



Российская Академия Наук

А Т АВТОМАТИКА И ТЕЛЕМЕХАНИКА

Журнал основан в 1936 году

Выходит 12 раз в год

10

октябрь

Москва

2021

Учредители журнала:

Отделение энергетики, машиностроения, механики и процессов управления РАН,
Институт проблем управления им. В.А. Трапезникова РАН (ИПУ РАН),
Институт проблем передачи информации им. А.А. Харкевича РАН (ИППИ РАН)

Главный редактор:

Галяев А.А.

Заместители главного редактора:

Соболевский А.Н., Рубинович Е.Я., Хлебников М.В.

Ответственный секретарь:

Родионов И.В.

Редакционный совет:

Васильев С.Н., Желтов С.Ю., Каляев И.А., Кулешов А.П., Куржанский А.Б.,
Мартынюк А.А. (Украина), Пешехонов В.Г., Поляк Б.Т., Попков Ю.С.,
Рутковский В.Ю., Федосов Е.А., Черноусько Ф.Л.

Редакционная коллегия:

Алескеров Ф.Т., Бахтадзе Н.Н., Бобцов А.А., Виноградов Д.В., Вишневский В.М.,
Воронцов К.В., Глумов В.М., Граничин О.Н., Губко М.В., Каравай М.Ф.,
Кибзун А.И., Краснова С.А., Красносельский А.М., Крищенко А.П.,
Кузнецов Н.В., Кузнецов О.П., Кушнер А.Г., Лазарев А.А., Ляхов А.И.,
Маликов А.И., Матасов А.И., Меерков С.М. (США), Миллер Б.М.,
Михальский А.И., Мунасыпов Р.А., Назин А.В., Немировский А.С. (США),
Новиков Д.А., Олейников А.Я., Пакшин П.В., Пальчунов Д.Е.,
Поляков А.Е. (Франция), Рапопорт Л.Б., Рублев И.В., Степанов О.А.,
Уткин В.И. (США), Фрадков А.Л., Хрусталеv М.М., Цыбаков А.Б. (Франция),
Чеботарев П.Ю., Щербаков П.С.

Адрес редакции: 117997, Москва, Профсоюзная ул., 65

Тел./факс: (495) 334-87-70

Электронная почта: redacsia@ipu.ru

Зав. редакцией *Е.А. Мартехина*

Москва

ООО «Объединённая редакция»

**ВСТУПИТЕЛЬНОЕ СЛОВО ПРОГРАММНОГО
КОМИТЕТА КОНФЕРЕНЦИИ «ИНТЕЛЛЕКТУАЛИЗАЦИЯ
ОБРАБОТКИ ИНФОРМАЦИИ» (ИОИ)**

DOI: 10.31857/S0005231021100019

В специальном выпуске представлены избранные статьи 13-й Международной конференции «Интеллектуализация обработки информации» (ИОИ), прошедшей с 8 по 11 декабря 2020 г.

Конференция ИОИ является форумом исследователей и профессионалов, работающих в области интеллектуального анализа данных, площадкой для обсуждения, распространения и продвижения передовых идей, достижений и разработок. Конференция призвана способствовать обмену идеями между представителями науки и индустрии. Конференция организована представителями российской научной школы машинного обучения и нацелена на расширение взаимодействия между российскими и зарубежными исследователями и представителями высокотехнологичного IT бизнеса.

Впервые конференция ИОИ прошла в 1989 г., а с 2000 г. проводится регулярно один раз в два года. В 2020 г. конференция прошла в режиме онлайн. За эти годы область исследований претерпела существенные изменения. Завершилась очередная «зима искусственного интеллекта». Проблематика «распознавания образов», постепенно расширяясь, перешла в «машинное обучение». Под влиянием экспоненциально растущих объемов данных и стремительно расширяющейся сферы приложений появлялись новые практические направления — «интеллектуальный анализ данных» (data mining, business-intelligence) и «аналитика больших данных» (big data analytics). С развитием глубоких нейронных сетей термин «искусственный интеллект» (ИИ) снова вошел в употребление и перестал ассоциироваться с несбыточными мечтами. Машинное обучение осознается теперь как магистральное направление ИИ и как часть, едва ли не равная целому. Появился зонтичный термин «науки о данных» (data science), отражающий возросший масштаб данной области исследований, претендующей теперь на право называться отдельной наукой. Современные тенденции регулярно находили отражение в тематике конференции ИОИ. Традицией конференции остается сбалансированное сочетание теоретических математических исследований, инженерных разработок и решения практических задач интеллектуального анализа данных в различных предметных областях.

В 2020 г. в конференции ИОИ приняли участие 243 человека, было сделано 115 докладов по научным направлениям:

- анализ биомедицинских данных, биоинформатика;
- аналитика больших данных;

- вычислительная сложность и приближенные методы;
- индустриальные приложения науки о данных;
- интеллектуальная оптимизация и эффективный менеджмент;
- интеллектуальный анализ геопространственных данных;
- интеллектуальный анализ данных в задачах информационной безопасности;
- информационный поиск и анализ текстов;
- машинное обучение;
- методы математического моделирования в интеллектуальном анализе данных;
- методы оптимизации для интеллектуального анализа данных;
- нейронные сети и глубокое обучение;
- обработка и анализ изображений и сигналов, компьютерное зрение.

Для публикации в специальном выпуске журнала «Автоматика и телемеханика» программный комитет конференции отобрал работы, соответствующие тематике и требованиям журнала.

По результатам обсуждения сделанных докладов были отобраны работы для публикации в АиТ № 10 и № 11.

В настоящем 10-м номере журнала представлены работы, посвященные: исследованию стратегий комбинирования допустимых решений трехиндексной аксиальной задачи о назначениях, которые могут применяться в качестве дополнения к эвристическим или приближенным алгоритмам решения вместо общепринятого шага выбора рекорда среди найденных допустимых решений (Л.Г. Афраймович, М.Л. Емелин); задаче анализа данных с элементами из декартового произведения конечных частично упорядоченных множеств (И.Е. Генрихов, Е.В. Дюкова); рассмотрению знание-ориентированных моделей построения маршрутов в сложных сетях агентами-коммивояжерами, представимых в виде псевдобулевых оптимизационных моделей с сепарабельными целевыми функциями и ограничениями в виде дизъюнктивных нормальных форм (М.С. Германчук, М.Г. Козлова, В.А. Лукьяненко); изучению метода обнаружения и идентификации аномальных эффектов в сигнале сложной структуры, основанного на нелинейных аппроксимирующих схемах в словаре вейвлет-пакетов (В.В. Гепшенер, Б.С. Мандрикова); рассмотрению подходов к решению выпуклых задач $\min\text{-min}$ с гладкостью и сильной выпуклостью только по одной из двух групп переменных (Е.Л. Гладин, М. Алкуса, А.В. Гасников); рассмотрению трех новых подходов к решению задач теории расписаний: метрического, интерполяционного и комбинированного (А.А. Лазарев, Д.В. Лемтюжникова, А.А. Тюняткин); исследованию задачи оптимизации работы системы массового обслуживания, в которой число рабочих каналов обслуживания может управляемо изменяться в моменты контроля, отстоящие друг от друга на фиксированный временной шаг (В.А. Лапин, А.С. Мандель); исследованию задачи визуальной одометрии по последовательности видеок кадров, которые формируются с использованием направленной перпендикулярно вниз камеры (В.А. Фурсов, Е.Ю. Минаев,

А.П. Котов); изучению нового подхода к построению алгоритмов кластеризации k-means, в котором вместо евклидова расстояния используется расстояние Мехаланобиса (З.М. Шибзухов); исследованию легковесной шумоподавляющей фильтрующей нейронной сети, которая реализует этап фильтрации в алгоритме томографической реконструкции свертки и обратного проецирования (А.В. Ямаев, М.В. Чукалина, Д.И. Николаев, А.В. Шешкус, А.И. Чуличков).

Каждая рукопись прошла слепое рецензирование как минимум двумя рецензентами, была одобрена к публикации программным комитетом конференции и редколлекцией журнала.

10 июля 2021 года после подписания номера в печать ушел из жизни академик РАН К.В. Рудаков. Вклад Константина Владимировича в развитие области интеллектуального анализа данных трудно переоценить. Ставшие классическими его собственные научные результаты, десятки учеников, успешные прикладные проекты, проведенные международные и российские конференции, руководство кафедрой, участие в редколлекциях научных журналах и экспертных советах — вот краткое и далеко неполное перечисление его вклада. К.В. Рудаков был одним из инициаторов и неизменным вдохновителем конференции ИОИ, проводимой с 1989 года. Программный комитет конференции ИОИ выражает глубокое соболезнование родным, близким и коллегам Константина Владимировича.

*К.В. Воронцов, Ю.И. Журавлев, А.А. Лазарев, Д.В. Лемтюжникова,
К.В. Рудаков, В.В. Стрижов, Юл.В. Чехович, Ю.В. Чехович*

© 2021 г. Л.Г. АФРАЙМОВИЧ, д-р физ.-мат. наук (levafrainovich@gmail.com),
М.Д. ЕМЕЛИН (maksim888e@mail.ru)
(Нижегородский государственный университет)

ЭВРИСТИЧЕСКИЕ СТРАТЕГИИ КОМБИНИРОВАНИЯ РЕШЕНИЙ ТРЕХИНДЕКСНОЙ АКСИАЛЬНОЙ ЗАДАЧИ О НАЗНАЧЕНИЯХ

Рассматривается NP-трудная целочисленная трехиндексная аксиальная задача о назначениях. Исследуются стратегии комбинирования допустимых решений задачи. Такое комбинирование может быть применено в качестве дополнения к эвристическим или приближенным алгоритмам решения вместо общепринятого шага выбора рекорда среди найденных допустимых решений. Приводятся результаты вычислительных экспериментов, демонстрирующие перспективность предложенного подхода.

Ключевые слова: аксиальная задача о назначениях, многоиндексная задача, приближенные алгоритмы.

DOI: 10.31857/S0005231021100020

1. Введение

Многоиндексные аксиальные задачи о назначениях возникают при решении множества прикладных задач [1–3]. Обзор результатов анализа подклассов многоиндексных задач о назначениях приведен в [1]. Класс трехиндексных аксиальных задач о назначениях является NP-трудным [4]. В [5] доказано отсутствие полиномиальных ε -приближенных алгоритмов решения трехиндексной аксиальной задачи о назначениях (здесь ε – произвольная константа), в противном случае $P = NP$.

Известны приближенные и эвристические алгоритмы решения NP-трудной аксиальной задачи о назначениях [2, 5–10]. Такие алгоритмы, как правило, позволяют строить серию допустимых решений задачи. Далее на финальном шаге таких алгоритмов общепринятым подходом является выбор рекорда среди построенных допустимых решений. В качестве улучшения финального шага выбора рекорда Л.Г. Афраймович и М.Д. Емелин предлагают решение задачи оптимального комбинирования найденных допустимых решений. Ранее в [11] был построен алгоритм оптимального комбинирования пары допустимых решений, обладающий линейной сложностью. В данной статье исследуется проблема комбинирования в случае произвольного числа решений.

Далее статья построена следующим образом. В разделе 2 приводится постановка трехиндексной аксиальной задачи о назначениях и ставится задача комбинирования допустимых решений. В разделе 3 исследуется задача комбинирования допустимых решений и описываются эвристические стратегии комбинирования. В разделе 4 приводятся результаты вычислительных экспериментов.

2. Постановка задачи

Пусть I, J, K – непересекающиеся множества индексов, $I \cap J = \emptyset, I \cap K = \emptyset, J \cap K = \emptyset$ и $|I| = |J| = |K| = n$; $c_{ijk}, i \in I, j \in J, k \in K$ – трехиндексная матрица стоимостей; $x_{ijk}, i \in I, j \in J, k \in K$ – трехиндексная матрица неизвестных. Тогда трехиндексная аксиальная задача о назначениях ставится как следующая задача целочисленного линейного программирования:

$$(1) \quad \sum_{i \in I} \sum_{j \in J} x_{ijk} = 1, \quad k \in K,$$

$$(2) \quad \sum_{i \in I} \sum_{k \in K} x_{ijk} = 1, \quad j \in J,$$

$$(3) \quad \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad i \in I,$$

$$(4) \quad x_{ijk} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad k \in K,$$

$$(5) \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \rightarrow \min.$$

Далее пусть задано множество $W \subseteq I \times J \times K$, которое определяет подмножество разрешенных назначений. Тогда рассмотрим задачу (1)–(4), (6), (5)

$$(6) \quad x_{ijk} = 0, \quad (i, j, k) \notin W.$$

Для удобства изложения задачу (1)–(4), (6), (5) для заданного множества W будем обозначать через $Z(W)$. Очевидно, задача (1)–(5) соответствует задаче $Z(I \times J \times K)$.

В общем случае задача $Z(W)$ является NP-трудной [1]. Более того, проблема проверки совместности системы (1)–(4), (6) для произвольного множества W является NP-полной [1]. Будем рассматривать такие множества W , которые соответствуют набору назначений некоторых допустимых решений задачи (1)–(5).

Введем вспомогательные обозначения. Пусть x – допустимое решение системы ограничений (1)–(4). Тогда через $W(x)$ обозначим следующее множество разрешенных назначений:

$$W(x) = \{(i, j, k) \mid x_{ijk} = 1, i \in I, j \in J, k \in K\}.$$

Для удобства через $C(x)$ обозначим значение критерия, соответствующего решению x , $C(x) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk}$. Пусть x^1, x^2, \dots, x^m – произвольные допустимые решения системы ограничений (1)–(4). Тогда обозначим $W(x^1, x^2, \dots, x^m) = \bigcup_{t=1}^m W(x^t)$. Через S_m обозначим множество перестановок размера m .

Пусть x^1, x^2, \dots, x^m – некоторые допустимые решения системы ограничений (1)–(4), найденные известными приближенными или эвристическими

методами решения аксиальной задачи о назначениях (1)–(5). Общепринятым подходом является выбор рекорда среди данных допустимых решений на финальном шаге алгоритма: $C' = \min_{t=\overline{1,m}} C(x^t)$. В качестве улучшения финального шага выбора рекорда предлагается решение задачи $Z(W(x^1, x^2, \dots, x^m))$, т.е. выбор рекорда предлагается заменить на поиск решения, скомбинированного из компонент известных допустимых решений.

3. Исследования задачи комбинирования решений

Для случая $m = 2$ в [11] был разработан полиномиальный алгоритм решения задачи $Z(W(x^1, x^2, \dots, x^m))$. Вопрос построения алгоритма для случая $m > 2$ является открытым.

Покажем, что решение задачи комбинирования $m > 2$ допустимых решений $Z(W(x^1, x^2, \dots, x^m))$ в общем случае не может быть сведено к последовательному комбинированию пар допустимых решений. В качестве базового алгоритма последовательного комбинирования пар допустимых решений рассмотрим следующий алгоритм.

Алгоритм 1. Последовательное комбинирование пар допустимых решений.

Вход: Исходные допустимые решения x^1, x^2, \dots, x^m и перестановка $p \in S_m$.

Шаг 1. y^1 является решением задачи $Z(W(x^{p_1}, x^{p_2}))$.

Шаг 2. y^t является решением задачи $Z(W(y^{t-1}, x^{p_{t+1}}))$, $t = \overline{2, m-1}$.

Выход: y^{m-1} .

Замечание. На шагах 1 и 2 алгоритма 1 применяется алгоритм оптимального комбинирования пар решений (случай $m = 2$), предложенный в [11].

Теорема. Алгоритм 1 не является алгоритмом решения задачи $Z(W(x^1, x^2, \dots, x^m))$ при $m > 2$.

Доказательство. Для доказательства данной теоремы приведем контрпример. Пусть $n = 3$, $m = 3$,

$$c = \begin{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} & \begin{bmatrix} 2 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} & \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 1 \end{bmatrix} \end{bmatrix},$$

$$x^1 = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{bmatrix},$$

$$x^2 = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{bmatrix},$$

$$x^3 = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}.$$

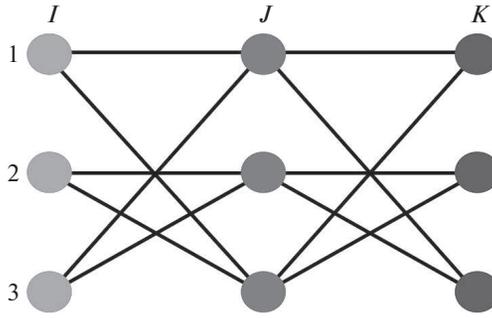


Рисунок.

Несложно увидеть, что граф $G = (V, A)$, где $V = \{I \cup J \cup K\}$, $A = \{(i, j), (i, k), (j, k) \mid (i, j, k) \in W(x^{l_1}, x^{l_2})\}$, $l_1 \neq l_2$, $l_1, l_2 \in \{1, 2, 3\}$, имеет одну компоненту связности (на рисунке в качестве примера приведен граф, построенный для $l_1 = 1$, $l_2 = 2$).

Отсюда согласно [11] решением задачи $Z(W(x^{l_1}, x^{l_2}))$ является x^{l_1} или x^{l_2} . Следовательно, при применении алгоритма 1 комбинирования решений x^1 , x^2 , x^3 с произвольной перестановкой $p \in S_3$ получим на выходе одно из этих решений. При этом оптимальным решением задачи $Z(W(x^1, x^2, x^3))$ является

$$x^* = \left[\begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \right],$$

в котором тройка $(1, 1, 1)$ выбрана из решения x^1 , тройка $(2, 2, 2)$ из x^2 , тройка $(3, 3, 3)$ из x^3 . И выполняется условие $C(x^1) = C(x^2) = C(x^3) = 5 > 3 = C(x^*)$. Теорема доказана.

По причине отсутствия на данный момент известного эффективного алгоритма решения задачи $Z(W(x^1, x^2, \dots, x^m))$ при $m > 2$ в данной статье предлагается ряд эвристических стратегий решения данной задачи, основанных на последовательном комбинировании пар решений. Согласно теореме данные стратегии не гарантируют построение оптимального решения, однако на численных задачах предложенные стратегии показывают улучшение результатов по сравнению с общепринятым выбором рекорда.

Предлагаются следующие эвристические стратегии комбинирования решений:

Стратегия 1.

Применить алгоритм 1 к решениям x^1, x^2, \dots, x^m со случайной перестановкой $p \in S_m$. Полученное алгоритмом 1 решение выбрать в качестве выходного.

Стратегия 2.

Применить алгоритм 1 к решениям x^1, x^2, \dots, x^m с перестановкой $p \in S_m$, удовлетворяющей свойству $C(x^{p_i}) \leq C(x^{p_{i+1}})$, $i = \overline{1, m-1}$, т.е. решения упорядочены в порядке неубывания критерия. Полученное алгоритмом 1 решение выбрать в качестве выходного.

Стратегия 3.

Применить алгоритм 1 к решениям x^1, x^2, \dots, x^m с перестановкой $p \in S_m$, удовлетворяющей свойству $C(x^{p_i}) \leq C(x^{p_{i+1}})$, $i = \overline{1, m-1}$. Обозначим полученное алгоритмом 1 решение через y_1 .

Получить k решений y_t , $t = \overline{2, k+1}$, следующим способом. Для каждого $t = \overline{2, k+1}$ создать перестановку $p \in S_m$, удовлетворяющую свойству $C(x^{p_i}) \leq C(x^{p_{i+1}})$, $i = \overline{1, m-1}$, выбрать случайным образом $d = \lfloor \frac{m}{2} \rfloor$ элементов перестановки и случайно поменять их местами. Применить алгоритм 1 к решениям x^1, x^2, \dots, x^m с перестановкой $p \in S_m$. Полученное решение обозначим через y_t .

Применить алгоритм 1 к решениям y_1, y_2, \dots, y_{k+1} с тождественной перестановкой $e = (1, 2, \dots, k+1)$. Полученное алгоритмом 1 решение выбрать в качестве выходного.

4. Вычислительный эксперимент

По аналогии с [12] построим тестовый набор с матрицами стоимостей, элементы которых сгенерированы с целочисленными значениями, равномерно распределенными в интервале $[0, 300]$. Будем строить серии экспериментов с задачами размерности $n \in \{10, 11, \dots, 19\}$, в каждой серии построим $M = 10$ задач. Для каждой из тестовых задач случайным образом сгенерируем $m = n^3$ допустимых решений, к каждому из которых применим алгоритм локальной оптимизации, предложенный в [6]. Полученные допустимые решения задачи (1)–(5) обозначим через x^t , $t = \overline{1, m}$. Тогда рекорд среди полученных решений обозначим $C' = \min_{t=\overline{1, m}} C(x^t)$. Далее применим разработанные стратегии оптимального комбинирования решений к полученным локально оптимальным решениям и обозначим значения критерия, соответствующего решению, полученному с использованием стратегии s , через C^s , $s = \overline{1, 3}$. Наконец, через C^* обозначим оптимальное значение критерия исходной задачи. Будем сравнивать отклонение от оптимума рекорда среди локально оптимизированных случайных решений и отклонение от оптимума среди локально

Таблица

n	M	$\frac{C' - C^*}{C^*} \cdot 100\%$	$\frac{C^1 - C^*}{C^*} \cdot 100\%$	$\frac{C^2 - C^*}{C^*} \cdot 100\%$	$\frac{C^3 - C^*}{C^*} \cdot 100\%$
10	10	3,195	2,399	2,399	1,478
11	10	8,883	7,413	6,332	4,274
12	10	15,054	14,185	14,185	10,923
13	10	24,319	24,319	24,319	22,010
14	10	41,210	34,207	34,387	23,276
15	10	54,856	47,248	52,372	46,553
16	10	72,872	69,792	70,677	69,473
17	10	68,145	68,145	59,684	51,387
18	10	87,201	81,402	82,272	66,769
19	10	100,464	88,211	90,114	81,358

оптимальных случайных решений, к которым применили соответствующие разработанные стратегии комбинирования. Для серии экспериментов будем оценивать среднее отклонение в серии. Полученные результаты приведены в таблице.

Таким образом, среднее отклонение для C' по всем сериям составляет 47,620 %, для C^1 составляет 43,732 %, для C^2 составляет 43,674 %, для C^3 составляет 37,750 %.

5. Заключение

В литературе известны приближенные и эвристические алгоритмы решения NP-трудной аксиальной задачи о назначениях. Такие алгоритмы, как правило, позволяют строить серию допустимых решений задачи. На финальном шаге таких алгоритмов общепринятым является выбор рекорда среди построенных допустимых решений. В качестве улучшения финального шага выбора рекорда Л.Г. Афраймович и М.Д. Емелин предложили решение задачи оптимального комбинирования найденных m допустимых решений. Для случая $m = 2$ ранее был разработан эффективный алгоритм решения. В данной статье было показано, что решение задачи комбинирования при $m > 2$ в общем случае не сводится к последовательному комбинированию пар допустимых решений. Тем не менее предложенные эвристические стратегии последовательного комбинирования пар решений для случая $m > 2$ позволяют получить улучшение результатов по сравнению с общепринятой стратегией выбора рекорда. Построение эффективного алгоритма оптимального комбинирования при $m > 2$ остается открытой проблемой.

СПИСОК ЛИТЕРАТУРЫ

1. *Spieksma F.C.R.* Multi Index Assignment Problems. Complexity, Approximation, Applications / P.M. Pardalos, L.S. Pitsoulis (Eds.). Nonlinear Assignment Problems: Algorithms and Applications. Dordrecht: Kluwer Acad. Publishers, 2000. P. 1–11.
2. *Афраймович Л.Г.* Эвристический метод решения целочисленных декомпозиционных многоиндексных задач // *АиТ.* 2014. № 8. С. 3–18.
Afraimovich L.G. A Heuristic Method for Solving Integer-Valued Decompositional Multiindex Problems // *Autom. Remote Control.* 2014. V. 75. No. 8. P. 1357–1368.
3. *Афраймович Л.Г., Прилуцкий М.Х.* Многоиндексные задачи оптимального планирования производства // *АиТ.* 2010. № 10. С. 148–155.
Afraimovich L.G., Prilutskii M.Kh. Multiindex Optimal Production Planning Problems // *Autom. Remote Control.* 2010. V. 71. No. 10. P. 2145–2151.
4. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
5. *Crama Y., Spieksma F.C.R.* Approximation Algorithms for Three-Dimensional Assignment Problems with Triangle Inequalities // *Eur. J. Oper. Res.* 1992. V. 60. P. 273–279.
6. *Huang G., Lim A.* A Hybrid Genetic Algorithm for the Three-Index Assignment Problem // *Eur. J. Oper. Res.* 2006. V. 172. P. 249–257.

7. *Karapetyan D., Gutin D.* A New Approach to Population Sizing for Memetic Algorithms: A Case Study for the Multidimensional Assignment Problem // *Evolutionary Computation*. 2011. V. 19. No. 3. P. 345–371.
8. *Медведев С.Н., Медведева О.А.* Адаптивный алгоритм решения аксиальной трехиндексной задачи о назначениях // *АиТ*. 2019. № 4. С. 156–172.
Medvedev S.N., Medvedeva O.A. An Adaptive Algorithm for Solving the Axial Three-Index Assignment Problem // *Autom. Remote Control*. 2019. V. 80. No. 4. P. 718–732.
9. *Gabrovšek B., Novak T., Povh J., Rupnik Poklukar D., Žerovnik J.* Multiple Hungarian Method for k-Assignment Problem // *Mathematics*. 2020. V. 8. 2050.
10. *Гимади Э.Х., Коржишко Н.М.* Об одном алгоритме решения трехиндексной аксиальной задачи о назначениях на одноциклических подстановках // *Дискретный анализ и исследование операций*. Сер. 1. 2003. Т. 10. № 2. С. 56–65.
11. *Афраймович Л.Г., Емелин М.Д.* Комбинирование решений аксиальной задачи о назначениях // *АиТ*. 2021. (принято к печати)
12. *Balas E., Saltzman M.J.* An Algorithm for the Three-Index Assignment Problem // *Oper. Res*. 1991. V. 39. No. 1. P. 150–161.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 20.01.2021

После доработки 01.06.2021

Принята к публикации 30.06.2021

© 2021 г. И.Е. ГЕНРИХОВ, канд. физ.-мат. наук (ingvar1485@rambler.ru)
(ООО “Мобайл парк ИТ”, Химки),
Е.В. ДЮКОВА, д-р физ.-мат. наук (edjukova@mail.ru)
(ФИЦ “Информатика и управление” РАН, Москва)

ПОИСК ЧАСТЫХ ЭЛЕМЕНТОВ ПРОИЗВЕДЕНИЯ ЧАСТИЧНЫХ ПОРЯДКОВ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ¹

Рассматриваются вопросы анализа данных с элементами из декартового произведения конечных частично упорядоченных множеств. Для эффективного поиска частых элементов, порождаемых всеми возможными вариантами бинаризации исходных небинарных данных, используется модификация классического FP-дерева (Frequent Pattern Tree). Сокращение временных затрат достигается за счет использования параллельных вычислений на основе технологии CUDA (Compute Unified Device Architecture). Приводятся результаты тестирования построенных параллельных процедур синтеза искомых частых элементов на модельных и реальных данных.

Ключевые слова: декартовое произведение частичных порядков, база данных, частый элемент, FP-дерево, пороговое FP-дерево, параллельные вычисления, технология CUDA.

DOI: 10.31857/S0005231021100032

1. Введение

Задача поиска частых элементов в данных востребована для ряда прикладных областей, среди которых следует выделить нахождение ассоциативных правил в базах данных и машинное обучение. Приведем классическую постановку рассматриваемой задачи для наиболее исследованного случая, а именно для случая бинарных данных [1].

Дано некоторое множество P , элементы которого называются атрибутами, и дана некоторая совокупность D подмножеств множества P (не обязательно различных), называемая базой данных. Подмножества множества P называются наборами атрибутов, а те из них, которые содержатся в D , называются транзакциями. Набор атрибутов Z называется s -частым, если отношение числа транзакций, содержащих Z , к числу всех транзакций не менее s . Требуется найти все s -частые наборы атрибутов.

Среди алгоритмов поиска частых наборов атрибутов в бинарных данных наиболее известными являются алгоритмы Apriori, Eclat, FP-Growth, SaM, RElim, LCM, DepthProject ([2–9]). Рассматриваемый в настоящей статье алгоритм FP-Growth ([2, 5]) делает предобработку бинарной базы данных, бла-

¹ Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 19-01-00430-а).

годаря которой поиск частых наборов атрибутов сводится к анализу компактной древовидной структуры, называемой Frequent Pattern Tree или FP-деревом. В результате алгоритм FP-Growth имеет в ряде случаев преимущество перед другими алгоритмами, например, позволяет избежать слишком затратной процедуры поиска решений, характерной для алгоритмов Apriori [2] и DepthProject [9].

В случае небинарных данных каждый атрибут имеет некоторое множество числовых значений, вместо наборов атрибутов рассматриваются наборы их значений и, как правило, задача сводится к бинарному случаю путем задания для каждого небинарного атрибута некоторого числа (порога), позволяющего перекодировать исходные небинарные данные в бинарные [10, 11].

На практике возникает необходимость нахождения зависимостей в частично упорядоченных данных. В [12] рассмотрена задача поиска ассоциативных правил в данных, представленных в виде декартового произведения частичных порядков, и в связи с этим введено понятие s -частого элемента для множества $P = P_1 \times \dots \times P_n$ при условии, что P_1, \dots, P_n – конечные частично упорядоченные числовые множества.

На множестве P устанавливается частичный порядок. Считается, что элемент $y = (y_1, \dots, y_n) \in P$ следует за элементом $x = (x_1, \dots, x_n) \in P$, если y_i следует за x_i при $i = 1, \dots, n$ (запись $x \preceq y$). Элементы $x, y \in P$ называются *сравнимыми*, если один из этих элементов следует за другим, иначе x и y называются *несравнимыми*.

Предполагается, что каждое множество (атрибут) P_i содержит *наименьший элемент*, т.е. такой элемент l_i , для которого выполнено $l_i \preceq x_i$ для любого $x_i \in P_i$. Элемент $x_i \in P_i$ называется *существенным* значением элемента $x = (x_1, \dots, x_i, \dots, x_n) \in P$, если $x_i \neq l_i$. Допускается, что база D не содержит $l = (l_1, \dots, l_n)$.

В случае бинарных данных $P_i = \{0, 1\}$, $i = 1, \dots, n$, и в P_i установлен порядок $0 \preceq 1$, $0 \neq 1$ (здесь $l_i = 0$, $i = 1, \dots, n$).

Пусть $x \in P$, $x \neq l$ и $S_D(x)$ – число транзакций y в D таких, что $x \preceq y$. Элемент x называется *s-частым*, если $S_D(x) / |D| \geq s$, иначе x называется *s-нечастым* (здесь и далее $|D|$ – число транзакций в базе D). Заметим, что если $x \preceq y$, то $S_D(x) \geq S_D(y)$. Элемент x называется *максимальным s-частым* элементом, если $S_D(x) / |D| \geq s$ и $S_D(y) / |D| < s$ для любого y , такого что $x \preceq y$, $x \neq y$ (т.е. из условия $x \preceq y$, $x \neq y$ следует, что y – s -нечастый элемент в P).

В [13–15] рассмотрены вопросы бинаризации множества $P = P_1 \times \dots \times P_n$. Для каждого множества P_i , $i \in \{1, \dots, n\}$, строится множество “значимых порогов” $Q_i \subseteq P_i$. Каждый набор порогов $H = \{p_1, \dots, p_n\}$, в котором $p_i \in Q_i$ при $i = 1, \dots, n$, порождает произведение бинарных частичных порядков P^H . Частые элементы множества P^H названы *пороговыми* частыми элементами. Поставлена задача перечисления максимальных пороговых частых элементов, порождаемых всеми возможными вариантами бинарной перекодировки. Заметим, что находить частые элементы, не являющиеся максимальными,

неэффективно как по времени, так и по памяти. Для решения поставленной задачи предложен подход, основанный на построении порогового FР-дерева (TFР-дерева).

Модель TFР-дерева является модификацией модели классического бинарного FР-дерева [2, 5]. В TFР-дереве для каждого небинарного атрибута P_i строится так называемая полная вершина, содержащая информацию о возможных вариантах бинаризации этого атрибута. При спуске из полной вершины строится либо новая полная вершина, либо корневая вершина классического бинарного FР-дерева. Бинарное FР-дерево строится тогда, когда в текущей ветви все значения небинарных атрибутов перекодированы в бинарные значения.

В реальных задачах может формироваться большое число значимых наборов порогов и для поиска максимальных пороговых частых элементов с использованием TFР-дерева требуются существенные вычислительные ресурсы. В настоящей статье с целью ускорения процедуры поиска искомых частых элементов разработаны параллельные алгоритмы на основе технологии CUDA [16, 17], позволяющей выполнять операции, не требующие больших временных затрат на центральном процессоре, а все сложные операции на графическом процессоре (GPU). Реализованы две схемы распараллеливания: блочная и одиночная. В блочной схеме множество всех “значимых” наборов порогов H_D разбивается на непересекающиеся подмножества, каждое из которых подается на отдельный вычислительный блок GPU для синтеза максимальных пороговых частых элементов. При одиночном распараллеливании для нахождения максимальных пороговых частых элементов, порождаемых набором порогов из H_D , используется один вычислительный блок GPU.

В разделе 2 введены основные понятия. В разделе 3 описана процедура построения TFР-дерева. В разделах 4 и 5 дано описание алгоритмов поиска максимальных пороговых частых элементов в TFР-дереве на основе параллельных вычислений с использованием технологии CUDA и приведены результаты тестирования этих алгоритмов на модельных данных и на реальных задачах из репозитория UCI [18].

2. Основные понятия

Рассмотрим случай небинарных данных. Задачу поиска частых элементов в произведении частичных небинарных порядков P сведем к бинарному случаю путем задания набора порогов $H = \{p_1, \dots, p_n\}$, в котором $p_i \in P_i$ при $i = 1, \dots, n$, $p_i \neq l_i$. В каждом элементе $x = (x_1, \dots, x_n)$ множества P заменим значение x_i , $i \in 1, \dots, n$, на 1, если $p_i \preceq x_i$, и заменим на 0 в противном случае. В результате такой кодировки получаем вместо множества P множество $P^H = P_1^H \times \dots \times P_n^H$, где $P_i^H = \{0, 1\}$, $i \in \{1, \dots, n\}$, и в каждом P_i^H установлен порядок $0 \preceq 1$, $0 \neq 1$. Вместо базы данных D получаем базу данных D_H .

Порогу p , $p \in P_i$, поставим в соответствие элемент $\varphi_i(p) = (x_1, \dots, x_n)$ из P , в котором $x_i = p$ и $x_j = l_j$ при $j \neq i$. Тогда порог p называется *значимым*, если $S_D(\varphi_i(p)) \geq s$, т.е. элемент $\varphi_i(p)$ является s -частым в P . Предполагается, что каждое P_i имеет хотя бы один значимый порог.

Набор порогов $H = (p_1, \dots, p_n)$ называется *значимым*, если для любого $i \in \{1, \dots, n\}$ порог p_i – значимый. Для бинарной перекодировки будем выбирать только значимые наборы порогов.

Набор порогов (p_1, \dots, p_n) , в котором для каждого $p_i, i \in \{1, \dots, n\}$, выполняется условие $S_D(\varphi_i(p_i)) = \max_{p \in P_i} S_D(\varphi_i(p))$, называется *оптимальным значимым*. Оптимальный значимый набор порогов позволяет находить наибольшее число пороговых частых элементов.

Ставится задача нахождения всех максимальных пороговых частых элементов, которые порождаются всеми значимыми наборами порогов. В силу большого числа s -частых элементов имеет смысл искать и хранить только те из них, которые являются максимальными.

Рассматриваются три способа решения поставленной задачи. Первый способ основан на последовательном поиске значимых наборов порогов и синтезе для каждого такого набора соответствующего бинарного FP-дерева. Второй способ основан на нахождении искомым элементов в TFP-дереве. Третий способ основан на параллельном поиске частых элементов в TFP-дереве с применением технологии CUDA. Результаты экспериментального сравнения по скорости счета указанных подходов приводятся в разделе 5.

3. Синтез порогового FP-дерева (TFP-дерева) и пороговых частых элементов произведения частичных небинарных порядков

Отметим, что описание варианта FP-дерева для произведения бинарных частичных порядков (далее бинарного FP-дерева) приведено в [14].

Введем обозначения: $a_{ri}, r \in \{1, \dots, m\}, i \in \{1, \dots, n\}$, – значение атрибута P_i для транзакции S_r в базе D ; X – множество небинарных атрибутов в D ; H – набор вида $(1, \dots, 1)$ длины n ; H_D – множество всех значимых наборов порогов. Не ограничивая общности, можно считать, что каждый небинарный атрибут имеет хотя бы один значимый порог.

Опишем алгоритм синтеза TFP-дерева. Данный алгоритм является рекурсивным. Положим на первом шаге $D' = D, \hat{X} = X, \hat{H} = H, H_D = \emptyset$. Далее на каждом шаге рекурсии осуществляется следующая последовательность действий 1–3.

1. Если $\hat{X} = \emptyset$, то осуществляется переход к 3, иначе осуществляется 2.
2. Выбирается атрибут $P_i \in \hat{X}$ с множеством значимых порогов Q_i и создается полная вершина с меткой (P_i, Q_i) . Далее осуществляется ветвление из вершины (P_i, Q_i) по каждому порогу $p \in Q_i$. Для порога p строится одна дуга и создается база D_p путем бинарной перекодировки атрибута P_i в текущей для данной ветви базе D' по правилу: $a_{ri} = 1$, если $p \preceq a_{ri}$, иначе $a_{ri} = 0$. Полагается $D' = D_p, \hat{X} = \hat{X} \setminus \{P_i\}$, в наборе \hat{H} координата с номером i заменяется на p и осуществляется рекурсивный переход к 1.
3. Полагается $H_D = H_D \cup \{\hat{H}\}$, и для набора порогов \hat{H} строится бинарное FP-дерево. Рекурсия останавливается.

После построения TFP-дерева для каждого построенного значимого набора порогов осуществляется поиск максимальных пороговых s -частых элементов.

Пусть $H_1 = \{p_1, \dots, p_n\}$ и $H_2 = \{q_1, \dots, q_n\}$ – два значимых набора порогов таких, что $p_t \neq q_t$ для некоторого $t \in \{1, \dots, n\}$ и $p_i = q_i$, $i \neq t$, $i \in \{1, \dots, n\}$.

Пусть $W(H_1)$ – множество пороговых s -частых элементов в P^{H_1} , $W(H_2)$ – множество пороговых s -частых элементов в P^{H_2} . Имеют место приводимые далее утверждения 1 и 2.

Утверждение 1. Если $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$, то $W(H_2) \subseteq W(H_1)$.

Доказательство. Пусть $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$. Покажем, что $W(H_2) \subseteq W(H_1)$. Возьмем произвольный элемент $x = (x_1, \dots, x_n)$ из $W(H_2)$. Заметим, что наборы H_1 и H_2 отличаются друг от друга по значению только одного порога с индексом t , поэтому база D_{H_1} отличается от базы D_{H_2} только по атрибуту с номером t . Тем самым если $x_t = 0$, то x будет также s -частым элементом и в P^{H_1} по построению базы D_{H_1} . Если же $x_t \neq 0$, то из условия $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$ получаем, что в D_{H_1} содержится не меньше транзакций, чем в D_{H_2} , следующих за элементом x . Тем самым элемент x также принадлежит и множеству $W(H_1)$. Утверждение 1 доказано.

Пусть $W_t(H_1)$ – множество пороговых s -частых элементов в P^{H_1} , в которых значение атрибута с индексом t равно нулю, $W_t(H_2)$ – множество пороговых s -частых элементов в P^{H_2} , в которых значение атрибута с индексом t равно нулю.

Утверждение 2. Если $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$, то $W_t(H_2) = W_t(H_1)$.

Доказательство. Пусть $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$. Покажем, что $W_t(H_2) = W_t(H_1)$. Возьмем произвольный элемент $x = (x_1, \dots, x_n)$ из $W_t(H_1)$. Заметим, что наборы H_1 и H_2 отличаются друг от друга по значению только одного порога с индексом t , поэтому база D_{H_1} отличается от базы D_{H_2} только по атрибуту с номером t . Из $x_t = 0$ и $S_D(\varphi_t(q_t)) \leq S_D(\varphi_t(p_t))$ получаем, что в базе D_{H_2} найдется $s|D_{H_2}|$ транзакций, следующих за элементом x . Тем самым элемент x также принадлежит и множеству $W_t(H_2)$. Аналогично рассматривается обратная ситуация, когда элемент x принадлежит $W_t(H_2)$. Утверждение 2 доказано.

Применяя утверждения 1 и 2, процесс поиска всех максимальных пороговых s -частых элементов с использованием TFP-дерева можно существенно ускорить. С этой целью в каждом Q_i , $i \in \{1, \dots, n\}$, пороги упорядочиваются по убыванию значения величины $S_D(\varphi_i(p))$, $p \in Q_i$, и ветвление из полной вершины (P_i, Q_i) осуществляется в соответствии с выбранным порядком. Множество H_D упорядочивается в порядке построения его элементов. Нетрудно видеть, что в таком случае в H_D первым по порядку будет оптимальный значимый набор порогов H_{opt} (раздел 2).

Пусть $F_{\max}(H)$, $H \in H_D$, – множество максимальных пороговых s -частых элементов для набора порогов H ; $F(H_D)$ – множество пар вида $(F_{\max}(H), H)$, $H \in H_D$. Для поиска требуемых частых элементов применяются процедуры AllMaxSets и MaxSets, результатом работы которых является соответственно построение множества $F(H_D)$ и пары $(F_{\max}(H), H)$, $H \neq H_{opt}$.

На первом шаге процедура AllMaxSets по FFP-дереву, построенному для H_{opt} , классическим способом, находит множество $F_{\max}(H_{opt})$ и пару $(F_{\max}(H_{opt}), H_{opt})$ добавляет в $F(H_D)$ (первоначально $F(H_D) = \emptyset$). Далее процедура AllMaxSets осуществляет построение множества $F(H_D)$ путем последовательного вызова процедуры MaxSets для каждого $H \in H_D$, $H \neq H_{opt}$ (в порядке следования наборов в H_D).

На вход процедуры MaxSets подаются набор порогов $H = \{q_1, \dots, q_n\}$, множество $F_{\max}(H) = \emptyset$ и текущее множество (последовательность) $F(H_D)$. На выходе из MaxSets пара $(F_{\max}(H), H)$ добавляется в $F(H_D)$. В процедуре MaxSets осуществляется следующая последовательность действий 1–3.

1. В последовательности $F(H_D)$ среди пар $(F_{\max}(H'), H')$, $H' \neq H$, $H' = \{p_1, \dots, p_n\}$, $S_D(\varphi_i(q_i)) \leq S_D(\varphi_i(p_i))$, $i \in \{1, \dots, n\}$, ищутся пары с минимальным числом неравенств $S_D(\varphi_i(q_i)) < S_D(\varphi_i(p_i))$. Среди этих пар берется пара $(F_{\max}(H^*), H^*)$ с максимальным номером и создается множество $T(H^*)$ индексов i , для каждого из которых при $H' = H^*$ верно указанное выше строгое неравенство. Далее осуществляется переход к 2.
2. Просматриваются элементы из $F_{\max}(H^*)$. Элемент A из $F_{\max}(H^*)$, в котором значения атрибутов с индексами из $T(H^*)$ равны 0, добавляется в $F_{\max}(H)$, иначе осуществляется переход к 3. После просмотра всех элементов в $F_{\max}(H^*)$ построение множества $F_{\max}(H)$ заканчивается и осуществляется выход из процедуры MaxSets.
3. Пусть $M(A)$ – множество различных элементов из P^H , каждое из которых получается из A заменой одного элемента 1 на элемент 0. Если A – s -частый элемент в D_H и $A \notin F_{\max}(H)$, то элемент A добавляется в $F_{\max}(H)$. Если же A – s -нечастый элемент в D_H и $|A| > 2$, то действие 3 осуществляется для каждого элемента из $M(A)$. Во всех других случаях элемент A не добавляется в $F_{\max}(H)$.

Следует заметить, что в связи с особенностью построения TFP-дерева на первом шаге работы процедуры MaxSets всегда найдется хотя бы одна пара $(F_{\max}(H^*), H^*)$. Дело в том, что спуск из полной вершины осуществляется последовательно начиная с рассмотрения наиболее значимого порога.

4. Поиск пороговых частых элементов на базе TFP-дерева и технологии параллельных вычислений CUDA

В случае больших данных для построения пороговых частых элементов произведения частичных порядков требуются существенные вычислительные ресурсы. В данном разделе с целью уменьшения временных затрат при поиске искомых элементов реализованы параллельные схемы вычислений на основе применения технологии параллельных вычислений CUDA (Compute Unified Device Architecture), которая позволяет использовать графический процессор [16, 17].

Графический процессор (GPU) состоит из однородных вычислительных элементов (мультипроцессоров) с общей памятью. Каждый мультипроцессор способен исполнять параллельно тысячи вычислительных “нитей”. Нити могут быть сгруппированы в вычислительные потоки, имеющие общий кэш и быструю разделяемую память для обмена данными между нитями потока.

Вычислительные потоки также могут быть сгруппированы в вычислительные блоки для лучшего распараллеливания вычислений. Применение вычислений на GPU наиболее эффективно в решении задач, в которых число арифметических операций велико по сравнению с операциями над памятью.

Далее описаны два параллельных алгоритма поиска максимальных пороговых частых элементов произведения частичных порядков: PTFFP-tree и DPTFFP-tree. В алгоритме PTFFP-tree реализована одиночная схема распараллеливания, а в алгоритме DPTFFP-tree применена блочная схема. В качестве базового алгоритма поиска максимальных пороговых частых элементов предлагается алгоритм, описанный в разделе 3. В этом алгоритме наиболее существенное время требуется для работы процедуры AllMaxSets. Поэтому в разработанных параллельных алгоритмах данные вычисления полностью перенесены на GPU.

В алгоритме PTFFP-tree реализована процедура MaxSetsGPU, осуществляющая построение множества $F_{\max}(H)$, $H \in H_D$, $H \neq H_{opt}$, на GPU. На вход MaxSetsGPU подаются исходная база данных D , значение уровня поддержки s , число вычислительных нитей z , набор порогов $H = \{q_1, \dots, q_n\}$, множество $F_{\max}(H) = \emptyset$ и пара $(F_{\max}(H^*), H^*) = (F_{\max}(H_{opt}), H_{opt})$. На выходе из MaxSetsGPU пара $(F_{\max}(H), H)$ добавляется в $F(H_D)$.

В процедуре MaxSetsGPU выполняются следующие действия 1–3.

1. Просматриваются элементы из $F_{\max}(H^*)$, $H^* = \{p_1, \dots, p_n\}$. Если для $A \in F_{\max}(H^*)$, $A = \{a_1, \dots, a_n\}$, найдется $t \in \{1, \dots, n\}$ такое, что $a_t = 1$, $p_t \neq q_t$, то осуществляется переход к 2. В противном случае элемент A добавляется в $F_{\max}(H)$. Данная проверка элемента A выполняется параллельно z вычислительными нитями GPU. После просмотра всех элементов в $F_{\max}(H^*)$ происходит переход к 3, после которого построение множества $F_{\max}(H)$ заканчивается и осуществляется выход из процедуры MaxSetsGPU.
2. Пусть $M(A)$ – множество различных элементов из P^H , каждое из которых получается из A заменой одного элемента 1 на элемент 0. Если A – s -частый элемент в D_H и $A \notin F_{\max}(H)$, то элемент A добавляется в $F_{\max}(H)$. Если же A – s -нечастый элемент в D_H и $|A| > 2$, то действие 2 осуществляется для каждого элемента из $M(A)$. Во всех других случаях элемент A не добавляется в $F_{\max}(H)$.
3. Из множества $F_{\max}(H)$ удаляются элементы, которые предшествуют другим элементам из $F_{\max}(H)$. Данный блок выполняется параллельно z вычислительными нитями GPU.

В алгоритме DPTFFP-tree в отличие от алгоритма PTFFP-tree построение множества $F_{\max}(H)$ осуществляется параллельно для всех $H \in H_D$, $H \neq H_{opt}$. Для этого создается $|H_D| - 1$ вычислительных потоков, в каждом из которых вызывается процедура MaxSetsGPU. Также для лучшей загрузки мультипроцессоров GPU в алгоритме DPTFFP-tree в действиях 1 и 2 процедуры MaxSetsGPU активно применяется динамический параллелизм (возможность динамически порождать внутри потока новые вычислительные потоки без возврата к коду, исполняемому на CPU [16]).

Дополнительное отличие состоит в том, что каждый вычислительный поток “сообщает” свой результат всем вычислительным потокам, которые находятся в очереди на выполнение, и те из них, которые запустятся далее, перед каждым вызовом процедуры MaxSetsGPU действуют следующим образом. Среди пар $(F_{\max}(H'), H')$, $H' \neq H$, $H' = \{p_1, \dots, p_n\}$, $H = \{q_1, \dots, q_n\}$, $S_D(\varphi_i(q_i)) \leq S_D(\varphi_i(p_i))$, $i \in \{1, \dots, n\}$, множества $F(H_D)$ ищутся пары с минимальным числом неравенств $S_D(\varphi_i(q_i)) < S_D(\varphi_i(p_i))$, $i \in \{1, \dots, n\}$, из которых выбирается пара $(F_{\max}(H^*), H^*)$ с максимальным номером. Поиск пары $(F_{\max}(H^*), H^*)$ полностью выполняется на GPU с применением динамического параллелизма.

5. Результаты численных экспериментов на реальных и модельных данных

Выполнены эксперименты на случайных модельных данных, содержащих как бинарные атрибуты, так и небинарные целочисленные атрибуты. Число небинарных атрибутов составляло 10% от общего числа атрибутов. Значение бинарного атрибута в транзакции полагалось равным 0 с вероятностью v , $v \in [0,5; 0,9]$, и равным 1 с вероятностью $1 - v$. Число v выбиралось с применением датчика случайных чисел. Значения небинарного атрибута выбирались из интервала $[0; 9]$ с равной вероятностью. Число транзакций m изменялось от 60 до 1 800 000, число атрибутов n изменялось от 30 до 40. Максимальные пороговые s -частые элементы искались при $s = 0,3$. Время счета измерялось в миллисекундах.

На рис. 1–4 приведено время поиска всех максимальных пороговых s -частых элементов алгоритмом FP-tree, основанным на последовательном синтезе для каждого значимого набора порогов классического бинарного FP-дерева, алгоритмом TFP-tree, основанным на построении TFP-дерева, и параллельными алгоритмами, описанными в разделе 4. Для каждого алгоритма приведено среднее время поиска при анализе 20 случайных баз данных размера $m \times n$. Дополнительно указаны число значимых наборов порогов $|H_D|$ и число найденных максимальных пороговых s -частых элементов $|F_D|$. На рис. 4 представлены результаты работы алгоритмов DP-TFP-tree и TFP-tree при числе небинарных атрибутов k , равном 1, ..., 7.

Из полученных результатов можно сделать следующие выводы.

Более быстрым из двух разработанных параллельных алгоритмов является алгоритм DP-TFP-tree (рис. 1). При $n = 40$ алгоритм DP-TFP-tree в среднем в 6 раз работает быстрее алгоритма PTFP-tree.

Алгоритм TFP-tree работает существенно быстрее алгоритма FP-tree. Например, при $n = 40$ время счета TFP-tree в 20–40 раз меньше времени счета FP-tree (рис. 2). Этот вывод ранее был получен в [13–15].

При большом числе транзакций (более 30 000 транзакций) и относительно небольшом числе значимых наборов порогов (не более 2500) алгоритм DP-TFP-tree работает быстрее алгоритма TFP-tree в среднем в четыре раза (рис. 3). Заметим, что при $n = 40$, $m \in \{100, 200, 300\}$ время счета TFP-tree меньше времени счета DP-TFP-tree в 3–10 раз (рис. 1–2) за счет того, что копирование данных между GPU и CPU занимает слишком большое время.

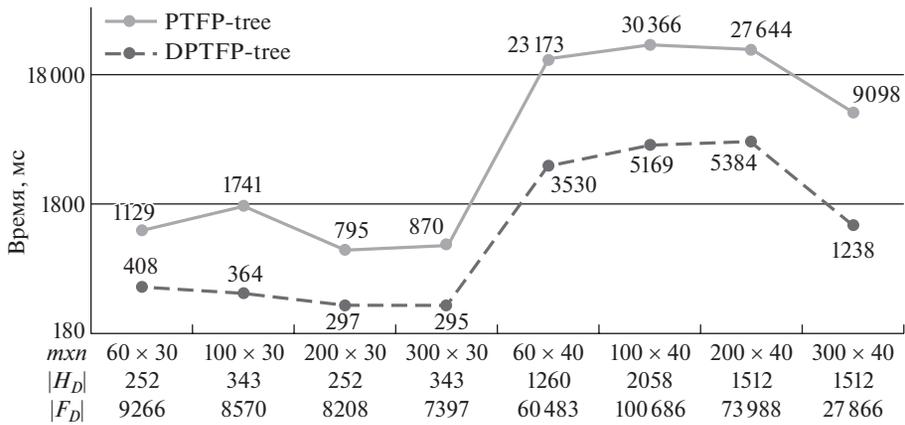


Рис. 1. Зависимость времени поиска максимальных пороговых s -частых элементов параллельными алгоритмами от размерности базы данных.

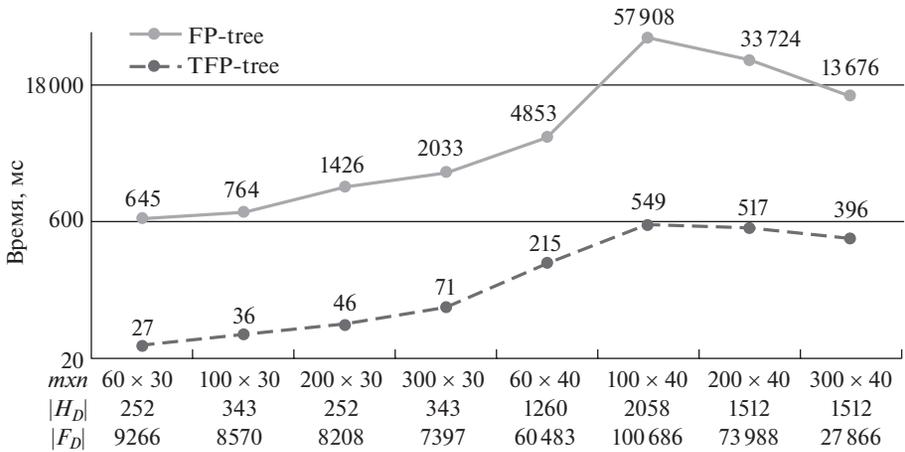


Рис. 2. Зависимость времени поиска максимальных пороговых s -частых элементов последовательными алгоритмами от размерности базы данных.

При большом числе значимых наборов порогов (более 600 000) и небольшой размерности базы данных ($n = 40$, $m = 300$) алгоритм DPTFP-tree работает быстрее алгоритма TFP-tree в среднем в три раза (рис. 4).

Тестирование алгоритмов DPTFP-tree и TFP-tree также осуществлялось на 7 реальных задачах из репозитория UCI [18]: Stone Flakes (задача № 1), Cloud (задача № 2), Wine Quality White (задача № 3), Clickstream data for online shopping (задача № 4), Wholesale customers (задача № 5), Adult (задача № 6), Heart Disease (задача № 7).

В таблице представлены результаты счета для задач 1–7. В задачах 4–7 варьировались параметры: s (значение уровня поддержки) и h (максимальное число значимых порогов для небинарного атрибута). При отсутствии ограничения на значение параметра h в соответствующей ячейке таблицы поставлен

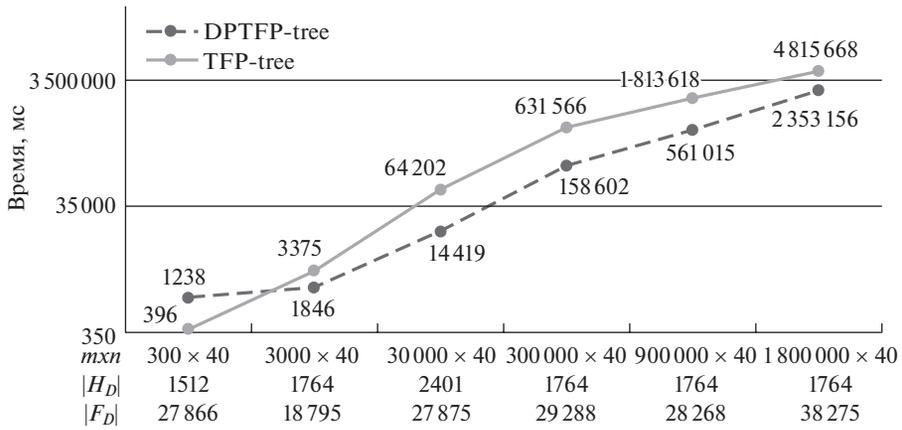


Рис. 3. Зависимость времени поиска максимальных пороговых s -частых элементов от m и при $n = 40$.

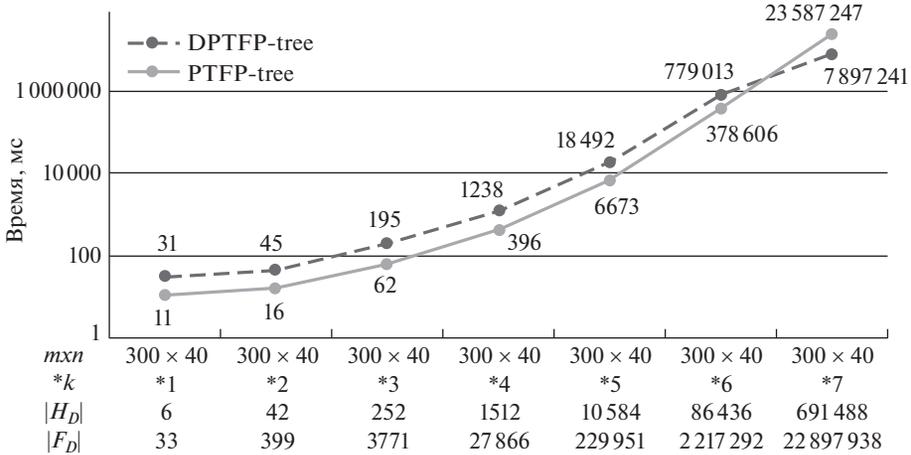


Рис. 4. Зависимость времени поиска максимальных пороговых s -частых элементов от числа небинарных атрибутов k при $n = 40$ и $m = 300$.

знак *. Для наглядности результатов счета приведены значения величины r , равной $(-t_G/t_C)$, если $t_G > t_C$, иначе равной t_C/t_G , где t_C – время работы алгоритма TFP-tree, t_G – время работы алгоритма DPTFP-tree. Серым цветом выделены значения величины r в случае, когда достигается ускорение алгоритма DPTFP-tree относительно алгоритма TFP-tree.

Из результатов, представленных в таблице, следует, что при большом числе значимых наборов порогов и при большом числе транзакций алгоритм DPTFP-tree до 3,6 раз работает быстрее алгоритма TFP-tree. На задачах с небольшим числом транзакций или с небольшим числом значимых наборов порогов алгоритм DPTFP-tree до 2,3 раз работает медленнее алгоритма TFP-tree.

Эффективность алгоритмов TFP-tree и DPTFP-tree на реальных задачах

Описание задачи				$ H_D $	$ F_D $	TFP-tree, с	DPTFP-tree, с	r
№	$m \times n$	s	h					
1	79×8	90	7	84 672	695 634	119	167	-1,4
2	2048×10	90	3	786 432	786 432	22 740	6400	3,6
3	4898×12	90	2	354 294	354 294	4520	1609	2,8
4	165474×12	90	*	768	2501	41	14	2,9
4	165474×12	80	*	20 300	103 052	1159	372	3,1
4	165474×12	70	*	216 384	1 511 248	16 442	4884	3,4
5	440×8	90	5	46 656	46 656	25	31	-1,2
5	440×8	90	6	117 649	117 649	180	186	-1,0
5	440×8	90	7	262 144	262 144	1087	864	1,3
6	32560×15	90	60	42 090	186 900	504	398	1,3
6	32560×15	90	70	48 990	220 480	548	496	1,1
6	32560×15	90	80	55 890	254 532	840	588	1,4
7	303×14	90	10	15 488	94 696	6	9	-1,5
7	303×14	80	10	29 282	123 022	18	42	-2,3
7	303×14	70	10	197 568	787 141	727	1230	-1,7

Таким образом, показана целесообразность применения в случае данных большого размера параллельного поиска максимальных пороговых s -частых элементов, осуществляемого на базе TFP-дерева и технологии CUDA. Установлено, что время счета заметно снижается при использовании динамического параллелизма в вычислениях на графическом ускорителе GPU.

6. Заключение

Рассмотрена актуальная задача логического анализа данных – задача поиска (максимальных) пороговых частых элементов произведения частичных порядков. Для ее решения на основе использования TFP-дерева (порогового FP-дерева) и технологии CUDA (Compute Unified Device Architecture) построены и исследованы параллельные алгоритмы. Конструкция TFP-дерева является модификацией конструкции классического бинарного FP-дерева на случай небинарных данных и ранее была предложена в [13–15].

Приведены результаты тестирования построенных алгоритмов на модельных и реальных данных. Показано, что в случае больших данных использование TFP-дерева заметно снижает время синтеза искомым частых элементов по сравнению с их последовательным поиском на базе классического FP-дерева, и установлена целесообразность применения технологии CUDA для решения поставленной задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Agrawal R., Imielinski T., Swami A. Mining Association Rules between Sets of Items in Large Databases // Proc. 1993 ACM SIGMOD Int. Conf. on Management of Data. 1993. P. 207–216.

2. *Aggarwal C., Jiawei H.* Frequent Pattern Mining. Springer, 2014.
3. *Agrawal R., Srikant R.* Fast Algorithms for Mining Association Rules in Large Databases // VLDB Conf. 1994. P. 487–499.
4. *Zaki M.J.* Scalable Algorithms for Association Mining // IEEE Trans. Knowledge and Data Engineering. 2000. V. 12. No. 3. P. 372–390.
5. *Han J., Pei H., Yin Y.* Mining Frequent Patterns without Candidate Generation // Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). 2000. P. 1–12.
6. *Borgelt C., Wang X.* SaM: A Split and Merge Algorithm for Fuzzy Frequent Item Set Mining // IFSA/EUSFLAT Conf. 2009. P. 968–973.
7. *Borgelt C.* Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination // OSDM. 2005. P. 66–70.
8. *Takeaki U., Masashi K., Hiroki A.* LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets // FIMI '04. 2004.
9. *Agrawal R., Aggarwal C., Prasad V.* Depth First Generation of Long Patterns // KDD '00: Proc. 6th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining. 2000. P. 108–118.
10. *Imberman S.P., Domanski B.* Finding Association Rules from Quantitative Data Using Data Booleanization // AMCIS Proc. 2001. P. 369–375.
11. *Angiulli F., Ianni G., Palopoli L.* On the Complexity of Inducing Categorical and Quantitative Association Rules // Theoretical Comput. Sci. 2004. V. 314. No. 1. P. 217–249.
12. *Elbassioni K.M.* On Finding Minimal Infrequent Elements in Multi-dimensional Data Defined over Partially Ordered Sets // arXiv:1411.2275. 2014.
13. *Генрихов И.Е., Дюкова Е.В.* О поиске ассоциативных правил в небинарных данных // 19-я Всеросс. конф. с междунар. участием “Математические методы распознавания образов”. М.: РАН, 2019. С. 15–19.
14. *Генрихов И.Е., Дюкова Е.В.* Поиск частых элементов произведения частичных порядков и ассоциативные правила // Информационные технологии и нанотехнологии (ИТНТ-2020). Сб. тр. по матер. VI Междунар. конф. и молодежной школы (Самара, 26–29 мая). Самара: Изд-во Самар. ун-та, 2020. Том 4. Науки о данных. С. 620–629.
15. *Genrikhov I.E., Djukova E.V.* Finding Frequent Elements for a Product of Partial Orders and Association Rules // Int. Conf. on Information Technology and Nanotechnology (ITNT), Samara, Russia. 2020. P. 1–5.
16. *Боресков А.В., Харламов А.А., Марковский Н.Д.* Параллельные вычисления на GPU. Архитектура и программная модель CUDA. М.: Изд-во Моск. ун-та, 2012.
17. *Cheng J., Grossman M., McKercher T.* Professional CUDA C Programming. N.Y.: Wrox, 2014.
18. *Lichman M.* UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. 2013. <http://archive.ics.uci.edu/ml>.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 05.04.2021

Принята к публикации 30.06.2021

© 2021 г. М.С. ГЕРМАНЧУК (m.german4uk@yandex.ru),
М.Г. КОЗЛОВА, канд. физ.-мат. наук (art-inf@mail.ru),
В.А. ЛУКЪЯНЕНКО, канд. физ.-мат. наук (art-inf@yandex.ru)
(Крымский федеральный университет им. В.И. Вернадского, Симферополь)

ПСЕВДОБУЛЕВЫЕ МОДЕЛИ УСЛОВНОЙ ОПТИМИЗАЦИИ ДЛЯ КЛАССА ЗАДАЧ МНОГИХ КОММИВОЯЖЕРОВ

Рассматриваются знаниеориентированные модели, задачи и алгоритмы построения маршрутов в сложных сетях агентами-коммивояжерами. Формализация приводит к моделям псевдодобулевой дискретной оптимизации с ограничениями, учитывающими специфику задачи многих коммивояжеров. Рассмотрен класс задач, который представим в виде псевдодобулевых оптимизационных моделей с сепарабельными целевыми функциями (монотонные, линейные) и ограничениями в виде дизъюнктивных нормальных форм (ДНФ). Показана возможность приближенного синтеза ДНФ ограничений на основе прецедентной информации. Приведена методология, теоретические положения и алгоритмы решения такого класса задач. Показано, что решение задач маршрутизации может базироваться на применении многоагентного подхода в сочетании с кластеризацией исходной задачи, алгоритмах псевдодобулевой оптимизации с дизъюнктивными ограничениями и метаэвристиках.

Ключевые слова: многоагентные задачи коммивояжера, модели псевдодобулевой условной оптимизации с дизъюнктивными ограничениями, метаэвристики.

DOI: 10.31857/S0005231021100044

1. Введение

Прикладная теория задач маршрутизации на сложных сетях (типа многих агентов-коммивояжеров) базируется на точных решениях выделенных классов задач с полиномиальными алгоритмами решения, использовании приближенных алгоритмов решения (например, с гарантированной функциональностью) и декомпозиции (кластеризации) исходной задачи, т.е. сведения к задачам меньшей размерности и уточняющих преобразованиях для возврата к исходной задаче. Важным в этом процессе является учет всей имеющейся информации, знаний, фактов и прецедентов как для построения иерархии моделей (извлечение моделей), так и для разработки практических алгоритмов решения [1–10].

Разнообразие алгоритмов также связано с наличием априорных знаний о решении или структуре сети, прецедентным характером знаний и требованиями к точности решения. Рационально использование как точных, так и приближенных алгоритмов и их композиций. Заметим, что задачи прикладной маршрутизации возникают в сочетании с другими известными задачами:

задача о ранце, распределение ресурсов, кластеризации, максимального разреза, покрытия и т.п.

Многоагентные системы с роевым интеллектом используются для решения сложных задач дискретной оптимизации, которые нельзя эффективно решать классическими алгоритмами. Агентная модель для сложной сети задачи типа многих коммивояжеров (Multiple Traveling Salesman Problem, *mTSP*) становится интеллектуализированной системой, определяющей эвристические алгоритмы поиска оптимального решения реактивными агентами (следующих заложенным в них правилам).

Синтез многоагентных систем (МАС) искусственного интеллекта (ИИ) по частичной, прецедентной, априорной информации базируется на результатах наблюдения за поведением МАС на основе накопленной информации в виде [6]: "... вектора состояния, значения качества функционирования системы, бинарного индикатора допустимости этого состояния". Для МАС маршрутизации типа *mTSP* используется модель скалярной псевдодвулевой условной оптимизации с ограничениями в виде дизъюнктивной нормальной формы (ДНФ). Такие модели естественным образом учитывают линейные ограничения по прохождению вершин сети, декларативные требования, требования предшествования, обязательного прохождения выделенного множества дуг и другую прецедентную информацию.

Псевдодвулевые оптимизационные модели с сепарабельными целевыми функциями и ДНФ ограничениями, имеющими ограниченную постоянную величиной длину, являются полиномиально разрешимыми. Представляют интерес классы задач, которые приведены или легко приводятся к форме с ДНФ ограничениями, так как в общем случае такие приведения являются экспоненциальными. Синтез модели с ДНФ ограничениями из данных можно осуществлять приближенно, и сложность такой аппроксимации оказывается полиномиальной. В [6] показано, что число конъюнкций в извлеченной ДНФ не превышает числа примеров в исходной прецедентной информации. При этом указывается, что для построения ДНФ ограничений целесообразно использовать решающие деревья. В случае монотонности и линейности частично заданной целевой функции в публикациях В.И. Донского [3, 6] и М.Г. Козловой [7, 8] предложены алгоритмы решения задач псевдодвулевой скалярной оптимизации при наличии неполной, прецедентной начальной информации. Идея этого подхода будет применена для решения многоагентных задач типа многих коммивояжеров.

В настоящей статье приводится часть проекта, представленного в декабре 2020 г. на Международной конференции "Интеллектуализация обработки информации" [2]. Исторические аспекты по задачам коммивояжера, их обобщениям, точным и приближенным алгоритмам решения можно найти в [11–13]. В [14] показано применение композиции алгоритмов: модификация генетического алгоритма, муравьиный, роевой (пчелиной колонии), имитации отжига. Предложен и реализован обобщенный алгоритм, в котором исходной сети ставится в соответствие более простая сеть (сеть облета). Алгоритм инспирирован рядом актуальных прикладных задач: задачей планирования много-

дневных туристических маршрутов на инфраструктурной сети достопримечательностей Крыма и задачей доставки ресурсов агентами-коммивояжерами по территории Ялты в условиях чрезвычайных ситуаций (ЧС). Численный эксперимент проведен для задачи маршрутизации по карте ГИС для городской инфраструктуры. Реализованы алгоритмы кластеризации, в которых первоначально пройденные маршруты уточняются с помощью алгоритмов 2-opt, имитации отжига и других метаэвристик [14].

В данной статье выделено важное направление – построение МАС $mTSP$ на базе моделей псевдодобулевой оптимизации с дизъюнктивными ограничениями.

2. Предварительные сведения. Задачи псевдодобулевой оптимизации

Оптимизационные задачи с булевыми переменными имеют широкие приложения [1, 3, 15]. В связи с задачами маршрутизации на графах особый интерес представляют задачи псевдодобулевой оптимизации. Достаточно подробно такие задачи исследовались в [16, 17], где разработаны методы решения в случае аналитически заданных моделей псевдодобулевой оптимизации. В [18] псевдодобулевые функции рассматриваются как отображения из семейства подмножеств конечного исходного множества действительных чисел. Оптимизация на графах в классе псевдодобулевых функций представлена в [19, 20]. Базовые результаты содержатся в [21, 22].

Введем обозначения:

$B^n = \{0, 1\}^n$ – единичный n -мерный куб, $P_2(n) = \{F : B^n \rightarrow \{0, 1\}\}$ – класс функций алгебры логики (ФАЛ), зависящих от n переменных, $\tilde{x} = (x_1, x_2, \dots, x_n) \in B^n$.

Функция вида $f : B^n \rightarrow \mathbb{R}$, где \mathbb{R} – множество действительных чисел, называется псевдодобулевой [1, 3, 18]. Для обозначения класса таких функций будем использовать обозначение $PS_2(n)$, а для обозначения класса линейных псевдодобулевых функций – $LPS_2(n)$. Функции из $PS_2(n)$ определены на множестве вершин единичного n -мерного куба B^n и могут принимать вещественные значения.

Задача вида

$$(1) \quad \text{extr } f(\tilde{x}), \quad \tilde{x} \in \Omega \subseteq B^n, \quad f \in PS_2(n),$$

называется задачей псевдодобулевой оптимизации.

Введем характеристическую функцию множества ограничений Ω :

$$F_\Omega(\tilde{x}) = \begin{cases} 1, & \tilde{x} \in \Omega; \\ 0, & \tilde{x} \in B^n \setminus \Omega. \end{cases}$$

Задачу (1) можно представить в эквивалентной форме:

$$(2) \quad \text{extr } f(\tilde{x}), \quad F_\Omega(\tilde{x}) = 1, \quad f \in PS_2(n), \quad F_\Omega \in P_2(n), \quad \tilde{x} \in B^n,$$

где $P_2(n)$ – класс функций алгебры логики от n переменных.

Пусть $D_{F_\Omega} = \bigvee_{j=1}^m K_j$ – любая дизъюнктивная нормальная форма функции $F_\Omega(\tilde{x})$; тогда задача, эквивалентная задачам (1) и (2), имеет вид:

$$(3) \quad \text{extr } f(\tilde{x}), \quad D_{F_\Omega}(\tilde{x}) = 1, \quad \tilde{x} \in B^n.$$

Задача (3) называется задачей псевдодобулевой оптимизации с дизъюнктивным ограничением, и форму ее представления называют канонической.

Определение 1. Переменная x_i называется существенной для $f \in PS_2(n)$, если найдется такой набор значений переменных $\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n$, что $f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) \neq f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$. В противном случае переменная называется фиктивной.

Определение 2. Псевдодобулевые функции f_1 и f_2 называются равными, если функция f_2 может быть получена из f_1 путем введения или удаления фиктивных переменных.

Каноническая форма псевдодобулевой функции f аналогична совершенной дизъюнктивной нормальной форме в $P_2(n)$ и имеет вид

$$(4) \quad f(x_1, \dots, x_n) = \sum_{\tilde{\sigma} \in B^n} a_{\tilde{\sigma}} x_1^{\sigma_1} \cdot \dots \cdot x_i^{\sigma_i} \cdot \dots \cdot x_n^{\sigma_n},$$

где

$$\tilde{\sigma} = (\sigma_1, \dots, \sigma_n), \quad \sigma_i \in \{0, 1\}, \quad i = \overline{1, n}, \quad a_{\tilde{\sigma}} \in \mathbb{R}, \quad x^\sigma = \begin{cases} x, & \sigma = 1, \\ \bar{x}, & \sigma = 0. \end{cases}$$

Каждая псевдодобулевая функция может быть представлена в полиномиальной форме над полем действительных чисел

$$(5) \quad f(x_1, \dots, x_n) = \sum_{j=1}^{k_f} c_j x_{j_1} \cdot \dots \cdot x_{j_{r_j}} + c_0, \quad c_j \in \mathbb{R}, \quad j = 1, \dots, k_f.$$

Задача псевдодобулевой оптимизации в форме слабых неравенств имеет вид:

$$(6) \quad \text{extr } f(x_1, \dots, x_n), \quad g_j(x_1, \dots, x_n) \leq 0, \quad j = 1, 2, \dots, m, \\ (x_1, \dots, x_n) \in B^n, \quad f, g_j \in PS_2(n).$$

Определение 3. Две формы представления оптимизационной задачи называются эквивалентными, если множества их решений совпадают.

Теорема 1 [4]. Для любой задачи псевдодобулевой оптимизации в форме (1) существует эквивалентная форма представления

$$(7) \quad \text{extr } f(\tilde{x}), \quad h(\tilde{x}) \leq 0, \quad \tilde{x} \in B^n,$$

с единственным ограничением в виде нестрогого неравенства, где $f, h \in PS_2(n)$ есть некоторые полиномы.

Полиномиальное представление для функции $f \in PS_2(n)$ существует всегда.

Теорема 2 [4]. Любая задача оптимизации псевдодобулевой функции с ограничениями, определяющими непустое множество допустимых решений Ω задачи (1), может быть представлена в эквивалентной форме с дизъюнктивным условием (3).

Доказательство теоремы 2 следует из существования эквивалентной формы задачи с характеристическими функциями на наборе допустимых значений $F_\Omega(\tilde{x})$ и полноты представления ФАЛ в виде дизъюнктивных нормальных форм.

Определение 4. Представление задач произвольного класса Z в форме F называется полным в Z , если любая задача этого класса может быть представлена в форме F .

Из данного определения следует, что представление задач условной оптимизации псевдодобулевой функции в форме с дизъюнктивным условием является полным.

Область допустимых решений задачи (3) и эквивалентной ей задачи (1) может быть представлена в виде:

$$\Omega = \bigcup_{j=1}^m N_{K_j},$$

где N_{K_j} – интервал ранга r_j , соответствующий элементарной конъюнкции $K_j = x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_{r_j}}^{\sigma_{j_{r_j}}}$, что приводит к еще одной эквивалентной форме задачи (1):

$$(8) \quad \underset{1 \leq j \leq m}{extr} \quad \underset{\tilde{x} \in N_{K_j}}{extr} \quad f(\tilde{x}).$$

Действительно, учитывая, что область допустимых решений есть объединение интервалов N_{K_j} , $j = \overline{1, m}$, легко убедиться, что экстремальное решение задачи можно определить путем его выбора из предварительно найденных допустимых решений, являющихся экстремальными в интервалах N_{K_j} .

Рассмотрим основные алгоритмы решения задач псевдодобулевой оптимизации. Пусть дана задача псевдодобулевой оптимизации с линейной целевой функцией

$$(9) \quad \max \sum_{i=1}^n c_i x_i, \quad \bigvee_{j=1}^m K_j(\tilde{x}) = 1, \quad \tilde{x} \in B^n,$$

где $K_j(\tilde{x}) = x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_{r_j}}^{\sigma_{j_{r_j}}}$, $j = \overline{1, m}$. Приведем к эквивалентной форме:

$$(10) \quad \max \sum_{i=1}^n c_i x_i, \quad \tilde{x} \in \bigcup_{j=1}^m N_{K_j},$$



Рис. 1. Схема решения задачи.

где N_{K_j} – интервал в B^n , соответствующий конъюнкции K_j ; интервал N_{K_j} определяется набором значений $\{j_1, \dots, j_{r_j}\}$ (направление) и множеством $\{\sigma_{j_1}, \dots, \sigma_{j_{r_j}}\}$ (код интервала).

Решение (10) сводится к решению задачи

$$(11) \quad \max_{1 \leq j \leq m} \max_{\tilde{x} \in N_{K_j}} \sum_{i=1}^n c_i x_i,$$

которое в свою очередь требует решения m задач вида:

$$(12) \quad \max_{\tilde{x}} \sum_{i=1}^n c_i x_i, \quad \tilde{x} : (x_{j_1} = \sigma_{j_1}) \& \dots \& (x_{j_{r_j}} = \sigma_{j_{r_j}}).$$

Задача (12) решается следующим образом: допустимыми являются только те булевы наборы \tilde{x} , у которых зафиксированы координаты $x_{j_1} = \sigma_{j_1}, \dots, x_{j_{r_j}} = \sigma_{j_{r_j}}$, а остальные могут иметь любые значения из множества $\{0, 1\}$. Свободные переменные (вне множества номеров $\{j_1, \dots, j_{r_j}\}$) можно назначать единичными или нулевыми в зависимости от значения c_i , $i \in \{1, 2, \dots, n\} \setminus \{j_1, \dots, j_{r_j}\}$ – коэффициентов целевой функции. Экстремальные решения \tilde{x}^* задачи (12) будут определяться формулой [23]:

$$(13) \quad x_i^* = \begin{cases} \sigma_i, & i \in \{j_1, \dots, j_{r_j}\}, \\ \varphi(c_i), & i \notin \{j_1, \dots, j_{r_j}\}, \end{cases}$$

где

$$\varphi(c_i) = \begin{cases} 1, & c_i > 0, \\ 0, & c_i < 0, \\ \alpha, & c_i = 0, \end{cases}$$

α – любое значение из $\{0, 1\}$. В случае когда все $c_i \neq 0$, задача (12) имеет единственное решение, а исходная задача (11) – не более m решений.

На каждом интервале N_{K_j} при вычислении x_i^* согласно (13) просматривается n значений, а интервалов всего m , поэтому сложность решения $O(mn)$.

Так как любую задачу псевдодвулевой оптимизации можно представить в эквивалентной форме с ДНФ ограничением, то это справедливо и для задач с линейной целевой функцией. Следовательно, решение любой линейной задачи (в том числе задачи коммивояжера) можно осуществлять по схеме, представленной на рис. 1.

Теорема 3 [4]. *Если задача условной оптимизации линейной псевдодвулевой функции с ограничениями-неравенствами приводится к эквивалентной форме с дизъюнктивным ограничением за число шагов, ограниченное полиномом от размерности задачи, то она разрешима за полиномиальное время.*

3. Псевдодвулевая модель задачи коммивояжера

Рассмотрим задачу коммивояжера, формализованную в виде модели линейной псевдодвулевой условной оптимизации с неотрицательными коэффициентами ($c_{ij} \geq 0$) целевой функции:

$$(14) \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min,$$

$$(15) \quad \sum_{i=1}^n x_{ij} = 1, \quad \sum_{j=1}^n x_{ij} = 1, \quad i, j = \overline{1, n},$$

$$(16) \quad u_i - u_j + n x_{ij} \leq n - 1, \quad i, j = \overline{2, n}, \quad i \neq j.$$

Здесь u_i — произвольные действительные числа (в частности, им может соответствовать нумерация вершин, по которым проходит коммивояжер). Ограничения (16) препятствуют образованию подциклов. Чтобы упростить выкладки, будем использовать обозначение двухиндексных величин через одноиндексные: $\tilde{x} = (x_1, x_2, \dots, x_N) = (x_{11}, x_{12}, \dots, x_{nn}) \in B^N$, $c = (c_1, c_2, \dots, c_N) = (c_{11}, c_{12}, \dots, c_{nn})$, где $N = n^2$. Ограничениям можно поставить в соответствие функции $F_j(\tilde{x}) \in P_2(N)$, $j = \overline{1, M}$, где M — число ограничений. Можно в \tilde{x} использовать и другую нумерацию элементов: x_{ij} , $i, j = \overline{1, n}$.

Определение 5. *Вершина $\tilde{\alpha} \in B^N$ называется верхним нулем монотонной функции алгебры логики $f(\tilde{x})$, если $f(\tilde{\alpha}) = 0$ и для всякой вершины $\tilde{\beta}$ из $\tilde{\alpha} < \tilde{\beta}$ следует, что $f(\tilde{\beta}) = 1$ [24].*

Лемма 1. *Если*

$$\left\{ \sum_{j=1}^n x_{ij} = 1, \sum_{i=1}^n x_{ij} = 1, i, j = \overline{1, n}, u_i - u_j + n x_{ij} \leq n - 1, i \neq j \right\} \Leftrightarrow \{F_j(\tilde{x}) = 0\},$$

то F_j — монотонные функции алгебры логики (ФАЛ) и задачу (14)–(16) можно записать в виде

$$(17) \quad (c, \tilde{x}) \rightarrow \min, \quad F_0(\tilde{x}) = \bigvee_{j=1}^M F_j(\tilde{x}) = 0,$$

где $c = (c_{11}, c_{12}, \dots, c_{nn})$, (c, \tilde{x}) — скалярное произведение, $F_0(\tilde{x})$ — монотонная ФАЛ.

Доказательство. Покажем монотонность функций $F_j(\tilde{x})$. Пусть $\tilde{\alpha}, \tilde{\beta} \in B^N$ такие, что $\tilde{\alpha} \preceq \tilde{\beta}$, т.е. $\alpha_{ij} \leq \beta_{ij} \quad \forall i, j = \overline{1, n}$. Тогда

$$S(\tilde{\alpha}) = \left\{ \begin{array}{l} \sum_{j=1}^n \alpha_{ij} - 1, \quad \sum_{i=1}^n \alpha_{ij} - 1, \quad i, j = \overline{1, n}, \\ u_i - u_j + n\alpha_{ij} - (n-1), \quad i, j = \overline{2, n} \end{array} \right\} \leq$$

$$\leq \left\{ \begin{array}{l} \sum_{j=1}^n \beta_{ij} - 1, \quad \sum_{i=1}^n \beta_{ij} - 1, \quad i, j = \overline{1, n}, \\ u_i - u_j + n\beta_{ij} - (n-1), \quad i, j = \overline{2, n} \end{array} \right\} = S(\tilde{\beta}).$$

Неравенства выполняются покомпонентно в силу $\alpha_{ij} \leq \beta_{ij} \quad \forall i, j = \overline{1, n}$.

По условию леммы $(S(\tilde{\alpha}) \leq 0) \Leftrightarrow (F_j(\tilde{\alpha}) = 0)$, $(S(\tilde{\beta}) \leq 0) \Leftrightarrow (F_j(\tilde{\beta}) = 0)$. Учитывая, что $S(\tilde{\alpha}) \preceq S(\tilde{\beta})$, имеем $(F_j(\tilde{\alpha}) = 1) \Rightarrow (F_j(\tilde{\beta}) = 1)$, поэтому $F_j(\tilde{\alpha}) \leq F_j(\tilde{\beta})$. Следовательно, $F_j(\tilde{x})$ – монотонная ФАЛ.

Функция $F_0(\tilde{x}) = \bigvee_{j=1}^M F_j(\tilde{x})$ является монотонной. Это следует из того, что класс монотонных ФАЛ является замкнутым и содержит дизъюнкцию. Функция $F_0(\tilde{x})$ равна нулю тогда и только тогда, когда выполняются все ограничения (14)–(16), так как дизъюнкция $\bigvee_{j=1}^M F_j(\tilde{x})$ равна нулю только при $F_j(\tilde{x}) = 0$ для всех $j = \overline{1, M}$. Лемма 1 доказана.

Отсюда следует, что если область допустимых решений $\Omega = \{ \tilde{x} \in B^N : F_0(\tilde{x}) = 0 \} \neq \emptyset$, то решением задачи является верхний ноль функций $F_0(\tilde{x})$. Задача (17) сводится к задаче расшифровки монотонной ФАЛ или к поиску ее верхних нулей [4].

Псевдодулевая задача линейного программирования (17)

$$(18) \quad \min_{x \in \Omega} (c, \tilde{x}), \quad \Omega = \{ \tilde{x} \in B^N : F_0(\tilde{x}) = 0 \},$$

с ДНФ ограничениями позволяет учитывать знания о решении задачи коммивояжера, которые представимы в ДНФ форме. Например, если необходимо включить прохождение дуг x_{kl} и x_{pm} , тогда к ограничениям добавляется условие $(x_{kl} - 1) \vee (x_{pm} - 1) = 0$. Если, наоборот, не включать, то $x_{kl} \vee x_{pm} = 0$ [25].

Полученный формализм позволяет учитывать знания о модели и решениях, использовать их в теоретических обоснованиях и конкретных алгоритмах решения.

4. Модели псевдобулевой условной оптимизации с дизъюнктивными ограничениями для задачи многих коммивояжеров

Задаче многих коммивояжеров ($mTSP$) с общими интересами соответствует одна целевая функция, выражающая минимум общего расстояния, как и в задаче для одного коммивояжера. Ограничения линейные. Задача приводится к задаче псевдобулевой оптимизации с дизъюнктивными ограничениями.

При большой размерности задачи (сложность сети) условной псевдобулевой оптимизации с ДНФ ограничениями применять полиномиальные алгоритмы, предназначенные для такого класса задач, может быть нерацionalmente. Требуется упрощение задачи (применение приближенных, эвристических методов) с помощью отсечения излишних вариантов перебора на основе имеющихся знаний. Прежде всего снижение сложности (размерности) достигается с помощью кластеризации сети. При этом количество агентов, количество депо и их расположение, количество кластеров может быть задано, искомо или быть произвольным.

Следуя идеологии сведения исходной задачи к нескольким задачам меньшей размерности, рассмотрим следующий подход к решению задачи маршрутизации для многих агентов. Пусть $m = 2$ (два агента).

Алгоритм решения задачи для двух коммивояжеров ($AmTSP$).

Вход: сеть $S = (G, C)$, $G = (U, V)$, $n = |V|$, U – множество дуг,
 C – матрица расстояний (весов); информация о структуре сети.

Выход: маршруты коммивояжеров, длина общего маршрута.

- 1: Провести кластеризацию сети: $S = S_1 \cup S_2$, $G = G_1 \cup G_2$, $V = V_1 \cup V_2$ ($V_1 \cap V_2 = \emptyset$).
 - 2: На сетях S_1, S_2 сформировать задачи коммивояжеров, выписать все основные и дополнительные ограничения.
 - 3: Трансформировать задачи коммивояжеров к задачам псевдобулевой оптимизации с ДНФ ограничениями.
 - 4: Найти решения задач с ДНФ ограничениями.
 - 5: Провести локальные преобразования, обмениваясь вершинами множеств V_1, V_2 . Добавление вершины приводит к изменению ДНФ ограничений (добавление интервалов конъюнкций).
Алгоритм остается полиномиальным.
 - 6: Выбрать лучший вариант (или провести заданное число итераций).
 - 7: Получить решение исходной задачи.
-

Каждый шаг алгоритма *AmTSP* конкретизируется в зависимости от структуры (сложности) исходной сети и всей имеющейся информации (знаний).

Задача каждого коммивояжера на выделенном кластере является задачей скалярной псевдодвулевой условной оптимизации, т.е. может быть представлена в канонической форме с дизъюнктивными ограничениями (9):

$$(19) \quad \min \left\{ f_k(\tilde{x}) = (c^k, \tilde{x}) = \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} c_{ij}^k x_{ij}, \bigvee_{j=1}^{m_k} x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_{r_j}}^{\sigma_{j_{r_j}}} = 1, k = \overline{1, m} \right\}.$$

Каноническая модель является исчерпывающей в своем классе в силу полноты. Левая часть ограничения (19) является ДНФ характеристической функции множества Ω^k -ограничений искомой задачи на k -м кластере, в которой может быть учтена дополнительная информация о структуре кластера и искомого решения (запреты, предписания и др.).

В случае общих интересов модель будет однокритериальной:

$$(20) \quad \min f_0(\tilde{x}) = \min \sum_{k=1}^m f_k(\tilde{x}), \bigvee_{j=1}^M x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_{r_j}}^{\sigma_{j_{r_j}}} = 1.$$

Процессом выбора решения будем называть поиск такого набора значений $\alpha \in B^N$, признаковых предикатов, чтобы (одновременно или по отдельности):

- обращался в единицу один или несколько целевых предикатов;
- достигала экстремального значения несколько (или одна в однокритериальной постановке) псевдодвулевых функций $f_k, k = \overline{1, m}$.

Единственное ограничение канонической модели (20) в виде ДНФ характеристического множества ограничений задает И/ИЛИ граф, которому соответствует логическая система продукций. Существование логической системы продукций (ЛСП)

$$\begin{aligned} x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_{r_j}}^{\sigma_{j_{r_j}}} \rightarrow g_j, \\ g_j \rightarrow g_0, \quad j = \overline{1, M}, \end{aligned}$$

позволяет выводить целевой факт $g_0 = \text{“}\tilde{x} \text{ — допустимое решение”}$. Граф И/ИЛИ ограничения канонической модели задачи *mTSP* является трех-ярусным. Любой граф ЛСП, не имеющий циклов, может быть сведен к трех-ярусному и представлен в виде ДНФ. Отсюда следует, что соответствующая база знаний (БЗ) системы построения допустимого решения *mTSP* должна удовлетворять следующим требованиям:

1) решения *mTSP* должны удовлетворять ограничениям задачи, следовательно, ЛСП должна обеспечивать возможность вывода целевых предикатов, соответствующих этим ограничениям;

2) группа ограничений, которые должны выполняться одновременно, задаются вершиной типа “И”, связывающей эти ограничения вместе.

Заметим, что ДНФ ограничение может быть получено с помощью обучения по прецедентной (эмпирической) информации [6, 7]. Для синтеза ДНФ по заданным ЛСП можно использовать D - и DS -алгоритмы [3], реализующие соответственно стратегии “сверху вниз” и “снизу вверх”, т.е. реализуется синтез областей допустимости решений в знаниеориентированных продукционных системах моделей псевдодобулевой условной оптимизации, соответствующих $mTSP$ (в этих же публикациях можно найти оценки сложности алгоритмов).

Вопрос полноты знаний об ограничениях в БЗ задачи коммивояжера является важным и рассматривается самостоятельно в теории знаниеориентированных систем.

Уточним шаги 4, 5 алгоритма $AmTSP$ с точки зрения преодоления неопределенности (для $m = 2$, аналогичная ситуация для $m > 2$). Задачи псевдодобулевой оптимизации для каждого кластера имеют вид:

$$(21) \quad \begin{aligned} \min f_1(\tilde{x}) &= \min(c^1, \tilde{x}), & F_1(\tilde{x}) &= 0, \\ \min f_2(\tilde{x}) &= \min(c^2, \tilde{x}), & F_2(\tilde{x}) &= 0, \\ c^k &= (c_{11}^k, c_{12}^k, \dots, c_{nn}^k) \equiv (c_1^k, c_2^k, \dots, c_j^k, \dots, c_N^k), \\ \tilde{x} &\in B^N, \quad N = n^2, \quad F_k(\tilde{x}) \in P_2(N), \quad k = 1, 2. \end{aligned}$$

В том случае, когда кластеры определены единственным образом, а целевые функции и ограничения заданы точно, решение полученных задач на кластерах сводится к расшифровке монотонной функции алгебры логики (или к поиску ее верхних нулей). В более общем случае модели $mTSP$ получены как неполное представление исходной задачи $mTSP$: когда в результате кластеризации (или при другом сведении и задаче меньшей размерности) при исследовании линейной модели не удалось получить полную информацию о ее ограничениях. Но по доказанному выше $F_j(\tilde{x})$ являются монотонными функциями алгебры логики.

Будем предполагать, что существуют множества

$$(22) \quad \begin{aligned} M_{k0}^{F_k} &= \left\{ \tilde{x} \in B^N : F_k(\tilde{x}) = 0 \right\}; & M_{k1}^{F_k} &= \left\{ \tilde{x} \in B^N : F_k(\tilde{x}) = 1 \right\}; \\ M_{k0}^{f_k} &= \left\{ \tilde{x} \in B^N : f_k(\tilde{x}) = 0 \right\}; & M_{k1}^{f_k} &= \left\{ \tilde{x} \in B^N : f_k(\tilde{x}) = 1 \right\}, \quad k = 1, 2. \end{aligned}$$

С позиции первого коммивояжера ($k = 1$) $F_k(\tilde{x})$ — функция, определяющая допустимые решения, задана частично с помощью указания множеств наборов $M_{k0}^{f_k}, M_{k1}^{f_k}$ (прецедентов или фактов), т.е. заданы некоторые частичные функции алгебры логики f_k , $k = 1, 2$.

Пусть Φ_k — множество монотонных функций алгебры логики из $P_2(n)$, принимающих значение “0” на множестве $M_{0k}^{f_k}$ и значение “1” на множестве $M_{1k}^{f_k}$, а $Z_k(\Phi_k)$ — множество всех верхних нулей всех функций из Φ_k .

Непротиворечивым решением задач (21) называется такой набор $\tilde{z}_k^* \in Z_k(\Phi_k)$, что

$$\sum_{j=1}^N c_j^k z_j^* = \min_{z \in Z_k(\Phi_k)} \sum_{j=1}^N c_j^k z_j.$$

Не теряя общности, можно считать, что булевы переменные упорядочены так, что $c_1^k > \dots > c_N^k$. Это легко выполнить для любой исходной задачи.

Теорема 4. Функция $f_k \in P_2(N)$, не являющаяся константой, монотонна тогда и только тогда, когда для любых пар вершин $\tilde{x}, \tilde{y} \in B^N$ таких, что $f_k(\tilde{x}) = 1$, $f_k(\tilde{y}) = 0$, найдется переменная с номером $i \in \{1, 2, \dots, N\}$ такая, что $x_i = 1$, $y_i = 0$.

Доказательство.

Необходимость. Докажем необходимость методом от противного. Пусть $f_k \in P_2(N)$ не константа, монотонна и не существует переменной с номером $i \in \{1, 2, \dots, N\}$ такой, что $x_i = 1$, $y_i = 0$, т.е. $x_i \leq y_i$, $i = \overline{1, N}$. Тогда $\tilde{x} \preceq \tilde{y}$, но $f_k(\tilde{x}) > f_k(\tilde{y})$, что противоречит условию монотонности функции f_k .

Достаточность. Рассмотрим три множества пар наборов $\tilde{x}, \tilde{y} \in B^N$:

$$W_{k1} = \{(\tilde{x}, \tilde{y}) : f_k(\tilde{x}) = 1, f_k(\tilde{y}) = 0\};$$

$$W_{k2} = \{(\tilde{x}, \tilde{y}) : f_k(\tilde{x}) = 0, f_k(\tilde{y}) = 1\};$$

$$W_{k3} = \{(\tilde{x}, \tilde{y}) : f_k(\tilde{x}) = f_k(\tilde{y})\}.$$

Пусть $\tilde{x} \in W_{k1}$. Тогда $f_k(\tilde{x}) > f_k(\tilde{y})$ и по условию теоремы 4 найдется такой индекс i , что $x_i > y_i$. Следовательно, либо $\tilde{x} \succ \tilde{y}$, либо наборы \tilde{x} и \tilde{y} – несравнимы. Для всех сравнимых наборов из W_{k1} имеем: $\tilde{x} \succ \tilde{y}$ и $f_k(\tilde{x}) > f_k(\tilde{y})$.

Аналогично проверяется выполнение условия монотонности функции f_k на множества W_{k2} .

Пусть $\tilde{x} \in W_{k3}$, тогда $f_k(\tilde{x}) = f_k(\tilde{y})$, в том числе всегда, когда $\tilde{x} \preceq \tilde{y}$. Учитывая, что объединение $W_{k1} \cup W_{k2} \cup W_{k3}$ содержит любую пару вершин куба B^N , получаем, что f_k – монотонная функция: если $\tilde{x} \preceq \tilde{y}$, то $f_k(\tilde{x}) \leq f_k(\tilde{y})$. Теорема 4 доказана.

На основании теоремы 4 можно сделать следующий вывод. Если во множествах $M_{k0}^{f_k}$ и $M_{k1}^{f_k}$ частичной функции алгебры логики f_k найдутся такие наборы $\tilde{\alpha} \in M_{k0}^{f_k}$ и $\tilde{\beta} \in M_{k1}^{f_k}$, что не существует переменной с номером $i \in \{1, \dots, N\}$, для которой $\alpha_i < \beta_i$, то f_k не может быть доопределена монотонной функцией.

Пусть частичная функция f_k доопределена монотонной функцией φ_k . Необходимо, чтобы $M_{k1}^{f_k} \subseteq M_{k1}^{\varphi_k}$, $M_{k0}^{f_k} \subseteq M_{k0}^{\varphi_k}$, следовательно, любой набор из $M_{k1}^{f_k}$ должен покрываться некоторым интервалом $N_j^k \subseteq M_{k1}^{\varphi_k}$, но N_j^k не должен содержать точек из $M_{k0}^{f_k}$, каждый набор из $M_{k0}^{f_k}$ должен покрываться некоторым интервалом $N_L^k \subseteq M_{k0}^{\varphi_k}$, но N_L^k не должен содержать точек из $M_{k1}^{f_k}$.

Класс монотонных функций φ_k , доопределяющих f_k , обозначим $\Phi_k \subset M_k$. Функции класса Φ_k определяют множество:

$$\overline{\Phi}_k = \left\{ g_k \in P_2(N) : g_k(\tilde{x}) = \overline{\varphi}_k(\tilde{x}), \varphi_k \in \Phi_k \right\}.$$

Любая функция может быть представлена сокращенной дизъюнктивной нормальной формой (ДНФ) так, что может быть указан набор максимальных вне M_{k1}^{fk} интервалов, покрывающих все точки из множества M_{k0}^{fk} .

Рассмотрим любой максимальный интервал $N_L^k \subseteq M_{k1}^{g1}$ произвольной функции $g_k \in \overline{\Phi}_k$ и соответствующую ему элементарную конъюнкцию $L = \overline{x}_{i_1} \& \dots \& \overline{x}_{i_r}$. Вхождение переменных в простую импликанту только с инверсиями доказывается с учетом монотонности функции $\overline{g}_k(\tilde{x})$. Набор $\tilde{\alpha} \in N_L^k$ является допустимым решением, и в этом наборе $\alpha_{i_1} = 0, \dots, \alpha_{i_r} = 0$. Среди всех наборов $\tilde{\alpha} \in N_L^k$ наибольшее значение целевой функции будет достигаться на наборе, в котором $\alpha_j = 1$ для всех j из множества $\{1, 2, \dots, N\} \setminus \{i_1, i_2, \dots, i_r\}$. Назовем такой набор экстремальным.

Если теперь для каждой простой импликанты всех функций из множества Φ_k выбрать экстремальный набор, то в полученном множестве будут содержаться все непротиворечивые решения задачи.

Различные доопределения функции f_k функциями $\varphi_k \in \Phi_k$ отличаются значениями $\varphi_k(\tilde{x})$ на множестве $B^N \setminus \{M_{k0}^{fk} \cup M_{k1}^{fk}\}$, поэтому простые импликанты различных функций g_k из $\overline{\Phi}_k$ могут отличаться рангом. Экстремальная постановка задачи требует из всех простых импликант всех функций $g_k \in \overline{\Phi}_k$ выделить кратчайшие. Для построения таких простых импликант с инверсиями, необходимыми для любых доопределений, можно использовать следующий алгоритм.

Алгоритм построения простых импликант.

- 1: Для каждого набора $\tilde{\alpha} \in M_{k0}^{fk}$ выписать конъюнктивную нормальную форму (КНФ) $K_k(\tilde{\alpha})$, каждая дизъюнкция которой состоит из переменных \overline{x}_i (с инверсиями), таких, что $\alpha_i < \beta_i$ для одного из наборов $\tilde{\beta} \in M_{k1}^{fk}$; КНФ $K_k(\tilde{\alpha})$ будет содержать $m_1^k = |M_{k1}^{fk}|$ дизъюнкций — число наборов в множестве M_{k1}^{fk} .
 - 2: В полученных КНФ $K_k(\tilde{\alpha}_1), \dots, K_k(\tilde{\alpha}_{m_0^k})$, где $m_0^k = |M_{k0}^{fk}|$ (число наборов в M_{k0}^{fk}), раскрыть скобки и выполнить операции поглощения, получая ДНФ $D_1, \dots, D_{m_0^k}$.
 - 3: Записать ДНФ $D_1 \vee \dots \vee D_{m_0^k}$ и выполнить все возможные операции поглощения. Будет получена ДНФ $D(\overline{\Phi}_k)$.
-

Множества $M_{k0}^{f_k}$ и $M_{k1}^{f_k}$, являющиеся частью исходной информации в задаче (21) и содержащие $m_0^k + m_1^k$ двоичных наборов, можно рассматривать как стандартную обучающую информацию задачи Z_k распознавания: в обучающей таблице $T_{m_0^k m_1^k}^k = M_{k0}^{f_k} \cup M_{k1}^{f_k}$, наборы $\tilde{x} \in M_{k0}^{f_k}$ относятся к классу K_1^k допустимых решений задачи (21), а $\tilde{x} \in M_{k1}^{f_k}$ — к классу K_2^k недопустимых решений.

Обозначим через $A_z^k = A_z^k(T_{m_0^k m_1^k}, \tilde{x})$ алгоритм распознавания класса произвольного набора $\tilde{x} \in B^N \setminus T_{m_0^k m_1^k}^k$; пусть A_Z^{k*} — корректный алгоритм:

$$A_z^{k*} \left(T_{m_0^k m_1^k}^k, \tilde{x} \right) = \begin{cases} 1, & \tilde{x} \in K_2^k = B^n \setminus \Omega_k, \\ 0, & \tilde{x} \in K_1^k = \Omega_k, \quad k = 1, 2. \end{cases}$$

Очевидно, что если информация в $T_{m_0^k m_1^k}^k$ достоверна и алгоритм A_z^{k*} относит экстремальный набор \tilde{x}^* , являющийся непротиворечивым решением задачи (21), к классу $K_1^k = \Omega_k$, то \tilde{x}^* является решением задачи

$$\min \sum_{i=1}^n c_i^k x_i / \tilde{x} \in \Omega_k.$$

Пусть алгоритм A_z^k — экстремальный в некотором классе алгоритмов распознавания или построен с применением корректирующих (алгебраических) методов, т.е. является в некотором смысле наилучшим для решения задачи $Z_k(T_{m_0^k m_1^k}, \tilde{x})$.

Подход к решению $mTSP$ как задачи линейного псевдоболевого программирования с частично заданными ограничениями с применением алгоритмов распознавания образов состоит в следующем:

1) при помощи алгоритма находится множество экстремальных наборов $\aleph^k = \{\tilde{x}^*\}$ для задачи (21)–(22);

2) алгоритм A_z^k определяет принадлежность экстремальных наборов из \aleph^k к классу K_1 ; $\aleph_A^k \subseteq \aleph^k$; $\aleph_A^k = \left\{ \tilde{x}^* \in \aleph^k : A \left(T_{m_0^k m_1^k}, \tilde{x}^* \right) = 0 \right\}$;

3) если $\aleph_A^k \neq 0$, то входящий в него экстремальный набор, которому соответствует наибольшее значение целевой функции, объявляется решением задачи;

4) если $\aleph_A^k = 0$, то к $M_{k1}^{f_k}$ добавляются наборы \aleph^k , т.е. $M_{k1}^{f_k} := M_{k1}^{f_k} \cup \aleph^k$, и повторяется п. 1, внутри которого обеспечивается проверка монотонности, обеспечивающая линейность модели.

Замечание 1. Добавление к множеству $M_{k1}^{f_k}$ множества экстремальных наборов \aleph^k равносильно переопределению для некоторых функций алгебры логики верхних нулей единицами.

Линейность задачи $mTSP$ позволила эффективно “сузить” область поиска решения, что обеспечивается указанным алгоритмом (см. [9] по сужающим запросам).

5. Многокритериальные задачи многих коммивояжеров, представленные в канонической форме

Пусть задача $mTSP$ сводится к многокритериальной псевдодобулевой оптимизации с дизъюнктивным ограничением:

$$(23) \quad \begin{cases} \min f_1(\tilde{x}), \min f_2(\tilde{x}), \dots, \min f_m(\tilde{x}), \\ \bigvee_{j=1}^m x_{j_1}^{\sigma_{j_1}} \& \dots \& x_{j_r}^{\sigma_{j_r}} = 1, \\ f_k \in LPS_2(N), \quad k = \overline{1, m}, \quad \tilde{x} \in B^N. \end{cases}$$

Отметим, что задачи псевдодобулевой оптимизации возникают как результат синтеза моделей $mTSP$ на основе индуктивного обобщения или построения логического описания области дедуктивной выводимости в системах, основанных на знаниях.

Необходимо найти паретовское множество \mathcal{P} задачи (23), его логическое описание в виде дизъюнктивной нормальной формы и подходов к выбору решения $\tilde{x}^* \in \mathcal{P}$. Для этого используем необходимое условие принадлежности точки множеству Парето и принцип ветвей и границ.

В задаче $mTSP$ учитывается информация о распределении весов дуг. Выбор прохождения тех или иных дуг для коммивояжера зависит от среднего значения веса дуги, дисперсии (при большой дисперсии преобладают дуги с большими весами). Если в матрице весов вычесть среднее значение веса, получим новую матрицу весов с положительными и отрицательными значениями. Такие матрицы появляются в процессе реализации некоторых алгоритмов TSP . Поэтому необходимое условие принадлежности точки множеству Парето учитывает знаки коэффициентов c_j^k , $j = \overline{1, N}$, $N = n^2$, $k = \overline{1, m}$.

Рассмотрим необходимое условие принадлежности точки множеству Парето в задаче безусловной оптимизации.

Обозначим через P_i множество номеров переменных, имеющих положительный, а через N_i — множество номеров переменных, имеющих отрицательный коэффициент в линейной функции f_i , $i = \overline{1, m}$. Пусть

$$P_0 = P_1 \cap P_2 \cap \dots \cap P_m, \quad N_0 = N_1 \cap N_2 \cap \dots \cap N_m.$$

Лемма 2. Если для задачи безусловной многокритериальной оптимизации $mTSP$

$$(24) \quad \begin{cases} \min f_1(\tilde{x}), \min f_2(\tilde{x}), \dots, \min f_m(\tilde{x}), \\ \tilde{x} \in B^N, \quad f_1, \dots, f_m \in LPS_2(N), \end{cases}$$

множества P_0 и N_0 непусты и точка \tilde{x}^ является паретовской, то она удовлетворяет уравнению*

$$(25) \quad \left(\&_{i \in P_0} x_i \right) \left(\&_{i \in N_0} \bar{x}_i \right) = 1.$$

Доказательство. Пусть $\tilde{\alpha}$ — любая точка, удовлетворяющая уравнению (25). Тогда найдется такое i , что $\alpha_i = 0$ при $i \in P_1 \cap P_2 \cap \dots \cap P_m$ или $\alpha_i = 1$ при $i \in N_1 \cap N_2 \cap \dots \cap N_m$. Заменяя α_i на $\bar{\alpha}_i$, получим точку $\tilde{\alpha}'$ такую, что $f_1(\tilde{\alpha}') < f_1(\tilde{\alpha})$, $f_2(\tilde{\alpha}') < f_2(\tilde{\alpha})$, \dots , $f_m(\tilde{\alpha}') < f_m(\tilde{\alpha})$ и тогда $\tilde{\alpha}$ — не является паретовской точкой. Лемма 2 доказана.

Замечание 2. Если $P_0 \neq \emptyset$ и $N_0 \neq \emptyset$, то необходимыми условиями эффективности точки \tilde{x}^* в задаче (24) являются $\bigwedge_{i \in P_0} x_i = 1$ и $\bigwedge_{i \in N_0} \bar{x}_i = 1$. Если $P_0 \neq \emptyset$ и $N_0 = \emptyset$, то необходимым условием эффективности точки \tilde{x}^* в задаче (24) является $\bigwedge_{i \in P_0} x_i = 1$. Если $P_0 = \emptyset$ и $N_0 \neq \emptyset$, то необходимым условием эффективности точки \tilde{x}^* в задаче (24) является $\bigwedge_{i \in N_0} \bar{x}_i = 1$.

Определение 6. Нижней векторной оценкой допустимого множества X называется вектор

$$\left(\min_{\tilde{x} \in X} f_1(\tilde{x}), \min_{\tilde{x} \in X} f_2(\tilde{x}), \dots, \min_{\tilde{x} \in X} f_m(\tilde{x}) \right).$$

Определение 7. Вектор (a_1, a_2, \dots, a_m) мажорируется вектором (b_1, b_2, \dots, b_m) , если $a_j \leq b_j$ для всех $j = \overline{1, m}$, причем хотя бы для одного j выполняется строго неравенство $a_j < b_j$.

Определение 8. Рекордом называется вектор значений скалярных критериев в некоторой допустимой точке $\tilde{\gamma}$, который не мажорируется никаким другим имеющимся рекордом или нижней векторной оценкой, полученной для какого-либо подмножества допустимого множества решений.

Будем использовать метод ветвей и границ (см. [3, 8] для данного класса задач). Ветвление будем осуществлять путем фиксации значений 0 и 1 переменных x_i , $i = \overline{1, N}$. На каждом шаге ветвления будет происходить измельчение множества B^N и порождение подмножеств-интервалов, подлежащих исследованию.

Интервал подлежит исключению из рассмотрения в следующих случаях:

а) существует рекорд, мажорирующий верхнюю векторную оценку этого интервала;

б) известен другой интервал, нижняя векторная оценка которого мажорирует верхнюю векторную оценку этого интервала. Интервал подлежит ветвлению, если он не подлежит исключению и его верхняя векторная оценка отличается от нижней. Выбор переменной и интервала, подлежащего ветвлению, является эвристическим элементом метода и будет рассмотрен далее.

Рассмотрим задачу (24) с добавлением дизъюнктивных ограничений.

Теорема 5. Пусть в задаче (24) существует непустое множество Парето \mathcal{P} , и к данной задаче добавляется ограничение $\tilde{x} \in \Omega$; $\Omega \neq \emptyset$; $\Omega \subset B^N$, $\Omega \neq B^N$. Для полученной задачи множество $\mathcal{P} \cap \Omega$, если оно не пусто, будет состоять только из паретовских точек.

Доказательство. Пусть $\tilde{x}^* \in \{\mathcal{P} \cap \Omega\}$. Тогда $\tilde{x}^* \in \mathcal{P}$ и не мажорируется ни одной точкой из B^N и, тем более, — ни одной точкой из $\mathcal{P} \cap \Omega$, а так как

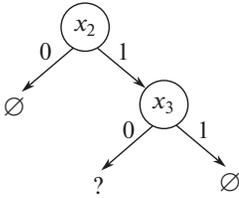


Рис. 2. Начальное дерево.

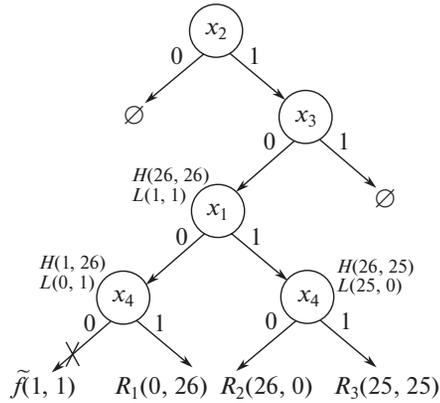


Рис. 3. Результат решения задачи.

$\tilde{x}^* \in \Omega$, то она является допустимой. Таким образом, \tilde{x}^* — немажорируемая допустимая точка, следовательно, является паретовской. Теорема 5 доказана.

Условие (25) не является достаточным. В этом можно убедиться, рассмотрим следующий упрощенный пример:

$$\begin{cases} \min f_1(\tilde{x}) = -25x_1 - x_2 + x_3 + x_4; \\ \min f_2(\tilde{x}) = x_1 - x_2 + x_3 - 25x_4; \\ \tilde{x} \in B^N. \end{cases}$$

Очевидно, что $P_1 \cap P_2 = \{2\}$; $N_1 \cap N_2 = \{3\}$. Условие (25) принимает вид: $x_2\bar{x}_3 = 1$. Этому условию удовлетворяет точка $\tilde{\beta} = (0, 1, 0, 0)$, но она не паретовская: взяв точку $\tilde{\gamma} = (1, 1, 0, 1)$, убеждаемся, что $-25 = f_1(\tilde{\gamma}) < f_1(\tilde{\beta}) = -1$, $-25 = f_2(\tilde{\gamma}) < f_2(\tilde{\beta}) = -1$.

Найдем множество Парето в задаче безусловной многокритериальной оптимизации. Используем необходимое условие, которому должны удовлетворять эффективные точки: $x_2\bar{x}_3 = 1$. Начальное дерево представлено на рис. 2.

Знак “ \emptyset ” указывает на отсутствие эффективных точек в интервале, соответствующем ветви; знак “?” — на необходимость дальнейшего ветвления.

Обозначим через $H(y_1, y_2)$ вектор верхних и через $L(y_1, y_2)$ вектор нижних достижимых оценок функций f_1, f_2 , $R(\alpha, \beta)$ — рекорд.

Результат решения задачи представлен на рис. 3.

Знак “ \times ” указывает на отсечение интервала.

Множество Парето состоит из трех точек $\{0101, 1100, 1101\}$, являющихся рекордными (R_1, R_2, R_3) , и имеет логическое описание: $\mathcal{P} = \left\{ \tilde{x} : x_1x_2\bar{x}_3 \vee x_2\bar{x}_3x_4 = 1 \right\}$.

Замечание 3. Условие $\tilde{x} \in \mathcal{P} \cap \Omega$, как следует из теоремы 5, является достаточным для того, чтобы точка \tilde{x} была паретовской в задаче с ограничением $\tilde{x} \in \Omega$ при $\mathcal{P} \cap \Omega \neq \emptyset$. Однако это условие не является необходимым.

Действительно, в задаче

$$\left\{ \begin{array}{l} \min f_1(\tilde{x}) = -25x_1 - x_2 + x_3 + x_4; \\ \min f_2(\tilde{x}) = x_1 - x_2 + x_3 - 25x_4; \\ \tilde{x} \in \Omega = \{\tilde{x} : \bar{x}_1 x_2 = 1\} \subset B^N \end{array} \right.$$

паретовскими являются точки $\{0100, 0101\}$ со значениями векторов критериев $(-1, -1)$ и $(0, -26)$ соответственно, причем точка $\{0100\}$ не принадлежит множеству $\mathcal{P} \cap \Omega$.

Рассмотрим варианты выбора интервалов и переменных для ветвления.

От последовательности выбора интервалов и переменных для ветвления зависит скорость нахождения решения задачи. Стратегии ветвления являются эвристиками, например (возможны другие):

1) разбиению по переменной с номером i подвергается тот интервал множества допустимых решений, конъюнкция которого не содержит литерала переменной с номером i , и изменение этой переменной с единицы на нуль обеспечивает одновременное уменьшение как можно большего числа скалярных критериев;

2) разбиению подвергается тот интервал, для которого является максимальной следующая мера различия между верхней $H = (h_1, \dots, h_m)$ и нижней $L = (l_1, \dots, l_m)$ его векторными оценками:

$$D(H, L) = \min_{1 \leq j \leq m} \left(\frac{h_j - l_j}{M_j - \mu_j} \right),$$

где $M_j = \max_{\tilde{x} \in \Omega} f_j(\tilde{x})$; $\mu_j = \min_{\tilde{x} \in \Omega} f_j(\tilde{x})$; $M_j - \mu_j > 0$, поскольку в противном случае критерий f_j может быть исключен из рассмотрения.

Можно привести пример задачи многокритериальной псевдодобулевой оптимизации, для которой процесс принятия решения по изложенному методу будет близок к полному перебору. В расчете на такие ситуации возможен приближенный подход к решению, суть которого состоит в следующем.

Пусть заданы значения $\varepsilon_1, \dots, \varepsilon_m \in \mathbb{R}^+$. Будем говорить, что два вектора (a_1, a_2, \dots, a_m) и (b_1, b_2, \dots, b_m) являются $\tilde{\varepsilon}$ -равными, если $|a_i - b_i| \leq \varepsilon_i$, $i = 1, m$.

Если верхняя и нижняя векторные оценки некоторого интервала, полученного при ветвлении, $\tilde{\varepsilon}$ -равны, то такой интервал называется $\tilde{\varepsilon}$ -интервалом.

Если нижняя векторная оценка некоторого $\tilde{\varepsilon}$ -интервала мажорируется рекордом или нижней векторной оценкой другого интервала, то такой $\tilde{\varepsilon}$ -интервал подлежит исключению.

Совокупность немажорируемых $\tilde{\varepsilon}$ -интервалов вместе с рекордами дает приближение к искомому паретовскому множеству. Логическое описание паретовского множества \mathcal{P} получается обратным проходом по ветвям деревьев ветвлений, листья которых соответствуют немажорируемым элементам.

Таким образом, предложены методы решения многоэкстремальных задач, представленных в канонической форме. Показано, как использование метода

ветвей и границ решения таких задач позволяет строить логическое описание паретовского множества.

6. Заключение

В статье представлен класс задач для многих коммивояжеров, который приведен или достаточно просто приводится к моделям псевдодвулевой условной оптимизации с ограничениями в виде дизъюнктивных нормальных форм. Предложен алгоритм, основанный на кластеризации графа, согласно которому на каждом кластере решаются задачи для одного коммивояжера, представленные в канонической форме скалярной псевдодвулевой оптимизации и дизъюнктивными ограничениями. Указывается на возможность формирования ДНФ ограничений с помощью обучения по прецедентной информации, а тем самым — синтез областей допустимости решений в продукционных системах моделей псевдодвулевой оптимизации, соответствующих $mTSP$. Предложен подход с частично заданными ограничениями, основанный на применении алгоритмов распознавания образов. Методика распространяется на многокритериальный случай задач $mTSP$.

Прикладные аспекты применения полученных результатов требуют учета всех компонент задачи $mTSP$ в сложных сетях.

Разработка приближенных алгоритмов выбора маршрутов в сложных сетях может быть связана с учетом знаний о свойствах структуры сети, ее сложности, наличием ограничений, предписаний, условий достижимости, числа агентов-коммивояжеров.

Необходимо будет учитывать специфику задач маршрутизации в сложных сетях, которая, в отличие от классической теории графов, связана с рядом уникальных задач: о нахождении метрических характеристик сложных сетей; поиск минимального (максимального) среднего пути в сети; коэффициентов кластеризации; изучения информационных потоков в сети; выявление критичных мест в сети; определения кластеров; выявление блоков, компонент, мостов, точек сочленения (перемычек).

Методология разработки алгоритма решения задач маршрутизации может быть основана на формировании по исходной сложной сети более простой (относительно реализации алгоритмов маршрутизации) по своей структуре сети. Построение рациональных решений $mTSP$ на сетях большой размерности реализуется по схеме алгоритма $AmTSP$ для m коммивояжеров. В случае разных интересов агентов приходим к многокритериальной задаче псевдодвулевой оптимизации с дизъюнктивными ограничениями. Здесь возможны игровые модели.

Дальнейшие исследования также связаны с обучением агентов-коммивояжеров, их автономностью и организацией обмена прецедентной информацией между агентами (системами управления). При этом в многоагентной системе (МАС) $mTSP$ должны сочетаться задачи выбора решения; управления; распределения ресурсов; синтеза сети (вершин-источников ресурсов); устойчивости сети в зависимости от удаления вершины, дуги или некоторого маршрута;

кластеризации сети в зависимости от изменяющихся условий; обмена информацией между агентами; потоковые задачи; задачи прокладки кратчайших путей и замкнутых маршрутов.

СПИСОК ЛИТЕРАТУРЫ

1. *Антамошкин А.А., Масич И.С.* Поисквые алгоритмы псевдоболевой оптимизации // Системы управления, связи и безопасности. 2016. № 1. С. 103–145.
2. *Германчук М.С., Козлова М.Г., Лукьяненко В.А.* Знаниеориентированные модели маршрутизации многих коммивояжеров // Интеллектуализация обработки информации // Тез. докл. 13-й Междунар. конф., Москва, 2020. С. 352–355.
3. *Донской В.И., Башта А.И.* Дискретные модели принятия решений при неполной информации. Симферополь: Таврия, 1992.
4. *Донской В.И.* Задачи псевдоболевой оптимизации с дизъюнктивным ограничением // Журн. выч. матем. и матем. физ. 1994. № 4. С. 461–472.
5. *Donskoy V., Perekhod I.* Multiple Criteria Models with the Linear Pseudoboolean Functions and Disjunctive Restrictions / Fandel G., Gal T. (eds). Multiple Criteria Decision Making. Lecture Notes in Economics and Mathematical Systems. V. 448. Berlin–Heidelberg: Springer, 1997. https://doi.org/10.1007/978-3-642-59132-7_2
6. *Donskoy V.I.* A Synthesis of Pseudo-Boolean Empirical Models by Precedential Information // Bulletin SUSU MMCS. 2018. V. 11. No. 2. P. 96–107.
7. *Козлова М.Г.* Знаниеориентированные модели принятия решений // Ученые записки СГУ. 1998. № 7 (46). С. 76–83.
8. *Козлова М.Г.* Многокритериальные модели принятия решений с линейными псевдоболевыми функциями и дизъюнктивным ограничением // Искусственный интеллект. 2000. № 2. С. 67–73.
9. *Козлова М.Г.* Синтез сужающих запросов // Динамические системы. 2000. Вып. 16. С. 208–211.
10. *Масич И.С.* Поисквые алгоритмы условной оптимизации: монография. Красноярск: СибГАУ. 2013.
11. *Меламед И.И., Сергеев С.И., Сигал И.Х.* Задача коммивояжера. Вопросы теории // АиТ. 1989. № 9. С. 3–33.
Melamed I.I., Sergeev S.I., Sigal I.K. The Traveling Salesman Problem. Issues in Theory // Autom. Remote Control. 1989. V. 50. No. 9. P. 1147–1173.
12. *Меламед И.И., Сергеев С.И., Сигал И.Х.* Задача коммивояжера. Точные методы // АиТ. 1989. № 10. С. 3–29.
Melamed I.I., Sergeev S.I., Sigal I.K. The Traveling Salesman Problem. Exact Methods // Autom. Remote Control. 1989. V. 50. No. 10. P. 1303–1324.
13. *Меламед И.И., Сергеев С.И., Сигал И.Х.* Задача коммивояжера. Приближенные алгоритмы // АиТ. 1989. № 11. С. 3–26.
Melamed I.I., Sergeev S.I., Sigal I.K. The Traveling Salesman Problem. Approximate Algorithms // Autom. Remote Control. 1989. V. 50. No. 11. P. 1459–1479.
14. *Германчук М.С., Лемтюжникова Д.В., Лукьяненко В.А.* Метаэвристические алгоритмы для многоагентных задач маршрутизации // Проблемы управления. 2020. Т. 6. С. 3–13.
15. *Grana Y., Hammer P.L.* Boolean Functions: Theory, Algorithms and Applications. – N.Y.: Cambridge University Press, 2011.

16. *Hammer P.L., Rudeanu S.* Boolean Methods in Operations Research and Related Areas. Berlin–Heidelberg–N.Y.: Springer-Verlag, 1968.
17. *Foldes S., Hammer P.L.* Disjunctive and Conjunctive Normal Forms of Pseudo-Boolean Functions // *Discrete Appl. Math.* 2000. No. 107. P. 1–26.
18. *Boros E., Hammer P.L.* Pseudo-Boolean Optimization // *Discrete Appl. Math.* 2002. No. 123. P. 155–225.
19. *Hammer P.L.* Pseudo-Boolean Remarks on Balanced Graphs // *Int. series of Numerical Math.* 1977. No. 36. P. 69–78.
20. *Ebenegger Ch., Hammer P.L., de Werra D.* Pseudo-Boolean Functions and Stability of Graphs // *Annals of Discrete Math.* 1984. No. 19. P. 83–97.
21. *Журавлев Ю.И.* О локальных алгоритмах над дизъюнктивными нормальными формами // *Докл. АН СССР.* 1979. Т. 245. № 2. С. 289–292.
22. *Журавлев Ю.И., Коган А.Ю.* Реализация булевых функций с малым числом нулей дизъюнктивными нормальными формами и смежные задачи // *Докл. АН СССР.* 1985. Т. 285. № 4. С. 795–799.
23. *Hammer P.L., Rudeanu S.* Pseudo-Boolean Methods for Bivalent Programming // *Lecture Notes in Math.* Sept. 2–7, 1966.
24. *Сапоженко А.А.* О поиске максимального верхнего нуля монотонных функций на ранжированных множествах // *Журн. выч. матем. и матем. физ.* 1991. Т. 31. № 12. С. 1871–1884.
25. *Германчук М.С., Козлова М.Г., Лукьяненко В.А.* Задачи практической маршрутизации // *Анализ, моделирование, управление, развитие социально-экономических систем.* Сб. науч. тр. XI Междунар. школы-симпозиума АМУР, 2017. С. 116–120.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 16.03.2021

Принята к публикации 30.06.2021

© 2021 г. В.В. ГЕППЕНЕР, д-р техн. наук (gerpener@mail.ru)
(Санкт-Петербургский государственный электротехнический
университет “ЛЭТИ” им. В.И. Ульянова (Ленина)),
Б.С. МАНДРИКОВА (555bs5@mail.ru)
(Институт космических исследований и распространения
радиоволн ДВО РАН, Камчатский край, село Паратунка)

ОБНАРУЖЕНИЕ И ИДЕНТИФИКАЦИЯ АНОМАЛЬНЫХ ЭФФЕКТОВ В СЛОЖНОМ СИГНАЛЕ¹

Предложен метод обнаружения и идентификации аномальных эффектов в сигнале сложной структуры, основанный на нелинейных аппроксимирующих схемах в словаре вейвлет-пакетов. С учетом свойств частотно-временного окна вейвлет-преобразования вводится адаптивный порог. Повышение эффективности обнаружения разных типов структур достигается путем применения суперпозиции конструкций вейвлет-преобразования. На примере данных нейтронных мониторов показано, что метод позволяет подавить шум и идентифицировать аномальные эффекты разной формы и длительности. Результаты подтвердили эффективность предлагаемого метода для обнаружения малоамплитудных Форбуш-эффектов в вариациях космических лучей.

Ключевые слова: анализ данных, нелинейная аппроксимация, вейвлет-пакеты, вариации космических лучей.

DOI: 10.31857/S0005231021100056

1. Введение

В настоящее время учеными активно ведутся исследования, связанные с разработкой методов обработки и анализа сложных нелинейных данных [1, 2]. Основная проблема таких исследований заключается в отсутствии априорной информации о полезном сигнале, сложной структуре изучаемых нестационарных данных и наличии высокого уровня шума [3]. Острая необходимость в создании методов анализа сложных данных возникает при проведении ряда фундаментальных и прикладных научных исследований в таких областях, как биомедицина, геофизика, космофизика и т.д.

Космофизические исследования являются одним из ярких примеров необходимости создания математических моделей и методов, способных с приемлемой точностью и, что не менее важно, скоростью выполнять анализ геофизических данных и прогноз космической погоды [4]. Известно, что аномальные события в околоземном космическом пространстве отрицательно,

¹ Работа выполнена в рамках Государственного задания по теме “Физические процессы в системе ближнего космоса и геосфер при солнечных и литосферных воздействиях” (2021–2023 гг.), регистрационный номер АААА-А21-121011290003-0.

а в некоторых случаях губительно отражаются на технических и техногенных объектах, а также на здоровье и жизни людей [5]. Изучение динамики космических лучей занимает широкую область в сфере солнечно-земной физики, а также является одним из факторов мониторинга и прогноза космической погоды. Исследование вариаций космических лучей проводят по данным мировой сети нейтронных мониторов [6]. Вариации космических лучей имеют сложную структуру, в которой находят отражение события, возникающие в атмосфере, галактике, а также имеющие внегалактическое происхождение [7]. Помимо полезной информации, в данных нейтронных мониторов содержится высокий уровень шума, включающий помехи природного и техногенного характера [8]. Периодические вариации космических лучей соответствуют регулярному ходу, аномальные (спорадические) эффекты представляют собой Форбуш-эффекты и сильные наземные протонные возмущения (GLE-события). Форбуш-эффекты могут иметь разную форму и длительность и проявляться в виде резкого снижения уровня, его восстановления, а также мелкомасштабных изменений перед началом магнитных бурь. Особый интерес представляют своевременное обнаружение и идентификация малоамплитудных Форбуш-эффектов, служащих предикторами магнитных бурь [9, 10].

Спектр применяемых в настоящее время методов и подходов к анализу данных космических лучей весьма широк. Например, метод глобальной съемки [9] или метод кольца станций [10], разработанные учеными ИЗМИРАН, позволяют оценить свойства углового распределения космических лучей, но точность методов напрямую зависит от наличия данных определенно расположенных станций, что, к сожалению, не всегда выполнимо. Также данные методы включают сложные трудоемкие расчеты и не автоматизируемы. Известны весьма успешные попытки применения методов машинного обучения для анализа данных нейтронных мониторов [11–13]. Например, в публикациях [11, 12] используются глубокие нейронные сети, позволяющие с учетом мощностей современных ЭВМ в оперативном режиме выполнять анализ данных космических лучей. Но весомым недостатком данных методов является потребность периодической перенастройки параметров системы (например, переобучения нейронной сети) в связи с существенной зависимостью методов от внешних условий. Данный фактор сильно влияет на качество и оперативность получаемых результатов.

Учитывая указанные недостатки, в статье предложен подход, основанный на построении нелинейных адаптивных аппроксимирующих схем. Поскольку распределение вероятности сигнала космических лучей имеет сложную форму, линейная аппроксимация не является эффективной и лучшие результаты дают нелинейные пороговые оценки [14, 15]. В качестве аппроксимирующих функций в статье используются базисы вейвлет-пакетов. Известно, что вейвлет-фильтрация позволяет эффективно детектировать структуры сложного сигнала и подавить шум [11–15]. Разные вейвлет-базисы по-разному аппроксимируют сигнал, поэтому выбор наилучшего базиса, в смысле выделения определенных структур, обеспечивает эффективное решение

поставленной задачи. Но необходимо учитывать, что если сигнал содержит различные типы структур, локализованные в разные моменты времени, то нельзя построить базис, адаптированный ко всем структурам. В этом случае надо использовать большие словари базисов [16], например, как предложено в [15], можно расширить класс ортогональных функций словарями линейно независимых функций. Эффективный результат в этом случае дает аппроксимация с преследованием [17]. Но большая вычислительная сложность данного метода делает его малоэффективным. Алгоритмы согласованного преследования [18] с использованием стратегии “жадности” позволяют оптимизировать процесс построения базиса и получить достаточно точные аппроксимации, но в случае высокого уровня шума такое решение задачи, к сожалению, не дает хороших результатов. В этом случае энергия сигнала мала относительно энергии шума, поэтому такая оценка дает низкий порог и его применение не позволяет подавить весь шум. Поскольку в настоящее время отсутствует строгий математический аппарат построения оценок для сигналов с такими свойствами [15], в статье предлагается метод, основанный на эвристическом подходе. В статье показано, что использование большего порога увеличивает риск, но позволяет получить более точные оценки. С учетом свойств частотно-временного окна вейвлет-преобразования в статье вводится адаптивный порог. Повышение эффективности обнаружения разных типов структур достигается путем применения суперпозиции непрерывного вейвлет-преобразования и нелинейных адаптивных аппроксимаций в словаре вейвлет-пакетов. Представленные результаты подтверждают эффективность предлагаемого метода для задачи обнаружения малоамплитудных Форбуш-эффектов разной формы и длительности в вариациях космических лучей.

2. Предлагаемый метод

2.1. Построение нелинейных аппроксимирующих схем в словаре вейвлет-пакетов

При нелинейной аппроксимации сигнал $f \in H$ (H — пространство Гильберта) аппроксимируется M векторами, адаптивно выбранными из ортонормированного базиса $\mathfrak{B} = \{g_m\}_{m \in N}$ пространства H [15]:

$$(1) \quad f_M = \sum_{m \in I_M} \langle f, g_m \rangle g_m.$$

Тогда погрешность аппроксимации есть

$$\epsilon[M] = \|f - f_M\|^2 = \sum_{m \notin I_M} |\langle f, g_m \rangle|^2.$$

Очевидно, минимизация погрешности $\epsilon[M]$ достигается выбором абсолютных значений вейвлет-коэффициентов $|\langle f, g_{m_k} \rangle|$, наилучшим образом корре-

лирующих с сигналом. Аппроксимация может быть получена путем применения пороговой функции, и соотношение (1) примет вид

$$(2) \quad f_M = \sum_{k=1}^M f_{\mathfrak{B}}^r[k] g_{m_k},$$

где $f_{\mathfrak{B}}^r[k]$ — пороговая функция.

Для оптимизации представления (2) базис \mathfrak{B} выбирают адаптивно сигналы, используя, например, словари вейвлет-пакетов, которые позволяют построить наилучшие аппроксимации сигналов конечной длины путем минимизации вогнутой функции стоимости [15]. В качестве функции стоимости может быть использована вогнутая сумма Шура [19]:

$$C(f, \mathfrak{B}^\lambda) = \sum_{m=1}^N \Phi \left(\frac{|(f, g_m^\lambda)|^2}{\|f\|^2} \right), \quad \Phi(x) = -x \ln x,$$

и наилучший базис \mathfrak{B}^α может быть определен как $C(f, \mathfrak{B}^\alpha) = \min_{\lambda \in \Lambda} C(f, \mathfrak{B}^\lambda)$ (Λ — словари базисов). Каждый узел дерева вейвлет-пакетов соответствует пространству W_j^p , которое определяет ортонормированный базис \mathfrak{B}_j^p . Пространство W_j^p разбивается на ортогональные подпространства [15] $W_j^p = W_{j+1}^{2p} \oplus W_{j+1}^{2p+1}$, и стоимость f в семействе $M \leq N$ определяется как $C(f, \mathfrak{B}) = \sum_{m=0}^{M-1} \Phi \left(\frac{|(f, g_m)|^2}{\|f\|^2} \right)$. Тогда наилучший базис O_j^p пространства W_j^p есть базис [19]:

$$(3) \quad O_j^p = \begin{cases} O_{j+1}^{2p} \cup O_{j+1}^{2p+1}, & \text{если } C(f, O_{j+1}^{2p}) + C(f, O_{j+1}^{2p+1}) < C(f, \mathfrak{B}_j^p), \\ \mathfrak{B}_j^p, & \text{если } C(f, O_{j+1}^{2p}) + C(f, O_{j+1}^{2p+1}) \geq C(f, \mathfrak{B}_j^p). \end{cases}$$

Рекурсивное вычисление лучших базисов (3) при движении снизу вверх по дереву позволяет найти наилучший базис вейвлет-пакетов для сигнала f .

Поскольку не знаем распределения вероятности сигнала, задача состоит в оценке $f \in \Theta$ по зашумленным данным: $X[m] = f[m] + W[m]$, где $X[m]$ — регистрируемые данные, $f[m]$ — сигнал, $W[m]$ — шум. В этом случае риск оценки $\tilde{F} = DX$ есть $r(D, f) = E \left\{ \|DX - f\|^2 \right\}$ (D — оператор решения, E — математическое ожидание). Цель авторов — минимизировать максимальный риск (минимаксный риск) [15]:

$$r_n(\Theta) = \inf_{D \in O_n} \sup_{f \in \Theta} E \left\{ \|DX - f\|^2 \right\}.$$

Донохо и Джонстон в [20] показали, что диагональные пороговые оценки как в случае белого, так и цветного шума имеют риск, близкий к нижней границе минимаксного риска:

$$(4) \quad \tilde{F} = DX = \sum_{m=0}^{N-1} d_m (X_{\mathfrak{B}}[m]) g_m = \sum_{m=0}^{N-1} a[m] X_{\mathfrak{B}}[m] g_m \quad \text{при } 0 \leq m \leq N,$$

$X_{\mathfrak{B}}[m] = \langle X, g_m \rangle$, $f_{\mathfrak{B}}[m] = \langle f, g_m \rangle$, $W_{\mathfrak{B}}[m] = \langle W, g_m \rangle$, $a[m]$ нелинейный проектор:

$$a[m] = \begin{cases} 1, & \text{если } |f_{\mathfrak{B}}[m]| \geq \sigma, \\ 0, & \text{если } |f_{\mathfrak{B}}[m]| < \sigma. \end{cases}$$

Но необходимо учитывать, что риск (4) может быть связан с погрешностью аппроксимации f в базисе \mathfrak{B} :

$$f_M = \sum_{|f_{\mathfrak{B}}[m]| \geq \sigma} f_{\mathfrak{B}}[m] g_m, \quad \epsilon_n[M] = \|f - f_M\|^2 = \sum_{|f_{\mathfrak{B}}[m]| < \sigma} |f_{\mathfrak{B}}[m]|^2.$$

Риск оценивается как [15]:

$$r_p(f) = \sum_{m=0}^{N-1} \min(|f_{\mathfrak{B}}[m]|^2, \sigma^2) = \epsilon_n[M] + M\sigma^2.$$

Поэтому задача выбора базиса важна и определяет риск получаемой оценки $r_p(f)$. Также необходимо учитывать, что применение диагональных пороговых оценок (4) требует знания дисперсии сигнала, что не всегда реализуемо на практике.

В случае жесткой пороговой обработки

$$d_m(x) = \rho_T(x) = \begin{cases} x, & \text{если } |x| > T, \\ 0, & \text{если } |x| \leq T, \end{cases}$$

для минимизации риска порог T выбирается так, что есть большая вероятность того, что он больше максимального уровня коэффициентов шума $|W_{\mathfrak{B}}[m]|$. В [20] доказано, что порог $T = \sigma\sqrt{2\ln N}$ дает оптимальную диагональную оценку в базисе \mathfrak{B} (см. (4)). Дисперсия шума σ^2 может быть оценена [20]: $\bar{\sigma}^2 = \frac{M_X}{0,6745}$, где M_X — медиана множества $\{|\langle X, \psi_{j,m} \rangle|\}_{0 \leq m < N/2}$, $\psi_{j,m}$ — базис пространства W_j^p .

Таким образом, если используем вейвлет-пакеты и выбираем порог T выше максимальной амплитуды коэффициентов шума $|\langle W, \psi_{j,m} \rangle|$, то коэффициенты $|\langle X, \psi_{j,m} \rangle|$, превышающие данный порог, с высокой вероятностью будут детектированы. Известно [15], что на мелких масштабах такие коэффициенты возникают в окрестностях резких изменений сигнала. Поэтому, выбирая такой подход, с большой вероятностью сохраняем коэффициенты в окрестностях аномальных значений анализируемой функции. Также при выборе порога необходимо учесть, что в соответствии со свойствами частотно-временного окна вейвлет-преобразования [14] детектирующая способность вейвлетов уменьшается с ростом масштаба. Поэтому логично адаптировать порог к масштабу. Получаем следующий алгоритм построения аппроксимирующих схем в словаре базисов вейвлет-пакетов:

1. Выполняем разложение функции f в вейвлет-пакеты:

$$W_j^0 : W_j^0 = \bigoplus_{i=0}^I W_{j_i}^{p_i}, \quad \left\{ \Psi_{j_i}^{p_i}(2^{j_i}t - m) \right\}_{m \in N} \quad \text{есть базис пространства } W_{j_i}^{p_i};$$

2. Определяем ветви дерева, соответствующие структурным компонентам сигнала, базис $O_{j_i}^p$ пространства $W_{j_i}^p$ есть базис:

$$O_{j_i}^p = \begin{cases} \left\{ \Psi_{j_i}^p (2^{j_i} t - m) \right\}_{m \in Z}, \\ \text{если } \sum_{m \in I_M^p} |\langle X, \Psi_{j_i, m}^p \rangle|^2 \geq \sum_{m \in I_M^{2p}} |\langle X, \Psi_{j_i+1, m}^{2p} \rangle|^2 + \sum_{m \in I_M^{2p+1}} |\langle X, \Psi_{j_i+1, m}^{2p+1} \rangle|^2, \\ \left\{ \Psi_{j_i+1}^{2p} \right\}_{m \in Z} \cup \left\{ \Psi_{j_i+1}^{2p+1} \right\}_{m \in Z}, \\ \text{если } \sum_{m \in I_M^p} |\langle X, \Psi_{j_i, m}^p \rangle|^2 < \sum_{m \in I_M^{2p}} |\langle X, \Psi_{j_i+1, m}^{2p} \rangle|^2 + \sum_{m \in I_M^{2p+1}} |\langle X, \Psi_{j_i+1, m}^{2p+1} \rangle|^2, \end{cases}$$

где множество индексов I_M^l , $l = P, 2P, 2P + 1$ определяются так:

$$\text{индекс } m \in I_M^l, \quad \text{если } |\langle X, \Psi_{j_i, m}^l \rangle| \geq T_{j_i}, \quad \text{порог } T_{j_i} = K \cdot \sigma_{j_i}^l,$$

$$\sigma_{j_i}^l = \sqrt{\frac{1}{l} \sum_{m=1}^l \left(\langle X, \Psi_{j_i, m}^l \rangle - \overline{\langle X, \Psi_{j_i, m}^l \rangle} \right)^2}.$$

Выполненные в статье оценки показали, что получаемые пороги $T_{j_i} = K \cdot \sigma_{j_i}^l$ при $K = 2,5$ превышают оптимальный порог $T = \sigma \sqrt{2 \ln N}$, что увеличивает риск. Но учитывая, что пороговые оценки в словаре вейвлет-пакетов дают почти минимаксный риск для сигналов с ограниченной вариацией, можно с уверенностью предположить, что потери при используемом подходе невелики.

На основе алгоритма получаем

$$(5) \quad \tilde{f}(t) = \sum_{(j_i, m) \in Q_M^l} \langle X, \psi_{j_i, m} \rangle \psi_{j_i, m}(t) = \sum_{(j_i, m) \in Q_M^l} v_{j_i, m} \psi_{j_i, m}(t),$$

где

$$(j_i, m) \in Q_M^p : j_i \in O_{j_i}^p, \quad m \in I_M^l.$$

На рис. 1 представлен результат построения аппроксимирующей схемы с применением вейвлета Койфлет 2. Построение выполнялось по данным нейтронного монитора станции Инувик (68.35N, -133.72W) за период высокой солнечной активности (2013–2015 гг.). Цифрами в узлах дерева показаны значения функции стоимости. Выбранные узлы показаны серым цветом. Анализ показывает, что структурные составляющие сигнала включают аппроксимирующие (левая ветка дерева) и детализирующие (правая ветка дерева) компоненты 4-го уровня разложения.

На рис. 2 представлены выделенные детализирующие компоненты дерева. В верхней части рис. 2 показан исходный сигнал. Анализ результатов подтверждает эффективность метода, который детектировал особенности малой

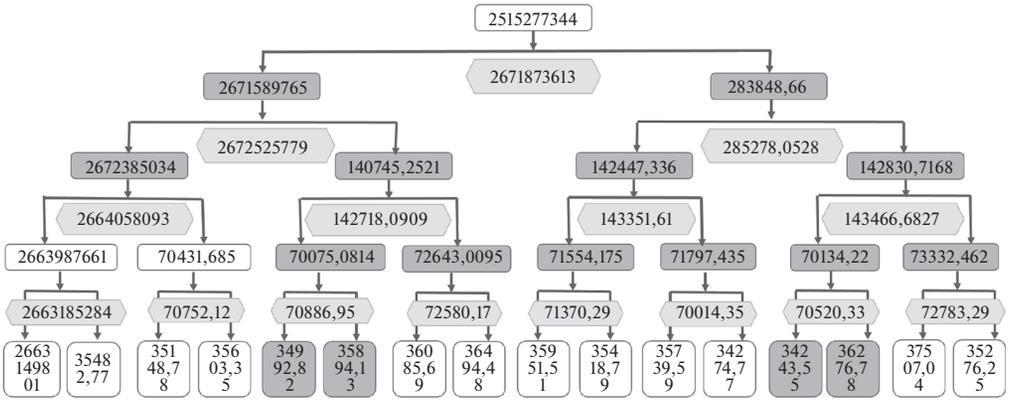


Рис. 1. Построенное дерево вейвлет-пакета с применением вейвлета Койфлет 2, выделенные компоненты обозначены серым цветом.

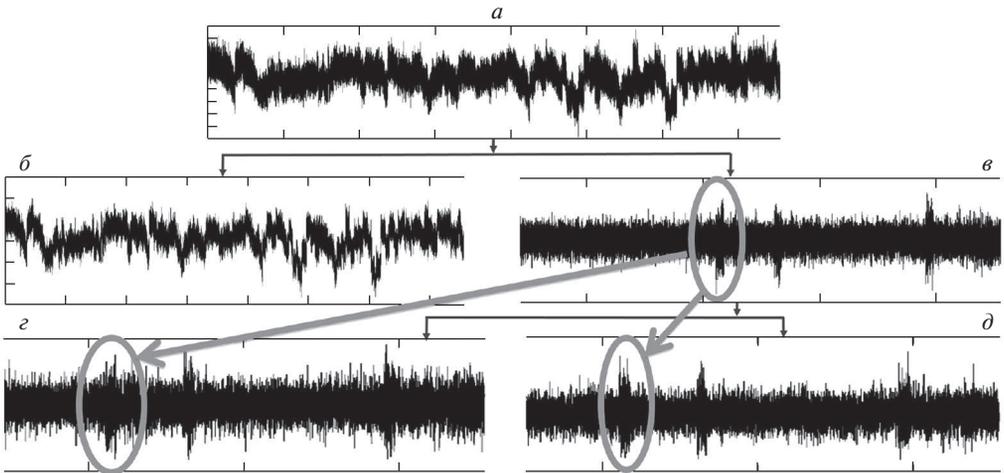


Рис. 2. *a* — Отображение в пространство W_L^0 ; *б* — отображение в пространство W_{L+1}^0 ; *в* — отображение в пространство W_{L+1}^1 ; *г* — отображение в пространство W_{L+2}^2 ; *д* — отображение в пространство W_{L+2}^3 .

амплитуды, незначительно превышающие фоновые вариации космических лучей и практически неразличимые в шуме.

Пример применения метода для обнаружения Форбуш-эффекта малой амплитуды показан на рис. 3. По данным Центра прогноза космической погоды ИЗМИРАН [16] Форбуш-эффект малой амплитуды произошел 3 марта (отмечено вертикальной пунктирной линией). На рис. 3,а и 3,б изображены соответственно данные нейтронного монитора станции Москва за 1–5 марта 2014 г. и их вейвлет-спектр (применялось непрерывное вейвлет-преобразование). На рис. 3,в и 3,г показана выделенная компонента и ее вейвлет-спектр. Анализ результатов показывает, что из-за наличия высокого

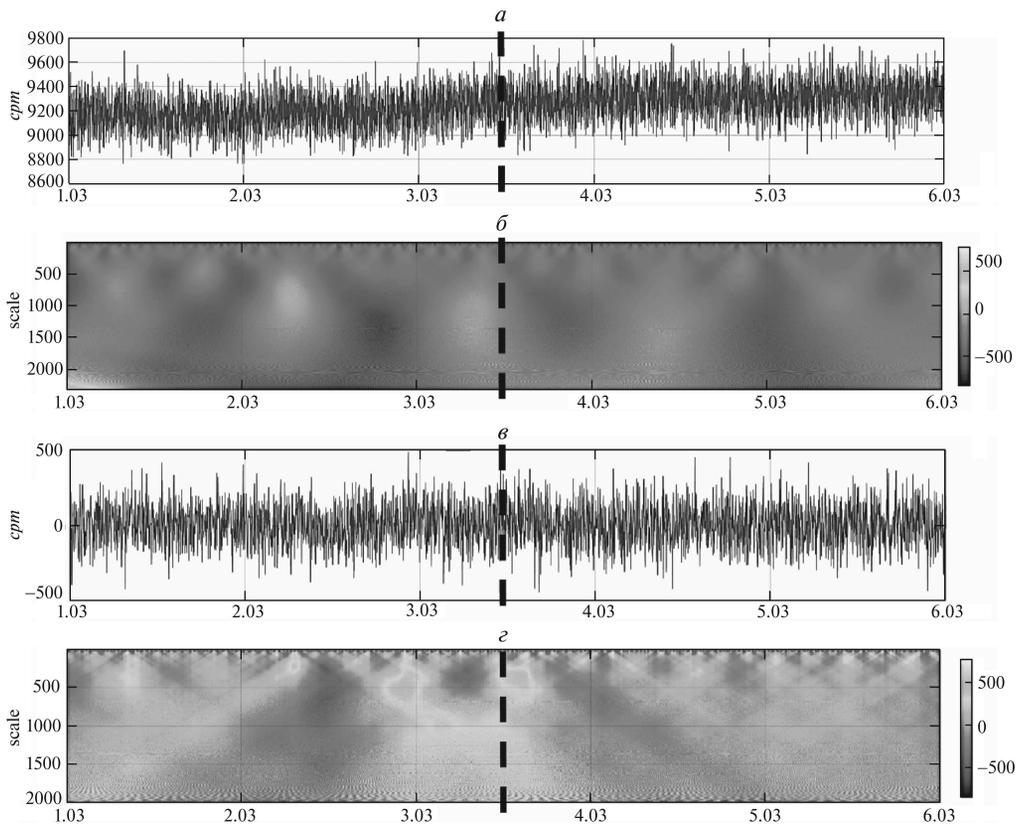


Рис. 3. *a* — Регистрируемые данные нейтронного монитора ст. Москва, *б* — вейвлет-спектр сигнала, *в* — выделенная компонента дерева, *г* — вейвлет-спектр компоненты.

уровня шума обнаружить Форбуш-эффект в зашумленных данных не удалось (см. рис. 3,*б*). Применение метода позволило детектировать Форбуш-эффект в сигнале (рис. 3,*г*), что подтверждает эффективность предлагаемого подхода.

На рис. 4 показан период, содержащий несколько спорадических эффектов. По данным ИЗМИРАН [21] Форбуш-эффекты произошли 18, 20, 25, 29 и 31 марта. Периоды Форбуш-эффектов отмечены на рис. 4 овалами. Результат применения метода (рис. 4,*б*) показывает его эффективность для задачи обнаружения спорадических эффектов. Отметим, что по данным обработки в анализируемый период Форбуш-эффект также произошел 14 марта (рис. 4,*б*). Но в базе данных ИЗМИРАН [21] он не указан. По данным прогноза космической погоды [22, 23] в связи с пересечением секторной границы межпланетного магнитного поля (ММП) и из-за прихода ускоренного потока от слабоконтрастной корональной дыры вертикальная компонента ММП 13 марта приняла значение $Bz = -10$ нТл, скорость солнечного ветра увеличилась от 300 до 600 км/с и 14 марта превысила 650 км/с. По указанным

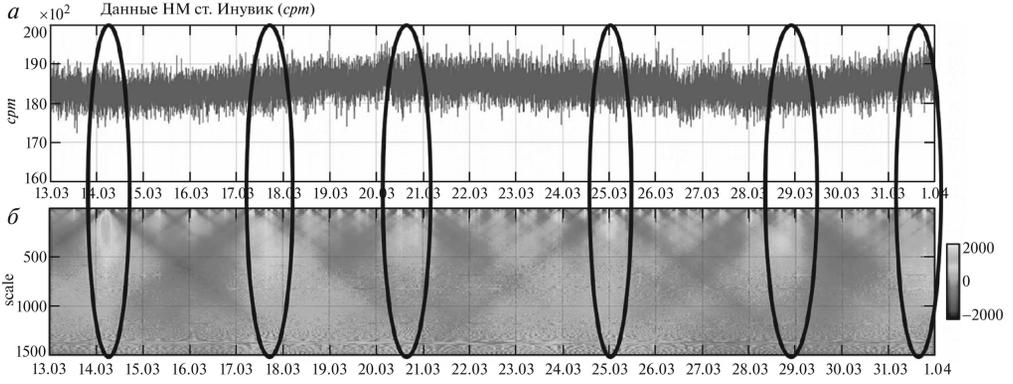


Рис. 4. *a* — Данные нейтронного монитора ст. Инувик, *b* — вейвлет-спектр выделенной компоненты сигнала.

факторам космической погоды можно сделать вывод о высокой возможности обнаружения спорадических эффектов в космических лучах, что подтверждает эффективность метода.

2.2. Алгоритм обнаружения аномальных эффектов на основе суперпозиции конструкций вейвлет-преобразования

По результатам экспериментов спорадические эффекты в вариациях космических лучей имеют многомасштабную структуру и детектируются в разных компонентах дерева вейвлет-пакетов. В этом случае эффективность их обнаружения может быть повышена на основе совмещения нелинейных аппроксимирующих схем с непрерывным вейвлет-преобразованием. Учитывая свойства частотно-временного окна вейвлет-преобразования [14], лучший результат в этом случае даст применение адаптивного порога, зависящего от масштаба. Предлагаемый алгоритм представлен.

1. Построение аппроксимирующей схемы в базе вейвлет-пакетов (см. (5)):

$$\tilde{f}(t) = \sum_{(j_i, m) \in Q_M^l} \langle X, \psi_{j_i, m} \rangle \psi_{j_i, m}(t) = \sum_{(j_i, m) \in Q_M^l} v_{j_i, m} \psi_{j_i, m}(t),$$

где $(j_i, m) \in Q_M^p : j_i \in O_{j_i}^p, m \in I_M^l$.

2. Выполнение непрерывного вейвлет-преобразования и применение пороговой функции:

$$(6) \quad P(W_\Psi f_{a,b}) = \begin{cases} W_\Psi f_{a,b}, & \text{если } |W_\Psi f_{a,b} - W_\Psi f_{a,b}^{med,l}| \geq T_a^l, \\ 0, & \text{если } |W_\Psi f_{a,b} - W_\Psi f_{a,b}^{med,l}| < T_a^l, \end{cases}$$

где $W_\Psi \tilde{f}_{a,b} = |b|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} \tilde{f}(t) \Psi\left(\frac{t-a}{b}\right) dt$ — непрерывное вейвлет-преобразование, $W_\Psi f_{a,b}^{med,l}$ — медианное значение, $T_a^l = U \cdot \sigma_a^l$ — порог, $\sigma_a^l =$

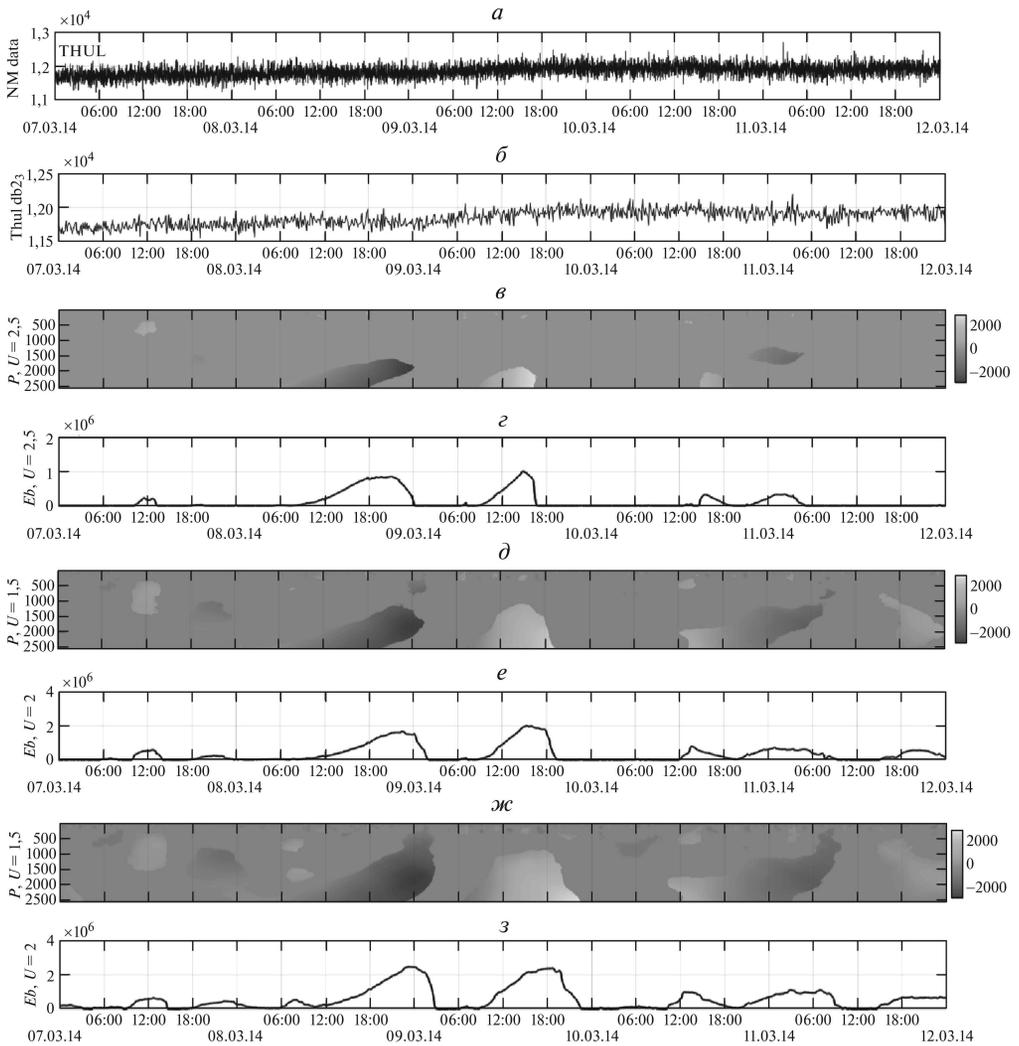


Рис. 5. *a* – данные нейтронного монитора, *б* – операция (5), *в* – операция (6), $U = 2,5$, положительные аномалии отмечены белым, отрицательные – черным, *г* – операция (7), $U = 2,5$, *д* – операция (6), $U = 2$, *е* – операция (7), $U = 2$, *ж* – операция (6), $U = 1,5$, *з* – операция (7), $U = 1,5$.

$= \sqrt{\frac{1}{l} \sum_{k=1}^l (W_{\Psi} f_{a,b} - \overline{W_{\Psi} f_{a,b}})^2}$ – стандартное отклонение, $\overline{W_{\Psi} f_{a,b}}$ – медиана, U – пороговый коэффициент.

3. Оценка интенсивности аномалий:

$$(7) \quad SUM = \sum_b P_a (W_{\Psi} \tilde{f}_{a,b}).$$

Результаты применения алгоритма по данным станции Туле представлены на рис. 5. На рис. 5, *ж*, 5, *з*, 5, *д* и 5, *е* показаны результаты алгоритма с

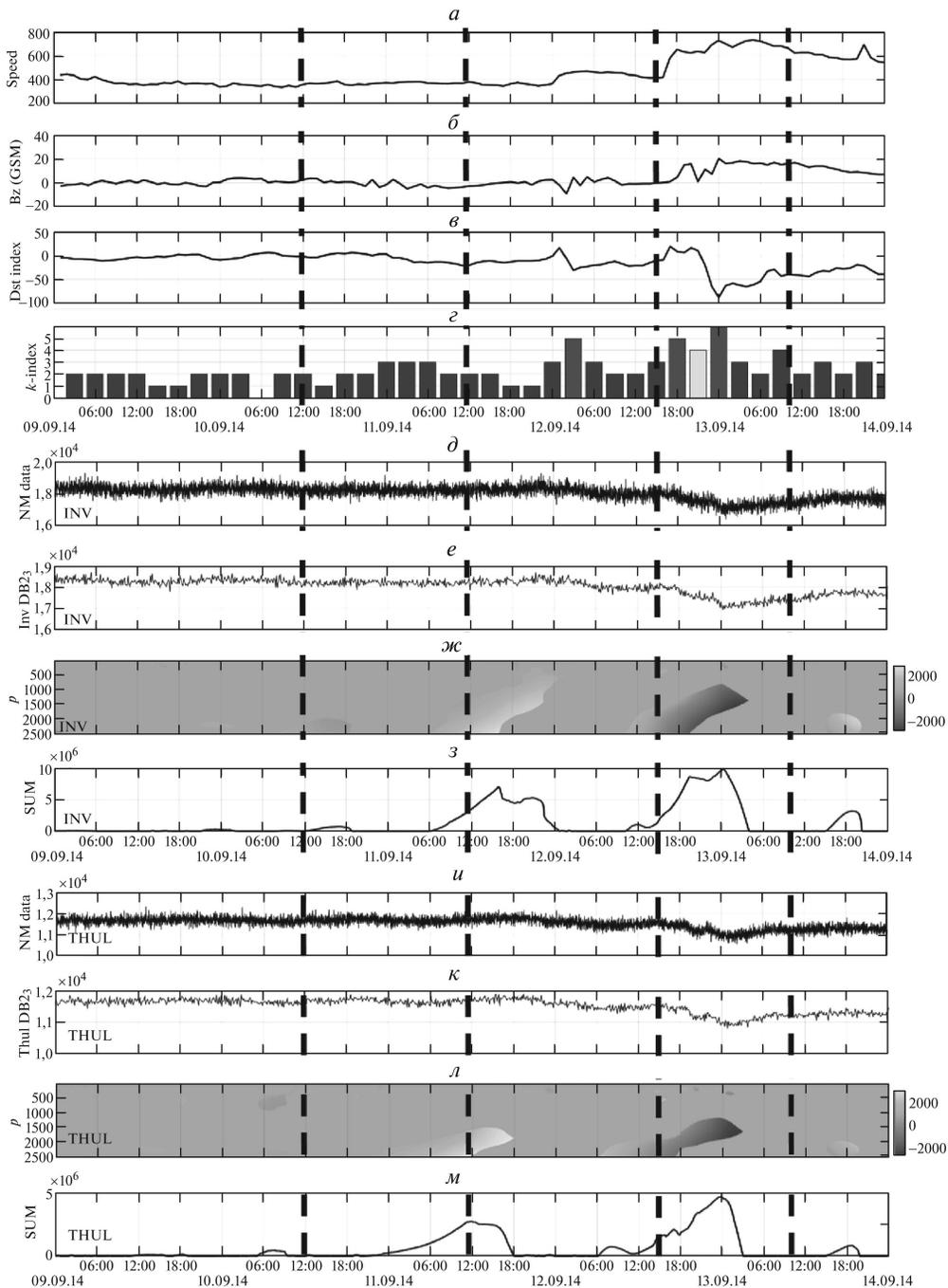


Рис. 6. *a* — скорость солнечного ветра км/с, *б* — B_z ММП, *в* — Dst-индекс, *з* — k -индекс геомагнитной активности, *д* — регистрируемые данные НМ ст. Инувик, *е* — операция (5), *ж* — операция (6), *з* — операция (7), *и* — регистрируемые данные НМ ст. Туле, *к* — операция (5), *л* — операция (6), *м* — операция (7).

коэффициентами $U = 1,5$ и $U = 2$ (см. операции (6) и (7)). В соответствии с алгоритмом коэффициент $U = 1,5$ позволяет детектировать вариации, превышающие по амплитуде от $1,5\sigma$ и более, а коэффициент $U = 2$ — от 2σ и более. Анализ результатов показывает, что при использовании коэффициента $U = 1,5$ (рис. 5,ж и 5,з) детектируются колебания интенсивности космических лучей, связанные с суточным ходом. Применение коэффициента $U = 2$ (рис. 5,д и 5,е) позволяет детектировать аномальные изменения в данных космических лучей, но суточные вариации также присутствуют. Наилучшие результаты показывает коэффициент $U = 2,5$ (рис. 5,в и 5,г) — отсутствие вариаций, связанных с суточным ходом и детектирование спорадических эффектов.

На рис. 6 представлено применение метода с использованием данных станции Инувик и Туле [6]. В начале анализируемого периода 11 сентября из-за ускоренного потока от коронального выброса скорость солнечного ветра возросла до 480 км/с (рис. 6,а) [22], южная V_z ММП опустилось до -14 нТл (рис. 6,б). Далее, 12 сентября скорость солнечного ветра резко увеличилась до 800 км/с, южная компонента ММП опустилась до значений $V_z = -17$ нТл. По данным космической погоды [22] в течение данного периода зарегистрировано 4 геомагнитных бури: 10, 11, 12 и 13 сентября. Моменты регистрации магнитных бурь отмечены на рис. 6 вертикальной пунктирной линией. При идентификации некоторых событий обнаружение аномалии предложенным методом отстает или опережает на несколько часов время, указанное в [22], что не является погрешностью метода, поскольку поток космических лучей приходит на Землю анизотропно и один и тот же Форбуш-эффект на разных станциях возникает с разной интенсивностью и в разное время. Результаты подтверждают эффективность метода.

3. Заключение

Эмпирически доказана эффективность предлагаемого метода для обнаружения спорадических эффектов в вариациях космических лучей. Рекомендовано использование большего порога, увеличивающего риск, но позволяющего получить более точные оценки. Показано, что применение суперпозиции вейвлет-пакетов и непрерывного вейвлет-преобразования позволяет повысить эффективность детектирования Форбуш-эффектов разной структуры. Полученные результаты представляют интерес в задаче прогноза космической погоды.

В дальнейшем планируется продолжить исследование в данном направлении с привлечением данных сети станций высоких и средних широт с целью оптимизации метода.

СПИСОК ЛИТЕРАТУРЫ

1. Цурко В.В., Михальский А.И. Моделирование данных при анализе рисков здоровью и продолжительности жизни человека // *АиТ*. 2018. № 10. С. 164–182.
Zurko V., Mikhailskii A. Data Modeling for the Analysis of Health Risks and Human Longevity // Autom. Remote Control. 2018. V. 79. No. 10. P. 1871–1885.

2. *Агеев И.А., Бурков В.Н., Зинченко В.И., Киселева Т.В.* Структурный анализ временных рядов данных // *АиТ.* 2005. № 6. С. 161–169.
Ageev I.A., Burkov V.N., Zinchenko V.I., Kiseleva T.V. Structural Analysis of the Time Data Series // *Autom. Remote Control.* 2005. V. 66. No. 6. P. 995–1002.
3. *Щербань И.В., Кириленко Н.Е., Красников С.О.* Метод поиска неизвестных высокочастотных осцилляторов в составе зашумленных сигналов на основе непрерывного вейвлет-преобразования // *АиТ.* 2019. № 7. С. 122–133.
Shcherban I.V., Kirilenko N.E., Krasnikov S.O. A Search Method for Unknown High-frequency Oscillators in Noisy Signals Based on the Continuous Wavelet Transform // *Autom. Remote Control.* 2019. V. 80. No. 7. P. 1279–1287.
4. *Mandrikova O., Stepanenko A.* Automated Method for Calculating the Dst-index Based on the Wavelet Model of Geomagnetic Field Variations // *Comp. Optics.* 2020. V. 44. Iss. 5. P. 797–808.
5. *Топтыгин И.Н.* Космические лучи в межпланетных магнитных полях. М.: Наука, 1983.
6. Real time data base for the measurements of high-resolution Neutron Monitor. [Электронный ресурс]. www.nmdb.eu (дата обращения 01.11.2020).
7. *Сокуров В.Ф.* Физика космических лучей: космическая радиация. Ростов н/Д.: Феникс, 2005.
8. *Дорман Л.И.* Экспериментальные и теоретические основы астрофизики космических лучей. М.: Наука, 1975.
9. *Belov A.V., et al.* Global Survey Method for the World Network of Neutron Monitors // *Geomagn. Aeron.* 2018. V. 58. P. 356–372.
10. *Abunina M.A., et al.* Ring of Stations Method in Cosmic Rays Variations Research // *Sol. Phys.* 2020. V. 69. No. 295.
11. *Mandrikova O.V., Solovev I.S., Zalyaev T.L.* Methods of Analysis of Geomagnetic Field Variations and Cosmic Ray Data // *Earth Planet Space.* 2014. V. 66. No. 148.
12. *Mandrikova O.V., et al.* Methods of Analysis of Geophysical Data During Increased Solar Activity // *Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications).* 2016. V. 26. No. 2. P. 406–418.
13. *Мандрикова О.В., Залаяев Т.Л.* Моделирование вариаций космических лучей на основе совмещения кратномасштабных вейвлет-разложений и нейронных сетей переменной структуры // *Цифровая обработка сигналов.* 2015. № 1. С. 11–16.
14. *Chui C.K.* An introduction in wavelets. N.Y.: Acad. Press, 1992.
15. *Mallat S.* A wavelet tour of signal processing. London: Acad. Press, 1999.
16. *Herley C., et al.* Tilings of the Time-Frequency Plane: Construction of Arbitrary Orthogonal Bases and Feist Tiling Algorithms // *IEEE Trans. Signal Process. Special Issue on Wavelets and Signal Processing.* 1993. P. 3341–3359.
17. *Chen S., Donoho D.* Atomic Decomposition by Basis Pursuit // *Technical Report.* Stanford University, 1995.
18. *Mallat S.G., Zhang Z.F.* Matching Pursuits with Time-frequency Dictionaries // *IEEE Trans. Signal Processing.* 1993. V. 41. No. 12. P. 3397–3415.
19. *Coifman R.R., Wickerhauser M.V.* Entropy-based Algorithms for Best Basis Selection // *IEEE Trans. Inform. Theory.* 1992. V. 38. No. 2. P. 713–718.
20. *Donoho D.L., Johnstone I.M.* Ideal Spatial Adaptation Via Wavelet Shrinkage // *Биометрика.* 1994. No. 81. P. 425–455.

21. Центр прогнозов космической погоды ИЗМИРАН. Каталог Форбуш-эффектов и межпланетных возмущений [Электронный ресурс].
<http://spaceweather.izmiran.ru/rus/fds2019.html> (дата обращения 11.11.2020).
22. Forecast of space weather according to the data of Federov Institute of Applied Geophysics [Электронный ресурс]. <http://ipg.geospace.ru> (дата обращения 01.12.2020).
23. NASA Interface to produce plots listings or output files from OMNI 2 [Электронный ресурс]. <https://omniweb.gsfc.nasa.gov/form/dx1.html> (дата обращения 11.11.2020).

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 26.04.2021

Принята к публикации 30.06.2021

© 2021 г. Е.Л. ГЛАДИН (gladin.el@phystech.edu),
М. АЛКУСА, канд. физ.-мат. наук (mohammad.alkousa@phystech.edu),
А.В. ГАСНИКОВ, д-р физ.-мат. наук (gasnikov.av@mipt.ru)
(Московский физико-технический институт, Долгопрудный;
Институт проблем передачи информации им. А. А. Харкевича РАН, Москва)

О РЕШЕНИИ ВЫПУКЛЫХ MIN-MIN ЗАДАЧ С ГЛАДКОСТЬЮ И СИЛЬНОЙ ВЫПУКЛОСТЬЮ ПО ОДНОЙ ИЗ ГРУПП ПЕРЕМЕННЫХ И МАЛОЙ РАЗМЕРНОСТЬЮ ДРУГОЙ¹

Статья посвящена некоторым подходам к решению выпуклых задач вида min-min с гладкостью и сильной выпуклостью только по одной из двух групп переменных. Показано, что предложенные подходы, основанные на методе Вайды, быстром градиентном методе и ускоренном градиентном методе с редукцией дисперсии, имеют линейную сходимость. Для решения внешней задачи предлагается использовать методы Вайды, для решения внутренней (гладкой и сильно выпуклой) — быстрый градиентный метод. Ввиду важности для приложений в машинном обучении отдельно рассмотрен случай, когда целевая функция является суммой большого числа функций. В этом случае вместо быстрого градиентного метода используется ускоренный градиентный метод с редукцией дисперсии. Приведены результаты численных экспериментов, иллюстрирующие преимущества предложенных процедур для задачи логистической регрессии, в которой есть априорное распределение на одну из двух групп переменных.

Ключевые слова: выпуклая оптимизация, метод секущей плоскости, метод Вайды, редукция дисперсии, быстрый градиентный метод, логистическая регрессия.

DOI: 10.31857/S0005231021100068

1. Введение

Одним из основных направлений исследований численных методов выпуклой оптимизации в последнее десятилетие стало повсеместное распространение конструкции ускорения обычного градиентного метода, предложенной в 1983 г. Ю.Е. Нестеровым [1], на различные другие численные методы оптимизации. За последние 15 лет ускоренный метод был успешно перенесен на гладкие задачи условной выпуклой оптимизации, на задачи со структурой

¹ Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госзадание) № 075-00337-20-03, номер проекта 0714-2020-0005. Работа А.В. Гасникова была также частично поддержана Российским фондом фундаментальных исследований (проект № 18-29-03071 мк).

(в частности, так называемые композитные задачи), безградиентные и рандомизированные методы (например, ускоренный градиентный метод с редуцией дисперсии для задач минимизации суммы функций [2]). Также ускорение было успешно перенесено на методы, использующие старшие производные. Детали и более подробный обзор публикаций можно найти в [3].

Задачи оптимизации вида min-max и седловые задачи широко изучались в литературе из-за их широкого спектра приложений в статистике, машинном обучении, компьютерной графике, теории игр и других областях. В последнее время многие исследователи активно работают над темой ускоренных методов решения этих задач, учитывающих их структуру: [4–8] — и это лишь некоторые из последних публикаций. В некоторых приложениях существует задача, аналогичная задаче min-max, которая остается в значительной степени неизученной — это задача вида min-min:

$$(1) \quad \min_{x \in Q_x} \min_{y \in Q_y} F(x, y),$$

где $Q_x \subset \mathbb{R}^d$, $Q_y \subset \mathbb{R}^n$ — непустые компактные выпуклые множества, размерность d относительно небольшая ($d \ll n$), функция $F(x, y)$ — выпуклая по совокупности переменных, а также L -гладкая и μ -сильно выпуклая по y . Под L -гладкостью по y понимается свойство

$$\|\nabla_y F(x, y) - \nabla_y F(x, y')\|_2 \leq L \|y - y'\|_2 \quad \forall x \in Q_x, y, y' \in Q_y.$$

Такая постановка возникает, например, при поиске равновесий в транспортных сетях [9]. В машинном обучении задачи такого типа соответствуют случаю, когда регуляризация применяется к одной из двух групп параметров модели (отсюда сильная выпуклость только по одной группе переменных из двух). Например, когда в датасете большая группа признаков являются разреженными, то регуляризация может использоваться только для весов модели, соответствующих этим признакам. В качестве еще одного примера можно привести логистическую регрессию, в которой есть априорное распределение на часть параметров. Задаче min-min посвящено несколько публикаций, среди которых [10–12]. Например, в [10] авторы предложили новые алгоритмы для задач min-max, шаги которых настраиваются автоматически, но предложенные методы также применяются и к задачам min-min.

В данной статье рассматриваются два подхода к решению задачи (1), имеющие линейную скорость сходимости. Предлагается свести рассматриваемую задачу к совокупности вспомогательных задач (внутренней и внешней). Внешняя задача (минимизация по x) решается методом Вайды (метод секущей плоскости) [13, 14].

В случае когда целевая функция F простая, т.е. не является суммой большого количества функций, внутренняя задача (минимизация по y) решается быстрым градиентным методом для задач сильно выпуклой оптимизации. В результате такого подхода приближенное решение задачи (1) может быть достигнуто за $\tilde{O}(d)$ вычислений $\partial_x F$ и $\tilde{O}\left(d\sqrt{\frac{L}{\mu}}\right)$ вычислений $\nabla_y F$, см. тео-

рему 5. Здесь и далее $\tilde{O}(\cdot) = \mathcal{O}(\cdot)$ с точностью до небольшой степени логарифмического множителя, обычно эта степень равна единице или двум.

Оптимизация суммы большого количества функций в течение последних нескольких лет является предметом интенсивных исследований из-за широкого спектра приложений в машинном обучении, статистике, обработке изображений и других математических и инженерных приложениях. Поэтому отдельно рассматривается случай, когда целевая функция F представляет собой сумму (или среднее арифметическое) большого числа m функций, в котором использование быстрого градиентного метода для задач сильно выпуклой оптимизации потребовало бы вычисления градиентов m слагаемых на каждом шаге, что может занимать много времени. Вместо этого предлагаем использовать ускоренный градиентный метод с редукцией дисперсии [2, 15], который также имеет линейную сходимость. В результате такого подхода решение задачи может быть достигнуто за $\tilde{O}(md)$ вычислений $\partial_x F$ и за $\tilde{O}\left(md + d\sqrt{\frac{mL}{\mu}}\right)$ вычислений $\nabla_y F$, см. теорему 6.

Используя два предложенных подхода, получаем линейную скорость сходимости для задачи min-min (1). Отметим, что гладкость и сильная выпуклость требуются только по одной из двух групп переменных.

Статья состоит из 5 разделов и Приложения. В разделе 2 приводятся используемые алгоритмы и их сложность, а именно: быстрый градиентный метод, метод Вайды (метод секущей плоскости) и метод ускоренного градиентного спуска с редукцией дисперсии. В разделе 3 формулируется постановка задачи и приводятся подходы к рассматриваемой задаче для различных случаев целевой функции, в одном из которых целевая функция является суммой или средним арифметическим большого числа функций. В разделе 4 приводятся результаты вычислительных экспериментов и сравнение скорости работы предложенных подходов. Отметим, что полные доказательства теорем 4, 5, 6 и вспомогательного утверждения 1 приводятся в Приложении.

2. Используемые алгоритмы

Приведем алгоритмы, используемые в предлагаемых в статье подходах к решению задачи (1). Сначала приводится быстрый градиентный метод, затем метод Вайды (метод секущей плоскости) и, наконец, ускоренный градиентный метод с редукцией дисперсии.

2.1. Быстрый градиентный метод

В [16] предложен адаптивный алгоритм для решения задачи оптимизации

$$(2) \quad f(y) \rightarrow \min_{y \in Q_y},$$

где $Q_y \subset \mathbb{R}^n$ — непустое компактное выпуклое множество, f — L -гладкая выпуклая функция. Этот алгоритм, получивший название быстрого градиентного метода, позволяет ускорить сходимость обычного градиентного спуска

с $\mathcal{O}\left(\frac{1}{N}\right)$ до $\mathcal{O}\left(\frac{1}{N^2}\right)$, где N — количество итерации алгоритма. Быстрый градиентный метод (не адаптивный вариант) приведен далее как алгоритм 1.

Алгоритм 1. Быстрый градиентный метод [16].

Вход: Количество шагов N , начальная точка $y^0 \in Q_y$, параметр $L > 0$.

1: 0-шаг: $z^0 := y^0$, $u^0 := y^0$, $\alpha_0 := 0$, $A_0 := 0$.

2: for $k = 0, 1, \dots, N - 1$ do

3: Находим наибольший корень α_{k+1} такой, что $A_k + \alpha_{k+1} = L\alpha_{k+1}^2$,

4: $A_{k+1} := A_k + \alpha_{k+1}$,

5: $z^{k+1} := \frac{\alpha_{k+1}u^k + A_k y^k}{A_{k+1}}$,

6: $u^{k+1} := \arg \min_{y \in Q_y} \left\{ \alpha_{k+1} \left\langle \nabla f(z^{k+1}), y - z^{k+1} \right\rangle + \frac{1}{2} \|y - u^k\|_2^2 \right\}$,

7: $y^{k+1} := \frac{\alpha_{k+1}u^{k+1} + A_k y^k}{A_{k+1}}$,

8: end for

Выход: y^N .

Следующая теорема дает оценку сложности (скорости сходимости) алгоритма 1.

Теорема 1 [16]. Пусть функция $f : Q_y \rightarrow \mathbb{R}$ является L -гладкой и выпуклой, тогда алгоритм 1 возвращает такую точку y^N , что

$$f(y^N) - f(y_*) \leq \frac{8LR^2}{(N+1)^2},$$

где y_* — решение задачи (2), $R^2 = \frac{1}{2} \|y^0 - y_*\|_2^2$.

Опишем далее технику рестартов (перезапусков) быстрого градиентного метода (алгоритм 1) для случая μ -сильно выпуклой функции.

Ввиду μ -сильной выпуклости f имеем

$$\frac{\mu}{2} \|z - y\|_2^2 \leq f(z) - (f(y) + \langle \nabla f(y), z - y \rangle) \leq \frac{L}{2} \|z - y\|_2^2 \quad \forall y, z \in Q_y.$$

Тогда после N_1 итераций алгоритма 1 с учетом теоремы 1 получаем

$$(3) \quad \frac{\mu}{2} \|y^{N_1} - y_*\|_2^2 \leq f(y^{N_1}) - f(y_*) \leq \frac{4L \|y^0 - y_*\|_2^2}{N_1^2},$$

отсюда

$$\|y^{N_1} - y_*\|_2^2 \leq \frac{8L}{\mu N_1^2} \|y^0 - y_*\|_2^2.$$

Поэтому, выбирая $N_1 = \left\lceil 4\sqrt{\frac{L}{\mu}} \right\rceil$, где $\lceil \cdot \rceil$ — округление вверх, получим

$$\|y^{N_1} - y_*\|_2^2 \leq \frac{1}{2} \|y^0 - y_*\|_2^2.$$

После этого выберем для алгоритма 1 в качестве точки старта y^{N_1} , снова сделаем N_1 итераций и т.д. Для достижения приемлемого качества решения можно выбрать количество рестартов алгоритма 1 (параметр p алгоритма 2) следующим образом:

$$p = \left\lceil \frac{1}{2} \ln \left(\frac{\mu R^2}{\varepsilon} \right) \right\rceil.$$

В таком случае общее число итераций алгоритма 2 будет

$$N = \left\lceil \frac{1}{2} \ln \left(\frac{\mu R^2}{\varepsilon} \right) \right\rceil \cdot \left\lceil 4\sqrt{\frac{L}{\mu}} \right\rceil,$$

т.е.

$$(4) \quad N = \mathcal{O} \left(\sqrt{\frac{L}{\mu}} \ln \left(\frac{\mu R^2}{\varepsilon} \right) \right) = \tilde{\mathcal{O}} \left(\sqrt{\frac{L}{\mu}} \right).$$

Алгоритм 2. Быстрый градиентный метод для задач сильно выпуклой оптимизации, рестарты алгоритма 1.

Вход: начальная точка $y^0 \in Q_y$, $L > 0$, число рестартов $p = \left\lceil \frac{1}{2} \ln \left(\frac{\mu R^2}{\varepsilon} \right) \right\rceil$.

1: for $j = 1, \dots, p$ do

2: Выполнить $N_j = \left\lceil 4\sqrt{\frac{L}{\mu}} \right\rceil$ итераций алгоритма 1,

3: $y^0 := y^{N_j}$.

4: end for

Выход: $\hat{y} := y^{N_p}$.

2.2. Метод Вайды

Метод Вайды (метод секущей плоскости) был предложен Вайдой в [13, 14] для решения условной задачи оптимизации

$$(5) \quad f(x) \rightarrow \min_{x \in Q_x},$$

где $Q_x \subset \mathbb{R}^d$ — выпуклое компактное множество с непустой внутренностью, а целевая функция f , определенная на Q_x , непрерывна и выпукла.

Пусть $P = \{x \in \mathbb{R}^d : Ax \geq b\}$ — ограниченный d -мерный многогранник, где $A \in \mathbb{R}^{m \times d}$ и $b \in \mathbb{R}^m$. Логарифмический барьер множества P определяется как

$$Barr(x) = - \sum_{i=1}^m \log(a_i^\top x - b_i),$$

где a_i^\top — i -я строка матрицы A . Гессиан $H(x)$ функции $Barr(x)$ равен

$$H(x) = \sum_{i=1}^m \frac{a_i a_i^\top}{(a_i^\top x - b_i)^2}.$$

Матрица $H(x)$ положительно определена для всех x из внутренности P . Волюметрический барьер (volumetric barrier) \mathcal{V} определяется как

$$\mathcal{V}(x) = \frac{1}{2} \log(\det(H(x))),$$

где $\det(H(x))$ обозначает детерминант $H(x)$. Будем называть точку минимума функции \mathcal{V} на P волюметрическим центром множества P .

Обозначим

$$(6) \quad \sigma_i(x) = \frac{a_i^\top (H(x))^{-1} a_i}{(a_i^\top x - b_i)^2}, \quad 1 \leq i \leq m,$$

тогда градиент волюметрического барьера \mathcal{V} может быть записан как

$$\nabla \mathcal{V}(x) = - \sum_{i=1}^m \sigma_i(x) \frac{a_i}{a_i^\top x - b_i}.$$

Пусть $\mathcal{Q}(x)$ определяется как

$$\mathcal{Q}(x) = \sum_{i=1}^m \sigma_i(x) \frac{a_i a_i^\top}{(a_i^\top x - b_i)^2}.$$

Заметим, что $\mathcal{Q}(x)$ положительно определена на внутренности P , а также $\mathcal{Q}(x)$ является хорошим приближением гессиана функции $\mathcal{V}(x)$, т.е. $\nabla^2 \mathcal{V}(x)$.

Метод Вайды производит последовательность пар $(A_k, b_k) \in \mathbb{R}^{m \times d} \times \mathbb{R}^m$ таких, что соответствующие многогранники содержат решение. В качестве начального многогранника, задаваемого парой (A_0, b_0) , обычно берется симплекс (алгоритм может начинать с любого выпуклого ограниченного n -мерного многогранника, для которого легко вычислить волюметрический центр — например, с n -прямоугольника).

Параметром алгоритма является небольшое число $\gamma \leq 0,006$, смысл которого более подробно раскрывается в книге [17]. Пусть x_k ($k \geq 0$) обозначает волюметрический центр многогранника, заданного парой (A_k, b_k) , и пусть для него вычислены величины $\{\sigma_i(x_k)\}_{1 \leq i \leq m}$ (см. (6)). Следующий многогранник (A_{k+1}, b_{k+1}) получается из текущего в результате либо присоединения, либо удаления ограничения:

- 1) Если для некоторого $i \in \{1, \dots, m\}$ выполняется $\sigma_i(x_k) = \min_{1 \leq j \leq m} \sigma_j(x_k) < \gamma$, тогда (A_{k+1}, b_{k+1}) получается исключением i -й строки из (A_k, b_k) ;
- 2) иначе (если $\min_{1 \leq j \leq m} \sigma_j(x_k) \geq \gamma$) оракул, вызванный в текущей точке x_k , возвращает вектор c_k такой, что $f(x) \leq f(x_k) \forall x \in \left\{ z \in Q_x : c_k^\top z \geq c_k^\top x_k \right\}$, т.е. $c_k \in -\partial f(x_k)$. Выберем $\beta_k \in \mathbb{R}$ таким, что

$$\frac{c_k^\top (H(x_k))^{-1} c_k}{(x_k^\top c_k - \beta_k)^2} = \frac{1}{5} \sqrt{\gamma}.$$

Определим (A_{k+1}, b_{k+1}) добавлением строки (c_k, β_k) к (A_k, b_k) .

Волюметрический барьер \mathcal{V}_k является самосогласованной функцией, поэтому может быть эффективно минимизирован методом Ньютона. Достаточно одного шага метода Ньютона для \mathcal{V}_k , сделанного из x_{k-1} . Подробности и анализ метода Вайды можно найти в [13, 14, 17].

Следующая теорема дает оценку сложности алгоритма Вайды.

Теорема 2. Пусть \mathcal{B}_ρ и \mathcal{B}_R — некоторые евклидовы шары радиусов ρ и R соответственно такие, что $\mathcal{B}_\rho \subseteq Q_x \subseteq \mathcal{B}_R$, и пусть число $B > 0$ таково, что $|f(x) - f(x')| \leq B \forall x, x' \in Q_x$. Тогда метод Вайды находит ε -решение задачи (5) за $\mathcal{O}\left(d \log \frac{dBR}{\rho\varepsilon}\right)$ шагов.

Замечание 1. Как показано в [18], метод Вайды можно использовать с неточным субградиентом без накопления ошибки.

Замечание 2. Помимо вычисления субградиента, в стоимость итерации метода Вайды входит стоимость обращения матрицы размера $d \times d$ и решения системы линейных уравнений.

2.3. Ускоренный градиентный метод с редукцией дисперсии

Рассмотрим задачу

$$(7) \quad f(y) \rightarrow \min_{y \in Q_y},$$

где $Q_y \subseteq \mathbb{R}^n$ — замкнутое выпуклое множество, а целевая функция f представляет собой сумму (или среднее арифметическое) большого числа m гладких выпуклых функций f_i , т.е. $f(y) = \frac{1}{m} \sum_{i=1}^m f_i(y)$. При решении (7) с помощью быстрого градиентного метода для задач сильно выпуклой оптимизации (алгоритм 2) потребуется вычислять градиент m функций на каждой итерации, что очень дорого. Поэтому предпочтительнее вместо алгоритма 2 использовать рандомизированный градиентный метод, а именно ускоренный градиентный метод с редукцией дисперсии, также называемый Varag [2, 15]. Приведенный далее алгоритм 3 представляет собой ускоренный градиентный метод с редукцией дисперсии (Varag) для гладкой сильно выпуклой задачи оптимизации конечной суммы (7). Этот алгоритм был предложен Г. Ланом и др. в [15].

Предположим, что для каждого $i \in \{1, \dots, m\}$, существует $L_i > 0$ такое, что

$$\|\nabla f_i(y) - \nabla f_i(z)\|_2 \leq L_i \|y - z\|_2 \quad \forall y, z \in Q_y.$$

Ясно, что f имеет липшицев градиент с константой не более $L := \frac{1}{m} \sum_{i=1}^m L_i$. Предположим также, что целевая функция f сильно выпуклая с константой $\mu > 0$, т.е.

$$f(z) \geq f(y) + \langle \nabla f(y), z - y \rangle + \frac{\mu}{2} \|y - z\|_2^2 \quad \forall y, z \in Q_y.$$

Определение 1. Случайный вектор \bar{y} , принимающий значения из Q_y , называется стохастическим ε -решением задачи (7), если $\mathbb{E}[f(\bar{y}) - f(y_*)] \leq \varepsilon$, где y_* — точное решение задачи (7).

Алгоритм Vagag содержит вложенные циклы — внешний и внутренний (индексируемые переменными s и t соответственно). На каждой итерации внешнего цикла вычисляется полный градиент $\nabla f(\tilde{y})$ в точке \tilde{y} , который затем используется во внутреннем цикле для определения оценок градиента G_t . Каждая итерация внутреннего цикла требует информацию о градиенте только одного случайно выбранного слагаемого f_{i_t} и содержит три основные последовательности: $\{y_t\}$, $\{y_t\}$ и $\{\bar{y}_t\}$.

Обозначим $s_0 := \lfloor \log_2 m \rfloor + 1$, где $\lfloor \cdot \rfloor$ — округление вниз. Параметры алгоритма $\{q_1, \dots, q_m\}$, $\{\theta_t\}$, $\{\alpha_s\}$, $\{\gamma_s\}$, $\{p_s\}$ и $\{T_s\}$ описываются следующим образом:

- Вероятности $q_i = \frac{1}{\sum_{i=1}^m L_i} L_i \quad \forall i \in \{1, \dots, m\}$;
- Веса $\{\theta_t\}$ при $1 \leq s \leq s_0$ или $s_0 < s \leq s_0 + \sqrt{\frac{12L}{m\mu}} - 4$, $m < \frac{3L}{4\mu}$ равны

$$(8) \quad \theta_t = \begin{cases} \frac{\gamma_s}{\alpha_s} (\alpha_s + p_s), & 1 \leq t \leq T_s - 1, \\ \frac{\gamma_s}{\alpha_s}, & t = T_s. \end{cases}$$

В остальных случаях они равны

$$(9) \quad \theta_t = \begin{cases} \Gamma_{t-1} - (1 - \alpha_s - p_s) \Gamma_t, & 1 \leq t \leq T_s - 1, \\ \Gamma_{t-1}, & t = T_s, \end{cases}$$

где $\Gamma_t = (1 + \mu\gamma_s)^t$;

- Параметры $\{T_s\}$, $\{\gamma_s\}$ и $\{p_s\}$ определяются как

$$(10) \quad T_s = \begin{cases} 2^{s-1}, & s \leq s_0, \\ T_{s_0}, & s > s_0, \end{cases} \quad \gamma_s = \frac{1}{3L\alpha_s}, \quad p_s = \frac{1}{2};$$

- Наконец,

$$(11) \quad \alpha_s = \begin{cases} \frac{1}{2}, & s \leq s_0, \\ \max \left\{ \frac{2}{s - s_0 + 4}, \min \left\{ \sqrt{\frac{m\mu}{3L}}, \frac{1}{2} \right\} \right\}, & s > s_0. \end{cases}$$

Алгоритм 3. Ускоренный градиентный метод с редукцией дисперсии (Varag) [15].

Вход: $y^0 \in Q_y, \{T_s\}, \{\gamma_s\}, \{\alpha_s\}, \{p_s\}, \{\theta_t\}$ и распределение вероятностей $\{q_1, \dots, q_m\}$ на $\{1, \dots, m\}$.

- 1: $\tilde{y}^0 := y^0$.
 - 2: for $s = 1, 2, \dots$, do
 - 3: $\tilde{y} := \tilde{y}^{s-1}, \tilde{g} := \nabla f(\tilde{y})$.
 - 4: $y_0 := y^{s-1}, \bar{y}_0 = \tilde{y}, T := T_s$.
 - 5: for $t = 1, 2, \dots, T$ do
 - 6: Выбрать $i_t \in \{1, \dots, m\}$ случайным образом согласно $\{q_1, \dots, q_m\}$.
 - 7: $\underline{y}_t := \frac{1}{(1 + \mu\gamma_s(1 - \alpha_s))} [(1 + \mu\gamma_s)(1 - \alpha_s - p_s)\bar{y}_{t-1} + \alpha_s y_{t-1} + (1 + \mu\gamma_s)p_s \tilde{y}]$.
 - 8: $G_t := \frac{1}{(q_{i_t} m)} \left(\nabla f_{i_t}(\underline{y}_t) - \nabla f_{i_t}(\tilde{y}) \right) + \tilde{g}$.
 - 9: $y_t := \arg \min_{y \in Q_y} \left\{ \gamma_s \left(\langle G_t, y \rangle + \frac{\mu}{2} \|\underline{y}_t - y\|_2^2 \right) + \frac{1}{2} \|y_{t-1} - y\|_2^2 \right\}$.
 - 10: $\bar{y}_t := (1 - \alpha_s - p_s)\bar{y}_{t-1} + \alpha_s y_t + p_s \tilde{y}$.
 - 11: end for
 - 12: $y^s := y_T, \tilde{y}^s := \frac{1}{\sum_{t=1}^T \theta_t} \sum_{t=1}^T (\theta_t \bar{y}_t)$.
 - 13: end for
-

Следующий результат дает оценку сложности алгоритма 3.

Теорема 3 [15]. Если параметры алгоритма 3 $\{\theta_t\}, \{\alpha_s\}, \{\gamma_s\}, \{p_s\}$ и $\{T_s\}$ заданы согласно формулам (8), (9), (10) и (11), то общее количество вычислений градиентов функций f_i , выполняемых алгоритмом 3 для нахождения стохастического ε -решения задачи (7), ограничено

$$(12) \quad N := \begin{cases} \mathcal{O} \left\{ m \log \frac{D_0}{\varepsilon} \right\}, & m \geq \frac{D_0}{\varepsilon} \text{ или } m \geq \frac{3L}{4\mu}, \\ \mathcal{O} \left\{ m \log m + \sqrt{\frac{mD_0}{\varepsilon}} \right\}, & m < \frac{D_0}{\varepsilon} \leq \frac{3L}{4\mu}, \\ \mathcal{O} \left\{ m \log m + \sqrt{\frac{mL}{\mu}} \log \frac{D_0/\varepsilon}{3L/4\mu} \right\}, & m < \frac{3L}{4\mu} \leq \frac{D_0}{\varepsilon}, \end{cases}$$

где $D_0 = 2(f(y^0) - f(y_*)) + \frac{3L}{2} \|y^0 - y_*\|_2^2$, где y_* — решение задачи (7).

Заметим, что оценку (12) можно записать как $N = \tilde{\mathcal{O}} \left(m + \sqrt{\frac{mL}{\mu}} \right)$, где $\tilde{\mathcal{O}}(\cdot) = \mathcal{O}(\cdot)$ с точностью до логарифмического множителя по m, L, μ, ε и D_0 .

3. Постановка задачи и полученные результаты

Рассмотрим задачу

$$(13) \quad \min_{x \in Q_x} \min_{y \in Q_y} F(x, y),$$

где $Q_x \subset \mathbb{R}^d$, $Q_y \subset \mathbb{R}^n$ — непустые компактные выпуклые множества, размерность d относительно небольшая ($d \ll n$), функция $F(x, y)$ — выпуклая по совокупности переменных, а также L -гладкая и μ -сильно выпуклая по y . Под L -гладкостью по y понимается свойство

$$\|\nabla_y F(x, y) - \nabla_y F(x, y')\|_2 \leq L \|y - y'\|_2 \quad \forall x \in Q_x, y, y' \in Q_y.$$

Введем функцию

$$(14) \quad f(x) = \min_{y \in Q_y} F(x, y).$$

Задачу (13) можно переписать в виде

$$(15) \quad f(x) \rightarrow \min_{x \in Q_x}.$$

При решении (15) некоторым итерационным методом необходимо на каждом его шаге решать вспомогательную задачу (14), чтобы приближенно находить субградиент $\partial f(x)$. Обратимся к следующему определению.

Определение 2 ([19], с. 123). Пусть $\delta \geq 0$, $Q_x \subseteq \mathbb{R}^d$ — выпуклое множество, $f : Q_x \rightarrow \mathbb{R}$ — выпуклая функция. Вектор $g \in \mathbb{R}^d$ называется δ -субградиентом f в точке $x' \in Q_x$, если

$$f(x) \geq f(x') + \langle g, x - x' \rangle - \delta \quad \forall x \in Q_x.$$

Множество δ -субградиентов f в точке x' обозначается $\partial_\delta f(x')$.

Обозначим $D := \max_{y, z \in Q_y} \|y - z\|_2$, $y(x) := \arg \min_{y \in Q_y} F(x, y)$. Следующая теорема говорит о том, как вычислить δ -субградиент функции $f(x)$, приближенно решая вспомогательную задачу (15).

Теорема 4. Пусть найден такой $\tilde{y} \in Q_y$, что $F(x, \tilde{y}) - f(x) \leq \varepsilon$, тогда

$$\partial_x F(x, \tilde{y}) \in \partial_\delta f(x), \quad \delta = (LD + \|\nabla_y F(x, y(x))\|_2) \sqrt{\frac{2\varepsilon}{\mu}}.$$

Эта теорема непосредственно следует из двух утверждений.

Утверждение 1. Пусть $g : Q_y \rightarrow \mathbb{R}$ — L -гладкая μ -сильно выпуклая функция, точка $\tilde{y} \in Q_y$ такова, что $g(\tilde{y}) - g(y_*) \leq \varepsilon$, тогда

$$\max_{y \in Q_y} \langle \nabla g(\tilde{y}), \tilde{y} - y \rangle \leq \delta, \quad \delta = (LD + \|\nabla g(y_*)\|_2) \sqrt{\frac{2\varepsilon}{\mu}},$$

где $y_* = \arg \min_{y \in Q_y} g(y)$.

Утверждение 2 ([20], с. 12). Пусть найден такой $\tilde{y} \in Q_y$, что

$$\max_{y \in Q_y} \langle \nabla_y F(x, \tilde{y}), \tilde{y} - y \rangle \leq \delta,$$

тогда $\partial_x F(x, \tilde{y}) \in \partial_\delta f(x)$.

Интуитивно теорема 4 говорит о том, что, решив вспомогательную задачу (14) достаточно точно, получим хорошее приближение субградиента $\partial f(x)$, которое может быть использовано для решения внешней задачи (15). На этой идее основан предлагаемый подход к решению (13).

Подход 1 (основной случай). Внешняя задача (15) решается методом Вайды. Вспомогательная задача (14) решается быстрым градиентным методом для задач сильно выпуклой оптимизации (алгоритм 2).

Теорема 5. Подход 1 позволяет получить ε -решение задачи (13) после $\tilde{O}(d)$ вычислений $\partial_x F$ и обращений матриц размера $d \times d$, а также $\tilde{O}\left(d\sqrt{\frac{L}{\mu}}\right)$ вычислений $\nabla_y F$.

Замечание 3. Обращение матриц появляется в сложности предлагаемого подхода из-за того, что оно производится на каждом шаге метода Вайды.

3.1. Минимизация суммы большого числа функций

Пусть в задаче (13)

$$(16) \quad F(x, y) = \frac{1}{m} \sum_{i=1}^m F_i(x, y),$$

где функции F_i являются выпуклыми по совокупности переменных и L_i -гладкими по y , а F является μ -сильно выпуклой по y . Из этого следует, что F является выпуклой по совокупности переменных и гладкой по y с константой гладкости не более $L := \frac{1}{m} \sum_{i=1}^m L_i$.

Подход 2 (сумма функций). Внешняя задача (15) решается методом Вайды. Вспомогательная задача (14) решается ускоренным градиентным методом с редукцией дисперсии (алгоритм 3).

Теорема 6. Подход 2 позволяет получить ε -решение задачи (13) за $\tilde{O}(md)$ вычислений $\partial_x F_i$, $\tilde{O}(d)$ обращений матриц размера $d \times d$ и $\tilde{O}\left(dm + d\sqrt{\frac{mL}{\mu}}\right)$ вычислений $\nabla_y F_i$.

4. Эксперименты

Рассмотрим модель логистической регрессии для задачи бинарной классификации. Ошибка модели с параметрами w на обучающем объекте с вектором признаков z , принадлежащем классу $t \in \{-1, 1\}$, записывается как

$$\ell_z(w) = \log\left(1 + e^{-t\langle w, z \rangle}\right).$$

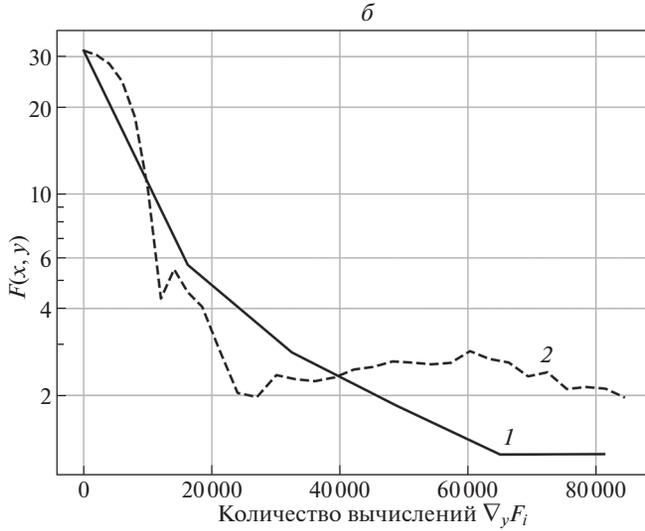
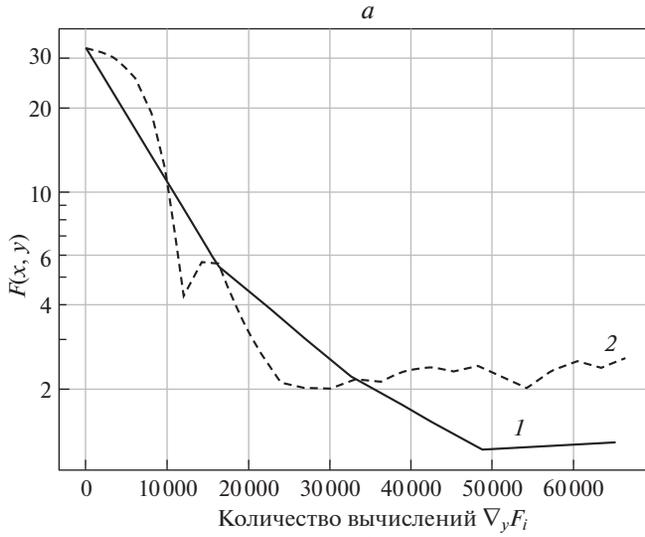


Рис. 1. *a* и *б* соответствуют размерностям $d = 20$ и $d = 30$ соответственно. Графики 1 и 2 показывают сходимость предлагаемого подхода и метода Varag соответственно.

Пусть параметры модели состоят из двух групп: $w = (x, y)$, $x \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, причем для группы y задано гауссовское априорное распределение:

$$y \sim \mathcal{N}(0, \sigma^2 I_n),$$

где I_n — единичная матрица размера n . Максимизация апостериорной вероятности приведет (см. [21], § 4.5.1) к задаче

$$(17) \quad \min_{x \in Q_x} \min_{y \in Q_y} \left\{ F(x, y) := \frac{1}{m} \sum_{i=1}^m \ell_{z_i}(x, y) + \frac{1}{\sigma^2} \|y\|_2^2 \right\},$$

где в качестве Q_x и Q_y можно взять евклидовы шары достаточно большого радиуса.

Будем решать задачу (17) при помощи подхода 2 и сравним его работу с работой метода Varag (алгоритм 3). Заметим, что эта задача не является сильно выпуклой по совокупности переменных. Для такой постановки можно использовать Varag, задавая параметры θ_t по формуле (8), а все остальные параметры по формулам для сильно выпуклого случая, положив $\mu = 0$, см. [15].

При этом стохастическое ε -решение будет найдено за $\mathcal{O}\left(\sqrt{\frac{mD_0}{\varepsilon}} + m \log m\right)$ вычислений градиентов функций F_i , где $D_0 = 2(F(x^0, y^0) - F(x_*, y_*)) + \frac{3L}{2}\|(x^0, y^0) - (x_*, y_*)\|_2^2$, (x_*, y_*) — решение задачи (17). Эта сублинейная оценка уступает предлагаемому в статье подходу, см. теорему 2.

Для экспериментов использовался датасет madelon, представленный 2000 объектов, имеющих 500 признаков. Был выбран небольшой коэффициент регуляризации $\frac{1}{\sigma^2} = 0,005$ и проведены эксперименты для двух размерностей d , равных 20 и 30.

На рис. 1 отражены результаты эксперимента. По оси x откладывается количество вычислений градиентов $\nabla_y F_i$, которое для Varag совпадает с количеством вычислений $\nabla_x F_i$. Отметим, что предложенный подход требует меньше вычислений $\nabla_x F_i$, поскольку они выполняются только во внешнем цикле. Так, график 1 на рис. 1,а соответствует четырем итерациям внешнего цикла (т.е. 8000 вычислений $\nabla_x F_i$), а график 1 на рис. 1,б — пяти итерациям (т.е. 10 000 вычислений $\nabla_x F_i$). В данном эксперименте подход 2 позволил достичь меньших значений целевой функции.

Исходный код и результаты экспериментов могут быть найдены в репозитории https://github.com/egorgladin/min_min.

5. Заключение

В статье рассмотрена задача вида min-min:

$$(18) \quad \min_{x \in Q_x} \min_{y \in Q_y} F(x, y),$$

где $Q_x \subset \mathbb{R}^d$, $Q_y \subset \mathbb{R}^n$ — непустые компактные выпуклые множества, размерность d относительно небольшая ($d \ll n$), функция $F(x, y)$ — выпуклая по совокупности переменных, а также L -гладкая и μ -сильно выпуклая по y .

Предложено два подхода к решению задачи (18), в которых она сводится к совокупности вспомогательных задач (внутренней и внешней). Внешняя задача (минимизация по x) решается методом Вайды, а внутренняя (минимизация по y) — быстрым градиентным методом для задач сильно выпуклой оптимизации или, если минимизируется сумма большого количества функций, ускоренным градиентным методом с редукцией дисперсии. Это позволяет достигать приближенного решения задачи (18) за $\tilde{\mathcal{O}}(d)$ вычислений $\partial_x F$ и $\tilde{\mathcal{O}}\left(d\sqrt{\frac{L}{\mu}}\right)$ вычислений $\nabla_y F$, см. теорему 5. Для сравнения, если бы задача (18) была гладкой по совокупности переменных, то ее решение при

использовании только быстрого градиентного метода имело бы сложность $\mathcal{O}\left(\sqrt{\frac{LR^2}{\varepsilon}}\right)$, где R — расстояние от начального приближения до решения. В случае суммы с m слагаемыми решение задачи может быть достигнуто за $\tilde{\mathcal{O}}(md)$ вычислений $\partial_x F$ и за $\tilde{\mathcal{O}}\left(md + d\sqrt{\frac{mL}{\mu}}\right)$ вычислений $\nabla_y F$, см. теорему 6.

Проведен численный эксперимент, в котором один из предлагаемых подходов применен к задаче логистической регрессии с регуляризацией, применяемой к одной из двух групп параметров модели. По сравнению с алгоритмом Varag, предложенный подход достиг меньших значений функции при меньшем числе вызовов оракулов.

Отметим также, что если функция $F(x, y)$ — μ -сильно выпуклая по совокупности переменных, то функция $g(y) = \min_{x \in Q_x} F(x, y)$ также будет μ -сильно выпуклая. Более того, все это можно сформулировать в терминах (δ, μ, L) -оракула (см. [3] и цитированную там литературу). При $\mu = 0$ это сделано в [20], при $\mu > 0$ доказательство практически дословно повторяет утверждения 1 и 3 из [20] (см. также [9]). Приведенное наблюдение позволяет обоснованно (с теоретической проработкой) использовать для решения внутренней задачи метод Вайды, а для решения внешней задачи использовать, например, быстрый градиентный метод. Однако такой подход будет предпочтительнее рассмотренного в данной статье только при весьма специальных (как правило, трудно выполнимых) условиях [5].

ПРИЛОЖЕНИЕ

Доказательство утверждения 1. Рассмотрим произвольный $y \in Q_y$

$$(П.1) \quad \langle \nabla g(\tilde{y}), \tilde{y} - y \rangle = \langle \nabla g(\tilde{y}) - \nabla g(y_*), \tilde{y} - y \rangle + \langle \nabla g(y_*), \tilde{y} - y \rangle.$$

Оценим сверху первое слагаемое, используя неравенство Коши–Буняковского и определение липшицевости градиента:

$$(П.2) \quad \begin{aligned} \langle \nabla g(\tilde{y}) - \nabla g(y_*), \tilde{y} - y \rangle &\leq \|\nabla g(\tilde{y}) - \nabla g(y_*)\|_2 \|\tilde{y} - y\|_2 \leq \\ &\leq L \|\tilde{y} - y_*\|_2 \|\tilde{y} - y\|_2. \end{aligned}$$

Из сильной выпуклости следует, что

$$g(\tilde{y}) \geq g(y_*) + \langle \nabla g(y_*), \tilde{y} - y_* \rangle + \frac{\mu}{2} \|\tilde{y} - y_*\|_2^2.$$

Воспользовавшись неравенствами $g(\tilde{y}) - g(y_*) \leq \varepsilon$ и $\langle \nabla g(y_*), y - y_* \rangle \geq 0 \forall y \in Q_y$, получим

$$(П.3) \quad \|\tilde{y} - y_*\|_2 \leq \sqrt{\frac{2\varepsilon}{\mu}} \stackrel{(П.2)}{\implies} \langle \nabla g(\tilde{y}) - \nabla g(y_*), \tilde{y} - y \rangle \leq L \|\tilde{y} - y\|_2 \sqrt{\frac{2\varepsilon}{\mu}}.$$

Теперь оценим сверху второе слагаемое в (П.1)

$$\langle \nabla g(y_*), \tilde{y} - y \rangle = \langle \nabla g(y_*), \tilde{y} - y_* \rangle + \langle \nabla g(y_*), y_* - y \rangle.$$

Снова воспользовавшись критерием оптимальности точки y_* и неравенством Коши–Буняковского, получим

$$\langle \nabla g(y_*), \tilde{y} - y \rangle \leq \|\nabla g(y_*)\|_2 \|\tilde{y} - y_*\|_2 \stackrel{(\text{П.3})}{\leq} \|\nabla g(y_*)\|_2 \sqrt{\frac{2\varepsilon}{\mu}}.$$

Объединив верхние оценки для обоих слагаемых, получим

$$\langle \nabla g(\tilde{y}), \tilde{y} - y \rangle \leq (L \|\tilde{y} - y\|_2 + \|\nabla g(y_*)\|_2) \sqrt{\frac{2\varepsilon}{\mu}},$$

откуда следует доказываемое утверждение 1.

Доказательство теоремы 4. Зафиксировав $x \in Q_x$, применим утверждение 1 к функции $g(y) := F(x, y)$ и утверждение 2. Теорема 4 доказана.

Доказательство теоремы 5. Согласно (4) алгоритм 2 сходится линейно, поэтому можно считать, что вспомогательная задача $\min_{y \in Q_y} F(x, y)$ решается сколь угодно точно за время $\tilde{O}\left(\sqrt{\frac{L}{\mu}}\right)$. Согласно теореме 4 это позволяет использовать δ -субградиент, где δ убывает со скоростью геометрической прогрессии. Для внешней задачи используется метод Вайды, который также сходится линейно и имеет сложность $\tilde{O}(d)$. Таким образом, для решения задачи (13) достаточно $\tilde{O}(d)$ вычислений $\partial_x F$ и обращений матриц размера $d \times d$, а также $\tilde{O}\left(d\sqrt{\frac{L}{\mu}}\right)$ вычислений $\nabla_y F$. Теорема 5 доказана.

Доказательство теоремы 6. Согласно теореме 3 Varag сходится линейно, поэтому можно считать, что вспомогательная задача $\min_{y \in Q_y} F(x, y)$ решается сколь угодно точно за время $\tilde{O}\left(m + \sqrt{\frac{mL}{\mu}}\right)$. Согласно теореме 4 это позволяет использовать δ -субградиент, где δ убывает со скоростью геометрической прогрессии. Для внешней задачи используется метод Вайды, который также сходится линейно и имеет сложность $\tilde{O}(d)$ итераций. На каждой его итерации необходимо вычислять субградиенты всех m слагаемых $\partial_x F_i$. Таким образом, для решения задачи достаточно $\tilde{O}(md)$ вычислений $\partial_x F_i$, $\tilde{O}(d)$ обращений матриц размера $d \times d$ и $\tilde{O}\left(dm + d\sqrt{\frac{mL}{\mu}}\right)$ вычислений $\nabla_y F_i$. Теорема 6 доказана.

СПИСОК ЛИТЕРАТУРЫ

1. *Нестеров Ю.Е.* Метод минимизации выпуклых функций со скоростью сходимости $O(1/k^2)$ // Докл. АН СССР. 1983. Т. 269. № 3. С. 543–547.
2. *Lan G.* First-order and Stochastic Optimization Methods for Machine Learning. Atlanta: Springer, 2020.
3. *Гасников А.В.* Современные численные методы оптимизации. Метод универсального градиентного спуска. М.: МЦНМО, 2020.
4. *Alkousa M.S., Dvinskikh D.M., Stonyakin F.S., Gasnikov A.V., Kovalev D.* Accelerated Methods for Saddle Point Problems // Comput. Math. Math. Phys. 2020. V. 60. No. 11. P. 1787–1809.

5. *Gladin E., Kuruzov I., Stonyakin F., Pasechnyuk D., Alkousa M., Gasnikov A.* Solving strongly convex-concave composite saddle point problems with a small dimension of one of the variables. <https://arxiv.org/pdf/2010.02280.pdf>
6. *Tianyi L., Chi J., Michael I.J.* Near-Optimal Algorithms for Minimax Optimization. <https://arxiv.org/pdf/2002.02417v5.pdf>
7. *Yuanhao W., Jian L.* Improved Algorithms for Convex-Concave Minimax Optimization. <https://arxiv.org/pdf/2006.06359.pdf>
8. *Zhongruo Wang, Krishnakumar Balasubramanian, Shiqian Ma, Meisam Razaviyayn.* Zeroth-Order Algorithms for Nonconvex Minimax Problems with Improved Complexities. <https://arxiv.org/pdf/2001.07819.pdf>
9. *Гасников А.В., Гасникова Е.В.* Модели равновесного распределения транспортных потоков в больших сетях. Уч. пос. М.: МФТИ, 2020.
10. *Bolte J., Glaudin L., Pauwels E., Serrurier M.* A Hölderian backtracking method for min-max and min-min problems. <https://arxiv.org/pdf/2007.08810.pdf>
11. *Jungers M., Trélat E., Abou-Kandil H.* Min-Max and Min-Min Stackelberg Strategies with Closed-Loop Information Structure // J. Dynamical and Control Syst. Springer Verlag, 2011. No. 17 (3). P. 387–425.
12. *Konur D., Farhangi H.* Set-based Min-max and Min-min Robustness for Multi-objective Robust Optimization // Proc. 2017 Industrial and Systems Engineering Research Conf. K. Coperich, E. Cudney, H. Nembhard, eds.
13. *Vaidya P.M.* A New Algorithm for Minimizing Convex Functions over Convex Sets // Foundations of Computer Science, 1989. 30th Annual Sympos. 1989. P. 338–343.
14. *Vaidya P.M.* A new algorithm for minimizing convex functions over convex sets // Mathematical Programming 73. Springer, 1996. P. 291–341.
15. *Lan G., Zhize Li, Yi Zhou.* A unified variance-reduced accelerated gradient method for convex optimization // 33rd Conf. on Neural Information Processing Systems (NeurIPS 2019). Vancouver, Canada. <https://arxiv.org/pdf/1905.12412.pdf>
16. *Tyurin A.I., Gasnikov A.V.* Fast Gradient Descent Method for Convex Optimization Problems with an Oracle That Generates a (δ, L) -model of a Function in a Requested Point // Comput. Math. Math. Phys. 2019. V. 59. No. 7. P. 1137–1150.
17. *Bubeck S.* Convex Optimization: Algorithms and Complexity // Foundations and Trends in Machine Learning. 2015. V. 8. No. 3-4. P. 231–357.
18. *Gladin E., Sadiev A., Gasnikov A., Stonyakin F., Dvurechensky P., Beznosikov A., Alkousa M.* Solving smooth min-min and min-max problems by mixed oracle algorithms. <https://arxiv.org/pdf/2103.00434.pdf>
19. *Поляк Б.Т.* Введение в оптимизацию. М.: Наука, 1983.
20. *Гасников А.В., Двуреченский П.Е., Камзолов Д.И., Нестеров Ю.Е., Спокойный В.Г., Стецюк П.И., Суворикова А.Л., Чернов А.В.* Поиск равновесий в многостадийных транспортных моделях // Тр. Московского физико-технического института. 2015. № 7.4 (28).
21. *Bishop C.* Pattern recognition and machine learning. Springer, 2006.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 28.01.2021

После доработки 26.04.2021

Принята к публикации 30.06.2021

© 2021 г. В.П. КОРНЕЕНКО, канд. техн. наук (vkorn@ipu.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ЭФФЕКТИВНЫЙ АЛГОРИТМ ТУПИКОВЫХ УПРАВЛЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Предлагается алгоритм тупиковых управлений, предназначенный для точного решения NP -трудных задач комбинаторной оптимизации. Эффективность алгоритма демонстрируется на примерах решения задачи разбиения на равные части и задачи об одномерном рюкзаке. В статье также показано, что применение идеи тупиковых управлений при реализации метода динамического программирования позволяет значительно сократить на каждом шаге оптимизации число переменных состояний задачи. Проведен сравнительный анализ предлагаемого метода с известными алгоритмами решения этих задач.

Ключевые слова: тупиковое управление, функция Беллмана, алгоритм, задача разбиения, задача о рюкзаке.

DOI: 10.31857/S000523102110007X

1. Введение

В настоящее время на практике получили распространение два основных оптимальных метода решения задач комбинаторной оптимизации, к которым относится задача об одномерном рюкзаке (0-1 knapsack problem) и сводящаяся к ней задача разбиения на равные части (set-partition problem), а именно: метод ветвей и границ в рамках статичной модели, когда параметры задачи не меняются во времени, и различные модификации метода динамического программирования [1–3]. Подробный обзор различных методов и алгоритмов, разработанных для решения задачи о рюкзаке, изложены в [4].

Наряду с модификациями метода динамического программирования для решения задачи о рюкзаке применяются и приближенные алгоритмы, в частности жадный алгоритм и аппроксимационный алгоритм, подробно изложенные в [2, с. 448–478; 5, с. 417–424] соответственно.

Комбинированные эвристические алгоритмы для задачи о рюкзаке подробно изложены в [6]. В [7–9] представлен графический подход к решению задачи разбиения на равные части и задачи об одномерном рюкзаке.

Несмотря на то что данные задачи относятся к классу NP -трудных, в последнее время в большом количестве научных публикаций появляются новые алгоритмы для задачи о ранце в виде как точных алгоритмов, так и приближенных, включая и эвристические алгоритмы, обладающие различной временной трудоемкостью [10–12].

Рассмотрим математические постановки задач комбинаторной оптимизации.

1. Задача разбиения на равные части (partition) состоит в следующем: задано множество целых чисел $B = \{b_1, \dots, b_n\}$. Требуется разбить множество B на два непересекающихся подмножества B_1 и B_2 так, чтобы минимизировать значение:

$$(1) \quad \left| \sum_{b_j \in B_1} b_j - \sum_{b_j \in B_2} b_j \right| \rightarrow \min.$$

2. Задача об одномерном рюкзаке (0-1 knapsack). В общем виде вербальная постановка задачи сводится к следующему: из заданного множества предметов, характеризующихся для j -го предмета “ценностью” p_j и “весом (объемом)” w_j , требуется отобрать такое число предметов, чтобы получить максимальную суммарную ценность при одновременном соблюдении ограничения b на суммарный вес или объем.

Математическую постановку целочисленной задачи представим в виде задачи булевого линейного программирования:

$$(2) \quad \begin{cases} f(x_1, \dots, x_n) = \sum_{j=1}^n p_j x_j \rightarrow \max_{x_1, \dots, x_n}, \\ \sum_{j=1}^n w_j x_j \leq b, \quad x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n. \end{cases}$$

Когда $p_j = w_j = b$, $j = \overline{1, n}$, и $b = \frac{1}{2} \sum_{j=1}^n b_j$, то задачи (1) и (2) являются эквивалентными.

Идея тупиковых управлений взята из задачи минимизации и получения сокращенной (тупиковой) дизъюнктивной нормальной формы (ДНФ) [13], представляющей собой произвольную дизъюнкцию элементарных конъюнкций логических функций булевой алгебры, которую нельзя упростить. По аналогии с ДНФ допустимое управление, в котором замена произвольной нулевой компоненты в булевском векторе на единицу приводит к нарушению ресурсных ограничений, в задаче о ранце будем называть тупиковым.

В данной статье предлагается новый оптимальный алгоритм тупиковых управлений, который по своей эффективности на данный момент превосходит опубликованные алгоритмы для решения задачи разбиения на равные части и задачи об одномерном рюкзаке.

Кроме того, предлагаемый алгоритм может быть применим в задачах выполнения комплекса взаимосвязанных работ, математическая постановка которых представлена на динамических управляемых моделях [14]. В статье демонстрируется применение тупиковых управлений при решении задачи о рюкзаке методом динамического программирования.

2. Алгоритм тупиковых управлений

Рассмотрим шаги *алгоритма* тупиковых управлений более подробно.

Шаг 1. Упорядочение номеров предметов в порядке убывания весов. Не умаляя общности, будем предполагать, что для любого предмета справедливо неравенство:

$$w_j \leq b \quad \forall j = \overline{1, n}.$$

Построение тупиковых управлений начинается с упорядочения номеров предметов в порядке убывания весов (объемов):

$$w_{j_1} \geq \dots \geq w_{j_k} \geq \dots \geq w_{j_n}.$$

Пусть $u_k \equiv x_{j_k} \in \{0, 1\}$ – переменная управления, принимающая значение единица, если k -й предмет по порядку помещается в рюкзак, и нулевое значение в противном случае. В связи с перенумерацией переменных математическая постановка задачи о рюкзаке сводится к нахождению такого вектора управления $\vec{u} = (u_1, \dots, u_n)$, который доставляет максимум целевой функции:

$$(3) \quad f(\vec{u}) = \sum_{k=1}^n p_k u_k \rightarrow \max_{u_1, \dots, u_n}$$

при условии

$$(4) \quad \sum_{k=1}^n w_k u_k \leq b, \quad u_k \in \{0, 1\}, \quad k = 1, 2, \dots, n.$$

Очевидно, что задача (3)–(4) эквивалентна задаче (2).

Шаг 2. Построение тупиковых управлений. В первое тупиковое управление включаем первый предмет, который соответствует управлению $u_1 = 1$, если он не нарушает ограничения (4), т.е. выполняется неравенство: $w_1 u_1 \leq b$, в противном случае полагаем $u_1 = 0$. Точно так же поступаем со вторым, третьим и так далее по порядку и с n -м предметом в соответствии с формулой:

$$(5) \quad u_k = \begin{cases} 1, & \text{если } \sum_{i=1}^k u_i w_i \leq b; \\ 0, & \text{если } \sum_{i=1}^k u_i w_i > b, \end{cases}$$

последовательно для $k = 1, 2, \dots, n$.

В результате получим первое тупиковое управление, состоящее из нулей и единиц

$$(6) \quad \vec{u}_1 = \left(u_1^{(1)}, u_2^{(1)}, \dots, u_n^{(1)} \right).$$

Построенному вектору управления \vec{u}_1 (6) соответствует некоторое двоичное число

$$(7) \quad \xi_1 = (11010..1..01001),$$

где единицы стоят в тех разрядах, номера которых совпадают с номерами предметов, включенных в управление \vec{u}_1 (6). Замена любого нуля единицей делает это управление недопустимым по ограничению (4).

С помощью первого тупикового управления построим второе. Для этого найдем самый младший разряд числа ξ_1 , в котором записан ноль. Во всех разрядах справа от него вместо единиц записываем нули. В полученном двоичном числе первую справа единицу перенесем на один разряд вправо. Если полученное управление недопустимо по ограничению (4), то эту единицу сдвигаем еще на один разряд вправо до тех пор, пока управление не окажется допустимым.

Далее в разряды справа от этой единицы помещаем единицы по тому же правилу (5), что и при построении числа ξ_1 (7). В результате получаем двоичное число $\xi_2 < \xi_1$. Этому числу соответствует тупиковое управление $\vec{u}_2 = (u_1^{(2)}, u_2^{(2)}, \dots, u_n^{(2)})$. Точно таким же образом из тупикового управления \vec{u}_2 строится тупиковое управление \vec{u}_3 , которому соответствует двоичное число $\xi_3 < \xi_2 < \xi_1$ и т.д. В результате получаем множество тупиковых управлений

$$(8) \quad U = \left\{ \vec{u}_l = \left(u_1^{(l)}, \dots, u_k^{(l)}, \dots, u_n^{(l)} \right) \mid l = 1, 2, \dots, N \right\},$$

которому соответствует упорядочение двоичных чисел:

$$\xi_N < \xi_{N-1} < \dots < \xi_3 < \xi_2 < \xi_1.$$

Описанная процедура дает возможность получить все тупиковые управления, удовлетворяющие ограничению (4).

Шаг 3. Вычисление оптимального тупикового управления. Для каждого тупикового управления \vec{u}_l , $l = \overline{1, N}$, находим значение целевой функцией $f(\vec{u}_l)$ (3).

За оптимальное управление принимаем тупиковое $\vec{u}_* \in U$ (8), обеспечивающее максимальное значение целевой функции $f(\vec{u})$ (3):

$$\vec{u}_* = (u_1^*, \dots, u_k^*, \dots, u_n^*) = \arg \max_{\vec{u}_l \in U} f \left(u_1^{(l)}, \dots, u_k^{(l)}, \dots, u_n^{(l)} \right).$$

Обоснование изложенного алгоритма решения задачи о рюкзаке базируется на следующих теоремах.

Теорема 1 (о существовании оптимального решения). *Среди всех тупиковых управлений из множества U (8) найдется по крайней мере одно тупиковое управление, обеспечивающее максимум целевой функции задачи о рюкзаке (3)–(4).*

Доказательство. Пусть \vec{u}_* – оптимальное тупиковое управление, являющееся решением задачи (3)–(4). Предположим обратное, что \vec{u}_* не является оптимальным и тупиковым управлением, т.е. $\vec{u}_* \notin U$ (8). Дополним его до тупикового $\vec{u}_0 \in U$ (8).

Это значит, что найдется предмет, который войдет в рюкзак, и значение целевой функции при этом увеличится на величину ценности этого предмета. При этом для целевой функции $f(\vec{u})$ (3) будет выполняться неравенство, т.е.

$$f(\vec{u}_0) > f(\vec{u}_*).$$

Получили противоречие с тем, что нашлось решение \vec{u}_0 лучшее, чем \vec{u}_* . Следовательно, предположение \vec{u}_* не верно, что и требовалось доказать. Теорема 1 доказана.

От исходной прямой задачи (3)–(4) максимизации ценности рюкзака перейдем к двойственной задаче минимизации остатка веса (объема) рюкзака по критерию:

$$(9) \quad g(\vec{u}_l) = b - \sum_{k=1}^n w_k u_k^{(l)} \rightarrow \min_{\vec{u}_l \in U},$$

который равносителен максимизации суммы весов предметов, помещаемых в рюкзак.

Очевидно, что для функции $g(\vec{u}_l)$ (9) должно выполняться неравенство

$$(10) \quad g(\vec{u}_l) = b - \sum_{k=1}^n w_k u_k^{(l)} \geq 0 \quad \forall l = 1, 2, \dots, N,$$

где b – объем (вес) рюкзака, w_k – вес k -го предмета управления $\vec{u}_l \in U$ (8).

При этом имеет место следующее утверждение.

Теорема 2 (о связи прямой и двойственной задач). Пусть

$$\vec{u}_f = (u_1^{(f)}, \dots, u_k^{(f)}, \dots, u_{j_n}^{(f)}) \quad \text{и} \quad \vec{u}_g = (u_1^{(g)}, \dots, u_k^{(g)}, \dots, u_{j_n}^{(g)})$$

– оптимальные решения прямой $f(\vec{u}_l)$ (3)–(4) и двойственной $g(\vec{u}_l)$ (9)–(10) задач соответственно.

Тогда для весов предметов, помещаемых в рюкзак, справедливо неравенство

$$(11) \quad \sum_{k=1}^n w_k u_k^{(f)} \leq \sum_{k=1}^n w_k u_k^{(g)}.$$

Доказательство. Пусть

$$(12) \quad U_{\min} = \left\{ \vec{u}_g \in U(8) \mid \vec{u}_g = \arg \min_{\vec{u}_l \in U} g(\vec{u}_l) \right\}$$

— подмножество множества U (8) тупиковых управлений, на которых критерий $g(\vec{u}_l)$ (9) достигает минимального значения, которое не всегда совпадает с точной нижней гранью U (8), а именно: $\inf g(\vec{u}_l) = 0$, $\vec{u}_l \in U$ (8).

Очевидно, что если оптимальные решения прямой $f(\vec{u}_l)$ (3)–(4) задачи $\vec{u}_f \in U_{\min}$ (12), то неравенство (11) выполняется как равенство.

Покажем, что если $\vec{u}_f \notin U_{\min}$ (12), то неравенство (11) выполняется как строгое. Действительно, в соответствии с определением минимума целевой функции [5] должно выполняться неравенство

$$g(\vec{u}_g) < g(\vec{u}_f) \quad \forall \vec{u}_f \notin U_{\min} \quad (12),$$

т.е.

$$b - \sum_{k=1}^n w_k u_k^{(g)} < b - \sum_{k=1}^n w_k u_k^{(f)} \Rightarrow \sum_{k=1}^n w_k u_k^{(g)} > \sum_{k=1}^n w_k u_k^{(f)}.$$

Отсюда следует справедливость неравенства (11), что и требовалось доказать. Теорема 2 доказана.

3. Метод динамического программирования с тупиковыми управлениями на примере решения задачи о рюкзаке в прямом времени

Используя основную идею динамического программирования [1], сведем решение задачи (3)–(4), поставленной на статичной математической модели, к задаче оптимизации управляемой динамической системы, решение которой сводится к следующим этапам.

1. Этап инвариантного погружения. Для этого вложим задачу о рюкзаке в семейство задач той же природы, в результате чего получим управляемую систему в прямом времени:

$$(13) \quad Z = \left\{ Z_k : \max_{u_1, \dots, u_k} \sum_{j=1}^k p_j u_j, \quad \sum_{j=1}^k w_j u_j \leq s_k, \quad k = \overline{1, n} \right\},$$

где $u_k \in \{0, 1\}$ – управление на k -м шаге оптимизации, s_k – переменная состояния, характеризующая остаточный вес рюкзака.

Множество допустимых значений переменной состояния s_k управляемой системы (13) для u_k -го управления будем обозначать через S_k , являющееся подмножеством множества числовых значений параметров $S = \{0, 1, \dots, b\}$, связанным с семейством задач (13). Исходная задача очевидным образом входит в рассматриваемое семейство, если в (13) положить $k = n$ и $s_n = b$. Поскольку из семейства задач выделяется исходная задача, то семейство задач Z (13) реализует принцип инвариантного погружения в прямом времени.

Пусть на первом шаге $k = 1$ осуществляется выбор переменной управления u_1 при некотором выборе переменных u_2, u_3, \dots, u_n таким, что

$$s_1 = b - \sum_{j=2}^n w_j u_j,$$

где $0 \leq s_1 \leq b$.

Величина s_1 характеризует тот остаток общего ресурса b , который можно использовать при выборе u_1 . Перейдя ко второму шагу, а затем и к третьему и далее к k -му шагу, будем рассматривать остаток общего ресурса b на k -м шаге как s_k состояние процесса выбора управления u_k управляемой системы (13).

Из ресурсных ограничений

$$(14) \quad 0 \leq \sum_{j=1}^k w_j u_j \leq s_k, \quad k = \overline{1, n},$$

имеем:

а) параметры состояния управляемой системы на k -м шаге оптимизации

$$(15) \quad S_k = s_k = \left\{ b - \sum_{j=k+1}^n w_j u_j \mid u_j \in \{0, 1\}, \quad k = 1, \dots, n \right\},$$

где $s_0 = 0$ — начальное состояние на первом шаге $k = 1$,

$s_n = b$ — конечное состояние на последнем шаге $k = n$;

б) уравнения состояний в прямом времени

$$s_{k-1} = s_k - w_k u_k,$$

связывающие переменные состояний функций Р. Беллмана на шагах оптимизации $k-1$ и k -м.

Множества значений функции Беллмана на k -м шаге оптимизации представим в виде

$$(16) \quad S_k = \{ \underline{s}_k, \underline{s}_k + 1, \dots, b \},$$

где минимальное значение переменной состояния $s_k \in S_k$ (15) определяется из условия:

$$(17) \quad \underline{s}_k = \min_{u_{k+1}, \dots, u_n} \left\{ b - \sum_{j=k+1}^n w_j u_j \right\},$$

причем с учетом неравенства (14) переменные управления u_{k+1}, \dots, u_n должны удовлетворять соотношению (5).

Поскольку для любых переменных u_{k+1}, \dots, u_n и не обязательно тупиковых на k -м шаге справедливо неравенство $\sum_{j=k+1}^n w_j u_j \leq \sum_{j=k+1}^n w_j$, то минимальное значение переменной состояния $s_k \in S_k$ (15) для задач большой размерности можно определять и из условия:

$$\underline{s}_k = \min_{u_{k+1}, \dots, u_n} \left\{ 0, \left| b - \sum_{j=k+1}^n w_j \right| \right\}.$$

Множество допустимых управлений k -го шага представим в виде:

$$U(s_k) = \left\{ u_k \in \{0, 1\} \mid 0 \leq u_k \leq \left\lfloor \frac{s_k}{w_k} \right\rfloor \right\},$$

где $\left\lfloor \frac{s_k}{w_k} \right\rfloor$ — целая часть числа $\frac{s_k}{w_k}$.

Отсюда под действием управления u_k система, находящаяся в состоянии s_{k-1} , перейдет в состояние s_k . Показатель эффективности k -го шага определим как $f_k = p_k u_k$.

Заключаем, что задача (3)–(4) поставлена как задача динамического программирования оптимизации управляемой системы.

2. Этап построения рекуррентных функциональных уравнения Р. Беллмана. На решениях задач Z (13) определим функцию Беллмана от k переменных управления u_1, \dots, u_k в виде

$$(18) \quad B_k(s_k) = \max_{u_1, \dots, u_k} \sum_{j=1}^k p_j u_j, \quad k = 1, 2, \dots, n,$$

с областью определения S_k (16), характеризующую суммарную ценность рюкзака от первого шага до k -го шага.

Так как вычисление последовательности функций Беллмана $B_k(s_k)$ (18) происходит в направлении возрастания дискретного аргумента k (идет слева направо: $1, 2, \dots$), то для k -го шага имеем рекуррентное уравнение Беллмана в прямом времени в виде:

$$(19) \quad B_k(s_k) = \max_{u_k \leq \left\lfloor \frac{s_k}{w_k} \right\rfloor} \{p_k u_k + B_{k-1}(s_k - w_k u_k)\},$$

которое удовлетворяет начальному условию

$$B_1(s_1) = \max_{0 \leq u_1 \leq \left\lfloor \frac{s_1}{w_1} \right\rfloor} p_1 u_1.$$

Выбрав на k -м шаге некоторое произвольное управление u_k , система из состояния s_{k-1} придет в состояние s_k . Так как в оптимальном решении задач Z (13) должно быть либо 0, либо 1, то уравнения (19) запишутся в виде

$$B_k(s_k) = \max_{0 \leq u_k \leq \left\lfloor \frac{s_k}{w_k} \right\rfloor} \left\{ \begin{array}{ll} p_k, & u_k = 1 \\ B_{k-1}(s_{k-1}), & u_k = 0 \end{array} \right\}, \quad k = 1, 2, \dots, n.$$

Таким образом, для любого значения $s_k \in S_k$ определение величины $B_k(s_k)$ сводится к простейшей задаче оптимизации — сравнению двух чисел, начальное условие при этом для начального шага $k = 1$ запишется в виде:

$$B_1(s_1) = \max_{0 \leq u_1 \leq \left\lfloor \frac{s_1}{w_1} \right\rfloor} \left\{ \begin{array}{ll} p_1, & u_1 = 1 \\ 0, & u_1 = 0 \end{array} \right\}.$$

Дойдя до $k = n$, определим оптимальное значение целевой функции $B_n(s_n)$, совпадающей со значением целевой функции исходной задачи.

3. Этап решения рекуррентных функциональных уравнений. На данном этапе алгоритмом обратной прогонки, дойдя до $k = n$ шага, определим оптимальное значение функции $B_n(\bar{s}_n)$, $\bar{s}_n \in S_n$, совпадающей со значением целевой функции исходной задачи (3)–(4). Из условия $B_n(\bar{s}_n) = f_n(u_n^*) + B_{n-1}(\bar{s}_n - w_n u_n^*)$ имеем оптимальное управление $u_n^* = u_n^*(\bar{s}_n)$.

Далее последовательно на каждом шаге для $k = n, n-1, \dots, 1$, определяем оптимальные управления:

$$\begin{aligned} B_n(\bar{s}_n) &\rightarrow u_n^* = u_n^*(\bar{s}_n) \rightarrow \bar{s}_{n-1} = \bar{s}_n - w_n u_n^*; \\ &\dots \\ B_k(\bar{s}_k) &\rightarrow u_k^* = u_k^*(\bar{s}_k) \rightarrow \bar{s}_{k-1} = \bar{s}_k - w_k u_k^*; \\ &\dots \\ B_1(\bar{s}_1) &\rightarrow u_1^* = u_1^*(\bar{s}_1). \end{aligned}$$

Тогда в конце работы пошаговой процедуры получим оптимальное управление

$$(20) \quad \vec{u}_* = (u_1^*, \dots, u_n^*).$$

Необходимо заметить, что если на каждом шаге запоминать вектор управления вида $\vec{u}(k) = (u_1, \dots, u_k)$, $k = \overline{1, n}$, то на последнем шаге сразу можно выделить оптимальное решение \vec{u}^* (20).

4. Сравнительная оценка эффективности алгоритма

Для оценки эффективности алгоритма сравним его трудоемкость с традиционными методами решения задачи о рюкзаке. Продемонстрируем работу алгоритма на примерах решения задачи разбиения и задачи о рюкзаке.

Пример 1. На данных примера из [9, с. 320–323] решим задачу разбиения. Пусть числа множества $B = \{100, 70, 50, 20\}$, пронумерованы по невозрастанию, $n = 4$. Сведем задачу разбиения (1) к эквивалентной задаче о рюкзаке (3), что можно представить в виде

$$\begin{aligned} f(u_1, u_2, u_3, u_4) = 100u_1 + 70u_2 + 50u_3 + 20u_4 &\rightarrow \max_{u_1, \dots, u_4}, \\ 100u_1 + 70u_2 + 50u_3 + 20u_4 &\leq 120, \end{aligned}$$

где $b = \frac{1}{2} \sum_{j=1}^4 b_j = 120$.

Выполнив шаги 2 и 3 алгоритма, тупиковые управления и соответствующие им значения целевой функции представим в виде табл. 1.

В результате, построив четыре тупиковых управления, получаем два оптимальных решения:

$$\begin{aligned} B_1 = \{b_1, b_4\} &= \{100, 20\}, \quad B_2 = \{b_2, b_3\} = \{70, 50\}; \\ B_1 = \{b_2, b_3\} &= \{70, 50\}, \quad B_2 = \{b_1, b_4\} = \{100, 20\}. \end{aligned}$$

Таблица 1

l	U	u_1	u_2	u_3	u_4	$f(\vec{u}_l)$
1	\vec{u}_1	1	0	0	1	120
2	\vec{u}_2	0	1	1	0	120
3	\vec{u}_3	0	1	0	1	70
4	\vec{u}_4	0	0	1	1	70

Таблица 2

l	U	u_1	u_2	u_3	u_4	$f(\vec{u}_l)$
1	\vec{u}_1	1	0	0	1	8
2	\vec{u}_2	0	1	1	0	13
3	\vec{u}_3	0	1	0	1	11
4	\vec{u}_4	0	0	1	1	12

Как видно из табл. 1, понадобилось всего лишь четыре тупиковых управления, чтобы найти точное оптимальное решение. Для решения данной задачи разбиения графическим алгоритмом понадобилось рассмотреть семь точек (см. [9, с. 322]), а для алгоритма тупиковых управлений потребовалось построить всего лишь четыре управления (см. табл. 1).

Пример 2. Продемонстрируем работу алгоритма при решении задачи о рюкзаке на данных примера из [9, с. 326–333].

Постановку исходной задачи представим с учетом убывания весов предметов в виде

$$\begin{cases} f(\vec{u}) = 3u_1 + 6u_2 + 7u_3 + 5u_4 \rightarrow \max_{u_1, \dots, u_4}, \\ 7u_1 + 5u_2 + 3u_3 + 2u_4 \leq 9, \\ u_j \in \{0, 1\}, \quad j = 1, 2, 3, 4. \end{cases}$$

Выполнив шаги 2 и 3 алгоритма, тупиковые управления и соответствующие им значения целевой функции представим в виде табл. 2.

Построив четыре тупиковых управления, получаем оптимальное решение:

$$\vec{u}_* = \vec{u}_2 = (0, 1, 1, 0),$$

где

$$u_1 = x_4 = 0, \quad u_2 = x_3 = 1, \quad u_3 = x_2 = 1, \quad u_4 = x_1 = 0.$$

Для решения данной задачи о рюкзаке графический алгоритм вычисляет только 14 элементов [9, с. 333], в то же время для алгоритма тупиковых управлений потребовалось построить всего лишь четыре управления (см. табл. 2).

Пример 3. Рассмотрим задачу о рюкзаке вместимостью $b = 10$ для множества из 7 предметов, т.е. $J_0 = \{1, 2, 3, 4, 5, 6, 7\}$, вес и стоимость которых представлены в табл. 3 (исходные данные взяты из [15, с. 437]).

Таблица 3. Исходные данные задачи о рюкзаке

j	1	2	3	4	5	6	7
w_j	4	1	2	3	2	1	2
p_j	299	73	159	221	137	89	157

Таблица 4. Результаты решения методом тупиковых управлений

l	$\vec{u}_l = (u_1^{(l)}, \dots, u_7^{(l)})$	ξ_l – значение в десятичной системе счисления	$\sum_{k=1}^n w_k u_k$	$f(\vec{u}_l)$
1	1110010		114	752
2	1110001		113	768
3	1101010		106	730
4	1101001		105	746
5	1100110		102	750
6	1100101		101	766
7	1100011		99	682
8	1011100		92	752
9	1011011		91	757
10	1010111		87	777
11	1001111		79	755
12	0111110		62	747
13	0111101		61	763
14	0111011		59	679
15	0110111		55	699
16	0101111		47	677
17	0011111		31	615

1. Решение псевдополиномиальным алгоритмом динамического программирования ДП-III. Результатом решения является подмножество

$$(21) \quad J_* = \{1, 2, 3, 6, 7\},$$

где $p = \sum_{j \in J_*} p_j = 777$.

При этом алгоритм проходит через построение 91 пары (J, p) , где $J \subset J_0$. Трудоемкость данного алгоритма $O(n^2 p)$, где p – значение оптимальной стоимости [15, с. 436].

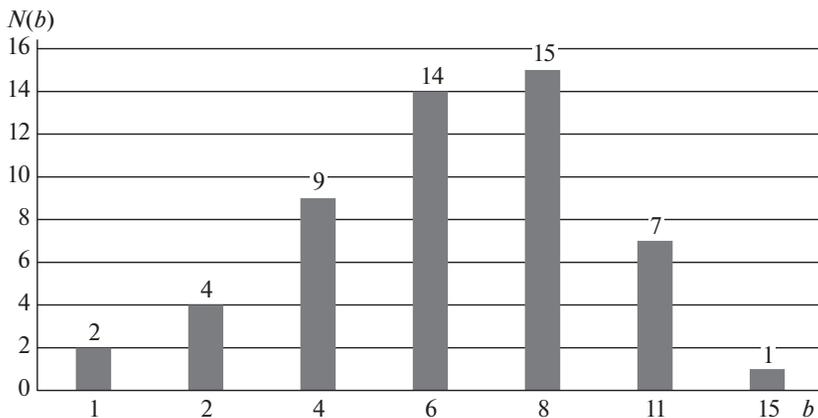
2. Решение задачи о рюкзаке методом тупиковых управлений. Упорядочим номера предметов в порядке убывания весов (объемов)

$$w_{11} > w_{42} > w_{33} \geq w_{54} \geq w_{75} > w_{26} \geq w_{67},$$

где $w_{11} = 4$, $w_{42} = 3$, $w_{33} = 2$, $w_{54} = 2$, $w_{75} = 2$, $w_{26} = 1$, $w_{67} = 1$.

Пусть $u_k \in \{0, 1\}$ – переменная управления, принимающая значение единица, если k -й предмет по порядку помещается в рюкзак, и нулевое значение в противном случае. Тогда постановку задачи о ранце представим в виде:

$$299u_1 + 221u_2 + 159u_3 + 137u_4 + 157u_5 + 73u_6 + 89u_7 \rightarrow \max$$



Зависимость числа тупиковых управлений от веса рюкзака.

при ограничениях

$$4u_1 + 3u_2 + 2u_3 + 2u_4 + 2u_5 + 1u_6 + 1u_7 \leq 10.$$

Построенные тупиковые управления и ценность предметов, попадающих в рюкзак, представлены в табл. 4.

Решением задачи является тупиковое управление $\vec{u}_{10} = (1010111)$, которое совпадает с решением J_* (21) в исходных обозначениях с величиной ценности рюкзака:

$$\sum_{j \in J_*} p_j = 299 + 159 + 157 + 73 + 89 = 777.$$

Возникает вопрос о зависимости числа тупиковых управлений от веса рюкзака. Можно предположить, что максимальное число тупиковых управлений будет приходиться на вес рюкзака, равный примерно половине суммы весов всех предметов.

На рисунке приведен график зависимости числа тупиковых управлений $N(b)$ от веса b рюкзака для исходных данных из табл. 3, где величина веса (объема) рюкзака определялась по формуле

$$b_k = \sum_{j=1}^k w_j, \quad k = 1, 2, \dots, 7; \quad 1 \leq w_1 \leq w_2 \leq \dots \leq w_7 \leq 4.$$

Из рисунка видно, что максимальное число тупиковых управлений приходится на вес рюкзака, равный не менее половины суммы весов всех предметов, и убывает, когда вес рюкзака возрастает, т.е.

$$\left[\frac{1}{2} \sum_{j=1}^n w_j \right] \leq b < \sum_{j=1}^n w_j,$$

где $\left[\frac{1}{2} \sum_{j=1}^n w_j \right]$ — целая часть числа $\frac{1}{2} \sum_{j=1}^n w_j$.

Таблица 5. Рекуррентные функциональные уравнения Р. Беллмана

Номер шага k	Функция Р. Беллмана $B_1(s_1)$	Тупиковое управление \vec{u}_l	Множество состояний $S_k(\vec{u}_l)$
1	$B_1(s_1) = \max_{0 \leq u_1 \leq \lfloor s_1/4 \rfloor} 299u_1, s_1 \in S_1$	0111110	$0 \div 10$
2	$B_2(s_2) = \max_{0 \leq u_2 \leq \lfloor \frac{s_2}{3} \rfloor} \{221u_2 + B_1(s_2 - 3u_2)\}$	0011111	$2 \div 10$
3	$B_3(s_3) = \max_{0 \leq u_3 \leq \lfloor \frac{s_3}{2} \rfloor} \{159u_3 + B_2(s_3 - 2u_3)\}$	0001111	$4 \div 10$
4	$B_4(s_4) = \max_{0 \leq u_4 \leq \lfloor \frac{s_4}{2} \rfloor} \{137u_4 + B_3(s_4 - 2u_4)\}$	0000111	$6 \div 10$
5	$B_5(s_5) = \max_{0 \leq u_5 \leq \lfloor \frac{s_5}{2} \rfloor} \{157u_5 + B_4(s_5 - 2u_5)\}$	0000011	$8 \div 10$
6	$B_6(s_6) = \max_{0 \leq u_6 \leq \lfloor \frac{s_6}{1} \rfloor} \{73u_6 + B_5(s_6 - u_6)\}$	0000001	9, 10
7	$B_7(s_7) = \max_{0 \leq u_7 \leq \lfloor \frac{s_7}{2} \rfloor} \{89u_7 + B_7(s_7 - u_7)\}$	0000001	10

Таблица 6. Результаты расчетов функций Р. Беллмана в прямом времени

k	Шаг 1		Шаг 2		Шаг 3		Шаг 4		Шаг 5		Шаг 6		Шаг 7		
	s_k	u_1	$B_1(s_1)$	u_2	$B_2(s_2)$	u_3	$B_3(s_3)$	u_4	$B_4(s_4)$	u_5	$B_5(s_5)$	u_6	$B_6(s_6)$	u_7	$B_7(s_7)$
0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
2	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-
3	0	0	0	1	221	-	-	-	-	-	-	-	-	-	-
4	0	1	299	0	299	0	299	-	-	-	-	-	-	-	-
5	0	1	299	0	299	0	380	-	-	-	-	-	-	-	-
6	0	1	299	0	299	0	458	0	458	-	-	-	-	-	-
7	0	1	299	0	520	0	520	0	520	-	-	-	-	-	-
8	0	1	299	0	520	0	520	0	595	0	615	-	-	-	-
9	0	1	299	0	520	0	679	0	657	0	677	0	688	-	-
10	0	1	299	0	520	0	679	1	679	0	752	0	750	0	777

В этом случае можно предположить, что мощность $|U|$ множества тупиковых U (8), как правило, значительно меньше числа bn , т.е. $|U| \leq cn$, где $c > 0$, поэтому трудоемкость алгоритма можно оценить как $O(n)$.

Таблица 7. Результаты сравнений методов решения задачи о рюкзаке

Метод решения	Число прямых вычислений	Процент от полного перебора
Полный перебор	128	100%
Метод динамического программирования ДП III	91	$\frac{91}{128} \times 100\% \approx 71\%$
Метод динамического программирования с ТУ	38	$\frac{38}{128} \times 100\% \approx 30\%$
Метод тупиковых управлений	17	$\frac{17}{128} \times 100\% \approx 13\%$

Представляет интерес нахождение функциональной зависимости числа тупиковых управлений от веса рюкзака. Данную проблему, хотя бы для частных случаев, автор предлагает исследовать читателю.

3. Решение задачи методом динамического программирования с тупиковым управлением. Расчет значений функций Беллмана ведется по всем допустимым u_k от начала к концу, $k = 1, 2, \dots, 7$.

Исходное дискретное множество области значений функции Р. Беллмана $S_0 = \{0, 1, 2, \dots, 10\}$. В табл. 5 сведены функции Р. Беллмана на каждом шаге оптимизации и множество состояний в зависимости от тупиковых управлений рекуррентных функциональных уравнений Р. Беллмана.

Результаты расчетов $B_k(s_k)$ представлены в табл. 6, где оптимальное управление выделено подчеркиванием (в скобках после значения функции Р. Беллмана текущего шага указано значение переменной состояния предшествующей функции Р. Беллмана).

Алгоритмом обратной прогонки находим оптимальное управление:

$$\begin{aligned}
 B_7(\bar{s}_7)_{|\bar{s}_7=10} &= 777 \rightarrow u_7^*(\bar{s}_7)_{|\bar{s}_7=10} = 1 \rightarrow \bar{s}_6 = \bar{s}_7 - 1u_7^* = 10 - 1 = 9; \\
 B_6(\bar{s}_6)_{|\bar{s}_6=9} &= 688 \rightarrow u_6^*(\bar{s}_6)_{|\bar{s}_6=9} = 1 \rightarrow \bar{s}_5 = \bar{s}_6 - 1u_6^* = 9 - 1 = 8; \\
 B_5(\bar{s}_5)_{|\bar{s}_5=8} &= 615 \rightarrow u_5^*(\bar{s}_5)_{|\bar{s}_5=8} = 1 \rightarrow \bar{s}_4 = \bar{s}_5 - 2u_5^* = 8 - 2 = 6; \\
 B_4(\bar{s}_4)_{|\bar{s}_4=6} &= 458 \rightarrow u_4^*(\bar{s}_4)_{|\bar{s}_4=6} = 0 \rightarrow \bar{s}_3 = \bar{s}_4 - 2u_4^* = 6 - 0 = 6; \\
 B_3(\bar{s}_3)_{|\bar{s}_3=6} &= 458 \rightarrow u_3^*(\bar{s}_3)_{|\bar{s}_3=6} = 1 \rightarrow \bar{s}_2 = \bar{s}_3 - 2u_3^* = 6 - 2 = 4; \\
 B_2(\bar{s}_2)_{|\bar{s}_2=4} &= 299 \rightarrow u_2^*(\bar{s}_2)_{|\bar{s}_2=4} = 0 \rightarrow \bar{s}_1 = \bar{s}_2 - 3u_2^* = 4 - 0 = 4; \\
 B_1(\bar{s}_1)_{|\bar{s}_1=4} &= 299 \rightarrow u_1^*(\bar{s}_1)_{|\bar{s}_1=4} = 1.
 \end{aligned}$$

Отсюда $\vec{u}_* = (u_1^*, u_2^*, u_3^*, u_4^*, u_5^*, u_6^*, u_7^*) = (1010111)$, что соответствует в исходных обозначениях решению J_* (21). Сравнительный анализ по трудоемкости методов решений задачи о рюкзаке представлен в табл. 7.

Из табл. 7 следует, что применение тупиковых управлений оказалось эффективнее в $\frac{91}{17} \approx 5,3$ раза, чем алгоритм ДП III [2, 16], и в $\frac{38}{17} \approx 2,2$ раза, чем метод динамического программирования с тупиковым управлением.

Традиционно считается, что временная сложность метода динамического программирования линейна по числу этапов, что является его достоинством. Если число состояний на каждом шаге ограничено константой b , то временная сложность для задачи распределения ресурсов с небулевым управлением может быть оценена как $O(b^2n)$ [5]. Временная сложность алгоритма с булевым управлением обычно не превышает величины $O(nb)$ [8]. Покажем, что если определять множество допустимых состояний k -го шага по формуле S_k (16), то временную сложность вычислений можно оценить как $O(n)$ за n шагов алгоритма.

Теорема 3 (о трудоемкости метода динамического программирования с тупиковым управлением). Пусть нижняя граница переменной состояния $s_k \in S_k$ (15) на k -м шаге определяется по формуле \underline{s}_k (17), тогда временная сложность алгоритма динамического программирования с тупиковым управлением решения задачи о рюкзаке в прямом времени будет удовлетворять неравенству

$$(22) \quad \sum_{k=1}^n |S_k| \leq cn, \quad 0 < c < b,$$

что равносильно оценке временной сложности как $O(n)$, где $|S_k|$ – число состояний на k -м шаге оптимизации.

Доказательство. Поскольку для переменной состояний на k -м шаге выполняется неравенство

$$s_k = b - \sum_{j=k+1}^n w_j u_j \geq 0,$$

то для любых (тупиковых) управлений

$$\vec{u}(k) = (u_k, u_{k+1}, \dots, u_n) \quad \text{и} \quad \vec{u}(k+1) = (u_{k+1}, u_{k+2}, \dots, u_n)$$

справедливо неравенство

$$b = \sum_{j=1}^n w_j \geq \dots \geq \sum_{j=k}^n w_j u_j \geq \sum_{j=k+1}^n w_j u_j \geq \dots \geq w_n > 0 \quad \forall k = \overline{1, n-1},$$

т.е.

$$b \geq |S_1| \geq |S_2| \geq \dots \geq |S_n| = 1.$$

Отсюда следуют справедливость неравенства (22) и оценка временной сложности $O(n)$. Теорема 3 доказана.

5. Заключение

В статье рассмотрен эффективный алгоритм тупиковых управлений для решения задач комбинаторной оптимизации, относящийся к классу точных оптимальных алгоритмов с временной сложностью $O(n)$. В [8, 9] показано, что графический алгоритм решения задач комбинаторной оптимизации обладает временной сложностью $O(n)$, однако при этом, как показано в примерах 1 и 2, алгоритм тупиковых управлений оказался более эффективным. По своей эффективности, на данный момент, алгоритм тупиковых управлений превосходит известные алгоритмы, включая алгоритм *Balsub*, представленный в [4].

Также показано, что применение идеи тупиковых управлений при реализации метода динамического программирования позволяет значительно сократить на каждом шаге оптимизации число переменных состояний задачи. Достоинствами метода тупиковых управлений являются его вычислительная простота и более высокое быстродействие по сравнению с известными алгоритмами, что позволяет решать с его помощью характерные для практики задачи большой размерности.

СПИСОК ЛИТЕРАТУРЫ

1. *Беллман Р.* Динамическое программирование. М.: Мир, 1960.
2. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ. М.: Издательский дом “Вильямс”, 2013.
3. *Pisinger D.* A Minimal Algorithm for the 0-1 Knapsack Problem // University of Copenhagen. Oper. Res. 1997. V. 46. No. 5. P. 758–767.
4. *Kellerer H., Pferschy U., Pisinger D.* Knapsack Problems. Springer Science. Business Media, 2010.
5. *Корнеев В.П.* Методы оптимизации. М.: Высш. шк., 2007.
6. *Сигал И.Х., Иванова А.П.* Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. М.: Физматлит, 2002.
7. *Гафаров Е.Р., Долгий А., Лазарев А.А., Вернер Ф.* Новый эффективный алгоритм решения задачи об инвестициях // АиТ. 2016. № 9. С. 150–166.
Gafarov E.R., Dolgui A., Lazarev A.A., et al. A New Effective Dynamic Program for an Investment Optimization Problem // Autom. Remote Control. 2016. V. 77. No. 9. P. 1633–1648. <https://doi.org/10.1134/S0005117916090101>
8. *Лазарев А.А.* Графический подход к решению задач комбинаторной оптимизации // АиТ. 2007. № 4. С. 13–23.
9. *Лазарев А.А.* Теория расписаний. Методы и алгоритмы. М.: ИПУ РАН, 2019.
10. *Brethauer K.M., Shetty B.* The Nonlinear Knapsack Problem – Algorithms and applications // Eur. J. Oper. Res. 2002. V. 138. Iss. 3. P. 459–472.
11. *Riedhammer K., Gillick D., Favre B., Hakkani-Tür D.* Packing the Meeting Summarization Knapsack // Proc. Interspeech. Brisbane, Australia, 2008.
12. *Robson J.M.* Finding a Maximum Independent set in Time $O(2n/4)$ // Technical Report 1251-01, LaBRI, Université de Bordeaux I, 2001.
13. *Яблонский С.В.* Введение в дискретную математику. М.: Наука, 1986.

14. *Korneenko V.P., Nazyuta S.V., Chursin A.A.* System for Uncertainty Factors Accounting When Optimizing and Choosing Effective Options for Network Work Schedules on a Dynamic Model with Dead-End Controls / IOP Conference Series: Earth and Environmental Science // Proc. Int. Science and Technology Conf. on Earth Science. Vladivostok, Russian: IOP Publishing Ltd, 2021. Sci. 666 062129. <https://doi.org/10.1088/1755-1315/666/6/062129>.
15. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 20.01.2020

После доработки 17.03.2021

Принята к публикации 30.06.2021

© 2021 г. А.А. ЛАЗАРЕВ, д-р. физ.-мат. наук (jobmath@mail.ru),
Д.В. ЛЕМТЮЖНИКОВА, канд. физ.-мат. наук (darabbt@gmail.com)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва),
А.А. ТЮНЯТКИН (andtun@yandex.ru)
(Московский государственный университет им. М.В. Ломоносова)

МЕТРИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ ДЛЯ ЗАДАЧИ МИНИМИЗАЦИИ МАКСИМАЛЬНОГО ВРЕМЕННОГО СМЕЩЕНИЯ ДЛЯ ОДНОГО ПРИБОРА¹

Статья основана на использовании методов непрерывной математики в дискретных задачах. Рассматриваются три новых подхода к решению задач теории расписаний: метрический, интерполяционный и комбинированный — метрическая интерполяция. Метрическая интерполяция является объединением двух других и совмещает в себе их преимущества. Каждый из этих подходов позволяет сокращать время решения соответствующих задач и вычислять значения гарантированной абсолютной погрешности целевой функции.

Ключевые слова: теория расписаний, метрический подход, интерполяция, аппроксимация, дискретная оптимизация.

DOI: 10.31857/S0005231021100081

1. Введение

Подавляющее большинство задач теории расписаний NP-трудны [1]. Для решения таких задач часто используются быстрые алгоритмы, производительность которых сильно зависит от входных данных. Предлагается новый подход нахождения приближенных решений задач теории расписаний — метрическая интерполяция. Данный метод — это комбинация двух различных подходов, интерполяционного и метрического [1, 2], благодаря совместному использованию которых возможно:

- 1) использовать приближенное значение целевой функции как начальное решение для итерационных методов;
- 2) аппроксимировать решение задачи теории расписаний, используя некоторый класс полиномиально разрешимых примеров (при его существовании);
- 3) аппроксимировать значение целевой функции задачи даже при отсутствии полиномиально разрешимых примеров, используя интерполяционный подход.

Следует сразу отметить, что метрическая интерполяция — подход, который используется для ускорения работы уже существующих алгоритмов решения задач теории расписаний. В связке с рассматриваемым подходом могут

¹ Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 20-58-S52006).

быть использованы алгоритмы решения задач теории расписаний, рассмотренные, например, в [1, 3, 4]. Для работы со случайными данными могут быть использованы алгоритмы и методы из [5], а метрическая интерполяция ускоряет их выполнение при обработке трудных случаев.

Поскольку интерполяционный подход работает лишь со значениями целевой функции, с его помощью можно составлять и расписания многостадийных систем, решая задачи, например, алгоритмами из [6]. Метрический подход можно использовать для этой цели при небольшой модификации, рассмотренной в [2].

Для определенности, в данной статье рассматривается решение задачи минимизации максимального временного смещения $1|r_j|L_{\max}$ с помощью двойственного алгоритма [7] и трех подходов, которые упомянуты выше.

2. Задача минимизации максимального временного смещения для одного прибора

2.1. Постановка задачи минимизации максимального временного смещения

В задаче $1|r_j|L_{\max}$, которую будем рассматривать, задано множество из n требований $N = \{1, \dots, n\}$. Для каждого требования j заданы: время поступления r_j , длительность выполнения p_j и директивный срок d_j [1]. Под *расписанием* π будем подразумевать некоторую перестановку требований множества N . Введем время завершения требования j при расписании π :

$$(1) \quad C_j(\pi) = \max_{\pi} \left\{ r_j, \max_{(k \rightarrow j)_{\pi}} C_k(\pi) \right\} + p_j.$$

Здесь $(k \rightarrow j)_{\pi}$ — требования, которые выполняются перед работой j при расписании π .

Таким образом, задача минимизации максимального временного смещения состоит в нахождении такого расписания π_0 , при котором целевая функция принимает минимальное значение:

$$(2) \quad L_{\max}(\pi_0) = \min_{\pi} \max_{j=1, \dots, n} \{C_j(\pi) - d_j\}.$$

Данная задача является NP-трудной в сильном смысле [8].

2.2. Пространство примеров размерности n

Рассмотрим пример A , который состоит из n заявок. Такой пример можно представить точкой в $3n$ -мерном пространстве с координатами $A(r_1, r_2, \dots, r_n, p_1, p_2, \dots, p_n, d_1, d_2, \dots, d_n)$. Для наглядности будем записывать эти координаты в виде матрицы $3 \times n$:

$$\begin{pmatrix} r_1 & r_2 & \dots & r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix}.$$

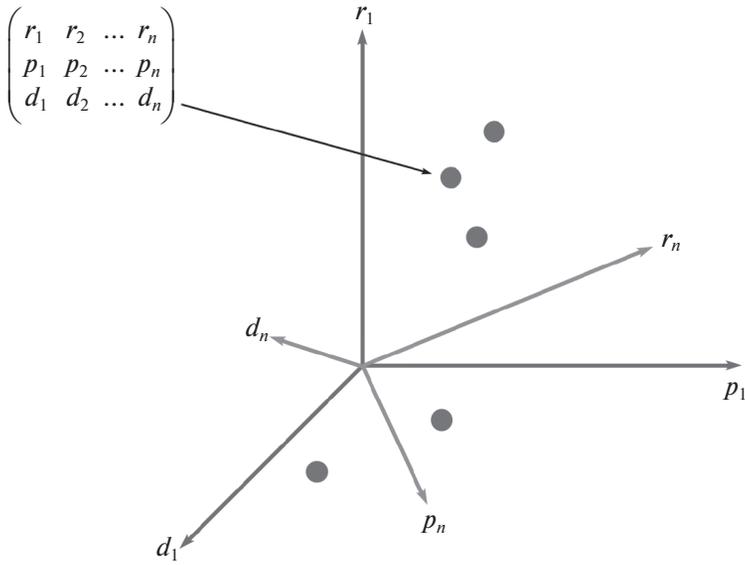


Рис. 1. $3n$ -мерное пространство примеров размерности n .

Далее в работе будем рассматривать каждый пример как точку в $3n$ -мерном пространстве примеров, состоящих из n требований (см. рис. 1). А само это пространство будем называть **пространством примеров размерности n** .

Более подробно о пространстве примеров размерности n см. [9].

3. Метрический подход

3.1. Идея подхода

В этом разделе представлены основы метрического подхода, который был предложен и описан более подробно в [1, 2].

Во введенном пространстве примеров размерности n зададим метрику.

Определение 1. Метрикой для произвольных примеров A и B называется функция $\rho(A, B)$, удовлетворяющая свойствам:

- (3) $\rho(A, B) \geq 0$;
- (4) $\rho(A, B) = 0 \Leftrightarrow A = B$;
- (5) $\rho(A, B) = \rho(B, A)$;
- (6) $\rho(A, C) \leq \rho(A, B) + \rho(B, C)$ для всех A, B, C ;
- (7) $\rho(A, B) = \rho_d(A, B) + \rho_r(A, B) + \rho_p(A, B)$;
- (8) $\rho_d(A, B) = \max_{j \in N} \{d_j^A - d_j^B\} - \min_{j \in N} \{d_j^A - d_j^B\}$;
- (9) $\rho_r(A, B) = \max_{j \in N} \{r_j^A - r_j^B\} - \min_{j \in N} \{r_j^A - r_j^B\}$;
- (10) $\rho_p(A, B) = \sum_{j \in N} (|p_j^A - p_j^B|)$.

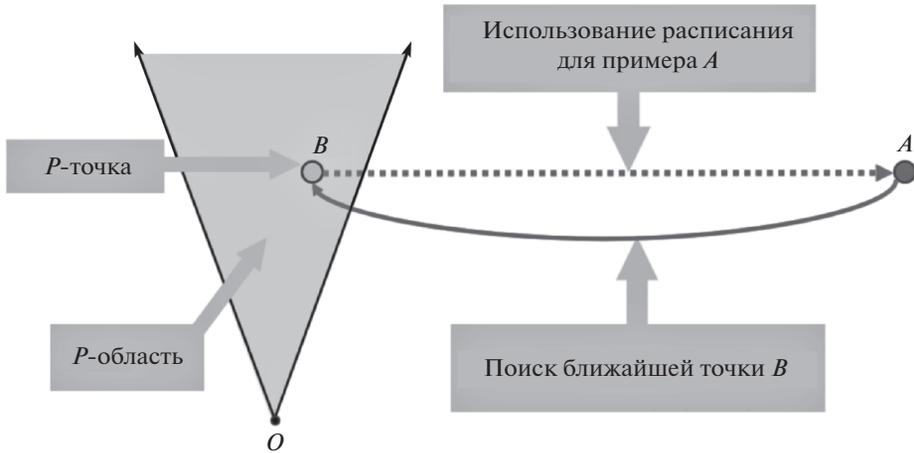


Рис. 2. Метрический подход.

Можно доказать, что если решить задачу $1|r_j|L_{\max}$ для примера B и получить оптимальное расписание π^B , можно оценить погрешность при использовании расписания π^B в качестве решения той же задачи для исходного примера A .

Теорема 1. Пусть $\tilde{\pi}^B$ – приближенное решение примера B , удовлетворяющего $L_{\max}^B(\tilde{\pi}^B) - L_{\max}^B(\pi^B) \leq \delta_B$. Тогда

$$(11) \quad 0 \leq L_{\max}^A(\tilde{\pi}^B) - L_{\max}^A(\pi^A) \leq \rho(A, B) + \delta_B.$$

Доказательство теоремы 1 приведено в [1, 2].

Согласно этой теореме, получается, что при принадлежности примера B некоторой полиномиально разрешимой области можно поступить следующим образом: решить пример B (за полиномиальное время), использовать полученное расписание для примера A и оценить погрешность, вычислив значение метрики $\rho(A, B)$.

Качественно опишем алгоритм решения задачи $1|r_j|L_{\max}$ при помощи метрического подхода. Будем обозначать через $L_{\max}^A(\pi)$ значения целевой функции для примера A при расписании π .

Алгоритм 1 (метрический подход).

- 1) Найти пример B из полиномиально разрешимой области, наиболее близкий к A по метрике $\rho(A, B)$;
- 2) найти оптимальное расписание π^B для примера B ;
- 3) применить расписание π^B к исходному примеру A ;
- 4) оценить погрешность согласно теореме 1: $0 \leq L_{\max}^A(\pi^B) - L_{\max}^A(\pi^A) \leq \rho(A, B)$.

На рис. 2 проиллюстрирована работа алгоритма 1.

Остается лишь поставить задачу нахождения примера B , наиболее близкого к исходному примеру A по метрике $\rho(A, B)$.

3.2. Поиск ближайшего полиномиально разрешимого примера

Рассмотрим множество точек, которые соответствуют полиномиально разрешимым примерам задачи. В пространстве примеров это множество задается следующей системой неравенств:

$$(12) \quad XR + YP + ZD \leq H,$$

где $R = (r_1, \dots, r_n)^T$, $P = (p_1, \dots, p_n)^T$ ($p_j \geq 0$ для всех $j \in N$), $D = (d_1, \dots, d_n)^T$ и X, Y, Z — некоторые матрицы размерности $k \times n$; а $H = (h_1, \dots, h_k)^T$ является k -мерным вектором, где верхний индекс T обозначает транспонирование.

Решив эту систему, сможем построить многомерный конус в пространстве примеров, все точки которого полиномиально разрешимы [9]. Например, в случае значений $Y = 0$, $Z = 0$ и произвольной матрицы X решением системы будет множество примеров Джексона [10].

Тогда в классе примеров (12) определим пример B с минимальным значением метрики $\rho(A, B)$ относительно исходного примера A , решив следующую задачу линейного программирования [9]:

$$(13) \quad \begin{cases} (x^d - y^d + x^r - y^r) + \sum_{j \in N} x_j^p \rightarrow \min, \\ y^d \leq d_j^A - d_j^B \leq x^d \quad \forall j \in N, \\ y^r \leq r_j^A - r_j^B \leq x^r \quad \forall j \in N, \\ -x_j^p \leq p_j^A - p_j^B \leq x_j^p \quad \forall j \in N, \\ 0 \leq x_j^p \quad \forall j \in N, \\ XR^B + YP^B + ZD^B \leq H. \end{cases}$$

После решения этой подзадачи остается лишь найти расписание для исходного примера, пользуясь алгоритмом 1.

Более подробно про метрический подход, классы полиномиально разрешимых примеров и задачи линейного программирования для этих классов см. [1, 2, 9].

3.3. Оценка сложности алгоритма

Сложность решения задачи с помощью метрического подхода напрямую зависит от трудоемкости алгоритма, используемого для решения примера из полиномиальной области. Для класса полиномиально разрешимых примеров Джексона [10], например, сложность алгоритма будет порядка $O(n \log n)$ без учета трудоемкости решения задачи (13).

4. Интерполяционный подход

4.1. Преобразования в пространстве примеров размерности n

Определение 2. Преобразованием $r'_j = \alpha r_j$ называется такое преобразование, которое ставит в соответствие примеру $\begin{pmatrix} r_1 & r_2 & \dots & r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix}$ пример $\begin{pmatrix} \alpha r_1 & \alpha r_2 & \dots & \alpha r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix}$, где α – произвольное неотрицательное число: $\alpha \in \mathbb{R}$, $\alpha \geq 0$.

Аналогично определяются преобразования $p'_j = \alpha p_j$ и $d'_j = \alpha d_j$.

Теорема 2. Все точки, полученные в результате какого-либо из преобразований $r'_j = \alpha r_j$, $p'_j = \alpha p_j$ или $d'_j = \alpha d_j$, лежат на одной прямой в пространстве примеров размерности n .

Из теоремы 2 следует, что любые точки, полученные в результате какого-либо из предложенных преобразований исходного примера, лежат на одной прямой в этом $3n$ -мерном пространстве, причем эта прямая содержит и исходный пример (при $\alpha = 1$). Иллюстрация к этому утверждению приведена на рис. 3. Доказательство теоремы 2 приведено в Приложении 1.

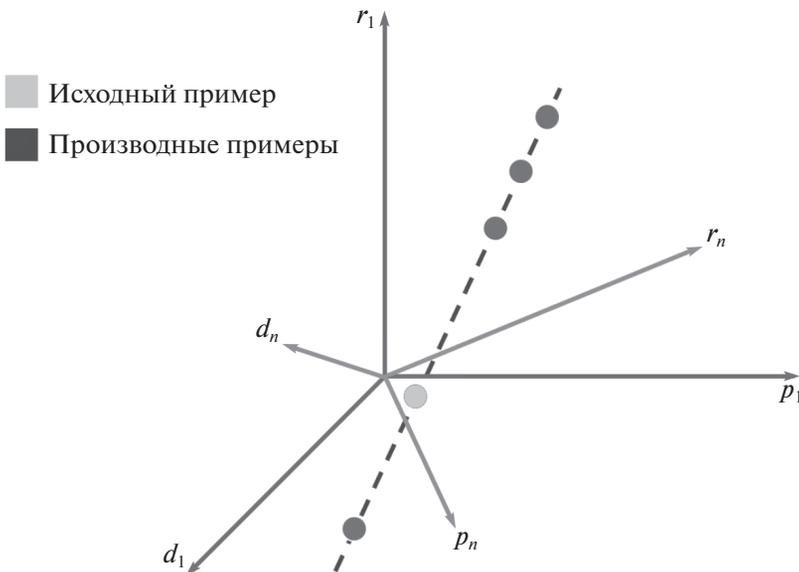


Рис. 3. Иллюстрация к теореме 2.

4.2. Идея подхода

Рассмотрим алгоритм решения задачи $1|r_j|L_{\max}$ при помощи интерполяционного подхода. Для определенности будем использовать преобразование $r'_j = \alpha r_j$, а также интерполяцию Лагранжа [11].

Интерполяционный полином Лагранжа задается следующей формулой:

$$(14) \quad L_m(x) = \sum_{k=0}^m \frac{\prod_{i \neq k} (x - x_i)}{\prod_{j \neq k} (x - x_j)} f(x_k),$$

где x_k , $k = 1, \dots, m$ – узлы интерполяции, значения целевой функции в которых соответственно равны $f(x_k)$, причем $x \neq x_j$.

Кроме того, в [11] показано, как оценить погрешность интерполяции функции по m узлам с учетом, что $x \neq x_1 \neq \dots \neq x_m$:

$$(15) \quad \begin{aligned} R_m(x) &= f(x) - L_m(x) = \\ &= \left(\prod_{i=1, \dots, m} (x - x_i) \right) \left(\frac{f(x)}{(x - x_1)(x - x_2) \dots (x - x_m)} + \right. \\ &\quad \left. + \frac{f(x_1)}{(x_1 - x)(x_1 - x_2) \dots (x_1 - x_m)} + \dots \right. \\ &\quad \left. \dots + \frac{f(x_m)}{(x_m - x)(x_m - x_1) \dots (x_m - x_{m-1})} \right). \end{aligned}$$

Таким образом, качественный вид алгоритма интерполяционного подхода будет следующим:

Алгоритм 2 (интерполяционный подход).

- 1) Определить множество m численных значений α_i ;
- 2) для каждого α_i получить i -й производный пример A_i путем преобразования $r'_j = \alpha r_j$ (см. определение 2) исходного примера с коэффициентом $\alpha = \alpha_i$;
- 3) для каждого i -го производного примера A_i решить соответствующую задачу и вычислить значение целевой функции;
- 4) зная значения целевых функций производных примеров, оценить значение целевой функции исходного примера, используя (14);
- 5) оценить погрешность аппроксимации целевой функции по формуле (15), т.е. задать в этой формуле $x_i = \alpha_i$ и $x = 1$.

Для наглядного объяснения принципа работы алгоритма приведен график на рис. 4. По оси x отложены значения α_i , по оси y – значения целевой функции. Темные точки – производные примеры – используются как узлы интерполяции, по ним и строится интерполяционная кривая. Затем для $\alpha = 1$, т.е. в точке, соответствующей исходному примеру, вычисляется значение интерполирующего полинома $L_m(x)$, которое и является аппроксимацией значения целевой функции исходного примера.

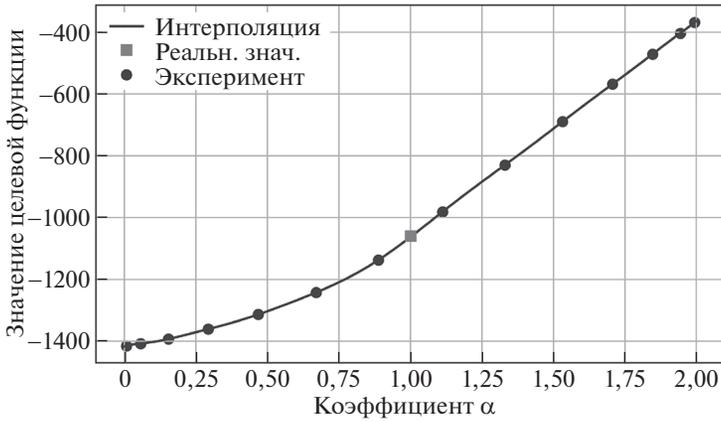


Рис. 4. Иллюстрация к алгоритму 2.

4.3. Модификации алгоритма

Акцентируем внимание на том, что преобразование $r'_j = \alpha r_j$, т.е. домножение всех параметров r исходного примера на некоторый коэффициент α , является лишь одним из возможных преобразований и приводится здесь в качестве примера.

Полученные в результате какого-либо из преобразований точки можно решить некоторым полиномиальным в среднем алгоритмом (например, двойственным [7]), который долго работает над решением исходного примера, но, как показали эксперименты, решит производные примеры за полиномиальное в среднем время.

Стоит также подчеркнуть, что представленный алгоритм отличается гибкостью: можно использовать любые алгоритмы решения производных примеров, а также различные методы интерполяции.

Например, по принципу оптимального выбора узлов интерполирования [11] рекомендуется выбирать эти узлы с помощью многочленов Чебышева.

При большом количестве узлов интерполирования рекомендуется использовать кусочно-полиномиальную интерполяцию ввиду неравномерной сходимости интерполяционных полиномов Лагранжа [11].

В численных экспериментах, параметры которых будут указаны в табл. 1, а результаты — в табл. 2, 3, для решения примеров использовался двойственный алгоритм [7], а для интерполяции значений целевой функции — методы Лагранжа и Чебышева.

4.4. Оценка сложности алгоритма

Время выполнения алгоритма будет наиболее низким при следующей модификации: необходимо задать некоторый лимит времени вычислений τ и при итерации через множество значений α_i , как только время вычисления примера A_i превышает τ , сразу переходить к примеру A_{i+1} . Благодаря такой модификации, время работы алгоритма гарантированно не будет превышать значения $m\tau$, где m — количество всех производных примеров.

Таким образом, при использовании алгоритма решения задач с помощью интерполяционного подхода возможно заранее оценить максимальное время работы алгоритма.

5. Метод метрической интерполяции

Метрическая интерполяция — комбинированный метод, совмещающий в себе оба предыдущих. Введем его для общего случая, не опираясь на конкретные задачи и алгоритмы их решения.

Алгоритм 3 (метод метрической интерполяции).

- 1) На вход подается исходный пример A , принадлежащий классу примеров некоторой задачи теории расписаний;
- 2) при наличии полиномиально разрешимого класса примеров рассматриваемой задачи с помощью алгоритма 1 находится некоторый пример B с минимальным расстоянием до исходного примера A по метрике $\rho(A, B)$;
- 3) по теореме 1 оценивается погрешность метрического подхода;
- 4) с помощью некоторого заранее выбранного преобразования заполняется множество производных примеров A_i ;
- 5) проводится интерполяция значений целевой функции для производных примеров согласно алгоритму 2;
- 6) для производного решенного примера A_i с наиболее близким к исходному примеру A значением целевой функции составляется оптимальное расписание π^{A_i} ;
- 7) по теореме 1 оценивается расстояние по метрике $\rho(A, A_i)$ между примерами, а значит, и погрешность аппроксимации;
- 8) получив окрестность, содержащую точное решение, используем найденную точку этой окрестности в качестве начального значения для других точных алгоритмов.

Схема алгоритма 3 приведена на рис. 5.

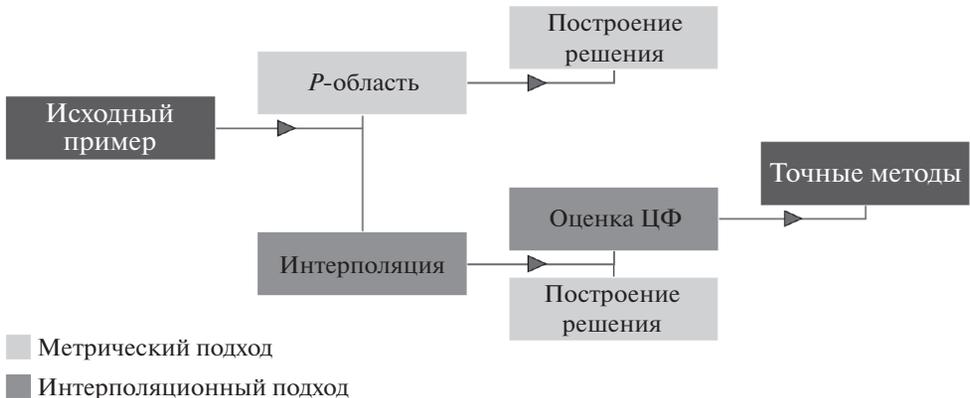


Рис. 5. Схема алгоритма 3.

6. Пример решения задачи

1) Дан пример $A = \begin{pmatrix} r_1 & r_2 & r_3 \\ p_1 & p_2 & p_3 \\ d_1 & d_2 & d_3 \end{pmatrix}$.

2) Воспользуемся для определенности классом L полиномиально разрешимых примеров Лазарева [1, 2, 9]. Решив задачу (13), найдем пример $B \in L$, принадлежащий классу L , с минимальной метрикой $\rho(A, B)$.

3) Решим пример B двойственным алгоритмом [7]. Из теоремы 1 следует оценка погрешности этого решения: она не превосходит $\rho(A, B)$.

4) Рассмотрим два производных примера: $A_0 = \begin{pmatrix} 0 \cdot r_1 & 0 \cdot r_2 & 0 \cdot r_3 \\ p_1 & p_2 & p_3 \\ d_1 & d_2 & d_3 \end{pmatrix}$, $A_\xi = \begin{pmatrix} \xi \cdot r_1 & \xi \cdot r_2 & \xi \cdot r_3 \\ p_1 & p_2 & p_3 \\ d_1 & d_2 & d_3 \end{pmatrix}$, $\xi \gg \max(r_1 + p_1, r_2 + p_2, r_3 + p_3)$. Пример A_0 — поли-

номиально разрешимый случай Джексона [10], решаемый очевидным перебором: подставляя на первое место расписания любую из работ, все остальные работы упорядочиваем в порядке возрастания директивных сроков. Пример A_ξ — случай, в котором момент поступления следующей работы превосходит продолжительность предыдущей. Оптимальное расписание составляется сортировкой требований по возрастанию времен поступления r_j .

5) Вычислим оптимальные значения целевой функции $L_{\max}(\pi_0^*)$ и $L_{\max}(\pi_\xi^*)$ для примеров A_0 и A_ξ соответственно. Используя (14) и (15), получим оценку целевой функции исходного примера $L_{\max}(\pi^*)$ и погрешность этой оценки.

7. Результаты экспериментов

7.1. Метрический подход

В численных экспериментах с метрическим подходом подсчитывались три параметра:

- 1) μ — процент примеров, для которых алгоритм нашел оптимальное решение;
- 2) β_{av} — средняя относительная погрешность значения целевой функции;
- 3) β_{\max} — максимальная относительная погрешность значения целевой функции.

Погрешность определяется относительно оптимального значения целевой функции тестового примера. Перечисленные параметры рассчитываются по следующим формулам:

$$(16) \quad \beta_{av} = \sum_{i=1}^{\bar{K}} \frac{L_{\max}(\pi_i) - L_{\max}(\pi^*)}{L_{\max}(\pi^*)} \cdot 100\%;$$

$$(17) \quad \beta_{\max} = \max_{i=1, \dots, \bar{K}} \left\{ \frac{L_{\max}(\pi_i) - L_{\max}(\pi^*)}{L_{\max}(\pi^*)} \cdot 100\% \right\};$$

$$(18) \quad \mu = \frac{K^* \cdot 100\%}{K}.$$

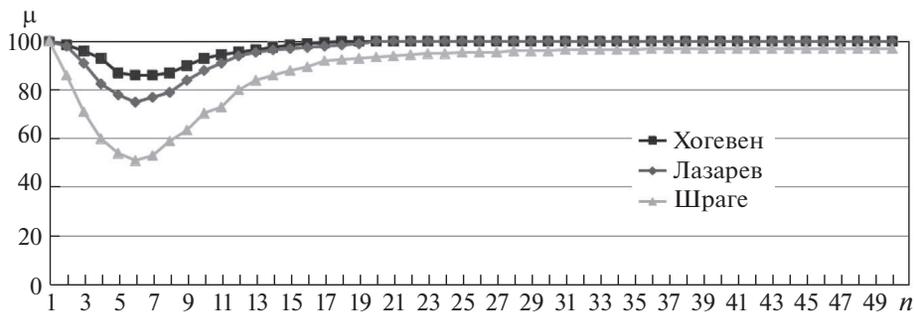


Рис. 6. График зависимости значения μ от количества требований в примере.

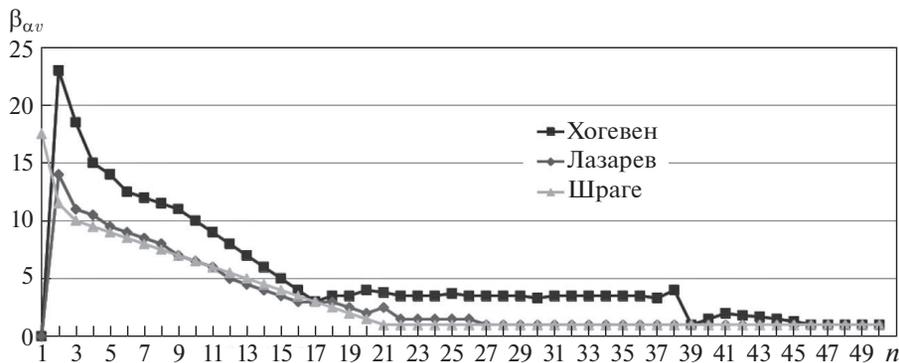


Рис. 7. График зависимости значения $\beta_{\alpha v}$ от количества требований в примере.

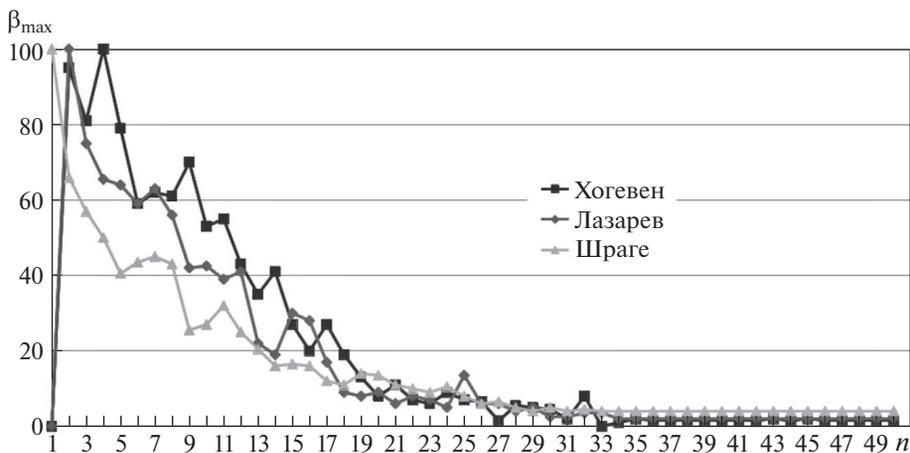


Рис. 8. График зависимости значения β_{\max} от количества требований в примере.

Здесь K – количество сгенерированных примеров, \bar{K} – количество примеров, для которых решение, полученное алгоритмом, не было оптимальным ($\bar{K} = K - K^*$), π_i и π_i^* – найденное алгоритмом расписание и оптимальное расписание для i -го сгенерированного примера, решенного неоптимально, а K^* – количество тестовых примеров, оптимально решаемых алгоритмом.

Результаты экспериментов приведены на графиках рис. 6–8.

Три класса примеров (Хогевена, Лазарева и Шраге), указанные на графиках, – это разновидности областей полиномиально разрешимых примеров, на которые проецировался исходный пример в ходе проведения эксперимента. Выбор множества полиномиально разрешимых примеров рассмотрен подробнее в [9].

Подробнее о метрическом подходе, полиномиально разрешимых классах примеров, методике проведения экспериментов и их результатах см. [2].

7.2. Интерполяционный подход

На численных экспериментах с интерполяционным подходом стоит остановиться подробнее, поскольку их результаты представляются впервые. В табл. 1 приведена основная информация о методике проведения этих экспериментов.

Сто простых примеров, каждый — из 10 требований, были сгенерированы. Критерий простоты — время решения каждого такого примера двойственным алгоритмом [7] не должно превышать 1 секунду. Это множество примеров использовалось во всех последующих экспериментах, чтобы иметь возможность сравнивать результаты. Простота этих примеров — способ ускорить вычисления.

На приведенных графиках рис. 9, 10 представлены экспериментальные результаты для одного из этих примеров с использованием двух методов интерполяции: Лагранжа и Чебышева [11, 15]. Коэффициент α изменялся в диапазоне $\alpha \in [0; 2]$ с шагом 0,1. Таким образом, для интерполяции использовалось 20 точек, а затем 21-я, соответствующая исходному примеру, также отображалась на графике для сравнения аппроксимации с действительностью.

Обозначения следующие: значение целевой функции исходного примера показано серым квадратом, узлы и кривая интерполяции отображены темно-серым. Визуально оценить точность метода можно, сравнивая положение серого квадрата (реальное значение целевой функции) относительно темно-серой кривой (аппроксимированное значение).

Таблица 1. Основная информация о методике проведения экспериментов с интерполяционным подходом

Операционная система	Windows 10
ЦП	Intel core i3
Оперативная память	4 Гб
Язык программирования	Python 3.6 [12]
Среда программирования	Jupyter Notebook [12]
Основная библиотека для вычислений	numpy [13]
Библиотека для построения графиков	matplotlib/pyplot [14]
Способ генерации r_j	$r_i = r_{i-1} + X_i; X_i \sim \exp(\lambda); \lambda = 0,01$
Способ генерации p_j	$p_j \sim N(\mu, \sigma); \mu = 100, \sigma = 40$
Способ генерации d_j	$d_j = r_j + E[p_j]$

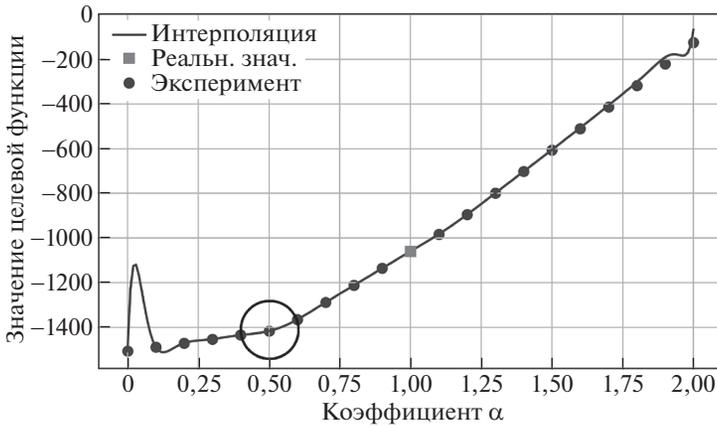


Рис. 9. Результаты эксперимента для преобразования $r'_j = \alpha r_j$ с использованием интерполяции Лагранжа.

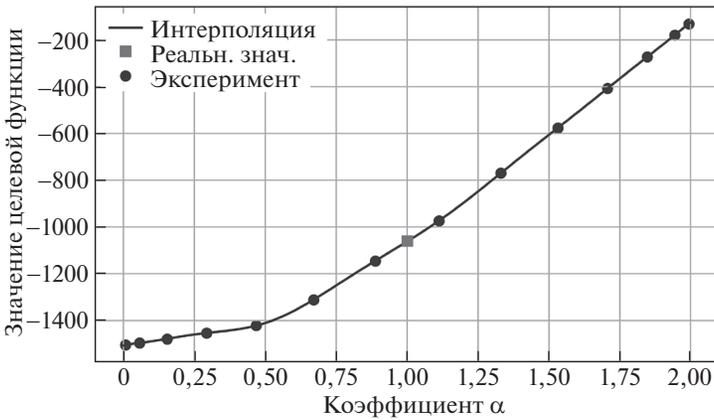


Рис. 10. Результаты эксперимента для преобразования $r'_j = \alpha r_j$ с использованием интерполяции Чебышева.

На графике рис. 9 рассмотрим точку перегиба, соответствующую значению $\alpha = 0,5$ – она помещена в круг. Тем не менее оптимальные расписания в точках $\alpha = 0,4, 0,5, 0,6$ одинаковы. Таким образом, наличие точки перегиба *не означает* изменение оптимального порядка расстановки требований.

При использовании в рассматриваемой задаче метода Лагранжа характерны всплески по краям отрезка интерполяции. Возможная причина этого – большое количество узлов интерполяции [11], однако это явление требует дальнейшего исследования.

При использовании интерполяции Чебышева такие всплески не характерны (см. график рис. 10). Несмотря на это, у метода Чебышева есть свой недостаток – невозможно выбирать узлы интерполяции произвольным образом. Узлы интерполяции строго выбираются при помощи многочленов Чебышева [15].

Таблица 2. Таблица результатов экспериментов для интерполяции Лагранжа

Величина	Значение	Формула
Максимальный модуль ошибки, %	287,66	$\max_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Минимальный модуль ошибки, %	0,00	$\min_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Средний модуль ошибки, %	5,48	$\frac{1}{n} \sum_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Среднее значение ошибки, %	3,73	$\frac{1}{n} \left \sum_{i \in N} \left(\frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right) \right \cdot 100\%$

Таблица 3. Таблица результатов экспериментов для интерполяции Чебышева

Величина	Значение	Формула
Максимальный модуль ошибки, %	137,32	$\max_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Минимальный модуль ошибки, %	0,00	$\min_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Средний модуль ошибки, %	1,93	$\frac{1}{n} \sum_{i \in N} \left \frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right \cdot 100\%$
Среднее значение ошибки, %	0,28	$\frac{1}{n} \left \sum_{i \in N} \left(\frac{L_{\max}(i) - L_{\max}^T(i)}{L_{\max}^T(i)} \right) \right \cdot 100\%$

В табл. 2 и 3 представлены результаты численных экспериментов со всеми сгенерированными примерами при использовании преобразования $r'_j = \alpha r_j$ и обоих методов интерполяции.

Обозначения следующие: $L_{\max}^T(i)$ – реальное значение целевой функции i -го примера, $L_{\max}(i)$ – аппроксимированное значение целевой функции i -го примера; N – множество n различных примеров.

8. Заключение

В статье рассмотрен метрический подход [2] к решению задач теории расписаний. Кроме того, представлен новый — интерполяционный — метод, приведены результаты численных экспериментов. Также представлена идея комбинированного подхода — метрической интерполяции — как способа объединения двух предыдущих методов.

Перечисленные подходы отличаются универсальностью и гибкостью, а экспериментальные результаты подтверждают также и их высокую точность, позволяющую использовать рассмотренные подходы для решения широкого круга задач теории расписаний.

Доказательство теоремы 2. Задача ставится следующим образом: пусть выбраны три произвольных неотрицательных числа α^A , α^B , α^C и задан пример $\begin{pmatrix} r_1 & r_2 & \dots & r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix}$. Доказать, что в рассмотренном выше $3n$ -мерном пространстве примеры

$$\begin{pmatrix} \alpha^A r_1 & \alpha^A r_2 & \dots & \alpha^A r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (A),$$

$$\begin{pmatrix} \alpha^B r_1 & \alpha^B r_2 & \dots & \alpha^B r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (B),$$

$$\begin{pmatrix} \alpha^C r_1 & \alpha^C r_2 & \dots & \alpha^C r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (C)$$

лежат на одной прямой. В силу произвольности выбора значений $\alpha^A \dots \alpha^C$ это утверждение будет означать, что исходный пример лежит на той же прямой, что и примеры (A)–(C).

Таким образом, решение задачи докажет, что любые точки, полученные в результате преобразования $r'_j = \alpha r_j$ исходного примера, лежат на одной прямой в $3n$ -мерном пространстве, причем эта прямая содержит и исходный пример.

Рассмотрим некоторое $3n$ -мерное пространство и три точки на нем: $A(x_1^A \dots x_{3n}^A)$, $B(x_1^B \dots x_{3n}^B)$, $C(x_1^C \dots x_{3n}^C)$. Каноническое уравнение прямой (AB) имеет вид [16]:

$$(П.1) \quad \frac{x_1 - x_1^A}{x_1^B - x_1^A} = \frac{x_2 - x_2^A}{x_2^B - x_2^A} = \dots = \frac{x_{3n} - x_{3n}^A}{x_{3n}^B - x_{3n}^A}.$$

Тогда критерий того, что точка С лежит на прямой (AB):

$$(П.2) \quad \frac{x_1^C - x_1^A}{x_1^B - x_1^A} = \frac{x_2^C - x_2^A}{x_2^B - x_2^A} = \dots = \frac{x_{3n}^C - x_{3n}^A}{x_{3n}^B - x_{3n}^A}.$$

В рассматриваемой задаче анализируется расположение трех точек:

$$\begin{pmatrix} \alpha^A r_1 & \alpha^A r_2 & \dots & \alpha^A r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (A),$$

$$\begin{pmatrix} \alpha^B r_1 & \alpha^B r_2 & \dots & \alpha^B r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (B),$$

$$\begin{pmatrix} \alpha^C r_1 & \alpha^C r_2 & \dots & \alpha^C r_n \\ p_1 & p_2 & \dots & p_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} (C).$$

Заметим, что координаты $p_1, p_2, \dots, p_n, d_1, d_2, \dots, d_n$ у всех трех точек одинаковы. Тогда критерий (П.2) в приложении к рассматриваемой задаче примет вид

$$(П.3) \quad \frac{\alpha^C r_1 - \alpha^A r_1}{\alpha^B r_1 - \alpha^A r_1} = \frac{\alpha^C r_2 - \alpha^A r_2}{\alpha^B r_2 - \alpha^A r_2} = \dots = \frac{\alpha^C r_n - \alpha^A r_n}{\alpha^B r_n - \alpha^A r_n}.$$

Преобразуем его:

$$(П.4) \quad \frac{r_1(\alpha^C - \alpha^A)}{r_1(\alpha^B - \alpha^A)} = \frac{r_2(\alpha^C - \alpha^A)}{r_2(\alpha^B - \alpha^A)} = \dots = \frac{r_n(\alpha^C - \alpha^A)}{r_n(\alpha^B - \alpha^A)}.$$

И после сокращения получим тождество

$$(П.5) \quad \frac{\alpha^C - \alpha^A}{\alpha^B - \alpha^A} = \frac{\alpha^C - \alpha^A}{\alpha^B - \alpha^A}.$$

Таким образом, критерий (П.2) выполняется при любых $\alpha^A, \alpha^B, \alpha^C$. То есть любые три точки, полученные в результате преобразования $r'_j = \alpha r_j$ исходного примера, лежат на одной прямой в $3n$ -мерном пространстве.

Преобразования $p'_j = \beta p_j$ и $d'_j = \gamma d_j$ вводятся и доказываются аналогично. Так же как и в случае с $r'_j = \alpha r_j$, полученные в результате этих преобразований примеры лежат на одной прямой в $3n$ -мерном пространстве, содержащей и исходный пример.

Теорема 2 доказана.

СПИСОК ЛИТЕРАТУРЫ

1. Лазарев А.А. Теория расписаний. Методы и алгоритмы. М.: ИПУ РАН, 2019.
2. Lazarev A.A., Lemtyuzhnikova D.V., Werner F. A metric approach for scheduling problems with minimizing the maximum penalty // Appl. Math. Modell. 2021. No. 89. P. 1163–1176.
3. Tanaev V.S., Gordon V.S., Shfransky Y.M. Scheduling theory: single-stage systems Dordrecht, The Netherlands: Kluwer Academic Publishers, 1994.
4. Brucker P. Scheduling algorithms. Berlin: Springer, 1995.
5. Sotskov Y.N., Werner F. (Editors). Sequencing and Scheduling with Inaccurate Data. New York, USA: Nova Science Publishers, Inc., 2014.
6. Tanaev V.S., Sotskov Y.N., Strusevich V.A. Scheduling theory: multi-stage systems. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1994.
7. Lazarev A.A., Pravdivets N., Werner F. On the Dual and Inverse Problems of Scheduling Jobs to Minimize the Maximum Penalty // Mathematics. 2020. V. 8. No. 7. P. 1131.

8. *Lenstra J.K., Rinnooy Kan A.H.G., Brucker P.* Complexity of machine scheduling problems // *Annals of discrete mathematics*. Elsevier, 1977. V. 1. P. 343–362.
9. *Lazarev A.A., Lemtyuzhnikova D.V., Pravdivets N.A., Werner F.* Polynomially Solvable Subcases for the Approximate Solution of Multi-machine Scheduling Problems // *Communications in Computer and Information Science (Advances in Optimization and Applications, 11th International Conference, OPTIMA 2020)*. 2021. No. 1340. P. 211–223.
10. *Jackson J.R.* Scheduling a production line to minimize maximum tardiness. Los Angeles, CA: University of California, 1955.
11. *Самарский А.А., Гулин А.В.* Численные методы: Учеб. пособие для вузов. М.: Наука. Гл. ред. физ-мат. лит., 1989.
12. *Kluyver T., Ragan-Kelley B., Perez F., et.all.* Jupyter Notebooks — a publishing format for reproducible computational workflows. Netherlands: InELPUB, 2016. P. 87–90.
13. *Harris C.R., Millman K.J., van der Walt S.J., et al.* Array programming with NumPy // *Nature*. 2020. V. 585. No. 7825. P. 357–362.
14. *Hunter J.D.* Matplotlib: A 2D Graphics Environment // *Computing in Science & Engineering*. 2007. V. 9. No. 3. P. 90–95.
15. *Rivlin T.J.* Chebyshev polynomials. Courier Dover Publications, 2020.
16. *Розенфельд Б.А.* Многомерные пространства. М.: Рипол Классик, 2013.

Статья представлена к публикации членом редколлегии А.А. Галяевым.

Поступила в редакцию 20.01.2021

После доработки 25.05.2021

Принята к публикации 30.06.2021

© 2021 г. А.С. МАНДЕЛЬ, д-р техн. наук (almandel@yandex.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва),
В.А. ЛАПТИН (straqker@bk.ru)
(Московский государственный университет им. М.В. Ломоносова)

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ СИСТЕМАМИ МАССОВОГО ОБСЛУЖИВАНИЯ С ПЕРЕКЛЮЧЕНИЕМ КАНАЛОВ ОБСЛУЖИВАНИЯ

Рассматривается задача оптимизации работы системы массового обслуживания, в которой число рабочих каналов обслуживания может управляемо изменяться в моменты контроля, отстоящие друг от друга на фиксированный временной шаг. Предполагается, что при переходе от шага к шагу интенсивность простейшего входящего потока изменяется в соответствии с некоторой однородной марковской цепью. Критерием выбора стратегии переключения каналов обслуживания является минимум суммарных средних затрат на многошаговом периоде планирования. Выявлена параметрическая структура оптимальной стратегии переключения каналов обслуживания.

Ключевые слова: управляемые системы массового обслуживания, марковский входящий поток, оптимизация, стратегии переключения каналов, параметризация стратегий.

DOI: 10.31857/S0005231021100093

1. Введение

Интерес к управляемым системам массового обслуживания (СМО) восходит к 70-м гг. XX в., когда была опубликована серия работ В.В. Рыкова. Отметим его фундаментальную публикацию [1], в которой приведена достаточно общая постановка задачи управления СМО и предложены подходы к ее решению. Оригинальным вкладом в теорию управляемых СМО стало исследование Ю.И. Неймарка [2], в котором изучались так называемые конфликтные системы, см. также развивающую эти результаты применительно к СМО публикацию [3].

К концу 80-х гг. интерес к управляемым системам массового обслуживания заметно вырос в связи с тем, что возник, начал и продолжает бурно развиваться такой класс прикладных систем как вычислительные и информационные сети и опирающиеся на них системы связи. Именно поэтому основные усилия специалистов в области теории массового обслуживания оказались направленными на решение задач управления, связанных именно с такими прикладными системами. Среди постановок задач и исследуемых моделей как у нас в стране, так и за ее рубежами рассматривались прежде всего

модели, в которых в качестве критериев выступали различные характеристики времен реакции на запрос [4], значения пропускных способностей [5, 6] и энергетической эффективности [7]. При этом решались вопросы оптимизации параметров [5, 7] и структуры этих систем [4, 6]. На эти темы имеется весьма значительное число публикаций, см., например, обзор [8].

В отличие от упомянутых работ в настоящей статье рассматривается задача оптимизации функционирования управляемой СМО по критерию минимума суммарных средних затрат за счет включения дополнительных или отключения лишних каналов обслуживания. Соответствующие модели весьма удобны для описания процессов, которые характерны для различных систем социально-экономической природы, включая производственно-торговые комплексы, системы продажи различного рода билетов, бронирования мест в гостиницах и многие другие. Настоящая статья развивает и обобщает постановки задач и методы их решения, которые были рассмотрены в предыдущих публикациях авторов настоящей статьи [9–14].

2. Постановка и алгоритм решения задачи

2.1. Содержательная постановка задачи

Рассматривается СМО с параллельно работающими каналами обслуживания, структурная схема которой представлена на рис. 1. При этом помимо работающих каналов обслуживания, которые на рис. 1 названы основными, имеются и резервные каналы обслуживания. В некоторые моменты времени (назовем их моментами контроля) любое доступное число резервных каналов может быть переведено в статус основных (включение) так же, как и любое число имеющихся основных каналов может быть переведено в статус резервных (выключение). Система рассматривается на конечном интервале времени — периоде планирования — продолжительности $T = N\tau$, где τ — длительность промежутка времени между соседними моментами контроля. Этот промежуток будет называться шагом. Таким образом, рассматриваемая СМО является системой с периодическим контролем, когда в равноудаленные моменты контроля можно включать дополнительные или отключать лишние

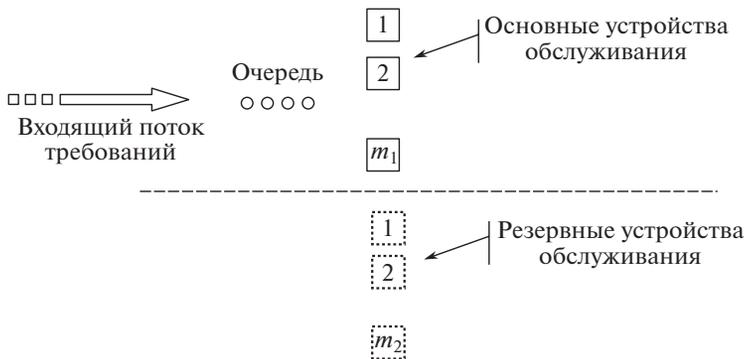


Рис. 1. Структурная схема исследуемой управляемой СМО.

каналы обслуживания. В предположении, что СМО должна выполнить все поступающие в нее требования, требуется минимизировать суммарные средние затраты на функционирование СМО в периоде планирования $[0, T]$. На каждом шаге в перечень затрат включаются затраты на содержание и эксплуатацию основных и резервных каналов обслуживания, затраты на решения о включении или отключении каналов (затраты на переключения) и затраты СМО на требования, пребывающие в очереди. При этом считается, что в СМО поступает простейший входящий поток, интенсивность которого $\lambda(t)$ на протяжении шага постоянна, а в моменты контроля претерпевает скачкообразные изменения, принимая конечное число k значений λ_i из дискретного множества $\Lambda = \{\lambda_i, i \in \overline{1, k}\}$.

2.2. Математическая постановка и решение задачи

Итак, рассматривается СМО, в которой число рабочих каналов обслуживания является управляемой величиной и может быть изменено в периодически (с шагом, равным единице) расположенные на оси времени моменты контроля за состоянием СМО. Задана матрица вероятностей перехода соответствующей однородной марковской цепи $\mathbf{P} = \|p_{ij}\|$, где p_{ij} — вероятность перехода (в момент контроля) от интенсивности $\lambda_i, i \in \overline{1, k}$, на предыдущем шаге к интенсивности $\lambda_j, j \in \overline{1, k}$, на следующем шаге.

Если на данном шаге интенсивность входящего потока равна λ_i , а интенсивность обслуживания на одном рабочем канале составляет μ , то, чтобы в рассматриваемой СМО на данном шаге мог установиться стационарный в вероятностном смысле режим функционирования, число основных рабочих каналов u в СМО должно выбираться, чтобы удовлетворять неравенству [4, 15]:

$$(1) \quad u \geq u_{\text{крит}}(\lambda_i) = \underline{u}_i = \left\lceil \frac{\lambda_i}{\mu} \right\rceil + 1,$$

где через $\lceil \bullet \rceil$ обозначена целая часть числа.

Рассмотрим состояние СМО в момент контроля за n шагов длительности τ до конца периода планирования. Будем считать, что в начальный момент времени случайная интенсивность входящего потока приняла на этом шаге значение λ_i , к этому моменту в системе имелось m основных каналов обслуживания и принимается решение о том, что в СМО должно работать u основных каналов обслуживания.

Запишем выражение для суммарных средних затрат $C^{(1)}(\lambda_i, m, u)$ на первом шаге процесса управления (стартующем за n шагов до конца периода планирования), считая, что затраты на эксплуатацию резервных устройств обслуживания равны нулю. В силу сделанных выше замечаний эти затраты могут быть представлены в виде

$$(2) \quad C^{(1)}(\lambda_i, m, u) = C_{\text{экспл}}(u) + C_{\text{очереди}}(\lambda_i, u) + C_{\text{переключ}}(\lambda_i, m, u).$$

В (2) $C_{\text{экспл}}(u)$ – это одношаговые затраты на эксплуатацию u основных устройств ($C_{\text{экспл}}(u) = c_1 u$, где c_1 – цена эксплуатации основного устройства обслуживания на протяжении одного шага); $C_{\text{очереди}}(\lambda_i, u)$ – одношаговые затраты на очередь в стационарном режиме, которые будем считать пропорциональными средней длине очереди $\bar{L}_{\text{очереди}}(\lambda_i, u)$ как функции от интенсивности входящего потока λ_i и управляющего решения о числе u основных каналов; $C_{\text{переключ}}(\lambda_i, m, u)$ – затраты на переключения, которые зависят от интенсивности входящего потока λ_i , от начального (до выполнения операции переключения) числа основных каналов m и от управляющего решения о числе основных каналов u . Будем считать, что эта компонента затрат может быть представлена в виде

$$(3) \quad C_{\text{переключ}} = \begin{cases} A_1, & \text{если } u > m; \\ 0, & \text{если } u = m; \\ A_2 + c_2(m - u), & \text{если } u < m, \end{cases}$$

где A_1 – фиксированная цена принятия решения о подключении новых рабочих устройств (“включение”), A_2 – это фиксированная цена принятия решения об отключении новых рабочих устройств (“выключение”), а c_2 – стоимость отключения одного рабочего устройства обслуживания ($c_1 > c_2$). В формуле (2) осталось конкретизировать только значение $C_{\text{очереди}}(\lambda_i, u)$. Будем считать, что пребывание каждого требования в очереди на протяжении одного шага обходится в сумму d , т.е. $C_{\text{очереди}}(\lambda_i, u) = d\bar{L}_{\text{очереди}}(\lambda_i, u)$. Пользуясь классическими результатами [4, 15], можно записать:

$$(4) \quad \bar{L}_{\text{очереди}} = \left[\sum_{k=1}^{u-1} \frac{(u\rho_i)^k}{k!} + \frac{(u\rho_i)^u}{u!(1-\rho_i)} \right]^{-1} \frac{(u\rho_i)^u \rho_i}{u!(1-\rho_i)^2},$$

где $\rho_i = \frac{\lambda_i}{u\mu}$.

Собираясь воспользоваться для решения проблемы методом дискретного динамического программирования, введем следующий промежуточный функционал, а именно функционал $C_n^*(\lambda_i, m)$, описывающий минимально возможное (т.е. при использовании еще не найденной оптимальной стратегии переключения каналов) значение средних суммарных затрат на последних n шагах процесса управления, когда в начале первого шага текущее число основных рабочих каналов равно m , а интенсивность простейшего входящего потока принимает значение λ_i . В этих обозначениях окончательное решение проблемы переключений на протяжении всего периода планирования будет описываться по уровню минимально возможных средних затрат величиной $C_N^*(\lambda, m)$, где λ и m – текущие значения интенсивности входящего потока и числа основных каналов обслуживания, с которых стартует СМО в периоде планирования $[0, T]$.

Теперь нетрудно записать систему уравнений дискретного динамического программирования, которая и является решением проблемы управления

процессом переключения каналов:

$$(5) \quad C_1^*(\lambda_i, m) = \min_{u \geq \underline{u}_i} C^{(1)}(\lambda_i, m, u),$$

$$(6) \quad C_n^*(\lambda_i, m) = \min_{u \geq \underline{u}_i} \left\{ C^{(1)}(\lambda_i, m, u) + \alpha \sum_{j=1}^l p_{ij} C_{n-1}^*(\lambda_j, u) \right\}, \quad n \in \overline{2, N},$$

где значение \underline{u}_i задается формулой (1), значение $C^{(1)}(\lambda_i, m, u)$ – формулой (2), $i \in \overline{1, k}$, а α – коэффициент дисконтирования, $0 \leq \alpha \leq 1$.

Преобразуем уравнения (5) и (6), введя вспомогательные функции $G_{\text{включ}}^{(1)}(\lambda_i, u)$, $G_{\text{выключ}}^{(1)}(\lambda_i, u)$, $G_{\text{включ}}^{(n)}(\lambda_i, u)$, $G_{\text{выключ}}^{(n)}(\lambda_i, u)$:

$$(7) \quad G_{\text{включ}}^{(1)}(\lambda_i, u) = uc_1 + d\bar{L}_{\text{очереди}}(\lambda_i, u),$$

$$(8) \quad G_{\text{выключ}}^{(1)}(\lambda_i, u) = u(c_1 - c_2) + d\bar{L}_{\text{очереди}}(\lambda_i, u),$$

$$(9) \quad G_{\text{включ}}^{(n)}(\lambda_i, u) = uc_1 + d\bar{L}_{\text{очереди}}(\lambda_i, u) + \alpha \sum_{j=1}^l p_{ij} C_{n-1}^*(\lambda_j, u), \quad n = 2, 3, \dots,$$

$$(10) \quad G_{\text{выключ}}^{(n)}(\lambda_i, u) = u(c_1 - c_2) + d\bar{L}_{\text{очереди}}(\lambda_i, u) + \alpha \sum_{j=1}^l p_{ij} C_{n-1}^*(\lambda_j, u),$$

$$n = 2, 3, \dots$$

Теперь уравнения (5) и (6) можно переписать в единообразном виде:

$$(11) \quad C_n^*(\lambda_i, m) =$$

$$= \min \begin{cases} -c_1 m + \min \begin{cases} A_1 + \min_{u \geq \underline{u}_i} G_{\text{включ}}^{(n)}(\lambda_i, u), \\ G_{\text{включ}}^{(n)}(\lambda_i, m), \end{cases} \\ -(c_1 - c_2)m + \min \begin{cases} A_2 + \min_{u \geq \underline{u}_i} G_{\text{выключ}}^{(n)}(\lambda_i, u), \\ G_{\text{выключ}}^{(n)}(\lambda_i, m), \end{cases} \end{cases} \quad n \in \overline{1, N}.$$

Поясним запись уравнений дискретного динамического программирования (5)–(6) в форме (11). Очевидно, что появление нижних индексов “включ” и “выключ” в функциях с общим обозначением G соответствует принятию решений о включении дополнительных резервных каналов или, наоборот, отключении части рабочих каналов. Решениям о включении и выключении соответствуют первая (включение) и третья (выключение) строки в крайней правой части формулы (11). В то же время вторая и четвертая строки в крайней правой части формулы (11) отвечают за принятие решений об отказе от каких-либо переключений. При этом сама оптимальная стратегия в новой переформулированной записи уравнений (11) пока еще не определена.

Прежде чем перейти к обсуждению решений уравнений (5)–(6) или (11), отметим, что поставленная задача в “близорукой” постановке ($n = 1$), которая сводится к решению задачи минимизации (2), подробно исследовалась в [13], где обсуждение опиралось на качественное использование аналогий с математическими задачами теории управления запасами и производством [16]. Опора на эту аналогию, в конечном счете, привела к тому, что для управляемых СМО удалось в несколько видоизмененной форме воспроизвести некоторые классические свойства стратегий оптимального управления запасами, например их параметрический характер (наличие двух критических уровней R и r). При этом упомянутые аналогии выстраивались между задачами управления запасами с непрерывными переменными состояниями, тогда как рассматриваемые задачи теории массового обслуживания по одной из переменных (значение интенсивности входящего потока) являются дискретными. Чтобы восполнить этот пробел, в разделе 3 настоящей статьи приводятся некоторые развивающие известные и новые результаты по математическим инструментам изучения “похожих” на выпуклые дискретных задач оптимизации.

3. A -выпуклость в дискретных задачах оптимизации

Рассматривается функция $g(i)$ от переменной i , принимающей значения из ряда натуральных чисел ($i = 1, 2, 3, \dots$). Напомним несколько известных понятий, которые тщательно изучены для случая вещественных (непрерывных) переменных и функций, а для случая дискретных переменных используются несколько реже.

Определение 1. Функция $g(i)$ называется выпуклой (вниз), если для всех натуральных чисел i и j

$$(12) \quad g(i + j) - g(i) - \Delta^{(1)}g(i) \times j \geq 0,$$

где $\Delta^{(1)}g(i)$ – первая разность функции $g(i)$ в точке i : $\Delta^{(1)}g(i) = g(i) - g(i-1)$.

Необходимым и достаточным условием выпуклости (вниз) функции является выполнение неравенства $\Delta^{(2)}g(i) \geq 0$, где $\Delta^{(2)}g(i)$ – вторая разность функции $g(i)$ в точке i . Действительно, пусть j из определения 1 равно единице, тогда формула (11) записывается как $\Delta^{(1)}g(i + 1) \geq \Delta^{(1)}g(i)$, т.е. первая разность $g(i)$ оказывается монотонно возрастающей функцией, т.е. $\Delta^{(2)}g(i) \geq 0$. Аналогично доказывается и достаточность.

Определение 2. Функция $g(i)$ называется A -выпуклой¹ ($A \geq 0$), если для всех натуральных чисел i и j :

$$(13) \quad A + g(i + j) - g(i) - \Delta^{(1)}g(i) \times j \geq 0.$$

¹ Понятие A -выпуклости было предложено Г. Скарфом [17] для анализа свойств оптимальных стратегий принятия решений (управления) в задачах логистики запасов при наличии фиксированных цен поставки, не зависящих от размера поставки, с добавлением к ним сумм, обусловленных размером партии поставки.

Для функций от дискретных переменных сохраняются все свойства A -выпуклых функций, помеченные Гербертом Скарфом [8], а именно:

Свойство 1. Если функция $g(i)$ является A -выпуклой, то при любом натуральном j функция $g(i + j)$ также A -выпукла.

Свойство 2. Если функция $g_1(i)$ является A_1 -выпуклой, а функция $g_2(i)$ — A_2 -выпуклой, то при любых $\theta_1, \theta_2 > 0$ функция $g(i) = \theta_1 g_1(i) + \theta_2 g_2(i)$ также является $(\theta_1 A_1 + \theta_2 A_2)$ -выпуклой.

Свойство 3. Если функция $g(i)$ является A_1 -выпуклой, то она и A_2 -выпукла для любого $A_2 > A_1$.

Свойство 4. Пусть i — случайная величина с распределением $\{p_i\}_1^k$, $p_i > 0$, $\sum_{i=1}^k p_i = 1$, и пусть функция $g(i)$ является A -выпуклой. Тогда функция $\alpha \sum_{i=1}^k p_i g(i)$, где $\alpha \geq 0$, также A -выпукла.

Отметим, что доказательство устанавливаемых в разделе 4 настоящей статьи фактов опирается на аналогию между задачами теории управления запасами и задачами теории массового обслуживания. Первичный анализ этих аналогий был выполнен в публикации [16], в которой была рассмотрена задача управления запасами, названная авторами “фантазийной”². Тем не менее нельзя не признать, что для рассматриваемого класса задач (как “фантазийной” из теории управления запасами³, так и задачи о переключении каналов), была продемонстрирована некоторая неготовность, неприспособленность существующих теории оптимизации и теории выпуклости к решению этих задач. Дело в том, что в при математическом описании рассматриваемого класса задач используется не одно, а два разных видения оптимальности (критерия оптимальности). Говоря на языке, адекватном рассматриваемому в данной статье классу задач, это *оптимальность при включении* каналов (при подаче заказов в теории управления запасами), если перемещаться по оси целочисленной переменной *и слева-направо*, и *оптимальность при отключении* каналов (возвращении товара в теории управления запасами), если перемещаться по оси переменной *и справа-налево*. Используем это замечание, чтобы сформулировать новые определения.

Определение 3. Функция $g(i)$ называется A -выпуклой слева ($A \geq 0$) в интервале $(-\infty, s]$, если для некоторого натурального числа s , которое обладает тем свойством, что оно находится правее точки абсолютного минимума функции $g(i)$, и для всех натуральных чисел i и j , удовлетворяющих неравенствам $i, i + j \leq s$, справедливо неравенство (13).

Определение 4. Функция $g(i)$ называется A -выпуклой справа ($A \geq 0$) в интервале $[r, \infty)$, если для некоторого натурального числа r , которое обладает тем свойством, что оно находится левее точки абсолютного мини-

² Фантастическим в [16] было предположение о том, что склад может не только подавать заказы на пополнение запасов, но и возвращать товар его поставщикам.

³ В настоящее время подвергнутый критике на ряде семинаров термин “фантазийная задача” заменен на новый — “задача управления запасами с возвратами”.

мума функции $g(i)$, и для всех натуральных чисел i и j , удовлетворяющих неравенствам $p \leq i, i + j$, справедливо неравенство (13).

4. Исследование свойств оптимальных стратегий переключения каналов

Чтобы воспользоваться введенными выше понятиями A -выпуклости, а также A -выпуклости слева и справа для выявления свойств оптимальной стратегии переключения каналов, убедимся в том, что для введенных выше функционалов (5), (6) и (11) эти свойства имеют место. Рассмотрим сначала «близорукий» случай, когда число шагов $n = 1$.

4.1. Свойства «близоруких» стратегий переключения каналов

Рассмотрим одношаговую задачу переключения каналов, которая представляет собой задачу, описываемую уравнением (5) или уравнением (11) при $n = 1$. В этом случае целевой функционал состоит в минимизации функции (7) при включении каналов или функции (8) при выключении каналов. В обоих случаях эти функционалы представляют собой сумму возрастающей линейной функции от назначаемого к включению числа каналов u и взвешенной с коэффициентом d средней длины очереди $\bar{L}_{\text{очереди}}(\lambda_i, u)$. Изучим свойства характеристики $\bar{L}_{\text{очереди}}(\lambda_i, u)$ как функции переменной u .

Из чисто вероятностных соображений следует, что функция $\bar{L}_{\text{очереди}}(\lambda_i, u)$, как функция от u , является монотонно убывающей функцией u , при этом такой, что скорость этого убывания с ростом u также монотонно убывает. При этом очевидно, что для любого значения λ_i предел $\lim_{u \rightarrow \infty} \bar{L}_{\text{очереди}}(\lambda_i, u) = 0$, и вполне понятно, что по мере роста числа каналов от минимально возможного значения \underline{u}_i величина $\bar{L}_{\text{очереди}}(\lambda_i, u)$ уменьшается, причем с ростом u это сокращение средней длины очереди $\bar{L}_{\text{очереди}}(\lambda_i, u)$ само становится все меньше, поскольку введение еще одного дополнительного рабочего канала на фоне большего числа рабочих каналов приносит выигрыш, «доля» которого $1/(u + 1)$ постоянно снижается. Однако оказывается, что строгое доказательство этого факта требует довольно утомительных выкладок (по крайней мере авторам статьи не удалось технически упростить соответствующие доказательства).

Изменим обозначения, сняв индекс i состояния СМО и заменив величину ρ_i из формулы (4) величиной $\rho_0 = \lambda/\mu$. В результате формула (4) приобретает вид

$$(14) \quad \bar{L}_{\text{очереди}}(\lambda, u) = \left[\sum_{k=1}^{u-1} \frac{(\rho_0)^k}{k!} + \frac{(\rho_0)^u}{(u-1)!(u-\rho_0)} \right]^{-1} \frac{(\rho_0)^{u+1}}{(u-1)!(u-\rho_0)^2}.$$

При $u - 1 \geq u_{\text{крит}}(\lambda) = \underline{u} = [\rho_0] + 1$ рассмотрим первую разность функции $\bar{L}_{\text{очереди}}(\lambda_i, u)$, как функции от u :

$$(15) \quad \Delta \bar{L}_{\text{очереди}}(\lambda, u) = \bar{L}_{\text{очереди}}(\lambda, u) - \bar{L}_{\text{очереди}}(\lambda, u - 1).$$

Утверждение 1. При $u - 1 \geq u_{\text{крит}}(\lambda) = \underline{u} = [\rho_0] + 1$ подсчитываемая по формуле (15) первая разность $\bar{L}_{\text{очереди}}(\lambda, u)$ отрицательна.

Доказательство. Используя формулы (14) и (15) и опуская промежуточные выкладки, запишем:

$$\begin{aligned} & \Delta \bar{L}_{\text{очереди}}(\lambda, u) = \\ & = \left[\sum_{k=1}^{u-1} \frac{(\rho_0)^k}{k!} + \frac{(\rho_0)^u}{(u-1)!(u-\rho_0)} \right]^{-1} \frac{(\rho_0)^{u+1}}{(u-1)!(u-\rho_0)^2} - \\ & - \left[\sum_{k=1}^{u-2} \frac{(\rho_0)^k}{k!} + \frac{(\rho_0)^u}{(u-2)!(u-1-\rho_0)} \right]^{-1} \frac{(\rho_0)^u}{(u-2)!(u-1-\rho_0)^2} = \\ & = -C_1(\rho_0, u) + C_2(\rho_0, u) [\rho_0(u-1-\rho_0)^2 - (u-1)(u-\rho_0)^2] \sum_{k=1}^{u-2} \frac{(\rho_0)^k}{k!}. \end{aligned}$$

В последнем выражении величины $C_1(\rho_0, u)$ и $C_2(\rho_0, u)$ положительны, а для множителя в квадратных скобках можно записать:

$$\begin{aligned} \rho_0(u-1-\rho_0)^2 - (u-1)(u-\rho_0)^2 & < \rho_0(u-\rho_0)^2 - (u-1)(u-\rho_0)^2 = \\ & = (\rho_0 - u + 1)(u - \rho_0)^2 < 0. \end{aligned}$$

Последнее неравенство записано в силу первого условия в утверждении 1. Из этих выкладок следует утверждение 1.

Введем вторую разность функции $\bar{L}_{\text{очереди}}(\lambda, u)$, как функция от u :

$$(16) \quad \begin{aligned} \Delta^{(2)} \bar{L}_{\text{очереди}}(\lambda, u) & = \Delta \bar{L}_{\text{очереди}}(\lambda, u) - \Delta \bar{L}_{\text{очереди}}(\lambda, u-1) = \\ & = \bar{L}_{\text{очереди}}(\lambda, u) - 2\bar{L}_{\text{очереди}}(\lambda, u-1) - \bar{L}_{\text{очереди}}(\lambda, u-2). \end{aligned}$$

Выполнив аналогичные выкладки, можно доказать следующее утверждение.

Утверждение 2. При $u - 2 \geq u_{\text{крит}}(\lambda) = \underline{u} = [\rho_0] + 1$ рассчитываемая по формуле (16) вторая разность функции $\bar{L}_{\text{очереди}}(\lambda, u)$ положительна.

Из утверждений 1 и 2 следует, что средняя длина очереди $\bar{L}_{\text{очереди}}(\lambda, u)$ является выпуклой вниз функцией переменной u . Из последнего утверждения и свойств 2 и 3 вытекает, что функции $G_{\text{включ}}^{(1)}(\lambda, u)$ из формулы (7) и $G_{\text{выключ}}^{(1)}(\lambda, u)$ из формулы (8) выпуклы вниз. Последнее в силу формул (5) и (11) и по аналогии с рассуждениями при доказательстве факта A -выпуклости в главе 7 монографии [18], казалось бы, должно было привести к выводу о том, что функция $C_1^*(\lambda, m)$ является A -выпуклой. Однако такой вывод оказывается поспешным, поскольку структура функции $C_1^*(\lambda, m)$ более сложна, чем те конструкции, которые были рассмотрены в классической публикации по теории управления запасами и производством [17]: она “склеена” из функции средних затрат при включении дополнительных рабочих каналов (через функцию $G_{\text{включ}}^{(1)}(\lambda, u)$) и функции средних затрат при отключении части

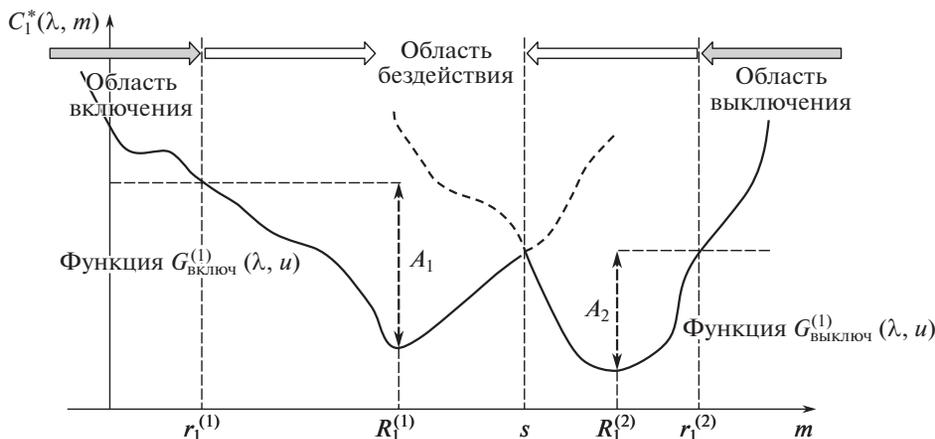


Рис. 2. Вид функции $C_1^*(\lambda, m)$.

рабочих каналов (через функцию $G_{\text{выключ}}^{(1)}(\lambda, u)$). Чтобы окончательно разобраться в характере этой "склейки", выделим точки абсолютных минимумов функций $G_{\text{включ}}^{(1)}(\lambda, u)$ и $G_{\text{выключ}}^{(1)}(\lambda, u)$, обозначив их через $R_1^{(1)}$ и $R_1^{(2)}$ соответственно. Из формул (7) и (8) очевидно, что $R_1^{(1)} < R_1^{(2)}$. Таким образом, в силу формулы (11) для случая $n = 1$ график функции $C_1^*(\lambda, m)$ может быть представлен таким, как это показано на рис. 2.

Из рис. 2 видно, что в силу упорядочения точек абсолютных минимумов $R_1^{(1)}$ и $R_1^{(2)}$ функций $G_{\text{включ}}^{(1)}(\lambda, u)$ и $G_{\text{выключ}}^{(1)}(\lambda, u)$ существует точка s пересечения соответствующих фрагментов графиков (пунктиром на рис. 2 показаны нереализуемые части графиков). Эта точка играет роль точки s и, одновременно, точки p из определений 3 и 4. Таким образом, становится понятным, что до точки s функция $C_1^*(\lambda, m)$ является A_1 -выпуклой слева, а после точки s — A_2 -выпуклой справа. Тогда, следуя логике формирования оптимальных двухуровневых стратегий в теории управления запасами [17, 18], оптимальное правило переключения каналов может быть формализовано в виде:

$$(17) \quad u = \begin{cases} R_1^{(1)}, & \text{если } m \leq r_1^{(1)} \text{ (включение),} \\ m, & \text{если } r_1^{(1)} < m \leq r_1^{(2)}, \\ R_1^{(2)}, & \text{если } m \geq r_1^{(2)} \text{ (отключение).} \end{cases}$$

При этом параметры $r_1^{(1)}$ и $r_1^{(2)}$ в формуле (17)⁴ представляют собой решения следующих уравнений (см. рис. 2):

$$(18) \quad A_1 + G_{\text{включ}}^{(1)}(\lambda, R_1^{(1)}) = G_{\text{включ}}^{(1)}(\lambda, r_1^{(1)}) \quad \text{для } r_1^{(1)} < R_1^{(1)},$$

$$(19) \quad A_2 + G_{\text{выключ}}^{(1)}(\lambda, R_1^{(2)}) = G_{\text{выключ}}^{(1)}(\lambda, r_1^{(2)}) \quad \text{для } r_1^{(2)} < R_1^{(2)}.$$

⁴ Применение формулы (17) связано с некоторыми оговорками, суть которых заключается в том, что в некоторых случаях, перечисленных в [13], параметр $R_{1,i}^{(2)}$ назначается равным \underline{u}_i . Имеются и другие нюансы.

Осталось заметить, что, на самом деле, четыре параметра “близорукой” стратегии управления запасами должны быть снабжены индексом i состояния, в котором СМО находится на первом шаге, т.е. фактически их надо обозначать как $r_{1,i}^{(1)}$, $R_{1,i}^{(1)}$, $r_{1,i}^{(2)}$ и $R_{1,i}^{(2)}$, рассчитывая эти параметры для всех $i \in \overline{1, k}$.

4.2. Многошаговые динамические задачи о переключении каналов

Вернемся к решению многошаговой задачи управления переключениями каналов, которая описывается уравнениями (5)–(6) или (11). В этом случае функции $G_{\text{вк\!л\!ю\!ч}}^{(1)}(\lambda, u)$ и $G_{\text{вык\!л\!ю\!ч}}^{(1)}(\lambda, u)$ уступают место функциям $G_{\text{вк\!л\!ю\!ч}}^{(n)}(\lambda, u)$ и $G_{\text{вык\!л\!ю\!ч}}^{(n)}(\lambda, u)$, $n = 2, 3, \dots$, задаваемым формулами (9) и (10). Повторим запись целевого функционала в уравнении (6):

$$C_n^*(\lambda_i, m) = \min_{u \geq \underline{u}_i} \left\{ C^{(1)}(\lambda_i, m, u) + \alpha \sum_{j=1}^l p_{ij} C_{n-1}^*(\lambda_j, u) \right\}.$$

Выражение в фигурных скобках в правой части этого уравнения, независимо от того, какая альтернатива оценивается (включение или отключение), содержит не меняющийся при смене альтернативы член $\alpha \sum_{j=1}^l p_{ij} C_{n-1}^*(\lambda_j, u)$. Одношаговый добавок $C^{(1)}(\lambda_i, m, u)$, являющийся, как показано в подразделе 4.1, выпуклой вниз функцией от u , наоборот, варьирует при смене альтернативы. Из этого можно сделать вывод, что есть все основания считать, что выражение в фигурных скобках в правой части формулы (6) (также двухвариантное) является одновременно A_1 -выпуклым слева и A_2 -выпуклым справа. Вопрос только в том, как будут упорядочены точки абсолютного минимума функций $G_{\text{вк\!л\!ю\!ч}}^{(n)}(\lambda, u)$ и $G_{\text{вык\!л\!ю\!ч}}^{(n)}(\lambda, u)$, которые обозначим как $R_{n,i}^{(1)}$ и $R_{n,i}^{(2)}$.

Утверждение 3. Для любого n справедливо неравенство $R_{n,i}^{(1)} < R_{n,i}^{(2)}$.

Доказательство. Используя формулы (9) и (10), которые можно переписать в виде

$$G_{\text{вк\!л\!ю\!ч}}^{(n)}(\lambda, u) = uc_1 + \beta^{(n)}(\lambda_i, u) \quad \text{и} \quad G_{\text{вык\!л\!ю\!ч}}^{(n)}(\lambda, u) = u(c_1 - c_2) + \beta^{(n)}(\lambda_i, u),$$

устанавливаем искомое неравенство для любого n .

Теперь выскажем предположение математической индукции о том, что это для любого номера n функции $G_{\text{вк\!л\!ю\!ч}}^{(n)}(\lambda, u)$ и $C_n^*(\lambda, m)$ являются A_1 -выпуклыми слева и A_2 -выпуклыми справа. Дальнейшее продвижение заключается в том, что следует установить факт A_1 -выпуклости слева и A_2 -выпуклости справа функции $G_{\text{вк\!л\!ю\!ч}}^{(n+1)}(\lambda, u)$. Тогда из уравнения (11) и утверждения 3 будет следовать, что A_1 -выпуклой слева и A_2 -выпуклой справа будет функция $C_{n+1}^*(\lambda, m)$. В свою очередь из факта A_1 -выпуклости слева и A_2 -выпуклости справа функции $G_{\text{вк\!л\!ю\!ч}}^{(n+1)}(\lambda, u)$ вытекает (по аналогии с подразделом 4.1), что

оптимальная стратегия переключения каналов будет определяться правилом:

$$(20) \quad u = \begin{cases} R_{n+1,i}^{(1)}, & \text{если } m \leq r_{n+1,i}^{(1)} \text{ (включение),} \\ m, & \text{если } r_{n+1,i}^{(1)} < m \leq r_{n+1,i}^{(2)}, \\ R_{n+1,i}^{(2)}, & \text{если } m \geq r_{n+1,i}^{(2)} \text{ (отключение).} \end{cases}$$

При этом параметры $r_{n+1,i}^{(1)}$ и $r_{n+1,i}^{(2)}$ в формуле (20) представляют собой решения следующих уравнений (см. рис. 2):

$$(21) \quad A_1 + G_{\text{выключ}}^{(n+1)}(\lambda, R_{n+1,i}^{(1)}) = G_{\text{выключ}}^{(n+1)}(\lambda, r_{n+1,i}^{(1)}) \quad \text{для } r_{n+1,i}^{(1)} < R_{n+1,i}^{(1)},$$

$$(22) \quad A_2 + G_{\text{выключ}}^{(n+1)}(\lambda, R_{n+1,i}^{(2)}) = G_{\text{выключ}}^{(n+1)}(\lambda, r_{n+1,i}^{(2)}) \quad \text{для } r_{n+1,i}^{(2)} < R_{n+1,i}^{(2)}.$$

Нетрудно видеть, что доказательство сформулированного выше утверждения математической индукции идейно повторяет доказательство аналогичного утверждения (для двухуровневых стратегий) в теории управления запасами [18] с поправкой на дискретность и отмеченную выше альтернативность (включение и выключение каналов) целевого функционала.

Как отмечалось выше, немаловажным фактом, который позволяет не опасаться “подводных камней” в форме скользящих режимов, является упорядочение точек абсолютного минимума альтернативных функционалов в форме неравенств $R_{n,i}^{(1)} < R_{n,i}^{(2)}$, $n \in \overline{1, N}$, $i \in \overline{1, k}$.

5. Заключение

Рассмотрена многошаговая задача оптимального переключения каналов в многолинейной системе массового обслуживания. Исследованы стратегии переключения каналов в дискретном времени при Марковском описании процесса изменения интенсивности входящего потока. Критерием для выбора стратегий переключения является минимизация суммарных средних затрат на многошаговом периоде планирования. Предполагается, что процедура переключения (включения или отключения) каналов приводит к расходам, которые состоят из фиксированных платежей и затрат, величина которых зависит от числа включаемых или отключаемых рабочих каналов. Доказано, что оптимальные стратегии переключения каналов оказываются параметрическими и на каждом шаге оптимальное решение задачи о переключениях зависит только от четырех характеристических параметров. Указаны способы расчета этих параметров.

СПИСОК ЛИТЕРАТУРЫ

1. Рыков В.В. Управляемые системы массового обслуживания // Теория вероятностей. Математическая статистика. Теоретическая кибернетика. 1975. Т. 12. С. 43–153.
2. Неймарк Ю.И. Динамические системы и управляемые процессы. М.: Наука, 1978.

3. *Гольшиева Н.М., Федоткин М.А.* Циклическое управление конфликтными потоками в условиях гибели и рождения очередей критических размеров // *АиТ.* 1990. № 4. С. 68–75.
Golyshева N.M., Fedotkin M.M. A Conflict Flows Cyclic Control in Conditions of Critical Queues Deaths and Births // *Autom. Remote Control.* 1990. V. 51. No. 4. P. 479–484.
4. *Вишневецкий В.М.* Теоретические основы проектирования компьютерных сетей. М.: Техносфера, 2003.
5. *Захаров П.П.* Разработка автоматизированного программного комплекса для исследования качества и эффективности функционирования моделей технических систем и управляемых систем массового обслуживания. Дисс. канд. техн. наук. М.: МИЭМ, 2006.
6. *Ivanov R., Mukhtarov A., Pershin O.* A Problem of Optimal Location of Given Set of Base Stations in Wireless Networks with Linear Topology // *Distributed Computer and Communication Networks. 22nd Int. Conf., DCCN 2019, Moscow, Russia, September 23–27, 2019, Revised Selected Papers.* Springer, 2019. P. 53–64.
7. *Галлинина О.С., Андреев С.Д., Кучерявый Е.А.* Оптимальное управление мощностью передачи устройства в гетерогенных сетях связи. // Матер. 21-й Междунар. науч. конф. “Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь” (DCCN-2018, Москва). М.: ИПУ РАН, 2018. С. 79–83.
8. *Вишневецкий В.М., Дудин А.Н.* Системы массового обслуживания с коррелированными входными потоками и их применение для моделирования телекоммуникационных сетей // *АиТ.* 2017. № 8. С. 3–59.
Vishnevskii V.M., Dudin A.N. Queueing Systems with Correlated Arrival Flows and Their Applications to Nodeling Telecommunication Networks // *Autom. Remote Control.* 2017. V. 74. No. 8. P. 1361–1403.
9. *Mandel A.* Econometric Models of Controllable Multiple Queuing Systems // *Proc. 18th Int. Conf. “Distributed Computer and Communication Networks” (DCCN 2015, Moscow, Russia).* Geneva: Springer, 2016. P. 296–304.
10. *Мандель А.С., Барладян И.И., Токмакова А.Б.* Многолинейная СМО с изменением числа рабочих каналов: стационарный случай. Ч. 1 // Материалы 18-й Международной научной конференции “Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь” (DCCN-2015, Москва). М.: ИПУ РАН, 2015. С. 345–354.
11. *Мандель А.С., Махукова В.В.* Многолинейная СМО с изменением числа рабочих каналов: нестационарный случай. Ч. 2 // Матер. 18-й Междунар. науч. конф. “Распределенные компьютерные и телекоммуникационные сети: управление, вычисление, связь” (DCCN-2015, Москва). М.: ИПУ РАН, 2015. С. 355–361.
12. *Mandel A., Bakulin K.* Models of Controllable Multiple Queuing Systems for Channel Switching Myopic Strategies // *Proc. 20th Int. Conf. “Distributed Computer and Communication Networks” (DCCN 2017, Moscow, Russia).* М.: Техносфера, 2017. С. 534–542.
13. *Mandel A., Laptin V.* Myopic Channel Switching Strategies for Stationary Mode: Threshold Calculation Algorithms / *Vishnevskiy V., Kozyrev D.* (eds). *Distributed Computer and Communication Networks. DCCN 2018. Communications in Computer and Information Science.* V. 919. Geneva: Springer, 2018.
https://doi.org/10.1007/978-3-319-99447-5_35

14. *Mandel A., Laptin V.* Channel Switching Threshold Strategies for Multichannel Controllable Queuing Systems // Communications in Computer and Information Science. 2020. V. 1337. P. 259–270. link.springer.com/chapter
https://doi.org/10.1007/978-3-030-66242-4_21
15. *Гнеденко Б.В., Коваленко И.Н.* Введение в теорию массового обслуживания. М.: Наука, 1966.
16. *Mandel A., Granin S.* Investigation of Analogies between the Problems of Inventory Control and the Problems of the Controllable Queuing Systems // Proc. 2018 Eleventh Int. Conf. “Management of large-scale system development” (MLSD’2018). IEEE, 2018. P. 1–4. <https://doi.org/10.1109/MLSD.2018.8551852>
17. *Arrow K., Karlin S., Scarf H.* Studies in the Mathematical Theory of Inventory and Production. Stanford University Press, 1958.
18. *Хедли Дж., Уайтин Т.* Анализ систем управления запасами. М.: Наука, 1969.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 20.01.2021

После доработки 20.03.2021

Принята к публикации 30.06.2021

© 2021 г. В.А. ФУРСОВ, д-р техн. наук (furso@ssau.ru),
Е.Ю. МИНАЕВ, канд. техн. наук (eminaev@gmail.com)

(Самарский университет),

А.П. КОТОВ, канд. техн. наук (kotov@ssau.ru)

(Институт систем обработки изображений — филиал ФНИЦ

“Кристаллография и фотоника” РАН, Самара)

ОЦЕНИВАНИЕ ПАРАМЕТРОВ ДВИЖЕНИЯ АППАРАТА ПО НАБЛЮДЕНИЯМ ОПОРНОЙ ПОВЕРХНОСТИ¹

Рассматривается задача визуальной одометрии по последовательности видеокладов, которые формируются с использованием направленной перпендикулярно вниз камеры. Предлагается адаптивная двухэтапная технология визуальной одометрии, основанная на последовательном определении межкадровых сдвигов и регулярной корректировке текущих оценок координат. На первом этапе сдвиг между двумя последовательными кадрами определяется корреляционным методом и сопоставляемые видеоклады с использованием найденных параметров сдвига совмещаются с точностью до пикселя. На втором этапе сдвиги уточняются с субпиксельной точностью с использованием метода оптического потока. Для повышения надежности осуществляется отбор наиболее согласованных оценок оптического потока. Приводятся результаты экспериментальных исследований на имеющихся в открытом доступе данных съемок, подтверждающие высокую надежность и точность оценок.

Ключевые слова: визуальная одометрия, корреляционный метод, оптический поток, опорная плоскость, автономная навигация.

DOI: 10.31857/S000523102110010X

1. Введение

Технологии управления движением транспортных средств с использованием средств визуальной одометрии все шире внедряются в практику [1–4]. Методы визуальной одометрии позволяют оценивать положение и ориентацию аппарата с помощью анализа изменений, вызванных движением камеры на последовательности изображений. Примером ранних исследований в этой области является программа исследования Марса НАСА, где визуальная одометрия использовалась для оценки положения марсоходов на пересеченной местности [5, 6].

Традиционные непрямые методы визуальной одометрии [7–9] используют геометрический анализ разреженных соответствующих ключевых точек для определения изменений между входными видеокладами. В таком случае задача визуальной одометрии сводится к задаче минимизации ошибок репроекции.

¹ Работа выполнена в рамках Государственного задания по теме FSSS-0777-2020-0017.

Прямые методы визуальной одометрии [10–12], наоборот, стремятся совместно определить плотную (полуплотную) регистрацию изменений входных видеок кадров и параметров модели движения камеры. В результате вычисления плотного поля соответствий между изображениями задача прямых методов сводится к минимизации фотометрической ошибки.

Для прямых методов визуальной одометрии в качестве алгоритмов оценки изменений между изображениями широко используют оптический поток [13] и корреляционные методы [14], для непрямых методов применяют широкий спектр различных детекторов ключевых точек Harris detector [15], SIFT [16], SURF [17], FAST [18] и ORB [19].

В публикации [14] продемонстрирована высокая надежность применения корреляционного метода на реальном маршруте. Однако в корреляционном методе относительный сдвиг определяется с точностью до целого числа пикселей. Возникающие при этом субпиксельные ошибки могут накапливаться и вносить заметный вклад в общую ошибку оценивания траектории. В [20] предпринята попытка преодолеть этот недостаток путем вычисления субпиксельных относительных сдвигов с использованием модифицированного коэффициента корреляции. Модифицированный метод субпиксельной корреляции основан на предположении, что корреляционный отклик убывает по экспоненте. Однако предположения в отношении величины показателя экспоненты при этом могут не выполняться.

В [21, 22] для определения относительных сдвигов использовался оптический поток. Параметры сдвига методом оптического потока в отличие от корреляционного метода могут быть определены с субпиксельной точностью. Однако надежность решений часто оказывается низкой вследствие неизбежных помех и искажений видеок кадров, которые обычно зависят от области локализации используемых фрагментов. Кроме того, методы оптического потока ненадежно работают при больших относительных сдвигах между видеок кадрами, которые являются следствием либо высокой скорости движения аппарата, либо малой скорости регистрации видеок кадров.

Нейронные сети для решения задачи визуальной одометрии [4, 23] показывают высокую точность, однако результаты сильно зависят от наборов данных, на которых обучались сети. Надежность этих методов при незнакомых сценах и условиях съемки не поддается оценке.

Одним из путей повышения точности одометрии является автоматическая калибровка в режиме функционирования по наблюдениям, поступающим от дополнительных источников информации. В частности, в последние годы строятся технологии, основанные на комбинировании нескольких сенсоров и методов обработки наблюдений [13, 24, 25]. Конечно, это связано с существенным возрастанием расходов на установку дополнительных сенсоров, тем не менее такой подход оправдан в случае, когда речь идет о безопасности движения, например в системах управления железнодорожными транспортными средствами, воздушными судами и др.

Таким образом, идея комбинирования сама по себе плодотворна и находит все больше сторонников. Однако представляется, что повышение надежно-

сти путем комбинирования только методов обработки информации без установки дополнительных сенсоров является также перспективным направлением. В рамках этого подхода появляется возможность строить не только более дешевые, но и более компактные системы управления транспортными средствами. В ряде случаев достижение минимальных габаритно-весовых характеристик систем визуальной одометрии является решающим фактором. Идея настоящей статьи состоит в комбинировании методов и алгоритмов на информационном уровне без применения дополнительных аппаратных средств.

Статья построена следующим образом. В разделе 2 рассматривается схема наблюдения и вводится система координат. В разделе 3 описан алгоритм построения оценок с использованием корреляционного метода. В разделе 4 приведен алгоритм субпиксельного уточнения оценок с использованием уравнений оптического потока. В разделе 5 описана общая схема предлагаемой двухэтапной технологии с эпизодической корректировкой текущих оценок параметров траектории. В разделе 6 приведены результаты экспериментальных исследований.

2. Постановка задачи

В последние годы все больший интерес проявляется к системам, в которых последовательность видеок кадров формируется с использованием направленной перпендикулярно вниз камеры [14, 23, 24]. Связано это, по-видимому, с тем, что последовательность изображений поверхности, по которой перемещается аппарат, формируется непрерывно, в то время как при обзоре окружающей сцены на некоторых участках траектории возможны пропуски вследствие отсутствия ориентиров на доступных для камеры расстояниях.

В статье рассматривается задача построения оценок двумерных координат траектории по последовательности изображений, получаемых при движении аппарата по плоской опорной поверхности. В случае летательного аппарата предполагается, что движение осуществляется на одной высоте над заданной плоскостью (в горизонтальном полете — это плоскость местного горизонта). Схема съемки при движении по опорной плоскости показана на рис. 1.

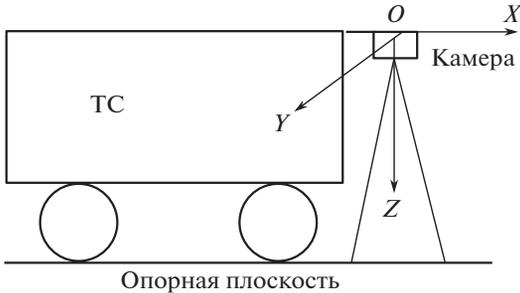


Рис. 1. Схема съемки.

Используем следующую систему координат. Начало системы координат, связанной с камерой, находится в центре фокальной плоскости камеры. Главная оптическая ось камеры направлена на опорную поверхность по перпендикуляру к плоскости в направлении оси Oz (см. рис. 1). Направление оси Ox системы координат, связанной с камерой, совпадает с направлением оси Ox системы координат, связанной с опорной плоскостью. Ось Oy образует правую систему координат.

При движении транспортного средства (ТС) система координат, связанная с камерой, совершает плоскопараллельные перемещения в плоскости, параллельной опорной. Это означает, что при жестком закреплении камеры изменение направления движения аппарата осуществляется без поворота вокруг некоторой оси. Если изменение направления движения аппарата осуществляется путем его поворота вокруг некоторой оси, камера должна стабилизироваться в заданном направлении с помощью специальных (например, гироскопических) устройств.

Для определения траектории в любой другой системе координат должен быть задан закон движения системы координат, связанной с опорной плоскостью, относительно этой системы координат. Не будем касаться этих аспектов и ограничимся задачей построения оценок траектории на опорной плоскости в прямоугольной системе координат, связанной с этой плоскостью.

Траекторию движения будем строить в виде непрерывной ломаной линии, составленной из отрезков прямых, точки *сопряжения* которых соответствуют моментам съемки камерой. Кадры, соответствующие концам каждого отрезка, далее будем называть *смежными*. Предполагаем, что частота съемки достаточно высокая, так что аппроксимация непрерывной траектории ломаной линией удовлетворяет по точности.

При указанном подходе технология визуальной одометрии состоит в определении последовательности относительных межкадровых сдвигов смежных кадров и формировании общей траектории аппарата путем их суммирования. Таким образом, оценки $\hat{x}_k, \hat{y}_k, k = 1, 2, \dots$, текущих координат аппарата в любой точке траектории определяются путем суммирования относительных сдвигов по осям Ox, Oy . Другими словами, текущие координаты траектории в каждой точке являются накопленной последовательностью относительных межкадровых сдвигов $\Delta x_k, \Delta y_k, k = 1, 2, \dots$, во всех предшествующих точках.

При определении сдвигов в каждой точке возможны ошибки, которые накапливаются по мере увеличения длины траектории. Ошибки одометрии имеют различные источники. Например, при использовании корреляционного метода относительные сдвиги могут быть определены лишь с точностью до целого числа пикселей. Свободный от этого недостатка метод оптического потока ненадежен при больших относительных сдвигах. Ошибки одометрии связаны также с отклонениями параметров движения аппарата от значений, при которых осуществлялась калибровка камеры. К сожалению, калибровка на длинных участках траекторий в расчете на средние значения параметров движения не спасает положение. Ошибки, связанные с кратковременными

нарушениями калибровки на малых отрезках траектории, обычно составляют значительную долю общей ошибки.

В настоящей статье для повышения точности и надежности оценок координат авторы предлагают двухэтапную технологию *монокулярной* визуальной одометрии. На первом этапе определяется межкадровый сдвиг корреляционным методом. На втором этапе сдвиги уточняются с субпиксельной точностью с использованием оптического потока. Кроме того, эпизодически осуществляется корректировка текущих оценок координат траектории по оценкам отклонений параметров движения аппарата от калибровочных значений на малых отрезках траектории.

3. Определение относительных сдвигов корреляционным методом

На первом этапе определяется относительный сдвиг кадров корреляционным методом. Для определения относительных сдвигов на каждой k -й паре кадров задается область поиска и определяются параметры сдвига $\Delta x_k(k)$, $\Delta y_k(k)$, $k = \overline{1, K}$, соответствующие максимальным значениям нормированных коэффициентов корреляции

$$(1) \quad K(f, g, k) = \frac{\sum_{(x,y) \in X(k)} (f(x, y) - f^0) (g(x, y) - g^0)}{\sqrt{\sum_{(x,y) \in X(k)} (f(x, y) - f^0)^2} \sqrt{\sum_{(x,y) \in X(k)} (g(x, y) - g^0)^2}},$$

где $f(x, y)$, $g(x, y)$, f^0 , g^0 – функции интенсивности и их средние значения на изображениях k -й пары кадров. В результате получаем оценки

$$(2) \quad \Delta \hat{x}(k), \Delta \hat{y}(k), \quad k = \overline{1, K},$$

сдвигов по осям Ox , Oy соответственно.

Реализация корреляционного метода связана с большим объемом вычислений. Это неизбежная плата за надежность оценок. Объем вычислений в значительной степени зависит от области поиска соответствующего фрагмента. Для сокращения вычислительных затрат применяется адаптивный алгоритм формирования области поиска.

Предполагается, что сдвиг между соседними кадрами не может превышать Δx_{thr} , Δy_{thr} . После каждого очередного определения координат сдвига к полученным абсолютным значениям сдвига (2) добавляются указанные пороговые значения, при этом получаем допустимые границы области поиска по обеим координатам:

$$(3) \quad \Delta x_{bord} = \Delta x(k) + \Delta x_{thr},$$

$$(4) \quad \Delta y_{bord} = \Delta y(k) + \Delta y_{thr}.$$

Эти значения сравниваются с размерами $\Delta x_{bord}(k-1)$, $\Delta y_{bord}(k-1)$ области поиска на предыдущем $(k-1)$ -м шаге определения относительных сдвигов

предшествующих кадров и осуществляется корректировка границ по правилам:

$$(5) \quad \text{если } \Delta x_{bord} > \Delta x_{bord}(k-1), \quad \text{то } \Delta x_{bord}(k) = \Delta x_{bord}(k-1) - 1,$$

$$(6) \quad \text{если } \Delta x_{bord} < \Delta x_{bord}(k-1), \quad \text{то } \Delta x_{bord}(k) = \Delta x_{bord}(k-1) + 1,$$

$$(7) \quad \text{если } \Delta y_{bord} > \Delta y_{bord}(k-1), \quad \text{то } \Delta y_{bord}(k) = \Delta y_{bord}(k-1) - 1,$$

$$(8) \quad \text{если } \Delta y_{bord} < \Delta y_{bord}(k-1), \quad \text{то } \Delta y_{bord}(k) = \Delta y_{bord}(k-1) + 1.$$

Получающаяся в результате прямоугольная область поиска оказывается максимально приближенной к ожидаемым возможным сдвигам. Такой метод формирования области поиска, кроме прочего, придает дополнительную устойчивость к единичным грубым сбоям оценивания сдвига, так как запрещает оценки сдвигов, которые не могут быть объяснены нормальной скоростью движения аппарата.

4. Субпиксельное уточнение сдвигов методом оптического потока

В классической задаче вычисления оптического потока предполагается, что пиксель с интенсивностью $I(x, y, t)$ в точке (x, y, t) кадра переходит к следующему кадру без изменений

$$(9) \quad I(x, y, t) \cong I(x + \Delta x, y + \Delta y, t + \Delta t).$$

Если сдвиги $(\Delta x, \Delta y, \Delta t)$ между этими кадрами невелики, то соотношение (9) с помощью ряда Тейлора можно записать как

$$(10) \quad I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + \tau,$$

где τ – малое значение, связанное с исключением членов высшего порядка. Из (10) с учетом (9) можно записать уравнение

$$(11) \quad I_x \Delta x + I_y \Delta y = -\Delta I_t,$$

где $\Delta I_t = \frac{\partial I}{\partial t} \cong \partial I$.

Уравнение (11) содержит два неизвестных. Особенность настоящей задачи состоит в том, что при отсутствии помех и искажений на изображениях все пиксели должны смещаться на одинаковое расстояние. Поэтому можно записать переопределенную систему уравнений

$$(12) \quad \mathbf{I}_{x,y} \mathbf{\Delta} = \mathbf{I}_t,$$

где $\mathbf{\Delta} = [\Delta x, \Delta y]^T$, \mathbf{I}_t – матрица-столбец размерности $N \times 1$, $\mathbf{I}_{x,y} = [\mathbf{I}_x, \mathbf{I}_y]$ – $(N \times 2)$ -матрица, а \mathbf{I}_x , \mathbf{I}_y , \mathbf{I}_t – $(N \times 1)$ -матрицы, составленные в соответствии

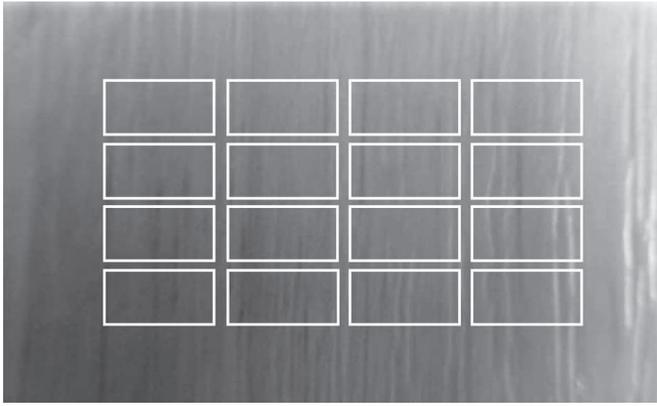


Рис. 2. Пример разбиения на фрагменты.

с соотношениями

$$\begin{aligned}
 \mathbf{I}_x &= \left[\frac{\partial I_1}{\partial x}, \frac{\partial I_2}{\partial x}, \dots, \frac{\partial I_N}{\partial x} \right]^T, \\
 \mathbf{I}_y &= \left[\frac{\partial I_1}{\partial y}, \frac{\partial I_2}{\partial y}, \dots, \frac{\partial I_N}{\partial y} \right]^T, \\
 \mathbf{I}_t &= \left[-\partial I_1, -\partial I_2, \dots, -\partial I_N \right]^T.
 \end{aligned}
 \tag{13}$$

В действительности, изображения всегда содержат шумы и искажения. Более того, характер и интенсивность искажений в разных частях изображений могут быть существенно различными вследствие колебаний аппарата во время движения. Для повышения надежности и точности определения относительных сдвигов применяем мультифрагментную схему наблюдения. Пример представления изображения в виде 16 фрагментов приведен на рис. 2.

Идея состоит в следующем. На первом кадре, например k -й пары кадров, зададим L фрагментов по всему полю изображения и определим относительные сдвиги $\Delta x_k(l)$, $\Delta y_k(l)$, $l = \overline{1, L}$, каждого l -го фрагмента на втором изображении. Далее из полученного множества оценок отбирается подмножество наиболее согласованных, для определенности обозначим это подмножество Θ .

Идея построения согласованных оценок была ранее реализована в ряде задач [26, 27]. В данном случае согласованность оценок выступает в качестве признака отсутствия ошибок. Если подмножество Θ существует, то имеем все основания полагать, что это подмножество содержит наиболее точные решения. Нахождение согласованных оценок обычно связано с перебором различных вариантов подмножеств, что требует значительных вычислительных затрат. В данном случае (имея в виду возможность реализации технологии в реальном времени) строим упрощенный алгоритм. В частности, подсчитываем число оценок с разными знаками. Результаты, образующие небольшое число оценок одного знака, исключаются. На оставшемся множестве оценок

ищется подмножество оценок, удовлетворяющее критерию минимума функции взаимной близости оценок [26]. Далее на этом подмножестве может быть построена точечная оценка любым известным методом. Полученные субпиксельные смещения добавляются к найденным корреляционным методом на первом этапе относительным смещениям.

5. Общая схема и алгоритм коррекции оценок координат

Основная проблема при определении текущих координат аппарата методами одометрии состоит в накоплении ошибок по мере увеличения длины траектории. Общая ошибка определения траектории включает ошибки оценивания межкадровых сдвигов, а также ошибки, связанные с тем, что параметры движения аппарата могут отличаться от тех, на которых осуществлялась калибровка камеры. Вариации параметров движения могут иметь место вследствие невозможности обеспечить точное значение заданной скорости в горизонтальной плоскости и точное расстояние от камеры до опорной плоскости. В частности, при колебаниях камеры в направлении оси Oz (см. рис. 1) нарушение калибровки воспринимается как кажущееся изменение горизонтальной скорости. Реальные (или кажущиеся) вариации параметров движения аппарата в окрестности заданного (среднего) калибровочного значения обычно носят кратковременный характер, тем не менее накопление этих ошибок происходит достаточно быстро.

В данной статье предлагается схема коррекции траектории в процессе формирования текущих оценок координат. Идея состоит в построении локальных оценок параметров движения аппарата на коротких отрезках траектории и эпизодическом внесении поправок в текущие оценки координат при их заметном отклонении от калибровочных значений. Здесь и далее полагаем, что длительность малого отрезка траектории $T = m \cdot \Delta t$, где m – заданное число кадров, по которым определяется текущая корректирующая поправка, а Δt – временной интервал между кадрами. Предлагаемая схема коррекции приведена на рис. 3.

Входной информацией в схеме оценки координат траектории с эпизодической коррекцией являются последовательность видеок кадров (блок 1) и ожидаемая средняя скорость, вычисленная по результатам предварительной калибровки камеры. Для простоты полагаем, что интервалы T одинаковы на всей траектории.

В приведенной схеме предлагаемая технология реализуется следующим образом. В блоке 1 с некоторого момента t на каждом такте из видеоследовательности выбирается два очередных кадра. Далее для каждого дискретного $t = k \cdot \Delta t$ определяются межкадровые сдвиги $\Delta \hat{x}_{Korr}(t)$, $\Delta \hat{y}_{Korr}(t)$ корреляционным методом по формуле (2). В блоке 2 осуществляется совмещение кадров путем сдвига второго кадра на найденное число пикселей. Ясно, что при отсутствии грубых ошибок корреляционного метода после совмещения кадров остающийся относительный сдвиг не превышает одного пикселя по любой координате.



Рис. 3. Схема коррекции оценок координат.

В блоке 3 относительный сдвиг кадров уточняется в пределах одного пикселя методом оптического потока. Найденные по формуле (12) субпиксельные сдвиги $\Delta\hat{x}_{Flow}(t)$, $\Delta\hat{y}_{Flow}(t)$ суммируются со сдвигами, найденными ранее корреляционным методом. В блоке 4 суммарные сдвиги, полученные на текущем такте корреляционным методом и из уравнений оптического потока, суммируются с оценками координат траектории, полученными на предыдущем такте. Таким образом, формируется последовательность отсчетов оценок координат траектории:

$$(14) \quad \hat{x}(t) = \hat{x}(t-1) + \Delta\hat{x}_{Korr}(t) + \Delta\hat{x}_{Flow}(t),$$

$$(15) \quad \hat{y}(t) = \hat{y}(t-1) + \Delta\hat{y}_{Korr}(t) + \Delta\hat{y}_{Flow}(t).$$

Блоки 5 и 6 образуют контур коррекции траектории. В блоке 5 из последовательности межкадровых сдвигов $\Delta\hat{x}(t) = \Delta\hat{x}_{Korr}(t) + \Delta\hat{x}_{Flow}(t)$, $\Delta\hat{y}(t) = \Delta\hat{y}_{Korr}(t) + \Delta\hat{y}_{Flow}(t)$ формируются сдвиги $\hat{x}(T)$, $\hat{y}(T)$ на заданном временном интервале T и с использованием информации о частоте кадров осуществляется оценка текущих локальных скоростей $\hat{V}_x(T)$, $\hat{V}_y(T)$:

$$(16) \quad \hat{V}_x(T) = \hat{x}(T)/T = \left(\sum_{i=t}^{t+T} \Delta\hat{x}(i) \right) / T,$$

$$(17) \quad \hat{V}_y(T) = \hat{y}(T)/T = \left(\sum_{i=t}^{t+T} \Delta\hat{y}(i) \right) / T.$$

Далее определяются отклонения оценок текущей скорости на малом отрезке траектории от заданной средней скорости движения:

$$(18) \quad \Delta V_x(T) = \hat{V}_x(T) - V_{cp},$$

$$(19) \quad \Delta V_y(T) = \hat{V}_y(T) - V_{cp}.$$

С использованием полученных отклонений в блоке 6 в дискретные моменты времени $t = NT$, $N = 1, 2, \dots$, формируются поправки координат траектории, которые затем суммируются с координатами, формируемыми на выходе блока 4. Соотношения, реализуемые блоком 6 и следующим за ним сумматором, с учетом (18), (19) можно представить в виде:

$$(20) \quad \hat{x}_{final}(NT) = \hat{x}(NT) + w_x \cdot \Delta V_x(NT),$$

$$(21) \quad \hat{y}_{final}(NT) = \hat{y}(NT) + w_y \cdot \Delta V_y(NT),$$

где w_x , w_y – весовые коэффициенты, учитывающие выбор системы отсчета, вероятность случайной составляющей ошибок, связанных с нарушениями калибровочных характеристик, диапазоны абсолютных значений отклонений скоростей ТС от среднего значения по осям Ox , Oy и др. Они уточняются на этапе предварительной калибровки.

6. Результаты экспериментальных исследований

Для экспериментов использовался набор данных Blackbird Dataset [28] для беспилотных летательных аппаратов (БПЛА), перемещающихся в помещениях, на основе специально созданной квадрокоптерной платформы. Набор Blackbird Dataset содержит данные 168 полетов, в каждом из которых регистрируются данные стереокамеры и обращенной вниз фотореалистичной виртуальной камеры с частотой 100 Гц. Известны также наблюдения эталонной траектории с частотой 360 Гц. В настоящей статье используется набор данных полета по траектории “Oval”, включающий 700 последовательных видеокадров. На рис. 4 приведена тестовая траектория в прямоугольной системе координат, построенная по эталонным точкам.

Для простоты полагаем, что все расстояния измеряются в пикселях, а интервал съемки между кадрами $\Delta t = 1$. Поскольку относительные сдвиги определяются по паре кадров, финальная оценка координат определена при $t_{fin} = 699$. Начальные значения координат задавались в соответствии с эталонными данными: $\Delta \hat{x}(1) = -1,412$, $\Delta \hat{y}(1) = -0,325$.

Цель экспериментов состояла в сравнительном исследовании точности корреляционного метода, субпиксельного уточнения оценок методом оптического потока и эпизодической корректировки текущих оценок координат траектории по оценкам отклонений параметров движения аппарата от калибровочных значений на малых отрезках траектории. Поскольку все указанные выше методы являются этапами сквозной технологии, результаты различных методов просто “выводились” на соответствующих промежуточных этапах ее реализации.

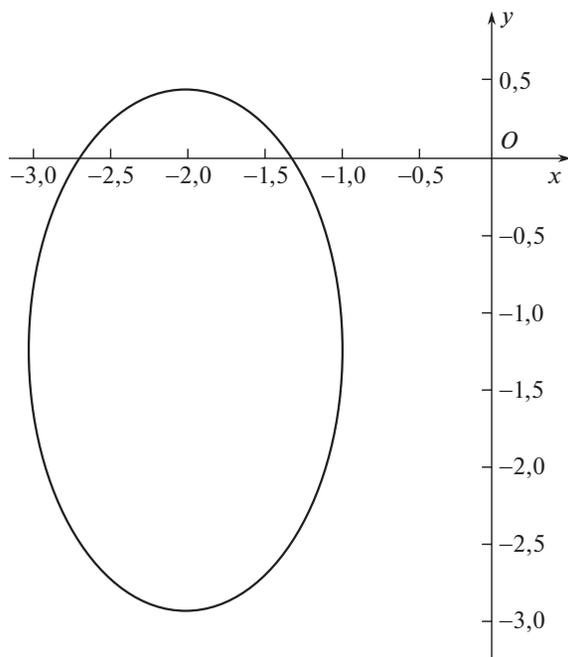


Рис. 4. Эталонная траектория.

На первом этапе определялась точность оценок траектории с использованием корреляционного метода. В этом эксперименте сдвиг второго кадра относительно первого определялся с использованием фрагментов в центре кадра размером 51×51 . Решение принималось по максимальному значению нормированного коэффициента корреляции (1). Область поиска на втором кадре задавалась на каждом шаге в соответствии с соотношениями (5)–(8). Графики зависимости ошибок оценивания координат X и Y корреляционным методом (символом o) приведены на рис. 5 и 6 соответственно.

Второй этап – субпиксельное уточнение оценок координат методом оптического потока. Для реализации второго этапа второй кадр очередной пары кадров сдвигался “назад” на целое число пикселей, которые предварительно были определены корреляционным методом на первом этапе. Искомые векторы оценок сдвига $\Delta = [\Delta x, \Delta y]^T$ определялись как решения переопределенной системы уравнений (12). Для каждой пары кадров формировалось 16 уравнений (12) для фрагментов с размерами 50×50 , расположенными по полю изображений так, как показано на рис. 2. Далее для каждой координаты из 16 полученных оценок строилась точечная оценка в виде среднего значения множества наиболее согласованных между собой оценок.

С использованием найденных на обоих этапах оценок сдвигов $\Delta \hat{x}_{Korr}$, $\Delta \hat{x}_{Flow}$, полученных корреляционным методом и по уравнениям оптического потока соответственно, определялись оценки координат в каждой точке траектории по соотношениям (14), (15). Начальные значения координат задавались в соответствии с тестовыми данными: $\Delta \hat{x}(1) = -1,412$, $\Delta \hat{y}(1) =$

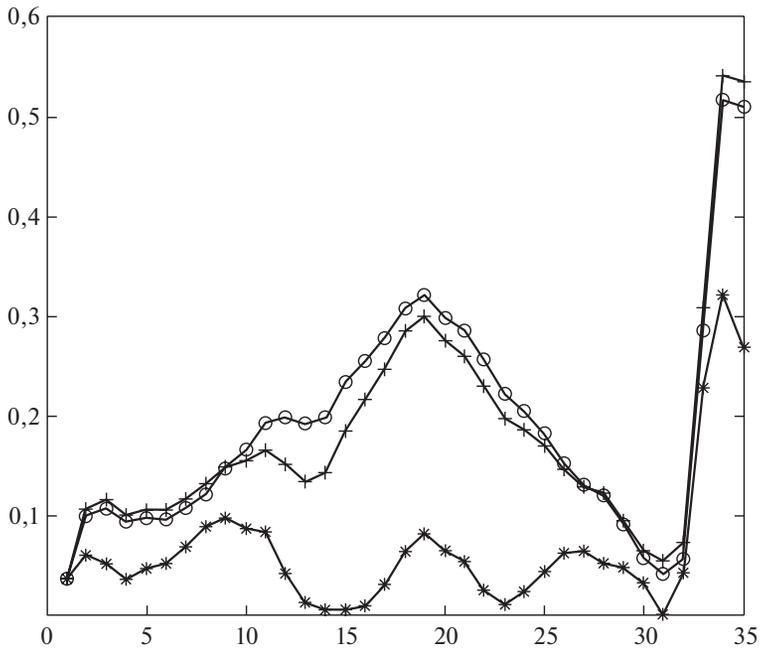


Рис. 5. Графики ошибок оценивания координаты X : корреляционный метод – символ o , с субпиксельной оценкой с эпизодической коррекцией – символ $*$.

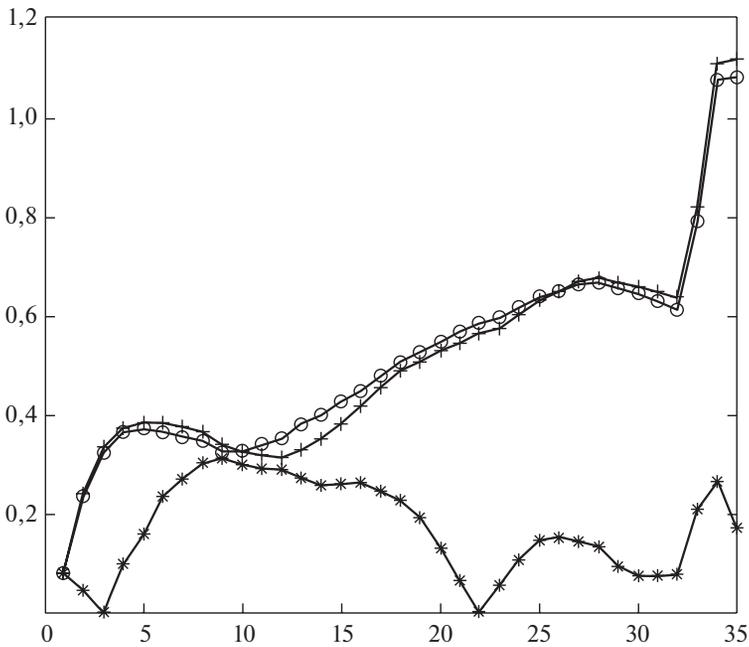


Рис. 6. Графики ошибок оценивания координаты Y : корреляционный метод – символ o , с субпиксельной оценкой с эпизодической коррекцией – символ $*$.

Ошибки оценивания координат траектории

№ п/п	Этапы получения оценок траектории	СКО на всей траектории		Финальные ошибки	
		Координата X	Координата Y	Координата X	Координата Y
1	Метод корреляции (первый этап)	0,2229	0,5564	0,5658	1,2069
2	Субпиксельные оценки (второй этап)	0,2148	0,5572	0,5952	1,2461
3	Результаты после коррекции оценок	0,0975	0,1981	0,3130	0,2516

$= -0,325$. Графики зависимости ошибок оценивания координат X и Y после второго этапа субпиксельного уточнения (символом \times) приведены на рис. 5 и 6 соответственно.

На завершающем этапе технологии проверялась эффективность предложенной схемы эпизодической коррекции траектории по локальным оценкам параметров движения аппарата на малых отрезках траектории. Для этого осуществили предварительную калибровку средней “цены” одного пикселя фотоприемного устройства (камеры). С использованием точных данных о текущих координатах на тестовом Dataset-е “Овал” подсчитали точный путь на всей траектории

$$S_{\text{sampl}} = \sum_{k=1}^{699} (x^2(k) + y^2(k))^{1/2} = 8,8635 \text{ метров.}$$

Затем построили оценки текущих координат траектории путем реализации описанной в настоящей статье двухэтапной технологии и подсчитали оцененный путь

$$S_{\text{fact}} = \sum_{k=1}^{699} (\Delta \hat{x}^2(k) + \Delta \hat{y}^2(k))^{1/2} = 5509,0 \text{ пикселей.}$$

Вычислили калибровочное значение цены пикселя

$$\Delta S_{\text{cost}} = S_{\text{sampl}}/S_{\text{fact}} = 8,8635/5509 = 0,0016$$

и средней скорости

$$V_{\text{cp}} = S_{\text{sampl}}/t_{\text{fin}} = 8,8635/699 = 0,0126.$$

Для выработки корректирующих сигналов на каждом интервале T по текущим оценкам сдвигов суммировались межкадровые сдвиги на $m = 20$ тактах. Оценки скорости на этих 20-ти тактах определялись по формулам (18), (19). Затем для каждой координаты формировались корректирующие оценки, которые в соответствии с формулами (20), (21) добавлялись к оценкам траектории, полученным до коррекции. При этом задавались весовые коэффициенты $w_x = 10,0$, $w_y = 15,0$.

Графики ошибок оценивания координат X и Y траектории с эпизодической коррекцией (символом $*$) приведены тоже на рис. 5 и 6 соответственно.

В таблице приведены данные о величине среднеквадратических ошибок (СКО) по всей длине траектории, полученные в ходе реализации всех этапов предложенной технологии.

7. Заключение

Реализована сквозная трехэтапная технология, включающая этапы оценки межкадровых сдвигов корреляционным методом, субпиксельное уточнение сдвигов методом оптического потока и коррекцию траектории по текущей информации об отклонениях сдвигов на локальных участках траектории от заданного среднего значения.

Как и следовало ожидать, субпиксельная поправка на втором этапе невелика. Связано это с тем, что после первого этапа совмещаются соседние кадры с точностью до пикселя и поправки не могут быть больше величины пикселя. В то же время фактический полный межкадровый сдвиг в данном примере на некоторых участках достигал около 20 пикселей.

Интересно, что среднеквадратическое отклонение в направлении оси Oy оказалось даже несколько выше (0,5572), чем было на первом этапе (0,5564), хотя визуально из графика на рис. 5 видно, что на большей части траектории ошибка после субпиксельного уточнения все-таки была меньше. Сбой оптического потока произошел на двух последних участках траектории, где действительно кадры оказались “смазанными”.

После завершающего этапа – коррекции оценок траектории – наблюдается существенное увеличение точности оценок координат траектории. Повышение точности носит устойчивый регулярный характер по всей длине траектории. Таким образом, показана эффективность предложенной схемы эпизодической коррекции траектории по текущим оценкам вариаций параметров движения на малых отрезках траекторий в окрестности заданного среднего значения, определяемого предварительной калибровкой камеры.

СПИСОК ЛИТЕРАТУРЫ

1. *Nister D., Naroditsky O., Bergen J.* Visual Odometry for Ground Vehicle Applications // *J. of Field Robotics*. 2006. V. 23. No. 1. P. 3–20.
2. *Fraundorfer F., Scaramuzza D.* Visual Odometry: Part i: The First 30 Years and Fundamentals // *IEEE Robotics and Automation Magazine*. 2011. V. 18. No. 4. P. 80–92.
3. *Fraundorfer F., Scaramuzza D.* Visual Odometry: Part ii: Matching, Robustness, Optimization, and Applications // *IEEE Robotics & Automation Magazine*. 2012. V. 19. No. 2. P. 78–90.
4. *Min Z., Yang Y., Dunn E.* VOLDOR: Visual Odometry From Log-Logistic Dense Optical Flow Residuals // *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2020. P. 4898–4909.

5. *Moravec H.P.* Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Ph.D. dissertation, Stanford, CA, USA. 1980.
6. *Matthies L., Shafer S.* Error Modeling in Stereo Navigation // IEEE J. Robot. Autom. 1987. V. 3. No. 3. P. 239–248.
7. *Kitt B., Geiger A., Lategahn H.* Visual Odometry based on Stereo Image Sequences with Ransac based Outlier Rejection Scheme // 2010 IEEE Intelligent Vehicles Sympos. 2010. P. 486–492.
8. *Pire T., Fischer T., Civera J., Cristoforis P., Berles J.* Stereo Parallel Tracking and Mapping for Robot Localization // 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). 2015. P. 1373–1378.
9. *Mur-Artal R., Tardos J.D.* Orb-slam2: An open-source slam system for monocular, stereo, and RGB-D cameras // IEEE Trans. Robotics. 2017. V. 33. No. 5. P.1255–1262.
10. *Newcombe R.A., Lovegrove S.J., Davison A.J.* DTAM: Dense tracking and mapping in real-time // 2011 Int. Conf. on Computer Vision. 2011. P. 2320–2327.
11. *Engel J., Schops T., Cremers D.* LSD-SLAM: Large-scale Direct Monocular SLAM // Eur. Conf. on Computer Vision. 2014. P. 834–849.
12. *Kerl C., Sturm J., Cremers D.* Dense Visual SLAM for RGB-D Cameras // 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. 2013. P. 2100–2106.
13. *Mohamed S.A., Haghbayan M.H., Westerlund T., Heikkonen J., Tenhunen H., Plosila J.* A Survey on Odometry for Autonomous Navigation Systems // IEEE Access. 2019. No. 7. P. 97466–97486.
14. *Nourani-Vatani N., Borges P.V.K.* Correlation-based Visual Odometry for Ground Vehicles // J. Field Robotics. 2011. V. 28. No. 5. P. 742–768.
15. *Harris C.G., Pike J.M.* 3D positional Integration from Image Sequences // Proc. Alvey Vis. Conf., Cambridge, U.K. 1987. P. 1–4.
16. *Lowe D.G.* Object Recognition from Local Scale-invariant Features // Proc. Int. Conf. Comput. Vis. (ICCV), Washington, DC, USA. 1999. V. 2. P. 1150–1157.
17. *Bay H., Ess A., Tuytelaars T., Van Gool L.* Speeded-up Robust Features (SURF) // Comput. Vis. Image Understand. 2008. V. 110. No. 3. P. 346–359.
18. *Rosten E., Drummond T.* Machine Learning for High-speed Corner Detection // Proc. 9th Eur. Conf. Comput. Vis. (ECCV). Berlin, Germany: Springer-Verlag. 2006. P. 430–443.
19. *Rublee E., Rabaud V., Konolige K., Bradski G.* Orb: An Efficient Alternative to SIFT or SURF // Proc. Int. Conf. Comput. Vis. 2011. P. 2564–2571.
20. *Миллер Б.М., Степанян К.В., Попов А.К., Миллер А.Б.* Навигация БПЛА на основе последовательностей изображений, регистрируемых бортовой видеокамерой // АИТ. 2017. № 12. С. 141–154.
Miller B.M., Stepanyan K.V., Popov A.K., Miller A.B. UAV Navigation based on Videosequences Captured by the Onboard Video Camera // Autom. Remote Control. 2017. V. 78. No. 12. P. 2211–2221.
21. *Goppert J., Yantek S., Hwang I.* Invariant Kalman Filter Application to Optical Flow based Visual Odometry for UAVs // 2017 Ninth Int. Conf. on Ubiquitous and Future Networks (ICUFN). 2017. P. 99–104.
22. *Muller P., Savakis A.* Flowdometry: An Optical Flow and Deep Learning based Approach to Visual Odometry // 2017 IEEE Winter Conf. on Applications of Computer Vision (WACV). 2017. P. 624–631.

23. *Gonzalez R., Rituerto A., Guerrero J.J.* Improving Robot Mobility by Combining Downward-looking and Frontal Cameras // *Robotics*. 2016. V. 5. No. 4. P. 25.
24. *Charrett T.O., Waugh L., Tatam R.P.* Speckle Velocimetry for High Accuracy Odometry for a Mars Exploration Rover // *Measurement Science and Technology*. 2009. V. 21. No. 2. P. 025301.
25. *Мясников В.В., Дмитриев Е.А.* Исследование зависимости точности одновременной реконструкции сцены и позиционирования камеры от погрешностей, вносимых датчиками мобильного устройства // *Компьютерная оптика*. 2019. Т. 43. № 3. С. 492–503.
26. *Fursov V.A., Kotov A.P., Goshin Y.V.* Solution of Overdetermined Systems of Equations Using the Conforming Subsystem Selection // *J. of Physics: Conference Series*. 2019. V. 1368. Iss. 5. P. 821–828.
27. *Fursov V.A., Gavrilov A.V., Goshin Y.V.* Conforming Identification of the Fundamental Matrix in the Image Matching Problem // *Computer Optics*. 2017. V. 41. No. 4. P. 559–563.
28. *Antonini A., Guerra W., Murali V., Sayre-McCord T., Karaman S.* The Blackbird Dataset: A Large-scale Dataset for UAV Perception in Aggressive Flight // *Int. Sympos. on Experimental Robotics*. 2018. P. 130–139.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 20.01.2021

После доработки 26.03.2021

Принята к публикации 30.06.2021

© 2021 г. З.М. ШИБЗУХОВ, д-р физ.-мат. наук (intellimath@mail.ru)
(Институт математики и информатики Московского
педагогического государственного университета;
Московский физико-технический институт)

ОБ ОДНОМ РОБАСТНОМ ПОДХОДЕ К ПОИСКУ ЦЕНТРОВ КЛАСТЕРОВ¹

Предложен новый подход к построению алгоритмов кластеризации k -means, в котором вместо евклидова расстояния используется расстояние Махаланобиса. Подход основан на минимизации дифференцируемых оценок среднего значения, нечувствительных к выбросам. На иллюстративных примерах убедительно показана возможность устойчивости предложенного алгоритма по отношению к большим объемам выбросов в данных.

Ключевые слова: центр кластера, робастное среднее, расстояние Махаланобиса, итеративное перевзвешивание, робастный алгоритм.

DOI: 10.31857/S0005231021100111

1. Введение

Задача кластеризации — одна из классических задач машинного обучения. В основе одного классического подхода для разбиения конечного множества точек на кластеры лежит процедура поиска центров кластеров. Центр кластера — это точка, от которой сумма обобщенных расстояний до всех его точек минимальна. Разбиение на кластеры осуществляется по простому правилу: точка относится к тому кластеру, до центра которого расстояние минимально. Формально это можно выразить следующим образом.

2. Классическая постановка задачи

Пусть $U \subset \mathbb{R}^n$ — открытое множество, $X = \{x_1, \dots, x_N\} \subset U$ — конечное множество, которое требуется разбить на K кластеров. Пусть $d: U \times U \rightarrow \mathbb{R}_+$ — функция обобщенного расстояния между точками из U . Она обладает свойствами, которые достаточны для того, чтобы следующая задача минимизации

$$c^* = \arg \min_{c \in S} \sum_{j=1}^m d(x_j, c)$$

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 18-01-00050).

для поиска центра произвольного кластера $\mathbf{C} \subset \mathbf{U}$ имела решение. Точное формальное определение функции d можно найти в [1].

Если заданы центры кластеров $\mathbf{c}_1, \dots, \mathbf{c}_K$, то произвольная точка \mathbf{x} будет относиться к кластеру с центром \mathbf{c}_t ($1 \leq t \leq K$), если

$$d(\mathbf{x}, \mathbf{c}_t) = \min_{1 \leq j \leq K} d(\mathbf{x}, \mathbf{c}_j).$$

Задачу поиска центров кластеров можно сформулировать как задачу минимизации

$$(1) \quad \mathbf{c}_1^*, \dots, \mathbf{c}_K^* = \arg \min_{\mathbf{c}_1, \dots, \mathbf{c}_K} \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K),$$

где

$$(2) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \sum_{k=1}^N \nu_k \min_{1 \leq j \leq K} d(\mathbf{x}_k, \mathbf{c}_j),$$

величина $\nu_k \geq 0$ соответствует значимости k -й точки, $\nu_1 + \dots + \nu_N = 1$.

3. Постановка задачи с выпуклыми функциями среднего по Колмогорову

В [1] рассматривается более общая постановка задачи, в которой наряду с функцией \min в формуле (2) используются выпуклые функции среднего по Колмогорову:

$$G_{h_s}(x_1, \dots, x_K) = h_s^{-1} \left\{ \frac{1}{K} \sum_{j=1}^K h_s(x_j) \right\},$$

где $h_s: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ — непрерывно-дифференцируемая строго монотонно убывающая неотрицательная биективная функция. При этом

$$\lim_{s \rightarrow \infty} G_{h_s}(x_1, \dots, x_K) = \min_{j=1, \dots, K} x_j.$$

Например:

$$1) \quad G_{e^{-sx}}(x_1, \dots, x_K) = -\frac{1}{s} \ln \left\{ \frac{1}{K} (e^{-sx_1} + \dots + e^{-sx_K}) \right\};$$

$$2) \quad G_{x^{-s}}(x_1, \dots, x_K) = \left(\frac{1}{K} (x_1^{-s} + \dots + x_K^{-s}) \right)^{-1/s}.$$

Целевая функция (2) принимает вид:

$$(3) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \sum_{k=1}^N \nu_k G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)).$$

Градиенты целевой функции \mathcal{Q} принимают вид:

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{c}_j} = \sum_{k=1}^N \nu_k v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) \frac{\partial d(\mathbf{x}_k, \mathbf{c}_j)}{\partial \mathbf{c}_j},$$

где

$$v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \frac{h'_s(d(\mathbf{x}_k, \mathbf{c}_j))}{h'_s(G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)))}.$$

В случае $G_{h_\infty} = \min$

$$v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \begin{cases} 1, & \text{если } d(\mathbf{x}, \mathbf{c}_j) = \min_{t=1, \dots, K} d(\mathbf{x}, \mathbf{c}_t), \\ 0 & \text{иначе.} \end{cases}$$

Алгоритм SKM.

Для поиска центров в [1] предложен алгоритм SKM (*Smooth k-Means*) и при определенных условиях доказана его сходимость к решению задачи минимизации (3).

```

procedure SKM( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t)$ ,
       $\mathbf{c}_j^{t+1} \leftarrow \arg \min_{\mathbf{c}} \sum_{k=1}^N \tilde{\nu}_{jk}^t d(\mathbf{x}_k, \mathbf{c})$ , где  $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1^t, \dots, \mathbf{c}_K^t$ 
end

```

Если d – Евклидово расстояние, а G – \min , то SKM представляет собой алгоритм KMeans. В [1] показано, что SKM обобщает хорошо известные алгоритмы кластеризации, такие как FCM [2], EM [3], DA [4], Bergman Soft Clustering [5].

3.1. Вариант алгоритма SKM для расстояния Махаланобиса

Рассмотрим случай, когда

$$d(\mathbf{x}, \mathbf{c}, \mathbf{S}) = \ln |\mathbf{S}| + (\mathbf{x} - \mathbf{c})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{c})$$

— регуляризованный квадрат расстояния Махаланобиса с ковариационной матрицей \mathbf{S} .² Поэтому кроме центров кластеров необходимо искать ковариационные матрицы, ассоциированные с ними. Целевая функция записывается

$$^2 d(\mathbf{x}, \mathbf{c}, \mathbf{S}) = -\ln(|\mathbf{S}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{c})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{c})\}).$$

в виде

$$(4) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K) = \sum_{k=1}^N \nu_k G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1, \mathbf{S}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K, \mathbf{S}_K)).$$

Чтобы избежать вырождения, добавим дополнительное условие $|\mathbf{S}| = 1$. При фиксированных значениях $\mathbf{S}_1, \dots, \mathbf{S}_K$ центры находятся при помощи алгоритма SKM вида:

```

procedure SKM_M1( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}; \{\mathbf{S}_1, \dots, \mathbf{S}_K\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t; \mathbf{S}_1, \dots, \mathbf{S}_K)$ ,
       $\mathbf{c}_j^{t+1} \leftarrow \sum_{k=1}^N \tilde{\nu}_{jk}^t \mathbf{x}_k$ , где  $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1, \dots, \mathbf{c}_K$ 
end

```

При фиксированных значениях $\mathbf{c}_1, \dots, \mathbf{c}_K$ ковариационные матрицы можно найти при помощи следующего алгоритма:

```

procedure SKM_M2( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}; \{\mathbf{c}_1, \dots, \mathbf{c}_K\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1^t, \dots, \mathbf{S}_K^t)$ ,
       $\mathbf{S}_j^{t+1} \leftarrow \sum_{k=1}^N \tilde{\nu}_{jk}^t (\mathbf{x}_k - \mathbf{c}_j)' (\mathbf{x}_k - \mathbf{c}_j)$ , где
       $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1^t, \dots, \mathbf{c}_K^t$ 
end

```

Для поиска центров и ковариационных матриц применяется метод альтернативных направлений:

```

procedure SKM_M( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}$ )
   $s \leftarrow 0$ 
  repeat
     $\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1} \leftarrow \text{SKM\_M1}(\mathbf{X}, \{\nu_1, \dots, \nu_N\}, \{\mathbf{c}_1^s, \dots, \mathbf{c}_K^s\}; \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $\mathbf{S}_1^{s+1}, \dots, \mathbf{S}_K^{s+1} \leftarrow \text{SKM\_M2}(\mathbf{X}, \{\nu_1, \dots, \nu_N\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\}; \{\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1}\})$ 
     $s \leftarrow s + 1$ 
  until значения целевой функции не стабилизируются
  return  $(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s), (\mathbf{S}_1^s, \dots, \mathbf{S}_K^s)$ 
end

```

4. Проблема выбросов

Алгоритм SKM_MM хорошо справляется с задачей поиска центров кластеров и ковариационных матриц расстояний Махаланобиса, ассоциированных с ними, до тех пор, пока в эмпирическом распределении значений $\{D_1(\mathbf{x}_1), \dots, D_N(\mathbf{x}_N)\}$, где $D_k(\mathbf{x}) = G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1, \mathbf{S}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K, \mathbf{S}_K))$, не появятся *выбросы*.

К выбросам можно отнести те данные, которые совершенно (или в достаточно большой степени) не соответствуют модели распределения точек. Их присутствие приводит к искажению положений центров кластеров.

Можно указать две основные причины появления выбросов:

- часть данных (<50 %) содержит существенные ошибки или неустранимые искажения;
- часть данных (<50 %) подчиняется другому закону распределения, основная часть (>50 %).

В таких случаях среднее взвешенное среднее арифметическое может сохранять свою устойчивость по отношению к выбросам, если веса точек ν_k , соответствующих выбросам, достаточно малы, чтобы предотвратить их влияние на величину оценки эмпирического среднего. Но на практике изначально веса примеров, как правило, неизвестны, а поиск значений весов примеров, способных подавить влияние выбросов, является задачей, сопоставимой по трудности с задачей идентификации самих выбросов. Поэтому естественный выход из такой ситуации — это использование оценок среднего значения, которые нечувствительны к выбросам. Например, медиана, квантили, усеченные средние арифметические, винзоризированные средние. Но проблема состоит в том, что частные производные таких оценок среднего значения имеют сингулярные свойства, что создает неустранимое препятствие для применения градиентных алгоритмов минимизации, которые используются при построении алгоритмов SKM, SKM_M1, SKM_M2 и SKM_M.

В данной статье для преодоления *проблемы выбросов* предлагается расширить принцип разбиения на кластеры. Для этого в определении (3) целевой

функции \mathcal{Q} будем использовать непрерывно-дифференцируемые усредняющие агрегирующие функции, устойчивые к выбросам, вместо взвешенного среднего арифметического. Теперь

$$(5) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = M\{D(\mathbf{x}_1; \mathbf{c}_1, \dots, \mathbf{c}_N), \dots, D(\mathbf{x}_N; \mathbf{c}_1, \dots, \mathbf{c}_N)\},$$

где $M\{z_1, \dots, z_N\}$ — дифференцируемая усредняющая агрегирующая функция, нечувствительная к определенной доле выбросов в $\{z_1, \dots, z_N\}$,

$$D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N) = \min_{1 \leq j \leq K} d(\mathbf{x}_k, \mathbf{c}_j)$$

или

$$D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N) = G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)).$$

Это позволяет расширить описанный выше подход для построения робастных алгоритмов кластеризации.

Для случая расстояний Махаланобиса

$$\begin{aligned} \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K) &= \\ &= M\{D(\mathbf{x}_1; \mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K), \dots, D(\mathbf{x}_N; \mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K)\}. \end{aligned}$$

Если M — среднее арифметическое, то получим задачу, эквивалентную задаче минимизации (2) и (3).

Далее рассмотрим некоторые способы построения дифференцируемых усредняющих агрегирующих функций, нечувствительных к определенной доле выбросов.

4.1. Оценки среднего, нечувствительные к выбросам

Оценки среднего, нечувствительные к выбросам, можно построить несколькими способами.

Первый способ основан на приближении медианы на базе M -средних [6, 7]. Определим M -среднее как решение следующей задачи [8]:

$$M_\rho\{r_1, \dots, r_m\} = \arg \min_s \sum_{j=1}^m \rho(r_j - s).$$

Если $\rho(r)$ — строго выпуклая функция и $\rho(0) = 0$, то M_ρ — усредняющая агрегирующая функция [9, 10].

Если существует $\rho''(r)$, то

$$\frac{\partial M_\rho}{\partial r_j} = \frac{\rho''(r_j - \bar{r})}{\rho''(r_1 - \bar{r}) + \dots + \rho''(r_m - \bar{r})},$$

где $\bar{r} = M_\rho\{r_1, \dots, r_m\}$.

Второй способ основан на применении цензурированного среднего арифметического, в котором пороговое значение оценивается при помощи сглаженного варианта α -квантиля:

$$(6) \quad \text{WM}_{\rho, \alpha}\{z_1, \dots, z_N\} = \frac{1}{N} \sum_{k=1}^N \min\{z_k, \bar{z}_{\rho, \alpha}\},$$

где $\rho_\alpha(r)$ – функция, при которой M_{ρ_α} выступает в качестве приближения медианы,

$$(7) \quad \rho_\alpha(r) = \begin{cases} \alpha\rho(r), & \text{если } r > 0, \\ \frac{1}{2}(\alpha\rho(0_+) + (1 - \alpha)\rho(0_+)), & \text{если } r = 0, \\ (1 - \alpha)\rho(r), & \text{если } r < 0, \end{cases}$$

является функцией, для которой $M_{\rho_\alpha}\{z_1, \dots, z_N\}$ выступает в качестве приближения α -квантиля.

Частные производные имеют вид

$$\frac{\partial \text{WM}_{\rho_\alpha}}{\partial z_k} = \left(\frac{1}{N} [z_k < \bar{z}_{\rho_\alpha}] + \frac{m}{N} \frac{\partial M_{\rho_\alpha}}{\partial z_k} \right),$$

где m – число значений $z_k \geq \bar{z}_{\rho_\alpha}$. В обоих случаях $\frac{\partial M}{\partial z_k} \geq 0$ и $\frac{\partial M}{\partial z_1} + \dots + \frac{\partial M}{\partial z_N} = 1$.

Алгоритм SRKM.

Рассмотрим обобщение путем замены взвешенного среднего арифметического в (2) и (3) на дифференцируемые оценки среднего M , определенные выше.

Градиент \mathcal{Q} по \mathbf{c}_j ($1 \leq j \leq K$) имеет вид

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{c}_j} = \sum_{k=1}^N \nu_k(\mathbf{c}_1, \dots, \mathbf{c}_N) v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_N) \frac{\partial d}{\partial \mathbf{c}_j},$$

где

$$\nu_k(\mathbf{c}_1, \dots, \mathbf{c}_N) = \frac{\partial M_\rho\{r_1, \dots, r_N\}}{\partial r_k}, \quad r_k = D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N).$$

Для минимизации (5) запишем алгоритм SRKM (*Smooth Robust k-Means*) – обобщение алгоритма SKM.

```

procedure SRKM( $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ )
   $t \leftarrow 0$ 
  repeat
     $\nu_k^t = \nu_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t)$ 
     $\mathbf{c}_1^{t+1}, \dots, \mathbf{c}_K^{t+1} \leftarrow \text{SKM}(\mathbf{X}, \{\nu_1^t, \dots, \nu_N^t\}, \{\mathbf{c}_1^t, \dots, \mathbf{c}_K^t\})$ 
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1^t, \dots, \mathbf{c}_K^t$ 
end

```

Алгоритм SRKM можно рассматривать как робастный вариант алгоритма SKM.

В случае применения расстояния Махаланобиса получим вариант SRKM с применением метода альтернативных направлений для поиска центров и ковариационных матриц.

```

procedure SRKM_M( $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}$ )
   $s \leftarrow 0$ 
  repeat
     $\nu_k^s = \nu_k(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s)$ 
     $\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1} \leftarrow \text{SKM\_M1}(\mathbf{X}, \{\nu_1^s, \dots, \nu_N^s\}, \{\mathbf{c}_1^s, \dots, \mathbf{c}_K^s\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $\mathbf{S}_1^{s+1}, \dots, \mathbf{S}_K^{s+1} \leftarrow \text{SKM\_M2}(\mathbf{X}, \{\nu_1^s, \dots, \nu_N^s\}, \{\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1}\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $s \leftarrow s + 1$ 
  until значения целевой функции не стабилизируются
  return  $(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s), (\mathbf{S}_1^s, \dots, \mathbf{S}_K^s)$ 
end

```

Нетрудно видеть, что алгоритм SRKM отличается от SKM тем, что на каждом шаге s вычисляются веса точек ν_1^s, \dots, ν_N^s как частные производные дифференцируемой робастной оценки среднего значения от минимальных расстояний от точек до ближайшего кластера, вычисляемых при помощи \min или G_{h_s} .

Алгоритм SRKM_M является робастным вариантом SKM_M, которые представляют собой варианты алгоритмов SRKM и SKM соответственно, когда используется расстояние Махаланобиса $d(\mathbf{x}, \mathbf{c}, \mathbf{S})$, в котором учитывается ковариационная матрица, с помощью которой может учитываться эллипсоидальная форма распределения точек в кластере. В прилагаемых далее иллюстративных примерах применяется расстояние Махаланобиса.

5. Иллюстративные примеры

Для иллюстрации возможностей алгоритма SRKM-M рассмотрим ряд примеров. Они наглядно демонстрируют его способность находить центры, которые лежат достаточно близко к настоящим центрам кластеров в условиях, когда данные содержат выбросы или когда ищутся центры не всех кластеров, а лишь некоторой части из них. Если в первом случае причиной смещения найденных центров являются выбросы, то во втором случае причиной смещения являются избыточные кластеры.

Пример 1. В этом примере сравнивается применение алгоритмов SKM_M и SRKM_M с M-средним $M_{\rho_{\alpha, \varepsilon}}$, где $\rho_{\varepsilon}(r) = \sqrt{\varepsilon^2 + r^2}$, $\varepsilon = 10^{-3}$, для поиска центров трех кластеров. Здесь расстояние d – Евклидово. Точки искусственно сгенерированного набора данных принадлежат трем кластерам с несимметричным распределением расстояний точек кластера от своего центра, которое имеет эллипсоидную форму с центром в одном из фокусов, плотность распределения убывает по мере удаления от фокуса. Все кластеры содержат по

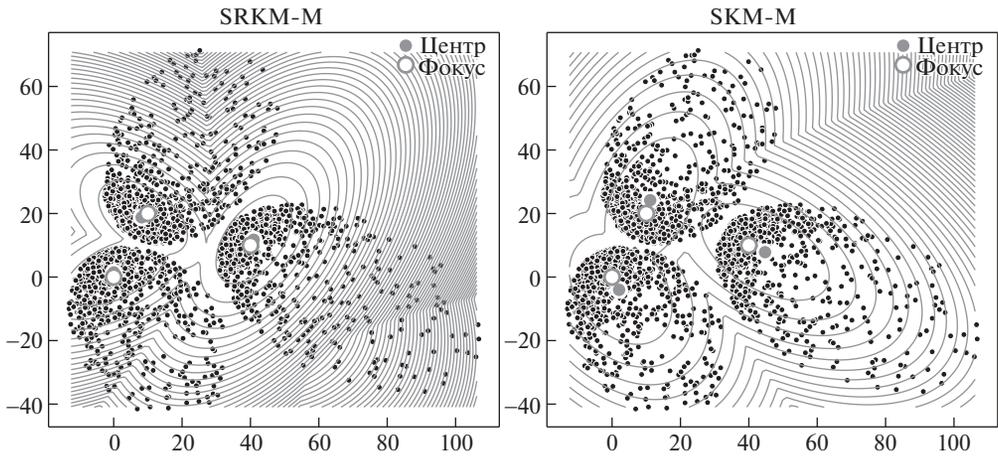


Рис. 1. Применение алгоритмов SKM_M и SRKM-M в примере 1.

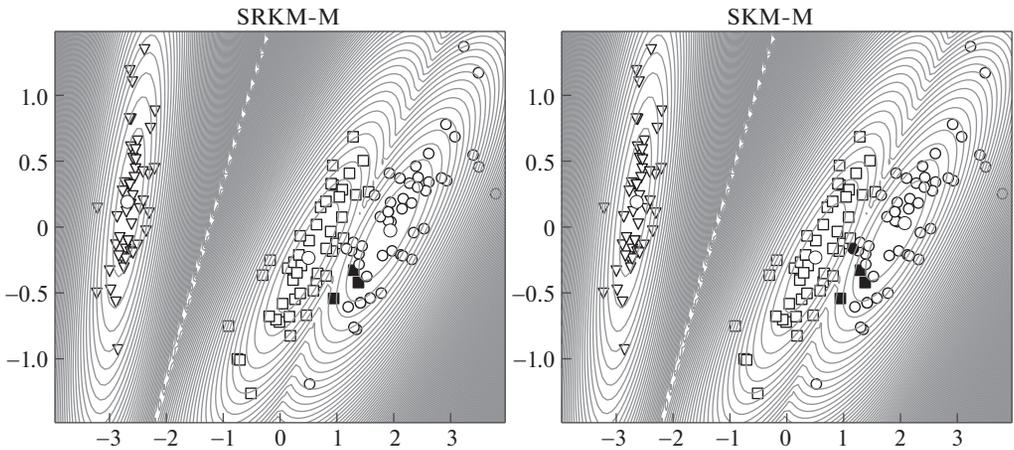


Рис. 2. Применение алгоритма SRKM_M и SKM_M в примере 2.

1000 точек. Метод SKM_M выдает центры кластеров, смещенные в направлении противоположного фокуса. Робастный алгоритм SRKM_M с $\alpha = 0,4$ позволяет найти центры, которые расположены вблизи фокусов, в то время как алгоритм SKM_M находит центры, которые более удалены от настоящих фокусов эллипсов. Это достигается за счет того, что основная масса точек сосредоточена около фокусов эллипсов. Поиск фокусов эллипсов можно искать по той части точек, которые как раз находятся около фокусов. Робастный алгоритм обеспечивает возможность нахождения центра множества точек, расположенных вблизи фокусов, не испытывая существенного влияния со стороны более удаленных от фокусов точек. Результаты представлены на рис. 1.

Пример 2. Рассмотрим классический набор данных *iris*. Как правило, он используется для задач классификации. Попытаемся идентифицировать

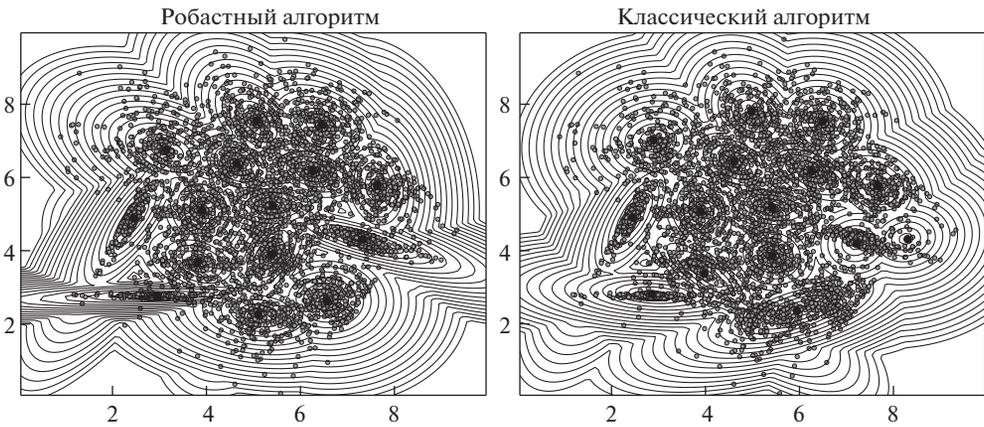


Рис. 3. S4: Результаты алгоритмов SRKM_M и SKM_M из примера 3.

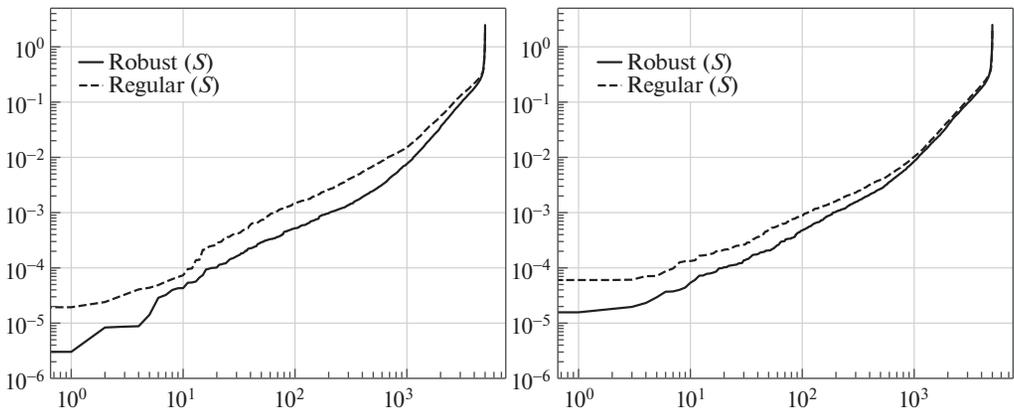


Рис. 4. S4: Распределение расстояний до центра ближайшего кластера для алгоритмов SRKM_M (слева) и SKM_M (справа) из примера 3.

классы при помощи кластеризации, используя расстояния Махаланобиса вместо Евклидового. На рис. 1 представлен результат кластеризации при помощи алгоритмов SRKM_M и SKM_M. При помощи алгоритма SRKM_M можно получить разбиение, которое отличается от заданного в 3 точки из 150. При помощи алгоритма SKM_M можно получить разбиение, которое отличается от заданного в 4 точках из 150. Хотя преимущество минимальное, но с учетом того что наилучшие надежные алгоритмы классификации по данным набора *iris* как раз дают 98 % точности и выше, то это можно считать хорошим результатом. Он показывает, что с применением предложенного здесь подхода на основе реалистичного набора признаков можно получать разбиения, которые практически соответствуют естественной классификации. Результаты представлены на рис. 2.

Пример 3. Рассмотрим наборы данных S3–S4 из [11, 12]. Они содержат 5000 точек, 15 кластеров. Среди наборов данных S1–S4 [12] в S4 наблюдается

наибольшая вариация в распределении точек на плоскости, что затрудняет поиск центров кластеров. Именно поэтому он представляет наибольший интерес. Для сравнения приведем результаты применения алгоритмов SRKM_M и SKM_M. На рис. 3 представлены результаты кластеризации для S4. Рисунок 4 иллюстрирует зазоры между распределением расстояний от точек до ближайшего к ним кластера. Применение робастного варианта алгоритмов показывает уменьшение значений расстояний. Он также наглядно иллюстрирует увеличение этого зазора при применении алгоритма SRKM_M.

6. Заключение

Применение дифференцируемых оценок среднего значения, нечувствительных к выбросам, позволило построить новые робастные процедуры поиска центров кластеров SRKM и SRKM_M, которые обобщают SKM и SKM_M, позволяя использовать широкий спектр методов поиска среднего значения как для четкого, так и для нечеткого отнесения точек к кластерам. Предложенный метод и алгоритмы SRKM и SRKM_M наследуют структуру алгоритмов SKM и SKM_M, однако отличаются методом расчета весов точек. Особенность предложенного подхода состоит в том, что веса точек вычисляются как значения частных производных дифференцируемой робастной оценки среднего значения от минимальных расстояний от точек до ближайшего кластера, вычисляемых при помощи \min или G_{h_s} . В результате веса точек убывают с ростом модуля разности между соответствующим аргументом робастной усредняющей агрегирующей функции и величиной среднего значения. Наибольшие значения весов соответствуют точкам, которые соответствуют аргументам усредняющей агрегирующей функции, которые находятся вблизи величины среднего значения. Это объясняет, почему предложенный подход способен преодолевать влияние выбросов. Приведенные иллюстративные примеры убедительно показывают устойчивость предложенных алгоритмов по отношению к относительно большим объемам выбросов.

СПИСОК ЛИТЕРАТУРЫ

1. *Teboulle M.* A Unified Continuous Optimization Framework for Center-Based Clustering Method // J. Machine Learning Research. 2007. No. 8. P. 65–102.
2. *Bezdek J.C.* Pattern Recognition with Fuzzy Objective Function Algorithms. N.Y.: Plenum Press, 1981.
3. *Duda R.O., Hart P.E., Stork D.G.* Pattern Classification. John Wiley & Sons, Inc., 2-nd edition. 2001.
4. *Rose K.A., Gurewitz E., Fox C.G.* Deterministic annealing approach to clustering // Pattern Recognition Letters. 1990. V. 11. No. 9. P. 589–594.
5. *Banerjee A., Merugu S., Dhillon I.S., Ghosh J.* Clustering with Bregman Divergences // J. Machine Learning Research. 2005. No. 6. P. 1705–1749.
6. *Mesiar R., Komornikova M., Kolesarova A., Calvo T.* Aggregation functions: A revision / H. Bustince, F. Herrera, J. Montero, eds., Fuzzy Sets and Their Extensions: Representation, Aggregation and Models. Berlin–Heidelberg: Springer, 2008.

7. *Grabich M., Marichal J.-L., Pap E.* Aggregation Functions. Series: Encyclopedia of Mathematics and its Applications. No. 127. Cambridge University Press, 2009.
8. *Шибзухов З.М.* О принципе минимизации эмпирического риска на основе усредняющих агрегирующих функций // Докл. РАН. 2017. Т. 476. № 5. С. 495–499.
9. *Calvo T., Beliakov G.* Aggregation functions based on penalties // Fuzzy Sets and Systems. 2010. V. 161. No. 10. P. 1420–1436.
10. *Beliakov G., Sola H., Calvo T.* Practical Guide to Averaging Functions. Springer, 2016.
11. *Franti P., Sieranoja S.* K-means Properties on Six Clustering Benchmark Datasets // Applied Intelligence. 2018. V. 48. No. 12. P. 4743–4759.
12. Clustering Basic Benchmark. <http://cs.joensuu.fi/sipu/datasets/>

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 26.04.2021

Принята к публикации 30.06.2021

© 2021 г. А.В. ЯМАЕВ (rewin1996@gmail.com)
(МГУ им. М.В. Ломоносова; ООО Смарт Энджинс Сервис, Москва),
М.В. ЧУКАЛИНА, канд. физ.-мат. наук (m.chukalina@smartengines.com)
(Федеральный научно-исследовательский центр
“Кристаллография и фотоника” РАН, Москва;
ООО Смарт Энджинс Сервис, Москва),
Д.П. НИКОЛАЕВ, канд. физ.-мат. наук (d.p.nikolaev@smartengines.com)
(Институт проблем передачи информации РАН, Москва;
ООО Смарт Энджинс Сервис, Москва),
А.В. ШЕШКУС (asheshkus@smartengines.com)
(Федеральный исследовательский центр
“Информатика и управление” РАН, Москва;
ООО Смарт Энджинс Сервис, Москва),
А.И. ЧУЛИЧКОВ, д-р физ.-мат. наук (achulichkov@gmail.com)
(МГУ им. М.В. Ломоносова)

НЕЙРОННАЯ СЕТЬ ДЛЯ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ В КОМПЬЮТЕРНОЙ ТОМОГРАФИИ¹

Предложена легковесная шумоподавляющая фильтрующая нейронная сеть, которая реализует этап фильтрации в алгоритме томографической реконструкции свертки и обратного проецирования (FBR). Приведено обоснование нейросетевой архитектуры, выбранной на основе возможности аппроксимации с достаточной точностью операции рамп-фильтрации. Работоспособность сети продемонстрирована на синтетических данных, которые имитируют томографические проекции, зарегистрированные с малой экспозицией. При генерации синтетических данных учитывались квантовая природа рентгеновского излучения, время экспозиции одного кадра и нелинейный отклик детектора ионизирующего излучения. Время выполнения реконструкции с использованием предложенной сети в 11 раз меньше по сравнению с выбранными для сравнения тяжелыми сетями, при качестве реконструкции по метрике *SSIM* выше 0,9.

Ключевые слова: низкодозовая компьютерная томография, нейронные сети, UNet, быстрые вычисления.

DOI: 10.31857/S0005231021100123

1. Введение

Рентгеновская компьютерная томография (КТ) является широко используемым неразрушающим методом восстановления внутренней структуры объекта. В медицине при использовании КТ стремятся минимизировать получаемую пациентом дозу рентгеновского излучения, но чтобы по результату реконструкции врач мог определить патологии и поставить диагноз.

¹ Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проекты № 19-01-00790 и № 18-29-26017).

Уменьшают получаемую дозу, к примеру, следующими двумя способами: уменьшением числа измеренных изображений [1, 2] и уменьшением времени экспозиции.

Уменьшение времени экспозиции ведет к увеличению шума на зарегистрированных данных. Как следствие, это приводит к увеличению шума и на реконструкции. Для его подавления используют алгоритмы следующих классов: на основе статистического анализа [3], нелинейные фильтры [3], итеративные оптимизационные алгоритмы [4] и нейронные сети. Нейронные сети при зашумленных данных могут выступать в роли операции шумоподавления после реконструкции (постпроцессинг), в роли операции шумоподавления на синограмме (препроцессинг) [5] и в роли полноценного оператора реконструкции [6]. Для решения этих задач используют различные нейросетевые модели, к примеру, сверточные нейронные сети, нейронные сети, работающие в вейвлет пространстве (шумоподавление постпроцессингом), а также множество модификаций [7–10] и нейросети Хопфилда (постпроцессинг [11]), сети, работающие как в пространстве реконструкций, так и в пространстве измеренных данных, генеративные нейронные сети (шумоподавление постпроцессингом [12]).

Несмотря на большое разнообразие различных подходов к решению задачи низкодозовой компьютерной томографии, существует мало представителей быстрых нейросетевых методов. Использование быстрых алгоритмов дает возможность производить реконструкцию во время проведения исследования и, следовательно, останавливать исследование в процессе его проведения в момент достижения достаточного качества реконструкции для понижения полученной пациентом дозы [13].

В статье предложен подход, при котором нейронная сеть выполняет операцию фильтрации данных перед этапом обратного проецирования в алгоритме FBP. Таким образом, нейронная сеть выступает не только в качестве фильтра перед обратным проецированием, но и в качестве шумоподавляющего алгоритма. В статье показано, что фильтр в алгоритме FBP можно заменить одним сверточным слоем сети. Нейронная сеть, выполняющая функцию фильтрации данных, позволяет уменьшить объем вычислений и повысить точность реконструкции. Для обучения нейронной сети использовалась функция потерь в виде суммы по обучающей выборке $\{x_i, y_i\}$

$$(1) \quad Loss = - \sum_i SSIM(y_i, BP(F(x_i, v))),$$

где F – функция нейронной сети, x_i – зашумленная синограмма, подающаяся на вход нейронной сети и полученная из проекций с малыми временами экспозиций, v – параметры и веса нейронной сети, BP – операция обратного проецирования, y_i – фантом, послуживший основой для создания синограммы x_i , $SSIM$ (Structure Similarity) – метрика структурного подобия [14]. Метрика $SSIM$ была выбрана по причине того, что проведенные эксперименты показали значительное увеличение качества результатов работы сети после обучения по этой метрике, чем при обучении по метрике L2. Архитек-

тура предложенной нейронной сети основана на модификации архитектуры UNet [15] и свойствах FBP алгоритма и описана в разделе 4. Таким образом, в настоящей статье предлагается обоснованная архитектура нейронной сети, используется специально подобранная под задачу функция потерь и обсуждаются полученные в ходе работы результаты, которые сравниваются с другими обученными нейросетевыми моделями на синтетических данных.

Обобщая изложенное, выделим основную новизну предлагаемого подхода. Во-первых, использование одномерной нейронной сети приводит к независимости алгоритма от угловой схемы измерения, что аналитически невозможно для обычных двумерных сетей по типу UNet. Во-вторых, использование *SSIM* функции потерь. Использование этой функции потерь повышает метрику PSNR и *SSIM* в финале обучения по сравнению с использованием среднеквадратичной функции потерь.

Статья организована следующим образом: в разделе 2 приводится обоснование минимальных требований к фильтрующей синограмме нейросетевой архитектуре, в разделе 3 описывается процесс генерации синтетических данных, используемых для обучения и проверки нейросетевых моделей, в разделе 4 приводится предлагаемая в этой статье нейросетевая архитектура, в разделе 5 описываются и объясняются результаты сравнительных экспериментов, заключение находится в разделе 6.

2. Нейросетевая аппроксимация рамп-фильтра

Каждое измеренное изображение представляется как результат операции Радона над некоторым неизвестным объемом плотностей объекта при некотором заданном угле. Операция Радона – это интегральная сумма вдоль заданного луча. Для двумерной реконструкции, рассматриваемой в этой статье, координаты луча задаются полярной системой координат (r, ϕ) . И соответственно измеренные изображения являются одномерными и зависят только от координаты r . Такие измеренные изображения называются проекциями. Набор проекций же называется синограммой. Для восстановления неизвестного объема плотностей широко используют алгоритм свертки и обратного проецирования (FBP, Filtered Back Projection). Он состоит из двух шагов: (Filter) фильтрация проекций и (Back Projection) обратное проецирование, представимое в виде транспонированного оператора Радона. FBP алгоритм можно представить в операторном виде

$$(2) \quad FBP(Sin) = BP(FT^{-1}(h(w) * FT(Sin))),$$

где BP – обратное проецирование, Sin – синограмма, $h(w)$ – рамп-фильтр, FT – одномерное фурье преобразование вдоль r . При этом рамп-фильтр имеет вид в частотной области

$$(3) \quad h(w) = \frac{|w|}{W},$$

где w – частота, W – нормировочный коэффициент, обычно принимаемый вид максимальной частоты дискретного преобразования Фурье, примененно-

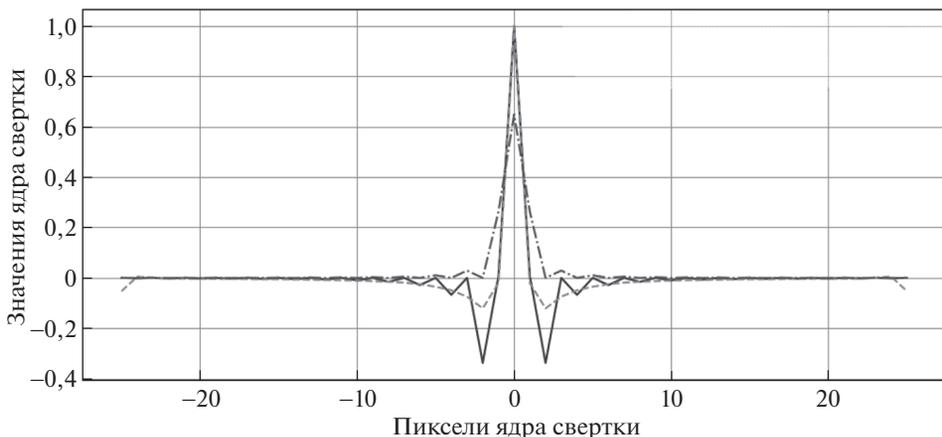


Рис. 1. Ядра свертки рамп-фильтра из приближения и нейросетевого обучения.

го к исходным данным синограммы. Основные особенности этого алгоритма – простота реализации и скорость работы.

Следуя операторному виду FBP , $FT^{-1}(h(w) * FT(Sin))$ представимо в виде операции свертки

$$(4) \quad (H * Sin)(x),$$

где $H(x)$ – представление рамп-фильтра в пространстве синограммы имеет интегральный вид

$$(5) \quad H(x) = \int_{-\infty}^{\infty} h(w)e^{ixw} dw = \int_{-\infty}^{\infty} \frac{|w|e^{ixw}}{W} dw.$$

Поскольку этот интеграл расходящийся, то используют различные его приближения [16]. Одним из них является формула

$$(6) \quad H(x) = W^2 \frac{1 + \cos(Wx)}{\pi^2 - x^2 W^2}.$$

На рис. 1 в соответствии с (6) сплошной линией изображено приближенное ядро свертки, штриховой линией изображено обученное ядро свертки и штрихпунктирной линией – обратное Фурье преобразование рамп-фильтра.

Однако FBP является точным решением только для случая непрерывной незашумленной синограммы [17]. Реальные же данные зашумлены и имеют дискретный вид, что делает нецелесообразным использование FBP для получения высокой точности реконструкций. Однако также следуя изложенному, реконструирующий алгоритм должен иметь возможность с определенно заданной точностью повторить рамп-фильтр. Исследуя свойства стандартных нейросетевых слоев, авторы настоящей статьи пришли к выводу, что рамп-фильтр локализован и с некоторой точностью может быть воспроизведен

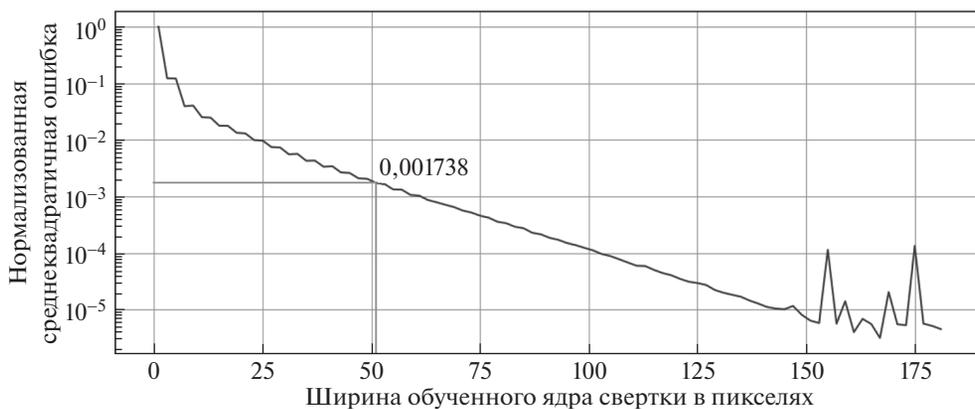


Рис. 2. Нормированный график зависимости среднеквадратичной ошибки обученной линейной одномерной свертки от ширины ядра свертки.

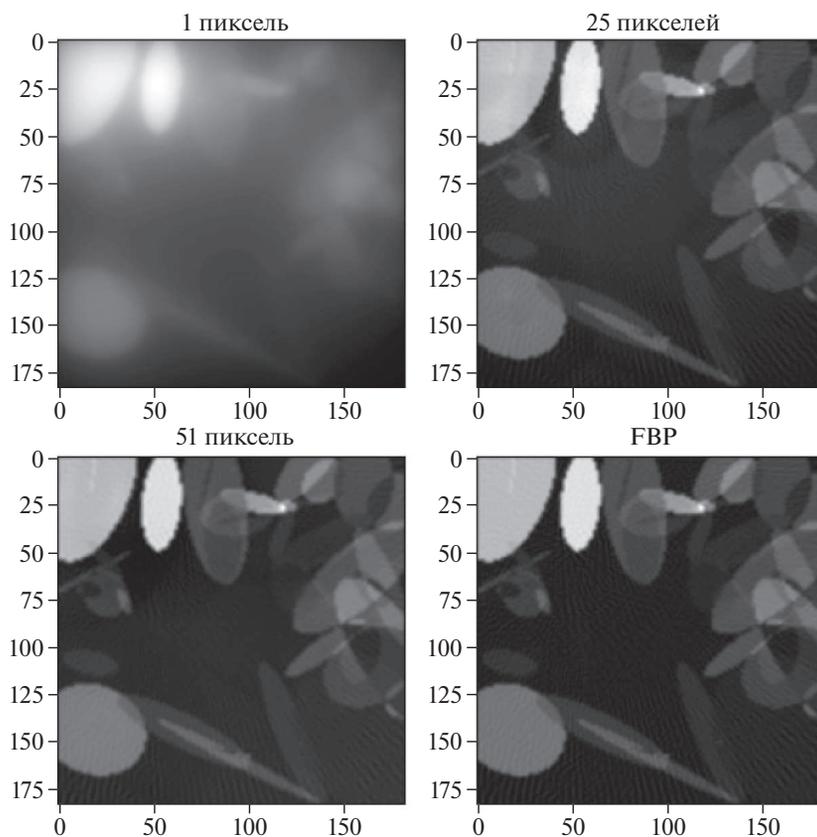


Рис. 3. Реконструкции, полученные при фильтрации синограмм обученными свертками различного размера.

обученной сверткой меньшего размера. Обучая свертки различного размера, от одного пикселя детектора до полного покрытия синограммы, была получена зависимость среднеквадратичной ошибки повторения сверткой рампы-фильтра от ширины ядра используемой свертки. График этой зависимости приведен на рис. 2. Данные для обучения состояли из синограмм и фильтрованных синограмм, построенных от фантомов из раздела 3 и фильтрованных рампы функцией наборов псевдо-дельта функций.

Визуально пропало различие в реконструкциях примерно при свертке размером в 51 пиксель детектора. На основе данных рис. 2 и 3 было решено считать рецептивное поле в 50–51 пиксель проекции достаточным для аппроксимации рампы-фильтра. Весы обученной свертки в выбранной точке показаны на рис. 1. Из этого следует, что нейронные сети со сверточной архитектурой, покрывающей рецептивным полем не менее 51 пикселя проекции, могут быть использованы как фильтрующие нейронные сети.

3. Генерация данных

Для генерации обучающих данных использовался пакет Adler [18]. Пакет позволяет моделировать двумерные изображения наложенных друг на друга эллипсов. Были выбраны только изображения с неотрицательными яркостями.

Синограммы строились с помощью пакетов ODL [18] и Astra Toolbox [19] и зашумлялись на основе квантовых свойств излучения. Синограммы имели ширину в 183 пикселей детектора и 128 углов, распределенных от 0° до 180° градусов (не включая 180°). Зашумленные синограммы использовались как входные данные нейронной сети.

Процедура зашумления происходила по следующей схеме. Выбиралась интенсивность $maxI$ и вычислялась дисперсия аддитивного шума $sigma = maxI/100\,000$. На основе квантовой природы рентгеновского излучения и его пуассоновского поведения рассчитывается зашумленная синограмма по формулам

$$(7) \quad S_e = maxI * e^{-S},$$

где S – незашумленная синограмма,

$$(8) \quad S'_e = P(S_e) + N(0, sigma),$$

где P и N – пуассоновские и нормальные случайные распределения соответственно,

$$(9) \quad I'_0 = P(maxI) + N(0, sigma),$$

$$(10) \quad S' = \log I'_0 - \log S'_e,$$

где S' – синограмма, искаженная случайным шумом. Зашумленная и незашумленная синограммы и их реконструкции показаны на рис. 4. Всего было набрано 20 000 тренировочных синограмм, 5% из которых были выделены под валидацию. Все используемые в статье нейронные сети обучались на одних и тех же тренировочных данных.

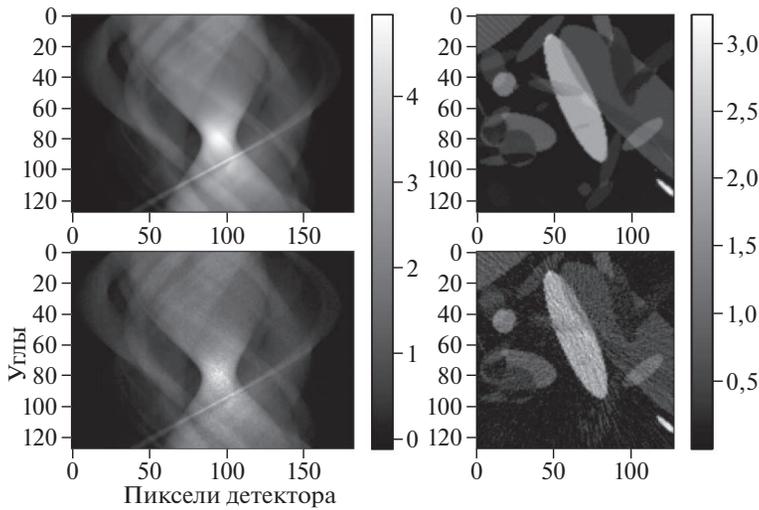


Рис. 4. Примеры данных, используемых в обучении. Слева – незашумленная и зашумленная синограммы, справа – результаты реконструкции.

4. Предлагаемый подход

В медицине требуется быстро обрабатывать данные, что мотивирует использовать одномерные модели вместо двумерных, так как они требуют меньшего объема вычислений при увеличении размерности, чем большинство двумерных моделей. В настоящей статье была разработана одномерная нейронная сеть, принимающая на вход строки синограммы.

Структура модели была основана на UNet-архитектуре, в которой все двумерные свертки заменены на одномерные. При этом количество слоев UNet-архитектуры было подобрано так, чтобы рецептивное поле нейронной сети составляло примерно, но не меньше, 50 пикселей детектора. Как было показано в разделе 2, такого по ширине рецептивного поля достаточно для выполнения операции рамп-фильтрации. Архитектура сети показана на рис. 5. UNet часто применяется для задач сегментации, шумоподавления или преобразования изображения. Более того, для UNet-архитектур характерны высокая обучаемость на небольшом количестве данных и высокая скорость работы. На основе этих характеристик UNet в одномерном виде была выбрана как целевая архитектура.

5. Результаты

Основываясь на том, что рамп-фильтр можно с достаточной точностью аппроксимировать локализованной одномерной сверткой, в статье вместо фильтра в ФВР для реконструкции зашумленных данных применялись легковесные нейронные сети.

В качестве исследуемых моделей были взяты: линейная одномерная свертка (рис. 6), трехслойная сверточная одномерная нейронная сеть (рис. 7), UNet-подобная одномерная архитектура (рис. 5). Эти сети сравниваются по

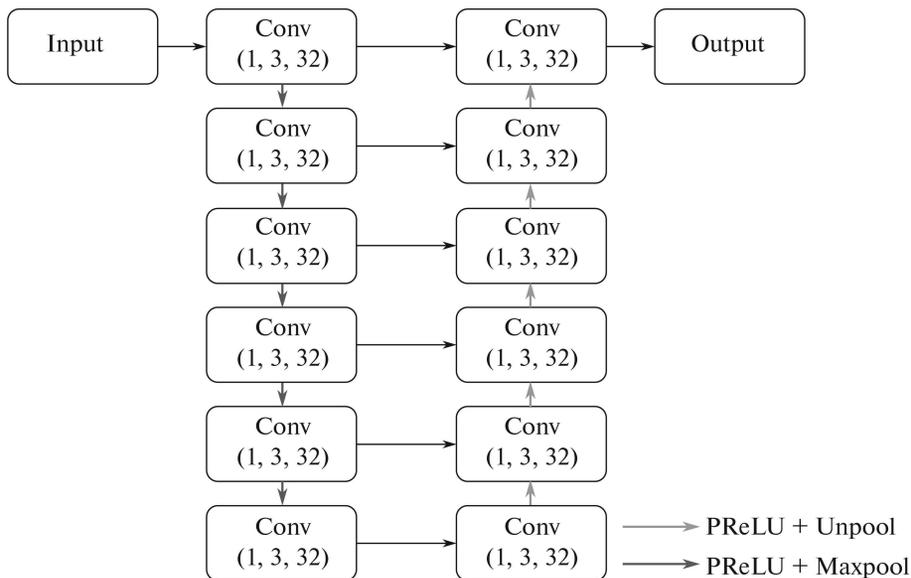


Рис. 5. Изображение архитектуры предложенного одномерного UNet, где в обозначение к Conv первое число – это высота по Y ядра свертки, второе число – это ширина по X ядра свертки, третье число – количество каналов выхода слоя.

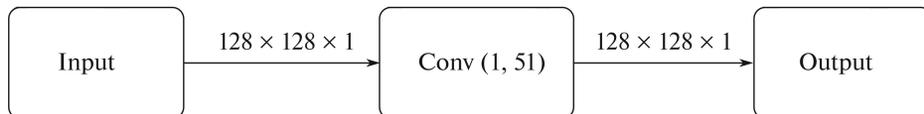


Рис. 6. Изображение архитектуры сети из одной линейной одномерной свертки.

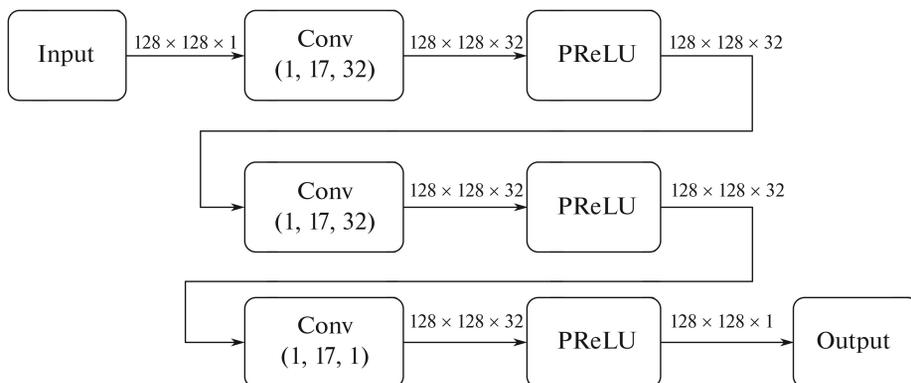


Рис. 7. Изображение архитектуры трехслойной нейронной сверточной одномерной сети.

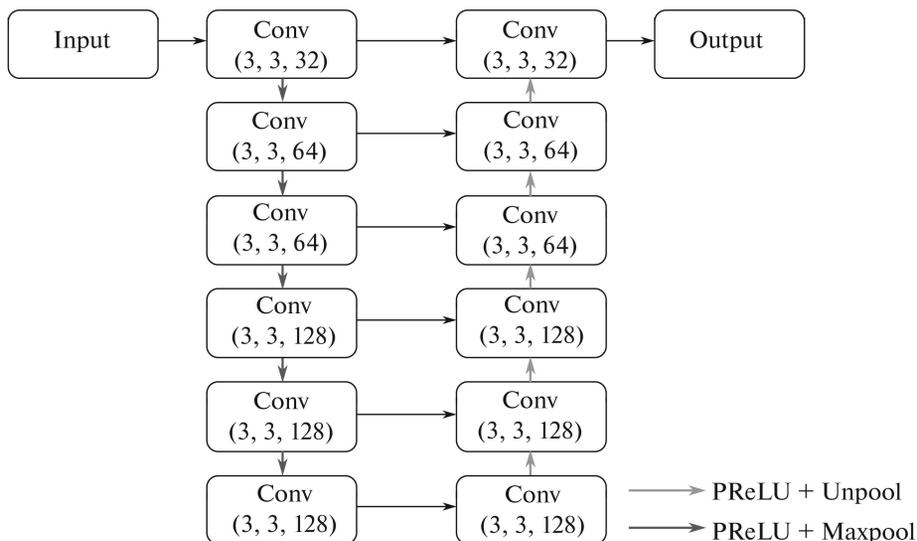


Рис. 8. Изображение архитектуры UNet.

Таблица 1. Таблица с результатами различных архитектур

Нейронная сеть	FBP	Линейная свертка	Трехслойная нейронная сеть	UNet 1D	UNet
Количество умножений на одну синограмму	0	$8,35 \cdot 10^5$	$2,62 \cdot 10^7$	$2,89 \cdot 10^8$	$1,72 \cdot 10^9$
Ширина рецептивная поля	183×128	51	50	50	50×50
Количество обучаемых параметров	0	52	1 633	46 400	1 125 633
Количество умножений на один параметр	0	16 069	16 053	6 237	1 530
PSNR	25	24,5	31,35	33,29	34,03
<i>SSIM</i>	0,5086	0,6143	0,8544	0,9058	0,9223
Время работы, мс	44	21	81	111	100

Таблица 2. Таблица с результатами нейронной сети из LPDR, обученной методом из этой статьи и с полученной в статье LPDR

Нейронная сеть	LPDR	LPDR (<i>SSIM</i>)
Количество умножений на одну синограмму	$4,11 \cdot 10^9$	$4,11 \cdot 10^9$
Ширина рецептивная поля	183×128	183×128
Количество обучаемых параметров	251 980	251 980
Количество умножений на один параметр	16 292	16 292
PSNR	38,73	35,49
<i>SSIM</i>	0,9635	0,9581
Время работы, мс	1214	1214

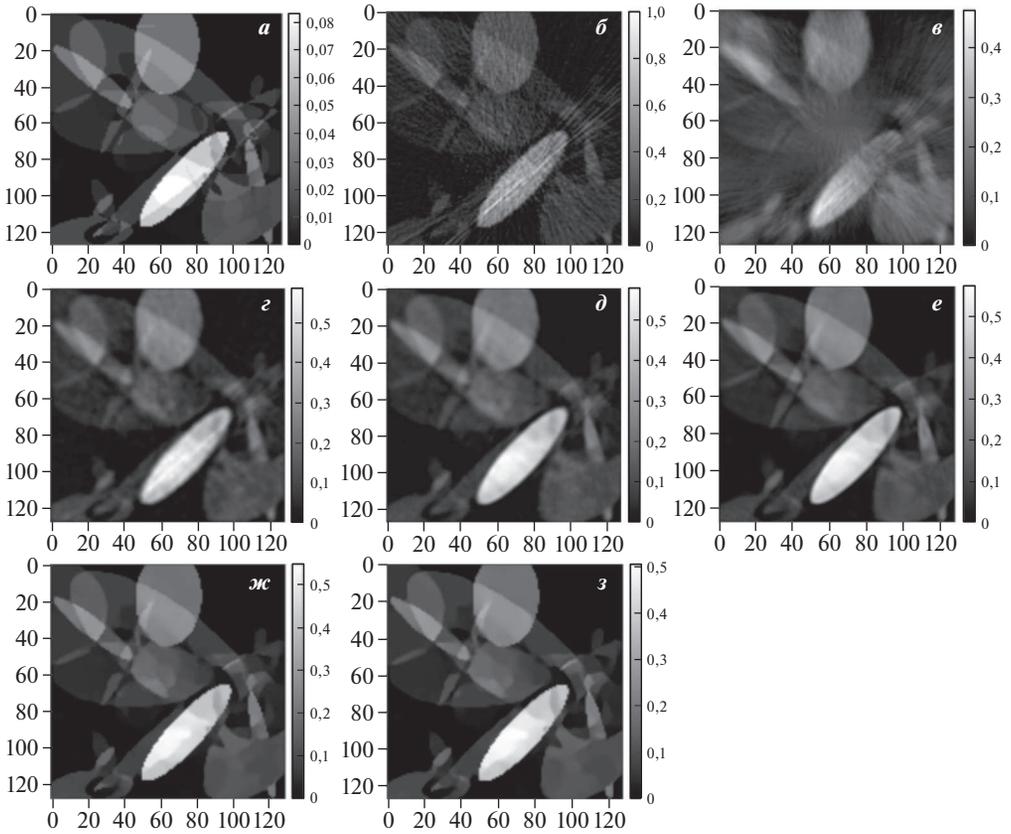


Рис. 9. Результаты различных реконструкций: *а* – фантом, *б* – FBP реконструкция, *в* – линейная свертка, *г* – трехслойная сеть, *д* – UNet1D, *е* – UNet, *ж* – LPDR, *з* – LPDR (*SSIM* функция потерь).

эффективности с двумерными сетями: UNet (рис. 8) и нейросетевой моделью из статьи Learned Primal Dual Reconstruction (LPDR) [6]. Результаты обучения нейронных сетей, а также измеренные характеристики выбранных архитектур представлены в табл. 1 и 2. Результаты реконструкций с помощью сравниваемых архитектур показаны на рис. 9.

При этом размеры свертков в трехслойной нейронной сети, одномерной UNet, двумерной UNet были подобраны так, чтобы рецептивные поля сетей составляли не менее 50 пикселей детектора в ширину, так как эта ширина достаточна для выполнения операции рамп-фильтрации. Также трехслойная нейронная сеть добавлена в сравнение как один из простейших возможных сверточных нейросетевых алгоритмов.

Согласно данным табл. 1 двумерная UNet является лучшей по метрикам PSNR и *SSIM*. Однако ее преимущества перед более простой UNet1D незначительны. Несмотря на то что использование *SSIM* в функции потерь вместо L2 улучшило PSNR и *SSIM* метрики для всех сетей из табл. 1, в экспериментах использование этой функции потерь не улучшило результаты для LPDR

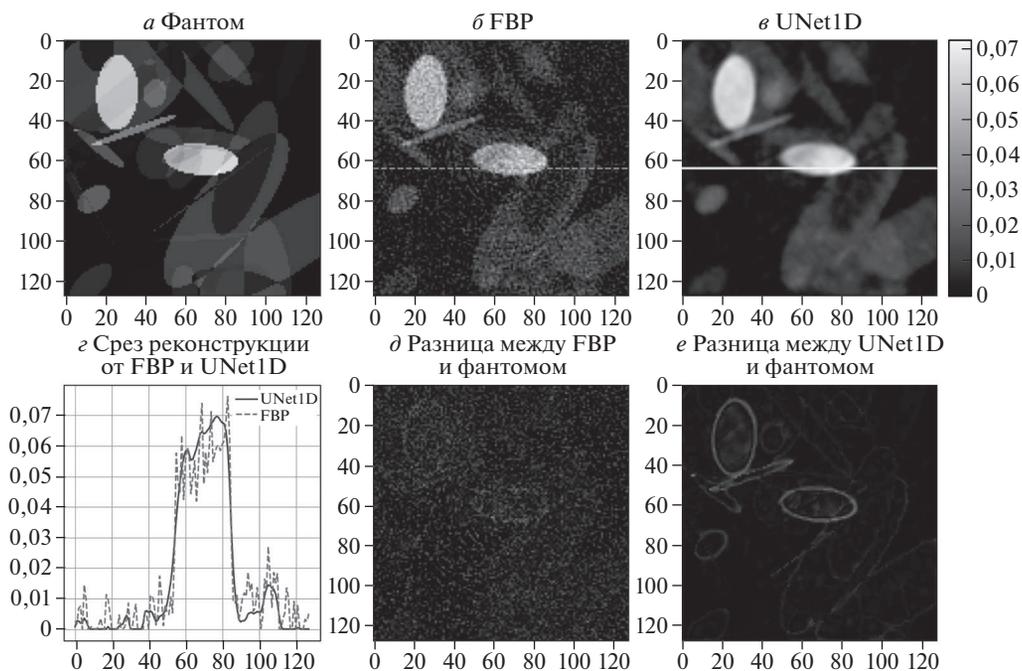


Рис. 10. Анализ ошибки предложенной UNet1D модели: *a* – фантом для проверки, *b* – FBP реконструкция, *в* – реконструкция предложенным в статье подходом, *г* – срезы реконструкций *b* и *в*, отмеченные штриховой и сплошной линиями соответственно, *д* – абсолютная разница между FBP реконструкцией и фантомом, *е* – абсолютная разница между реконструкцией предложенным в статье методом и фантомом. Все полутоновые картинки выполнены в одной шкале значений, указанной справа.

сети, как видно из табл. 2. Время работы нейронных сетей считалось исключительно на процессоре без использования видеокарты, включая требуемые для реконструкции операции преобразования Радона.

На рис. 9 видно, что при использовании UNet1D отсутствует однородный высокочастотный шум, однако границы объектов имеют однопиксельное размытие. Такие артефакты размытия, очевидно, отсутствуют в LPDR сети, это связано с отсутствием в UNet1D каких-либо нейронных слоев в пространстве реконструкций, которые бы поправили недостатки метода обратного проецирования.

Нейросеть, код для запуска нейросети и примеры синтетических данных доступны по ссылке

<https://github.com/ayamaev-se/NNPreprocessingTomography>

6. Заключение

В статье предложен метод фильтрации зашумленной синограммы для задачи низкодозовой компьютерной томографии. Основная идея метода заклю-

чается в применении нейронной сети перед выполнением оператора обратного проецирования. Показано, что рамп-фильтр возможно заменить с достаточной точностью на один одномерный сверточный слой нейронной сети, размер рецептивного поля которого равен 51 пикселю одномерного детектора при ширине рецептивного поля в 183 пикселя детектора. Добавление еще нескольких слоев позволяет подавлять шум, возникающий вследствие сокращения времени экспозиции при регистрации одного томографического изображения. Таким образом, построенная в статье одномерная нейронная сеть заменила фильтр в алгоритме FBP, обучаясь на невязке реконструкции с фантомом по метрике $SSIM$ и обогнав оригинальный FBP оператор. Исследование результатов работы сети проведено на синтетических данных и сравнивалось с широко используемыми нейросетевыми архитектурами. Показано, что предложенный подход производит реконструкцию с точностью выше 0,9 по метрике $SSIM$ в среднем. Кроме того, при таком подходе время выполнения реконструкции на CPU в среднем в 11 раз меньше, чем при использовании LPDR. Также показана эффективность использования оперативной памяти вычислительного устройства при использовании описанного одномерного подхода по сравнению с рассматриваемыми двумерными сетями.

СПИСОК ЛИТЕРАТУРЫ

1. *Hu Z., et al.* An Improved Statistical Iterative Algorithm for Sparse-view and Limited-angle CT Image Reconstruction // Scientific reports. 2017. V. 7. No. 1. P. 1–9.
2. *Swensen S.J., et al.* Screening for Lung Cancer with Low-dose Spiral Computed Tomography // Amer. J. Respiratory and Critical Care Medicine. 2002. V. 165. No. 4. P. 508–513.
3. *Thanh D., et al.* A Review on CT And X-ray Images Denoising Methods // Informatica. 2019. V. 43. No. 2.
4. *Johnson C.A., Sofer A.* A Data-parallel Algorithm for Iterative Tomographic Image Reconstruction // Proc. Frontiers' 99. Seventh Sympos. on the Frontiers of Massively Parallel Computation. IEEE, 1999. P. 126–137.
5. *Yang H.K., et al.* Slice-wise Reconstruction for Low-dose Cone-beam CT Using a Deep Residual Convolutional Neural Network // Nuclear Science and Techniques. 2019. V. 30. No. 4. P. 1–9.
6. *Adler J., Oktem O.* Learned Primal-dual Reconstruction // IEEE Trans. Medical Imaging. 2018. V. 37. No. 6. P. 1322–1332.
7. *Mizusawa S., et al.* Computed Tomography Image Reconstruction Using Stacked U-Net // Computerized Medical Imaging and Graphics. 2021. V. 90. P. 101920.
8. *Han Y., Ye J.C.* Framing U-Net via Deep Convolutional Framelets: Application to Sparse-view CT // IEEE Trans. Medical Imaging. 2018. V. 37. No. 6. P. 1418–1429.
9. *Nakai H., et al.* Quantitative and Qualitative Evaluation of Convolutional Neural Networks with a Deeper U-net for Sparse-view Computed Tomography Reconstruction // Academic radiology. 2020. V. 27. No. 4. P. 563–574.
10. *Zhu L., et al.* Metal Artifact Reduction for X-ray Computed Tomography Using U-net in Image Domain // IEEE Access. 2019. V. 7. P. 98743–98754.

11. *Cierniak R.* A 2D Approach to Tomographic Image Reconstruction Using a Hopfield-type Neural Network // *Artificial Intelligence in Medicine*. 2008. V. 43. No. 2. P. 113–125.
12. *Yang Q., et al.* Low-dose CT Image Denoising Using a Generative Adversarial Network with Wasserstein Distance and Perceptual Loss // *IEEE Trans. Medical Imaging*. 2018. V. 37. No. 6. P. 1348–1357.
13. *Bulatov K., et al.* Monitored Reconstruction: Computed Tomography as an Anytime Algorithm // *IEEE Access*. 2020. V. 8. P. 110759–110774.
14. *Hore A., Ziou D.* Image Quality Metrics: PSNR vs. SSIM // 2010 20th international conference on pattern recognition. IEEE, 2010. P. 2366–2369.
15. *Kofler A., et al.* A U-Nets Cascade for Sparse View Computed Tomography // *Int. Workshop on Machine Learning for Medical Image Reconstruction*. Springer, Cham, 2018. P. 91–99.
16. *Wei Y., Wang G., Hsieh J.* An Intuitive Discussion on the Ideal Ramp Filter in Computed Tomography (I) // *Comput. Math. Appl.* 2005. V. 49. No. 5–6. P. 731–740.
17. *Kak A.C., Slaney M., Wang G.* Principles of Computerized Tomographic Imaging. 2002.
18. *Adler J., Kohr H., Öktem O.* ODL — a Python Framework for Rapid Prototyping in Inverse Problems. 2017. Code and documentation available online: <https://github.com/odlgroup/odl>.
19. *Van Aarle W., et al.* The ASTRA Toolbox: A Platform for Advanced Algorithm Development in Electron Tomography // *Ultramicroscopy*. 2015. V. 157. P. 35–47.

Статъя представена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 01.06.2021

Принята к публикации 30.06.2021

Заметки, хроника, информация



ОСНОВНЫЕ РЕЗУЛЬТАТЫ НАУЧНОГО ТВОРЧЕСТВА И ДЕЯТЕЛЬНОСТИ В ОБЛАСТИ ПРАКТИЧЕСКОЙ КОСМОНАВТИКИ Ю.П. ПОРТНОВА-СОКОЛОВА (к 100-летию со дня рождения)

DOI: 10.31857/S0005231021100135

5 августа 2021 г. исполняется 100 лет со дня рождения Юрия Петровича Портнова-Соколова — крупного ученого в области теории построения бортовых систем управления объектами ракетно-космической техники.

После окончания в 1946 г. с отличием Московского авиационного института Ю.П. Портнов-Соколов, проработав три года на кафедре МАИ, поступил на работу в Институт автоматике и телемеханики АН СССР (ИАТАН) (в настоящее время — Институт проблем управления РАН). Здесь он прошел яркий творческий путь и проработал до последних дней своей жизни (до 2004 г.).

Уже в начале своей деятельности Юрий Петрович был привлечен к работам по ракетно-космической технике, которые возглавил в ИАТАН Б.Н. Петров — будущий академик, вице-президент АН СССР.

При исследовании динамики мощного жидкостного реактивного двигателя (ЖРД) Юрием Петровичем Портновым-Соколовым были получены приори-

тетные результаты — модели двигателя как объекта управления и принципы его регулирования, которые легли в основу многих дальнейших работ в области управления силовыми установками жидкостных ракет.

При создании межконтинентальной ракеты Королева, получившей рабочее название Р-7, возникла проблема терминального управления расходом топлива через воздействие на тягу и коэффициент соотношения многодвигательной установки ракеты блочной компоновки. Эта проблема была абсолютно новой, не было никаких прототипов решений, даже отдаленно приближающихся к возникшим здесь новым задачам.

При разработке этой проблемы коллективом сотрудников Института, руководимым Ю.П. Портновым-Соколовым, были получены основополагающие результаты в создании бортовых терминальных систем управления расходом топлива (СУРТ). В этих системах использовалась информация о запасах и расходах топлива, а также инерциальная информация для воздействия на расход топлива, т.е. на тягу ЖРД и на соотношение расходов компонентов топлива. Генеральная цель их функционирования состоит в минимизации гарантийных запасов топлива, что при неизменном стартовом весе топлива позволяло использовать высвободившиеся гарантийные запасы для увеличения сжигаемого в ЖРД количества компонентов топлива, т.е. для увеличения конечной скорости ракеты. Установка таких систем на борту ракеты повышает на 10–15 % энергетические характеристики носителя и соответственно увеличивает дальность стрельбы, массу полезного груза или высоту орбиты, на которую груз выводится.

При участии коллектива Института, руководимого Ю.П. Портновым-Соколовым, была создана система управления расходом топлива ракеты Р-7, с помощью которой 4 октября 1957 г. был осуществлен запуск первого спутника ПС-1, а 12 апреля 1961 г. — полет Ю.А. Гагарина на корабле “Восток”.

В итоге более чем полувековых работ под руководством Ю.П. Портнова-Соколова сформировалось новое направление совершенствования ракет-носителей с жидкостными ракетными двигательными установками на основе создания широкого класса терминальных систем управления расходом топлива. Были разработаны технология проектирования таких систем и ряд серийных систем управления СУРТ, являющихся неотъемлемой частью всех мощных жидкостных ракет, от первой ракеты Р-7 до ракет-носителей “Протон”, “Зенит”, “Энергия” и проектируемых в настоящее время перспективных ракет-носителей (“Союз-5”, семейство “Ангара”, “Аврора”).

Участие в этих работах потребовало адаптации коллектива научных сотрудников ИАТ, руководимого Ю.П. Портновым-Соколовым, к условиям работы в режиме опытно-конструкторской разработки (ОКР). Новые принципы построения и методы синтеза получали воплощение в конструкторской документации и технологии разработки систем управления. Помимо участия в разработке эскизных проектов, создания алгоритмов управления, максимально приближенных к условиям их реализации на бортовых вычислительных средствах, формирования эксплуатационных характеристик систем управле-

ния на основе оценки их динамики и точности, ОКР включает создание документации, регламентирующей технологию стендовой и натурной отработок. В состав такой документации входят тестовые примеры для контроля правильности функционирования бортового программного обеспечения, имитационные программные модули для проведения предпускового моделирования, методики адаптации алгоритмов управления к условиям конкретного пуска, оценки функционирования системы управления при стендовых и натурных испытаниях.

За работы в области практической космонавтики Ю.П. Портнов-Соколов награжден высокими правительственными наградами и дважды удостоен Государственной премии СССР.

В процессе разработки конкретных бортовых систем и исследований различных задач терминального управления возникло понимание, что бортовые терминальные системы имеют свою специфику и занимают определенное место в общем ряду систем автоматического управления. Практические задачи, возникающие в ракетно-космической технике, послужили толчком к разработке теории бортовых систем терминального управления и систематическому изложению теории и накопленного опыта в специальной монографии. Основная особенность развитых методов синтеза заключается в том, что все они инициированы практическими нуждами проектирования бортовых систем управления и учитывают реальные условия эксплуатации объектов ракетно-космической техники.

На основе обобщения результатов исследований в теории бортовых терминальных систем управления Ю.П. Портновым-Соколовым создана научная школа в области теории терминального управления.

В 90-х гг. XX в. Ю.П. Портновым-Соколовым был выполнен цикл работ по проблеме безопасности и управлению риском объектов ракетно-космической техники, отмеченный в 2004 г. Премией РАН им. акад. Б.Н. Петрова.

В основе этих работ лежит концепция управления безопасностью по обобщенному критерию, учитывающему различные факторы риска. Обеспечение безопасности охватывает основные этапы жизненного цикла объектов, начиная от формирования проектного облика до этапа сопровождения эксплуатации. Здесь используются средства проектно-технического и эксплуатационно-технического управления. В системах управления совершенствуются их принципы действия с приоритетом критерия безопасности и реализуются отказоустойчивые алгоритмы формирования управления.

Творческий жизненный путь Ю.П. Портнова-Соколова является ярким примером служения своей стране, науке, идее.

Фундаментальность и жизнестойкость сформировавшегося в итоге направления в ракетодинамике подтверждается в настоящее время востребованностью бортовых терминальных систем управления расходом топлива в современных средствах выведения. Институт продолжает и развивает работы, начатые под руководством Ю.П. Портнова-Соколова, активно участвуя в совершенствовании новых перспективных ракет-носителей, повышая их энергетические характеристики, безопасность и, как следствие, повышая рей-

тинг отечественных ракет-носителей на коммерческом рынке выведения на космические орбиты полезных нагрузок

Основные результаты пройденного пути составляют как научное наследие, содержащееся в многочисленных статьях, монографии и изобретениях, так и значительный личный вклад Ю.П. Портнова-Соколова в практическую космонавтику. Однако, пожалуй, самым главным в наследии Ю.П. Портнова-Соколова является создание коллектива единомышленников, который и в настоящее время развивает его идеи и сохраняет востребованность Института в области практической космонавтики.

В.П. Иванов, д-р техн. наук (vladguc@ipu.ru)

СОДЕРЖАНИЕ

Вступительное слово программного комитета конференции «Интеллектуализация обработки информации» (ИОИ)	3
Афраймович Л.Г., Емелин М.Д. Эвристические стратегии комбинирования решений трехиндексной задачи о назначениях	6
Генрихов И.Е., Дюкова Е.В. Поиск частых элементов произведения частичных порядков с использованием параллельных вычислений	13
Германчук М.С., Козлова М.Г., Лукьяненко В.А. Псевдодобулевые модели условной оптимизации для класса задач многих коммивояжеров	25
Гешпенер В.В., Мандрикова Б.С. Обнаружение и идентификация аномальных эффектов в сложном сигнале	46
Гладин Е.Л., Алкуса М., Гасников А.В. О решении выпуклых min-min задач с гладкостью и сильной выпуклостью по одной из групп переменных и малой размерностью другой	60
Корнеев В.П. Эффективный алгоритм тупиковых управлений для решения задач комбинаторной оптимизации	76
Лазарев А.А., Лемтюжникова Д.В., Тюняткин А.А. Метрическая интерполяция для задачи минимизации максимального временного смещения для одного прибора	93
Мандель А.С., Лаптин В.А. Оптимальное управление системами массового обслуживания с переключением каналов обслуживания	110
Фурсов В.А., Минаев Е.Ю., Котов А.П. Оценивание параметров движения аппарата по наблюдениям опорной поверхности	124
Шибзухов З.М. Об одном робастном подходе к поиску центров кластеров ...	140
Ямаев А.В., Чукалина М.В., Николаев Д.П., Шешкус А.В., Чуличков А.И. Нейронная сеть для предварительной обработки данных в компьютерной томографии	152

Заметки, хроника, информация

Иванов В.П. Основные результаты научного творчества и деятельности в области практической космонавтики Ю.П. Портнова-Соколова (к 100-летию со дня рождения)	165
--	-----

C O N T E N T S

Opening Remark by the Program Committee of “Intelligent Data Processing: Theory and Applications” (IDP) Conference	3
Afraimovich L.G., Emelin M.D. Heuristic Strategies for Combining Solutions of the Three-Index Axial Assignment Problem.....	6
Genrikhov I.E., Djukova E.V. Search for Frequent Elements of a Product of Partial Orders Using Parallel Calculations.....	13
Germanchuk M.S., Kozlova M.G., Lukianenko V.A. Pseudo-Boolean Models of Conditional Optimization for Multiple Traveling Salesman Problem.....	25
Geppener V.V., Mandrikova B.S. Detection and Identification of Anomalous Effects in a Complex Signal	46
Gladin E.L., Alkousa M., Gasnikov A.V. On Solving Convex Min-Min Problems with Smoothness and Strong Convexity in One Variable Group and Small Dimension of the Other	60
Korneenko V.P. An Effective Algorithm Dead-End Control for Solving Combinatorial Optimization Problems.....	76
Lazarev A.A., Lentiuzhnikova D.V., Tunyatkin A.A. Metric Interpolation for Single Machine Minimization Maximum Lateness Problem.....	93
Mandel A.S., Laptin V.A. Queuing Systems with Switching Channels Optimum Control	110
Fursov V.A., Minaev E.Y., Kotov A.P. Vehicle Motion Estimation Using Visual Observations of Elevation Surface.....	124
Shibzukhov Z.M. On One Robust Approach to Find the Centers of Clusters.....	140
Yamaev A.V., Chukalina M.V., Nikolaev D.P., Sheshkus A.V., Chulichkov A.I. Neural Network for Preprocessing Data in Computed Tomography	152

Notes, Meetings, Information

Ivanov V.P. Main Results of Scientific Work and Activities in the Field of Practical Cosmonautics Y.P. Portnov-Sokolov (on the 100th anniversary of his birth)	165
--	-----