На правах рукописи

Салий Ярослав Витальевич

# НЕКОТОРЫЕ МЕТОДЫ РЕШЕНИЯ МАРШРУТНЫХ ЗАДАЧ С УСЛОВИЯМИ ПРЕДШЕСТВОВАНИЯ

Специальность 05.13.18 — математическое моделирование, численные методы и комплексы программ

Автореферат диссертации на соискание ученой степени кандидата физико-математических наук

Екатеринбург — 2018

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук

*Научный руководитель*:

доктор физико-математических наук,
член-корреспондент РАН, профессор *Ченцов Александр Георгиевич*

*Официальные оппоненты*:

*Соколинский Леонид Борисович,*
доктор физико-математических наук, профессор,
Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет (национальный исследовательский университет)» (г. Челябинск),
заведующий кафедрой системного программирования

*Лемперт Анна Ананьевна,*
кандидат физико-математических наук,
Федеральное государственное бюджетное учреждение науки
Институт динамики систем и теории управления им. В. М. Матросова Сибирского отделения Российской академии наук (г. Иркутск),
ведущий научный сотрудник лаборатории 6.2 логических и оптимизационных методов анализа сложных систем

*Ведущая организация*:

Федеральное государственное бюджетное учреждение науки
Институт математики им. С. Л. Соболева Сибирского отделения
Российской академии наук (г. Новосибирск).

Защита состоится «19» сентября 2018 года в 13:00 на заседании диссертационного совета Д 212.285.25 на базе ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» по адресу: 620000, г. Екатеринбург, пр. Ленина, 51, к. 248.

С диссертацией можно ознакомиться в библиотеке и на сайте ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина», по адресу `http://lib.urfu.ru/mod/data/view.php?d=51&rid=279073`.

Автореферат разослан «___» ________ 2018 года.

Ученый секретарь
диссертационного совета
Д 212.285.25, д.ф.-м.н., проф. Пименов Владимир Германович

*Замечание.* Работы, опубликованные автором по теме диссертации, пронумерованы *римскими* цифрами. Ссылки на цитированные работы даны отдельным списком, пронумерованы *арабскими* цифрами.

## Степень разработанности. Актуальность

Рассматриваемые в диссертации задачи допустимо полагать обобщениями широко известной *задачи коммивояжера* (Traveling Salesman Problem), далее TSP; сравнительно недавний обзор TSP и некоторых обобщений представлен в [1]. Диссертация, в основном, посвящена задачам с *условиями предшествования*[1]: требуется некоторый город $a$ непременно посетить ранее города $b$ и т. п.Распространены три варианта *формализации* условий предшествования: *отношение порядка* [2], *ациклический орграф* [3] и «фундированное» *бинарное отношение* [4]. Для *конечного* числа городов, все три представления эквивалентны; для перехода к частичному порядку можно взять *транзитивное замыкание* (см. [5]) множества пар.

В диссертации условия предшествования формализуются в виде некоторого частичного порядка $P = (1 \mathinner{..} n, <_P)$; предполагается, что если для двух городов $a, b \in 1 \mathinner{..} n$ известно, что $a <_P b$, то в любом *допустимом* маршруте город $a$ должен стоять *прежде* $b$.

В русскоязычной литературе задача коммивояжера с условиями предшествования обычно называется *задачей курьера* [6]. В англоязычной литературе об. Sequential Ordering Problem (SOP), по [3]; также встречается Precedence Constrained TSP, сокращаемое как TSP-PC [7], PCATS, PCTSP[2] [8]; далее пишем TSP-PC. Эта задача имеет немало приложений: логистика, оптимизация производственных процессов, автоматизированных систем хранения. Наиболее распространен метод решения TSP-PC с помощью линейной релаксации задач целочисленного программирования (Mixed Integer-Linear Programming, MILP), напр. [8, 9]. Кроме MILP следует отметить решения «чистым» методом ветвей и границ [10] и решения на основе «многозначных диаграмм решений» [11] (multivalued decision diagrams).

*Динамическое программирование* (ДП), работа с которым представляет основное содержание диссертации, нечасто используется для общего случая TSP-PC: насколько известно автору, за исключением работ А. Г. Ченцова и соавторов, наиболее новой можно считать публикацию 1992 года [7]. Впервые ДП для TSP-PC было сформулировано в [12] — в прямой постановке, по Хелду и Карпу, и в [13] — в *попятной* постановке, по Беллману.

[1] в англоязычной литературе встречаются в форме «precedence constraints», реже «precedence relations»

[2] также обозначает Prize Collecting TSP [1, Ch. 14]

Модель с *абстрактной* функцией агрегирования затрат позволяет единообразно с задачами с *аддитивным* агрегированием стоимости перемещений рассматривать задачи *на узкие места* (то же *минимакс*) Bottleneck TSP (BTSP, [1, Ch. 15]); корректность ДП для задачи с условиями предшествования доказана, например, в [14].

Несмотря на малое внимание к B(G)TSP-PC[3] в литературе (исключая серию работ А. Г. Ченцова и соавторов, в частности, [III,IV]), следует отметить, что для многих приложений TSP-PC такая постановка тоже актуальна: там, где TSP-PC предполагает минимизацию *суммарных затрат* агента, в рамках BTSP-PC можно изучить *выполнимость* задания исходя из ограниченных ресурсов агента, пополняемых при каждом посещении города; например, можно поставить задачу о минимальной емкости *аккумулятора* электромобиля, электрокара или беспилотного летательного аппарата (дрона).

Кроме того, эти постановки могут рассматриваться как элемент *робастных* методов дискретной оптимизации [15], в частности, можно понимать их как эвристическое приближение к «истинно робастным» задачам. Важна и двойственную постановка: задачу на максимум с $min$-агрегированием, известная как Maximum Scatter TSP [16]. Одно из приложений этой постановки связано с избежанием *термической* деформации, например, при установке заклепок; подобная проблематика также важна в задачах о маршрутизации инструмента в машинах листовой резки, см. [17].

Важным обобщением TSP является задача с зависимостью стоимости перемещений от времени (Time-Dependent TSP, далее TD-TSP); см. обзор [18]. За редкими исключениями, зависимость от времени не исследуется в комбинации с другими обобщениями TSP. Из них упомянем [19], где применен комбинированный подход, задействующий *программирование в ограничениях* (constraint programming), «диаграммы решений» (decision diagrams) и лин. программирование для решения TD-TSP, TD-TSP с временными окнами (TD-TSP with Time Windows) и TD-TSP с условиями предшествованиями (далее TD-TSP-PC); в частности, предложен набор тестовых экземпляров на основе TSPLIB[4], с функцией стоимости типа «traveling deliveryman».

Дальнейшим обобщением TD-TSP можно считать задачи с «зависимостью от предыстории»: стоимость перемещения из города $a$ в город $b$ зависит от множества городов *уже посещенных* на момент перемещения из $a$ в $b$. По аналогии с задачами теории расписаний [20], будем называть соответствующий вариант задачи коммивояжера Past Sequence Dependent TSP-PC

[3]задача с max-агрегированием стоимости перемещений и условими предшествования или ее обобщенный (с кластерами/мегаполисами) вариант

[4]библиотека тестовых экземпляров TSP и других задач, включая TSP-PC, см. `https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

(PSD-TSP-PC)[5]. В TSP-подобной постановке зависимость от списка заданий, по-видимому, впервые описана в 2010 году: в [21] решается одна специфическая задача монтажа печатных плат; в [14] рассматривалась более общая постановка задачи, связанная с минимизацией облучения сотрудников АЭС.

Также отметим потребность в задачах с зависимостью от списка заданий в вопросах маршрутизации инструмента в машинах листовой резки: в них стоимость перемещений совмещает характеристики (а.) времени холостого хода и (б.) риска брака в процессе резки [17].

### Обобщенная задача

Рассмотрим подробнее *обобщенную*[6] задачу коммивояжера *с условиями предшествования*, которую обозначим GTSP-PC. По-видимому, первое ее формальное описание (строго говоря, то было CTSP-PC, см. сноску) представлено в [24] в применении к автоматизации обработки листа металла (сверление, развертывание отверстий, нарезка резьбы); для решения задачи предлагалась вариация метода ветвей и границ.

В [25] GTSP-PC применялась для маршрутизации инструмента в машинах листовой резки, предлагался эвристический алгоритм решения. В [13] предложено решение GTSP-PC, одно из приложений — минимизация радиационного облучения персонала АЭС; в [23] представлена постановка задачи с *внутренними работами*, позволяющая единообразно рассматривать GTSP-PC и CTSP-PC. В [26] представлено приложение к оптимизации работы координатно-измерительных машин на сборочном конвейере для автомобилей; наилучшие результаты были показаны гибридной эвристикой муравьиной колонии.

Необобщенная, «обычная» задача коммивояжера на узкие места (BTSP) *без условий предшествования* описана, например, в [27]. Обобщенная BTSP (Bottleneck Generalized TSP, BGTSP) в [28] рассматривалась для приложения к трассировке проводников на многослойных печатных платах. Условия предшествования в BGTSP (которую с ними будем называть BGTSP-PC) впервые добавлены в [29], где на BGTSP-PC распространен вариант ДП из [13]. Затем, в [30] постановка дополнена *зависимостью от списка невыполненных заданий* (следует называть такую задачу PSD-BGTSP-PC).

[5] в [IV,XII,XIII] мы использовали ту же аббревиатуру без «past» (SD-TSP-PC), вслед за [21]

[6] Generalized TSP предполагает посещение *одного* города из каждого мегаполиса; Clustered TSP [22] предполагает посещение *всех* городов в кластере — прежде перехода к следующему. Далее под GTSP-PC мы понимаем оба варианта задачи, с учетом их единого представления через формализм *внутренних работ* [23], см. пример [IV].

## Качественное исследование TSP-PC. Вопросы сложности

Задача TSP-PC содержит обычную TSP как частный случай, значит, NP-трудна. Между тем, очевидно, что сложность конкретного экземпляра связана с разновидностью условий предшествования. В одной из первых работ, посвященных решению задачи, близкой к TSP-PC [31] (для узкого класс условий предшествования «pickup and delivery constraints»), применялся метод ветвей и границ, где наихудшая оценка производительности есть перебор всех маршрутов; было рассчитано их количество — для $2n$ городов-заказчиков получается $(2n)!/2^n$ допустимых маршрутов. Автору неизвестны другие попытки рассчитать количество допустимых маршрутов в работах по *дискретной оптимизации*, однако, в целом теория, позволяющая это делать, хорошо разработана в рамках *теории порядков* [32], хотя задача их подсчета и труднорешаема (#P-полная [32]). Специфическая для методов решения, использующих MILP «сложность» TSP-PC — размерность политопа — описана в [9].

В контексте ДП сложность решения TSP-PC зависит от количества существенных списков заданий — порядковых идеалов (прямое ДП по Хелду и Карпу) или фильтров (попятное ДП по Беллману). Вопрос о количестве порядковых идеалов представляется менее разработанным чем вопрос о количестве линейных продолжений; в контексте дискретной оптимизации он рассматривался в [2]; кроме того, в [33] напрямую оценивалась сложность ДП-решения для простейшего случая условий предшествования (одна пара сравнимых мегаполисов); также этот вопрос исследовался в [II]. В общем случае задача подсчета числа порядковых идеалов *труднорешаема* (#P-полная, т. к. сводится к задаче о перечислении всех *антицепей* [2]), большой интерес представляют ее приближения, см. [2]. Отметим также статьи в которых, благодаря особым условиям предшествования, сложность ДП становится *линейной*: [34, 35].

## Цели и задачи работы.

Наша цель состоит в разработке и анализе основанных на ДП методов и алгоритмов решения задач *маршрутизации* перемещений *с ограничениями* и функциями стоимости перемещений, допускающими зависимость от списка невыполненных заданий, для достижении которой потребовалось решить следующие **задачи**:

1. Провести качественный анализ модели перемещений с условиями предшествования, охватывающей разнообразные практические задачи.

2. Построить конструкции решения задач маршрутизации на основе широко понимаемого *динамического программирования*.

3. Оценить влияние *условий предшествования* на вычислительную сложность.

4. Разработать и реализовать алгоритмы решения, провести вычислительные эксперименты.

**Основные положения, выносимые на защиту:**

1. Проведено качественное исследование комбинаторной модели перемещений с ограничениями в виде условий предшествования: получены гарантии вычислительной сложности, на основе которых выработан критерий практической разрешимости.

2. Реализован программный комплекс, выполняющий решение задач, соответствующих модели перемещений, точным (ДП) и эвристическим (усеченное ДП) методами.

3. Реализован экономичный вариант ДП для PSD-BGTSP-PC; предложен подход к параллелизации этого варианта для систем с общей памятью; проведено сравнение настоящего варианта ДП с другими методами для TSP-PC в классической постановке и в постановке с зависимостью стоимости перемещений от времени (минимизация суммарной задержки) на примере экземпляров из TSPLIB.

4. Продемонстрирована работоспособность эвристики усеченного ДП для PSD-BGTSP-PC на нескольких модельных задачах; получена оценка ее временно́й сложности; проведено сравнение эвристики усеченного ДП с другими методами решения TSP-PC в обычной постановке и в одной постановке с зависимостью функции стоимости перемещений от времени (минимизация суммарной задержки) на примере экземпляров из TSPLIB на различных значениях параметра эвристики.

**Научная новизна**

**Динамическое программирование для задачи курьера и ее обобщений [I–IV,VI,VIII–XIII].** В [IV] описана разработанная автором оригинальная схема программной реализации ДП, использовавшаяся в [I,III,IV,VIII,IX,XI]. Новизна состоит в генерации существенных списков заданий *снизу вверх*, что позволяет и сгенерировать все существенные списки заданий, и расчитать все значения функции Беллмана «одновременно», в прямом ходе алгоритма снизу вверх, в отличие от процедуры, описываемой, например, в [4], где предполагается сначала сгенерировать все существенные списки заданий посредством оператора **I** *сверху вниз* и затем, в обратном ходе алгоритма *снизу вверх* рассчитывать значения функции Беллмана.

Кроме того, оригинальным является *иерархическое* устройство основной структуры данных, хранящей значения функции Беллмана: каждому *существенному списку* сопоставляется множество пар «база–экстремум состояния», что позволяет сэкономить память за счет того что для всех состояний с одинаковым списком заданий последний хранится в *единственной* копии, а не повторяется каждый раз в отличие от, например, *плоской* реализации [36]; подобная экономия становится особенно важной в *обобщенных* задачах.

В [XII] с помощью точного ДП *впервые* получено оптимальное решение экземпляра TSP-PC ry48p.3.sop из TSPLIB. В [XIII] продемонстрирована эффективность ДП для некоторых «сильно ограниченных» экземпляров; напр., p43.4.sop впервые был оптимально решен в [37] за 20 часов и, независимо, в [38] за чуть более чем 4 часа; расход оперативной памяти в этих работах не замерялся; из недавних результатов, в [11] этот экземпляр решался за 88,49 секунд и в [8] за 11 секунд. Посредством ДП автору удалось решить этот экземпляр менее чем за 1 секунду [XIII] с расходом памяти менее 23 МБ.

В части решения TD-TSP-PC, в [XIII] автору удалось превзойти некоторые результаты [19]: точным ДП впервые получены оптимальные решения 7 экземпляров, включая экземпляры свыше 100 городов, размерность которых представлялась в [19] как слишком высокая; показано, что эвристика усеченного ДП (restricted dynamic programming [36]) достигает все известные оптимальные решения кроме одного экземпляра (p43.1.sop) на подходящем значении параметра и улучшены либо достигнуты *все* оценки сверху для экземпляров из [19], для которых не известно оптимального решения.

Испытания реализаций варианта точного ДП для обобщенных задач представлены на нескольких модельных экземплярах BGTSP-PC [I,IV], PSD-BGTSP-PC [III,IV] с 27 и 30 мегаполисами. Применительно к последним, в [IV] также описана схема параллелизации генерации существенных списков заданий и расчета значений функции Беллмана для систем с общей памятью на основе OpenMP, использовавшаяся в [III,IV,VIII,IX,XI]. Новизна состоит в применении конструкций `task`, введенных в OpenMP 3.0, для параллелизации обхода вложенной хеш-таблицы, которую, в отличие от *массива*, невозможно обработать параллельно директивой `parallel for`, поскольку *хеш-таблица* не гарантирует доступ к элементам за *постоянное* время. На нескольких модельных экземплярах BGTSP-PC и PSD-BGTSP-PC продемонстрирована работоспособность схемы. Также показана работоспособность для этих задач варианта эвристики усеченного ДП.

**Качественное исследование модели перемещений с условиями предшествования: пространственная и временная сложность [II,IV,XII,XIII].** Получены оценки пространственной и временной сложности решения TSP-PC с помощью ДП и их приближения сверху и снизу; впервые вычислены «приближения» оценок для общепринятых тестовых экземпляров TSP-PC (из TSPLIB). Метод получения оценок распространяет результаты [2] на TSP-PC; новизна проистекает из различий в пространстве состояний ДП и постановок задач. Также, в [IV] представлена оценка временной сложности эвристики усеченного ДП для PSD-BGTSP-PC.

Результаты о «количестве существенных списков заданий» [II] повторяют часть результатов о количестве идеалов в *последовательно-параллельных* частично упорядоченных множествах (см. [2]), но получены независимо и выражены на языке *фундированных* отношений (как в [4]), а не частично упорядоченных множеств. Кроме того, в [II] впервые формально описан класс условий предшествования, порождаемый требованием всегда «вырезáть *внутренний* контур прежде *внешнего*» в задачах маршрутизации инструмента в машинах листовой резки (лес с исходящей степенью не более 1).

## Теоретическая значимость

Определены пространственная и временная сложность ДП для TSP-PC и основанной на нем эвристики *усеченного* ДП; потребность в определении сложности последней связана с тем, что эвристика «в пределе» — при приближении параметра эвристики к мощности наибольшего слоя пространства состояний — переходит в *точное* ДП.

## Практическая значимость

Экспериментально показано, что средствами ДП можно эффективно решать определенный класс задач с условиями предшествования, а выявлять попадание произвольной данной задачи в этот класс можно затрачивая много меньше времени и ресурсов чем потребовалось бы на запуск и ожидание решения либо исчерпания лимита памяти. Предложена и апробирована практичная схема реализации ДП. Показана эффективность основанной на ДП эвристики, «автоматически» соблюдающей важные на практике ограничения.

## Методология и методы исследования

В статьях, посвященных задачам дискретной оптимизации, можно выделить четыре ключевых методологических компонента: (1.) строгая постановка задачи, обычно комбинаторная, напр. [4, 7, 19], или в виде задачи целочисленного программирования, напр. [8]; (2.) формальное описание метода решения, обоснование его корректности; (3.) исследование сложности решения задачи указанным методом; (4.) испытание метода на общепринятом наборе тестовых экземпляров, напр. TSPLIB, на нескольких задачах из практики, напр. [25, 26], либо на модельных задачах, как правило, случайно сгенерированных с учетом специфики [39].

Этой же методологии придерживался и автор диссертации в исследованиях ДП и основанной на нем эвристики *усеченного* ДП (restricted dynamic programming [36]) для задач с условиями предшествования.

## Степень достоверности

Сформулированные в диссертации научные положения, выводы и рекомендации обоснованы теоретическими построениями и экспериментальными данными, корректны математически и согласуются с известным опытом создания моделей и алгоритмов решения маршрутных задач.

## Апробация работы

Основные результаты работы представлялись в докладах на Всероссийской конференции «Статистика, моделирование, оптимизация» (Россия, Челябинск, 2011); 43-й, 46-й, 47-й Всероссийских молодежных школах-конференциях «Современные проблемы математики» (Россия, Екатеринбург, 2012, 2016, 2017 гг.); Международной конференции «Дискретная оптимизация и исследование операций» (Россия, Новосибирск, 2013); Всероссийской конф. с междунар. участием «Алгоритмический анализ неустойчивых задач», посвящ. памяти В. К. Иванова (Россия, Челябинск, 2014); XV Всерос. конф. «Математическое программирование и приложения» (Россия, Екатеринбург, 2015); 8-й Всерос. мультиконф. по проблемам управления (Россия, Дивноморское, 2015); Fourth Russian–Finnish Symposium on Discrete Mathematics (Finland, Turku, 2017). Также, диссертация представлялась на семинарах «Дискретные экстремальные задачи», ИМ им. С. Л. Соболева СО РАН (Россия, Новосибирск, 2017); «Математическое моделирование и дискретная оптимизация» ОФ ИМ им. С. Л. Соболева СО РАН (Россия, Омск, 2017); на Научном семинаре по информационным технологиям ЮУрГУ (Россия, Челябинск, 2017) и на семинаре Отдела управляемых систем ИММ УрО РАН им. Н. Н. Красовского.

**Разделение вкладов в публикациях в соавторстве.**

В работах [VI,I,VIII,IX,III,XI] доказательство корректности уравнения Беллмана принадлежит А. Г. Ченцову. Во всех этих работах Я. В. Салий писал реализацию алгоритма и генератор экземпляров задачи, проводил вычислительный эксперимент; в работах [VIII,IX,III,XI] также разрабатывал и реализовывал схему параллелизации алгоритма для систем с общей памятью; в работе [III] Я. В. Салию также принадлежит модель функции стоимости перемещений ремонтной бригады, допускающей экстренный вызов на не пройденный объект.

В работе [V] Я. В. Салий разработал обновленный, более эффективный вариант ДП для рассматриваемой задачи, реализовал его на C++ и испытал на наборе тестовых экземпляров; кроме того, реализовал на Haskell программу расчета количества состояний этого алгоритма и реализовал на C++ генератор экземпляров рассматриваемой задачи. Соавтор Е. Е. Иванко реализовал эвристику (жадный алгоритм) и испытал ее на наборе тестовых задач.

## Основное содержание диссертации

### Глава 1

Здесь фиксируются стандартные обозначения, вводятся специальные определения, описывающие конструкции ДП и позволяющие применять элементы теории частично упорядоченных множеств для качественного анализа рассматриваемой модели перемещений с условиями предшествования. Доказываются результаты, связанные с оценками пространственной и временно́й сложности решения (PSD)-TSP-PC — теоремы 1–2, леммы 1–2, предложения 1–4; последние даны с учетом исправлений [XIII] некоторых неточностей, допущенных автором в [XII]. Описываются теоретико-порядковые характеристики экземпляров задачи TSP-PC (то же «SOP») из TSPLIB, представляются оценки объема оперативной памяти, необходимого для их решения точным ДП; последние апробируются вычислительным экспериментом средствами разработанного автором программного комплекса. Также рассматривается эвристика усеченного ДП и решение одного варианта TSP-PC с зависимостью от времени (формализованного в виде задачи с зависимостью от списка невыполненных заданий) в сравнении с оригинальной работой [19].

Символ $\triangleq$ обозначает *равенство по определению*; *неупорядоченные пары* объектов записываются в фигурных скобках $\{a, b\}$, *упорядоченные пары* (УП) обозначаются круглыми скобками, $(a, b)$. Более двух упорядоченных объектов также пишем в круглых скобках, напр., $(a_1, \ldots, a_k)$, либо в виде кортежа $(a_i)_{i=1}^k$; *множество* элементов кортежа вводим так: $\overline{(a_i)_{i=1}^k} \triangleq \bigcup_{i \in 1..k}\{a_i\}$.

Символ *вложения* $\subset$ понимается в *нестрогом* смысле. Семейство всех (всех непустых) подмножеств некоторого $K$ обозначаются $\mathcal{P}(K)$ $(\mathcal{P}'(K))$. Под интервалом *целых* чисел понимается $p \mathinner{..} q \triangleq \{a \in \mathbb{Z} \,|\, (a \geqslant p) \wedge (a \leqslant q)\}$.

Фиксируем натуральное $n$ и интервал $1 \mathinner{..} n$; пусть на нем определено отношение частичного порядка $<_P$. Операторы взятия множества *максимальных* и *минимальных* элементов некоторого $K$, $K \subset 1 \mathinner{..} n$, в виду отношения частичного порядка $<_P$, определим так: $\mathrm{Max}[K] \triangleq \{m \in K \mid \forall k \in K (m \geqslant_P k) \vee (m \parallel k)\}$; $\mathrm{Min}[K] \triangleq \{m \in K \mid \forall k \in K (m \leqslant_P k) \vee (m \parallel k)\}$; здесь $\parallel$ — символ *несравнимости* в $<_P$, $a \parallel b \equiv \neg(a \leqslant_P b) \wedge \neg(b \leqslant_P a)$.

В контексте отношения порядка $<_P$, определим семейство *порядковых* идеалов $\mathcal{I} \triangleq \{I \in \mathcal{P}(1 \mathinner{..} n) \mid \forall i \in I\ \forall j \in 1 \mathinner{..} n\ (j \leqslant_P i) \to (j \in I)\}$ и упорядочим его по мощности, $\forall k \in 0 \mathinner{..} n\ \mathcal{I}_k \triangleq \{I \in \mathcal{I} \mid |I| = k\}$. Для порождения всех идеалов, используем оператор существенного продолжения $\mathbf{E}$, $\mathbf{E}\colon \mathcal{I} \setminus 1 \mathinner{..} n \to \mathcal{P}'(1 \mathinner{..} n)$, определенный по правилу $\mathbf{E}[I] \triangleq \{m \in 1 \mathinner{..} n \setminus I \mid I \cup \{m\} \in \mathcal{I}\}$.

**Теорема 1** (О характеризации $\mathbf{E}$, [XII])**.** *Для всех $I \in \mathcal{I} \setminus 1 \mathinner{..} n$, $m \in 1 \mathinner{..} n \setminus I$, эквивалентны условия* (i) $I \cup \{m\} \in \mathcal{I}$; (ii) $m \in \mathrm{Min}[1 \mathinner{..} n \setminus I]$.

Двойственный $\mathbf{E}$ оператор «допустимых точек выхода из идеала» $\mathbf{I}\colon \mathcal{I} \setminus \{\varnothing\} \to \mathcal{P}'(1 \mathinner{..} n)$ определяется следующим образом: $\mathbf{I}[I] \triangleq \{m \in I \mid I \setminus \{m\} \in \mathcal{I}\}$; для него справедлива аналогичная характеризация,

**Теорема 2** (О характеризации $\mathbf{I}$, [XII])**.** *Для всех $I \in \mathcal{I} \setminus \{\varnothing\}$ эквивалентны условия* (i) $I \setminus \{m\} \in \mathcal{I}$; (ii) $m \in \mathrm{Max}[I]$.

Пусть $n \in \mathbb{N}$ обозначает количество городов, которые должен обойти агент. В их число *не включаются* фиксированные база $0$ и «терминал» $\mathbb{t} \triangleq n + 1$; обозначения $\mathbb{t}$ и $n + 1$ равнозначны и взаимозаменяемы.

Затраты на перемещение между городами описываются *функцией стоимости* $\mathfrak{c}$, $\mathfrak{c}\colon \mathcal{P}(1 \mathinner{..} n) \times 0 \mathinner{..} n \times 1 \mathinner{..} n+1 \to \mathbb{R}$. Выражение $\mathfrak{c}(K, a, b)$ определяет затраты на перемещение из $a$ в $b$ при условии что во всех городах из $K$ агент уже побывал после выхода из базы $0$. Общие, *агрегированные* затраты на посещение *всех* городов (стоимость решения) — целевая функция — определяются в виде функции маршрута $\alpha$,

$$\mathfrak{C}[\alpha] \triangleq \mathfrak{c}(\varnothing, 0, \alpha_1) \oplus \mathfrak{c}(\{\alpha_1\}, \alpha_1, \alpha_2) \oplus \ldots \oplus \mathfrak{c}(1 \mathinner{..} n \setminus \{\alpha_n\}, \alpha_{n-1}, \alpha_n) \oplus (1 \mathinner{..} n, \alpha_n, \mathbb{t});$$

здесь $\oplus\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ — операция *агрегирования затрат*.

Условия предшествования задаем строгим *частичным порядком* $P \triangleq (1 \mathinner{..} n, <_P)$ на множестве городов. Множество *допустимых маршрутов* определяем так: $\mathbb{A} \triangleq \{\alpha \in (\mathrm{bi})[1 \mathinner{..} n] \mid \forall a, b \in 1 \mathinner{..} n\ (a <_P b) \to (\alpha_a^{-1} < \alpha_b^{-1})\}$.

Сама задача PSD-TSP-PC состоит в отыскании *допустимого* маршрута, доставляющего *минимум* стоимости.

*Состояниями* ДП полагаем *упорядоченные пары* вида $(I, x)$, где $I$ — множество городов, *уже посещенных* агентом после выхода из базы 0, также называемое *списком заданий*, а $x \in 1\,..\,n+1 \setminus I$ обозначает город, где агент в данный момент находится. Множество всех состояний: $\mathcal{S} \triangleq \{(I,x) \in \mathcal{I} \setminus \{1\,..\,n\} \times 1\,..\,n \mid x \in \mathbf{E}[I]\} \cup \{(1\,..\,n, \mathbb{t})\}$; стратифицируем его по мощности списков заданий, $\mathcal{S}_k \triangleq \{(I,x) \in \mathcal{S} \mid |I| = k\}$. Каждый элемент этого разбиения назовем *слоем.*

*Цена* (значение) состояния $v$ — оптимальное значение *целевой функции* $\mathfrak{C}$ на подзадаче, соответствующей данному состоянию, определяется следующим образом: в тривиальном случае $(I,x) \in \mathcal{S}_0$, положим $v(I,x) \triangleq \mathfrak{c}(\varnothing, 0, x)$; в общем случае, $v(I,x) \triangleq \min_{\alpha \in \mathbb{A}_I}\{\mathfrak{c}(\varnothing, 0, \alpha_1) \oplus \ldots \oplus \mathfrak{c}(\overline{(\alpha_i)_{i=1}^{|I|-1}}, \alpha_{|I|-1}, \alpha_{|I|}) \oplus \mathfrak{c}(\overline{(\alpha_i)_{i=1}^{|I|}}, \alpha_{|I|}, x)\}$ здесь $\mathbb{A}_I$ — множество *допустимых* маршрутов на $I$, определяемое аналогично $\mathbb{A} = \mathbb{A}_{1..n}$.

Функция Беллмана определяет цену состояния через цены *покрываемых* состояний, $\mathrm{BF}(I,x) \triangleq \min_{m \in \mathbf{I}[I]} \big\{v(I \setminus \{m\}, m) \oplus \mathfrak{c}(I, m, x)\big\}$, и *корректность* ДП означает, что для каждого состояния функция Беллмана *совпадает* с его ценой (исключая начальные условия $\mathcal{S}_0$).

**Теорема 3** ([XIII]). *Если операция $\oplus$ монотонно не убывает по первому аргументу и левоассоциативна, то для всех $(I,x) \in \mathcal{S} \setminus \mathcal{S}_0$ выполнено*

$$v(I,x) = \mathrm{BF}(I,x) = \min_{m \in \mathbf{I}[I]} \big\{v(I \setminus \{m\}, m) \oplus \mathfrak{c}(I, m, x)\big\}. \qquad \text{(BF)}$$

Псевдокод ДП для (PSD-)TSP-PC:

1: **Инициализация:** $\mathcal{I}_0 = \{\varnothing\}$, $\mathcal{S}_0 = \{(\varnothing, x), x \in \mathbf{E}[\varnothing]\}$,
2: $\forall(\varnothing, x) \in \mathcal{S}_0\ v(\varnothing, x) = \mathfrak{c}(\varnothing, 0, x)$, $\mathcal{I}_1 = \mathbf{E}[\varnothing]$.
3: **for all** $k \in 1\,..\,n-1$ **do**
4:     **for all** $I \in \mathcal{I}_k$ **do**
5:         `Вычислить` $\mathbf{E}[I]$
6:         **for all** $x \in \mathbf{E}[I]$ **do**
7:             `Добавить` $I \cup \{x\}$ `в` $\mathcal{I}_{k+1}$; `Добавить` $(I,x)$ `в` $\mathcal{S}_k$; `Вычислить` $v(I,x)$
8: `Вычислить` $v(1\,..\,n, \mathbb{t})$; `Восстановить решение по` $v[\mathcal{S}]$.

### Усеченное динамическое программирование

Усеченное ДП (restricted dynamic programming) — основанная на ДП эвристика, впервые предложенная в [36] для TD-TSP. Эвристика в некотором смысле обобщает жадный алгоритм: вместо того, чтобы сохранять, рассчитав BF, весь очередной слой $\mathcal{S}_i$ на шаге 7, в памяти вычислительного устройства сохраняются только $H \in \mathbb{N}$ *наилучших состояний*; при росте $H$ эвристика приближается к *точному* ДП. При $H = 1$ превращается в жадный алгоритм.

### Пространственная и временная сложность динамического программирования в (PSD)-TSP-PC

**Предложение 1** ([XII]). *$|\mathcal{S}| \leqslant w|\mathcal{I}|$, где $w$ — ширина $P$ (мощность наибольшей по включению антицепи в $P$). Пространственная сложность* полиномиально *(по количеству городов $n$) зависит от количества идеалов $|\mathcal{I}|$.*

Для обсуждения *временной* сложности введем две леммы; пусть множество элементов, *предшествующих* $x$, есть $\Downarrow x \triangleq \{m \in 1\,..\,n \mid m <_P x\}\ \forall x \in 1\,..\,n$.

**Лемма 1** ([XIII]). *Для $K \in \mathcal{P}(1\,..\,n)$, $|K| = k$, справедливо $\mathrm{Max}[K] = K \setminus \bigcup_{x \in K} \Downarrow x$. Если $\Downarrow k$ рассчитаны заранее для всех $k \in 1\,..\,n$, то временная сложность расчета $\mathrm{Max}[\cdot]$ есть $\mathcal{O}(k)$.*

**Лемма 2** ([XIII]). *Для $K \in \mathcal{P}(1\,..\,n)$, $|K| = k$, справедливо $\mathrm{Min}[K] = K \setminus \bigcup_{x \in K} \Uparrow x$, где $\Uparrow x \triangleq \{m \in 1\,..\,n \mid m >_P x\}$. Если $\Uparrow k$ рассчитаны заранее для всех $k \in 1\,..\,n$, то временная сложность расчета $\mathrm{Min}[\cdot]$ есть $\mathcal{O}(k)$.*

**Лемма 3** ([XIII]). *Временная сложность порождения всех идеалов $\mathcal{I}$ по процедуре $\mathcal{I}_{k+1} = \{I \cup \{m\} \mid (I \in \mathcal{I}_k) \wedge (m \in \mathbf{E}[I])\}$ составляет $\mathcal{O}(nw)$ на каждый идеал (итого, $\mathcal{O}(|\mathcal{I}| \cdot nw)$); данный результат улучшает известную оценку $\mathcal{O}(n^2)$ на идеал.*

**Предложение 2** ([XIII]). *Временная сложность процедуры ДП для (PSD)-TSP-PC составляет $\mathcal{O}(|\mathcal{I}| \cdot nw)$; кроме того, временная сложность полиномиально зависит от пространственной.*

*Следствие* 1. Временная сложность *полиномиально* зависит от пространственной — числа состояний $|\mathcal{S}|$.

*Следствие* 2. *Пространственная* сложность *полиномиально* зависит от количества идеалов $|\mathcal{I}|$.

*Следствие* 3. Если число идеалов $|\mathcal{I}|$ *полиномиально* зависит от числа городов $n$, то точное ДП *полиномиально* по времени.

**Предложение 3** ([XII]). *Пространственная сложность усеченного ДП не превышает $H(n+1)$.*

**Предложение 4** ([XIII]). *Временная сложность усеченного ДП не превышает $\mathcal{O}([H \cdot n \cdot \min\{\frac{n+1}{2}; w\}] \cdot (nw + \log_2 H))$.*

## Глава 2

Рассмотрены вопросы точного и эвристического вариантов ДП для наиболее сложной из рассматриваемых в диссертации моделей перемещений, PSD-BGTSP-PC, также обсуждается один подход к параллельной реализации точного ДП на основе систем с общей памятью. Соответствует пунктам 3–4 «основных результатов», на материале работ [I,III,IV,VIII,IX]. В оригинальных публикациях конструкции ДП описывались на языке фундированных отношений, аналогично [4]; стремясь к стилистическому единству диссертации и легкости сравнения построений гл. 1 и 2, автор «перевел» их на теоретико-порядковый язык. В отличие от гл. 1, описывается *попятная* процедура с существенными списками заданий-фильтрами.

Пусть $z = (a, b)$ — некоторая УП; обозначим $\mathrm{pr}_1(z)$ ее первый элемент, а $\mathrm{pr}_2(z)$ — второй. Фиксируем непустое множество $X$ и точку $x^0 \in X$ — *базу*, а также $n$ дизъюнктных конечных подмножеств $X$ — мегаполисы $M_1, \ldots, M_n$. Отношениями $\mathbb{M}_1 \in \mathcal{P}'(M_1 \times M_1), \ldots, \mathbb{M}_n \in \mathcal{P}'(M_n \times M_n)$ ограничим варианты внутренних работ в мегаполисах; если $(x_1, x_2) \in \mathbb{M}_i$, значит, допускается прибывать в мегаполис $M_i$ через город $x_1$ и отбывать из него через город $x_2$. С этими отношениями согласовывается *трасса* — кортеж из пар городов, описывающий конкретные пункты прибытия в и отправления из при движении по мегаполисам. Все возможные точки входа соберем в $\mathbb{X}_{\mathrm{in}}$, все возможные точки выхода, включая базу $x^0$ — в $\mathbb{X}_{\mathrm{out}}$.

Условия предшествия заданы строгим порядком на индексах мегаполисов $P = (1\,..\,n, <_P)$. Существенные списки заданий представим *порядковыми фильтрами* $\mathcal{F} \triangleq \{F \in \mathcal{P}(1\,..\,n) \mid \forall i \in F\ \forall j \in 1\,..\,n\ (j \geqslant_P i) \to (j \in F)\}$, связанными между собой операторами $\mathbf{E}[F] \equiv \mathrm{Max}[1\,..\,n \setminus F]$, $\mathbf{I}[F] \equiv \mathrm{Min}[F]$ (двойственны «тезкам» в гл. 1). Определение множества допустимых маршрутов $\mathbb{A}$ совпадает с гл. 1.

Введем $n+1$ функцию стоимости $\mathbb{c}\colon \mathbb{X}_{\mathrm{out}} \times \mathbb{X}_{\mathrm{in}} \times \mathcal{F} \to \mathbb{R}$; $c_i\colon \mathbb{M}_i \times \mathcal{F} \to \mathbb{R}$, $i \in 1\,..\,n$, где $\mathbb{c}(\cdot)$ отвечает за *внешние* перемещения, а $n$ «нумерованных» функций стоимости оценивают *внутренние работы*. Для краткости введем также «предагрегированную» функцию $\mathfrak{c}(x, z, K) \triangleq \mathbb{c}(x, \mathrm{pr}_1(z), K) + c_l(z, K)$, где $l$ — индекс мегаполиса, которому принадлежат города из УП $z$.

Качество решения PSD-BGTSP-PC, представленного маршрутом $\alpha \in \mathbb{A}$ и трассой $(z_i)_{i=0}^n$ — кортежем пар городов из мегаполисов, расположенных в порядке, заданным $\alpha$, определяется следующим критерием: $\mathfrak{C}^{(\alpha)}[(z_i)_{i=0}^n] = \max_{t \in 0..n-1}[\mathfrak{c}(\mathrm{pr}_2(z_t), z_{t+1}, \overline{(\alpha_i)_{i=t+1}^n})]$.

Множество подзадач-состояний ДП определяется в виде $\mathcal{S} \triangleq \{(x, F) \in \mathbb{X}_{\mathrm{out}} \times \mathcal{F} \mid (\exists m \in \mathbf{E}[F] : x \in \mathbb{M}_m^{(\mathrm{out})})\} \cup \{(x^0, 1\,..\,n)\}$, где $\mathbb{M}_m^{(\mathrm{out})}$ — множество допустимых точек выхода из $M_m$. Корректность ДП обеспечивается следующей теоремой, где $v(x, K)$ — «наилучшее решение», экстремум состояния $(x, K)$.

**Теорема 4** ([III]). *Для $(x, K) \in \mathcal{S} \setminus \mathcal{S}_0$ справедлив* ***принцип оптимальности:*** $v(x, K) = \min_{j \in \mathbf{I}[K]} \min_{z \in \mathbb{M}_j} \max\{\mathfrak{c}(x, z, K);\ v(\mathrm{pr}_2(z), K \setminus \{j\})\}$.

В схеме параллелизации под OpenMP [IV] для каждого существенного списка $K$ в один блок `task` входит (1.) вычисление $\mathbf{E}[K]$; (2.) порождение «продолжений» $K$ по схеме $K \cup \{j\}$, где $j \in \mathbf{E}[K]$; (3.) расчет $v(x, K)$, где $(x, K)$ — состояния, соответствующие $K$. Для исключения условий гонки (race conditions) операции записи в структуры данных на этапах (2.) и (3.) помечены директивой `omp critical`.

Усеченное ДП аналогично представленному в гл. 1.

**Предложение 5** ([IV]). *Временная сложность эвристики усеченного ДП имеет порядок $\mathcal{O}(n^2 w^2 m^2 t_{\mathrm{BF}} \cdot H \log_2 H)$, где $w$ — ширина $<_P$, $m$ — максимальное количество городов в мегаполисе, $H$ — параметр эвристики, а $t_{\mathrm{BF}}$ — постоянное время, требуемое на расчет* $\max\{\mathfrak{c}(x, (z_{\mathrm{in}}, z_{\mathrm{out}}), K); v(z_{\mathrm{out}}, K \setminus \{i\})\}$.

## Глава 3

## Задача о перестановке однотипных объектов

Впервые поставлена в [40] для приложений к полевой работе специалистов-биологов, изучающих динамику популяций некоторых видов животных, суть очень слабый вариант 1-Pickup and Delivery TSP [41]. Фиксируем число $n \in \mathbb{N}$ камер-ловушек. Множество *всех городов* обозначим $X \triangleq 0\,..\,2n = \{0\} \cup \mathcal{R} \cup \mathcal{B}$, где множество $\mathcal{R} \triangleq 1\,..\,n$ содержит индексы точек, где размещены камеры («красные»), а множество $\mathcal{B} \triangleq n+1\,..\,2n$ — индексы точек, куда их нужно переставить («синие»). Стоимость перемещения задается функцией $\mathfrak{c}\colon X \times X \to \mathbb{R}$.

*Синие* точки можно посещать только неся хотя бы одну камеру «в рюкзаке»; допустимые маршруты введем с помощью «оператора допустимой точки выхода» $\mathbf{I}\colon \mathcal{P}(X) \to \mathcal{P}(X)$, такого что для множества $K$ посещенных городов, если в нем красных *больше*, чем синих, то $\mathbf{I}[K] = K$; если красных и синих *поровну*, то $\mathbf{I}[K] = K \cap \mathcal{B}$. Множество допустимых маршрутов есть $\mathbb{A} \triangleq \{\alpha \in \mathbb{P} \mid \forall s \in 1\,..\,2n\ \alpha(s) \in \mathbf{I}[\overline{(\alpha_t)_{t=1}^{s}}]\}$. Качество маршрута $\alpha$ определяет целевая функция $\mathfrak{C}[\alpha] \triangleq \{\mathfrak{c}(0, \alpha_1) + \sum_{i=1}^{2n-1} \mathfrak{c}(\alpha_i, \alpha_{i+1}) + \mathfrak{c}(\alpha_{2n}, 0)\}$.

Список заданий $K \subset 1\,..\,2n$ *допустим* если синих точек в нем *не меньше* чем красных; обозначим множество таких списков $\mathcal{G}$. Введем *оператор допустимого продолжения* $\mathbf{E}\colon \mathcal{G} \to X$ так: если в $K$ красных *больше*, чем синих, то $\mathbf{E}[K] = X \setminus K$; если *поровну*, то $\mathbf{E}[K] = (X \setminus K) \cap \mathcal{R}$.

Тогда состояния ДП можно определить как $\mathcal{S} \triangleq \{(K,x) \in \mathcal{P}(X) \times X \mid K \in \mathcal{G} \setminus \mathcal{G}_{2n},\ x \in \mathbf{E}[K]\} \cup \{(1\,..\,2n, 0)\}$, и уравнение Беллмана выглядит следующем образом: н. у. $v(\varnothing, x) \triangleq \mathfrak{c}(0,x)$ для $x \in \mathcal{R}$ и общий шаг $v(K,x) \triangleq \min_{y \in \mathbf{I}[K]}\{v(K \setminus \{y\}, y) + \mathfrak{c}(y,x)\}$ для $(K,x) \in \mathcal{S} \setminus \mathcal{S}_0$; его корректность следует из корректности аналогичного уравнения в [40].

### Ультраметрическая BTSP

*Ультра*метрикой называется метрика $\mathfrak{c}$, удовлетворяющая дополнительно к обычным аксиомам еще *усиленному неравенству треугольника*, $\forall a,b,c \in 1\,..\,n\ \mathfrak{c}(a,c) \leqslant \max\{\mathfrak{c}(a,b); \mathfrak{c}(b,c)\}$. Известен более общий результат о характеризации ультраметрик [42, Pr. 3.1], однако о связи с BTSP в [42] не упомянуто; доказательство в [VII] получено независимо.

**Теорема** ([VII]). *Пусть* $(p^*, q^*) \triangleq \operatorname{argmax}_{i \in 0..n, j \in 1..n \setminus \{i\}}\{\mathfrak{c}(i,j)\}$. *В любом* $\alpha \in (\mathrm{bi})[1\,..\,n]$ *найдется перемещение, вес которого равен* $\mathfrak{c}(p^*, q^*)$, *т. о. целевая функция постоянна и равна* $\mathfrak{c}(p^*, q^*)$ *для всех решений* BTSP.

## Глава 4. Программный комплекс

Комплекс реализован на C++11, обеспечивает точное (ДП) и эвристическое (усеченное ДП) решения задач типа TSP-PC и GTSP-PC с различными вариантами агрегирования затрат, в частности, арифметическая сумма реализует TSP-PC и ее обобщенные варианты, max реализует BTSP-PC и ее обобщенные варианты; также доступна TD-TSP-PC с зависимостью типа *traveling deliveryman* [19]. Комплекс может применяться в 64-битных средах под управлением ОС семейств Windows (компилятор Microsoft Visual C++ 12.0) и Linux (компилятор gcc 6.3.0); межплатформенная совместимость достигается условной компиляцией через директивы препроцессора C.

Универсальность по разновидностям решаемых задач достигается за счет разделения «геометрической» части (расстояние между городами, условия предшествования) и признаков специфической задачи (конкретная разновидность функции агрегирования затрат). Также, благодаря конструкциям ДП из [XII,XIII], обеспечена «прозрачность по направлению решения»: за исключением н. у. и определений операторов **E** и **I**, решение как в прямом так и в попятном направлении производит общая кодовая база, что упрощает обслуживание программного кода; для усеченного ДП, в зависимости от конкретного экземпляра задачи, одно из направлений может оказаться более предпочтительным [XIII]. Элементы ООП позволяют частично объединить программный кода точного и усеченного ДП: вычисление значений функции Беллмана и восстановление оптимального маршрута по ней унифицированы. При решении задач комплекс замеряет истекшее (физическое) время с точностью до 1 мс посредством `steady_clock::now()` из библиотеки C++ `<std::chrono>` и расход оперативной памяти с помощью интерфейса к ОС.

## Заключение

### Итоги и рекомендации

**Точное решение. Динамическое программирование.** Исследования [XII,XIII] демонстрируют пригодность рассматриваемого варианта точного ДП как метода решения «сильно нагруженных» (условиями предшествования) задач на примере решения экземпляров из TSPLIB, как в обычной постановке TSP-PC, так и в постановке с зависимостью от времени [19, TD-SOP], и конкурентоспособность с методом программирования в ограничениях (constraint programming) [19].

Можно рекомендовать решать TSP-PC и PSD-TSP-PC посредством ДП согласно критерию из [XII,XIII]: (1.) если оценка *сверху* $\log_2|\mathcal{S}|$ не превосходит 40, экземпляр наверняка удастся разрешить (затратив не более 256 ГБ оперативной памяти); (2.) если оценка *снизу* $\log_2|\mathcal{S}|$ превосходит 40, экземпляр наверняка *не* удастся разрешить.

Отметим ключевые особенности ДП: (а.) не требует использования коммерческих пакетов для решения задач линейного программирования (например, IBM ILOG CPLEX); (б.) хорошо справляется с дополнительными ограничениями и обобщениями, как то условия предшествования, обобщенность, зависимость от времени и др.; (в.) более универсально чем другие методы: модель с абстрактной функцией агрегирования позволяет единообразно и со сравнимыми затратами решать как обычные, аддитивные задачи, так и задачи с агрегированием *на узкие места*; (г.) *полиномиально* по количеству идеалов, следовательно, (PSD)-TSP-PC *полиномиально* разрешима при полиномиальной зависимости $|\mathcal{I}|$ от количества городов $n$.

**Эвристика «усеченное динамическое программирование».** Эвристика достаточно эффективна для задач с «сильными» и «средними» условиями предшествования. Зачастую достигает оптимального решения много быстрее точного ДП и с существенно меньшим расходом памяти. Особенности: (а.) всегда находит *допустимое* решение (нехарактерно для эвристик); (б.) полиномиальна по $H$ и $n$; (в.) решение стремится к точному при приближении $H$ к $\operatorname{argmax}_{i\in 0..n}\{|\mathcal{S}_i|\}$.

### Перспективы дальнейших исследований

**Точное решение (PSD)-(B)(G)TSP-PC.** Предполагаются два общих подхода к разрешению задач *большей* размерности: (а.) *экстенсивный*: использовать «вертикальную» схему параллелизации с *распределенной* памятью [43] для задействования большего объема, чем доступен на одном (суперкомпьютерном) узле; (б.) *интенсивный*: найти эффективную оценку *снизу* и реализовать схему ветвей и границ в ДП [7, 44].

**Эвристика «усеченное динамическое программирование».** Представляется полезными гарантированные оценки точности, связанные с соотношением параметра эвристики $H$ и мощностью «наиболее населенного» слоя пространства состояний, возможно, на стохастических моделях[7].

**Оценки сложности экземпляров TSP-PC.** Представляется ценным (а.) дополнить оценки [XII] верхними оценками из разложений в цепи. Получить оценки размерности экземпляров TSP-PC в виде количества допустимых маршрутов (линейных продолжений); (б.) определить «оптимальное» разложение в цепи; исследовать связь пространственной сложности и иных теоретико-порядковых характеристик (в дополнение к ширине). Получить оценки на количество теоретико-порядковых идеалов заданной мощности. Формально описать верхнюю границу теоретического прироста производительности «вертикальной» схемы распараллеливания [43].

**Создание библиотеки экземпляров GTSP-PC.** Единственная на настоящий момент библиотека экземпляров GTSP-PC представлена в [47]. Можно предложить несколько вариантов ее пополнения: (1.) кластеризация экземпляров TSP-PC из TSPLIB и SOPLIB с сохранением имеющихся условий предшествования внутри кластеров и выявлением условий предшествования между кластерами исходя из условий на городах; (2.) наделение случайными условиями предшествования экземпляров GTSP из библиотеки `http://www.cs.nott.ac.uk/~pszdk/gtsp.html`.

## Список цитируемой литературы


1. The traveling salesman problem and its variations / Ed. by G. Gutin, A. P. Punnen. Dordrecht: Kluwer Academic Publishers, 2002. Vol. 12 of *Combinatorial optimization*.

2. Steiner G. On the complexity of dynamic programming for sequencing problems with precedence constraints // Annals of Operations Research. 1990. Vol. 26, no. 1. P. 103–123.


[7]например, по аналогии с оценками для жадного алгоритма в TSP [45], [46, § 4.2]

3. Escudero L. F. An inexact algorithm for the sequential ordering problem // European Journal of Operational Research. 1988. Vol. 37, no. 2. P. 236–249.
4. Ченцов А. Г. Экстремальные задачи маршрутизации и распределения заданий: вопросы теории. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2008.
5. Schmidt G., Ströhlein T. Relations and graphs: discrete mathematics for computer scientists. EATCS Monographs on Theoretical Computer Science. Berlin: Springer-Verlag, 1993.
6. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Вопросы теории // Автоматика и телемеханика. 1989. № 9. С. 3–33.
7. The Traveling Salesman Problem with Precedence Constraints / L. Bianco, A. Mingozzi, S. Ricciardelli et al. // Papers of the 19th Annual Meeting/Vorträge der 19. Jahrestagung / Springer. 1992. P. 299–306.
8. Gouveia L., Ruthmair M. Load-dependent and precedence-based models for pickup and delivery problems // Computers & Operations Research. 2015. Vol. 63. P. 56–71.
9. Balas E., Fischetti M., Pulleyblank W. R. The precedence-constrained asymmetric traveling salesman polytope // Mathematical programming. 1995. Vol. 68, no. 1-3. P. 241–265.
10. Shobaki G., Jamal J. An exact algorithm for the sequential ordering problem and its application to switching energy minimization in compilers // Computational Optimization and Applications. 2015. Vol. 61, no. 2. P. 343–372.
11. Cire A. A., van Hoeve W.-J. Multivalued decision diagrams for sequencing problems // Operations Research. 2013. Vol. 61, no. 6. P. 1411–1428.
12. Lawler E. L. Efficient implementation of dynamic programming algorithms for sequencing problems: Tech. Rep.: BW 106/79: Stichting Mathematisch Centrum, 1979.
13. Ченцов А. Г., Ченцов П. А. Маршрутная задача с условиями предшествования (задача курьера): метод динамического программирования // Вестн. УГТУ-УПИ. 2004. № 15. С. 148–151.
14. Сесекин А. Н., Ченцов А. А., Ченцов А. Г. Маршрутизация с абстрактной функцией агрегирования стоимостей перемещений // Труды Института математики и механики УрО РАН. 2010. Т. 16, № 3. С. 240–264.
15. Kouvelis P., Yu G. Robust discrete optimization and its applications. Dordrecht: Kluwer Academic Publishers, 1997. Vol. 14 of *Nonconvex Optimization and Its Applications*.
16. On the maximum scatter traveling salesperson problem / E. M. Arkin, Y.-J. Chiang, J. S. Mitchell et al. // SIAM Journal on Computing. 1999. Vol. 29, no. 2. P. 515–544.
17. Петунин А. А. О некоторых стратегиях формирования маршрута инструмента при разработке управляющих программ для машин термической резки материала // Вестник Уфимского государственного авиационного технического университета. 2009. Т. 13, № 2. С. 280–286.
18. Gendreau M., Ghiani G., Guerriero E. Time-dependent routing problems: A review // Computers & Operations Research. 2015. Vol. 64. P. 189–197.
19. Kinable J., Cire A. A., van Hoeve W.-J. Hybrid optimization methods for time-dependent sequencing problems // European Journal of Operational Research. 2017. Vol. 259, no. 3. P. 887–897.

20. Koulamas C., Kyparisis G. J. Single-machine scheduling problems with past-sequence-dependent setup times // European Journal of Operational Research. 2008. Vol. 187, no. 3. P. 1045–1049.

21. Alkaya A. F., Duman E. A new generalization of the traveling salesman problem // Appl. Comput. Math. 2010. Vol. 9, no. 2. P. 162–175.

22. Chisman J. A. The clustered traveling salesman problem // Computers & Operations Research. 1975. Vol. 2, no. 2. P. 115–119.

23. Ченцов А. Г. К вопросу о маршрутизации комплексов работ // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2013. № 1. С. 59–82.

24. Lokin F. Procedures for travelling salesman problems with additional constraints // European Journal of Operational Research. 1979. Vol. 3, no. 2. P. 135–141.

25. Castelino K., D'Souza R., Wright P. K. Toolpath optimization for minimizing airtime during machining // Journal of Manufacturing Systems. 2003. Vol. 22, no. 3. P. 173–180.

26. An industrially validated CMM inspection process with sequence constraints / R. Salman, J. S. Carlson, F. Ekstedt et al. // Procedia CIRP. 2016. Vol. 44. P. 138–143.

27. LaRusic John, Punnen Abraham P. The asymmetric bottleneck traveling salesman problem: Algorithms, complexity and empirical analysis // Computers & Operations Research. 2014. T. 43. C. 20–35.

28. Коротаева Л. Н., Трухин М. П., Ченцов А. Г. К вопросу о маршрутизации соединений // Автоматика и телемеханика. 1997. № 12. С. 175–192.

29. Ченцов А. А., Ченцов А. Г. Экстремальная задача маршрутизации "на узкие места" с ограничениями в виде условий предшествования // Тр. Ин-та математики и механики УрО РАН. 2008. Т. 14, № 2. С. 129–142.

30. Сесекин А. Н., Ченцов А. А., Ченцов А. Г. Об одной задаче маршрутизации "на узкие места" // Труды Института математики и механики УрО РАН. 2010. Т. 16, № 1. С. 152–170.

31. Kalantari B., Hill A. V., Arora S. R. An algorithm for the traveling salesman problem with pickup and delivery customers // European Journal of Operational Research. 1985. Vol. 22, no. 3. P. 377–386.

32. Brightwell G., Winkler P. Counting linear extensions // Order. 1991. Vol. 8, no. 3. P. 225–242.

33. Григорьев А. М., Иванко Е. Е., Ченцов А. Г. Динамическое программирование в обобщенной задаче курьера с внутренними работами: элементы параллельной структуры // Моделирование и анализ информационных систем. 2011. Т. 18, № 3. С. 101–124.

34. Balas E., Simonetti N. Linear time dynamic-programming algorithms for new classes of restricted TSPs: A computational study // INFORMS Journal on Computing. 2001. Vol. 13, no. 1. P. 56–75.

35. Chentsov A., Khachay M., Khachay D. Linear time algorithm for Precedence Constrained Asymmetric Generalized Traveling Salesman Problem // IFAC-PapersOnLine. 2016. Vol. 49, no. 12. P. 651–655.

36. Malandraki C., Dial R. B. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem // European Journal of Operational Research. 1996. Vol. 90, no. 1. P. 45–55.

37. Hernádvölgyi I. T. Solving the sequential ordering problem with automatically generated lower bounds // Operations Research Proceedings 2003. Springer, 2004. P. 355–362.

38. Gouveia L., Pesneau P. On extended formulations for the precedence constrained asymmetric traveling salesman problem // Networks. 2006. Vol. 48, no. 2. P. 77–89.

39. Ченцов А. А., Ченцов А. Г., Ченцов П. А. Элементы динамического программирования в экстремальных задачах маршрутизации // Проблемы управления. 2013. № 5. С. 12–21.

40. Иванко Е. Е. Динамическое программирование в задаче перестановки однотипных объектов // Труды Института математики и механики УрО РАН. 2013. Т. 19, № 4. С. 125–130.

41. Hernández-Pérez H., Salazar-González J.-J. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery // Discrete Applied Mathematics. 2004. Vol. 145, no. 1. P. 126–139.

42. Leclerc B. Description combinatoire des ultramétriques // Mathématiques et Sciences humaines. 1981. Vol. 73. P. 5–37.

43. Chentsov A. G., Grigoryev A. M. A Scheme of Independent Calculations in a Precedence Constrained Routing Problem // International Conference on Discrete Optimization and Operations Research / Springer. 2016. P. 121–135.

44. Morin T. L., Marsten R. E. Branch-and-bound strategies for dynamic programming // Operations Research. 1976. Vol. 24, no. 4. P. 611–627.

45. Rosenkrantz D. J., Stearns R. E., Lewis P. M. An Analysis of Several Heuristics for the Traveling Salesman Problem // SIAM Journal on Computing. 1977. Vol. 6, no. 3. P. 563–581.

46. Гимади Э. Х., Хачай М. Ю. Экстремальные задачи на множествах перестановок. Екатеринбург: Издательство УМЦ УПИ, 2016.

47. Salman R. Algorithms for the Precedence Constrained Generalized Travelling Salesperson Problem: Master's thesis: Chalmers University of Technology. University of Gothenburg. 2015.

## Список работ, опубликованных автором по теме диссертации

Основные результаты по теме диссертации изложены в 13 публикациях, 3 из которых изданы в журналах, рекомендованных ВАК [I,II,III], 2 — в виде статьей в трудах конференций, индексируемых системой Scopus [IV,V]. На программные комплексы, использованные в исследованиях, отраженных в диссертации, получено 3 свидетельства о гос. регистрации программ для ЭВМ (Роспатент).

## Статьи, опубликованные в изданиях, определенных ВАК

I Ченцов А. Г., Салий Я. В., Об одной маршрутной задаче на узкие места с внутренними работами // Вестник Тамбовского Университета (сер. Естественные и технические науки), 2012. Т. 17. № 3. С. 827–847. (2,625 п.л. / 0,25 п.л.)

II Салий Я. В., Влияние условий предшествования на вычислительную сложность решения маршрутных задач методом динамического программирования // Вестн. Удмурт. унив. Мат. Мех. Компьют. науки, 2014. № 1. С. 76–86. (1,25 п.л.)

III Chentsov A. G., Salii Ya. V., A model of "nonadditive" routing problem where the costs depend on the set of pending tasks // Vestnik YuUrGU. Ser. Mat. Model. Progr., 2015. V. 8. no. 1. P. 24–45. (Scopus) (2,75 п.л. / 0,15 п.л.)

IV Salii Ya., Restricted Dynamic Programming Heuristic for Precedence Constrained Bottleneck Generalized TSP // Proc. of the 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists (Yekaterinburg, November 17th, 2015), CEUR Workshop Proceedings. V. 1513. P. 85–108. http://ceur-ws.org/Vol-1513/#paper-10. (Scopus) (3 п.л.)

V Salii Ya. V., Ivanko E. E., Dynamic programming and greedy heuristic in Camera Trap Traveling Salesman Problem // Proc. of the 47th International Youth School-conference "Modern Problems in Mathematics and its Applications" (Yekaterinburg, Russia, January 31–February 6, 2016), CEUR Workshop Proceedings. V. 1662. P. 215–219. http://ceur-ws.org/Vol-1662/mpr5.pdf. (Scopus) (0,625 п.л. / 0,6 п.л.)

## Другие публикации

VI Ченцов А. Г., Салий Я. В., Об одной задаче на узкие места // Сб. тр. Всерос. конф. СМО-2011 (Челябинск, 28 ноября – 3 декабря 2011 г.). Челябинск: Изд. ЮУрГУ, 2011. С. 85–92. (0,5625 п.л. / 0,0625 п.л.)

VII Салий Я. В., Об ультраметрической задаче коммивояжера на узкие места // Тез. Междунар. (43-й Всерос.) мол. школы-конф. (Екатеринбург, 29 января – 5 февраля 2012 г.). Екатеринбург: Изд. УМЦ УПИ, 2012. С. 287–289. (0,1875 п.л.)

VIII Салий Я. В., Ченцов А. Г., О маршрутной задаче на узкие места с внутренними работами и условиями предшествования // Мат. конф. «Международная конференция "Дискретная оптимизация и исследование операций"» (Новосибирск, 24 – 28 июня 2013 г.), Новосибирск: Изд-во Ин-та математики, 2013. С. 134. (0,125 п.л. / 0,0625 п.л.)

IX Ченцов А. Г., Салий Я. В., Неаддитивная задача маршрутизации с условиями предшествования // Тезисы докладов Всерос. конф. с междунар. участием, посвященной памяти В. К. Иванова Алгоритмический анализ неустойчивых задач (Челябинск, 10 – 14 ноября 2014 года). Челябинск: Изд. ЮУрГУ, 2014. С. 166–167. (0,125 п.л. / 0,0625 п.л.)

X Салий Я. В., О прямом и попятном динамическом программировании в маршрутных задачах с условиями предшествования и алгоритмах генерации допустимых подзадач // Инф. бюлл. Асс. мат. программ., 2015. № 13. С. 168–169. (0,125 п.л.)

XI Ченцов А. Г., Салий Я. В., Задача маршрутизации «на узкие места» с ограничениями и усложненными функциями стоимости // Мат. 8-й Всерос. мультиконф. (МКПУ-2015): в 3-х т. Ростов-на-Дону: Изд. ЮФУ, 2015. Т. 1. С. 214–216. (0,1875 п.л. / 0,0625 п.л.)

XII Salii Ya., Order-theoretic characteristics and dynamic programming for Precedence Constrained Traveling Salesman Problem // Proc. of the 4th Russian Finnish Symposium on Discrete Mathematics (Turku, May 16–19 2017).,Turku Centre for Computer Science. V. 26. P. 152–164. http://urn.fi/URN:ISBN:978-952-12-3547-4 (1,625 п.л.)

XIII Salii Y., Revisiting Dynamic Programming for Precedence-Constrained Traveling Salesman Problem and Its Time-Dependent Generalization // European Journal of Operational Research, 2018. DOI:`https://doi.org/10.1016/j.ejor.2018.06.003` (4 п.л.)

## Свидетельства о государственной регистрации программ для ЭВМ

XIV Салий Я. В., Программа вычисления оптимального решения обобщенной задачи курьера на узкие места с условиями предшествования и зависимостью от списка невыполненных заданий методом динамического программирования // Фед. сл. по инт. собств., (12) гос. рег. программы для ЭВМ. № 2015661435. дата регистрации 27.10.2015.

XV Салий Я. В., Программа расчета сложности задачи о маршрутизации инструмента в машинах листовой резки с целью сокращения холостого хода // Фед. сл. по инт. собств., (12) гос. рег. программы для ЭВМ. № 2015661434. дата регистрации 27.10.2015.

XVI Салий Я. В., Модульная программа вычисления оптимальных и эвристических решений задачи курьера и ее обобщений методом динамического программирования // Фед. сл. по инт. собств., (12) гос. рег. программы для ЭВМ. № 2018614740. дата регистрации 17.04.2018.