

Российская академия наук

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ИНСТИТУТ МАТЕМАТИКИ ИМ. С.Л. СОБОЛЕВА СИБИРСКОГО ОТДЕЛЕНИЯ
РОССИЙСКОЙ АКАДЕМИИ НАУК

(ИМ СО РАН)

УДК 519.7

№ госрегистрации 01201064560

Инв. №

УТВЕРЖДАЮ

И.о. директора
член-корреспондент РАН

_____ Гончаров С.С.

«13» июня 2012 г.

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

В рамках федеральной целевой программы «Научные и научно-педагогические кадры
инновационной России» на 2009-2013 годы

по государственному контракту № 14.740.11.0362
шифр заявки «2010-1.1-113-130-032»

по теме:

СОВРЕМЕННЫЕ ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ

Наименование этапа: «Проведение фундаментальных исследований»
(промежуточный, этап № 4)

Руководитель НИР,
член-корреспондент РАН

В.Д. Мазуров

Новосибирск 2012

СПИСОК ИСПОЛНИТЕЛЕЙ

Рук. темы, зав. отделом ИМ СО РАН, член-корр. РАН	_____	В.Д. Мазуров (Введение, Заключение)
Отв. исполнитель темы, исп. директор НОЦ, д.т.н.	_____	С.М. Лавлинский (Реферат, Приложения А-Б)
проф. НГУ, д.ф.-м.н.	_____	Береснев В.Л. (раздел 1.2)
проф. НГУ, д.ф.-м.н.	_____	Гимади Э.Х. (раздел 1.1)
зав. кафедрой НГУ, д.ф.-м.н.	_____	Ерзин А.И. (раздел 1.9)
гл.н.с. ИМ СО РАН, д.ф.-м.н.	_____	Кельманов А.В. (раздел 1.4)
в.н.с. ИМ СО РАН, д.ф.-м.н.	_____	Соловьева Ф.И. (раздел 1.7)
в.н.с. ИМ СО РАН, д.ф.-м.н.	_____	Севастьянов С.В. (раздел 1.3)
с.н.с. ИМ СО РАН, к.ф.-м.н.	_____	Кононов А.В. (раздел 1.3)
н.с. ИМ СО РАН, к.ф.-м.н.	_____	Алексеева Е.В. (раздел 1.2)
н.с. ИМ СО РАН, к.ф.-м.н.	_____	Орозбеков Н.А. (раздел 1.2)
инж. ИМ СО РАН, к.ф.-м.н.	_____	Тахонов И.И. (раздел 1.10)
зав. лабораторией ИМ СО РАН, к.ф.-м.н.	_____	Августинович С.В. (раздел 1.5, 1.6)
асс. НГУ, к.ф.-м.н.	_____	Токарева Н.Н. (раздел 1.7)
доц. НГУ, д.ф.-м.н.	_____	Кротов Д.С. (раздел 1.6,1.5)
с.н.с. ИМ СО РАН, к.ф.-м.н.	_____	Потапов В. Н. (раздел 1.6)
асс. НГУ, к.ф.-м.н.	_____	Могильных И.Ю. (раздел 1.6)

	_____	Васильева А.Ю. (раздел 1.7)
с.н.с. ИМ СО РАН, к.ф.-м.н.	_____	Горкунов Е.В. (раздел 1.7)
асс. НГУ, к.ф.-м.н.	_____	
аспирант ИМ СО РАН	_____	Коломиец Н. А. (раздел 1.7) Батуева Ц.Ч. (раздел 1.6)
асс. НГУ, к.ф.-м.н.	_____	
аспирант ИМ СО РАН	_____	Павлов С.В. (раздел 1.3)
аспирант ИМ СО РАН	_____	Сухорослов А.А. (раздел 1.3)
асп. НГУ	_____	Плотников Р.В. (раздел 1.10)
аспирант ИМ СО РАН	_____	Романченко С.М. (раздел 1.4)
аспирант ИМ СО РАН	_____	Мельников А.А. (раздел 1.2)
студент НГУ	_____	Сотникова Е.В. (раздел 1.7)
студент НГУ	_____	Валюженич А.А. (раздел 1.5)
студент НГУ	_____	Семина Ю.Д. (раздел 1.5)
студент НГУ	_____	Паршина О.Г. (раздел 1.7)
студент НГУ	_____	Хандеев В.И. (раздел 1.4)
Нормоконтролер	_____	Кравченко С.В.

Реферат

Отчет 97 с., 1 ч., 89 источников, 4 табл., 2 прил.

Тема: СОВРЕМЕННЫЕ ПРОБЛЕМЫ ТЕОРЕТИЧЕСКОЙ КИБЕРНЕТИКИ

Ключевые слова: ЗАДАЧА КОММИВОЯЖЁРА С НЕСКОЛЬКИМИ КОММИВОЯЖЁРАМИ, АЛГОРИТМЫ ЛОКАЛЬНОГО ПОИСКА С ОБОБЩЕННОЙ ОКРЕСТНОСТЬЮ, ЭФФЕКТИВНЫЕ АЛГОРИТМЫ С ОЦЕНКАМИ ТОЧНОСТИ ДЛЯ ЗАДАЧ КЛАСТЕРНОГО АНАЛИЗА, БЕНТ-ФУНКЦИИ, БЛОЧНЫЙ ШИФР, КОММУНИКАЦИОННОЕ ДЕРЕВО В БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЯХ.

Основным объектом исследования являются актуальные проблемы теоретической кибернетики.

Основной целью проекта является получение научных результатов мирового уровня, позволяющих закрепить приоритет российской школы теоретической кибернетики, повысить уровень подготовки и способствовать закреплению в сфере науки и образования научных и научно-педагогических кадров, а также формированию эффективных и жизнеспособных научных коллективов.

В процессе работ использовались классические методы кластерного анализа, методы теории кодирования, методы оптимизации и дискретного анализа, аппарат теории расписаний.

В результате фундаментальных исследований 4 этапа получены новые результаты мирового уровня.

1) Найдена нижняя оценка числа бент-функций на минимальном расстоянии от бент-функции из класса Мэйорана – МакФарланда.

2) В терминах преобразования Фурье введено понятие реконструктивного множества в булевом кубе. Получена характеристика реконструктивных множеств, являющихся линейными подпространствами. Установлены необходимые и достаточные условия реконструктивности сферы. Приведено достаточное условие реконструктивности двух концентрических сфер.

3) Спектром гамильтонова цикла (кода Грея) в булевом n -мерном кубе называется набор $a = (a_1, \dots, a_n)$, где a_i – число рёбер i -го направления в цикле. Известны необходимые условия существования кода Грея со спектром a : числа a_i чётные и для любого $k = 1, \dots, n$ сумма k произвольных компонент набора a не меньше чем $2k$. Доказано существование такой размерности N , что если необходимые условия на спектр являются достаточными для

существования гамильтонова цикла с таким спектром в булевом N -мерном кубе, то сформулированные выше условия являются достаточными и для всех размерностей n .

4) Установлено, что сложность реализации в классе обобщённых (троичных) π -схем троичного счётчика кратности 3, зависящего от трёх переменных, равна 18.

5) Подмножество вершин графа называется k -кратным совершенным кодом радиуса r , если для каждой вершины шар радиуса r с центром в этой вершине содержит в точности k кодовых вершин. Получен критерий, который по параметрам совершенной 2-раскраски двоичного n -куба определяет, является ли она кратным совершенным кодом заданного радиуса $r > 1$ некоторой кратности.

6) Для NP-трудной в сильном смысле задачи построения оптимального коммуникационного дерева в беспроводной сенсорной сети найдены частные случаи полиномиальной разрешимости; показано, что минимальный остов, веса рёбер которого

принадлежат отрезку $[a, b]$, является $\left(2 - \frac{2a}{a + b + 2b/(n-2)}\right)$ -приближенным решением и, что задача построения 1,00048-приближенного решения NP-трудна; предложен эвристический полиномиальный алгоритм и осуществлён его апостериорный анализ.

7) Доказана NP-полнота нескольких актуальных задач разбиения последовательности векторов, содержащих конечное число элементов, по критерию минимума суммы квадратов расстояний.

8) Предложены 2-приближённые полиномиальные алгоритмы, а также точные псевдополиномиальные алгоритмы для ряда труднорешаемых задач поиска подпоследовательностей векторов.

9) Для задачи об инвестировании проектов с одновременным поступлением проектов в систему построен алгоритм, основанный на идее многопараметрического динамического программирования. Трудоёмкость алгоритма становится псевдополиномиальной в случае, когда число (m) различных моментов поступления проектов в систему ограничено константой. Доказана неувлучшаемость полученного результата: невозможность избавиться от псевдополиномиального характера оценки трудоёмкости (поскольку даже при $m=2$ задача NP-трудна) и невозможность построения аналогичного псевдополиномиального алгоритма для случая неограниченного m , поскольку в этом случае задача NP-трудна в сильном смысле.

Кроме того, для частного случая задачи, когда вклад каждого проекта в общий ресурсный пул неотрицателен, разработан полиномиальный алгоритм точного решения. В то же время, «симметричная» задача, в которой вклад каждого проекта в общий ресурсный пул неположителен, является (как показано нами) NP-трудной в сильном смысле, а значит, вряд

ли допускает точное решение за полиномиальное время (если верна гипотеза о несовпадении классов P и NP).

10) Установлена полиномиальная разрешимость смешанной задачи $(J + \bar{O})|p_{ij} = 1, \eta \leq 2|C_{\max}$, т.е. задачи, обобщающей классические задачи job shop и обобщённую задачу open shop (на минимум длины расписания) на случай, когда в примере могут присутствовать работы обоих типов (как job shop, так и open shop). При этом рассматривается частный случай этого обобщения, когда длины всех операций равны 1 либо 0, а каждая работа имеет не более двух операций. Установлено также, что данный результат является наилучшим возможным для задач такого типа (в плане построения эффективного алгоритма точного решения), поскольку любая попытка ослабления какого-либо из ограничений на входные данные задачи приводит к потере свойства её эффективной разрешимости.

11) Установлено, что оптимальное расписание в задаче job shop с разрешением прерываний, на минимум произвольной регулярной функции от моментов завершения операций может быть найдено простым жадным алгоритмом, при условии, что найдены (угаданы) подходящие приоритеты выполнения операций на каждой из машин. Таким образом, впервые описан конструктивный метод решения данной задачи.

12) Построен алгоритм A_Δ , который для любого заданного примера двухмашинной задачи flow shop с n работами и для любого заданного $\Delta \geq 0$ перечисляет (без повторений) все Δ -оптимальные перестановки из n работ, затрачивая не более чем $O(n \log n)$ единиц времени на отыскание каждой Δ -оптимальной перестановки и на остановку алгоритма после отыскания последнего решения. При этом на каждом шаге алгоритма используется не более чем $O(n)$ ячеек памяти вычислительной машины. Таким образом, алгоритм эффективен, поскольку имеет полиномиальную задержку и обходится полиномиальной памятью.

Установлено свойство связности множества Δ -оптимальных перестановок: доказано, что любая Δ -оптимальная перестановка может быть получена из любой другой Δ -оптимальной перестановки последовательной транспозицией двух соседних элементов, так что при этом все промежуточные перестановки также Δ -оптимальны.

13) Рассмотрена цеховая задача открытого типа с маршрутизацией, которая является обобщением двух классических задач дискретной оптимизации: цеховой задачи открытого типа $(O \| C_{\max})$ и метрической задачи коммивояжёра. Для различных версий задачи построены быстрые алгоритмы, строящие приближённые решения с гарантированными оценками точности:

а) для задачи с двумя машинами и двумя вершинами предложена вполне полиномиальная приближённая схема (FPTAS), то есть семейство алгоритмов, которые для

произвольного фиксированного ε строят $(1+\varepsilon)$ -приближённые решения за время, ограниченное полиномом от размера входа задачи и от величины $\frac{1}{\varepsilon}$;

б) для задачи на двух машинах на произвольной транспортной сети построен $13/8$ -приближённый алгоритм; в случае, когда задача коммивояжёра на транспортной сети может быть решена точно, оценку точности этого алгоритма можно улучшить до $4/3$;

в) для задачи с нефиксированным числом вершин на произвольной транспортной сети предложен $O(\sqrt{m})$ -приближённый алгоритм.

Степень внедрения - результаты используются в образовательном процессе Новосибирского государственного университета при чтении таких курсов лекций, как «Исследование операций», «Совершенные структуры», «Теория расписаний», «Анализ данных и распознавание образов».

Полученные результаты фундаментального характера, прежде всего, являются вкладом в общую математическую теорию. Результаты исследований могут быть использованы в практической сфере, связанной с процессами управления.

Эффективность и значимость работ, помимо чисто научных результатов, заключается в подготовке молодых ученых, непосредственно участвовавших в работах наряду с признанными специалистами, и способствуют закреплению в сфере науки и образования научных и научно-педагогических кадров.

В развитии результатов четвертого этапа в последующих работах этого направления следует ожидать формирование эффективного инструментария исследования проблем теоретической кибернетики, использующего сформулированные подходы и новые постановки ключевых задач.

В результате исследований по ряду направлений получены новые фундаментальные результаты мирового уровня, которые доложены на различных научных форумах и опубликованы в статьях в высокорейтинговых журналах.

Обозначения и сокращения

ИМ СО РАН - Институт математики Сибирского отделения Российской академии наук.

НГУ – Новосибирский государственный университет.

НОЦ – научно-образовательный центр.

СБИС – сверхбольшие интегральные схемы.

Содержание

	Введение	11
1	Проведение фундаментальных исследований	12
1.1	Построение оценок точности алгоритмов решения задачи коммивояжёра с несколькими коммивояжёрами	12
1.1.1	Введение	12
1.1.2	Задача 2-PSP-max, алгоритм A7/9	13
1.1.3	Алгоритм с оценкой $(3q+2)/(4q+1)$ для 2-PSP-max $[1,q]$	14
1.1.4	Алгоритм A7/5 для задачи 2-PSP(1,2)-min-2w	14
1.1.5	Алгоритм A4/3 для задачи 2-PSP(1,2)-min-2w	15
1.1.6	Оценка точности для задачи 2-PSP(1,2)-max-2w с разными весовыми функциями ребер в маршрутах	16
1.1.7	Обоснование полиномиальности и условий асимптотической точности алгоритма решения задачи m-PSP на максимум в многомерном евклидовом пространстве R^k	17
1.1.8	Задача маршрутизации с ограничениями на пропускные способности ребер графа	17
1.1.9	Задача k-VRP – маршрутизации многих транспортных средств (ТС) с ограниченным числом клиентов в каждом маршруте	18
1.2	Разработка и численный анализ алгоритмов локального поиска с обобщенной окрестностью для построения приближенных решений задач конкурентного размещения	19
1.2.1	Обобщенная окрестность	19
1.2.2	Алгоритм локального поиска по обобщенной окрестности для задачи конкурентного размещения предприятий	22
1.3	Разработка эффективных алгоритмов приближённого решения задач теории расписаний	28
1.3.1	Задача об инвестировании проектов	29
1.3.2	Задачи JOB SHOP, OPEN SHOP и смешанная	31
1.3.3	Задача JOB SHOP с разрешением прерываний, на минимум произвольной регулярной функции от моментов завершения операций	33
1.3.4	Эффективный алгоритм перечисления оптимальных и приближённых решений в задаче flow shop на двух машинах	34
1.3.5	Задача OPEN SHOP с маршрутизацией	39
1.4	Оценка сложностного статуса и построение эффективных алгоритмов с	43

	оценками точности для задач кластерного анализа	
1.4.1	Введение	43
1.4.2	Задачи выбора подпоследовательности	45
1.4.3	Анализ алгоритмической сложности	47
1.4.4	Заключение	53
1.5	Разработка новых методов использования бент-функций в криптографии и теории хэш-функций	54
1.6	Реализация метода квадратичного криптоанализа, использующего k-бент-функции для блочного шифра	56
1.7	Представление равномерно упакованных кодов через гармонические функции на n-кубе в терминах преобразования Фурье	59
1.8	Разбиение вершин плоского графа на подграфы малой степени и большого обхвата	63
1.9	Разработка эффективных алгоритмов построения коммуникационного дерева в беспроводных сенсорных сетях	76
1.10	Проведение численных экспериментов и апостериорного анализа для разработанного метода глобальной маршрутизации	78
2	Показатели	82
3	Заключение	83
4	Список использованных источников	86
	Приложение А. Список публикаций исполнителей	93
	Приложение Б. Список сделанных исполнителями докладов	96

Введение

Выполнение НИР направлено на проведение фундаментальных исследований в области теоретической кибернетики, с целью получения научных результатов мирового уровня, на подготовку и закрепление в сфере науки и образования научных и научно-педагогических кадров, а также формирование эффективных и жизнеспособных научных коллективов.

Запланированные исследования 4 этапа посвящены проведению фундаментальных исследований и играют важную роль в рамках всей НИР. В ходе работ предполагается построить оценки точности алгоритмов решения задачи коммивояжёра с несколькими коммивояжёрами, разработать эффективные алгоритмы приближённого решения задач теории расписаний, кластерного анализа и задач конкурентного размещения.

Предполагается разработать новые методы использования бент-функций в криптографии и теории хэш-функций, позволяющие реализовать методы квадратичного криптоанализа с использованием k -бент-функции для блочного шифра. Важная роль отведена проблеме представления равномерно упакованных кодов через гармонические функции на n -кубе в терминах преобразования Фурье, а также разбиения вершин плоского графа на подграфы малой степени и большого охвата.

Совместно с разработкой новых эффективных алгоритмов построения коммуникационного дерева в беспроводных сенсорных сетях эти исследования определяют фронт работ 4 этапа.

1. Проведение фундаментальных исследований

В рамках работ четвертого этапа НИР основной акцент сделан на конкретные проблемы хранения, обработки, передачи и защиты информации. Эти задачи определяют основное содержательное русло исследований, результаты которых могут быть применены на практике.

В отчете приведено описание работ по пунктам календарного плана в соответствии с техническим заданием.

1.1. Построение оценок точности алгоритмов решения задачи коммивояжера с несколькими коммивояжерами

1.1.1 Введение

Одним из естественных обобщений классической задачи коммивояжера (Traveling Salesman Problem – TSP) [26] является задача об m коммивояжерах (m -Peripatetic Salesman Problem – m -PSP) [25], состоящая в поиске в полном взвешенном неориентированном графе m реберно непересекающихся гамильтоновых циклов с минимальным или максимальным суммарным весом ребер. С тех пор как задача 2-PSP была впервые упомянута в работе [26] (Krarup-1975), появилось множество публикаций, посвященных ее исследованию. Было доказано [21] (DeKort-1991), что задача о существовании двух реберно непересекающихся гамильтоновых циклов в неориентированном графе NP-полна, что влечет NP-трудность задачи 2-PSP как на минимум так и на максимум даже в случае, если веса ребер принимают лишь значения 1 и 2. Рассматривались некоторые полиномиально разрешимые случаи задачи на минимум (2-PSP-min) [20] (DeBrey, Volgenant-1997). В статьях [20, 21] были предложены и проанализированы некоторые способы нахождения нижних и верхних оценок для применения в методе ветвей и границ. В работе [22] был представлен также полиэдральный подход к решению m -PSP.

Ввиду NP-трудности известных модификаций задач TSP и m -PSP большинство работ, посвященных их исследованию, связано с анализом полиномиально разрешимых случаев, а также с построением приближенных эвристических алгоритмов и полиномиальных алгоритмов (как детерминированных так и рандомизированных) с гарантированными оценками точности полученного решения.

За последние годы при участии авторов проекта был построен ряд полиномиальных алгоритмов с гарантированными оценками точности для задач одного и двух коммивояжеров

на максимум. Рассматривался как общий случай задачи, когда веса ребер принимают произвольные неотрицательные значения, так и метрический случай (выполняется неравенство треугольника), а также случай весов ребер из отрезка $[1, q]$ (при заданном $q > 1$). Задача двух и более коммивояжеров на максимум рассматривалась в случае графов в многомерном евклидовом пространстве.

1.1.2 Задача 2-PSP-max, алгоритм A_{7/9}

В настоящее время оценка точности $3/4$ в [1] (Агеев, Бабурин, Гимади–2006, далее [АБГ]) для задачи о двух коммивояжерах на максимум (2-PSP-max) на полном неориентированном графе с симметричными расстояниями улучшена до $7/9$ Глебовым и Замбалаевой за счет развития структурных идей, ранее использованных в работе [7] (Гимади, Глазков, Глебов–2007, далее [ГГГ]) при построении приближенного алгоритма с оценкой $6/5$ для задачи о двух коммивояжерах на минимум с весами ребер 1 и 2 (задача 2-PSP-min(1,2)). Интересно отметить, что оценка точности $7/9$ для задачи 2-PSP-max превосходит наилучшую на сегодняшний день оценку ($25/33$ -eps) для аналогичной задачи одного коммивояжера, полученную в [27] (Van Zuylen–2010).

Отправной точкой предлагаемого ниже алгоритма A_{7/9} является обращение к алгоритму Габова [23] (Gabow–1983), с помощью которого в графе G находится остовный 4-регулярный подграф G_4 с максимальным суммарным весом ребер. В общем случае алгоритм Габова позволяет отыскать в полном взвешенном n -вершинном графе $G(V, E)$ такой подграф H максимального реберного веса, что степени его вершин удовлетворяют следующим ограничениям снизу и сверху: $l_v \leq d(v) \leq h_v, v \in V$. При этом временная сложность алгоритма Габова оценивается сверху величиной

$$O(\min(|E|, n^2) \sum_{v \in V} h_v).$$

Ясно, что при $l_v = h_v = 4$ для всех $v \in V$ алгоритм Габова за время $O(n^3)$ находит в G остовный 4-регулярный подграф G_4 максимального реберного веса.

Дальнейший ход алгоритма A_{7/9} связан с применением процедуры из статьи [ГГГ], позволяющей в каждой компоненте связности G_4 найти пару реберно-непересекающихся туров специального вида с большим числом ребер. Далее найденные туры преобразуются в туры T_1^*, T_2^* в G со свойством

$$w(T_1^*) + w(T_2^*) \geq (7/9)OPT.$$

На завершающей стадии алгоритма туры T_1^*, T_2^* достраиваются до пары реберно-непересекающихся гамильтоновых циклов H_1 и H_2 в G , представляющих из себя искомое приближенное решение задачи 2-PSP-max.

Теорема. Алгоритм $A_{7/9}$ находит в полном n -вершинном графе G пару реберно-непересекающихся гамильтоновых циклов H_1, H_2 , для которых выполняется оценка $W(H_1) + W(H_2) \geq 7/9 \text{ OPT}$. Временная сложность алгоритма равна $O(n^3)$.

1.1.3 Алгоритм с оценкой $(3q+2)/(4q+1)$ для 2-PSP-max $[1, q]$

Важным подклассом задачи 2-PSP-max на полном неориентированном графе является случай, когда веса ребер графа принимают значения из заданного промежутка $[1, q]$. Ранее для этой подзадачи авторами проекта был построен ряд приближенных алгоритмов для решения соответствующей задачи на минимум. Так для задачи 2-PSP-min $[1, q]$ алгоритм из работы [ГТГ] дает оценку точности $(q+4)/5$, что позволяет построить на его основе алгоритм с оценкой $5/(q+4)$ для задачи 2-PSP-max $[1, q]$. Совместное применение результатов работ [АБГ] и [ГТГ] позволило получить для этой же задачи алгоритм A2 с улучшенной оценкой $(3q+2)/(4q+1)$.

Теорема. Алгоритм A2 за время $O(n^3)$ находит допустимое решение 2-PSP-max с весами ребер на отрезке $[1, q]$, вес которого составляет не менее $(3q+2)/(4q+1)$ от веса оптимального решения.

Следствие. Для соответствующей подзадачи 2-PSP-max с весами ребер 1 и 2 из теоремы следует оценка $8/9$.

С другой стороны, использование алгоритма из пункта 1 дает для указанной задачи оценку точности $(7q+2)/9q$, что лучше чем $(3q+2)/(4q+1)$ при $q > 2$, но уступает при $q < 2$ [10] (Глебов, Замбалаева, Иволина–2011). Наконец, модификация этого алгоритма, с учетом структурных свойств, доказанных в [ГТГ], позволяет обосновать наилучшую на сегодняшний день оценку $(7q+3)/(9q+1)$ для 2-PSP-max $[1, q]$. В частности, для соответствующей подзадачи 2-PSP-max с весами ребер 1 и 2 построенный алгоритм имеет оценку точности $17/19$, что на $1/153$ лучше оценки $8/9$.

1.1.4 Алгоритм $A_{7/5}$ для задачи 2-PSP(1,2)-min-2w

Отметим, что рассмотренные выше алгоритмы и оценки корректны лишь в том случае, когда весовые функции ребер у первого и второго маршрутов коммивояжера одинаковы. Построение соответствующих алгоритмов в случае, когда весовые функции

различны, представляет из себя существенно более трудную задачу. Это, в частности, объясняется невозможностью свободного обмена ребрами между строящимися гамильтоновыми циклами (частичными турами) по ходу работы алгоритма.

В случае двух различных весовых функций (задача 2-PSP(1,2)-min-2w) в статье [16] (Бабурин и др. в Discrete Applied Mathematics, 2009, далее [DAM]) был предложен алгоритм с оценкой $11/7$.

Усиление этого результата было получено в статье [11] (Глебов, Гордеева, Замбалаева – 2011), а именно, построение для задачи 2-PSP(1,2)-min-2w приближенного алгоритма с оценкой точности $7/5$ (без учета аддитивной константы) и оценкой временной сложности n^3). В основу алгоритма положена идея метода из статьи [17] (Бермана и Карпински, далее [БК]), заключающаяся в построении и последовательном "улучшении" двух реберно непересекающихся частичных туров наборов цепей и циклов, покрывающих все вершины графа) из ребер единичного веса, и последующем замыкании этих туров в непересекающиеся гамильтоновы циклы. Под "улучшением" туров понимается такое их локальное преобразование, при котором уменьшается либо общее число цепей и циклов, составляющих туры, либо число одновершинных цепей (синглов). Для того, чтобы гарантировать возможность улучшающего преобразования в случае, если нужное качество решения еще не достигнуто, применяется введенная в работе [БК] техника так называемых зарядов вершин, т.е. чисел, определяемых для каждой вершины графа на основе туров оптимального решения, и последующего перераспределения этих зарядов между вершинами графа с сохранением их суммы.

Дополнительные трудности при разработке и анализе алгоритма (по сравнению со случаем двух одинаковых весовых функций) связаны с тем, что в рассматриваемом случае невозможна свободная переброска ребер из одного тура в другой. Из-за этого оказались неприменимыми многие методы, ранее использовавшиеся в задаче о двух коммивояжерах с весами ребер 1 и 2 (см. [АБГ] и [ГГГ]). Доказательство результата было сопряжено со значительным объемом просмотра большого числа случаев, соответствующих различным типам вершин графа. Тип вершины определяется тем, является ли она в туре концевой или внутренней вершиной цепи, вершиной цикла или синглом (независимо для каждого тура).

1.1.5 Алгоритм $A_{4/3}$ для задачи 2-PSP(1,2)-min-2w

Алгоритм $A_{4/3}$ усиливает результаты из [11] и [16] по точности (но не по трудоемкости), а именно, для задачи 2-PSP(1,2)-min-2w строит приближенный алгоритм с оценкой точности $4/3$ (без учета аддитивной константы) и оценкой временной сложности

$O(n^5)$. В основу алгоритма положена та же идея метода БК из [17] с техникой зарядов вершин.

Понятие частичного тура, традиционно используемое в большинстве работ, посвященных построению приближенных алгоритмов для задач одного и двух коммивояжеров, существенно модифицировано за счет включения в туры наряду с цепями и циклами так называемых (s,q) -деревьев (обобщение синглов). Дополнительные трудности при разработке и анализе алгоритма (по сравнению со случаем двух одинаковых весовых функций) связаны с тем, что в рассматриваемом случае невозможна свободная переброска ребер из одного тура в другой. Из-за этого оказались неприменимыми многие методы, ранее использовавшиеся в задаче о двух коммивояжерах с весами ребер 1 и 2. Значительный объем работы также связан с рассмотрением большого числа случаев, соответствующих различным типам вершин графа. Тип вершины определяется тем, является ли она в туре концевой или внутренней вершиной цепи, вершиной цикла, синглом (S -вершиной) или так называемой Q -вершиной (s,q) -дерева (независимо для каждого тура). Есть все основания полагать, что разработанные методы, связанные с применением раскрасок вершин и ребер, перераспределением зарядов и использованием (s,q) -деревьев, окажется полезными при разработке других алгоритмов решения задач маршрутизации, в первую очередь, задач одного, двух и более коммивояжеров на минимум и на максимум.

1.1.6 Оценка точности для задачи 2-PSP(1,2) -max-2w с разными весовыми функциями ребер в маршрутах

Теорема. Если имеется алгоритм с оценкой γ для задачи 2-PSP-min{1,2} с двумя весовыми функциями, принимающими значениями 1 и 2, то на его основе можно построить алгоритм A_4 с оценкой $(11\gamma - 8)/(18\gamma - 15)$ для соответствующей задачи на максимум.

С одной стороны, результат основан на очевидной связи между 2-PSP-max {1,2} и 2-PSP-min {1,2}. С другой стороны, идеи работы [DAM] вместе с алгоритмом для TSP-min(1,2) с оценкой $7/6$ [25] (Papadimitriou, Yannakakis–1993) позволили получить оценку точности $(11\gamma - 8)/(18\gamma - 15)$ для 2-PSP-max(1,2). Исходя из известных алгоритмов решения 2-PSP-min(1,2) с двумя весовыми функциями и оценками $\gamma = 11/7$ [DAM], $7/5$ [11] и оценкой $4/3$ [13], можно получить следующие оценки точности для комбинированного алгоритма решения 2-PSP-max(1,2): $65/113 < 37/51 < 20/27$, соответственно.

1.1.7 Обоснование полиномиальности и условий асимптотической точности алгоритма решения задачи m-PSP на максимум в многомерном евклидовом пространстве R^k

Для задачи об m реберно-непересекающихся маршрутах коммивояжера на графах в многомерном евклидовом пространстве построен приближенный алгоритм с временной сложностью $O(n^3)$ и установлены условия асимптотической точности алгоритма [4, 18]. Результаты построения алгоритма и его анализа существенно опираются на прежние работы, связанные с обоснованием возможности асимптотически точного решения в пространстве R^k задачи коммивояжера на максимум [14] (Сердюков–1984) и [5] (Гимади–2000), а также задачи 2-PSP-max [6] (Гимади–2008). Отправной точкой алгоритма является построение максимального взвешенного паросочетания, которое представляется в виде совокупности $\lfloor n/2 \rfloor$ прямолинейных отрезков в многомерном евклидовом пространстве R^k . Ребра максимального взвешенного паросочетания используются как бы в качестве "строительных лесов", с помощью которых происходит построение искомых реберно-непересекающихся маршрутов коммивояжера H_1, H_2, \dots, H_m .

Теорема. Задача m-PSP на максимум в графе G с расстояниями в многомерном евклидовом пространстве R^k решается асимптотически точно при $m=o(n)$. Причем при $m < n^{1/2}$ алгоритм имеет временную сложность $O(n^3)$.

1.1.8 Задача маршрутизации с ограничениями на пропускные способности ребер графа

Рассматривается задача m коммивояжеров с ограничениями пропускной способности ребер (m-Capacitated Peripatetic Salesman Problem, m-CPSP). В полном неориентированном взвешанном графе G , каждое ребро e которого имеет заданную пропускную способность C_e со значениями из множества $\{1, \dots, m\}$, требуется найти m гамильтоновых циклов наименьшего суммарного веса с использованием каждого ребра e не более C_e раз. Задача m-CPSP NP-трудна, поскольку в частном случае единичных пропускных способностей каждого ребра имеем задачу m-Peripatetic Salesman Problem (m-PSP), которая NP-трудна при $m > 1$ даже на графе с весами ребер со значения 1 и 2.

Для задачи $2\text{-CPSP-min}\{1,2\}$ в случае, когда каждое ребро e графа имеет пропускную способность $C_e=2$ с вероятностью p и $C_e=1$ с вероятностью $1-p$ построен алгоритм с математическим ожиданием оценки точности $(19-5p)/12$. Алгоритм основан на

идеях работы [DAM] и использует алгоритм решения задачи TSP-min{1,2} с оценкой точности $7/6$ [25] (Papadimitriou, Yannakakis–1993).

1.1.9 Задача k-VRP – маршрутизации многих транспортных средств (ТС) с ограниченным числом клиентов в каждом маршруте

В полном графе G выделена вершина-депо, в которой изначально содержатся ТС. Остальные вершины – клиенты, расстояния между которыми и с вершиной-депо представлены ребрами графа. Множество клиентов представлено вершинами полного графа. Расстояния между клиентами и вершиной-депо представлены ребрами графа. Каждый клиент обслуживается только одним ТС, которое стартует из депо, посещает (обслуживает) не более k клиентов и возвращается в депо. При этом минимизируется общая длина маршрутов.

Для задачи с одним депо в работе [8] (Гимади, Шахштейн – 2012) представлены алгоритмы с временной сложностью $O(n^2)$ на случайных входах. Для усеченно-нормального и экспоненциального распределений с параметрами σ_n и λ_n , равномерного распределения в интервале $[a_n, b_n]$ и мажорирующего распределения входных данных представлены условия асимптотической точности, верные как в случае неориентированных, так и ориентированных графов. Показано, что приближенное решение задачи k-VRP на рассматриваемых классах входов может быть получено с оценкой относительной погрешности $O(\beta_n/a_n)$ и вероятности несрабатывания $O(\exp(n/k) / n^2)$, где параметр β_n в случае усеченно-нормального и экспоненциального распределений равен соответствующим параметрам σ_n и λ_n , а в случае равномерного распределения равен b_n . Так что имеем условия асимптотической точности:

$$\beta_n/a_n = o(n/\ln n); \quad k \geq 3n/\ln n.$$

В задаче k-VRP с несколькими депо транспортные средства распределены по s депо и каждое депо вмещает не более $m = \lfloor n/ks \rfloor$ ТС. При этом рассмотрены варианты задачи как с требованием возврата транспортного средства в исходное депо, так и без такого требования.

1.2 Разработка и численный анализ алгоритмов локального поиска с обобщенной окрестностью для построения приближенных решений задач конкурентного размещения

В предыдущем отчете рассмотрены модели (L, F) и (L, F') размещения предприятий и введены понятия оптимальных кооперативных и некооперативных решений полученных задач. Кроме того, показано, что задачи поиска оптимальных кооперативных и некооперативных решений задач (L, F) и (L, F') сводятся к задаче максимизации псевдобулевых функций. Особенность этих функций состоит в том, что они заданы неявным образом и для вычисления значения таких функций необходимо решить две задачи целочисленного линейного программирования. Также ранее мы привели алгоритмы поиска локально-оптимальных решений задачи максимизации псевдобулевых функций указанного вида. Алгоритмы включают два этапа. На первом вычисляется верхняя граница для значений рассматриваемых псевдобулевых функций и одновременно строится некоторое начальное решение. На втором этапе это решение улучшается до локально-оптимального решения. Соответствующий алгоритм представляет собой стандартную процедуру локального поиска с окрестностью специального вида.

В настоящем разделе для задачи оптимизации псевдобулевых функций рассматривается алгоритм локального поиска с обобщенной окрестностью. Такая окрестность строится для локально-оптимальных решений и включает в себя другие локально-оптимальные решения “окружающие” данное решение. Приводятся результаты вычислительных экспериментов с использованием псевдобулевых функций, оптимизация которых эквивалентна задачам конкурентного размещения предприятий. Целью экспериментов является сравнительная оценка локально-оптимальных решений, получаемых стандартным алгоритмом локального поиска и алгоритмом локального поиска с обобщенной окрестностью.

1.2.1 Обобщенная окрестность

Рассмотрим задачу минимизации псевдобулевой функции $f(x)$, определенной на множестве B^m (0, 1)-векторов $x = (x_i)$, $i \in I = \{1, \dots, m\}$. Пусть для всякого $x \in B^m$ задано множество $N(x) \subset B^m$, называемое окрестностью решения x . В случае множества B^m в качестве окрестности $N(x)$ точки $x \in B^m$ обычно используют следующие множества:

$$N_1(x) = \{y \in B^m \mid d(x, y) = 1\}$$

$$N_2(x) = \{y \in B^m \mid d(x, y) = 2, d(0, x) = d(0, y)\},$$

где $d(x, y)$ — расстояние Хэмминга, равное числу несовпадающих компонент $(0, 1)$ -векторов x и y .

При заданной окрестности $N(x)$ решение $x_0 \in B^m$ называется *локально-оптимальным*, если $f(x_0) \leq f(x)$ для всякого $x \in N(x_0)$.

Стандартный алгоритм локального поиска по заданной окрестности $N(x)$, $x \in B^m$, включает конечное число однотипных шагов, на каждом из которых рассматривается некоторое текущее решение x_0 . На первом шаге в качестве x_0 может быть взят любой $(0, 1)$ -вектор. Шаг состоит в поиске элемента $x' \in N(x_0)$, *улучшающего* текущее решение x_0 , т.е. такого элемента $x' \in N(x_0)$, что $f(x') < f(x_0)$. Если решения x' найти не удастся, то алгоритм останавливается, и текущее решение x_0 есть результат его работы. В противном случае текущее решение x_0 заменяется на решение x' , и начинается следующий шаг.

Способ выбора решения x' в алгоритме локального поиска нуждается в уточнении. Если задан некоторый порядок просмотра элементов множества $N(x_0)$, то решение x' может быть выбрано в результате *частичного* просмотра окрестности $N(x_0)$. Таким решением будет первый в заданном порядке элемент $x \in N(x_0)$, для которого $f(x) < f(x_0)$. При *полном* просмотре окрестности $N(x_0)$ в качестве улучшающего решения выбирается такой элемент x' , что $f(x') < f(x_0)$ и $f(x') \leq f(x)$ для каждого $x \in N(x_0)$.

Пусть для всякого $x \in B^m$ задана окрестность $N(x) \in B^m$, которую будем называть *базовой*, и пусть x_0 — локально-оптимальное решение относительно этой окрестности. Определим обобщенную окрестность $\tilde{N}(x_0)$ локально-оптимального решения x_0 . Это множество содержит не более чем m других локально-оптимальных решений, каждое из которых совпадает с одним из векторов $\tilde{x}_0^k, k \in I$. Для всякого $k \in I$ локально-оптимальное решение \tilde{x}_0^k определяется следующим образом.

При фиксированном $k \in I$ рассмотрим, наряду с базовой окрестностью $N(x)$, окрестность $N^k(x) = \{y \in N(x) \mid y_k = x_k\}$ и $(0, 1)$ -вектор $y^k = (y_i^k), i \in I$, отличающийся от решения $x_0 = (x_{0i}), i \in I$, только тем, что $y_k^k = 1 - x_{0k}$. Локально-оптимальное решение \tilde{x}_0^k строится в два этапа. Сначала, используя стандартную процедуру локального поиска по окрестности $N^k(x)$ и вектору y^k в качестве начальной точки, определяется решение y_0^k . Это решение является локально-оптимальным относительно вспомогательной окрестности $N^k(x)$.

Затем по решению y_0^k с помощью стандартной процедуры локального поиска по окрестности $N(x)$ строится локально-оптимальное решение \tilde{x}_0^k .

Обобщенной окрестностью локально-оптимального решения x_0 назовем множество

$$\tilde{N}(x_0) = \{x \in B^m \mid x = \tilde{x}_0^k \text{ для некоторого } k \in I\}.$$

Вектор x_0 будем называть центром обобщенной окрестности $\tilde{N}(x_0)$.

Отметим, что для различных $k \in I$ построенные локально-оптимальные решения \tilde{x}_0^k могут совпадать между собой и могут совпадать с центром окрестности x_0 . Поэтому число элементов в окрестности $\tilde{N}(x_0)$ может быть меньше чем m . В связи с этим, возникает вопрос о строении множества $\tilde{N}(x_0)$, которое можно охарактеризовать числом элементов в множестве и расстояниями от этих элементов до центра окрестности.

Алгоритм локального поиска по обобщенной окрестности, так же как и стандартный алгоритм локального поиска представляет собой процедуру последовательного улучшения текущего решения. Но в случае обобщенной окрестности текущим решением является локально-оптимальное решения, для улучшения которого используется также локально-оптимальное решение.

Пусть для всякого $x \in B^m$ задана базовая окрестность $N(x)$. Алгоритм локального поиска по обобщенной окрестности при заданной базовой окрестности $N(x)$, $x \in B^m$, состоит из предварительного шага и конечного числа однотипных основных шагов, на каждом из которых рассматривается некоторое текущее локально-оптимальное решение x_0 .

На предварительном шаге по заданному $(0,1)$ -вектору с использованием стандартной процедуры локального поиска строится начальное локально-оптимальное решение x_0 . После этого начинается основной шаг.

На основном шаге имеется текущее локально-оптимальное решение x_0 . Шаг состоит в построении обобщенной окрестности $\tilde{N}(x_0)$ и поиске элемента $x' \in \tilde{N}(x_0)$, такого что $f(x') < f(x_0)$. Если решение x' найти не удастся, то алгоритм заканчивает работу, результатом которой является текущее локально-оптимальное решение x_0 . В противном случае текущее локально-оптимальное решение x_0 заменяется на локально-оптимальное решение x' и начинается следующий шаг.

Так же как и в случае стандартного алгоритма локального поиска, будем использовать два способа выбора решения x' , из окрестности $\tilde{N}(x_0)$, улучшающего текущее решение x_0 .

При полном просмотре окрестности $\tilde{N}(x_0)$ решение x' должно удовлетворять условиям $f(x') < f(x_0)$ и $f(x') \leq f(x)$ для каждого $x \in \tilde{N}(x_0)$, а при частичном просмотре элементы окрестности $\tilde{N}(x_0)$ исследуются в некотором заданном порядке и в качестве решения x' выбирается первый элемент $x \in \tilde{N}(x_0)$, для которого $f(x) < f(x_0)$.

1.2.2. Алгоритм локального поиска по обобщенной окрестности для задачи конкурентного размещения предприятий

Задача конкурентного размещения предприятий формулируется [1,2] как задача двухуровневого целочисленного программирования вида

$$\max_{(x_i), (x_{ij})} \left\{ - \sum_{i \in I} f_i x_i + \sum_{j \in J} \left(\sum_{i \in I} p_{ij} x_{ij} \right) \left(1 - \sum_{i \in I} \tilde{z}_{ij} \right) \right\} \quad (1.2.2.1)$$

$$x_i + \sum_{k | i \succ_j k} x_{kj} \leq 1, \quad i \in I, j \in J; \quad (1.2.2.2)$$

$$x_i \geq x_{ij}, \quad i \in I, j \in J; \quad (1.2.2.3)$$

$$x_i, x_{ij} \in \{0,1\}, \quad i \in I, j \in J; \quad (1.2.2.4)$$

$$((\tilde{z}_i), (\tilde{z}_{ij})) \text{ — оптимальное решение задачи} \quad (1.2.2.5)$$

$$\max_{(z_i), (z_{ij})} \left\{ - \sum_{i \in I} g_i z_i + \sum_{i \in I} \sum_{j \in J} p_{ij} z_{ij} \right\} \quad (1.2.2.6)$$

$$x_i + z_i + \sum_{k | i \succ_j k} z_{kj} \leq 1, \quad i \in I, j \in J; \quad (1.2.2.7)$$

$$z_i \geq z_{ij}, \quad i \in I, j \in J; \quad (1.2.2.8)$$

$$z_i, z_{ij} \in \{0,1\} \quad i \in I, j \in J; \quad (1.2.2.9)$$

где $I = \{1, \dots, m\}$, $J = \{1, \dots, n\}$, (f_i) , $i \in I$, (g_i) , $i \in I$, — векторы с неотрицательными элементами, (p_{ij}) , $i \in I, j \in J$, — матрица с неотрицательными элементами, а \succ_j , $j \in J$, — отношение порядка на множестве I .

Данная задача, как и всякая задача двухуровневого программирования, включает задачу верхнего уровня (1.2.2.1)–(1.2.2.4), которую будем обозначать L и задачу нижнего уровня (1.2.2.6)–(1.2.2.9), которую будем обозначать через F . Для задачи (1.2.2.1)–(1.2.2.9) используется обозначение (L, F) .

Обозначим через $X = ((x_i), (x_{ij}))$ допустимое решение задачи L , а через $\tilde{Z} = ((\tilde{z}_i), (\tilde{z}_{ij}))$ – оптимальное решение задачи F при заданном решении X . Пару (X, \tilde{Z}) будем называть *допустимым решением* задачи (L, F) . Обозначим через $L(X, \tilde{Z})$ значение целевой функции задачи (L, F) на допустимом решении (X, \tilde{Z}) .

В качестве оптимального значения целевой функции задачи (L, F) в [2] принято её значение на так называемом оптимальном некооперативном решении. Допустимое решение (X, \tilde{Z}) называется допустимым некооперативным решением задачи (L, F) , если $L(X, \tilde{Z}) \leq L(X, \bar{Z})$ для любого допустимого решения (X, \bar{Z}) , а допустимое некооперативное решение (X^*, \bar{Z}^*) называется оптимальным некооперативным решением задачи (L, F) , если $L(X^*, \bar{Z}^*) \geq L(X, \bar{Z})$ для любого допустимого некооперативного решения (X, \bar{Z}) .

В предшествующем отчете отмечено, что попроизвольному $(0,1)$ -вектору $x = (x_i)$ строится допустимое некооперативное решение (X, \tilde{Z}) , где $X = ((x_i), (x_{ij}))$, и однозначно определяется значение целевой функции задачи (L, F) .

Таким образом, получаем, что задача (L, F) сводится к задаче максимизации некоторой псевдобулевой функции $f(x)$, $x \in B^m$. Только данная функция $f(x)$, в отличие от рассмотренной выше, не задана в явном виде, а для вычисления её значения необходимо, в частности, найти решения двух задач целочисленного линейного программирования.

Ранее был приведен алгоритм вычисления верхней границы для максимального значения функции $f(x)$ и предложены алгоритмы локального поиска для задачи отыскания максимального значения данной функции. С учётом трудоёмкости вычисления значения этой функции в предложенных алгоритмах используется окрестность $N_0(x) \subset N_1(x) \cup N_2(x)$ специального вида, содержащая не более m элементов. Кроме того, в этих алгоритмах в качестве начального решения используется $(0,1)$ -вектор, полученный в результате вычисления верхней границы для максимального значения функции $f(x)$.

Приведём результаты вычислительного эксперимента с алгоритмом локального поиска по обобщённой окрестности для задачи максимизации рассмотренной псевдобулевой функции $f(x)$. В этом алгоритме в качестве базовой окрестности используется окрестность $N_0(x)$, а выбор улучшающего элемента при построении локально-оптимальных решений производится по правилу неполного просмотра окрестностей. Выбор улучшающего элемента в обобщённой окрестности $\tilde{N}_0(x)$ так же производится в результате неполного её просмотра

за исключением первого шага алгоритма, где улучшающий элемент определяется в результате полного просмотра окрестности. Кроме того, на предварительном шаге алгоритма в качестве начальной точки используется $(0,1)$ -вектор, получаемый одновременно с вычислением верхней границы для оптимального значения функции $f(x)$.

Цель данного вычислительного эксперимента состоит в том, чтобы выяснить строение обобщённой окрестности $\tilde{N}_0(x)$, рассматриваемой на первом шаге алгоритма, и оценить насколько локально-оптимальное решение, найденное на предварительном шаге алгоритма, отличается от локально-оптимального решения, полученного в результате локального поиска по обобщённой окрестности.

Вычисления проводились для примеров задачи конкурентного размещения предприятий, взятых из библиотеки тестовых задач [28]. Использовались примеры из подклассов A20, A30, A40 и A50 класса A, для которых число переменных псевдобулевой функции $f(x)$ равняется соответственно 20, 30, 40 и 50. Для примеров из подклассов A20 и A30 известны оптимальные значения целевых функций.

В таблицах 1 и 2 для 20 примеров из каждого подкласса приводятся следующие величины:

f_1 — значение целевой функции на локально-оптимальном решении x_0 , полученном на предварительном шаге алгоритма.

f_2 — значение целевой функции на наилучшем локально-оптимальном решении из обобщённой окрестности, рассматриваемой на первом шаге алгоритма.

f_3 — значение целевой функции на локально-оптимальном решении \tilde{x}_0 , полученном в результате работы алгоритма локального поиска по обобщённой окрестности.

K — число элементов в обобщённой окрестности на первом шаге алгоритма.

d — среднее расстояние от элементов обобщённой окрестности, построенной на первом шаге алгоритма, до её центра.

Кроме того, в таблице 1 для каждого примера приводится оптимальное значение f^* целевой функции и величины относительно точности f^*/f_1 и f^*/f_3 локально-оптимальных решений x_0 и \tilde{x}_0 .

Из таблиц 1 и 2 видно, что исследуемая обобщенная окрестность локально-оптимального решения x_0 не является вырожденной. Элементы обобщенной окрестности в большинстве примеров в среднем отличаются от решения x_0 двумя или тремя компонентами.

Из таблиц 1 и 2 видно так же, что для большинства примеров локально-оптимальное решение \tilde{x}_0 лучше, чем локально-оптимальное решение x_0 , а из таблицы 1 следует, что использование обобщенной окрестности для подавляющего большинства примеров позволяет получать оптимальное решение.

Таблица 1 – Результаты вычислений.

Задача	f_1	f_2	f_3	K	d	f^*	f^*/f_1	f^*/f_3
a20-01	62	62	62	3	1,67	62	1,00	1,00
a20-02	53	53	53	2	1,00	53	1,00	1,00
a20-03	39	41	61	5	2,40	61	1,56	1,00
a20-04	23	35	35	8	2,50	35	1,52	1,00
a20-05	58	58	58	2	0,50	58	1,00	1,00
a20-06	41	41	41	2	1,00	51	1,24	1,24
a20-07	5	10	10	6	2,00	10	2,00	1,00
a20-08	27	27	27	4	2,25	38	1,41	1,41
a20-09	8	16	16	6	2,00	16	2,00	1,00
a20-10	9	24	24	9	3,56	29	3,22	1,21
a20-11	30	42	42	7	2,43	42	1,40	1,00
a20-12	15	22	22	10	2,00	22	1,47	1,00
a20-13	25	25	25	3	1,67	25	1,00	1,00
a20-14	51	51	51	2	1,50	51	1,00	1,00
a20-15	40	40	40	4	2,25	40	1,00	1,00
a20-16	12	27	27	6	2,67	27	2,25	1,00
a20-17	30	42	42	7	2,86	42	1,40	1,00
a20-18	50	50	50	2	1,00	50	1,00	1,00
a20-19	40	46	46	8	3,00	46	1,15	1,00
a20-20	32	32	32	6	2,83	32	1,00	1,00
a30-01	62	62	62	10	2,70	67	1,08	1,08
a30-02	37	44	44	9	1,89	44	1,19	1,00
a30-03	53	74	74	9	1,89	74	1,40	1,00
a30-04	58	63	67	8	1,25	67	1,16	1,00
a30-05	100	115	115	6	2,33	115	1,15	1,00
a30-06	41	41	41	6	2,33	41	1,00	1,00
a30-07	59	78	84	12	3,17	84	1,42	1,00
a30-08	44	51	56	8	1,63	56	1,27	1,00
a30-09	83	83	83	4	1,75	83	1,00	1,00
a30-10	54	67	67	16	2,56	67	1,24	1,00
a30-11	72	82	82	7	1,86	82	1,14	1,00
a30-12	61	61	61	7	2,43	71	1,16	1,16
a30-13	31	46	49	8	1,38	49	1,58	1,00

a30-14	44	68	73	6	1,50	73	1,66	1,00
a30-15	21	43	43	10	2,70	45	2,14	1,05
a30-16	76	76	76	4	2,00	76	1,00	1,00
a30-17	95	114	114	6	3,33	114	1,20	1,00
a30-18	55	87	87	10	3,90	87	1,58	1,00
a30-19	39	57	57	8	2,75	57	1,46	1,00
330-20	52	62	70	7	1,00	70	1,35	1,00

Таблица 2 – Результаты вычислений.

Задача	f_1	f_2	f_3	K	d
a40-01	62	62	62	9	2,33
a40-02	135	135	135	10	2,50
a40-03	81	81	81	2	2,50
a40-04	96	96	96	10	2,20
a40-05	65	94	97	13	3,69
a40-06	81	81	81	8	2,38
a40-07	81	90	96	11	2,27
a40-08	121	125	125	7	2,29
a40-09	81	83	83	7	2,29
a40-10	105	107	107	3	2,33
a40-11	101	101	101	6	1,50
a40-12	47	53	57	7	1,71
a40-13	163	163	163	5	1,80
a40-14	61	67	67	10	1,90
a40-15	93	100	100	9	3,00
a40-16	69	77	77	6	1,67
a40-17	107	132	136	6	2,67
a40-18	131	131	131	10	2,30
a40-19	95	112	112	11	2,45
a40-20	52	54	54	7	3,00
a50-01	128	158	158	8	2,75
a50-02	100	100	100	9	2,11
a50-03	85	106	106	12	2,50
a50-04	85	86	92	9	2,89
a50-05	103	103	103	3	1,33
a50-06	94	122	122	10	4,00
a50-07	68	91	91	12	3,83
a50-08	81	92	92	8	1,75
a50-09	65	93	93	12	2,50
a50-10	80	119	119	15	2,80

a50-11	115	119	120	9	2,22
a50-12	117	127	127	6	1,83
a50-13	54	84	94	21	3,76
a50-14	63	77	77	5	2,00
a50-15	112	122	122	9	2,78
a50-16	85	85	85	14	2,71
a50-17	156	156	156	11	2,00
a50-18	65	97	107	13	2,46
a50-19	99	107	111	13	2,69
a50-20	40	89	89	16	4,25

1.3. Разработка эффективных алгоритмов приближённого решения задач теории расписаний

Результаты, достигнутые нами в исследованиях на предыдущих этапах работы по настоящему проекту (а именно, исследование свойств оптимальных решений рассматриваемых задач и анализ сложности этих задач), позволяют перейти к следующему шагу в решении данных задач, а именно – к разработке эффективных алгоритмов решения данных задач. Действительно, выявление внутренних свойств оптимальных решений зачастую облегчает «обнаружение» этих решений и выделение их из (как правило, огромной) массы прочих (допустимых и недопустимых) решений. Задача поиска существенно облегчается, когда то, что мы ищем, имеет яркий отличительный признак. С другой стороны, проведённый предварительный анализ сложности позволяет иногда утверждать, что хотя и найденный нами способ решения не является формально эффективным (узкая трактовка понятия «эффективного алгоритма» включает в себя лишь алгоритмы полиномиальной сложности), но поскольку ничего лучшего просто не существует (если опираться на справедливость гипотезы о несовпадении классов P и NP), то найденный нами способ решения является просто наилучшим из возможных (а стало быть, вполне «эффективным»).

Таким образом, понятие «эффективного алгоритма» решения будет трактоваться нами расширительно, т.е. либо по отношению к другим (известным на сегодняшний день алгоритмам), либо по отношению к наилучшим алгоритмам, что могут существовать для данной задачи теоретически. При этом в ряде случаев мы будем называть «эффективными» алгоритмы псевдо-полиномиальной трудоёмкости (при условии, что нами доказано несуществование более быстрых – полиномиальных – алгоритмов решения этих задач), в других же случаях само существование алгоритма будет рассматриваться как позитивный результат (если до этих исследований вопрос о существовании какого-либо решения задачи за конечное время оставался открытым).

Другое «расширение» рассматриваемой темы лежит в термине «алгоритм приближённого решения». Мы будем считать (вполне справедливо), что «приближённый» алгоритм вовсе не «обязан» ошибаться. Таким образом, представленные нами алгоритмы точного решения также входят в понятие «алгоритмов приближённого решения».

Иными словами, на данном этапе исследований нас будут интересовать различные алгоритмические аспекты решения задач дискретной оптимизации.

1.3.1 Задача об инвестировании проектов

У некоей организации имеется Проект, состоящий из конечного множества мини-проектов $N = \{1, \dots, n\}$. Разработка мини-проекта $i \in N$ требует начального инвестирования в количестве α_i , в то время как завершение мини-проекта характеризуется получением единовременной прибыли в количестве β_i . Таким образом, общий баланс прибыли, получаемой от выполнения мини-проекта $i \in N$, составляет $\delta_i = \beta_i - \alpha_i$.

Ясно, что для реализации заданного Проекта организации необходимо иметь какой-то первоначальный капитал (будем называть его «ресурс»), – хотя бы для того, чтобы начать реализацию самого первого мини-проекта. Вопрос состоит в том, какое минимальное количество начального капитала (Q_0) достаточно для реализации всего Проекта, и какой должна быть при этом последовательность реализации мини-проектов?

В своей исходной постановке «Задача об инвестировании проектов» рассматривала в качестве входных данных лишь Q_0, α_i и β_i и отыскивала допустимую последовательность реализации мини-проектов при заданном начальном уровне ресурса. Предполагалось также, что любой мини-проект может выполняться в любой момент времени.

В отличие от исходной постановки, мы рассматриваем более общую ситуацию, когда время готовности к исполнению (r_i) разных мини-проектов $i \in N$ различно. При этом существенными становятся и прочие временные параметры задачи, такие как длительность (p_i) выполнения i -го мини-проекта. В результате возникает совсем другая задача – об отыскании допустимой последовательности реализации мини-проектов заданного Проекта за наименьшее время.

Говоря формально, мы решаем задачу о выполнении работ из заданного конечного множества $N = \{1, \dots, n\}$ на одной машине с дополнительным ресурсным ограничением. Исходно в ресурсном пуле имеется Q_0 единиц ресурса. Каждая работа $i \in N$ появляется в системе в заранее заданный момент времени r_i (лишь только после этого момента работа может быть «поставлена в расписание на выполнение») и имеет заданную длительность p_i . Для того, чтобы можно было начать выполнение этой работы, требуется изъять из ресурсного пула (в момент начала выполнения работы) заданное количество α_i единиц ресурса. В то же время, по окончании выполнения работы она «возвращает» в ресурсный пул β_i единиц ресурса. Без ограничения общности мы можем предполагать, что эти величины неотрицательны. Машина выполняет в каждый момент времени не более одной работы. Прерывания работ запрещены. Таким образом, работа i может начать выполнение в момент t , только если (1) она уже появилась в системе ($t \geq r_i$), (2) уровень ресурса в ресурсном пуле достаточен (не меньше величины α_i), и (3) машина не занята другой работой. Все

упомянутые параметры принимают неотрицательные целые значения (за исключением *суммарного вклада* работы в ресурсный пул, определяемого как $\delta_i = \beta_i - \alpha_i$ и могущего принимать отрицательные значения). При заданном начальном уровне ресурса в ресурсном пуле, расписание выполнения работ считается допустимым, если оно удовлетворяет ограничениям на моменты поступления работ и выполнение никакой из работ не «заблокировано» вследствие отсутствия необходимого количества ресурса. Целью решения задачи является построение допустимого расписания минимальной длины (C_{max}).

Используя стандартные трёхпольные обозначения, введённые Грэмом и др. (1979), мы можем обозначить нашу задачу как $1|rp, r_i|C_{max}$. Обозначение « rp » во втором поле означает, что мы рассматриваем ресурсные ограничения, подобные тем, что рассматриваются в «задаче передислокации» (relocation problem).

Для точного решения данной задачи нами разработано несколько алгоритмов. Первый алгоритм, основанный на идее многопараметрического динамического программирования, решает сформулированную задачу в её общем виде. Опишем далее более подробно, что понимается под «многопараметрическим динамическим программированием».

Обычное (классическое) динамическое программирование предполагает наличие в описании задачи одного или двух выделенных параметров, принимающих дискретное множество значений (как правило, целочисленных). В алгоритме ДП решение задачи для заданного примера сводится к последовательному решению задачи для подпримеров меньшего размера, и при этом для всевозможных значений выделенных параметров вычисляются оптимальные значения целевой функции. В многопараметрическом ДП само число выделенных параметров не ограничено какой-либо константой и может принимать сколь угодно большие значения. В нашем случае была разработана схема ДП с $3m$ выделенными параметрами, где m обозначает число различных значений, которые могут принимать величины r_i в рассматриваемых примерах нашей задачи. При этом трудоёмкость решения составила $O(PP(I)^{3m})$, где $PP(I)$ – некоторый псевдополином от входных данных рассматриваемой задачи. Таким образом, трудоёмкость экспоненциально зависит от параметра m . При этом если мы решаем подзадачу, в которой параметр m ограничен константой, то трудоёмкость решения становится псевдополиномиальной.

Можно ли, однако, усилить этот результат и предложить псевдополиномиальный алгоритм и для общей задачи (в которой параметр m не ограничен никакой константой)? Как показывает анализ сложности задачи, проведённый нами на предыдущем этапе проекта, ответ на этот вопрос ОТРИЦАТЕЛЕН (в предположении о несовпадении классов P и NP), поскольку для исходной задачи в её общем виде доказана NP -трудность в сильном смысле.

Теперь зададим вопрос: а в случае, когда параметр m ограничен константой, является ли наш алгоритм наилучшим? Нельзя ли предложить более быстрые алгоритмы (например, полиномиальной трудоёмкости)? Ответ и на этот вопрос ОТРИЦАТЕЛЕН (в предположении $P \neq NP$): кардинального улучшения трудоёмкости алгоритма не предвидится, поскольку для исходной задачи доказана её NP-трудность, даже в случае $m \leq 2$.

Таким образом, хотя полученные оценки трудоёмкости разработанного нами алгоритма не внушают особого оптимизма, мы можем завершить эти исследования «с чувством выполненного долга», поскольку успокаивает мысль: мы сделали всё, что могли.

Другой алгоритм разработан нами для подзадачи исходной задачи. В эту подзадачу включаются лишь примеры, в которых все работы $i \in N$ имеют неотрицательные вклады (δ_i) в ресурсный пул. Для решения такой задачи оказывается достаточным простейший алгоритм, в котором в каждый момент времени, когда освобождается машина (после выполнения очередной работы), в качестве следующей выполняемой работы может быть взята любая допустимая работа (т.е., во-первых, уже пришедшая в систему, и во-вторых, для начала выполнения которой имеется достаточное количество ресурса в ресурсном пуле). Поскольку для реализации этого алгоритма достаточно упорядочить все работы по неубыванию моментов их появления, то трудоёмкость алгоритма не превосходит $O(n \log n)$.

Любопытно, что если рассматривать «симметричную» подзадачу, включающую лишь примеры, в которых все работы $i \in N$ имеют неположительные вклады δ_i , то результат будет совсем другим: задача становится NP-трудной в сильном смысле.

1.3.2 Задачи JOB SHOP, OPEN SHOP и смешанная

Вначале дадим формулировки исследуемых задач.

Задача job shop возникает в многостадийном производственном процессе, в котором заданное множество работ $\{J_1, \dots, J_n\}$ требуется выполнить на *машинах* из заданного множества $\{M_1, \dots, M_m\}$; при этом каждая работа бывает вынуждена пройти несколько последовательных стадий своего выполнения (называемых операциями), выполняемых на разных машинах. Более точно, работа J_j представляет собой цепь из η_j операций $O_{1j}, \dots, O_{\eta_j j}$, и при этом каждая операция O_{ij} может стартовать лишь после завершения предыдущей операции $O_{(i-1)j}$. Каждой операции O_{ij} заранее приписана машина, на которой она должна выполняться непрерывно в течение заданного времени p_{ij} . (При этом мы допускаем существование операций нулевой длительности, что является существенным для задач данного типа.) В любом допустимом расписании для этой задачи в каждый момент

времени (за исключением, быть может, конечного множества моментов) каждая работа может выполняться не более чем на одной машине, а каждая машина может выполнять не более одной работы.

Общая задача open shop отличается от задачи job shop тем, что порядок выполнения операций каждой работы не фиксирован заранее, а является предметом решения задачи. Кроме того, в классической задаче open shop мы дополнительно предполагаем, что каждая работа имеет не более одной операции на каждой машине.

Наконец, в смешанной задаче (mixed shop) каждый пример может содержать как работы типа job shop (с фиксированным порядком операций), так и работы типа open shop (со свободным порядком операций).

Целью является нахождение расписания минимальной длины. Иначе говоря, ставится задача минимизации момента (C_{max}) завершения наиболее поздней операции. В задачах распознавания нашей целью является проверка существования допустимого расписания, длина которого не превосходит заданного параметра C (В этом случае значение параметра является частью входа.)

Следуя стандартной (3-польной) схеме обозначения цеховых задач, значение J в первом поле будет использоваться нами для обозначения задачи job shop, O – для классической задачи open shop, значение \bar{O} – для обозначения *общей задачи open shop*, и $J + \bar{O}$ – для их смешанной задачи.

Разработанные нами полиномиальные алгоритмы для этих задач, конечно же, не могут решать данные задачи в их самом общем виде (поскольку последние NP-трудны в сильном смысле). Эффективно решены следующие частные случаи этих задач.

1. Задача job shop с не более чем двумя операциями каждой работы и единичными длительностями операций (т.е. задача $J|p_{ij} = 1, \eta \leq 2|C_{max}$) полиномиально разрешима. Алгоритм основывается на алгоритме рёберной раскраски двудольного графа в минимальное число цветов. (Поскольку полиномиальная разрешимость последней задачи известна уже давно, то для своих целей мы можем брать любой из известных алгоритмов.)
2. Более общая смешанная задача $(J + \bar{O})|p_{ij} = 1, \eta \leq 2|C_{max}$ также полиномиально разрешима. Алгоритм её решения является комбинацией линейного по сложности алгоритма ориентации рёбер и упомянутого выше алгоритма решения задачи job shop.

Хотя полученные результаты кажутся довольно частными, проведённый нами (на предыдущем этапе проекта) анализ сложности данных задач показывает, что результат является наилучшим возможным. Действительно, полиномиальная разрешимость задачи не распространяется на более общие задачи, лишь незначительно расширяющие

рассматриваемую задачу по какому-либо из параметров. Так, например, нами показано, что, если расширить область входных примеров за счёт разрешения примеров, в которых длительности работ принимают лишь два допустимых значения: 1 либо 2, но при этом значительно сузить область за счёт перехода к чистой job shop либо чистой (классической) open shop, да ещё ввести ограничение на длину расписания: $C_{max} \leq 4$, то задача распознавания существования допустимого решения становится NP-полной в сильном смысле. Аналогичный эффект (потеря полиномиальной разрешимости) имеет место в том случае, если некоторым работам разрешить иметь по 3 операции (вместо рассматриваемого нами ограничения: не более 2 операций на работу). – При этом «не помогает» ни сужение области входных примеров за счёт введения ограничения $C_{max} \leq 3$ на длину расписания, с одновременным переходом от общей задачи open shop к классической, ни ограничение $C_{max} \leq 4$ на длину расписания, с одновременным переходом от смешанной задачи к чистой job shop.

1.3.3 Задача JOB SHOP с разрешением прерываний, на минимум произвольной регулярной функции от моментов завершения операций

Постановка рассматриваемой в данном пункте задачи job shop отличается от приведённой в предыдущем пункте формулировки задачи (под тем же названием «job shop») двумя моментами. Во-первых, множество допустимых решений задачи (расписаний) существенно расширяется за счёт разрешения прерываний операций (в произвольные моменты времени и на сколь угодно много частей). Во-вторых, вместо хорошо изученного классического критерия «минимум длины расписания» мы рассматриваем задачу на минимум целевой функции, являющейся произвольной регулярной функцией от моментов завершения операций. Таким образом, ввиду допустимости прерываний операций в произвольные моменты времени множество допустимых расписаний становится континуальным, а ввиду произвольности целевой функции дискретизация исходного множества допустимых расписаний (т.е. выделение какого-то конечного множества расписаний, среди которых доказуемо содержится хотя бы одно оптимальное) представляет собой нетривиальную проблему.

Видимо, в силу указанных выше обстоятельств, цеховые (многостадийные) задачи с допущением прерываний операций остаются до сих пор относительно слабо исследованным разделом теории расписаний. Достаточно сказать, что из трёх классических цеховых задач (open shop, flow shop, job shop) с допущением прерываний операций точные алгоритмы решения были известны к 2000 году лишь для задачи open shop (классический полиномиальный алгоритм Гонзалеса и Сани, 1976), тогда как описание первого алгоритма

точного решения задачи flow shop с прерываниями операций (для классического критерия минимум длины расписания) появилось лишь в 2006 году (Севастьянов и др. 2006). Авторами этой работы было установлено, что оптимальное расписание находится довольно простым «жадным» алгоритмом, в котором для каждого момента времени t и каждой машины M_i выполняемая на ней работа определяется как наиболее приоритетная работа из множества работ, доступных для выполнения на этой машине в данный момент времени. Единственным «тонким» моментом в том алгоритме является установление («угадывание») «правильных» приоритетов работ на машинах. Ясно однако, что эта проблема уже разрешима за конечное время, поскольку число различных комбинаций приоритетов работ на машинах конечно.

В настоящем проекте используемая в (Севастьянов и др. 2006) идея анализа свойств оптимальных решений успешно реализована для существенно более общей задачи – задачи job shop с произвольным регулярным критерием от моментов завершения операций. Установленных нами свойств оптимальных решений оказалось достаточно для разработки конструктивного метода решения этой сложной задачи. Так же как и в (Севастьянов и др. 2006) нами установлено, что оптимальное расписание может быть найдено жадным алгоритмом, при условии, что найдены (угаданы) подходящие приоритеты выполнения операций на каждой из машин. И хотя полученный таким путём алгоритм едва ли можно назвать «эффективным» (в общепринятом смысле), но с учётом несуществования последнего (так как задача NP-трудна в сильном смысле даже в весьма частных случаях), полученный нами конструктивный метод решения имеет определённую теоретическую ценность.

1.3.4 Эффективный алгоритм перечисления оптимальных и приближённых решений в задаче flow shop на двух машинах

При рассмотрении оптимизационных задач обычно отыскивается «наилучшее» решение (по отношению к заданной целевой функции). Но в реальных ситуациях процесс принятия решений, как правило, не бывает настолько простым, и для реальных задач мы выбираем какое-то «подходящее» решение из множества «достаточно хороших» решений по отношению к заданному критерию; при этом наше окончательное решение может зависеть от множества трудно формализуемых условий. В таких случаях нам может оказаться полезной возможность перебора (желательно, без повторений) множества всех приближённых решений, удовлетворяющих заданной оценке точности, с тем чтобы из этого множества выбрать (по каким-то неформальным критериям) решение, наиболее удовлетворяющее нашим требованиям.

Интересным частным случаем этой проблемы является перебор всех оптимальных решений конкретной оптимизационной задачи. Очевидно, это всегда можно сделать полным перебором всех допустимых (и даже, необязательно допустимых) решений, но в большинстве случаев такой подход, мягко говоря, не является «практически оправданным». С теоретической точки зрения такой метод также вряд ли является эффективным. С другой стороны, в ряде случаев при решении перечислительной задачи не удаётся избежать полного перебора всех решений. (Как, например, в задаче коммивояжёра, когда все $(n-1)!$ возможных маршрутов по n городам имеют одну и ту же суммарную длину.) И такие случаи, очевидно, не могут являться индикаторами неэффективности применённого метода. Действительно, едва ли справедливо назвать «неэффективным» алгоритм, который перечисляет **только** нужные нам решения, – даже если на некоторых входных данных его «выход» (число решений, выводимых им на печать) экспоненциально велик. Таким образом, для перечислительных задач не подходит классическое определение эффективности алгоритма. Становится существенным не общее время, затрачиваемое на вывод всех найденных решений, а время, затрачиваемое на нахождение каждого подходящего нам решения. Так мы приходим к понятию алгоритма с полиномиальной задержкой (polynomial delay), – такого, который затрачивает полиномиальное время на отыскание каждого следующего подходящего решения.

Помимо временной оценки, существенной характеристикой перечислительных алгоритмов является оценка максимального объёма памяти, запрашиваемой алгоритмом в процессе его работы. В некоторых случаях важность данной характеристики алгоритма выступает даже на первый план, что, в принципе, довольно понятно: временной ресурс, в принципе, не ограничен (время мы считаем «практически бесконечным»), тогда как ресурс памяти каждой вычислительной машины жёстко ограничен «в железе». Для перечислительных задач (с характерным экспоненциальным ростом размера «выхода») этот параметр становится актуальным в тех случаях, когда требуется перечисление элементов какого-то множества без повторений. В этих случаях возникает нетривиальная проблема такой организации вычислительного процесса, чтобы для обеспечения требования «неповторяемости» не требовалось хранения в памяти машины текущего множества уже найденных решений. Так мы приходим к понятию алгоритма с полиномиальной памятью (polynomial space algorithm).

Определение. Мы говорим, что перечислительный алгоритм имеет полиномиальную задержку, если время, затрачиваемое на генерирование каждого нового элемента вывода (называемого «элементарным решением»), а также время, необходимое для остановки алгоритма после вывода последнего решения, является полиномиальным от размера входа.

Определение. Мы говорим, что перечислительный алгоритм является алгоритмом с полиномиальной памятью, если на любом шаге алгоритма используемая им память компьютера не превосходит заранее известного полинома от размера входной информации.

Хотя направление перечислительных алгоритмов для оптимизационных задач известно уже несколько десятилетий (как минимум, со времени выхода работы (Джонсон и др. 1988), в которой были введены приведённые выше понятия алгоритмов с полиномиальной задержкой и с полиномиальной памятью), однако в теории расписаний данное направление едва ли можно назвать «хорошо разработанным». Это тем более справедливо по отношению к проблеме построения алгоритмов перечисления приближённых решений. (Похоже на то, что это является новым направлением исследований и для дискретной оптимизации в целом.)

В настоящем проекте предпринимается первая (по нашим сведениям) попытка подобных исследований по отношению к классической задаче теории расписаний, сформулированной Джонсоном в его пионерной работе (Джонсон, 1954), с которой, по сути, берёт своё начало современная «машинная» теория расписаний (т.е. теория расписаний для многостадийных процессов). В той работе Джонсон представил эффективный алгоритм решения следующей двухмашинной задачи. (Как выяснилось позже благодаря работе (Гэри и др. 1976), подобный результат является наилучшим возможным для многомашинной модели такого типа, поскольку уже аналогичная задача для трёхмашинной модели является NP-трудной в сильном смысле.)

Задача Джонсона. Задано множество из n работ $\{J_1, \dots, J_n\}$, которые предстоит выполнить на двух машинах: M_1 и M_2 . Каждая работа J_j состоит из двух операций O_{j1} , O_{j2} , имеющих длительности a_j и b_j . Операции должны быть выполнены в строго определённом порядке: сначала O_{j1} на машине M_1 , а затем O_{j2} на машине M_2 . Каждая машина также может выполнять свои операции только последовательно: одну за другой, в некотором порядке, выбираемом в процессе решения задачи. Целью решения задачи является отыскание расписания S минимальной длины (обозначаемой как $C_{\max}(S)$). Согласно стандартной классификации задач (Грэм и др. 1979), такая задача условно обозначается как $F2 \parallel C_{\max}$.

Как следует из общей теории расписаний, для минимизации длины расписания в системе flow shop достаточно ограничиться так называемыми активными расписаниями, каждое из которых однозначно определяется последовательностями операций на машинах M_1 и M_2 . Более того, как показано Джонсоном (Джонсон, 1954), нет необходимости перебирать все комбинации двух последовательностей. Достаточно рассмотреть только пары одинаковых последовательностей на двух машинах (определяемые единственной

перестановкой работ π). – Такие расписания называются перестановочными и обозначаются S_π .

В своём анализе оптимальности заданной последовательности работ Джонсон определил отношение « \leq » на множестве работ: $J_i \leq J_j$, если $\min\{a_i, b_j\} \leq \min\{a_j, b_i\}$. Он нашёл достаточное условие оптимальности последовательности работ:

Перестановка работ $\pi = (\pi_1, \dots, \pi_n)$ оптимальна, если

$$J_{\pi_i} \leq J_{\pi_j} \text{ для всех } i < j. \quad (1.3.4.1)$$

И хотя отношение « \leq » не определяет линейный порядок, Джонсон показал, что для любого примера данной задачи существует последовательность работ со свойством (1.3.4.1). (Далее будем называть такие последовательности Джонсоновскими.) Некоторые из этих последовательностей могут быть найдены за время $O(n \log n)$ путём применения так называемого специального правила Джонсона.

Легко видеть, что в случае, когда длительности всех операций различны, существует единственная перестановка работ, удовлетворяющая условию (1.3.4.1), поскольку отношение « \leq » становится транзитивным и антисимметричным. Однако в общем случае правило Джонсона порождает довольно широкое семейство оптимальных последовательностей работ. Среди большого числа работ, посвящённых полному или неполному перечислению Джонсоновских последовательностей, мы хотели бы выделить монографию Беллмана и др. (Беллман и др. 1982), где было выведено так называемое Общее Рабочее Правило, позволяющее генерировать, в принципе, любую Джонсоновскую последовательность. Это правило может быть легко трансформировано в эффективную процедуру перечисления всех Джонсоновских последовательностей без повторения.

Тем не менее, достаточно ясно, в общем случае множество оптимальных последовательностей работ для заданного примера не исчерпывается лишь Джонсоновскими последовательностями, поскольку правило (1.3.4.1), являясь достаточным условием оптимальности перестановки, в то же время не является необходимым. Таким образом, проблема перечисления всех оптимальных последовательностей работ в задаче $F2 \parallel C_{max}$ является более общей и представляет самостоятельный интерес. Ряд авторов обращались к этой теме, пытаясь описать какую-нибудь перечислительную процедуру, которая была бы более эффективной чем прямой перебор всех $n!$ Перестановок из n работ (см., например, (Било и Лопез, 1997), (Лин и Дэнг, 1999), (Пандит и Субраманьям, 1975), (Шварц, 1981)), однако они не могли (и не пытались) гарантировать ни какой-либо эффективности своих алгоритмов (в смысле полиномиальной задержки и полиномиальной памяти), ни даже их способности перечисления нужных нам решений без повторений. Как было заявлено Ченгом

в 1992 году (Ченг, 1992), «проблема разработки алгоритма перечисления всех оптимальных последовательностей работ в задаче $F2 // C_{max}$ до сих пор остаётся открытой».

Она действительно оставалась открытой до настоящего времени. В процессе работы по данному проекту был разработан первый (насколько нам известно) эффективный алгоритм перечисления всех оптимальных последовательностей работ в двухмашинной задаче Джонсона (где «эффективность алгоритма» понимается в смысле его полиномиальной задержки и полиномиальной памяти). На самом деле, нами получен гораздо более общий результат, поскольку он касается перечисления всех приближённых решений, удовлетворяющих заданной оценке точности.

Определение. Обозначая через B суммарную нагрузку машины M_2 (которая, как известно, является нижней оценкой длины оптимального расписания), будем называть последовательность π номеров работ Δ -оптимальной, если выполнено неравенство:

$$C_{max}(S_{\pi}) \leq B + \Delta$$

Нашим главным результатом является разработка алгоритма A_{Δ} со следующими свойствами.

Теорема 1. Алгоритм A_{Δ} перечисляет все Δ -оптимальные перестановки n работ без повторений, затрачивая не более чем $O(n \log n)$ единиц времени на отыскание каждой Δ -оптимальной перестановки и на остановку алгоритма после отыскания последнего решения. При этом на каждом шаге алгоритма используется не более чем $O(n)$ ячеек памяти вычислительной машины.

Нетрудно понять, что данный алгоритм может быть также использован для перечисления всех оптимальных перестановок.

Нами также выявлено интересное свойство связности дискретного множества Δ -оптимальных решений. Пусть для заданного примера задачи и заданной величины $\Delta \geq 0$ определён граф $G(\Delta)$, в котором все Δ -оптимальные перестановки n работ представлены вершинами графа, а две вершины π' и π'' соединены ребром, если и только если соответствующие перестановки работ получаются одна из другой транспозицией двух соседних элементов. Тогда справедлива

Теорема 2. Для любого $\Delta \geq 0$ граф $G(\Delta)$ связан.

Отсюда, в частности, следует, что любая оптимальная перестановка работ может быть получена из любой Джонсоновской (оптимальной) перестановки работ последовательной транспозицией двух соседних элементов, так что при этом все промежуточные перестановки также оптимальны.

1.3.5 Задача OPEN SHOP с маршрутизацией

Рассматривается цеховая задача открытого типа с маршрутизацией, которая является обобщением двух классических задач дискретной оптимизации: цеховой задачи open shop (сформулированной выше, в п. Б, и обозначаемой как $O \parallel C_{\max}$) и метрической задачи коммивояжера (MTSP).

Если прерывания в процессе выполнения операций запрещены, то задача $O \parallel C_{\max}$ является NP-трудной для фиксированного числа машин $m \geq 3$ и NP-трудной в сильном смысле, если число машин является частью входа задачи (Вильямсон и др. 1997).

Задача MTSP. Задан неориентированный полный граф $G = (V, E)$. Вес $\tau_{ij} \in Z^+$ ребра $e_{ij} = [v_i, v_j]$ определяет расстояние между вершинами v_i и v_j . Расстояния удовлетворяют неравенству треугольника. Требуется найти гамильтонов цикл T (обход) минимального веса $|T| = \sum_{e_{ij} \in T} \tau_{ij}$.

Задача MTSP является NP-трудной в сильном смысле (Гэри и Джонсон, 1979).

Цеховая задача открытого типа с маршрутизацией. Рассмотрим задачу, которая является обобщением задач $O \parallel C_{\max}$ и MTSP. Предположим, что работы расположены в узлах транспортной сети G и для выполнения работы J_j каждая машина должна переместиться в точку v_j . Изначально все машины расположены в одной вершине v_0 и должны вернуться в нее после выполнения всех работ. Все машины перемещаются с одинаковой скоростью и затрачивают на перемещение из вершины v_i в вершину v_j время τ_{ij} . Пусть s_{ik} и C_{ik} обозначают моменты начала и завершения выполнения операции O_{ik} , соответственно. Тогда, если машина M_k выполняет последовательно работы J_i и J_j , то выполнение работы J_j на машине M_k не может начаться ранее чем через время τ_{ij} после окончания машиной M_k работы J_i , то есть $s_{jk} \geq C_{ik} + \tau_{ij}$. Все машины могут начать движение или обслуживание работ с момента времени 0. Под длиной расписания \tilde{C}_{\max} будем понимать длину временного интервала между моментом 0 и моментом времени, когда последняя машина вернется в вершину v_0 после выполнения всех работ. Требуется найти минимальное по длине расписание для перемещения машин и выполнения работ.

Согласно принятой трёхпольной системе обозначений для задач теории расписаний (Грэм и др. 1979) обозначим сформулированную задачу как $RO \parallel \tilde{C}_{\max}$ (или $ROm \parallel \tilde{C}_{\max}$ для

случая с m фиксированными машинами). В поле β будем записывать следующие дополнительные ограничения на параметры задачи. Запись $pmtn$ означает, что в процессе выполнения работ разрешены прерывания.

Запись $|V|=k$ означает, что число вершин в графе ограничено числом k . Запись $EasyTSP$ означает, что на заданной матрице расстояний задача коммивояжера разрешима за полиномиальное время.

Цеховые задачи открытого типа с маршрутизацией введены в работах Авербаха, Бермана и Черных (Авербах и др. 2005, 2006). Задача $RO \parallel \tilde{C}_{\max}$ является NP-трудной в сильном смысле, даже если работы требуется выполнить одной машиной или все работы лежат в одной вершине. В первом случае из задачи $RO \parallel \tilde{C}_{\max}$ получается задача MTSP, во втором случае – задача $O \parallel C_{\max}$. Более того, в (Авербах и др. 2005) показано, что задача $RO \parallel |V|=2 \mid \tilde{C}_{\max}$ с двумя машинами на двухвершинном графе NP-трудна в обычном смысле. В таблице 1 представлены известные результаты о сложности и аппроксимируемости цеховых задач открытого типа с маршрутизацией. В первой колонке указан тип рассматриваемой задачи. В случае, когда задача является полиномиально разрешимой, во второй колонке указывается трудоемкость алгоритма, находящего оптимальное решение, и третья колонка остается незаполненной. Если соответствующая задача является NP-трудной (NP-трудной в сильном смысле), то во второй колонке приведена запись NP-h. (NP-h.s.s.), со ссылкой на авторов результата, и в третьей колонке указана оценка точности наилучшего известного ρ -приближённого алгоритма её решения.

Таблица 3 – Результаты работы приближенных алгоритмов.

№	ЗАДАЧА	Сложность точного решения	Точность прил. алг.
1	$RO \parallel \tilde{C}_{\max}$	NP-h.s.s. (Гэри и Джонсон, 1979)	$O(\sqrt{m})$
2	$RO2 \parallel \tilde{C}_{\max}$	NP-h.s.s. (Гэри и Джонсон, 1979)	13/8
3	$RO2 \mid EasyTSP \mid \tilde{C}_{\max}$	NP-h. (Авербах и др. 2006)	4/3
4	$RO2 \mid V =2 \mid \tilde{C}_{\max}$	NP-h. (Авербах и др. 2006)	$(1+\epsilon)$ (FPTAS)
5	$RO \mid V =2 \mid \tilde{C}_{\max}$	NP-h.s.s. (Вильямсон и др. 1997)	7/2
6	$RO \mid pmtn, V =2 \mid \tilde{C}_{\max}$	NP-h.s.s. (Пяткин и Черных, 2012)	2
7	$RO2 \mid pmtn, V =2 \mid \tilde{C}_{\max}$	$O(n)$ (Пяткин и Черных, 2012)	

Все приведённые в третьем столбце таблицы 3 результаты по приближённым алгоритмам получены авторами проекта. Ниже дадим более подробное описание результата, представленного в строке 4 таблицы.

Отметим, что этот результат улучшает результат статьи (Авербах и др. 2005), в которой для решения задачи $RO2 \mid |V| = 2 \mid \tilde{C}_{\max}$ предложен 6/5-приближённый алгоритм.

Длиной работы J_j называется общее время выполнения всех ее операций, то есть $d_j = \sum_{i=1}^m p_{ij}$. Нагрузкой l_i машины M_i называется сумма длительностей всех операций, выполняемых на этой машине, $d_{\max} = \max_{J_j \in J} d_j$, $l_{\max} = \max_i l_i$ и $\bar{C}_{\max} = \max\{l_{\max}, d_{\max}\}$.

Две машины и две вершины

Пусть заданы множество работ J , две машины A и B и транспортная сеть, состоящая из двух вершин v_0 и v_1 . Пусть J^0 и J^1 множество работ, расположенных в вершинах v_0 и v_1 соответственно. В момент времени 0 обе машины находятся в вершине v_0 . Каждой машине требуется время τ для перемещения из одной вершины в другую. Требуется найти расписание минимальной длины. Пусть $d_{\max}^0 = \max_{J_j \in J^0} d_j$ и $d_{\max}^1 = \max_{J_j \in J^1} d_j$. Обозначим через p_{jA} и p_{jB} длительности работы $J_j \in J$ на машинах A и B соответственно. Выберем из множества J работу, на которой достигается величина $\max\{\max\{p_{jA} \mid p_{jA} \leq p_{jB}\}, \max\{p_{jB} \mid p_{jB} < p_{jA}\}\}$. Без ограничения общности предположим, что это работа J_1 и $p_{1A} = \max\{\max\{p_{jA} \mid p_{jA} \leq p_{jB}\}, \max\{p_{jB} \mid p_{jB} < p_{jA}\}\}$. Тогда имеет место следующий результат.

Теорема. Для любого примера I задачи $RO2 \mid |V| = 2 \mid \tilde{C}_{\max}$, в котором выполняется одно из двух условий:

- (i) $J_1 \in J^0$,
- (ii) $J_1 \in J^1$ и $d_1 \geq l_{\max}$,

длина оптимального расписания равна $\max\{l_{\max} + 2\tau, d_{\max}^0, d_{\max}^1\}$, и оно может быть найдено за время, линейное от числа работ.

Назовем пример трудным, если $J_1 \in J^1$ и $d_1 < l_{\max}$. Расписание называется каноническим, если множество работ может быть разбито не более чем в восемь непересекающихся подмножеств, для которых выполнены следующие условия.

1. Все работы, принадлежащие одному подмножеству, расположены в одной вершине.

2. Все работы, принадлежащие одному подмножеству, выполняются блоком на каждой машине.

3. Все работы, принадлежащие одному подмножеству, выполняются либо сначала на машине A , потом на машине B , либо наоборот.

4. Машины A и B выполняют работы каждого подмножества в одном и том же порядке. Более того, этот порядок совпадает с порядком Джонсона для соответствующей цеховой задачи потокового типа на двух машинах.

В работе (Кононов, 2012) показано, что для любого трудного примера существует каноническое оптимальное расписание. Более того, длина канонического расписания зависит только от разбиения исходного множества работ на непересекающиеся подмножества и может быть вычислена за полиномиальное от числа работ время. Используя свойства канонических расписаний, трудный пример задачи $RO2 \mid |V| = 2 \mid \tilde{C}_{\max}$ может быть решен алгоритмом динамического программирования за время $O(n\Delta^{24})$, где $\Delta = \sum_{j \in J} d_j$.

Используя стандартную технику округления (Ибарра и Ким, 1975), можно трансформировать алгоритм динамического программирования во вполне полиномиальную приближённую схему, то есть для фиксированного $\varepsilon > 0$ построить $(1+\varepsilon)$ -приближённый алгоритм, время работы которого ограничено полиномом от числа работ и величины $1/\varepsilon$.

Теорема. Для задачи $RO2 \mid |V| = 2 \mid \tilde{C}_{\max}$ существует вполне полиномиальная приближённая схема.

1.4. Оценка сложностного статуса и построение эффективных алгоритмов с оценками точности для задач кластерного анализа

1.4.1 Введение

Аннотация. Доказана NP-полнота некоторых задач выбора подпоследовательности из последовательности векторов евклидова пространства, состоящей из конечного числа членов. Предполагается, что искомая подпоследовательность содержит фиксированное число векторов близких между собой по критерию минимума суммы квадратов расстояний, причем выбор векторов подчинён условию: разность между номерами последующего и предыдущего выбираемых векторов ограничена сверху и снизу некоторыми константами.

Предметом исследования настоящей работы являются дискретные экстремальные задачи выбора из последовательности векторов евклидова пространства, состоящей из конечного числа членов, подпоследовательности элементов близких по критерию минимума суммы квадратов расстояний при наличии ограничений на номера выбираемых векторов. Цель исследования – анализ алгоритмической сложности этих задач.

Представленные ниже результаты дополняют работу [48], где было установлено, что к числу NP-трудных задач относятся следующие тесно связанные между собой оптимизационные задачи кластерного анализа и выбора подмножества в конечном множестве векторов евклидова пространства.

Задача VS-1 (Vector Subset 1).

Дано: множество $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ векторов из \mathcal{R}^q и натуральное число $M > 1$.

Найти: подмножество $C \subseteq \mathcal{Y}$ векторов такое, что целевая функция

$$F_1(C) = \frac{1}{|C|} \left\| \sum_{\mathbf{y} \in C} \mathbf{y} \right\|^2 + \sum_{\mathbf{y} \in \mathcal{Y} \setminus C} \|\mathbf{y}\|^2$$

максимальна, при ограничении $|C| = M$ на мощность подмножества C .

Задача VS-2 (Vector Subset 2).

Дано: множество $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ векторов из \mathcal{R}^q и натуральное число $M > 1$.

Найти: подмножество $C \subseteq \mathcal{Y}$ векторов такое, что целевая функция

$$F_2(C) = \sum_{\mathbf{y} \in C} \|\mathbf{y} - \bar{\mathbf{y}}(C)\|^2,$$

где $\bar{\mathbf{y}}(C) = \frac{1}{|C|} \sum_{\mathbf{y} \in C} \mathbf{y}$, минимальна, при ограничении $|C| = M$ на мощность искомого подмножества.

Задача VS-3 (Vector Subset 3).

Дано: множество $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ векторов из \mathcal{R}^q и натуральное число $M > 1$.

Найти: подмножество $C \subseteq \mathcal{Y}$ векторов такое, что целевая функция

$$F_3(C) = \sum_{\mathbf{y} \in C} \sum_{\mathbf{z} \in C} \|\mathbf{y} - \mathbf{z}\|^2$$

минимальна, при ограничении $|C| = M$ на мощность искомого подмножества.

Задача MSSC-Case (Minimum Sum-of-Squares Clustering, special Case).

Дано: множество $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ векторов из \mathcal{R}^q , натуральное число $M > 1$.

Найти: разбиение множества \mathcal{Y} на $J = N - M + 1$ непустых кластеров C_1, C_2, \dots, C_J такое, что мощность одного из кластеров равна M и

$$F_4(C_1, C_2, \dots, C_J) = \sum_{j=1}^J \sum_{\mathbf{y} \in C_j} \|\mathbf{y} - \bar{\mathbf{y}}(C_j)\|^2 \rightarrow \min,$$

где $\bar{\mathbf{y}}(C_j) = \frac{1}{|C_j|} \sum_{\mathbf{y} \in C_j} \mathbf{y}$, $j = 1, 2, \dots, J$, – центр j -го кластера.

Для целевых функций задач VS-1, VS-2 и VS-3 выполняются следующие соотношения [48]:

$$F_2(C) = \sum_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|^2 - F_1(C) = \frac{1}{2|C|} F_3(C). \quad (1.4.1.1)$$

Поэтому задачи VS-1, VS-2 и VS-3 полиномиально эквивалентны. Кроме того, если считать, что в задаче MSSC-Case мощность, например, первого кластера C_1 зафиксирована и равна M , то имеет место равенство [48]

$$F_4(C_1, C_2, \dots, C_J) = F_2(C_1), \quad (1.4.1.2)$$

так как из условий задачи следует, что мощности кластеров из совокупности $\{C_1, C_2, \dots, C_J\} \setminus C_1$ равны 1. Поэтому задачи MSSC-Case и VS-2 эквивалентны. Доказано [48], что в форме верификации свойств сформулированные задачи NP-полны в сильном смысле.

Одна из возможных содержательных трактовок проблемы анализа данных, которая приводит к решению сформулированных задач, состоит в следующем [48]. Имеется таблица, содержащая результаты измерения набора числовых информационно значимых характеристик для совокупности некоторых материальных объектов. Часть объектов из этой

совокупности идентичны и имеют одинаковые характеристики. Число идентичных объектов известно. Оставшиеся объекты различны и имеют отличающиеся характеристики. В каждом результате измерения, представленном в таблице, имеется ошибка, причем соответствие между объектом и набором неизвестно. Характеристики идентичных объектов, в отличие от характеристик остальных объектов, имеют принципиальную информационную ценность. Требуется, используя критерий минимума суммы квадратов расстояний, найти подмножество наборов, соответствующих идентичным объектам и оценить по результатам измерения набор характеристик этих объектов (учитывая, что данные содержат ошибку измерения).

Задачи, рассмотренные в настоящей работе, индуцируются близкой в содержательном плане проблемой. Отличие этой проблемы от сформулированной выше состоит лишь в том, что элементы таблицы упорядочены по времени, причем известно, что временной интервал между двумя последовательными результатами измерения характеристик идентичных объектов ограничен сверху и снизу некоторыми константами. Подобные этой содержательные проблемы с временными ограничениями на результаты измерения каких-либо информационно значимых характеристик весьма актуальны, в частности, при помехоустойчивой off-line обработке числовых и векторных последовательностей (см., например, [49]-[54] и цитированные там работы), которые в приложениях трактуются как дискретные одномерные или многомерные сигналы.

Поскольку модель анализируемых данных практически та же (за исключением дополнительных ограничений), что и в работе [48], рассмотренные ниже дискретные экстремальные задачи по своей сути являются аналогами приведенных выше задач. В рассматриваемых задачах предполагается, что входными данными являются не множества, а векторные последовательности, причем имеются ограничения на номера выбираемых векторов из входной последовательности. Эти ограничения соответствуют априорным данным о времени измерений характеристик идентичных объектов. Мотивацией исследований послужил тот факт, что статус сложности этих задач был неизвестен.

1.4.2 Задачи выбора подпоследовательности

Для учёта ограничений на номера выбираемых векторов в приведенных ниже формулировках задач при записи целевых функций вместо суммирования по элементам множеств (см. предыдущий параграф) используется суммирование по номерам (индексам) элементов последовательности. Кроме того, с этой же целью в формулировки задач вводятся натуральные константы T_{\min} и T_{\max} .

Положим $\mathcal{N} = \{1, 2, \dots, N\}$. В форме верификации свойств задачи на последовательностях с ограничениями формулируются следующим образом.

Задача VSS1 (T_{\min}, T_{\max}) (Vector Subsequence in a Sequence 1).

Дано: последовательность (набор) $\mathcal{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ векторов из \mathcal{R}^q , натуральное число $M > 1$ и положительное число A . *Вопрос*: существует ли подмножество $\mathcal{M} = \{n_1, n_2, \dots, n_M\} \subseteq \mathcal{N}$ номеров элементов набора \mathcal{Y} такое, что

$$f_1(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \left\| \sum_{m=1}^M \mathbf{y}_{n_m} \right\|^2 + \sum_{m \notin \mathcal{M}} \|\mathbf{y}_m\|^2 \geq A,$$

при ограничениях

$$1 \leq T_{\min} \leq n_m - n_{m-1} \leq T_{\max} \leq N - 1, \quad m = 2, 3, \dots, M, \quad (1.4.2.1)$$

на элементы подмножества \mathcal{M} ?

Задача VSS2 (T_{\min}, T_{\max}) (Vector Subsequence in a Sequence 2).

Дано: последовательность $\mathcal{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ векторов из \mathcal{R}^q , натуральное число $M > 1$ и положительное число B . *Вопрос*: существует ли подмножество $\mathcal{M} = \{n_1, n_2, \dots, n_M\} \subseteq \mathcal{N}$ номеров элементов последовательности \mathcal{Y} такое, что

$$f_2(\mathcal{M}) = \sum_{m=1}^M \|\mathbf{y}_{n_m} - \bar{\mathbf{y}}(\mathcal{M})\|^2 \leq B,$$

где $\bar{\mathbf{y}}(\mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{n \in \mathcal{M}} \mathbf{y}_n$, при ограничениях (1.4.2.1) на элементы искомого подмножества \mathcal{M} ?

Задача VSS3 (T_{\min}, T_{\max}) (Vector Subsequence in a Sequence 3).

Дано: последовательность $\mathcal{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ векторов из \mathcal{R}^q , натуральное число $M > 1$ и положительное число C . *Вопрос*: существует ли подмножество $\mathcal{M} = \{n_1, n_2, \dots, n_M\} \subseteq \mathcal{N}$ номеров элементов последовательности \mathcal{Y} такое, что

$$f_3(\mathcal{M}) = \sum_{i=1}^M \sum_{j=1}^M \|\mathbf{y}_{n_i} - \mathbf{y}_{n_j}\|^2 \leq C,$$

при ограничениях (1.4.2.1) на элементы искомого подмножества \mathcal{M} ?

Задача MSSC-Case-S (T_{\min}, T_{\max}) (Minimum Sum-of-Squares Clustering, special Case for a Sequence).

Дано: последовательность $\mathcal{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N)$ векторов из \mathcal{R}^q натуральное число $M > 1$ и положительное число D . Вопрос: существует ли разбиение множества \mathcal{N} на $J = N - M + 1$ непустых подмножеств $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_J$ такое, что $|\mathcal{M}_1| = M$ и

$$f_4(\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_J) = \sum_{j=1}^J \sum_{m \in \mathcal{M}_j} \|\mathbf{y}_m - \bar{\mathbf{y}}(\mathcal{M}_j)\|^2 \leq D,$$

где $\bar{\mathbf{y}}(\mathcal{M}_j) = \frac{1}{|\mathcal{M}_j|} \sum_{n \in \mathcal{M}_j} \mathbf{y}_n$, $j = 1, 2, \dots, J$, при ограничениях (1.4.2.1) на элементы подмножества \mathcal{M}_1 ?

1.4.3 Анализ алгоритмической сложности

Во-первых, отметим, что в случае $T_{\min} = T_{\max}$ все задачи, очевидно, решаются за полиномиальное время. Поэтому далее всюду будем считать, что $T_{\min} < T_{\max}$.

Во-вторых, заметим, что задачи в парах VS-1 и VSS1(1, $N - 1$), VS-2 и VSS2(1, $N - 1$), VS-3 и VSS3(1, $N - 1$), а также MSSC-Case и MSSC-Case-S(1, $N - 1$) эквивалентны. При этом задачи VSS1(1, $N - 1$), VSS2(1, $N - 1$), VSS3(1, $N - 1$), а также MSSC-Case-S(1, $N - 1$) являются обобщением задач VSS1(T_{\min}, T_{\max}), VSS2(T_{\min}, T_{\max}), VSS3(T_{\min}, T_{\max}) и MSSC-Case-S(T_{\min}, T_{\max}) соответственно в случае, когда параметры T_{\min} и T_{\max} являются частью входа. Поэтому в этом случае NP-полнота задач выбора подпоследовательностей следует из результатов работы [48], где была показана NP-полнота задач VS1, VS2, VS3 и MSSC-Case.

Наконец, в-третьих, легко проверить, что целевые функции задач выбора подпоследовательностей связаны формулами

$$f_2(\mathcal{M}) = \sum_{n \in \mathcal{M}} \|\mathbf{y}_n\|^2 - f_1(\mathcal{M}) = \frac{1}{2|\mathcal{M}|} f_3(\mathcal{M}), \quad (1.4.3.1)$$

$$f_4(\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_J) = f_2(\mathcal{M}_1), \quad (1.4.3.2)$$

которые аналогичны (1.4.1.1) и (1.4.1.2). Следовательно, задачи VSS1(T_{\min}, T_{\max}), VSS2(T_{\min}, T_{\max}) и VSS3(T_{\min}, T_{\max}) полиномиально эквивалентны, а задача MSSC-Case-S(T_{\min}, T_{\max}) эквивалентна задаче VSS2(T_{\min}, T_{\max}). Поэтому в случае, когда параметры T_{\min} и T_{\max} не являются частью входа, достаточно проанализировать сложность одной из этих задач. Докажем NP-полноту задачи VSS3(T_{\min}, T_{\max}).

Заметим, что доказательство существенно отличается для случаев $T_{\max} \geq 2T_{\min}$ и $T_{\max} \leq 2T_{\min} - 1$, а именно: к этим случаям $VSS3(T_{\min}, T_{\max})$ полиномиально сводятся две следующие разные NP-полные задачи.

Задача Clique in a regular graph (Клика в однородном графе).

Дано: однородный граф G степени $d \geq 2$ и натуральное число k . Вопрос: существует ли в этом графе такое подмножество вершин мощности k , что любые две вершины из этого подмножества связаны ребром?

Задача MaxCut (Максимальный разрез).

Дано: граф G и натуральное число t . Вопрос: существует ли в этом графе такое разбиение множества вершин на два подмножества, что число рёбер с концами в разных подмножествах не меньше t ?

NP-полнота первой из этих задач доказана в [55]. Вторая задача относится к числу классических NP-полных задач [56].

NP-полнота задачи $VSS3(T_{\min}, T_{\max})$ вытекает из следующих двух лемм. Перед их доказательством заметим, что эта задача, очевидно, принадлежит классу NP для любых натуральных T_{\min} и T_{\max} .

Лемма 1. Если $T_{\max} \geq 2T_{\min}$, то задача $VSS3(T_{\min}, T_{\max})$ NP-полна в сильном смысле.

Доказательство. Используя пример задачи Клика в однородном графе G степени d , содержащем p вершин и q рёбер, а также число k построим следующий пример задачи $VSS3(T_{\min}, T_{\max})$. Положим $M = k + p - 1$ и $C = 2(d - 1)(k - 1)k + 2kd(p - 1)$.

Входную последовательность \mathcal{U} составим из $(q + 1)$ -мерных векторов трёх типов: нулевых, основных и вспомогательных. Возьмем $p - 1$ нулевых, p основных и $2(p - 1)(T_{\min} - 1)$ вспомогательных векторов, т.е. всего $N = 2(p - 1)T_{\min} + 1$ членов.

Каждой вершине v_i графа G поставим в соответствие $(q + 1)$ -мерный основной вектор \mathbf{u}_i , в котором $(q + 1)$ -ая координата равна нулю, а j -я координата равна 1, если ребро e_j инцидентно вершине v_i , и 0 в противном случае ($i = 1, 2, \dots, p$, $j = 1, 2, \dots, q$). Заметим, что у основных векторов d координат равны 1, а остальные равны 0 по построению.

Положим $X = \lfloor \sqrt{C + 1} \rfloor$ и в качестве m -го вспомогательного вектора возьмем вектор, у которого $(q + 1)$ -ая координата равна mX , $m = 1, 2, \dots, 2(p - 1)(T_{\min} - 1)$, а все остальные координаты равны 0. В нулевом векторе все компоненты равны 0.

Последовательность \mathcal{U} векторов сформируем по следующему правилу. Сначала в произвольном порядке запишем все основные векторы. Затем в эту последовательность

между основными векторами вставим нулевые векторы. Далее, в полученную последовательность между каждыми двумя последовательными векторами вставим $T_{\min} - 1$ вспомогательных векторов. Схематически, построенная последовательность имеет следующий вид (для примера $T_{\min} = 4$):

$$\mathcal{Y} = y_1 y_2 \dots y_N = u_1 \mathbf{x} \mathbf{x} \mathbf{x} \mathbf{0} \mathbf{x} \mathbf{x} \mathbf{x} u_2 \mathbf{x} \mathbf{x} \mathbf{x} \mathbf{0} \mathbf{x} \mathbf{x} \mathbf{x} u_3 \dots \mathbf{0} \mathbf{x} \mathbf{x} \mathbf{x} u_p,$$

где $\mathbf{0}$ – нулевой вектор, \mathbf{x} – вспомогательный вектор, а u_i – основной вектор.

Покажем, что у построенной последовательности \mathcal{Y} (т.е. в примере задачи $VSS3(T_{\min}, T_{\max})$) существует набор \mathcal{M} из M номеров, компоненты которого удовлетворяют условиям (1.4.2.1) и $f_3(\mathcal{M}) \leq C$, тогда и только тогда, когда в графе G (т.е. в примере задачи Клика в однородном графе) есть клика мощности k .

Сначала вычислим квадраты расстояний между векторами. Очевидно, что квадрат расстояния от вспомогательного вектора до любого другого вектора не меньше $X^2 \geq C + 1$. Квадрат расстояния от нулевого до любого основного вектора равен d , так как у основного вектора ровно d компонент равны 1, а остальные равны 0. Наконец, квадрат расстояния между основными векторами равен $2d - 2$, если соответствующие им вершины в графе G смежны, и $2d$ если они несмежны.

Необходимость. Если в G имеется клика размера k , то выбрав в качестве совокупности \mathcal{M} номера всех нулевых векторов из последовательности \mathcal{Y} , а также номера тех основных векторов этой последовательности, которые соответствуют вершинам клики, получим $f_3(\mathcal{M}) = C$. Выполнение условия (1.4.2.1) следует из построения и неравенства $T_{\max} \geq 2T_{\min}$.

Таким образом, у последовательности \mathcal{Y} существует набор \mathcal{M} номеров, компоненты которого удовлетворяют условиям задачи $VSS3(T_{\min}, T_{\max})$.

Достаточность. Предположим, что у последовательности \mathcal{Y} существует набор \mathcal{M} номеров членов этой последовательности, удовлетворяющих условиям (1.4.2.1) и $f_3(\mathcal{M}) \leq C$. Тогда из последнего неравенства следует, что набор \mathcal{M} не содержит номеров вспомогательных векторов.

Обозначим через a число номеров основных векторов в наборе \mathcal{M} . Тогда \mathcal{M} содержит $k + p - 1 - a$ номеров нулевых векторов, так как $|\mathcal{M}| = M = k + p - 1$ и \mathcal{M} не содержит номеров вспомогательных векторов.

Из построения следует, что суммарное число нулевых векторов в последовательности не превосходит $p - 1$. Поэтому $k + p - 1 - a \leq p - 1$. Откуда следует неравенство $a \geq k$. С

другой стороны, из построения следует, что число a основных векторов в последовательности не превосходит p . Таким образом, $a \in \{k, k+1, \dots, p\}$.

Учитывая, что квадрат расстояния от нулевого вектора до любого основного равен d , а минимальный квадрат расстояния между основными векторами равен $2d-2$, для целевой функции задачи имеем оценку

$$f_3(\mathcal{M}) \geq 2d(k+p-1-a) + (2d-2)(a-1)a.$$

Нетрудно проверить, что правая часть этого неравенства, как функция $f(a)$, строго возрастает при $d \geq 2$ и $a \in [k, p]$. Следовательно, эта функция принимает наименьшее значение на левой границе отрезка $[k, p]$, т.е. в точке k , причём это значение $f(k) = C$. Поэтому для целевой функции имеем оценку $f_3(\mathcal{M}) \geq C$. Но по нашему предположению справедливо условие $f_3(\mathcal{M}) \leq C$. Следовательно, для выполнения этого условия требуется, чтобы набор \mathcal{M} содержал ровно k номеров основных векторов, квадрат расстояния между любыми двумя из которых равен $2d-2$. Но тогда в графе G (из задачи Клика в однородном графе) соответствующие этим векторам вершины образуют клику. Лемма 1 доказана.

Поскольку $T_{\max} > T_{\min}$, из леммы 1 следует NP-полнота задачи $VSS3(1, T_{\max})$ для любого $T_{\max} > 1$.

Замечание 1. NP-полнота в сильном смысле рассмотренного случая задачи $VSS3(T_{\min}, T_{\max})$ следует из того, что числовые значения входных параметров этого случая задачи ограничены полиномом от размера входа задачи Клика в однородном графе.

Лемма 2. Если $T_{\max} \leq 2T_{\min} - 1$ и $T_{\min} \geq 2$, то задача $VSS3(T_{\min}, T_{\max})$ NP-полна в сильном смысле.

Доказательство. Построим следующий пример задачи $VSS3(T_{\min}, T_{\max})$, используя пример задачи Максимальный разрез, т.е. граф G с p вершинами и q рёбрами, а также положительное целое число t . Обозначим через d_i , $i = 1, 2, \dots, p$, степень i -й вершины графа G . Положим $M = 2p - 1$ и $C = 8pq - 4q - 8t$.

Входную последовательность \mathcal{Y} , содержащую всего $N = 2(p-1)T_{\min} + p + 1$ членов, составим из векторов трёх типов: нулевых, вспомогательных и основных. Возьмем $p-1$ нулевых, $N = 2(p-1)(T_{\min} - 1)$ вспомогательных и $2p$ основных векторов.

В качестве вспомогательных и нулевых используем те же векторы, что и при доказательстве леммы 1. Напомним, что в нулевом векторе все компоненты равны 0, а в m -м вспомогательном векторе $(q+1)$ -ая координата равна mX , $m = 1, 2, \dots, 2(p-1)(T_{\min} - 1)$, где $X = \left\lfloor \sqrt{C+1} \right\rfloor$, а остальные координаты равны 0.

Ориентируем все рёбра графа G произвольным образом. Каждой вершине v_i , $i = 1, 2, \dots, p$, ориентированного графа поставим в соответствие $(q+1)$ -мерный вектор \mathbf{u}_i , в котором $(q+1)$ -ая координата равна нулю, а j -я координата ($j = 1, 2, \dots, q$) равна 1, если дуга e_j исходит из вершины v_i , равна -1, если дуга e_j входит в v_i , и равна 0, если дуга e_j неинцидентна v_i . Положим $\mathbf{z}_i = -\mathbf{u}_i$, $i = 1, 2, \dots, p$. В качестве основных векторов возьмем векторы \mathbf{u}_i и \mathbf{z}_i , $i = 1, 2, \dots, p$.

Последовательность \mathcal{Y} векторов сформируем по следующему правилу. Сначала в произвольном порядке запишем все пары, состоящие из основных векторов, и получим последовательность $\mathcal{Y} = \mathbf{u}_1 \mathbf{z}_1 \mathbf{u}_2 \mathbf{z}_2 \dots \mathbf{u}_p \mathbf{z}_p$. Далее, в каждую третью позицию этой последовательности вставим нулевой вектор. Наконец, слева и справа от каждого нулевого вектора вставим по $T_{\min} - 1$ вспомогательных векторов. Схематически, построенная последовательность имеет следующий вид (для примера $T_{\min} = 4$):

$$\mathcal{Y} = \mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_N = \mathbf{u}_1 \mathbf{z}_1 \text{xxx} \mathbf{0} \text{xxx} \mathbf{u}_2 \mathbf{z}_2 \text{xxx} \mathbf{0} \text{xxx} \mathbf{u}_3 \mathbf{z}_3 \dots \mathbf{0} \text{xxx} \mathbf{u}_p \mathbf{z}_p.$$

Покажем, что у построенной последовательности \mathcal{Y} существует набор \mathcal{M} из M номеров, удовлетворяющий условиям (1.4.2.1) и $f_3(\mathcal{M}) \leq C$, тогда и только тогда, когда в графе G (т.е. в задаче MaxCut) существует разрез, содержащий не менее t рёбер.

Сначала вычислим квадраты расстояний между векторами. От вспомогательного вектора до любого другого вектора квадрат расстояния не меньше $X^2 \geq C + 1$. Квадрат расстояния от нулевого вектора до векторов \mathbf{u}_i и \mathbf{z}_i равен степени d_i вершины v_i графа G . Легко проверить, что если вершины v_i и v_k несмежны, то

$$\|\mathbf{u}_i - \mathbf{u}_k\|^2 = \|\mathbf{z}_i - \mathbf{z}_k\|^2 = \|\mathbf{u}_i - \mathbf{z}_k\|^2 = \|\mathbf{z}_i - \mathbf{u}_k\|^2 = d_i + d_k, \quad (1.4.3.1)$$

а если вершины v_i и v_k смежны, то

$$\|\mathbf{u}_i - \mathbf{u}_k\|^2 = \|\mathbf{z}_i - \mathbf{z}_k\|^2 = d_i + d_k + 2, \quad (1.4.3.2)$$

$$\|\mathbf{u}_i - \mathbf{z}_k\|^2 = \|\mathbf{z}_i - \mathbf{u}_k\|^2 = d_i + d_k - 2. \quad (1.4.3.3)$$

Необходимость. Если граф G содержит разрез $\{V_1, V_2\}$, где $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V(G)$, с $t_0 \geq t$ рёбрами, то выберем в набор \mathcal{M} номера всех нулевых векторов и номера всех векторов \mathbf{u}_i , для которых $v_i \in V_1$, а также номера всех векторов \mathbf{z}_i , для которых $v_i \in V_2$. Нетрудно заметить, что разность номеров соседних элементов в полученном наборе \mathcal{M} равна T_{\min} или $T_{\min} + 1$, откуда следует выполнение условия (1.4.2.1).

Для i -й вершины графа G через a_i обозначим число инцидентных ей рёбер разреза.

Тогда мощность разреза равна $t_0 = \sum_{i=1}^p a_i / 2$.

Вычислим значение функции $f_3(\mathcal{M}) = \sum_{n_m \in \mathcal{M}} \sum_{n_k \in \mathcal{M}} \| \mathbf{y}_{n_m} - \mathbf{y}_{n_k} \|^2$. Сначала заметим, что если $\mathbf{y}_{n_m} = 0$, то

$$\sum_{n_k \in \mathcal{M}} \| \mathbf{y}_{n_m} - \mathbf{y}_{n_k} \|^2 = \sum_{i=1}^p d_i = 2q,$$

так как квадрат расстояния от нулевого до основного вектора равен d_i , набор \mathcal{M} содержит номера векторов последовательности \mathcal{Y} , соответствующих всем вершинам графа G , а сумма степеней всех вершин графа равна удвоенному числу рёбер.

Если $\mathbf{y}_{n_m} = \mathbf{u}_i$ или $\mathbf{y}_{n_m} = \mathbf{z}_i$, то сумма $\sum_{n_k \in \mathcal{M}} \| \mathbf{y}_{n_m} - \mathbf{y}_{n_k} \|^2$ содержит $M = 2p - 1$ слагаемых, среди которых: 1) $p - 1$ слагаемых равных d_i (это квадраты расстояний до нулевых векторов), 2) одно слагаемое равное 0 (при $n_k = n_m$), 3) $p - 1 - d_i$ слагаемых вида (1.4.3.1) (это квадраты расстояний до векторов, соответствующих несмежным с v_i вершинам v_k), 4) $d_i - a_i$ слагаемых вида (1.4.3.2), 5) a_i слагаемых вида (1.4.3.3). Поэтому, используя несложные преобразования и очевидное равенство $\sum_{k=1}^p d_k = 2q$, найдем

$$\sum_{n_k \in \mathcal{M}} \| \mathbf{y}_{n_m} - \mathbf{y}_{n_k} \|^2 = (p - 1)d_i + \sum_{k=1, k \neq i}^p (d_i + d_k) - 2a_i + 2(d_i - a_i) = (2p - 1)d_i + 2q - 4a_i.$$

Суммируя найденные значения по всем номерам $n_m \in \mathcal{M}$, получим

$$\begin{aligned} f_3(\mathcal{M}) &= 2q(p - 1) + \sum_{i=1}^p ((2p - 1)d_i + 2q - 4a_i) \\ &= 2q(p - 1) + 2q(p - 1) + 2pq - 8t_0 = 8pq - 4q - 8t_0 \leq C \end{aligned}$$

при $t_0 \geq t$.

Таким образом, у последовательности \mathcal{Y} существует набор \mathcal{M} номеров, удовлетворяющий условиям (1.4.2.1) и $f_3(\mathcal{M}) \leq C$.

Достаточность. Пусть имеется набор \mathcal{M} номеров последовательности \mathcal{Y} , удовлетворяющих условиям (1.4.2.1) и $f_3(\mathcal{M}) \leq C$. Тогда из последнего неравенства следует, что набор \mathcal{M} не содержит номеров вспомогательных векторов.

Далее, заметим, что ни для какого $i = 1, 2, \dots, p$ набор \mathcal{M} не может содержать одновременно номера векторов \mathbf{u}_i и \mathbf{z}_i , так как $T_{\min} \geq 2$. Отсюда, а также из условий $M = 2p - 1$ и $T_{\max} \leq 2T_{\min} - 1$ следует, что набор \mathcal{M} содержит номера всех нулевых векторов и ровно по одному номеру векторов \mathbf{u}_i или \mathbf{z}_i для каждого i .

Рассмотрим разрез вершин графа G на два множества $V_1 = \{v_i \mid \mathbf{y}_{n_m} = \mathbf{u}_i, n_m \in \mathcal{M}\}$ и $V_2 = \{v_i \mid \mathbf{y}_{n_m} = \mathbf{z}_i, n_m \in \mathcal{M}\}$. Используя приведённые выше вычисления, нетрудно убедиться, что $f_3(\mathcal{M}) = 8pq - 4q - 8t_0$, где t_0 – мощность разреза. Поскольку $f_3(\mathcal{M}) \leq C$, этот разрез имеет мощность не менее t . Лемма 2 доказана.

Замечание 2. NP-полнота в сильном смысле рассмотренного случая задачи VSS3(T_{\min}, T_{\max}) следует из того, что числовые значения входных параметров этого случая задачи ограничены полиномом от размера входа задачи MaxCut.

Из лемм 1 и 2 следует

Теорема 1. Для любых натуральных T_{\min} и T_{\max} , удовлетворяющих неравенству $T_{\min} < T_{\max}$, задача VSS3(T_{\min}, T_{\max}) NP-полна в сильном смысле.

Из этой теоремы и формул (1.4.3.1), (1.4.3.2) вытекает

Следствие 1. Для любых натуральных T_{\min} и T_{\max} , удовлетворяющих неравенству $T_{\min} < T_{\max}$, задачи VSS1(T_{\min}, T_{\max}), VSS2(T_{\min}, T_{\max}) и MSSC-Case-S(T_{\min}, T_{\max}) NP-полны в сильном смысле.

1.4.4 Заключение

Показана NP-полнота оптимизационных задач, которые индуцируются проблемой поиска в последовательности векторов евклидова пространства, содержащей конечное число членов, такой подпоследовательности, что она имеет фиксированное число элементов и включает векторы близкие между собой по критерию минимума суммы квадратов расстояний. Из полученного результата следует труднорешаемость соответствующей проблемы анализа упорядоченных по времени табличных данных. Остается заметить, что эффективные алгоритмы с гарантированными оценками точности для решения рассмотренных задач в настоящее время неизвестны.

1.5. Разработка новых методов использования бент-функций в криптографии и теории хэш-функций

Представлен способ улучшения метода криптоанализа шифра KeeLoq, описанного А.Ю.Богдановым в статье [Bogdanov A. Cryptanalysis of the KeeLoq Block Cipher. // Cryptology ePrint Archive, Report 2007/055, 2007, <http://eprint.iarc.org/2007/055>]. В упомянутой статье для выбора слайдовой пары происходит перебор пар булевых векторов длины 32, для каждой из которых запускается корреляционный этап криптоанализа.

Усовершенствование заключается в сокращении перебора пар и их отсеивании после проверки некоторых вероятностных соотношений. Точнее, предлагается на каждом шаге корреляционного криптоанализа использовать найденные биты ключа для отсеивания неподходящих пар, а именно, можно найти вероятностное соотношение между битами из правильной слайдовой пары и битами ключа, проверка которого позволяет для части неправильных пар остановить выполнение корреляционного криптоанализа уже на этом шаге. В этом соотношении используется линейное приближение некоторой нелинейной функции, выполняющееся с вероятностью $5/8$.

На каждом шаге корреляционной атаки используется дополнительное соотношение для битов входного и выходного текста, что позволяет останавливать криптоанализ для пары, которая не удовлетворяет полученным соотношениям.

Благодаря данному улучшению происходит сокращение трудоёмкости криптоанализа в 64 раза.

SMS4 – стандарт блочного шифра КНР для защиты беспроводных сетей LAN WAPI (Wired Authentical and Privacy Infrastructure). Алгоритм шифрования, написанный на китайском языке, был опубликован правительством КНР в январе 2006 г., английский перевод опубликован в 2008 г. [W. Diffie, G. Ledin (translators). SMS4 encryption algorithm for wireless networks, Cryptology ePrint Archive, Report 2008/329, 2008, <http://eprint.iacr.org/2008/329>]. Существует не так много работ по криптоанализу SMS4. Заметим, что все существующие методы до сих пор остаются непрактическими, в связи с чем задача криптоанализа SMS4 остаётся актуальной.

В основу данной работы был положен метод линейного криптоанализа, описанный в [T. Kim, J. Kim, S. Hong, J. Sung. Linear and differential cryptanalysis of reduced SMS4 block cipher, Cryptology ePrint Archive, Report 2008/281, 2008, <http://eprint.iacr.org/2008/281>]. Но там используется не самое лучшее линейное приближение раунда. В данной работе описан линейный криптоанализ девяти раундов блочного шифра SMS4, для этого были исследованы всевозможные линейные приближения S-блока, линейные приближения

раунда, проверена оптимальность схемы согласования раундов, исследована связь между объёмом статистики и трудоёмкостью вычислений. Получена оценка на минимальную трудоёмкость: минимальная трудоёмкость составляет 2^{115} зашифрований и требует 2^{84} пар статистики.

1.6. Реализация метода квадратичного криптоанализа, использующего k -бент-функции для блочного шифра

Исследуется построение бент-функций на минимальном расстоянии от квадратичной бент-функции, описываются все такие бент-функции от $2k$ переменных и показывается, что их число равно $2^k (2^1 + 1) \dots (2^k + 1)$. Находится нижняя оценка числа бент-функций на минимальном расстоянии от бент-функции из класса Мэйорана – МакФарланда.

Бент-функции – булевы функции от чётного числа переменных, максимально удалённые от класса аффинных функций. Впервые бент-функции рассмотрены Ротхаусом. Бент-функции имеют большое число приложений в криптографии, теории кодирования и теории информации. Тем не менее, для них до сих пор существует много нерешённых проблем. Наиболее важная проблема – описание всех бент-функций. В частности, нахождение конструкций бент-функций.

В работе рассматривается построение бент-функций на минимальном расстоянии от квадратичной бент-функции. Ранее было показано, что две бент-функции от $2k$ переменных находятся на расстоянии 2^k (минимально возможное расстояние между двумя различными бент-функциями) тогда и только тогда, когда они отличаются на аффинном подпространстве размерности k и аффинны на нём. В данной работе описываются все бент-функции на минимальном расстоянии от квадратичной бент-функции (теорема 1), а также показывается, что число таких бент-функций от $2k$ переменных равно $2^k (2^1 + 1) \dots (2^k + 1)$ (теорема 2). Известно, что все квадратичные бент-функции аффинно эквивалентны функции $x_1 x_{k+1} \oplus x_2 x_{k+2} \oplus \dots \oplus x_k x_{2k}$, которая принадлежит классу Мэйорана – МакФарланда. Поэтому далее рассматриваем более общую задачу нахождения нижней оценки числа бент-функций на минимальном расстоянии от произвольной бент-функции из класса Мэйорана – МакФарланда (теорема 3). В заключении приводим некоторые факты и гипотезу об оценке числа бент-функций на расстоянии 2^k от произвольной бент-функции.

Введем некоторые обозначения.

Через Z_2^n обозначим n -мерное векторное пространство над Z_2 , через \oplus – сложение по модулю 2. Под расстоянием между двумя булевыми функциями будем понимать расстояние Хэмминга (число векторов, на которых функции различаются). Степень алгебраической нормальной формы булевой функции называется алгебраической степенью функции. Булева функция называется аффинной, если её алгебраическая степень не превосходит 1, и квадратичной, если её алгебраическая степень равна 2. Множество $L \subseteq Z_2^n$ называется

аффинным подпространством, если $L = a \oplus U$, где a – вектор из Z_2^n и U – линейное подпространство в Z_2^n . Для векторов u и v через $\langle u, v \rangle$ обозначим их скалярное произведение по модулю 2. Булева функция f от n переменных называется аффинной на множестве $D \subseteq Z_2^n$, если существуют $a \in Z_2^n$, $c \in Z_2$ такие, что для всех $x \in D$ выполняется $f(x) = \langle a, x \rangle \oplus c$. Булева функция f от $2k$ переменных называется бент-функцией, если все её коэффициенты Уолша – Адамара $W_f(v)$ равны $\pm 2^k$. Множество всех бент-функций от 2^k переменных обозначается через \mathcal{B}_{2k} . Булевы функции f и g от n переменных называются аффинно эквивалентными, если существует невырожденная матрица A размера $n \times n$, вектор b длины n и аффинная функция l от n переменных такие, что $g(x) = f(Ax \oplus b) \oplus l(x)$.

Пусть $D \subseteq Z^n$, через Ind_D обозначим индикатор множества D , т. е. булеву функцию от n переменных, которая принимает значение 1 только на элементах множества D . Через $a^{(i)}$ обозначим i -й столбец матрицы A , а через a_{ij} – элемент этой матрицы. Минимальное возможное расстояние между двумя различными бент-функциями от $2k$ переменных равно 2^k . Обозначим это расстояние через d_k .

В нашей работе задача построения бент-функций на расстоянии d_k от бент-функции f сводится к поиску аффинных подпространств размерности k в Z_2^{2k} , на которых заданная бент-функция аффинна.

Поскольку любая квадратичная бент-функция от $2k$ переменных аффинно эквивалентна бент-функции $f_0^{2k}(x) = x_1x_{k+1} \oplus x_2x_{k+2} \oplus \dots \oplus x_kx_{2k}$, мы построим все бент-функции на расстоянии d_k от нее и подсчитаем их число, тогда от остальных квадратичных бент-функций число бент-функций на расстоянии d_k будет таким же.

Для рассмотрения бент-функций на расстоянии d_k от функции f_0^{2k} сделано следующее: приведены некоторые утверждения об аффинности функций в общем и функции f_0^{2k} в частности на некотором подпространстве, рассмотрены удобные базисы для представления подпространств, описаны аффинные подпространства размерности k , на которых аффинна функция f_0^{2k} , подсчитано число таких подпространств и приведены некоторые примеры для малых размерностей. С помощью этой техники доказана

Теорема 2. Любая квадратичная бент-функция от $2k$ переменных имеет ровно $2^k (2^1 + 1) \dots (2^k + 1)$ бент-функций на расстоянии d_k .

Поскольку бент-функция f_0^{2k} принадлежит классу Мэйорана – МакФарланда, все квадратичные бент-функции аффинно эквивалентны бент-функциям из этого класса.

Поэтому рассмотрена более общая задача нахождения некоторой нижней оценки числа бент-функций на расстоянии d_k от функции из этого класса. Доказана следующая

Теорема 3. Пусть f – бент-функция от $2k$ переменных из класса Мэйорана – МакФарланда. Тогда число бент-функций на расстоянии d_k от неё не меньше $2^{2k+1} - 2^k$.

Доказано также, что число бент-функций на расстоянии d_k от функции $f \in \mathcal{B}_{2k}$ меньше 2^{k^2+2k} .

1.7. Представление равномерно упакованных кодов через гармонические функции на n -кубе в терминах преобразования Фурье

В терминах преобразования Фурье вводится понятие реконструктивного множества в булевом кубе. Получена характеристика реконструктивных множеств, являющихся линейными подпространствами. Установлены необходимые и достаточные условия реконструктивности сферы. Приведено достаточное условие реконструктивности двух концентрических сфер.

Исследуемый вопрос можно сформулировать так: какова минимальная информация (определённого рода) об объекте из заданного класса, однозначно определяющая этот объект? Этот вопрос рассматривается для класса всех действительнозначных функций, определённых на множестве всех двоичных наборов длины n . Произвольная такая функция однозначно определяется своими коэффициентами Фурье, которые, в свою очередь, однозначно определяются по функции.

Таким образом, функция полностью задаётся набором из 2^n значений либо набором из 2^n коэффициентов Фурье. Возникает вопрос об однозначности задания функции некоторым набором из k значений и m коэффициентов Фурье. Ясно, что в общем случае $k + m$ должно быть не меньше 2^n . Здесь исследуется экстремальный случай, когда $k + m = 2^n$ и в каждой вершине булева куба задано значение функции или коэффициент Фурье.

Исследуемый вопрос возник в результате изучения совершенных кодов и центрированных функций. Произвольная центрированная функция (т. е. функция, сумма значений которой в любом шаре радиуса 1 не зависит от выбора шара) может быть однозначно восстановлена по её значениям на наборах веса $(n + 1)/2$. Именно на этих наборах (и только на них) коэффициенты Фурье центрированной функции могут принимать ненулевые значения. В нашей терминологии это означает, что множество наборов веса $(n + 1)/2$ является реконструктивным. Совершенные коды являются важным частным случаем центрированных функций, когда функция булева, а сумма её значений в шаре равна 1.

Пополнен список реконструктивных множеств и изучены их свойства.

Булев куб E^n (т. е. n -мерное векторное пространство над $\{0, 1\}$) снабдим метрикой Хэмминга, тем самым расстояние $\rho(x, y)$ между вершинами x и y будет равно числу позиций, в которых эти вершины различаются. Вес Хэмминга $wt(x)$ вершины x равен числу ненулевых позиций x . Через W_h обозначим множество всех вершин веса h и назовём его h -м уровнем куба.

Известно, что произвольная функция $f: E^n \rightarrow R$ может быть однозначно представлена своими коэффициентами Фурье $\hat{f}(a)$, $a \in E^n$. Зафиксируем некоторый порядок вершин булева куба. Пусть F и \hat{F} – наборы длины 2^n , состоящие из всех значений функций f и \hat{f} соответственно, и пусть A – матрица преобразования Фурье, т. е. квадратная матрица порядка 2^n с элементами $a_{x,y} = (-1)^{\langle x,y \rangle}$, $x, y \in E^n$. Заметим, что A является матрицей Адамара и потому ортогональна. Тогда в векторной форме имеем $AF = \hat{F}$, $2^{-n}A\hat{F} = F$. Обозначим через A^{PQ} подматрицу из A , строки которой соответствуют вершинам множества $P \subseteq E^n$, а столбцы – вершинам из $Q \subseteq E^n$, через F^P – вектор всех значений функции f в вершинах множества P , а через \hat{F}^P – вектор всех коэффициентов Фурье функции f в вершинах P .

Для произвольного множества L из булева куба через \bar{L} обозначим дополнение к L : $\bar{L} = E^n \setminus L$, а через L^\perp – ортогональное к нему множество: $L^\perp = \{y \in E^n : \langle x, y \rangle = 0 \forall x \in L\}$. Если L является линейным подпространством, то $(L^\perp)^\perp = L$. В дальнейшем нам также понадобятся хорошо известные многочлены Кравчука $K_k(x; N)$ и Эберлейна $E_k(x; h, n)$.

Схема отношений Джонсона $J(h, n)$, $h \leq n/2$, определяется как множество W_h с отношениями R_0, R_1, \dots, R_h на нём, где $(x, y) \in R_i \Leftrightarrow \rho(x, y) = 2i$. Отношения R_0, R_1, \dots, R_h можно описывать квадратными матрицами инцидентности $D_0^h, D_1^h, \dots, D_h^h$ порядка $\binom{n}{h}$, строки и столбцы которых соответствуют вершинам множества W_h с элементами

$$(D_i^h) = \begin{cases} 1, & \text{если } \rho(x, y) = 2i, \\ 0 & \text{в остальных случаях.} \end{cases}$$

Матрицы инцидентности образуют базис алгебры Боуза – Меснера (БМ-алгебры) схемы $J(h, n)$. Все матрицы из БМ-алгебры имеют общие собственные подпространства. Можно также выбрать другой базис БМ-алгебры – из примитивных идемпотентов $J_0^h, J_1^h, \dots, J_h^h$.

Два базиса связаны соотношениями $D_i^h = \sum_{k=0}^h E_k(i; h, n) J_k^h$. Легко увидеть, что многочлены $E_k(i; h, n)$ дают собственные значения матриц D_k^h . Нетрудно доказать, что если некоторая матрица X из БМ-алгебры представима как линейная комбинация всех примитивных идемпотентов с ненулевыми коэффициентами, то обратная к ней X^{-1} представима в виде аналогичной линейной комбинации с обратными коэффициентами.

Пусть L – произвольное множество вершин из E^n . Множество L назовём реконструктивным, если для любой функции $f: E^n \rightarrow R$ из условий $f(x) = 0$, $x \in L$, и $\hat{f}(x) = 0$, $x \in \bar{L}$, следует, что функция f тождественно равна нулю. Очевидно, что тогда и $\hat{f}(x) = 0$ для всех $x \in E^n$. Другими словами, если для любой функции $f: E^n \rightarrow R$ её значения $f(x)$, $x \in L$, и

значения её коэффициентов Фурье $\hat{f}(x)$, $x \in \bar{L}$, однозначно определяют всю функцию, то множество L реконструктивно. Таким образом, множество \bar{L} реконструктивно в том и только том случае, если множество $L \subseteq E^n$ реконструктивно.

Выразим свойство множества L быть реконструктивным в терминах подматриц матрицы преобразования Фурье.

Теорема 1. Произвольное множество $L \subseteq E^n$ реконструктивно тогда и только тогда, когда матрица A^{LL} обратима.

Следующая лемма устанавливает инвариантность реконструктивности относительно сдвигов множеств.

Лемма 1. Пусть $a \in E^n$ – произвольная вершина и $L \subseteq E^n$ – произвольное множество. Тогда множество $L_a = L+a$ реконструктивно в том и только том случае, когда L реконструктивно.

Исследуем на реконструктивность некоторые классы множеств: линейные подпространства, сферы и пары концентрических сфер.

В случае, когда рассматриваемое подмножество линейно, его реконструктивность тесно связана с ортогональным множеством. Доказана

Теорема 2. Пусть L – линейное подпространство E^n . Тогда L реконструктивно в том и только том случае, когда $L \cap L^\perp = \{0\}$.

Поскольку при условии реконструктивности линейного подпространства L матрица A^{LL} квазиортогональна, найти обратную к ней очень просто: $(A^{LL})^{-1} = |L|^{-1} A^{LL}$. С ее помощью получена формула восстановления всех значений функции.

Зададимся вопросом, реконструктивна ли сфера радиуса h ? В силу леммы 1 без потери общности вместо произвольной сферы можно рассматривать сферу с центром в нулевой вершине, т. е. h -й уровень W_h гиперкуба. Таким образом, в рассматриваемом случае $L = W_h$ и порядок квадратной матрицы A^{LL} равен $\binom{n}{h}$. При этом естественно предположить, что сфера имеет радиус не более половины n .

В дальнейшем матрицу $A^{L_1 L_2}$ будем обозначать через A^{kh} , если L_1 и L_2 являются соответственно k -м и h -м уровнями n -куба; вместо F^{Wh} будем писать F^h . Через $\lambda_i(h, n)$, $i = 0, \dots, h$, обозначим собственные значения матрицы A^{hh} . Доказана

Лемма 2. Собственные числа матрицы A^{hh} задаются формулой

$$\lambda_i(h, n) = (-2^i) K_{h-i}(h-i, n-2i), i = 0, 1, \dots, h.$$

Комбинацией теоремы 1 и леммы 5 получается следующая

Теорема 3. Сфера радиуса $h \leq n/2$ в n -кубе реконструктивна тогда и только тогда, когда $K_i(i; n - 2h + 2i) \neq 0, i = 0, 1, \dots, h$.

Выведена также формула восстановления при условии реконструктивности. Эта формула выражает все неизвестные значения функции f , все ненулевые коэффициенты Фурье которой находятся на h -м уровне куба, через её значения на этом уровне. Формула восстановления имеет вид линейной комбинации сумм значений функции f , где сумма берётся по всем вершинам слоя W_h на фиксированном расстоянии от x (т. е. с фиксированным скалярным произведением на x).

В качестве примера нереконструктивности сферы отметим случай $h = n/2$, когда половина из требуемых теоремой значений многочленов Кравчука $K_i(i; 2i), i = 0, 1, \dots, n/2$, оказываются нулевыми (а именно, $K_i(i; 2i) = 0$, когда i нечётно), и потому сфера радиуса $n/2$ не обладает свойством реконструктивности.

Получены также условия реконструктивности множества, состоящего из двух сфер с общим центром. Учитывая лемму 1, опять без ограничения общности можно считать, что центр — это нулевая вершина куба, тогда сферы совпадают с уровнями куба и полагаем $L = W_k \cup W_h$, причём считаем, что $k < h \leq n/2$ (случай $k < n/2 < h$ может быть рассмотрен аналогично). Как и в случае одной сферы, рассматриваются нули некоторых многочленов. Достаточное условие реконструктивности множества L выражено опять в терминах полиномов Кравчука и Эберлейна.

1.8. Разбиение вершин плоского графа на подграфы малой степени и большого обхвата

Спектром гамильтонова цикла (кода Грея) в булевом n -мерном кубе называется набор $a = (a_1, \dots, a_n)$, где a_i – число рёбер i -го направления в цикле. Известны необходимые условия существования кода Грея со спектром a : числа a_i чётные и для любого $k = 1, \dots, n$ сумма k произвольных компонент набора a не меньше чем $2k$. Доказано существование такой размерности N , что если необходимые условия на спектр являются достаточными для существования гамильтонова цикла с таким спектром в булевом N -мерном кубе, то сформулированные выше условия являются достаточными и для всех размерностей n .

В 4-м томе «Искусства программирования» в разделе, посвящённом кодам Грея (гамильтоновым циклам в булевом n -кубе), Кнут указал на три не решённые на момент издания книги задачи. Первая из них состоит в оценке числа различных кодов Грея в булевом n -кубе. Известны порядок и асимптотика логарифма этого числа (при $n \rightarrow \infty$).

Вторая задача – каждое ли совершенное паросочетание в булевом n -кубе можно дополнить до гамильтонова цикла. Положительный ответ на этот вопрос получен Финком. В третьей задаче требовалось выяснить, являются ли сформулированные выше необходимые условия на спектр гамильтонова цикла достаточными для существования кода Грея с таким спектром. В данной работе предложено асимптотическое решение последней задачи. А именно, если необходимые условия являются достаточными в булевом n -кубе для некоторого достаточно большого n , то они являются достаточными для любого n . Отметим, что известно несколько способов построения кодов Грея с различными свойствами, в частности, разными авторами были построены гамильтоновы циклы с максимально равномерным (для фиксированной размерности) спектром.

Цель работы – доказательство того, что любой допустимый набор является спектром гамильтонова цикла в любом булевом n -кубе, если это верно для булева N -куба при некотором достаточно большом N . В доказательстве применяется конструкция гамильтонова цикла, использующая представление булева n -куба как декартова произведения кубов размерности k и $n - k$.

Булевым n -кубом называется множество Q_n двоичных слов длины n , а также граф GQ_n , вершинами которого являются элементы Q_n , и пара вершин соединена ребром, если и только если соответствующие слова различаются ровно в одной позиции.

Рассмотрим некоторый гамильтонов цикл в GQ_k , состоящий из рёбер непересекающихся совершенных паросочетаний P_1 и P_2 . Естественным образом вложим

паросочетание P_1 в GQ_n . Поскольку каждой вершине из GQ_k в декартовом произведении $GQ_k \times GQ_{n-k}$ соответствует булев $(n - k)$ -куб, каждому ребру из P_1 можно поставить в соответствие пару параллельных $(n - k)$ -кубов, т. е. один $(n - k + 1)$ -куб. Заменяем каждое ребро $v \in P_1$ гамильтоновым циклом H_v в $(n - k + 1)$ -кубе, проходящим через это ребро. Удалив P_1 из объединения P_2 и циклов H_v , $v \in P_1$, получим новый гамильтонов цикл в $GQ_k \times GQ_{n-k} = GQ_n$.

Если паросочетание со спектром (b'_1, \dots, b'_k) и хотя бы один из использованных в конструкции циклов в GQ_{n-k+1} имеют полный ранг, то в результате конструкции можно получить гамильтонов цикл полного ранга.

Основным результатом работы является следующая

Теорема. Существует такое число N , что если любой допустимый целочисленный набор длины N является спектром некоторого гамильтонова цикла (полного ранга в случае, когда $\sum_{i=1}^k a_i > 2^k$ при любом $k < N$), то для любого целого $n > 2$ любой допустимый целочисленный набор длины n является спектром некоторого гамильтонова цикла в GQ_n .

Для полного решения задачи о спектрах кодов Грея нужно обеспечить базу индукции для применения этой теоремы. Для построения гамильтоновых циклов в GQ_n при $n \leq N$ со всевозможными допустимыми спектрами можно использовать также и другие конструкции, в частности, конструкцию Бакоша.

Бесконечное вправо слово над алфавитом Σ – это слово вида $\omega = \omega_1\omega_2\omega_3\dots$, где каждое $\omega_i \in \Sigma$. Для слова ω определим $R\omega(i) = 0.\omega_i\omega_{i+1}\dots$. Отображение $h : \Sigma^* \rightarrow \Sigma^*$ называется морфизмом, если $h(xy) = h(x)h(y)$ для любых слов $x, y \in \Sigma^*$. Будем говорить, что ω – неподвижная точка морфизма φ , если $\varphi(\omega) = \omega$. Всякий морфизм однозначно определяется образами символов алфавита Σ , которые мы будем называть *блоками*. Морфизм называется равноблочным, если его блоки имеют одинаковую длину. Морфизм $\varphi : \Sigma^* \rightarrow \Sigma^*$ называется маркированным, если его блоки имеют вид $\varphi(a_i) = b_i x c_i$, где x – произвольное слово, а b_i и c_i – символы алфавита Σ , причем все b_i (как и все c_i) различны. В дальнейшем мы будем рассматривать только маркированные равноблочные морфизмы с длиной блоков l . Отметим, что для всех неподвижных точек $\varphi(\omega) = \omega$ рассматриваемых нами морфизмов существует число L_ω такое, что любое подслово слова ω длины не менее L_ω однозначно разбивается на блоки.

Определим функцию $\gamma : \mathbb{R}^2 \setminus \{(a, a) \mid a \in \mathbb{R}\} \rightarrow \{<, >\}$, которая двум различным действительным числам ставит в соответствие их отношение.

Равноблочный морфизм $\varphi : \{0, 1\}^* \rightarrow \{0, 1\}^*$ будем называть сравнимым, если его неподвижная точка $\omega = \varphi(\omega)$ удовлетворяет следующему условию: пусть $\omega_i = \omega_j$, где

$i \equiv i' \pmod{l}$, $j \equiv j' \pmod{l}$ и $0 \leq i', j' \leq l-1$, причем i' и j' фиксированы. Если $i' \neq j'$ или если ω_i и ω_j лежат в блоках разного типа в правильном разбиении ω , то отношение $\gamma(R_\omega(i), R_\omega(j))$ определено однозначно.

В следующих трех утверждениях мы находим условие, по которому можно определить является ли маркированный морфизм сравнимым.

Утверждение 1. Пусть ω – неподвижная точка маркированного равноблочного морфизма φ , причем $\varphi(0) = A$, $\varphi(1) = B$. Тогда верны следующие утверждения:

- 1) если $0u1$ является подсловом A или B , причем $A = 0u0x$, где x – некоторое слово, то φ не является сравнимым морфизмом;
- 2) если $1u0$ является подсловом A или B , причем $B = 1u1x$, где x – некоторое слово, то φ не является сравнимым морфизмом.

Утверждение 2. Пусть ω – неподвижная точка маркированного равноблочного морфизма φ , причем $\varphi(0) = A$, $\varphi(1) = B$. Тогда верны следующие утверждения:

- 1) если $0u$ является суффиксом A или B , причем $A = 0u0x$, где x – некоторое слово, то φ не является сравнимым морфизмом;
- 2) если $1u$ является суффиксом A или B , причем $B = 1u1x$, где x – некоторое слово, то φ не является сравнимым морфизмом.

Утверждение 3. Пусть ω – неподвижная точка маркированного равноблочного морфизма φ , для которого не выполнены условия утверждений 1 и 2. Тогда φ – сравнимый морфизм.

Пусть ω – бесконечное вправо непериодическое слово над алфавитом Σ . Тогда определим бесконечную перестановку, порождаемую словом ω , как упорядоченную тройку $\delta = \langle N, <\delta, < \rangle$, где $<\delta$ и $<$ – линейные порядки на N . При этом $<\delta$ определяется следующим образом: $i <\delta j$ тогда и только тогда, когда $R_\omega(i) < R_\omega(j)$. Определим комбинаторную сложность $\lambda(n) = |\text{Perm}(n)|$ перестановки δ_ω , порождаемой некоторым словом ω как число различных её подперестановок. Понятие бесконечной перестановки было введено в [Fon-Der-Flaass D.-G., Frid A.E. On periodicity and low complexity of infinite permutations // European J. Combin. 2007. Vol. 28, N 8. Pp. 2106-2114], где, кроме того, исследовались свойства периодичности и низкая комбинаторная сложность перестановок. Понятие перестановки, порожденной бесконечным непериодическим словом, было введено Макаровым в [Makarov M. A. On permutations generated by infinite binary words // Sib. Elektron. Mat. Izv. 2006. Vol. 3. Pp. 304-311]. В работе [Makarov M.A On the permutations generated by the Sturmian words // Sib. Math. J. 2009. Vol. 50, N 3. Pp. 674-680] тот же автор вычислил комбинаторную сложность перестановок, порожденных хорошо известным семейством слов Штурма. В работе [Widmer S. Permutation complexity of the Thue-Morse word // Adv. in Appl. Math. 2010]

Уидмер вычислил комбинаторную сложность перестановки Туэ-Морса. В настоящей работе найдена комбинаторная сложность перестановок, порожденных неподвижными точками сравнимых морфизмов.

Определим функцию $\delta(n, z)$: если $n = l^s/z + 1$ для некоторого натурального s , то $\delta(n, z) = 1$, иначе $\delta(n, z) = 0$.

Теорема 1. Пусть ω – неподвижная точка сравнимого морфизма ϕ . Тогда комбинаторная сложность перестановки порожденной ω вычисляется следующим образом:

$$\lambda(n) = \sum_{a_1 \in A_1} [C_{a_1}^{nar}(n)(m_{a_1} + n_{a_1}) + (C_{a_1}^{bad}(n) + C_{a_1}^{wide}(n))(m_{a_1} + 2n_{a_1})] + \sum_{a_2 \in A_2} C_{a_2}(n)m_{a_2} - \sum_{z \in \mathbb{Z}} [S_z(n-1)(k_z + t_z + r_z)(1 - \delta(n, z)) + (k_z + r_z) \delta(n, z)] \text{ для } n \geq L_\omega.$$

Все функции, входящие в условие теоремы, определяются в работе [Valuzhenich A. Permutation complexity of the fixed points of some uniform binary morphisms // EPTCS 63 (2011), Proceedings of WORDS 2011. Pp. 257-264].

Установлено, что сложность реализации в классе обобщённых (троичных) π -схем троичного счётчика кратности 3, зависящего от трёх переменных, равна 18.

Под q -ичной параллельно-последовательной контактной схемой (π -схемой) понимается обычная (двоичная) π -схема, контактам которой приписаны символы x_i^δ , $i = 1, \dots, n$; $\delta = 0, 1, \dots, q-1$. При этом символ x_i^δ не булева переменная или её отрицание, а функция от x_i , определённая на $B_q = \{0, 1, \dots, q-1\}$ и принимающая значения из $\{0, 1\}$. Значение функции x_i^δ равно 1, если $x_i = \delta$, и равно 0, если $x_i \neq \delta$. Для определённых таким образом переменных естественно ввести операции дизъюнкции и конъюнкции. Поэтому любой обобщённой π -схеме будет соответствовать формула, сложность которой определяется числом вхождений в неё переменных.

Функция $f: B_q^n \rightarrow \{0, 1\}$ проводимости q -ичной π -схемы определяется по аналогии с двоичным случаем: по определению q -ичная π -схема реализует функцию $f(x_1, \dots, x_n) = V_C K_C$, где дизъюнкция берётся по всем простым (без самопересечений) цепям, соединяющим полюсы схемы, а K_C – это конъюнкция всех функций $x_{i1}^{\delta 1}, \dots, x_{ik}^{\delta k}$, приписанных контактам цепи C . Как и в двоичном случае, мы говорим, что контакт, помеченный x_i^δ , замкнут на наборе $(a_1, \dots, a_n) \in B_q^n$, если $a_i = \delta$, и разомкнут в противном случае.

Сложностью $L(S)$ q -ичной π -схемы S называется число контактов в S . Сложностью $L_\pi(f)$ функции $f: B_q^n \rightarrow \{0, 1\}$ в классе π -схем называется $\min_S L(S)$, где минимум берётся по всем q -ичным π -схемам, реализующим f . На множестве B_q^n определим следующую функцию (линейную функцию, существенно зависящую от всех своих переменных):

$$\varphi_q(x_1, \dots, x_n) = \begin{cases} 1, & \text{если } x_1 + \dots + x_n = 0 \pmod{q}, \\ 0 & \text{в противном случае.} \end{cases}$$

Результатом настоящей работы является точное значение сложности функции $\varphi_3(x_1, x_2, x_3)$.

Теорема 1. Для функции $\varphi_3(x_1, x_2, x_3)$ справедливо равенство $L_\pi(\varphi_3(x_1, x_2, x_3)) = 18$.

Приведен пример троичной π -схемы, которая имеет сложность 18 и реализует функцию $\varphi_3(x_1, x_2, x_3)$. Тем самым доказана справедливость неравенства $L_\pi(\varphi_3(x_1, x_2, x_3)) \leq 18$. Доказательство неравенства $L_\pi(\varphi_3(x_1, x_2, x_3)) \geq 18$ опирается на подход Храпченко В. М. к получению нижних оценок сложности π -схем, основная идея которого сформулирована в лемме 1.

Пусть $f(x_1, \dots, x_n)$ – произвольная функция, заданная на B_q^n , значения которой принадлежат $\{0, 1\}$,

$$N_0(f(x_1, \dots, x_n)) = \{(\alpha_1, \dots, \alpha_n) \in B_q^n \mid f(\alpha_1, \dots, \alpha_n) = 0\},$$

$$N_1(f(x_1, \dots, x_n)) = \{(\beta_1, \dots, \beta_n) \in B_q^n \mid f(\beta_1, \dots, \beta_n) = 1\}.$$

Для произвольной π -схемы S множество её контактов будем обозначать через $K(S)$.

Лемма 1. Для произвольных непостоянной функции $f: B_q^n \rightarrow \{0, 1\}$ и реализующей её q -ичной π -схемы S существует отображение

$H: N_0(f(x_1, \dots, x_n)) \times N_1(f(x_1, \dots, x_n)) \rightarrow K(S)$, удовлетворяющее условию: если произвольный контакт $k \in K(S)$ помечен символом $x_i \delta$, то $H^{-1}(k) = A \times B$, где A – подмножество $N_0(x_i \delta \vee f(x_1, \dots, x_n))$, B – подмножество $N_1(x_i \delta \vee f(x_1, \dots, x_n))$.

Доказательство состоит в построении такого отображения. Пусть S – произвольная q -ичная π -схема, реализующая функцию $f(x_1, \dots, x_n)$.

Цепью π -схемы S называется простая цепь в S , соединяющая полюсы схемы. Тупиковым сечением π -схемы S называется минимальное по включению множество контактов этой схемы, имеющее общий контакт с каждой цепью схемы. Индукцией по сложности π -схемы нетрудно доказать, что каждая цепь и каждое тупиковое сечение π -схемы S имеют ровно один общий контакт.

Отображение $H: N_0(f(x_1, \dots, x_n)) \times N_1(f(x_1, \dots, x_n)) \rightarrow K(S)$ определим следующим образом. Заметим, что для каждого набора $(\alpha_1, \dots, \alpha_n) \in N_0(f(x_1, \dots, x_n))$ найдётся такое тупиковое сечение π -схемы S , все контакты которого разомкнуты на этом наборе. Поставим это сечение в соответствие с $(\alpha_1, \dots, \alpha_n)$. Точно так же для каждого набора $(\beta_1, \dots, \beta_n) \in N_1(f(x_1, \dots, x_n))$ найдётся такая цепь π -схемы S , все контакты которой замкнуты на этом наборе. Поставим эту цепь в соответствие с $(\beta_1, \dots, \beta_n)$.

По определению отображение $H: N_0(f(x_1, \dots, x_n)) \times N_1(f(x_1, \dots, x_n)) \rightarrow K(S)$ каждой паре наборов $((\alpha_1, \dots, \alpha_n), (\beta_1, \dots, \beta_n)) \in N_0(f(x_1, \dots, x_n)) \times N_1(f(x_1, \dots, x_n))$ ставит в соответствие тот единственный контакт π -схемы S , который является общим для тупикового сечения, соответствующего $(\alpha_1, \dots, \alpha_n)$, и цепи, соответствующей $(\beta_1, \dots, \beta_n)$.

Можно показать, что отображение H удовлетворяет условию леммы.

Пусть S – произвольная π -схема, реализующая функцию $\varphi_3(x_1, x_2, x_3)$. Докажем, что $L(S) \geq 18$. Для этого рассмотрим отображение $H : N_0(\varphi_3(x_1, x_2, x_3)) \times N_1(\varphi_3(x_1, x_2, x_3)) \rightarrow K(S)$, существование которого утверждает лемма 1.

Пусть

$$A_1 = \{\alpha = (\alpha_1, \alpha_2, \alpha_3) \in B_3^3 \mid \alpha_1 + \alpha_2 + \alpha_3 = 1 \pmod{3}\},$$

$$A_0 = \{\beta = (\beta_1, \beta_2, \beta_3) \in B_3^3 \mid \beta_1 + \beta_2 + \beta_3 = 0 \pmod{3}\}.$$

Отметим, что $A_1 \subseteq N_0(\varphi_3(x_1, x_2, x_3))$, $A_0 = N_1(\varphi_3(x_1, x_2, x_3))$.

Множество B_3^3 будем рассматривать как векторное пространство над полем $GF(3)$.

Введём следующие обозначения:

$$\bar{e}^1 = (1, 0, 0), \quad \bar{e}^2 = (0, 1, 0), \quad \bar{e}^3 = (0, 0, 1);$$

$$R_i = \{(\bar{\alpha}, \bar{\beta}) \in A_1 \times A_0 \mid \bar{\alpha} - \bar{\beta} = \bar{e}^i\}, \quad i = 1, 2, 3;$$

$$R_i^\delta = \{(\bar{\alpha}, \bar{\beta}) \in R_i \mid \beta_i = \delta\}, \quad i = 1, 2, 3, \quad \delta = 0, 1, 2;$$

$$R = R_1 \cup R_2 \cup R_3.$$

Рассмотрим также отображение $h : R \rightarrow K(S)$, являющееся сужением отображения H на множество R .

Заметим, что если контакт $k \in K(S)$ помечен x_i^δ , $i \in \{1, 2, 3\}$, $\delta \in \{0, 1, 2\}$,

то $h^{-1}(k) \subseteq R_i^\delta$. Это следует из того, что в силу леммы 1

$$h^{-1}(k) \subseteq H^{-1}(k) \subseteq N_0(x_i^\delta \vee \varphi_3(x_1, x_2, x_3)) \times N_1(x_i^\delta \wedge \varphi_3(x_1, x_2, x_3)).$$

Поэтому $|h^{-1}(k)| \leq |R_i^\delta| = 3$.

Пусть $y_i = |\{k \in K(S) \mid |h^{-1}(k)| = i\}|$, $i = 1, 2, 3$.

Неравенство $L(S) \geq 18$ является следствием очевидного неравенства $L(S) \geq y_1 + y_2$ и двух следующих соотношений: $y_1 + 2y_2 + 3y_3 = 27$, $y_2 + 3y_3 \leq 9$.

Равенство $y_1 + 2y_2 + 3y_3 = 27$ вытекает из очевидных равенств

$$y_1 + 2y_2 + 3y_3 = |R|, \quad |R| = 27.$$

Докажем, что $y_2 + 3y_3 \leq 9$. Для каждого $i \in Z$ определим вектор $\bar{e}^i \in B_3^3$, отображение

$C_i : A_1 \times A_0 \rightarrow A_1 \times A_0$ и множество $D_i \subseteq A_1 \times A_0$ по следующим правилам:

$$\bar{e}^i = \bar{e}^1, \text{ если } i \equiv 1 \pmod{3}, \quad \bar{e}^i = \bar{e}^2, \text{ если } i \equiv 2 \pmod{3}, \quad \bar{e}^i = \bar{e}^3, \text{ если } i \equiv 3 \pmod{3};$$

$$C_i((\bar{\alpha}, \bar{\beta})) = (\bar{\beta} + \bar{e}^i, \bar{\alpha} - \bar{e}^i), \quad (\bar{\alpha}, \bar{\beta}) \in A_1 \times A_0;$$

$$D_i = \{(\bar{\alpha}, \bar{\beta}) \in A_1 \times A_0 \mid \bar{\alpha} - \bar{\beta} = 2\bar{e}^i + \bar{e}^{i+1} + \bar{e}^{i+2}\}.$$

Отметим, что если $i \equiv j \pmod{3}$, то $C_i \equiv C_j$, $D_i = D_j$. Заметим еще, что $|D_1| = |D_2| = |D_3| = 9$, множества D_1, D_2, D_3 попарно не пересекаются и для любого $d \in D_1$ имеют место включения $C_3(d) \in D_2$, $C_2(d) \in D_3$.

Определим множество T , элементами которого являются следующие девять трёхэлементных подмножеств множества $D_1 \cup D_2 \cup D_3$: $T = \{\{d, C_3(d), C_2(d)\} \mid d \in D_1\}$.

Множество $M \subseteq A_1 \times A_0$ будем называть независимым, если для любых двух пар $(\bar{\alpha}, \bar{\beta}), (\bar{\gamma}, \bar{\lambda}) \in M$ из неравенства $(\bar{\alpha}, \bar{\beta}) \neq (\bar{\gamma}, \bar{\lambda})$ следует, что $\bar{\alpha} \neq \bar{\gamma}$ и $\bar{\beta} \neq \bar{\lambda}$.

Для $M \subseteq A_1 \times A_0$ обозначим через

$$\hat{M} = \{\bar{\alpha} \in A_1 \mid \exists \bar{\alpha}' \in A_0 (\bar{\alpha}, \bar{\alpha}') \in M\} \times \{\bar{\beta} \in A_0 \mid \exists \bar{\beta}' \in A_1 (\bar{\beta}', \bar{\beta}) \in M\}.$$

Множество \hat{M} будем называть замыканием множества M .

Заметим, что каждое множество R_i^δ (а также каждое его подмножество), $i = 1, 2, 3$, $\delta = 0, 1, 2$, является независимым. Кроме того, для любого независимого множества M справедливо равенство $|\hat{M}| = |M|^2$.

Каждому контакту $k \in K(S)$ поставим в соответствие подмножество $T(k) = \{t \in T \mid \hat{h}^{-1}(k) \cap t \neq \emptyset\}$ множества T . Покажем, что система подмножеств $T(k)$, $k \in K(S)$, множества T обладает свойствами, сформулированными в леммах 2, 3.

Лемма 2. Множества $T(k)$, $k \in K(S)$, попарно не пересекаются.

Лемма 3. Для каждого $k \in K(S)$ справедливо равенство $|T(k)| = (|h^{-1}(k)|^2 - |h^{-1}(k)|) / 2$.

Следующие соотношения являются очевидными следствиями лемм 2 и 3:

$$\sum_{k \in K(S)} |T(k)| \leq |T|,$$

$$|T(k)| = \begin{cases} 0, & \text{если } |h^{-1}(k)| = 1, \\ 0, & \text{если } |h^{-1}(k)| = 2, \\ 0, & \text{если } |h^{-1}(k)| = 3, \end{cases} \quad k \in K(S).$$

Неравенство $y_2 + 3y_3 \leq 9$ непосредственно вытекает из этих соотношений. Тем самым теорема 1 доказана. Доказательства лемм 2 и 3 трудоемки и здесь не приводятся.

Понятие кратного совершенного кода является естественным обобщением совершенного кода, одного из центральных объектов теории кодирования. Подмножество вершин графа называется k -кратным совершенным кодом радиуса r , если для каждой вершины шар радиуса r с центром в этой вершине содержит в точности k кодовых вершин. В общем виде задача формулируется следующим образом: определить параметры k , r , n и q такие, что существуют k -кратные совершенные коды радиуса r в n -кубе над $GF(q)$. На сегодняшний день эта задача решена только для $k = 1$: В.А. Зиновьев, В.К. Леонтьев и Тьетвайнен показали, что все возможные параметры исчерпываются списком $n = 2^k - 1$, $r = 1$ и $n = 23$, $r = 3$ над $GF(2)$, $n = 11$, $r = 2$ над $GF(3)$. Раскраска вершин графа называется совершенной, если для каждой вершины цветовой набор её соседей зависит только от её цвета.

В работе исследуется связь совершенных 2-раскрасок и k -кратных совершенных кодов двоичного гиперкуба. Цель данного исследования – дать описание кратных совершенных кодов, которые одновременно являются совершенными раскрасками в 2 цвета. Основным результатом является критерий, который по параметрам совершенной 2-раскраски определяет, является ли она k -кратным совершенным кодом.

Пусть H_n – это гиперкуб размерности n . Вершины куба – двоичные наборы длины n , они смежны, если их наборы отличаются ровно в одной координате. Весом $wt(y)$ вершины $y \in H_n$ называется число единиц её набора. Расстояние Хэмминга $d(x, y)$ между вершинами $x, y \in H_n$ – это число позиций, в которых x и y различны. Будем называть сферой радиуса r с центром в точке x множество $S(x, r) = \{y \in H_n \mid d(x, y) = r\}$, а шаром радиуса r с центром в точке x множество $B(x, r) = \{y \in H_n \mid d(x, y) \leq r\}$.

Полиномом Кравчука степени r называется полином $Pr(x, n) = \sum_{i=0}^r (-1)^i \binom{x}{i} \binom{n-x}{r-i}$.

Отображение $T : H_n \rightarrow \{1, 2, \dots, k\}$ называется совершенной раскраской вершин куба в k цветов с матрицей параметров $(s_{ij})_{i,j \in \{1, \dots, k\}}$, если оно сюръективно и для каждой i, j у любой вершины цвета i число соседей цвета j равно s_{ij} .

Раскраска вершин куба в 2 цвета называется совершенной с матрицей параметров $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, если каждая вершина первого цвета имеет a соседей первого цвета и b соседей второго цвета, а каждая вершина второго цвета имеет c соседей первого цвета и d соседей второго. Не теряя общности, будем считать, что $b \geq c$.

Подмножество вершин графа называется k -кратным совершенным кодом радиуса r , если для каждой вершины шар радиуса r с центром в этой вершине содержит в точности k кодовых вершин. Таким образом, необходимое условие существования: $\frac{2^n k}{\sum_{i=0}^r \binom{n}{i}}$ должно быть целым. В случае $k = 1$ мы получаем классическое определение совершенного кода. Как уже упоминалось выше, задача перечисления всех параметров n, r , при которых такие коды существуют, решена. При произвольном k эта проблема ещё далека от решения.

В соответствии с введенными определениями ставится задача: найти все n, b, c такие, что соответствующая совершенная раскраска будет совершенным кодом некоторой кратности k . В данной работе получен критерий для параметров n, b, c , который решает эту задачу.

Теорема 1. Совершенная раскраска с параметрами $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ является кратным совершенным кодом радиуса r тогда и только тогда, когда $P_r(\frac{b+c}{2} - 1, n - 1) = 0$, при этом кратность кода $k = \frac{c}{b+c} \sum_{i=0}^r \binom{n}{i}$.

Рассмотрим некоторые частные случаи.

При $r = 1$ критерий выглядит следующим образом: $c = a + 1$. Таким образом, параметрами совершенных раскрасок, являющихся также совершенными кратными кодами, будут $\binom{c-1}{c} \binom{b}{d-1}$ с кратностью $k = c$. Известно, что такие совершенные раскраски существуют для любых допустимых b, c . Заметим, что такие совершенные раскраски будут кратными совершенными кодами любого нечётного радиуса, так как в силу определения полиномов Кравчука $\Pr(\frac{n-1}{2}, n-1) = 0$ при нечётных n . Таким образом, мы заключаем, что $\forall m, l, r \in \mathbb{N} : r \equiv 1 \pmod{2}$ существуют кратные совершенные коды радиуса 1 при $n = 2m+1$ кратности $k = i \cdot 1$ для всех $i \in \{1, \dots, 2m\}, i \equiv 1 \pmod{2}$. Эти же коды будут кратными совершенными кодами любого нечётного радиуса.

Рассмотрены критерии также для случаев $r = 2$ и $r = 3$. Приведены все совершенные раскраски, являющиеся кратными совершенными кодами радиуса 2 в 10-мерном кубе и соответствующие им кратности.

На сегодняшний день существуют конструкции, позволяющие строить большое число различных совершенных раскрасок с различными параметрами. Приведённые выше результаты позволяют искать среди них неизвестные ранее кратные совершенные коды. Описанные в работе определения и задачи легко переносятся и на другие классы графов, такие как кубические транзитивные, циркулярные и графы Джонсона.

Граф $G = (V, E)$ называется сильно регулярным с параметрами (v, k, λ, μ) , если $|V| = v$, граф G является связным регулярным степени k , любая смежная пара вершин имеет λ общих соседей, и любая пара несмежных вершин имеет μ общих соседей.

Д. Г. Фон-Дер-Флаассом было введено определение сильно регулярной системы троек. Пусть V – конечное множество, T – подмножество множества трехэлементных подмножеств V . T называется сильно регулярной системой троек с параметрами $\Lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ и $M = (\mu_0, \mu_1, \mu_2, \mu_3)$, если для любых трех различных вершин u, v, w верно следующее:

1. Если $\{u, v, w\} \in T$, то

- (a) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\}, \{u, v, x\} \notin T\}| = \lambda_0,$
- (b) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\} \notin T, \{u, v, x\} \in T\}| = \lambda_1,$
- (c) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\} \notin T, \{u, x, w\}, \{u, v, x\} \in T\}| = \lambda_2,$
- (d) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\}, \{u, v, x\} \in T\}| = \lambda_3.$

2. Если $\{u, v, w\} \notin T$, то

- (a) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\}, \{u, v, x\} \notin T\}| = \mu_0,$
- (b) $|\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\} \notin T, \{u, v, x\} \in T\}| = \mu_1,$

$$(c) |\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\} \notin T, \{u, x, w\}, \{u, v, x\} \in T\}| = \mu_2,$$

$$(d) |\{x \in V \setminus \{u, v, w\} \mid \{x, v, w\}, \{u, x, w\}, \{u, v, x\} \in T\}| = \mu_3.$$

Для любой системы троек T и любой ее вершины u определен граф G на $V \setminus \{u\}$: его ребрами являются те пары $\{v, w\}$, для которых $\{u, v, w\} \in T$. Будем говорить, что граф G *индуцирован* системой T и вершиной u . Несложно доказать, что индуцированные графы сильно регулярной системы троек являются сильно регулярными, при этом относительно любой вершины системы троек индуцируется сильно регулярный граф с одинаковым набором параметров (v, k, λ, μ) .

Сильно регулярная система троек T называется сильно регулярным расширением графа G , если в результате этой операции относительно любой вершины системы троек получается граф, изоморфный G . В работе исследуется существование сильно регулярных расширений сильно регулярных графов на небольшом количестве вершин. Основным результатом работы является

Теорема. Существуют сильно регулярные расширения графа Петерсена с параметрами $\Lambda = (2, 2, 0, 0)$ и $M = (2, 1, 1, 0)$ и графа решетки 3×3 с параметрами $\Lambda = (0, 2, 0, 1)$ и $M = (1, 0, 2, 0)$. Не существует сильно регулярных расширений графов Шрикхандэ, Пэли порядка 13 и $\overline{L(K6)}$.

Рассмотрим булев куб $E^n = \{0, 1\}^n$ с расстоянием Хэмминга, заданным на его вершинах: $d(x, y) = |\{i \mid x_i \neq y_i\}|$. *Двоичным кодом* длины n называется любое подмножество такого пространства. Минимальное расстояние между различными вершинами кода определяет его *кодовое расстояние*. Два кода *эквивалентны*, если существует изометрия булева куба, отображающая один код в другой.

Интересен вопрос о том, какие условия, параметры или части кодов позволяют восстанавливать код однозначно или, например, с точностью до эквивалентности. Один из естественных подходов к восстановлению кодов заключается в использовании их метрических свойств. Ситуация выглядит нетривиальной, поскольку наличие изометрии между кодами не гарантирует их задания в объемлющем пространстве даже с точностью до эквивалентности. Примером служат коды Адамара: хотя все эти коды изометричны друг другу, среди них имеются неэквивалентные. Здесь вопрос о восстановлении выступает как проблема метрической жёсткости кодов, которая решена для некоторых классов кодов, см., например, [Августинович С. В., Соловьёва Ф. И. К метрической жесткости двоичных кодов // Пробл. передачи информ. – 2003. – Т. 39, вып. 2. – С. 23–28.]. Напомним, что код называется метрически жёстким, если любая изометрия, определённая на вершинах кода, продолжается до изометрии всего пространства.

В качестве инвариантов для восстановления кода с некоторой степенью точности разными авторами рассматривались также: граф минимальных расстояний между кодовыми вершинами; множество вершин на заданном расстоянии p друг от друга; треугольники, построенные на кодовых вершинах, и соответствующие им тройки попарных расстояний.

Одним из наиболее сильных инвариантов оказался набор размерностей подкодов двоичного кода, где под размерностью $\text{Dim}(C)$ кода C понимается размерность минимальной грани булева куба, содержащей код [Августинович С. В. О сильной изометрии бинарных кодов // Дискретн. анализ и исслед. операций. Сер. 1. – 2000. – Т. 7, № 3. – С. 3–5]. Впоследствии удалось уточнить, что для восстановления двоичного кода с точностью до эквивалентности достаточно знать размерности его подкодов лишь чётной мощности. Точнее, эти размерности позволяют найти размерности всех остальных подкодов [Горкунов Е. В., Августинович С. В. О восстановлении двоичных кодов по размерностям их подкодов // Дискретн. анализ и исслед. операций. – 2010. – Т. 17, № 5. – С. 15–21].

Различные результаты по восстановлению кодов, были получены при изучении совершенных двоичных кодов. Для них найдено подмножество кодовых слов, зная которое, можно однозначно восстановить весь код: произвольный совершенный двоичный код длины n с кодовым расстоянием 3 единственным образом определяется расположением своих кодовых слов веса $(n - 1)/2$.

Пусть C – приведенный двоичный код мощности k длины n . Упорядочим векторы из $\{0\} \times E^{k-1}$ некоторым образом и обозначим через $\alpha_0, \dots, \alpha_{2^{k-1}-1}$ количество столбцов соответствующего вида в проверочной матрице кода C с нулевой верхней строкой. Зная размерности подкодов четной мощности кода C , можно составить $C_k^2 + C_k^4 + \dots = 2^{k-1} - 1$ уравнений, связывающих величины $\alpha_1, \dots, \alpha_{2^{k-1}-1}$. В дополнение имеем равенство для числа нулевых столбцов $\alpha_0 = n - \text{Dim}(C)$. В результате получаем квадратную систему линейных уравнений для α_i . Результаты вышеупомянутой статьи показывают, что эта система имеет единственное решение, и тем самым невырожденна.

Вопрос о минимальности набора размерностей подкодов четной мощности в терминах построенной системы формулируется следующим образом. Будет ли система при удалении одного уравнения всякий раз иметь два или более целочисленных решения? Ответом на этот вопрос служит

Теорема. Размерности подкодов четной мощности двоичного кода образуют минимальный набор размерностей, определяющий код с точностью до эквивалентности.

Одним из объектов, близких к кодам, для которого исследование восстановимости весьма актуально, является граф. Проблема изоморфизма графов является ориентиром в этом направлении. Колодой графа G называется мультимножество $\{G - v \mid v \in V(G)\}$, где $G - v$

– подграф графа G , полученный удалением вершины v и инцидентных ей рёбер. Известная гипотеза Улама о том, что колода графа определяет его с точностью до изоморфизма, была ранее доказана для отдельных классов графов. Аналогичная гипотеза о рёберной восстановимости графа также является открытой.

Рассмотрим регулярный граф $G = (V, E)$. Для всякого кода (т.е. подмножества вершин) C определим дистанционное разбиение множества вершин этого графа относительно C : $V = \bigcup_{i=0, \dots, p} C_i$, где $p = \max\{d(x, C) : x \in V\}$ является радиусом покрытия кода C . Код C в G называется полностью регулярным, если любая вершина из слоя C_j имеет $\gamma_j, \alpha_j, \beta_j$ соседей соответственно из слоев C_{j-1}, C_j и C_{j+1} . Набор чисел $\{\gamma_j, \alpha_j, \beta_j : j = 0, \dots, p\}$ называется массивом пересечения кода C .

Вершинами графа Джонсона $J(n, w)$ являются все w -элементные подмножества n -элементного множества; две вершины смежны, если их пересечение имеет мощность $w - 1$.

Совокупность w -элементных подмножеств n -элементного множества, именуемых блоками, называется $t - (n, w, \lambda)$ -схемой, если любое t -элементное подмножество содержится в точности в λ блоках. Мы рассматриваем лишь схемы, у которых все блоки различны. Блоки такой схемы можно рассматривать как вершины графа Джонсона. Силой схемы называется максимальное t , для которого она является $t - (n, w, \lambda)$ -схемой.

П. Дельсартом было установлено, что всякий полностью регулярный код в графе Джонсона является $t - (n, w, \lambda)$ -схемой. Более того, в случае $p = 1$ числа t, λ однозначно определяются массивом пересечения полностью регулярного кода: t . Таким образом, полностью регулярные коды в графах Джонсона можно рассматривать как подкласс t -схем со специальными комбинаторными свойствами.

Ранее разными авторами было получено конструктивное описание всех полностью регулярных кодов силы 0 в графах Джонсона, а также показано, что всякая блок-схема силы $w - 1$ с размером блока w является полностью регулярным кодом в графе $J(n, w)$. Эти схемы включают в себя широко известные системы троек и четверок Штейнера.

В данной работе мы рассматриваем классические конструкции Оллтопа расширения блок-схем из существующих и показываем, что их применение к полностью регулярным кодам дает новые полностью регулярные коды. В качестве исходных блок-схем для этих конструкций берутся блок-схемы с размером блока, равным половине от числа точек.

Пусть D является $t - (n, w, \lambda)$ -схемой. Введем следующие обозначения:

$$D1 = \{\Delta \cup (n+1) : \Delta \in D\}, D2 = \{\{1, \dots, n\} \setminus \Delta : \Delta \in D\}.$$

Рассмотрим случай, когда $n = 2w$. Любой полностью регулярный код в графе $J(2w, w)$ наследует свойство антиподальности графа: код является либо плюс-антиподальным (т.е. любые две антиподальные вершины являются одновременно либо кодовыми, либо

некодовыми), либо минус-антиподальным (в любой паре антиподальных вершин одна является кодовой, а другая – нет) [Avgustinovich S.V., Mogilnykh I.Yu. Perfect 2-colorings of Johnson graphs $J(6; 3)$ and $J(7; 3)$ // LNCS, Springer. 2008. Vol. 5228. Pp. 11–19]. Оллтопом в [Alltop W.O. Extending t -designs // Journal of Combinatorial Theory. Ser. A. 1975. Vol. 18, iss. 2. Pp. 177–186] доказаны теоремы 1-3.

Теорема 1. Пусть D является $t - (2w, w, \lambda)$ -схемой, t чётно и $D = D^2$. Тогда D является $(t + 1) - (2w, w, \eta)$ -схемой, где $\eta = \lambda(w - t) / (2w - t)$.

Следствие. Пусть C является плюс-антиподальным полностью регулярным кодом в $J(2w, w)$ с $p = 1$. Тогда сила C как блок-схемы нечётна.

Опишем две конструкции Оллтопа. Ниже \bar{D} обозначает подмножество вершин графа $J(2w + 1, w)$, дополнительное к D .

Теорема 2. Пусть D является $t - (2w + 1, w, \lambda)$ -схемой, t чётно. Тогда $D_1 \cup D_2$ является $(t + 1) - (2w + 2, w + 1, \lambda)$ -схемой.

Теорема 3. Пусть D является $t - (2w + 1, w, \lambda)$ -схемой, t нечётно и $|D| = \binom{2w+1}{w}/2$. Тогда $D_1 \cup \bar{D}_2$ является $(t + 1) - (2w, w, \lambda)$ -схемой.

Вариант этих утверждений для полностью регулярных кодов имеет вид:

Теорема 4. Пусть C является полностью регулярным кодом с $p = 1$ в

$J(2w + 1, w)$. Тогда код $C_1 \cup C_2$ является полностью регулярным в $J(2w + 2, w + 1)$.

Теорема 5. Пусть C является полностью регулярным кодом с $p = 1$ в $J(2w + 1, w)$, $|C| = \binom{2w+1}{w}/2$. Тогда код $C_1 \cup \bar{C}_2$ является полностью регулярным в $J(2w + 2, w + 1)$.

В данной работе, используя эти два утверждения и классификацию полностью регулярных кодов в $J(9, 4)$ с радиусом покрытия 1 [Avgustinovich S.V., Mogilnykh I.Yu. On completely regular codes in Johnson graphs $J(2w+1, w)$ with covering radius 1 // Proceedings of Twelfth International workshop on Algebraic Combinatorial Coding Theory (ACCT-2010), P. 20–26. September 5–11, 2010, Akademgorodok, Novosibirsk, Russia], мы построили полностью регулярные коды в графе $J(10, 5)$ с массивами пересечений $\{\alpha_0 = 13, \beta_0 = 12, \gamma_1 = 16, \alpha_1 = 9\}$ и $\{\alpha_0 = 5, \beta_0 = 20, \gamma_1 = 8, \alpha_1 = 17\}$. Отметим, что конструкции расширения полностью регулярных кодов применимы к кодам произвольной силы t (в отличие от варианта конструкций для блок-схем).

1.9. Разработка эффективных алгоритмов построения коммуникационного дерева в беспроводных сенсорных сетях

Рассмотрена следующая NP-трудная в сильном смысле задача построения коммуникационного дерева, которая возникает в беспроводных сенсорных сетях. В произвольном неориентированном n -вершинном графе с неотрицательными весами ребер требуется построить остовное дерево, в котором сумма по всем вершинам максимальных весов инцидентных вершине ребер минимальна.

Элементы многих коммуникационных сетей используют беспроводную связь для обмена информацией. При этом потери энергии элемента пропорциональны ds , где $s \geq 2$, а d – дальность передачи [59]. В некоторых сетях, например, в беспроводных сенсорных сетях, элементы (сенсоры) имеют ограниченный запас энергии, и эффективное использование энергии сенсоров позволяет продлить время функционирования (жизни) сети [57, 66, 68, 69]. Для рационального использования энергии современные сенсоры способны регулировать дальность радиопередачи, и тогда актуальной становится проблема определение дальности передачи каждого элемента сети таким образом, чтобы минимизировать общую энергию, затрачиваемую на поддержание связного графа. Если предположить, что радиосигнал одинаково распространяется во всех направлениях, то все элементы, находящиеся в зоне передачи (не далее, чем дальность передачи), получают сообщение. В этом случае можно считать, что коммуникационная сеть (остовный подграф по ребрам которого осуществляется передача) – это полный граф [59, 62, 65, 66]. Однако не всегда сигнал распространяется одинаково во всех направлениях и на любое расстояние. Поэтому в общем случае следует считать, что коммуникационный граф $G = (V, E)$ может быть произвольным остовным подграфом, как и потери энергии по обеспечению передачи по ребру графа. Таким образом, если $c_{ij} \geq 0$ – потери энергии, связанные с передачей данных из $i \in V$ в $j \in V$, то в связном подграфе $T = (V, E'), E' \subseteq E$ потери энергии вершины $i \in V$ равны $E_i(T) = \max_{j: (i,j) \in E'} c_{ij}$. Целью исследований является решение задачи построения такого остовного подграфа T , в котором сумма $\sum_{i \in V} E_i(T)$ минимальна. Без ограничения общности подграф T можно считать остовным деревом.

Подобные задачи возникают, например, в беспроводных сенсорных сетях, когда расположение сенсоров известно и требуется синтезировать энергоэффективный граф, связывающий все сенсоры [68]. В литературе в качестве коммуникационного графа сенсорной сети принято рассматривать остовное дерево *минимального* веса P , когда вес ребра, связывающего пару вершин, – это квадрат расстояния между этими вершинами [68]. Однако минимальный остов не всегда является оптимальным коммуникационным графом сенсорной сети.

В работе [65] исследовалась задача определения дальности передачи каждой вершины, размещённой в евклидовом пространстве таким образом, чтобы индуцировать сильно связный граф, в котором общие энергозатраты на связь минимальны. Для частных случаев, когда вершины расположены на прямой, предложены полиномиальные алгоритмы решения задачи. Доказана NP-трудность задачи в трехмерном евклидовом пространстве.

Авторами работы [59] предложен алгоритм с асимптотической точностью $5/3$, полиномиальный алгоритм, строящий $11/6$ -приближённое решение, а также точный алгоритм – метод ветвей и границ, в котором используется новая постановка задачи в виде задачи целочисленного линейного программирования.

В работе [62] рассмотрена задача определения мощностей радиопередатчиков для передачи данных на два расстояния: «малое» и «большое». Показана NP-трудность этой задачи. Предложен полиномиальный алгоритм, который строит решение с числом передатчиков на большие расстояния, не превосходящим $11/6$ от числа таких передатчиков в оптимальном решении. Также предложен экспоненциальный $9/5$ -приближенный алгоритм. Эти результаты получены для случая, когда элементы распределены в евклидовом пространстве, однако легко обобщаются и на произвольную метрику.

Нами получены следующие результаты:

- ✓ найдены частные случаи полиномиальной разрешимости задачи;
- ✓ показано, что минимальный остов, веса рёбер которого принадлежат отрезку $[a, b]$, является $\left(2 - \frac{2a}{a + b + 2b/(n-2)}\right)$ -приближенным решением и, что задача построения $1,00048$ -приближенного решения NP-трудна;
- ✓ предложен эвристический полиномиальный алгоритм и осуществлён его апостериорный анализ.

1.10. Проведение численных экспериментов и апостериорного анализа для разработанного метода глобальной маршрутизации

Глобальная трассировка является одним из важнейших этапов проектирования сверхбольших интегральных схем (СБИС), на котором для каждой цепи определяется множество используемых областей маршрутизации в условиях ограничений на трассировочные ресурсы и время прохождения сигнала. В литературе встречается несколько формулировок задачи глобальной трассировки (ЗГТ) с различными критериями и ограничениями, достаточно подробный обзор которых приведен в работе [71].

Основной целью глобальной трассировки является маршрутизация всех цепей СБИС без нарушения ограничений. При этом даже простейшая постановка, в которой требуется осуществить трассировку двухтерминальных цепей в условиях ограниченности трассировочных ресурсов, является NP-трудной задачей [72].

Для решения ЗГТ исследователями предложены различные подходы, в которых трассировка, как правило, осуществляется лишь на двух слоях. В основе этих подходов лежат алгоритмы последовательной маршрутизации [73], алгоритмы трассировки с разрывом связей и поиском новых соединений [74], алгоритмы, основанные на решении задач о многопродуктовом потоке [75], иерархические методы [76, 77], а также различные метаэвристики [78, 79, 80, 81, 82].

При проектировании современных СБИС на этапе глобальной маршрутизации, наряду с учетом трассировочных ресурсов, все большее внимание уделяется времени прохождения сигнала [71, 83, 84]. При этом плотность соединений и временная задержка являются, как правило, конкурирующими критериями, и в литературе практически отсутствуют публикации, в которых эти критерии рассматриваются совместно.

Нами предложен новый подход к решению ЗГТ, который учитывает как трассировочные ресурсы, так и задержки прохождения сигнала, и применим при проектировании СБИС с произвольным количеством слоев маршрутизации.

Разработан эффективный метод решения задачи. Для предложенного алгоритма был проведен численный эксперимент, в котором решались тестовые задачи с числом цепей 1900-3500, каждая из которых содержала от 2 до 40 терминалов. Соответствующие глобальные графы имели от 3000 до 5500 вершин.

Построенные решения сравнивались с трассировками, найденными глобальным маршрутизатором Labyrinth, в котором используется алгоритм, предложенный в [71]. Выбор данного маршрутизатора обусловлен тем, что Labyrinth показал высокую эффективность по сравнению с другими маршрутизаторами [88].

Мы используем аббревиатуру **A_R** для обозначения алгоритма **A**, который строит решение в области **R**, являющейся подграфом глобального графа. Исследовались два варианта предлагаемого алгоритма: **S** – последовательное построение деревьев (частный случай, когда $|Q^s|=1$ для всех $s=1, \dots, S$) и **C** – одновременное построение деревьев. При этом применялись различные способы учета емкостей поддеревьев, выполнялось не более 10 итераций алгоритма MAD для каждого варианта, и запоминались от 1 до 20 лучших (по задержкам) деревьев для каждой цепи. Если цепь s оказывалась не критической, то в Q^s выбирались деревья с наименьшими максимальными задержками. В противном случае выбирались деревья-кандидаты, в которых наименьшими были задержки в критические терминалы. Число итераций алгоритма было ограничено пятью, так как большее число итераций редко приводило к улучшению решения.

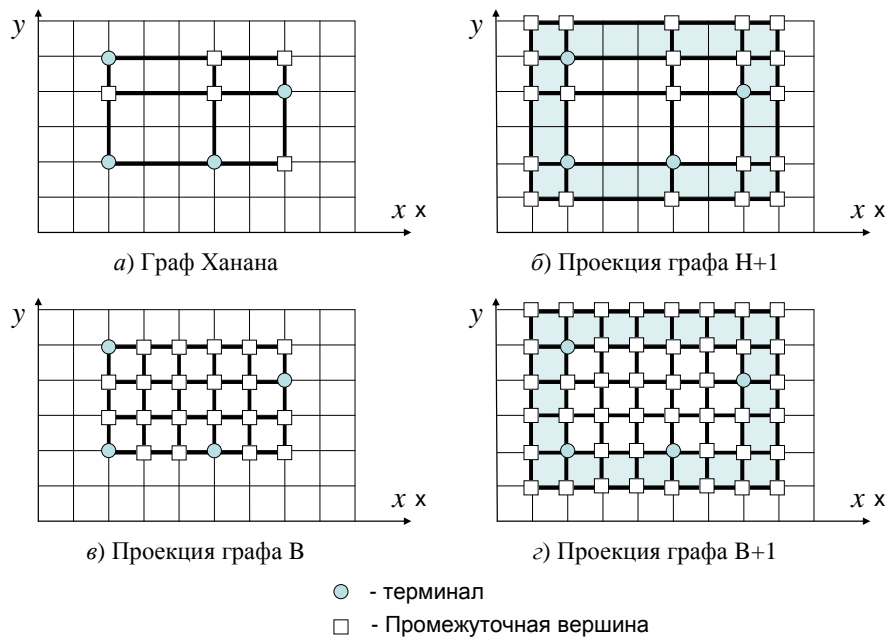


Рисунок 1.10.1 -- Примеры проекций графов.

В качестве подграфа $R = G'$ использовались различные конструкции. Например, через **H** обозначен подграф глобального графа, содержащий одинаковые плоские графы Ханана, порожденные проекциями терминалов, на каждом слое, в котором вершины смежных слоев с совпадающими (x, y) – координатами соединены ребром. Подграф **B** строится следующим образом. Сначала на плоскости xOy находится минимальный прямоугольник **B_p**, содержащий проекции всех терминалов цепи. Затем строится параллелепипед с основанием **B_p** и высотой, равной количеству слоев схемы. Граф **B** – это часть глобального графа, заключенная внутри построенного параллелепипеда. Кроме перечисленных подграфов **R**, строились их расширения **(R+1)** на одну координату в

направлениях возрастания и убывания по осям $0x$ и $0y$. Например, $C_{(H+1)}$ означает алгоритм одновременного построения деревьев для всех цепей, который осуществляет трассировку в графе H , расширенном на одну координату в направлениях возрастания и убывания по осям $0x$ и $0y$.

На рисунке 1.10.1 приведены примеры проекций графов R . Затемненные полосы – это расширения соответствующих графов. Граф R содержит аналогичные подграфы-проекции на всех слоях, при этом вершины соседних слоев с одинаковыми (x, y) -координатами соединены ребрами.

Таблица 4 — Усредненные характеристики алгоритмов.

Алгоритм	Превышение суммарной длины соединений (%)	Среднее превышение пропускной способности (%)	Максимальное превышение пропускной способности (%)	Процент переполненных ребер (%)	Процент превышения директивного времени (%)
Labyrinth	11.50	12.57	85.00	45.00	23.75
S_H	4.38	15.76	54.76	18.26	16.20
C_H	0.00	16.02	56.9	16.87	9.32
S_(H+1)	14.30	8.11	16.38	25.10	14.54
C_(H+1)	0.68	16.77	38.23	25.81	8.09
S_B	5.02	15.35	47.42	19.09	11.37
C_B	0.35	19.56	53.4	18.68	9.65
S_(B+1)	13.77	8.8	14.04	24.23	13.37
C_(B+1)	2.26	49.14	21.42	2.10	12.49

В таблице приведены усредненные показатели тестируемых алгоритмов. Используются следующие обозначения. Превышение суммарной длины соединений – это процент превышения общей длины соединений над минимальной длиной соединений в найденных решениях. Среднее превышение пропускной способности – это процентное выражение среднего превышения пропускных способностей переполненных глобальных ребер. Максимальное превышение пропускной способности – это процентное выражение максимального переполнения пропускных способностей ребер. Процент переполненных ребер – это процент глобальных ребер, по которым прошло больше соединений, чем их пропускная способность. Превышение директивного времени – это процентное выражение максимального превышения временной задержки в основных выходах схемы относительно допустимой задержки.

Большее затемнение клетки таблицы соответствует большему значению соответствующей характеристики. Как видно из результатов эксперимента, наш метод в

среднем предпочтительнее алгоритма Labyrinth почти по всем показателям. Особенно значительный выигрыш наблюдается по задержкам.

Отметим, что алгоритм Labyrinth, а также многие другие известные алгоритмы, неприменим в случае трех и более слоев маршрутизации. Кроме того, он не учитывает время прохождения сигнала. Предложенный в работе алгоритм позволяет учитывать как сетевые задержки, так и внутренние задержки элементов, и применим при произвольном числе слоев СБИС.

2 Показатели

2.1. Защиты диссертаций

2.2. Список студентов, аспирантов, докторантов и молодых исследователей, закрепленных в сфере науки и образования.

2.3. Количество подготовленных и опубликованных статей:

Опубликовано 13, принято к печати 11, сдано в печать 6 статей (см. Приложение А).

2.4. Количество сделанных докладов:

Сделано 10 докладов на международных научных форумах. (см. Приложение Б).

3. Заключение

В процессе выполнения 4 этапа НИР получены следующие основные результаты.

1) Найдена нижняя оценка числа бент-функций на минимальном расстоянии от бент-функции из класса Мэйорана – МакФарланда.

2) В терминах преобразования Фурье введено понятие реконструктивного множества в булевом кубе. Получена характеристика реконструктивных множеств, являющихся линейными подпространствами. Установлены необходимые и достаточные условия реконструктивности сферы. Приведено достаточное условие реконструктивности двух концентрических сфер.

3) Спектром гамильтонова цикла (кода Грея) в булевом n -мерном кубе называется набор $a = (a_1, \dots, a_n)$, где a_i – число рёбер i -го направления в цикле. Известны необходимые условия существования кода Грея со спектром a : числа a_i чётные и для любого $k = 1, \dots, n$ сумма k произвольных компонент набора a не меньше чем $2k$. Доказано существование такой размерности N , что если необходимые условия на спектр являются достаточными для существования гамильтонова цикла с таким спектром в булевом N -мерном кубе, то сформулированные выше условия являются достаточными и для всех размерностей n .

4) Установлено, что сложность реализации в классе обобщённых (троичных) π -схем троичного счётчика кратности 3, зависящего от трёх переменных, равна 18.

5) Подмножество вершин графа называется k -кратным совершенным кодом радиуса r , если для каждой вершины шар радиуса r с центром в этой вершине содержит в точности k кодовых вершин. Получен критерий, который по параметрам совершенной 2-раскраски двоичного n -куба определяет, является ли она кратным совершенным кодом заданного радиуса $r > 1$ некоторой кратности.

6) Для NP-трудной в сильном смысле задачи построения оптимального коммуникационного дерева в беспроводной сенсорной сети найдены частные случаи полиномиальной разрешимости; показано, что минимальный остов, веса рёбер которого

принадлежат отрезку $[a, b]$, является $\left(2 - \frac{2a}{a + b + 2b/(n-2)}\right)$ -приближенным решением и, что задача построения 1,00048-приближенного решения NP-трудна; предложен эвристический полиномиальный алгоритм и осуществлён его апостериорный анализ.

7) Доказана NP-полнота нескольких актуальных задач разбиения последовательности векторов, содержащих конечное число элементов, по критерию минимума суммы квадратов расстояний.

8) Предложены 2-приближённые полиномиальные алгоритмы, а также точные псевдополиномиальные алгоритмы для ряда труднорешаемых задач поиска подпоследовательностей векторов.

9) Для задачи об инвестировании проектов с одновременным поступлением проектов в систему построен алгоритм, основанный на идее многопараметрического динамического программирования. Трудоёмкость алгоритма становится псевдополиномиальной в случае, когда число (m) различных моментов поступления проектов в систему ограничено константой. Доказана неувлучшаемость полученного результата: невозможность избавиться от псевдополиномиального характера оценки трудоёмкости (поскольку даже при $m=2$ задача NP-трудна) и невозможность построения аналогичного псевдополиномиального алгоритма для случая неограниченного m , поскольку в этом случае задача NP-трудна в сильном смысле.

Кроме того, для частного случая задачи, когда вклад каждого проекта в общий ресурсный пул неотрицателен, разработан полиномиальный алгоритм точного решения. В то же время, «симметричная» задача, в которой вклад каждого проекта в общий ресурсный пул неположителен, является (как показано нами) NP-трудной в сильном смысле, а значит, вряд ли допускает точное решение за полиномиальное время (если верна гипотеза о несовпадении классов P и NP).

10) Установлена полиномиальная разрешимость смешанной задачи $(J + \bar{O})|p_{ij} = 1, \eta \leq 2|C_{max}$, т.е задачи, обобщающей классические задачи job shop и обобщённую задачу open shop (на минимум длины расписания) на случай, когда в примере могут присутствовать работы обоих типов (как job shop, так и open shop). При этом рассматривается частный случай этого обобщения, когда длины всех операций равны 1 либо 0, а каждая работа имеет не более двух операций. Установлено также, что данный результат является наилучшим возможным для задач такого типа (в плане построения эффективного алгоритма точного решения), поскольку любая попытка ослабления какого-либо из ограничений на входные данные задачи приводит к потере свойства её эффективной разрешимости.

11) Установлено, что оптимальное расписание в задаче job shop с разрешением прерываний, на минимум произвольной регулярной функции от моментов завершения операций может быть найдено простым жадным алгоритмом, при условии, что найдены (угаданы) подходящие приоритеты выполнения операций на каждой из машин. Таким образом, впервые описан конструктивный метод решения данной задачи.

12) Построен алгоритм A_A , который для любого заданного примера двухмашинной задачи flow shop с n работами и для любого заданного $\Delta \geq 0$ перечисляет (без повторений)

все Δ -оптимальные перестановки из n работ, затрачивая не более чем $O(n \log n)$ единиц времени на отыскание каждой Δ -оптимальной перестановки и на остановку алгоритма после отыскания последнего решения. При этом на каждом шаге алгоритма используется не более чем $O(n)$ ячеек памяти вычислительной машины. Таким образом, алгоритм эффективен, поскольку имеет полиномиальную задержку и обходится полиномиальной памятью.

Установлено свойство связности множества Δ -оптимальных перестановок: доказано, что любая Δ -оптимальная перестановка может быть получена из любой другой Δ -оптимальной перестановки последовательной транспозицией двух соседних элементов, так что при этом все промежуточные перестановки также Δ -оптимальны.

13) Рассмотрена цеховая задача открытого типа с маршрутизацией, которая является обобщением двух классических задач дискретной оптимизации: цеховой задачи открытого типа $(O \parallel C_{\max})$ и метрической задачи коммивояжёра. Для различных версий задачи построены быстрые алгоритмы, строящие приближённые решения с гарантированными оценками точности:

а) для задачи с двумя машинами и двумя вершинами предложена вполне полиномиальная приближённая схема (FPTAS), то есть семейство алгоритмов, которые для произвольного фиксированного ε строят $(1+\varepsilon)$ -приближённые решения за время, ограниченное полиномом от размера входа задачи и от величины $1/\varepsilon$;

б) для задачи на двух машинах на произвольной транспортной сети построен $13/8$ -приближённый алгоритм; в случае, когда задача коммивояжёра на транспортной сети может быть решена точно, оценку точности этого алгоритма можно улучшить до $4/3$;

в) для задачи с нефиксированным числом вершин на произвольной транспортной сети предложен $O(\sqrt{m})$ -приближённый алгоритм.

Полученные результаты имеют мировой уровень, а исполнители представляют передовой фронт науки в указанных областях.

По результатам 4 этапа НИР напрашивается вывод о целесообразности продолжения работ.

4. Список использованных источников

1. Агеев А.А., Бабурин А.Е., Гимади Э.Х. Полиномиальный алгоритм с оценкой точности $3/4$ для отыскания двух непересекающихся гамильтоновых циклов максимального веса // Дискрет. анализ и исслед. операций. Сер.1. – 2006. – Т.12, №2. – С.11–20.
2. Бабурин А.Е., Гимади Э.Х., Коркишко Н.М. Приближенные алгоритмы для нахождения двух реберно непересекающихся гамильтоновых циклов минимального веса // Дискрет. анализ и исслед. операций. Сер 2. – 2004. – Т.11, №1. – С.11–25.
3. Бабурин А.Е., Гимади Э.Х. Приближенный алгоритм отыскания d -однородного регулярного остовного связного подграфа максимального веса в полном графе со случайными весами ребер // Дискрет. анализ и исслед. операций. Серия 2. – 2006. – Т. 13, N 2, С. 3-20.
4. Бабурин А.Е., Гимади Э.Х. Об асимптотической точности эффективного алгоритма решения задачи m -PSP на максимум в многомерном евклидовом пространстве // Тр. ИММ УрО РАН – 2010 – Т. 6. № 3 – С. 12–24.
5. Гимади Э.Х. Новая версия асимптотически точного алгоритма решения евклидовой задачи коммивояжера // Труды XII Байкальской международной конференции. Методы оптимизации и их приложения. Том 1, Иркутск, 2001. С. 117–124.
6. Гимади Э.Х. Асимптотически точный алгоритм отыскания одного и двух реберно непересекающихся маршрутов коммивояжера максимального веса в евклидовых пространствах // Труды ИММ УрО РАН. – 2008. – Т.14. № 2. – С. 23–32.
7. Гимади Э.Х., Глазков Ю.В., Глебов А.Н. Приближенные алгоритмы решения задачи о двух коммивояжерах в полном графе с весами ребер 1 и 2 // Дискретный анализ и исследование операций. Сер. 2. 2007. Т.14. № 2. С. 41-61.
8. Гимади Э.Х., Шахшнейдер А.В. Приближенные алгоритмы с оценками для задач маршрутизации на случайных входах с ограниченным числом клиентов в каждом маршруте // Автоматика и телемеханика. 2012. Вып. 2. С. 126-140.
9. Гимади Э.Х., Ивоина Е.В. Приближенные алгоритмы решения задачи о двух коммивояжерах на максимум // Дискретный анализ и исследование операций. 2012. Т. 19, № 1. С. 17-32.
10. Глебов А.Н., Замбалаева Д.Ж., Ивоина Е.В. Эффективные алгоритмы с гарантированными оценками точности для задач одного и двух коммивояжеров на максимум // Сб. трудов XV Байкальской школы-семинара "Методы оптимизации и их приложения" - 2011, 6 стр.

11. Глебов А.Н., Гордеева А.В., Замбалаева Д.Ж. Алгоритм с оценкой $7/5$ для задачи о двух коммивояжерах на минимум с различными весовыми функциями // Сибирские электронные математические известия. 2011. Т. 8. С. 296-309.
12. Глебов А.Н., Замбалаева Д.Ж. Полиномиальный алгоритм с оценкой точности $7/9$ для задачи о двух коммивояжерах на максимум // Дискрет. анализ и исслед. операций. 2011. Т. 18, № 4. С. 17-48.
13. Глебов А.Н., Замбалаева Д.Ж. Приближенный алгоритм решения задачи о двух коммивояжерах на минимум с различными весовыми функциями // Дискрет. анализ и исслед. операций. 2011. Т. 18, № 5. С. 11-37.
14. Сердюков А.И. Асимптотически точный алгоритм для задачи коммивояжера на максимум в евклидовом пространстве // Методы целочисленной оптимизации (Управляемые системы). Новосибирск, 1987. Вып. 27. С. 79-87.
15. A.E. Baburin, E.Kh.Gimadi. On the asymptotic optimality of an algorithm for solving the maximum m-PSP in a multidimensional Euclidean space // Proceedings of the Steklov Institute of Mathematics, 2011, Vol. 272, Suppl. 1, P. 1-13.
16. A.E. Baburin, F. Della Croce, E.K. Gimadi, Y.V. Glazkov and V.Th. Paschos. Approximation algorithms for the 2-peripatetic salesman problem with edge weights 1 and 2 // Discrete Applied Mathematics, Vol. 157, No 9, 2009, P. 1988-1992.
17. Berman P., Karpinski M. $8/7$ -approximation algorithm for (1,2)-TSP // Proc. 17th ACM-SIAM SODA, 2006. P. 641-648.
18. Edward Kh. Gimadi, Alexey M. Istomin, Ivan A. Rykov. On Approximation Polynomial Algorithm for the Capacitated Peripatetic Salesman Problem. International Symposium on Combinatorial Optimization (CO 2012), 17–19 September 2012, University of Oxford, UK, 2012.
19. Christofides N. Worst-case analysis of a new heuristic for the traveling salesman problem // Technical Report CS-93-13, Carnegie Mellon University, 1976.
20. De Brey M. J. D., Volgenant A. Well-solved cases of the 2 -peripatetic salesman problem // Optimization. 1997. V. 39, N 3. P. 275-293.
21. De Kort J.~B. J.~M. Lower bounds for symmetric K -peripatetic salesman problems // Optimization. 1991. V. 22, N 1. P. 113-122.
22. Duchenne E., Laporte G., Semet F. Branch-and-cut algorithms for the undirected m -peripatetic salesman problem // European J. Oper. Res. 2005. V. 162, N 3. P. 700—712
23. Gabow H.~N. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems // Proc. of the 15th annual ACM symposium on theory of computing (Boston, April 25-27). New York: ACM Press, 1983. P. 48-456.

24. Krarup J. The peripatetic salesman and some related unsolved Problems // Combinatorial programming: methods and applications (Proc. NATO Advanced Study Inst., Versailles, 1974). Dordrecht: Reidel, 1975. P. 173-178.
25. Papadimitriou C.H., Yannakakis M. The travelling salesman problem with distances One and Two // Math. Oper. Res. 1993. V. 18, N 1. P. 1-11.
26. The Traveling Salesman Problem and its variations (ed. by A. Punnen and G. Gutin). Kluwer Academic Publishers. Dordrecht/Boston/London. 2002.
27. A. van Zuylen. Multiplying Pessimistic Estimators: Deterministic Approximation of Max TSP and Maximum Triangle Packing // Computing and Combinatorics, 16th Annual International Conference (COCOON 2010). – Nha Trang, Vietnam, July 19-21, 2010. – Proceedings. Lecture Notes in Computer Science. – V. 6196. – P. 60-69.
28. Дискретные задачи размещения. Библиотека тестовых примеров. Конкурентная задача размещения предприятий <http://math.nsc.ru/AP/benchmarks/Compet-FL/Compet-FL.html>
29. I. Averbakh, O. Berman, and I. Chernykh. A 6/5-approximation algorithm for the two-machine routing open shop problem on a 2-node network // European Journal of Operational Research, 166(1). 2005. P. 3-24.
30. I. Averbakh, O. Berman, and I. Chernykh. The Routing Open-Shop Problem on a Network: Complexity and Approximation // European Journal of Operational Research, 173(2). 2006. P. 521-539.
31. R. Bellman, A.O. Esogbue, and I. Nabeshima. Mathematical aspects of scheduling and applications // Pergamon Press, Oxford. 1982.
32. J.C. Billaut and P. Lopez. New results for the enumeration of optimal job sequences, Unpublished manuscript. 1997.
33. T.C.E. Cheng. Efficient implementation of Johnson's rule for the $n/2/F/F_{\max}$ scheduling problem // Computers and Industrial Engineering, 1992, № 22. P. 495- 499.
34. M. Garey and D. Johnson. Computers and Intractability, A Guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco, CA, 1979.
35. M.R. Garey, D.S. Johnson, and R. Sethi. The Complexity of Flowshop and Jobshop Scheduling // Mathematics of Operations Research, 1976, № 1. P. 117-129.
36. T. Gonzalez and S. Sahni. Open shop scheduling to minimize finish time // Journal of the Association for Computing Machinery, 1976, № 23(4). P. 665-679.
37. R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnoy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey // Annals of Discrete Mathematics, 1979, № 5. P. 287-326.

38. O. Ibarra and C.E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems Vehicle Scheduling on a Tree with Release and Handling Times // Journal of the Association for Computing Machinery, 1975, № 22. P. 463-468.
39. S.M. Johnson. Optimal two- and three-stage production schedules with setup times included // Naval Research Logistics Quarterly, 1954, № 1. P. 61-67.
40. D.S. Johnson, M. Yannakakis, and C.H. Papadimitriou. On generating all maximal independent sets // Information Processing Letters, 1988, № 27. P. 119-123.
41. Y. Lin and J. Deng. On the structure of all optimal solutions of the two-machine flowshop scheduling problem // OR Transactions, 1999, № 3(2). P. 10-20.
42. S.N.N. Pandit and Y.V. Subrahmanyam. Enumeration of all sequences // Opsearch, 1975, № 12. P. 35-39.
43. W. Szwarc. Extreme solutions of the two machine flow shop problem // Naval Research Logistics Quarterly, 1981, № 28(1). P. 103-114.
44. D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, S.V. Sevastianov, and D.B. Shmoys. Short shop schedules // Operations Research, 1997, № 45(2). P. 288-294.
45. Кононов А.В. О цеховой задаче открытого типа на двух машинах с маршрутизацией в двухвершинной сети // Дискретный анализ и исследование операций, 2012, № 19(2). С. 54-74.
46. Пяткин А.В., Черных И.Д. Задача открытого типа с маршрутизацией и разрешением прерываний на двухвершинной сети // Дискретный анализ и исследование операций, 2012, принята к печати.
47. С.В. Севастьянов, Д.А. Чемисова, И.Д. Черных. О некоторых свойствах оптимальных расписаний в задаче Джонсона с прерываниями // Дискретный анализ и исследование операций, 2006, Сер. 1. Т. 13, N 3. С. 83-102.
48. Кельманов А.В., Пяткин А.В. NP-полнота некоторых задач выбора подмножества векторов // Дискретный анализ и исследование операций. 2010. Т.17, №5. С. 37-45.
49. Кельманов А.В., Хамидуллин С.А. Апостериорное обнаружение заданного числа одинаковых подпоследовательностей в квазипериодической последовательности // Журн. вычисл. математики и мат. физики, 2001, Т.41, № 5, С. 807-820.
50. Гимади Э.Х., Кельманов А.В., Кельманова М.А. Хамидуллин С.А. Апостериорное обнаружение в числовой последовательности квазипериодического фрагмента при заданном числе повторов // Сиб. журн. индустр. математики. 2006. Т.9 №1(25). С.55-74.

51. Кельманов А.В., Михайлова Л.В. Совместное обнаружение в квазипериодической последовательности заданного числа фрагментов из эталонного набора и ее разбиение на участки, включающие серии одинаковых фрагментов // Журн. вычисл. математики и мат. физики, 2006, Т.46, №1, С. 172-189.
52. Кельманов А.В., Михайлова Л.В. Хамидуллин С.А. Об одной задаче поиска упорядоченных наборов фрагментов в числовой последовательности // Дискретный анализ и исследование операций. 2009. Т.16. № 4. С. 31-46.
53. Kel'manov A.V., Jeon B. A Posteriori Joint Detection and Discrimination of Pulses in a Quasiperiodic Pulse Train // IEEE Transactions on Signal Processing, 2004, Vol. 52, No. 3, pp. 1-12.
54. Kel'manov A.V., Khamidullin S.A. An Algorithm for Recognition of a Vector Alphabet Generating a Sequence with a Quasi-Periodic Structure // Pattern Recognition and Image Analysis. 2010. Vol. 20, No.4, pp. 451-458.
55. Papadimitriou C.H. Computational Complexity. New-York: Addison-Wesley, 1994. 523 P.
56. Garey M. R., Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman, 1979. 314 P.
57. С.Н. Астраков, А.И. Ерзин, В.В. Залюбовский. Сенсорные сети и покрытие плоскости кругами // Дискретный анализ и исследование операций, 2009, т. 16, \No 3, с. 3-19.
58. Л.Ф. Тот. Расположения на плоскости, на сфере и в пространстве. – М.: Изд. Физ.-мат. литературы. 1958. 365 с.
59. E. Althaus, et al. Power Efficient Range Assignment for Symmetric Connectivity in Static Ad Hoc Wireless Networks // Wireless Networks, 2006, v. 12, No. 3, p. 287-299.
60. P. Berman, M. Karpinski. On some tighter inapproximability results // Tech. Report TR98-065, 1998, ECCC.
61. A.E.F. Clementi, P. Penna, R. Silvestri. On the Power Assignment Problem in Radio Networks // Electronic Colloquium on Computational Complexity (ECCC), (054), 2000.
62. P. Carmi, M.L. Katz. Power Assignment in Radio Networks with Two Power Levels // Algorithmica, 2007, No. 47, p. 183-201.
63. M. Diane, J. Plesnik. An Integer Programming Formulation of the Steiner Problem in Fraphs // Methods and Models of Opetations Research, 1993, No. 37, p. 107-111.
64. R. Kershner. The Number of Circles Covering a Set // American Journal of Mathematics, 1939, v. 61, No. 3, p. 665-671.
65. L.M. Kirousis, E. Kranakis, D. Krizanc, A. Pelc. Power consumption in packet radio networks // Theoretical Computer Science, 2000, No. 243, p. 289-305.

66. G.J. Pottie, W.J. Kaiser. Wireless Integrated Network Sensors // Communications ACM, 2000, v. 43, No. 5, p. 51-58.
67. F.G. Toth. Covering the Plane with Two Kinds of Circles // Discrete & Computational Geometry, 1995, v. 13, No. 3, p. 445-457.
68. J. Wu, F. Dai. Virtual Backbone Construction in MANETs using Adjustable Transmission Ranges // IEEE Trans. On Mobile Computing, 2006, v. 5, No. 9, p. 1188-1200.
69. J. Wu, S. Yang. Energy-Efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges // Int. J. of Foundations of Computer Science, 2005, v. 16, No. 1, p. 3-17.
70. H. Zhang, J.C. Hou. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks // Ad Hoc & Sensor Wireless Networks, 2005, v. 1, No. 1-2, p. 89-124.
71. Kastner R., Bozorgzadeh E., Sarrafzadeh M. Pattern routing: use and theory for increasing predictability and avoiding coupling // IEEE Trans. on CAD. 2002. V. 21. P. 777-791.
72. Kramer M. R., van Leeuwen J. The complexity of wire routing and finding minimum area layouts for arbitrary VLSI circuits // Advances in computing research. 1984. V. 2: VLSI theory. (F. P. Preparata, ed.). P. 129-146.
73. Chiang C., Wong C. K., Sarrafzadeh M. A weighted Steiner tree-based global router with simultaneous length and density minimization // IEEE Trans. on CAD. 1994. V. 13. P. 1461-1469.
74. Ting B., Tien B. Routing techniques for gate array // IEEE Trans. on CAD. 1983. V. CAD-2. P. 301-312.
75. Albrecht C. Global routing by new approximation algorithms for multicommodity flow // IEEE Trans. on CAD. 2001. V. 20. P. 622-632.
76. Burstein M., Pelavin R. Hierarchical wire routing // IEEE Trans. on CAD. V. CAD-2. P. 223-234.
77. Hayashi M., Tsukiyama S. A hybrid hierarchical approach for multi-layer global routing // Proc. European Design and Test Conference. Paris, 1995. P. 492-496.
78. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing // Science. 1983. V. 220. P. 671-680.
79. Sechen C., Sangiovanni-Vincentelli A. The Timber-Wolf placement and routing package // IEEE J. of Solid-State Circuits. 1985. V. SC-20. P. 510-522.
80. Chen Y.A., Liu Y.L., Hsu Y.C. A new global router for ASIC design based on simulated evolution // Proc. Int. Symp. on VLSI Technology, Systems and Applications. 1989. P. 261-265.
81. Esbensen H. A macro-cell global router based on two genetic algorithms // Proc. EDAC. Paris, 1994. P. 428-433.

82. Youssef H., Sait S.M. Timing-driven global routing for standard-cell VLSI design // Computer systems: Science and Engineering. 1999. V. 14. P. 175-185.
83. Cong J., Madden P. Performance driven global routing for standard cell design // Proc. ACM ISPD. Napa Valley, California, 1997. P. 73-80.
84. Wang D., Kuh E.S. Performance-driven interconnect global routing // Proc. Great Lake Symp. VLSI. Montreal, Canada, 1996. P. 132-136.
85. Elmore W.C. The transient response of damped linear networks with particular regards to wide-band amplifies // J. Appl. Phys. 1948. V. 19. P. 55 – 63.
86. Rubinstein J., Penfield P., Horowitz M.A. Signal delay in RC tree networks // IEEE Trans. on CAD. 1983. V. 2. P. 201 – 211.
87. Hanan M. On Steiner's problem with rectilinear distance // SIAM Journal of Applied Mathematics. 1966. V. 14. P. 255-265.
88. <http://www.ece.ucsb.edu/~kastner/labyrinth>
89. Terlaky T., Vannelli A., Zhang H. On routing in VLSI design and communication networks // X. Deng and D. Du (eds.): ISAAC 2005. LNCS 3827. 2005. P. 1051-1060.

Приложение А. Список публикаций исполнителей

Опубликованные статьи:

1. Гимади Э.Х., Ивонина Е.В. Приближенные алгоритмы решения задачи о двух коммивояжерах на максимум // Дискретный анализ и исследование операций. Январь-февраль. 2012. Т. 19, № 1. С. 17–32.
2. Гимади Э.Х., Шахшнейдер А.В. Приближенные алгоритмы с оценками для задач маршрутизации на случайных входах с ограниченным числом клиентов в каждом маршруте // Автоматика и телемеханика. 2012. № 2, С. 126–140.
3. Edward Gimadi. Application of Petrov's probability inequalities to the justification of performance guarantees of approximation algorithms for solving some difficult problems of discrete optimization on random inputs // Proc. of 4th International Interdisciplinary Symposium on Chaos and Complex Systems (Chaos-2012), Antalya, Turkey, April 29–May 2, 2012, P. 27.
4. Edward Gimadi. On effective algorithms with performance guarantees for some hard-to-solve routing problems // Abstracts of the 25th Conference of the European Chapter on Combinatorial Optimization (ECCO-2012), Antalya, Turkey, April 24–28, 2012, P. 14.
5. Береснев В.Л. Алгоритмы локального поиска для задачи конкурентного размещения предприятий // Автоматика и телемеханика. 2012 № 3 С. 12–27.
6. Кононов А.В. О цеховой задаче открытого типа на двух машинах с маршрутизацией в двухвершинной сети // Дискретный анализ и исследование операций. – 2012, Т.19, № 2. – С. 54-74.
7. A.V. Kel'manov, S.M. Romanchenko. An Approximation Algorithm for Solving a Problem of Search for a Vector Subset // Journal of Applied and Industrial Mathematics. 2012. Vol. 6, No.1, pp. 90-96.
8. А.В. Кельманов, С.М. Романченко. Псевдополиномиальные алгоритмы для некоторых труднорешаемых задач поиска подмножества векторов и кластерного анализа // Автоматика и телемеханика. 2012. № 2, С. 156-162.
9. Астраков С.Н., Ерзин А.И. Построение эффективных моделей покрытия при мониторинге протяженных объектов // Вычислительные технологии, 2012. Т. 17, № 1, С. 26-34.
10. С.В. Августинovich Ю.Л. Васильев, К.Л. Рычков. Формульная сложность тернарной линейной функции // Дискретный анализ и исследование операций. 2012. Т.19, № 3. С. 3–12.

11. Васильева А. Ю. О реконструктивных подмножествах вершин в булевом кубе // Дискретный анализ и исследование операций. 2012. Т.19, № 1. С. 3–16.
12. Коломеец Н. А. Построение бент-функций на минимальном расстоянии от квадратичной бент-функции // Дискретный анализ и исследование операций. 2012. Т.19, № 1. С. 41–58.
13. Потапов В. Н. Построение гамильтоновых циклов с заданным спектром направлений рёбер в булевом n-мерном кубе // Дискретный анализ и исследование операций. 2012. Т.19, № 2. С. 75–83.

Статьи, принятые к печати:

1. Alexander Kononov, Sergey Sevastyanov, A complete 4-parametric complexity classification of short shop scheduling problems // Journal of Scheduling, 2011; DOI: 10.1007/s10951-011-0243-z.
2. A. Kononov. Quantity-based buffer-constrained two machine flowshop problem: active and passive prefetch models for multimedia applications // Journal of Scheduling, 2011, DOI: 10.1007/s10951-011-0235-z. (совместно с J-S. Hong, P. Kononova, F-C. Lin);
3. A. Kononov, S. Sevastyanov. Efficient approximation algorithms for the routing open shop problem // Computers and Operations Research, 2012; DOI: 10.1016/j.cor.2012.01.006 (совместно с I. Chernykh).
4. Кельманов А.В., Романченко С.М., Хамидуллин С.А. Приближённые алгоритмы для некоторых труднорешаемых задач поиска подпоследовательности векторов // Дискретный анализ и исследование операций. 2011 (принята в печать).
5. Кельманов А.В., Пяткин А.В. О сложности некоторых задач выбора подпоследовательности векторов // Журнал вычислительной математики и математической физики. 2012.
6. Кельманов А.В., Романченко С.М., Хамидуллин С.А. Точные псевдополиномиальные алгоритмы для некоторых труднорешаемых задач поиска подпоследовательности векторов // Журнал вычислительной математики и математической физики. 2012. (на рецензировании).
7. Ерзин А.И. Сенсорные сети и покрытие плоских областей эллипсами // 5 Всерос. конф. «Проблемы оптимизации и экономические приложения». Омск. 2-6 июля 2012 (принята в печать).
8. Ерзин А.И., Плотников Р.В., Шамардин Ю.В. Об одной задаче построения коммуникационного остоного дерева // 5 Всерос. конф. «Проблемы оптимизации и экономические приложения». Омск. 2-6 июля 2012 (принята в печать).

9. Ерзин А.И., Плотников Р.В., Шамардин Ю.В. Одна задача построения коммуникационного остоного дерева // Дискретный анализ и исследование операций (сдана в печать, получена положительная рецензия).
10. Тахонов И.И. Поиск наименее плотного регулярного покрытия плоскости кругами различных радиусов // 5 Всерос. конф. «Проблемы оптимизации и экономические приложения». Омск. 2-6 июля 2012 (принята в печать).
11. Воробьев К. В. Кратные совершенные коды в гиперкубе. гиперкубе // Дискретный анализ и исследование операций, 2012. Т.19, принято к печати в № 4.

Статьи, сданные в печать:

1. Гимади Э.Х., Курочкин А.А. Эффективный алгоритм решения двухэтапной задачи размещения на древовидной сети // Дискретный анализ и исследование операций. 2012 (сдана в печать).
2. Gimadi E.K., Ivonina Eugeniya. Approximation algorithms for maximum-weight 2-Peripatetic Salesman Problem in complete graph with restricted distances // Discrete Applied Mathematics. Elsevier, 2012 (submitted).
3. Ph. Baptiste, J. Carlier, A. Kononov, M. Queyranne, S. Sevastyanov, and M. Sviridenko, Integrality Property in Preemptive Parallel Machine Scheduling // Operations Research Letters, submitted.
4. Sergey Sevastyanov, Darya Chemisova, Ilya Chernykh, On some properties of optimal schedules in the job shop problem with preemption and an arbitrary regular criterion // Annals of Operations Research, submitted.
5. S. Sevastyanov and B.M.T. Lin, Efficient enumeration of optimal and approximate solutions for the two-machine flow-shop problem // *Naval Research Logistics*, submitted.
6. Ерзин А.И. Беспроводные сенсорные сети и наименее плотные покрытия плоских областей эллипсами // 9 Межд. конф. «Интеллектуализация обработки информации». Черногория, г. Будва. 16-22 сентября 2012 (сдана в печать).

Приложение Б. Список сделанных исполнителями докладов

1. Edward Gimadi. Polynomial algorithms with performance guarantees for some discrete hard-to-solve routing problems. XXV международная конференция по комбинаторной оптимизации (ЕССО-2012), Анталия, Турция, 24-28 апреля 2012 (секционный доклад).
2. Edward Gimadi. Application of Petrov's probability inequalities to the justification of performance guarantees of approximation algorithms for solving some difficult problems of discrete optimization on random inputs. 4th International Interdisciplinary Symposium on Chaos and Complex Systems (Chaos-2012), Анталия, Турция, 29 апреля – 2 мая 2012 (секционный доклад).
3. Sergey Sevastyanov, Bounds on the optimum makespan in open shops // 13th International Workshop on Project Management and Scheduling (PMS 2012), Leuven, Belgium, April 1-4, 2012 (совместно с И.Д. Черных); секционный;
4. Alexander Kononov, On routing open shop problem // 13th International Workshop on Project Management and Scheduling (PMS 2012), Leuven, Belgium, April 1-4, 2012; секционный.
5. Хандеев В.И. Приближенный полиномиальный алгоритм для решения одной задачи кластерного анализа // МНСК-2012, апрель 22-25, секционный доклад.
6. Erzin A.I., Zainutdinov R.T. Sensor Networks and Min-Density Covering of the Plane Regions by Ellipses // 25th Conference of European Chapter on Combinatorial Optimization. Antalya, Turkey, April 26 – 28, 2012 (секционный)
7. Августинovich С.В., Горкунов Е.В. Метрические инварианты для восстановления кодов // Международная (43-я Всероссийская) молодежная школа-конференция «Современные проблемы математики», 29 января - 5 февраля 2012 г., Екатеринбург, Россия (секционный).
8. Валюженич А.А. Комбинаторная сложность перестановок, порожденных неподвижными точками кодов // Международная (43-я Всероссийская) молодежная школа-конференция «Современные проблемы математики», 29 января - 5 февраля 2012 г., Екатеринбург, Россия (секционный).
9. Воробьев К.В. О сильно регулярных системах троек // Международная (43-я Всероссийская) молодежная школа-конференция «Современные проблемы математики», 29 января - 5 февраля 2012 г., Екатеринбург, Россия (секционный).
10. Могильных И. Ю. Полностью регулярные коды в графах Джонсона и блок-схемы троек // Международная (43-я Всероссийская) молодежная школа-конференция

«Современные проблемы математики», 29 января - 5 февраля 2012 г., Екатеринбург, Россия (секционный).