v , она не может иметь данный тип.

То есть ни одно дополнительное ребро не может быть использовано, а значит, необходимо проверить рёбра N -куба. Все $q_{i,N} = (r_1, \dots, r_{i-1}, r_i \oplus 1, r_{i+1}, \dots, r_{N-1}, r_N \oplus 1)$ смежны одновременно только с $(r_1, \dots, r_{N-1}, r_N \oplus 1)$, которая также смежна с r . Так как все смежные с r вершины уже являются вершинами типа 1, то остаётся последняя – u , так как r был выбран из множества N_u . Но она не смежна с w .

Таким образом, для последней вершины гиперкуба цвета 1 образ найти невозможно. Это значит, что граф $H = G^* - \{u, v\}$ не является 1-ВОТР, поэтому 1-ВОТР G^* является неприводимым. ■

2.4 Связь с результатами вычислительных экспериментов

Теоремы 3 и 4 позволяют строить неприводимые 1- и 2-вершинные отказоустойчивые реализации гиперкубов. Все минимальные отказоустойчивые реализации являются неприводимыми, но не все неприводимые отказоустойчивые реализации являются минимальными.

Однако проведённые вычислительные эксперименты показали, что полученные с помощью теоремы 3 Н1-ВОТР 3-куба и 4-куба не только являются минимальными, но также являются и единственными минимальными 1-вершинными отказоустойчивыми реализациями данных графов.

Также с помощью теоремы 4 может быть получено одно из найденных программным путём М2-ВОТР 3-куба. Второе же полученное М2-ВОТР 3-куба данной теоремой не описывается.

Глава 3. Моделирование рёберной отказоустойчивости гиперкубов

В данной главе даказывается, что у любого гиперкуба существует единственное с точностью до изоморфизма М1-РОТР.

Определение Q -графов было приведено в первой главе.

Каждый N -куб можно выразить через данное определение: $Q_N^1 = Q_N$.

Граф $Q_N^{1,N}$ является 1-рёберной отказоустойчивой реализацией N -куба Q_N . Впервые этот результат появился в [69]. Далее будет показано, что данная отказоустойчивая реализация является единственной минимальной 1-рёберной отказоустойчивой реализацией N -куба при $N \geq 2$.

Лемма 9 (О 1-рёберной отказоустойчивой реализации N -куба). Граф $Q_N^{1,N}$ является 1-РОТР N -куба Q_N .

Доказательство. Предположим, что из графа $Q_N^{1,N}$ удалено ребро $\{u, v\}$. Тогда возможны 2 случая.

$h(u, v) = N$ это первый случай. Так как $Q_N^{1,N} = Q_N^1 \cup Q_N^N$, было удалено ребро графа Q_N^N , и графы Q_N^1 и Q_N^N не имеют общих рёбер, то вложением N -куба $Q_N = Q_N^1$ в $Q_N^{1,N} - \{u, v\}$ будет тождественное отображение $\phi: \phi(x) = x$.

Второй случай, когда удалено ребро гиперкуба, то есть $h(u, v) = 1$. Следовательно, метки вершин u и v имеют различие в одной позиции. Пусть это будет позиция k . Пусть $x = (x_1, \dots, x_N)$, составим отображение $\psi_i: \{0, 1\}^N \rightarrow \{0, 1\}^N$ следующим образом:

$$\psi_i(x) = \begin{cases} x, & \text{если } x_i = 0 \\ x' = (x_1 \oplus 1, \dots, x_{i-1} \oplus 1, x_i, x_{i+1} \oplus 1, x_N \oplus 1) & \text{иначе} \end{cases}$$

Докажем, что ψ_i – вложение. Произвольно выберем ребро $\{a, b\}$ N -куба. Выделяется 3 случая:

1 случай: если $a_i = b_i = 0$, то данное ребро не равно $\{u, v\}$, так как вершины на его конце различаются только в позиции $j \neq i$. Ребро $\{\psi_i(u), \psi_i(v)\} = \{u, v\}$, а значит, оно присутствует в графе $Q_N^{1,N} - \{u, v\}$.

2 случай: если $a_i = b_i = 1$, то a и b имеют отличие только в позиции $j \neq i$, то есть

$$\begin{cases} a = (a_1, \dots, a_{j-1}, a_j, a_{j+1}, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_N) \\ b = (a_1, \dots, a_{j-1}, a_j \oplus 1, a_{j+1}, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_N) \end{cases} \quad \text{После}$$

применения отображения полученные вершины будут равны:

$$\begin{cases} \psi_i(a) = (a_1 \oplus 1, \dots, a_{j-1} \oplus 1, a_j \oplus 1, a_{j+1} \oplus 1, \dots, a_{i-1} \oplus 1, 1, a_{i+1} \oplus 1, \dots, a_N \oplus 1) \\ \psi_i(b) = (a_1 \oplus 1, \dots, a_{j-1} \oplus 1, a_j, a_{j+1} \oplus 1, \dots, a_{i-1} \oplus 1, 1, a_{i+1} \oplus 1, \dots, a_N \oplus 1) \end{cases}$$

3 случай: если $a_i \neq b_i$. Без ограничения общности пусть $a_i = 0$. Из этого следует, что метки вершин равны во всех прочих позициях:

$$\begin{cases} \psi_i(a) = (a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_N) \\ \psi_i(b) = (a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_N) \end{cases} \quad \text{После применения отображения к}$$

данным вершинам получим следующие метки:

$$\begin{cases} \psi_i(a) = (a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_N) \\ \psi_i(b) = (a_1 \oplus 1, \dots, a_{i-1} \oplus 1, 1, a_{i+1} \oplus 1, \dots, a_N \oplus 1) \end{cases} \quad \text{Это означает, что расстояние}$$

Хэмминга между образами вершин равно $h(\psi_i(a), \psi_i(b)) = N$, то есть данное ребро присутствует в $Q_N^{1,N} - \{u, v\}$.

Итак, для любого удалённого ребра $\{u, v\}$ рассматриваемой отказоустойчивой реализации N -куб вкладывается в $Q_N^{1,N} - \{u, v\}$, поэтому, $Q_N^{1,N}$ является 1-рёберной отказоустойчивой реализацией N -куба.

■

Покажем минимальность данной 1-РОТР гиперкуба.

Лемма 10 (О минимальности 1-РОТР $Q_N^{1,N}$ N -куба). 1-РОТР N -куба $Q_N^{1,N}$ является минимальным.

Доказательство. Напомним, что $Q_N^{1,N} = Q_N^1 \cup Q_N^N$. Для произвольной вершины $u = (u_1, \dots, u_N)$ существует ровно одна вершина $\bar{u} = (u_1 \oplus 1, \dots, u_N \oplus 1)$, расстояние Хэмминга до которой равно N , а также N вершин $u_1 = (u_1 \oplus 1, u_2, \dots, u_{N-1}, u_N), \dots, u_N = (u_1, u_2, \dots, u_{N-1}, u_N \oplus 1)$ расстояние до которых равно 1.

Следовательно, степени всех вершин графа $Q_N^{1,N}$ равны $N + 1$. Согласно лемме 3, минимальная степень вершины в М1-РОТР N -куба не может быть меньше $N + 1$. Поскольку это значение достигнуто, граф $Q_N^{1,N}$ является М1-РОТР и содержит 2^{N-1} дополнительных рёбер.

■

3.1 Единственность минимальной 1-рёберной отказоустойчивой реализацией гиперкубов малой размерности

М1-РОТР 2-куба является полным графом K_4 , поэтому оно единственное – нет больше ни одного 4-вершинного графа с таким же числом рёбер.

совпадает с полным графом K_4 , поэтому она единственна: не существует другого графа на 4 вершинах с таким же числом рёбер. Доказательство единственности далее будет проведено отдельно для трёх ситуаций: 3-куба, 4-куба и N -куба при $N \geq 5$. Для этого нам понадобится вспомогательная лемма 11.

Лемма 11. M1-РОтР G^* гиперкуба Q_N при $N > 2$ не содержит рёбер, соединяющих вершины, расстояние между которыми в гиперкубе Q_N равно 2.

Доказательство. Пусть G^* — минимальная 1-рёберная отказоустойчивая реализация $\setminus(N\setminus)$ -куба. Граф G^* получается из Q_N добавлением ровно одного ребра к каждой вершине, иными словами, множество вершин разбивается на пары несмежных вершин, которые затем соединяются дополнительными рёбрами. Допустим, что в G^* нашлось ребро, соединяющее вершины, метки которых в Q_N различаются в двух координатах. Без ограничения общности можно считать, что это вершины $0\dots 000$ и $0\dots 011$. Соответствующая иллюстрация приведена на рисунке 21.

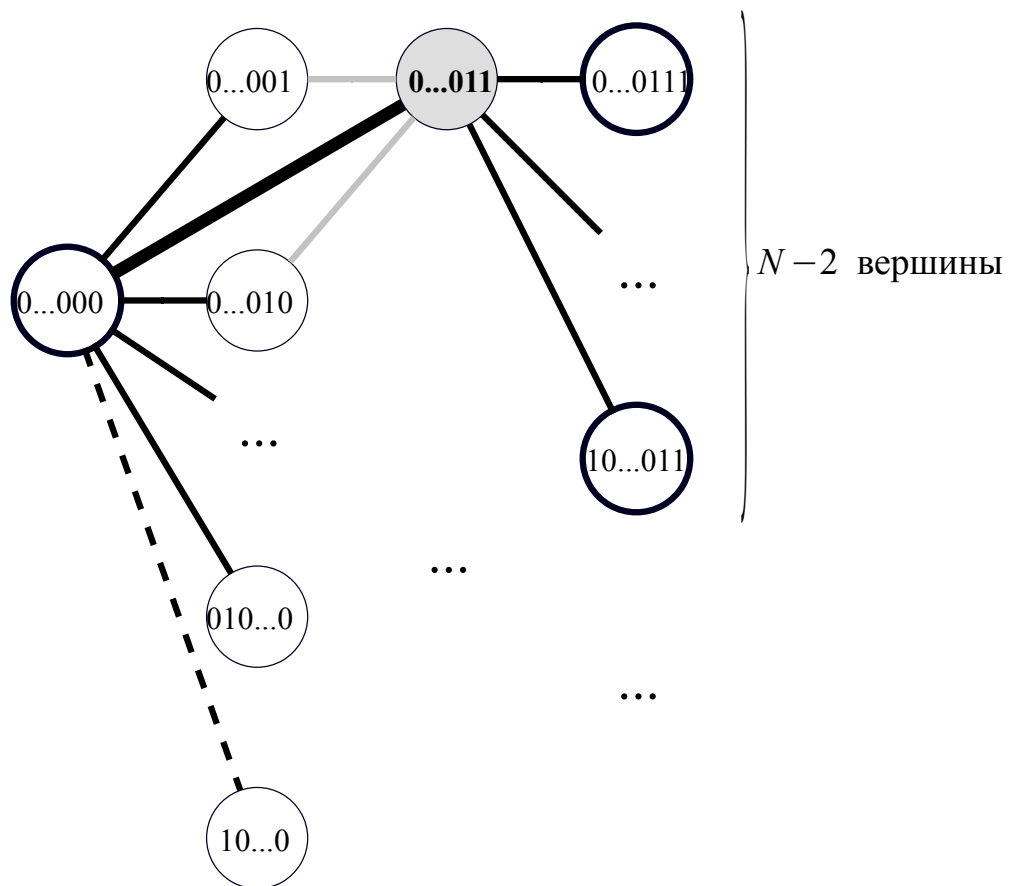


Рисунок 21 – Схема доказательства леммы 11

Добавленное ребро выделено более толстой чёрной линией.

Рассмотрим удаление любого ребра, инцидентного вершине $0\dots00$ и вершине, не смежной с $0\dots011$, например, $10\dots0$ (ребро выделено пунктиром). Обозначим получившийся граф через H .

Попробуем вложить в граф H гиперкуб Q_N . Вершина $0\dots0$ в графе H имеет степень N , следовательно, все смежные с ней вершины: $010\dots0$, $0010\dots0$, ..., $0\dots010$, $0\dots001$ и $0\dots011$ будут образами вершин 1 слоя Q_N .

Вершина $0\dots011$ смежна с $0\dots001$ и $0\dots010$, но все они во вложении являются вершинами типа 1, поэтому среди $N+1$ рёбер вершины $0\dots011$ в отказоустойчивой реализации останутся только $N-1$ рёбер, ведущих к вершинам других типов, что меньше N , а значит, вложение построить невозможно. Из этого следует, что в минимальной 1-рёберной отказоустойчивой реализации гиперкуба, не может быть ребра, соединяющего вершины, различающиеся в двух элементах метки.

■

Из этой леммы следует единственность М1-РОТР 3-куба.

Лемма 12. Граф $Q_3^{1,3}$ является единственным с точностью до изоморфизма М1-РОТР гиперкуба Q_3 .

Доказательство. В гиперкубе Q_3 расстояние между двумя различными вершинами может быть равно 1, 2 и 3. Если расстояние равно 1, то вершины уже смежны, и ребро добавлено быть не может. При добавлении ребра между вершинами на расстоянии 2 по лемме 11 граф не может быть М1-РОТР, а значит, для получения отказоустойчивой реализации остаётся добавлять рёбра между вершинами на расстоянии 3. Для каждой вершины существует только одна на расстоянии 3 парная вершина, поэтому существует единственная с точностью до изоморфизма отказоустойчивая реализация, предложенная в лемме 9.

■

Для 4-куба также существует единственная минимальная 1-рёберная отказоустойчивая реализация, что является результатом следующей леммы.

Лемма 13. Граф $Q_4^{1,4}$ является единственным с точностью до изоморфизма M1-РОТР гиперкуба Q_4 .

Доказательство. В M1-РОТР 4-куба не может быть дополнительного ребра между вершинами на расстоянии 2 по лемме 11. Если все дополнительные рёбра соединяют вершины на расстоянии 4, то получится единственная отказоустойчивая реализация из леммы 9. Следовательно, если существует отказоустойчивая реализация H неизоморфная последнему M1-РОТР, то в нём должно быть как минимум одно ребро между вершинами на расстоянии 3. Без ограничения общности предположим, что это вершины 0000 и 0111 (ввиду дистанционной транзитивности гиперкуба это можно сделать).

Попробуем выяснить, какие ещё рёбра в H должны быть. Для этого будем удалять рёбра и пытаться строить вложение гиперкуба в полученный граф, что позволит определить, какие рёбра также должны быть в графе H при наличии ребра $\{0000, 0111\}$.

Для начала удалим ребро $\{0000, 0001\}$. Пусть вершина первого слоя отображается в вершину типа 0, которая смежна ровно с 4 вершинами 0010, 0100, 1000 и 0111 после удаления ребра $\{0000, 0001\}$ – это будут вершины типа 1. Вершины 0011, 0101, 0110, 1010 и 1100 являются вершинами типа 2, так как каждая из них смежна как минимум с двумя вершинами типа 1. Также 0001, 1011, 1101 и 1110 имеют по 2 смежные вершины, которые однозначно являются вершинами типа 2, поэтому они являются вершинами типа 3.

Данное состояние изображено на рисунке 22. Под пунктом a вершины гиперкуба раскрашены по типам, под пунктом b вершины выставлены столбцами по типам. Пунктиром обозначено удалённое ребро. Белая вершина с чёрной меткой имеет тип 0, светло-серая с чёрной меткой – тип 1, тёмно-серая с чёрной меткой – тип 2, с белой меткой – тип 3. Вершины со

светло-серыми метками и обводкой имеют неопределённый тип. Длинным пунктиром обозначено удалённое ребро.

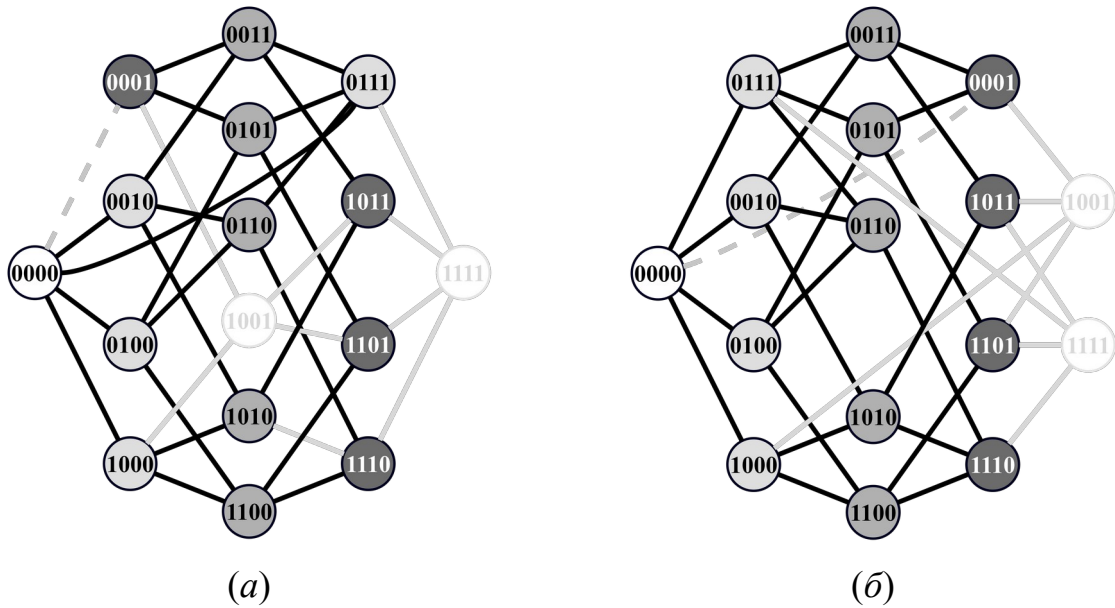


Рисунок 22 – Однозначно определённые типы вершин при вложении гиперкуба в предполагаемое М1-РОТР при удалении ребра $\{0000, 0001\}$ из графа H

Аналогичные изображения для случаев удаления рёбер $\{0000, 0010\}$ и $\{0000, 0100\}$ показаны на рисунке 23 под пунктами a и b соответственно.

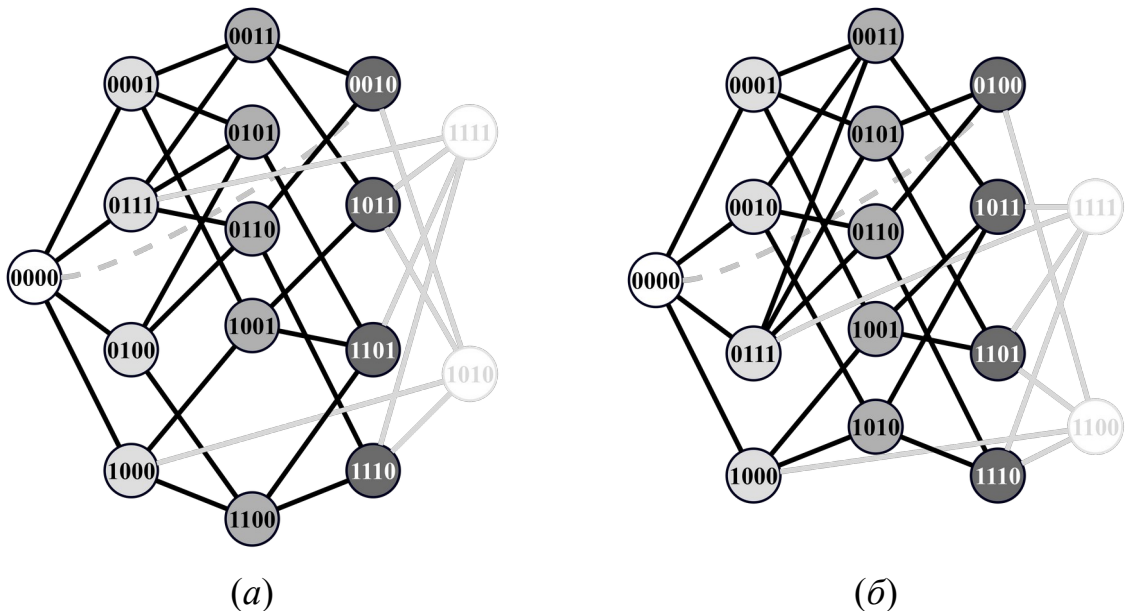


Рисунок 23 – Однозначно определённые типы вершин при вложении гиперкуба в предполагаемое М1-РОТР при удалении рёбер $\{0000, 0010\}$ под

пунктом *a* и {0000, 0100} под пунктом *б* из графа *H*

В гиперкубе каждые две вершины слоя 1 имеют по одной общей смежной вершине со слоя 0 и со слоя 2. Во всех трёх случаях для вершин 0111 и 1000 общей является вершина типа 0, но общих смежных вершин типа 2 нет даже с учётом вершин с неопределённым типом. Так как к вершине 0111 уже добавлено одно ребро, нужно вершину 1000 соединять с одной из смежных с 0111 вершин: 0011, 0101, 0110 или 1111.

Каждая вершина типа 2 должна быть смежна с 2-мя вершинами типа 1 и 2 вершинами типа 3. При удалении ребра {0000, 0001} вершина 0110 смежна только с одной вершиной типа 3, значит, дополнительное ребро должно соединять её с одной из вершин типа 3.

При удалении ребра {0000, 0010} вершина 0101 также смежна только с одной вершиной типа 3, поэтому дополнительное ребро будет её соединять с одной из вершин типа 3. Аналогичная ситуация при удалении ребра {0000, 0100} с вершиной 0011. Так как вершина 1000 имеет тип 1, она не может быть соединена дополнительным ребром ни с одной из этих трёх вершин, поэтому в *H* должно быть ребро {1000, 1111}, и вершина 1111 во всех трёх случаях будет вершиной типа 2, а значит, оставшаяся вершина с неопределённым типом должна иметь тип 4, однако каждая из них имеет только 3 смежных вершины типа 3, а должно быть 4, поэтому можно определить ещё 3 ребра, которые должны быть в *H*: {1001, 1110}, {1010, 1101}, {1100, 1011}.

Осталось определить, как дополнительные рёбра соединяют вершины 0001, 0010, 0100, 0011, 0101 и 0110. Так как ребро может быть добавлено только между вершинами на расстоянии 3 или 4, то единственный возможный вариант — это добавить рёбра {0001, 0110}, {0010, 0101}, {0100, 0011}, так как в других парах расстояние между вершинами равно 1 или 2, что можно видеть на рисунке 24.

	0001	0010	0100	0011	0101	0110
0001	0	2	2	1	1	3
0010	2	0	2	1	3	1
0100	2	2	0	3	1	1
0011	1	1	3	0	2	2
0101	1	3	1	2	0	2
0110	3	1	1	2	2	0

Рисунок 24 – Матрица расстояний между вершинами 0001, 0010, 0100, 0011, 0101 и 0110 в 4-кубе

В итоге, если в графе есть дополнительное ребро $\{0000, 0111\}$, то получится единственный вариант добавления рёбер: $\{1000, 1111\}$, $\{1001, 1110\}$, $\{1010, 1101\}$, $\{1100, 1011\}$, $\{1001, 1110\}$, $\{1010, 1101\}$, $\{1100, 1011\}$. Полученный граф изображён на рисунке 25 под пунктом *а*.

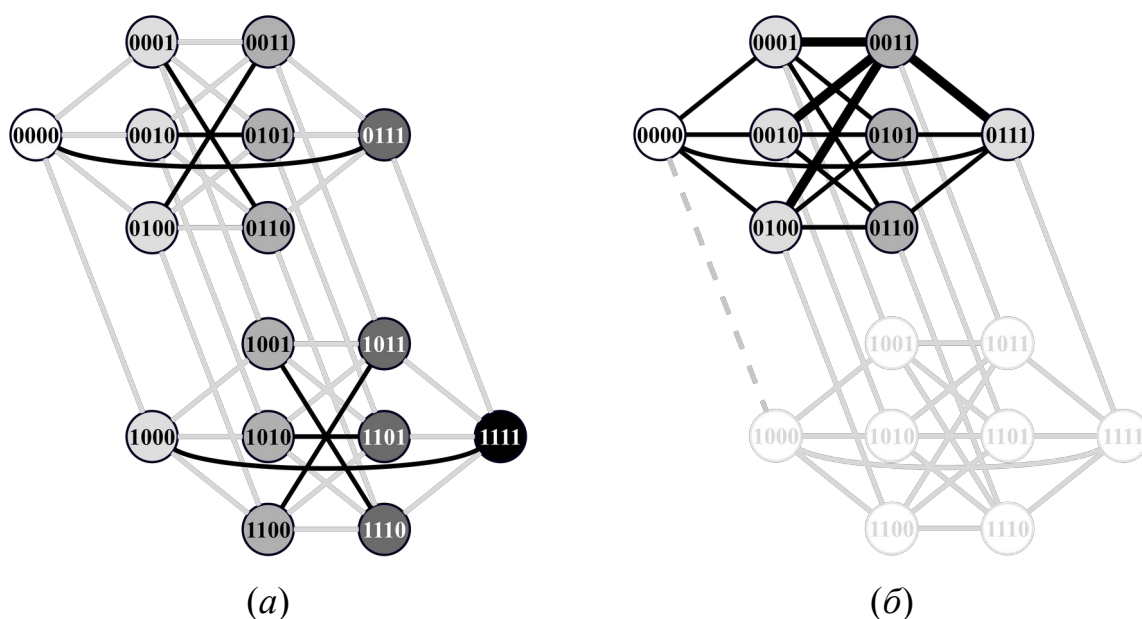


Рисунок 25 – Предполагаемое М1-РОТР 4-куба под пунктом *а* и иллюстрация невозможности построить отказоустойчивую реализацию ввиду существования вершины типа 3, которая смежна с 4-мя вершинами типа 2, под пунктом *б*

При удалении ребра $\{0000, 1000\}$ из данного графа получится ситуация, показанная под пунктом б. Пусть вершина 0000 является вершиной типа 0, тогда 0001, 0010, 0100 и 0111 являются вершинами типа 1. Так как вершина 0011 смежна с 4-мя вершинами типа 1, она не может не являться вершиной типа 2, но вершина типа 2 должна быть смежна с 2-мя вершинами типа 3, что невозможно ввиду наличия у неё 5-ти инцидентных рёбер, 4 из которых соединяют данную вершину с вершинами типа 4. Это означает, что вложение при удалении ребра $\{0000, 1000\}$ построить невозможно, поэтому по определению данный граф не является 1-РОТР 4-куба, следовательно, он не является М1-РОТР.

Учитывая дистанционную транзитивность 4-куба, это справедливо для любого графа, полученного из 4-куба путём добавления хотя бы одного ребра между вершинами на расстоянии 3, а значит, является единственным с точностью до изоморфизма. ■

Таким образом для всех $N < 5$ N -куб имеет единственное с точностью до изоморфизма М1-РОТР.

3.2 Единственность минимальной 1-рёберной отказоустойчивой реализации гиперкуба размерности более 4

Возникает закономерный вопрос, является ли М1-РОТР гиперкуба единственной для N -кубов при $N \geq 5$. Для дальнейшего доказательства необходима лемма 14.

Лемма 14. При $N \geq 5$ в М1-РОТР N -куба нет дополнительных рёбер, соединяющих вершины с метками, расстояние Хэмминга между которыми равно 3.

Доказательство. Введём обозначения:

$$u = 0 \dots 0, v = 0 \dots 0111.$$

$$C = \{0 \dots 0001, 0 \dots 0010, 0 \dots 0100\}.$$

$$W = \{10 \dots 0000, \dots, 00 \dots 0100\}.$$

Далее доказательство будет проиллюстрировано для случая $N = 5$.

1. Пусть между вершинами $u=0\dots 0$ и $v=0\dots 0111$ есть ребро.
2. В $M1\text{-}POtP$ гиперкуба к каждой вершине добавляется ровно одно ребро, поэтому в графе больше нет других дополнительных рёбер, соединяющих u или v с другой вершиной графа.
3. Удалим ребро между u и $w_1=10\dots 0$ и рассмотрим, как будет происходить вложение в отказоустойчивую реализацию.
4. Пусть u имеет тип 0. После удаления инцидентного этой вершине ребра, её степень равна N , поэтому все смежные с ней вершины будут иметь тип 1, в том числе и вершина v . Для Q_5 пример приведён на рисунке 26.

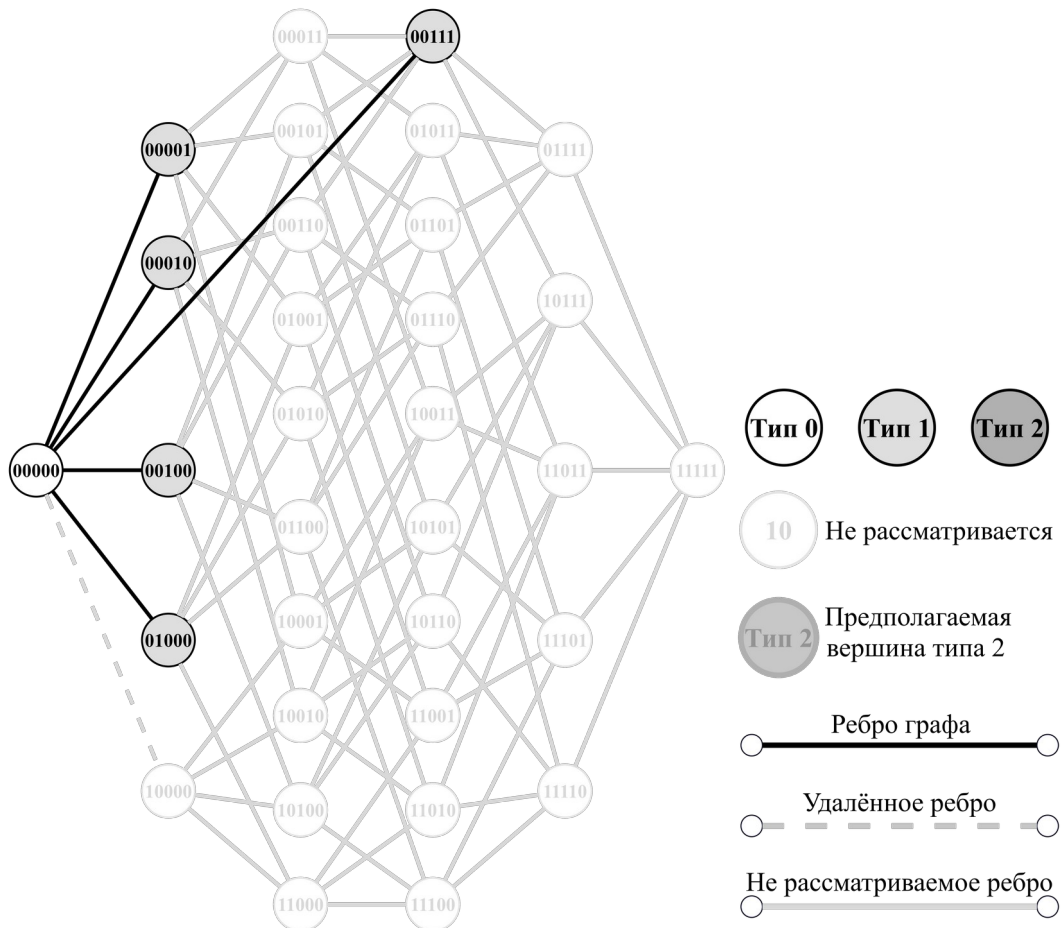


Рисунок 26 – Вершины графа типов 0 и 1 при вложении в предполагаемую отказоустойчивую реализацию 5-куба

5. Так как каждая вершина слоя 2 гиперкуба имеет ровно 2 ребра, соединяющих её с вершинами слоя 1, то, учитывая пункт 2, вершины типа 2 выбираются среди тех вершин, которые смежны с вершинами типа 1. Это все вершины с 2-мя единицами в метке и $N-3$ вершины с 4-мя единицами в метке, которые смежны с v . Данная ситуация для 5-куба проиллюстрирована на рисунке 27.

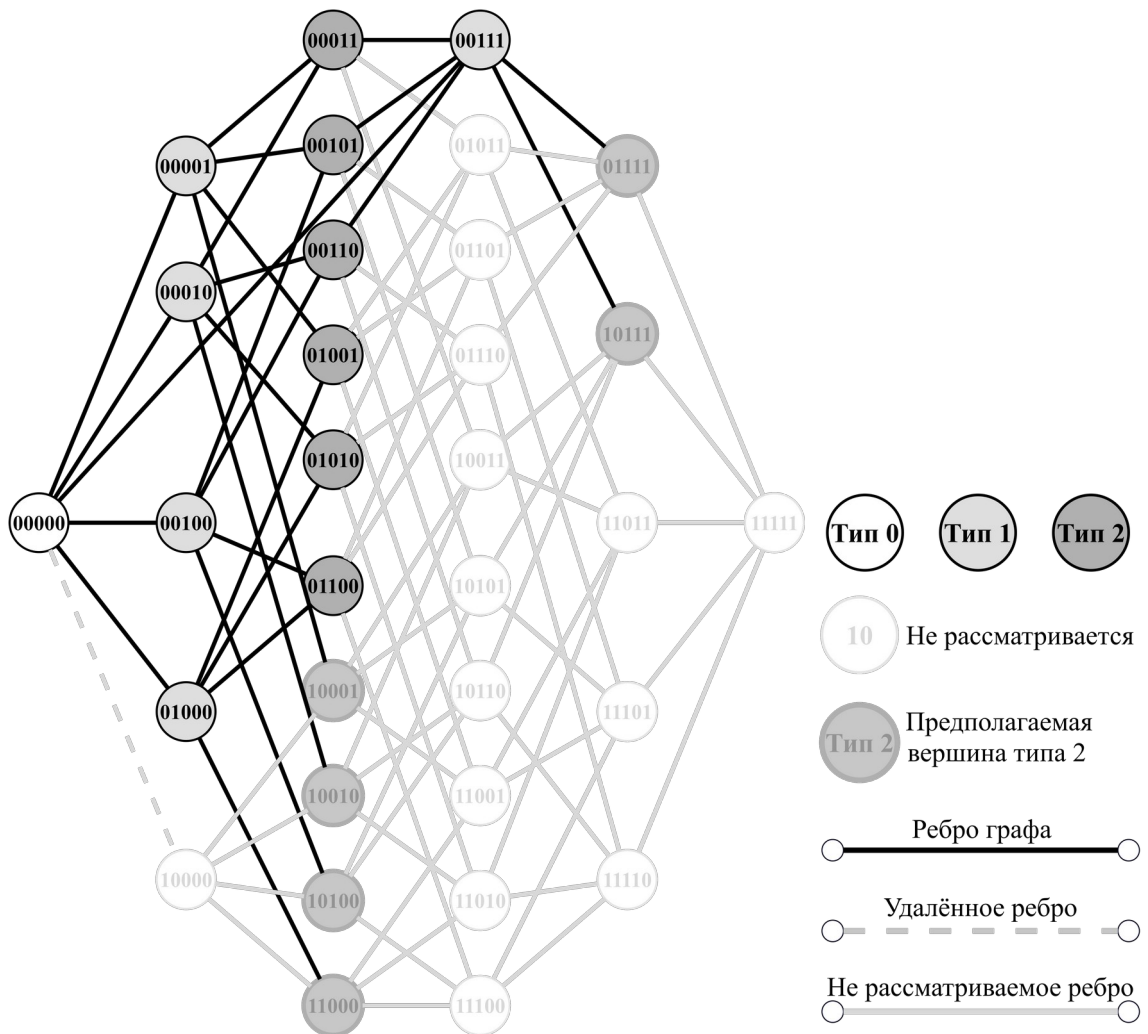


Рисунок 27 – Вершины типов 0, 1 и 2 и предполагаемые вершины типа 2 при вложении в предполагаемую отказоустойчивую реализацию 5-куба с удалённым ребром $\{0\dots 0, 10\dots 0\}$

6. 3 вершины $H_1 = \{0\dots 0110, 0\dots 0101, 0\dots 0011\}$ будут иметь 3 ребра, ведущие к вершинам типа 1: по 2 ребра к вершинам из множества $C = \{0\dots 0001, 0\dots 0010, 0\dots 0100\}$ и ребро к вершине v . Вершины типа 2

должны соединяться $N-2$ рёбрами с вершинами типа 2, но данные вершины имеют только $N-3$ таких ребра, поэтому каждая из этих вершин должна соединяться с ещё одной вершиной типа 3 дополнительным ребром.

7. $N-3$ вершины $00...01111$, ..., $10...00111$ с 4-мя единицами в метке, смежные с v , обозначим как множество H_2 – они смежны только с одной вершиной типа 1 – это вершина v . Среди смежных с v вершин минимум $N-1$ должна иметь тип 2. Учитывая 3 вершины типа 2 из H_1 , которые смежны с v , как минимум $N-4$ вершины из H_2 имеют тип 2.
8. Каждая пара вершин типа 1 должна иметь общую смежную вершину типа 2. Каждая вершина множества H_1 являются общей смежной для v и двух вершин из C . У других вершин типа 1 с одной единицей в метке общих смежных вершин с v нет. Количество таких вершин равно $N-4$ (всего вершин 1-го типа N штук, из них 1 вершина – это v , с 3 вершинами общие смежные у v есть). Это означает, что $N-4$ дополнительных ребра от вершин из H_2 нужно добавлять именно к ним. Обозначим множество этих вершин как W .
9. Покажем, что оставшаяся вершина должна быть соединена с вершиной $10...0$ дополнительным ребром. По пункту 8 $N-4$ ребра были добавлены между вершинами из H_2 и вершинами из W , среди которых есть и $w_2=010...0$. Теперь рассмотрим другой граф, полученный удалением из гипотетического M1-POtP гиперкуба ребра между u и $w_2=010...0$ (изображение для 5-куба показано на рисунке 28). Пункты 1-8 для него рассматриваются аналогично, при этом множество H_2 остаётся таким же, а W изменяется – в нём не будет вершины w_2 , но будет вершина w_1 . Если рассматриваемого ребра в

отказоустойчивой реализации не будет, то у вершин v и w_1 , имеющих тип 1, не будет общей смежной вершины типа 2, а значит, при удалении ребра между u и w_2 вложение построить будет невозможно.

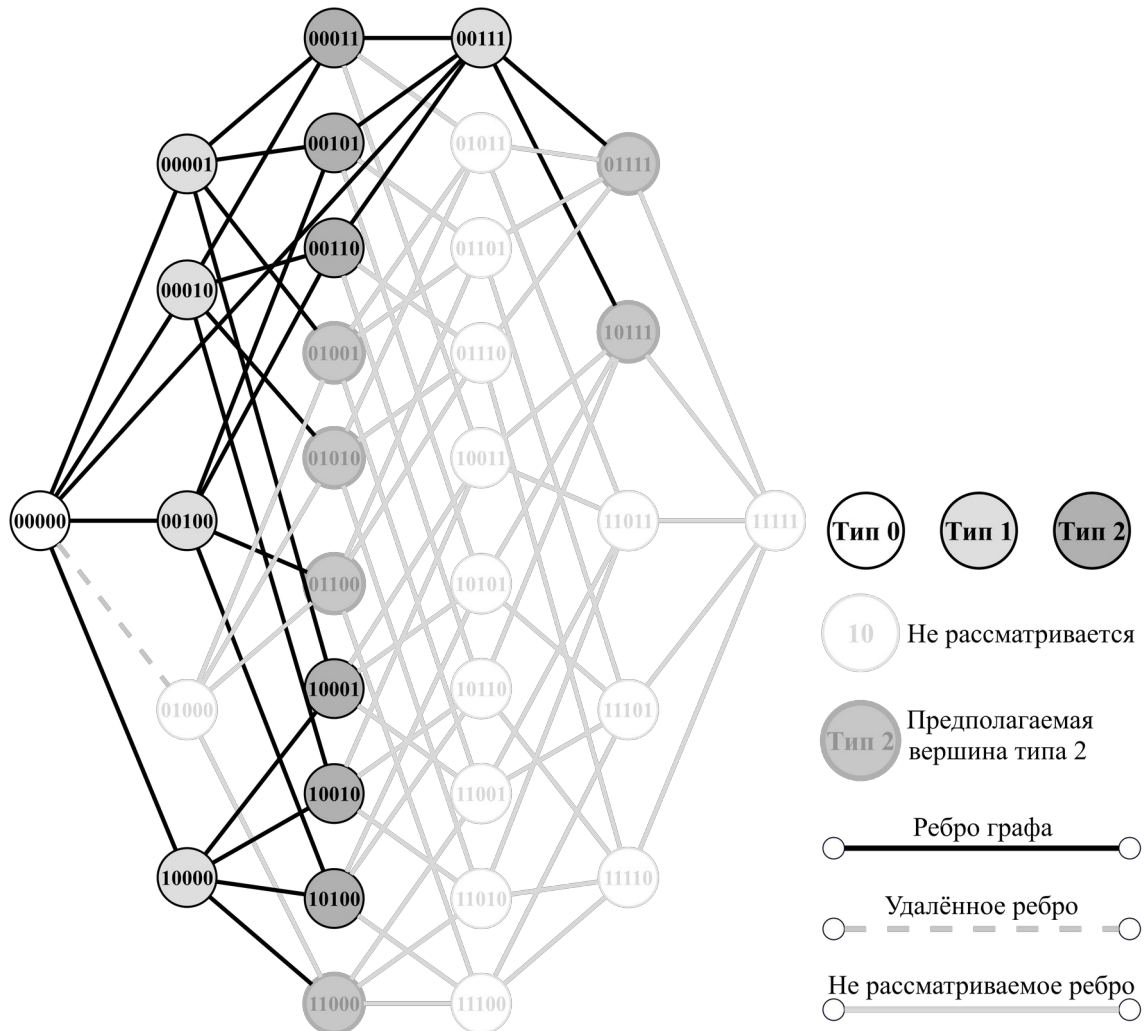


Рисунок 28 – Вершины типов 0, 1 и 2 и предполагаемые вершины типа 2 при вложении в предполагаемую отказоустойчивую реализацию 5-куба с удалённым ребром $\{0\dots 0, 010\dots 0\}$

10. Множество H_3 из $N-1$ вершин, которые смежны с w_1 и имеют 2 единицы в метке, а также имеют ровно 1 ребро, имеющее на другом конце образ вершины слоя 1. Эти вершины не смежны с v и имели в гиперкубе 2 ребра, которые их соединяли с вершинами с 1 единицей в

метке. Так как одной из них являлась w , то осталась ровно 1 смежная вершина типа 1.

11. Остальные вершины H_4 имеют 2 ребра, соединяющие их с вершинами типа 1, поэтому они имеют тип 2. Количество таких вершин равно $C_N^2 - (N-1) - 3 = C_N^2 - C_{N-1}^1 - 3 = C_{N-1}^2 - 3$.

12. Количество вершин, имеющих 2 и более смежных вершин типа 1, равно $(C_{N-1}^2 - 3) + 3 = C_{N-1}^2$ – это вершины множеств H_1 и H_4 . Таким образом, нужно как минимум $C_N^2 - C_{N-1}^2 = C_{N-1}^1 = N-1$ дополнительных рёбер, соединяющие образы вершин слоя 1 и вершины из множества $H_2 \cup H_3$, что равно числу образов вершин слоя 1, к которым можно добавить ребро (к последней вершине-образу v ребро уже было добавлено).

13. Учитывая пункты 8, 9, 10 и 12, получается, что от вершин типа 1 $N-4$ дополнительных ребра будут идти к вершинам из H_2 и 3 ребра – к вершинам из H_3 .

14. Рассмотрим случай удаления ребра $\{u, w_2\}$. По пункту 9 к вершине w_1 будет добавлено ребро к вершинам из H_2 . Таким образом, ко всем вершинам типа 1 уже были добавлены рёбра, но множество H_3 в данном случае будет отличаться. Множество H_3 , которое было получено при рассмотрении первого случая, обозначим как H_3^* . Пересечение этих множеств состоит из одной вершины в силу структуры гиперкуба, то есть $H_3 \cap H_3^* = \{110...0\}$. Так как пункты 1-13 без существенных изменений могут быть перенесены и на случай удаления ребра $\{u, w_2\}$, то по пункту 13 получим, что из H_3^* должно быть 3 ребра к вершинам типа 1, но ко всем этим вершинам уже проведены рёбра и новые добавлять нельзя. Учитывая, что среди этих

вершин может быть только одна $110...0$, к которой уже может быть проведено ребро, то нужно добавить минимум 2 ребра к образам вершин слоя 1, но это сделать невозможно по пункту 2, а значит, если в М1-РОТР гиперкуба присутствует ребро $\{u, v\}$, то отказоустойчивую реализацию построить невозможно.

15. В силу дистанционной транзитивности гиперкуба, это будет справедливо для любой пары вершин $\{u, v\}$, расстояние между которыми равно трём, следовательно, в М1-РОТР N -куба для $N \geq 5$ не должно быть ребра между вершинами, расстояние между которыми равно 3. ■

Опираясь на лемму 14, можно доказать теорему 8 о единственности М1-РОТР гиперкуба.

Теорема 8 (О единственности М1-РОТР гиперкуба). При $N \geq 2$ N -куб имеет единственное с точностью до изоморфизма М1-РОТР.

Доказательство. Для $N = 2$ единственность отказоустойчивой реализации очевидна ввиду того, что его М1-РОТР является полным графом, а других графов с таким же количеством вершин и рёбер не существует.

Случаи с $N = 3$ и $N = 4$ разобраны в леммах 12 и 13 соответственно.

Остался случай $N \geq 5$, который и будет разбираться далее.

N -куб состоит из двух $(N - 1)$ -кубов, соответствующие вершины которых соединены, которые будем называть *подкубами*.

Покажем, что каждое вложение Q_N в построенное добавлением рёбер его М1-РОТР G^* с одним удалённым ребром гиперкуба e отображает один из подкубов Q_N в подкуб оставшейся части Q_N из $G = G^* - e$.

Количество вершин в Q_N и G одинаково. Так как Q_N вершинно-симметричный граф, то вложение вершины v графа Q_N в вершину u графа G приведёт к тому, что будет существовать вложение любой другой

вершины графа в вершину u . Из-за одинакового количества вершин в графах, они все задействуются во вложении, поэтому можно построить другое вложение, при котором выбранная вершина Q_N может быть отображена в произвольную заданную вершину графа G , поэтому достаточно рассмотреть только одну пару таких вершин – остальные результаты можно получить с помощью композиции автоморфизма и вложения.

Слоем k N -куба будем называть вершины, в метках которых ровно k единиц. Так как граф G был построен добавлением рёбер, то у его вершин сохранилась разметка гиперкуба. Вершиной типа k будем называть вершину графа G , в метке которого ровно k единиц.

Для иллюстрации доказательства будут даны рисунки на примере 5-куба.

1. Пусть единственная вершина слоя 0 $u_0 = 0 \dots 0$ гиперкуба вкладывается в вершину $v_0 = 0 \dots 0$ графа G и ребро e инцидентно вершине v_0 .
2. Степени вершин u_0 и v_0 с учётом удалённого ребра равны, поэтому все смежные с u_0 вершины $u_1, \dots, u_N$ будут отображаться в соединённые с v_0 рёбрами гиперкуба вершины $v_1, \dots, v_{N-1}$, смежные с v_0 и v_N , к которой ведёт дополнительное ребро. Занумеруем их так, чтобы u_i отображалась в v_i (рисунок 29 пункт *a*).
3. Покажем, что вершина u_0 и вершины $u_1 \dots, u_{N-1}$ являются частью подкуба. Так как $h(u_0, u_i) = 1$ для $1 \leq i \leq N-1$, то существует ровно $N-1$ координат, в которых данные вершины совпадают с u_0 . Через j обозначим номер оставшейся координаты. Вершины, в индексах которых j -тая координата равна 0, образуют один из подкубов Q_N . Остальные вершины образуют второй подкуб. Метки вершин такие же, как и у Q_N , поэтому при отбрасывании j -той компоненты получаются метки гиперкуба Q_{N-1} . Таким образом, все вершины u_i , $0 \leq i \leq N-1$

образуют 0 и 1 слои подкуба. Данный подкуб будем называть главным. Второй подкуб этого же графа – побочным.

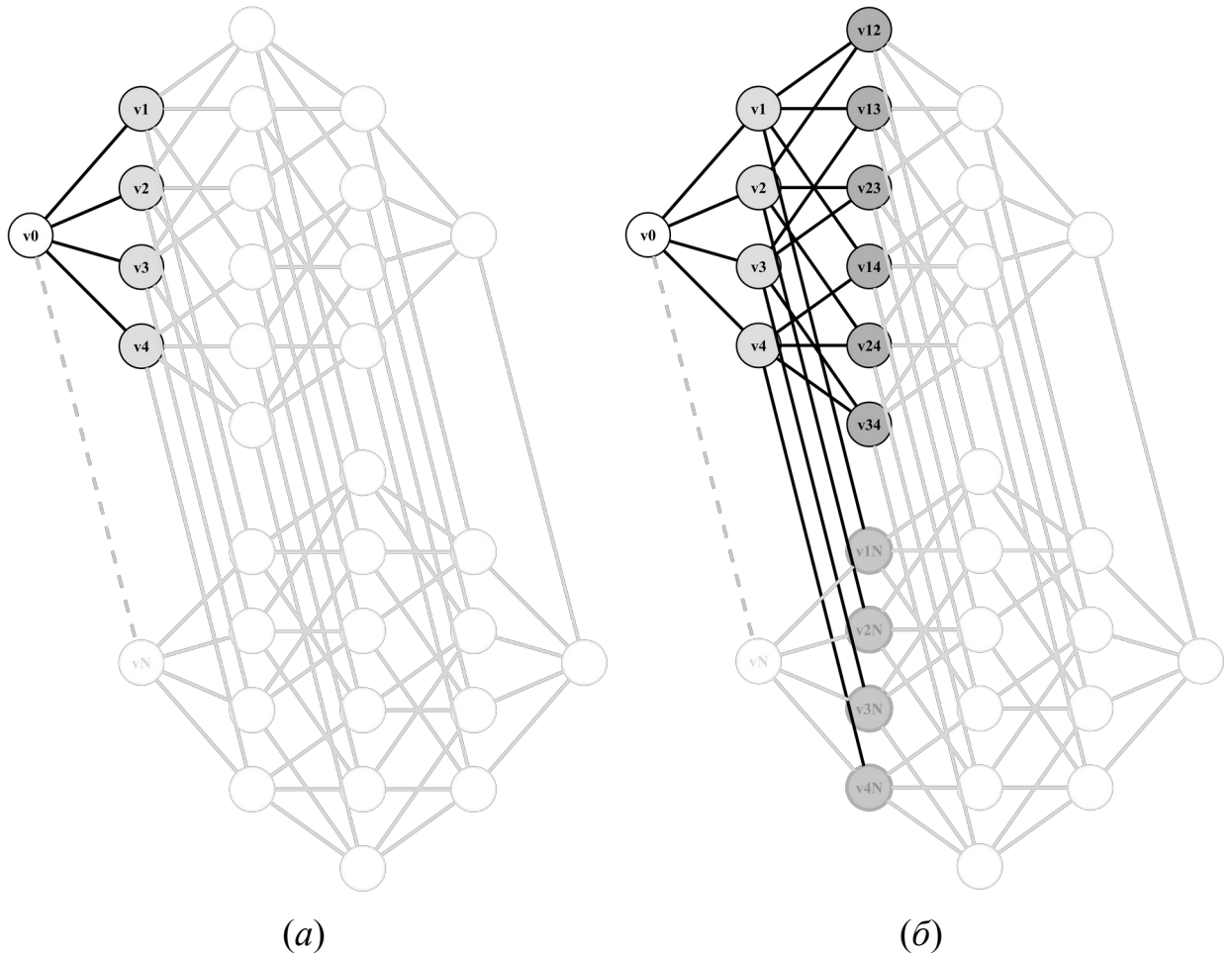


Рисунок 29 – Вершины типов 0 и 1 под пунктом а, и под пунктом б вершины типа 0, 1, 2 и предполагаемые вершины типа 2 при вложении в каждое М1-РОТР 5-куба с удалённым ребром $\{v_0, v_N\}$ при отображении вершины $0 \dots 0$ гиперкуба в v_0

4. Аналогичным образом для графа G и его вершин $v_0, \dots, v_{N-1}$ определим множество E , состоящее из вершин, у которых компонента, различная у вершин на концах ребра e в G^* , одинакова с вершиной v_0 .
5. Обозначим $V_0 = \{v_0\}$ и $V_1 = \{v_1, \dots, v_{N-1}\}$, $U_0 = \{u_0\}$ и $U_1 = \{u_1, \dots, u_{N-1}\}$
6. Рассмотрим вершины, которые могут быть образами вершин слоя 2 главного подкуба. Каждая вершина слоя 2 главного подкуба смежна с двумя вершинами этого подкуба слоя 1. Это означает, что каждый

кандидат на образ такой вершины должен быть смежен с 2-мя вершинами из V_1 . С учётом того, что к вершине можно добавить максимум одно ребро, то в качестве кандидатов могут рассматриваться те незадействованные во вложении вершины, которые соединены ребром оригинального гиперкуба хотя бы с одной вершиной из V_1 , так как второе ребро может быть дополнительным. Среди таких вершин только вершины типа 2 графа G . Вершины типа 2 и вершины V_1 имеют по 2 и 1 единице в метке соответственно, то есть они либо смежны (если позиция одной единицы совпадает), либо имеют расстояние Хэмминга равное 3 (если ни одной единицы не совпадает), поэтому по лемме 14 получим, что дополнительного ребра быть между ними не может, и значит, вершины слоя 2 главного подкуба могут отображаться только в те вершины, которые уже имеют 2 смежных среди вершин из V_1 . Учитывая то, что в гиперкубе каждая вершина соединена лишь с одной вершиной из другого подкуба, то в качестве образов могут быть только вершины из E . Количество образов и прообразов одинаково, поэтому все вершины будут участвовать во вложении. Будем обозначать вершины типа 2 из E как V_2 . На рисунке 29 под буквой b проиллюстрирован данный пункт.

7. Пусть $V_0 \dots V_{k-1}$ являются образами вершин слоёв $0, \dots, k-1$ главного подкуба соответственно при $3 \leq k < N$, тогда V_k — является множеством образов вершин слоя k главного подкуба, где V_k — множество вершин графа G типа k из E . Покажем это. Схема к доказательству проиллюстрирована на рисунке 30 под буквой a .

- 7.1. Вершины слоя k гиперкуба должны быть смежны с k вершинами слоя $k-1$ из G . Учитывая это, вершины не из множества E не

подходят, так как они соединяются с V_{k-1} максимум 2 рёбрами:

одно исходное ребро гиперкуба и, возможно, одно дополнительное.

7.2. Недействованные вершины из E типа $m > k$ могут иметь только 1 ребро.

7.3. Вершины из E типа $m < k$ уже задействованы во вложении.

7.4. Остаются только вершины из E типа k , количество которых равно количеству вершин слоя k главного подкуба.

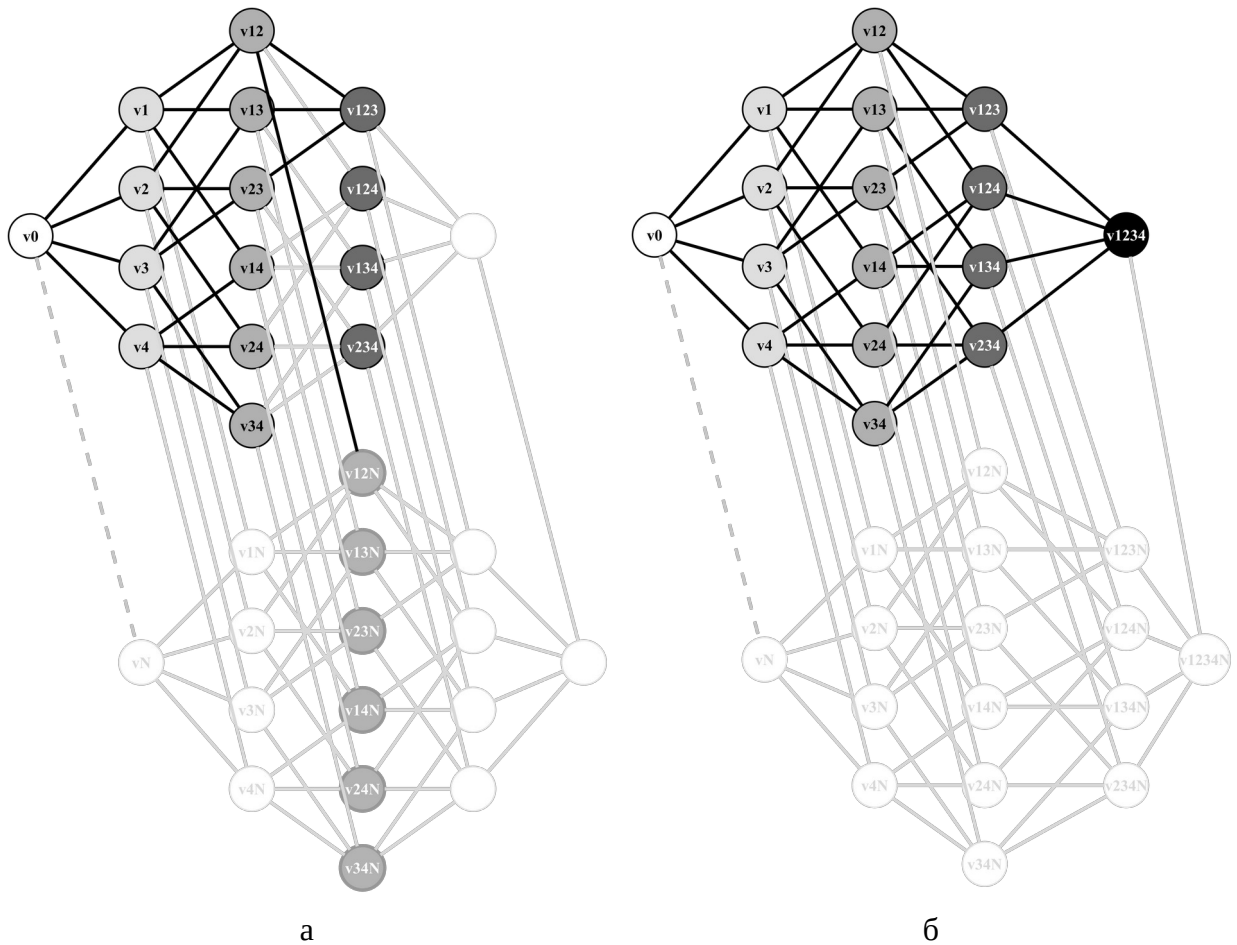


Рисунок 30 – Вершины типов 0, 1, 2 и 3 и предполагаемые вершины типа 2 при вложении в предполагаемую отказоустойчивую реализацию 5-куба с удалённым ребром $\{v_0, v_N\}$

8. Таким образом, при удалении выбранного ребра e из M1-РОТР G^* получили, что при вложении главного подкуба задействуются все вершины из E – множества вершин, у которых совпадает одна

компонента с v_0 . Это означает, что к вершинам из E не может вести дополнительное ребро из v_0 (рисунок 30 буква б).

9. Удаляя другие инцидентные v_0 рёбра, получим те же самые результаты, но при всех возможных равных компонентах метки: 1-ый, 2-ой, ..., N -ный. Это означает, что ни одна вершина из множеств E , формируемых при аналогичном рассмотрении удаления других инцидентных v_0 рёбер, не может соединяться с v_0 дополнительным ребром.

10. В итоге останется единственная вершина, к которой можно добавить ребро от v_0 : та, у которой значения всех компонент отличается от v_0 . Расстояние до неё превышает любое расстояние между двумя вершинами в $(N - 1)$ -кубе.

11. Учитывая вершинную симметричность гиперкуба, аналогичные результаты можно получить и для других вершин, поэтому M1-POtP гиперкуба при $N \geq 5$ единственно с точностью до изоморфизма.

Учитывая, что разобраны все случаи при $N \geq 2$, то для N -куба существует единственное с точностью до изоморфизма M1-POtP, которое описано леммой 9.

0-куб и 1-куб являются полными графами K_0 и K_1 , для которых не существует M1-POtP ввиду невозможности добавления рёбер. ■

Таким образом, M1-POtP гиперкуба единственно с точностью до изоморфизма.

3.3 Связь с полученными численными методами результатами по минимальным рёберным отказоустойчивым реализациям

Ранее в работах Судани (например, [53]) было указано построенное с помощью приложения FTConstructor M1-POtP 16-вершинного гиперкуба. Несмотря на то, что построение заняло менее секунды, следующий шаг

имеет слишком большую комбинаторную сложность, обусловленную двойным ростом числа вершин.

Описанные в данной главе результаты совпали с полученными вычислительным путём результатами, а также полностью описывают все возможные минимальные 1-рёберные отказоустойчивые реализации и полностью закрывают данную задачу.

ЗАКЛЮЧЕНИЕ

В диссертационной работе были получены следующие результаты по алгоритмам и моделям систем, основанных на топологии гиперкуба, устойчивых к отказу узлов и рёбер:

1. Разработана новая математическая модель представления графа, которая позволяет использовать метод канонических представителей и метод Рида–Фараджера для исключения изоморфных копий при построении суперграфов и частей графа.
2. Разработан и обоснован алгоритм построения минимальных k -вершинных отказоустойчивых реализаций, основанный на методе объединения изоморфных графов. Алгоритм реализован в программном комплексе на языке C++ и позволил впервые решить задачу построения всех минимальных 1-вершинных отказоустойчивых реализаций 16-вершинного гиперкуба в ходе вычислительного эксперимента.
3. Получены оптимизированные модели построения 1- и 2-вершинных отказоустойчивых реализаций, в которых количество рёбер на 1 и 2 меньше, чем у тривиальных 1- и 2-вершинных отказоустойчивых реализаций соответственно. Предложенные модели применимы ко всем двудольным графам, отличным от полных двудольных.
4. Найдено и аналитически обосновано исчерпывающее решение задачи построения оптимальной модели системы, устойчивой к отказу одного ребра на основе топологии гиперкуба.

Дальнейшие исследования могут быть направлены на:

1. Изучение графов $Q_N^{1, i_1, \dots, i_t}$ – при каких значениях $i_1, \dots, i_t$ данные графы будут являться минимальными отказоустойчивыми реализациями, а при каких нет.

2. Определение оптимальности описанных в теоремах 3 и 4 моделей при применении к системам, основанным на топологии гиперкубов.

Для дальнейших исследований рекомендуется использовать алгоритмические наработки данной работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абросимов, М. Б. Минимальные k -расширения предполных графов / М. Б. Абросимов // Известия высших учебных заведений. Математика. – 2003. – № 6. – С. 3-11.
2. Абросимов, М. Б. К вопросу о реберных расширениях предполных графов / М. Б. Абросимов // Компьютерные науки и информационные технологии : Материалы Международной научной конференции, Саратов, 01–04 июля 2009 года. – 2009. – С. 3-5.
3. Абросимов, М. Б. О вычислительной сложности расширений графов / М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2009. – № 1. – С. 94-95.
4. Абросимов, М. Б. Минимальные реберные расширения некоторых предполных графов / М. Б. Абросимов // Прикладная дискретная математика. – 2010. – № 1(7). – С. 105-117.
5. Абросимов, М. Б. О сложности некоторых задач, связанных с расширениями графов / М. Б. Абросимов // Математические заметки. – 2010. – Т. 88, № 5. – С. 643-650.
6. Абросимов, М. Б. О нижней оценке числа рёбер минимального рёберного 1-расширения сверхстройного дерева / М. Б. Абросимов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2011. – Т. 11, вып. 3, ч. 2. – С. 111-117.
7. Абросимов, М. Б. О числе дополнительных рёбер минимального вершинного 1-расширения сверхстройного дерева / М. Б. Абросимов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2012. – Т. 12, вып. 2. – С. 103-113.
8. Абросимов, М. Б. Графовые модели отказоустойчивости / М. Б. Абросимов // Саратов: Издательство Саратовского университета. – 2012. – 192 с.

9. Богомолов, А. М. Алгебраические основы теории дискретных систем / А. М. Богомолов, В. Н. Салий // М.: Наука. – 1997. – 368 с.
10. Гавриков, А. В. Т-неприводимые расширения объединений некоторых типов орграфов / А. В. Гавриков // Прикладная дискретная математика. – 2013. – № 4 (22). – С. 47-56.
11. Кабанов, М. А. Об отказоустойчивых реализациях графов / М. А. Кабанов // Теоретические задачи информатики и ее приложений. – Саратов: Издательство Саратовского университета, 1997. – вып. 1. – С. 50-58.
12. Камил, И. А. К. Вычислительный эксперимент по построению отказоустойчивых реализаций графов с числом вершин до 9 / И. А. К. Камил // International Journal of Open Information Technologies. – 2020. – Т. 8, № 9. – С. 43-47.
13. Камил, И. А. К. Методы и алгоритмы разработки вычислительных систем, устойчивых к отказам элементов, без проверки на изоморфизм : дис. ... канд. техн. наук : 05.13.18 / Камил Ихаб Абдулджаббар Камил. – Томск: 2021. – 121 с.
14. Каравай, М. Ф. Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. I. Одно-отказоустойчивые структуры / М. Ф. Каравай // Автоматика и телемеханика. – 2004. – № 12. – С. 159-177.
15. Каравай, М. Ф. Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. II. Решетки и k-отказоустойчивость / М. Ф. Каравай // Автоматика и телемеханика. – 2005. – № 2. – С. 175-189.
16. Каравай, М. Ф. Расширенный обобщенный гиперкуб как отказоустойчивая системная сеть для многопроцессорных систем / М. Ф. Каравай, В. С. Подлазов // Управление большими системами: сборник трудов. – 2013. – № 45. – С. 344-371.
17. Киреева, А. В. Отказоустойчивость в функциональных графах / А. В. Киреева // Упорядоченные множества и решетки. – 1995. – Саратов:

Изд-во Сарат. ун-та. – вып. 11. – С. 32-38.

18. Комаров, Д. Д. Минимальные рёберные расширения палым / М. Б. Абросимов, Д. Д. Комаров // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2013. – Т. 13, вып. 3. – С. 99-104.
19. Комаров, Д. Д. Минимальные вершинные 1-расширения палым / М. Б. Абросимов, Д. Д. Комаров // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2014. – Т. 14, вып. 2. – С. 233-241.
20. Комаров, Д. Д. Верхняя оценка количества дополнительных рёбер минимальных рёберных 1-расширений сверхстройных деревьев / Д. Д. Комаров // Прикладная дискретная математика. – 2015. – 4(30). – С. 91-99.
21. Курносова, С. Г. Т-неприводимые расширения объединений полных графов / С. Г. Курносова // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2005. – Т. 5, вып. 1. – С. 107-115.
22. Курносова, С. Г. Т-неприводимые расширения полных бинарных деревьев / С. Г. Курносова // Вестн. Томск. гос. ун-та. Приложение. – 2005. – № 14. – С. 158-160.
23. Свидетельство о государственной регистрации программы для ЭВМ № 2026614160 Российская Федерация. extvu : № 2026612945. – заявл. 06.02.2026 : опубл. 12.02.2026. / Лобов А. А.
24. Свидетельство о государственной регистрации программы для ЭВМ № 2020610497 Российская Федерация. GenSupg : № 2019666671. – заявл. 19.12.2019 : опубл. 15.01.2020. / Лобов А. А., Абросимов М. Б.
25. Свидетельство о государственной регистрации программы для ЭВМ № 2020614773 Российская Федерация. Построение оптимальных отказоустойчивых реализаций графов FTConstructor : № 2020612581. – заявл. 10.03.2020 : опубл. 24.04.2020. / Лобов А. А., Камил И. А. К., Судани Х. Х. К., Абросимов М. Б.

26. Лобов, А. А. О вершинном 1-расширении гиперкуба / А. А. Лобов, М. Б. Абросимов // Компьютерные науки и информационные технологии : Материалы Международной научной конференции, Саратов, 02–03 июля 2018 года. – Саратов: ИЦ "Наука". – 2018. – С. 249-251.
27. Лобов, А. А. О минимальном рёберном 1-расширении гиперкуба / А. А. Лобов, М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2018. – № 11. – С. 109-111.
28. Лобов, А. А. Оценка количества дополнительных рёбер в минимальном вершинном 1-расширении подкласса двудольных графов [Электронный ресурс] / Материалы Международного молодежного научного форума 1 «ЛОМОНОСОВ-2018». – Отв. ред. И. А. Алешковский, А. В. Андриянов, Е. А. Антипов // М.: МАКС Пресс. – 2018. – 1 электрон. опт. диск (DVD-ROM); 12 см - Систем. требования: ПК с процессором 486+; Windows 95; дисковод DVD-ROM; Adobe Acrobat Reader. — 1450 Мб. — 11000 экз. ISBN 978-5-317-05800-5.
29. Лобов, А. А. Построение минимальных расширений графа методом канонических представителей / А. А. Лобов, И. А. К. Камил, Х. Х. К. Судани, М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2019. – № 12. – С. 179-182.
30. Лобов, А. А. Построение всех неизоморфных минимальных вершинных расширений графа методом канонических представителей / А. А. Лобов, М. Б. Абросимов, И. А. К. Камил // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2019. – Т. 19, № 4. – С. 479-486.
31. Лобов, А. А. Построение минимальных рёберных расширений графа без проверки на изоморфизм / А. А. Лобов, Х. Х. К. Судани, М. Б. Абросимов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика.. – 2020. – Т. 20, вып. 1. – С. 105-115.
32. Лобов, А. А. Построение всех неизоморфных суперграфов без проверки на изоморфизм / А. А. Лобов, И. А. К. Камил, Х. Х. К. Судани,

- М. Б. Абросимов // Прикладная дискретная математика. – 2020. – № 48. – С. 82-92.
33. Лобов, А. А. Построение минимальных вершинных расширений графа методом Рида-Фараджера / А. А. Лобов, И. А. К. Камил, М. Б. Абросимов // International Journal of Open Information Technologies. – 2020. – Vol. 8, № 4. – P. 54-58.
34. Лобов, А. А. О вершинных 2-расширениях 4-слойных графов / М. Б. Абросимов, Лобов А. А. // Компьютерные науки и информационные технологии : Материалы Международной научной конференции, Саратов, 18–20 ноября 2021 года. – 2021. – С. 97-99.
35. Лобов, А. А. О вершинном 1-расширении некоторых торов [Электронный ресурс] / Материалы Международного молодежного научного форума 2 «ЛОМОНОСОВ-2021». – Отв. ред. И. А. Алешковский, А. В. Андриянов, Е. А. Антипов, Е. И. Зимакова // М.: МАКС Пресс. – 2021. – 1 электрон. опт. диск (DVD-ROM); 12 см. – 2000 экз. ISBN 978-5-317-06593-5.
36. Лобов, А. А. О единственности минимального рёберного 1-расширения гиперкуба Q_4 / А. А. Лобов, М. Б. Абросимов // Прикладная дискретная математика. – 2022. – № 58. – С. 84-93.
37. Лобов, А. А. Вершинные расширения 4-слойных графов и гиперкубов / А. А. Лобов, М. Б. Абросимов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2022. – Т. 22, № 4. – С. 536-548.
38. Лобов, А. А. О единственности минимального рёберного 1-расширения гиперкуба / А. А. Лобов, М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2022. – № 15. – С. 110-112.
39. Лобов, А. А. Единственность минимального рёберного 1-расширения гиперкубов / А. А. Лобов // International Journal of Open Information Technologies. – 2023. – Vol. 11, № 9. – P. 28-32.
40. Лобов, А. А. Построение минимальных вершинных расширений графа путём объединения его изоморфных копий / А. А. Лобов // Вестник

ТвГУ. Серия: Прикладная математика. – 2026. – № 1. – С. 79-88.

41. Лобов, А. А. О генерации графов, содержащих заданный подграф / А. А. Лобов, М. Б. Абросимов // Материалы XIII Международного семинара «Дискретная математика и её приложения», имени академика О. Б. Лупанова (Москва, МГУ, 17-22 июня 2019 г.) / Под редакцией О. М. Касим-Заде. – М.: Изд-во МГУ, 2019. – С. 223-226.
42. Моденова, О. В. О точных оценках числа дополнительных дуг минимального вершинного 1-расширения турнира / О. В. Моденова, М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2015. – № 8. – С. 111-113.
43. Моденова, О. В. О минимальных 1-расширениях ориентаций цепей / О. В. Моденова, М. Б. Абросимов // Прикладная дискретная математика. – 2017. – № 38. – С. 89-94.
44. Свидетельство о государственной регистрации программы для ЭВМ № 2021662022 Российская Федерация. ColorGraphExtensions : № 2021661289 : заявл. – 20.07.2021 : опубл. 20.07.2021 / Разумовский П. В., Абросимов М. Б.
45. Разумовский, П. В. Построение цветных графов без проверки на изоморфизм / П. В. Разумовский, М. Б. Абросимов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2021. – Т. 21, № 2. – С. 267-277.
46. Разумовский, П. В. О поиске минимальных вершинных расширений цветного неориентированного графа / П. В. Разумовский, М. Б. Абросимов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2021. – № 4(60). – С. 106-117.
47. Разумовский, П. В. Схемы построения минимальных вершинных 1-расширений полных двухцветных графов / П. В. Разумовский, М. Б. Абросимов // Прикладная дискретная математика. Приложение. – 2021. – № 14. – С. 165-168.
48. Разумовский, П. В. Минимальные вершинные расширения цветных полных графов / П. В. Разумовский, М. Б. Абросимов // Вестник Южно-

- Уральского государственного университета. Серия: Математика. Механика. Физика. . – 2021. – Т. 13, № 4. – С. 77-89.
49. Разумовский, П. В. Минимальные расширения цветных звездных графов / П. В. Разумовский, М. Б. Абросимов // International journal of open information technologies. – 2022. – Т. 10, № 2. – С. 1-7.
 50. Салий, В. Н. Оптимальные реконструкции графов / В. Н. Салий // Саратов : Издательство Саратовского университета. – 2008. – С. 59-65.
 51. Симонов, А. С. Высокоскоростная сеть Ангара: архитектура и результаты применения / А. С. Симонов, И. А. Жабин, Е. Р. Куштанов [и др.] // Вопросы кибербезопасности. – 2019. – № 4(32). – С. 46-53.
 52. Судани, Х. Х. К. Рёберно-отказоустойчивые расширения 8-, 9- и 10-вершинных графов / Х. Х. К. Судани // International Journal of Open Information Technologies. – 2020. – Т. 8, № 8. – С. 46-50.
 53. Судани, Х. Х. К. Модели и алгоритмы построения оптимальных рёберных отказоустойчивых реализаций вычислительных систем : дис. ... канд. техн. наук : 05.13.18 / Судани Хайдер Хуссейн Карим. – Томск: 2021. – 149 с.
 54. Свидетельство о государственной регистрации программы для ЭВМ № 2016610073 Российская Федерация. DSR Generator : № 2015660808 : заявл. – 10.11.2015 : опубл. 11.01.2016 / Сухов, С. А.
 55. Сухов, С. А. О количестве минимальных 1-расширений циклов с числом вершин до 26 и 28 / С. А. Сухов, М. Б. Абросимов, Г. Бринкман // Компьютерные науки и информационные технологии : Материалы Международной научной конференции, Саратов, 30 июня – 02 июля 2016 года – Саратов: ИЦ "Наука". – 2016. – С. 9-11.
 56. Ajima, Y. Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers / Y. Ajima, S. Sumimoto, T. Shimizu // IEEE Computer. – 2009. – Vol. 42, № 11. – P. 36-40.
 57. Ajima, Y. Tofu Interconnect 2: System-on-Chip Integration of High-Performance Interconnect / Y. Ajima, T. Inoue, S. Hiramoto // International Supercomputing Conference / Springer. – 2014. – P. 498-507.

58. Ajima, Y. The Tofu Interconnect D / Y. Ajima, K. Takahiro, T. Okamoto et.al. // 2018 IEEE International Conference on Cluster Computing (CLUSTER). IEEE. – 2018. – P. 646-654.
59. Ajima, Y. High-dimensional Interconnect Technology for the K Computer and the Supercomputer Fugaku [Электронный ресурс] / Y. Ajima // URL: <https://www.fujitsu.com/global/documents/about/resources/publications/technicalreview/topics/article005.pdf> (дата обращения 02.04.2023). – 2020. – 5 с.
60. Alverson, R. The Gemini System Interconnect / R. Alverson, D. Roweth, L. Kaplan // 2010 18th IEEE Symposium on High Performance Interconnects. – 2010. – P. 83-87.
61. Avižienis, A. Fault-tolerance and fault-intolerance : Complementary approaches to reliable computing / A. Avižienis // Proc. Intern. Conf. on Reliable Software. – 1975. – P. 458-464.
62. Biggs, N. L. Distance-Transitive Graphs / N. L. Biggs // Cambridge University Press. – 1993. – P. 155-163.
63. Brinkmann, G. Isomorphism rejection in structure generation programs / G. Brinkmann // Discrete Mathematical Chemistry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. – 2000. – V. 51. – P. 25-38.
64. Bruck, J. Fault-Tolerant Meshes and Hypercubes with Minimal Numbers of Spares / J. Bruck, R. Cypher, C.-T. Ho // IEEE Transactions on computers. – 1993. – Vol. 42, № 9. – P. 1089-1104.
65. Colburn, C. J. Orderly algorithms for generating restricted classes of graphs / C. J. Colburn, R. C. Read // Journal of Graph Theory. – 1979. – Vol. 3. – P. 187-195.
66. Colburn, C. J. Orderly algorithms for graph generation / C. J. Colburn, R. C. Read // Intern. J. Computer Math. – 1979. – Sec. A.7. – P. 167-172.
67. Faradzhev, I. A. Constructive enumeration of combinatorial objects / I. A. Faradzhev // Colloques internationaux C.N.R.S. №260, Problemes Combinatoires et Theorie des Graphes, Orsay. – 1976. – P. 131-135.

68. Harary, F. A survey of the theory of hypercube graphs / F. Harary, J. P. Hayes, H.-J. Wu // *Computers & Mathematics with Applications*. – 1988. – 15 (4). – P. 277-289.
69. Harary, F. Edge fault tolerance in graphs / F. Harary, J. P. Hayes // *Networks*. – 1993. – Vol. 23. – P. 135-142.
70. Harary, F. Node fault tolerance in graphs / F. Harary, J. P. Hayes // *Networks*. – 1996. – Vol. 27. – P. 19-23.
71. Haring, R. A. The IBM Blue Gene/Q Compute Chip / R. Haring, M. Ohmacht, T. W. Fox et. al. // *IEEE Micro*. – 2012. – 32(2). – P. 48-60.
72. Hayes, J. A Microprocessor-based Hypercube Supercomputer / J. Hayes, T. N. Mudge, Q. F. Stout, S. Colley // *IEEE Micro*. – 1986. – 6(5). – P. 6-17.
73. Hayes, J. P. A graph model for fault-tolerant computing system / J. P. Hayes // *IEEE Trans. Comput.* – 1976. – Vol. 25, № 9. – P. 875-884.
74. McKay, B. D. Practical graph isomorphism / B. D. McKay // *Congr. Numer.* – 1980. – Vol. 30. – P. 45-87.
75. McKay, B. D. Isomorphism-free exhaustive generation / B. D. McKay // *Journal of Algorithms*. – 1998. – Vol. 26. – P. 306-324.
76. McKay, B. D. Practical Graph Isomorphism, II / B. D. McKay, A. Piperno // *Journal of Symbolic Computation*, 60. – 2014. – P. 94-112.
77. Meringer, M. Fast Generation of Regular Graphs and Construction of Cages / M. Meringer // *Journal of Graph Theory*. – 1999. – № 30. – P. 137-146.
78. Meuer, H. TOP 500 The List [Электронный ресурс] / H. Meuer, E. Strohmaier, J. Dongarra, H. Simon // URL: <https://www.top500.org> (дата обращения 09.10.2026).
79. Neuwirth, S. Scalable communication architecture for network-attached accelerators / S. Neuwirth., D. Frey, M. Nuessle, U. Bruening // *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium / IEEE*. – 2015. – P. 627-638.
80. Ray-Shung, C. 1-edge fault-tolerant designs for meshes / C. Ray-Shung, H. Lih-Hsing // *Parallel Processing Letters*. – 1994. – Vol. 4, № 4. – P. 385-389.

81. Razumovsky, P. V. The search for minimal edge 1-extension of an undirected colored graph / P. V. Razumovsky // *Izvestiya of Saratov University. New Series. Series: Mathematics. Mechanics. Informatics.* – 2021. – Vol. 21, № 3. – P. 400-407.
82. Read, R. C. Every one a Winner or how to Avoid Isomorphism Search when Cataloguing Combinatorial Configurations / R. C. Read // *Annals of Discrete Mathematics.* – 1978. – Vol. 2. – P. 107-120.
83. Shantanu, D. Designing fault-tolerant systems using automorphisms / D. Shantanu, J. P. Hayes // *Journal of Parallel and Distributed Computing.* – 1991. – 12(3). – P. 249-268.
84. Sloane, N.J.A. The on-line encyclopedia of integer sequences [Электронный ресурс] // URL:<https://oeis.org> (дата обращения 01.09.2026).
85. Sung, T. Y. Optimal k-fault-tolerance network for token rings / T. Y. Sung, T. Y. Ho, C. P. Chang, L. H. Hsu // *J. Inform. Science and Engineering.* – 2000. – № 16. – P. 381-390.
86. Wang, J.-C. Scheduling of Unstructured Communication on the Intel iPSC/860 [Электронный ресурс] / J.-C. Wang, S. Ranka // *School of Computer and Information Science* // URL:https://surface.syr.edu/lcsmith_other/48 (дата обращения 01.09.2026).

Приложение А

Свидетельство о государственной регистрации программы для ЭВМ GenSupg

В данном приложении приведено изображение свидетельства о государственной регистрации программы для ЭВМ «GenSupg», предназначенной для генерации попарно неизоморфных суперграфов заданного графа. Данная программа является первой реализацией метода построения суперграфов, основанного на техниках исключения изоморфных копий без непосредственной проверки на изоморфизм.

Программа умеет разбивать задачи на подзадачи, и в ней реализована функция запуска для решения конкретной подзадачи, что позволяет её использовать как на системах общей памяти, так и на системах с распределённой памятью.

Сама программа написана на языке С.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020610497

GenSupg

Правообладатель: **Федеральное государственное бюджетное образовательное учреждение высшего образования «Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского» (RU)**

Авторы: **Лобов Александр Андреевич (RU),
Абросимов Михаил Борисович (RU)**



Заявка № **2019666671**

Дата поступления **19 декабря 2019 г.**

Дата государственной регистрации
в Реестре программ для ЭВМ **15 января 2020 г.**

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

Приложение Б

Свидетельство о государственной регистрации программы для ЭВМ FTConstructor

В данном приложении приведено изображение свидетельства о государственной регистрации программы для ЭВМ «Построение оптимальных отказоустойчивых реализаций графов FTConstructor».

FTConstructor предназначен для построения вершинных и рёберных отказоустойчивых реализаций графов с помощью техник исключений изоморфных копий без проверок на изоморфизм, основанных на методах канонических представителей и Рида-Фараджа. Сами эти методы используются в одной из подзадач (генерации суперграфов) и особую математическую модель представления графа. Программа поддерживает параллельные вычисления на вычислительных системах с распределённой памятью с помощью технологии MPI и реализованы на языке C++.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020614773

**Построение оптимальных отказоустойчивых реализаций
графов FTConstructor**

Правообладатели: *Абросимов Михаил Борисович (RU), Камил Ихаб
Абдулджаббар Камил (RU), Судани Хайдер Хуссейн Карим (RU),
Лобов Александр Андреевич (RU)*

Авторы: *Абросимов Михаил Борисович (RU), Камил Ихаб
Абдулджаббар Камил (RU), Судани Хайдер Хуссейн Карим (RU),
Лобов Александр Андреевич (RU)*

Заявка № 2020612581

Дата поступления 10 марта 2020 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 24 апреля 2020 г.



Руководитель Федеральной службы
по интеллектуальной собственности

 Г.П. Ивлиев

Приложение В

Свидетельство о государственной регистрации программы для ЭВМ extvu

В приложении приведено изображение свидетельства о государственной регистрации программы для ЭВМ «extvu». Данная программа решает задачу построения минимальных k -вершинных отказоустойчивых реализаций графа на основе метода объединения его изоморфных копий. В программе применяются описанные в работе оптимизации по сокращению перебора с использованием автоморфизмов графа.

Программа умеет разбивать задачи на подзадачи, и в ней реализована функция запуска для решения конкретной подзадачи, что позволяет её использовать как на системах общей памяти, так и на системах с распределённой памятью.

Алгоритм, реализованный в данной программе позволил построить все 1-вершинные отказоустойчивые реализации 16-вершинного гиперкуба за примерно 40 минут на одном ядре ноутбука с процессором AMD Ryzen 7 7840hs. Данную задачу ранее не удавалось решить другими алгоритмами, в том числе и теми, что реализованы в программе FTConstructor.

Сама программа реализована на языке C++.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2026614160

extvu

Правообладатель: *Лобов Александр Андреевич (RU)*

Автор(ы): *Лобов Александр Андреевич (RU)*



Заявка № 2026612945

Дата поступления 06 февраля 2026 г.

Дата государственной регистрации
в Реестре программ для ЭВМ 12 февраля 2026 г.

Руководитель Федеральной службы
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 00a570e477aadd8d531b4b8818e75f29506
Владелец **Зубов Юрий Сергеевич**
Действителен с 04.09.2025 по 28.11.2026

Ю.С. Зубов