ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ ИМ. Н.Н. КРАСОВСКОГО УРАЛЬСКОГО ОТДЕЛЕНИЯ РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Григорьев Алексей Михайлович

НЕКОТОРЫЕ ЗАДАЧИ МАРШРУТИЗАЦИИ С ОГРАНИЧЕНИЯМИ И ФУНКЦИЯМИ СТОИМОСТИ, ЗАВИСЯЩИМИ ОТ СПИСКА ЗАДАНИЙ

Специальность 05.13.18 —
Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
д. ф.-м. н., член-корреспондент РАН
Ченцов Александр Георгиевич

Екатеринбург — 2021

Оглавление

Стр.

Введение

Актуальность и степень разработанности.

В диссертации рассматриваются задачи маршрутизации перемещений, имеющие своим источником ситуации прикладного характера, возникающие в атомной энергетике при выполнении набора заданий, связанных с демонтажом радиационно опасных элементов. Отметим, что подобные постановки возникают и в других содержательных задачах. Среди них выделим особо задачу управления инструментом при листовой резке деталей на машинах с ЧПУ. Хотя исследуемые задачи маршрутизации, ориентированные на инженерные применения, имеют своим прототипом известную труднорешаемую задачу коммивояжера (TSP в англоязычной литературе) [1–12], в них возникают существенные особенности не только количественного, но и качественного характера: ограничения, усложненные функции стоимости, многовариантности возможных перемещений и др. Исследования в области решений т.н. обобщенной задачи коммивояжера (GTSP) [1; 13–16] также не покрывают потребности интересующих нас инженерных приложений, хотя и учитывают многовариантности возможных перемещений (объектами посещения являются кластеры или мегаполисы, а не города, как в TSP).

Полезно отметить задачу коммивояжера с условиями предшествования (TSP-PC) (см. [17–22] и др.), где, в частности, исследовалось влияние условий предшествования на сложности вычислений. Отметим, что в русскоязычной литературе используется термин "задача курьера" (см. [3–5; 23]). Отметим ряд работ, посвящённых TSP-PC, ориентированных на приложения; см. [24] (инспекция нефтяных вышек), [18; 25–28] (оптимизация производственных процессов), организация складской деятельности [29]. Отметим работы [30; 31], использующие элементы метода ветвей и границ для решения TSP-PC. В связи с применением для решения TSP динамического программирования (ДП) отметим особо работы [7; 8; 32]. Развиваемая в настоящей работе конструкция восходит в идейном отношении к подходу [7], имеющему смысл понятной процедуры.

В настоящем обзоре мы воздержимся от обсуждения работ, связанных с решением задач маршрутизации с неаддитивным агрегированием затрат, огра-

ничиваясь [33–35]. В задачах, исследуемых в настоящей работе, критерий качества предполагается аддитивным.

В связи с производственными задачами, приводящими к идеям маршрутизации с обобщениями, отметим исследование, посвящённое вопросам монтажа печатных плат [36]. Здесь же отметим работу [37], которая относится к GTSP-PC и связана с инженерной задачей обработки листа (сверление, развертывание отверстий, нарезка резьбы).

В настоящей работе исследуется комплекс вопросов, связанных с применением параллельных алгоритмов для решения задач маршрутизации, ориентированных на инженерные приложения, связанные с атомной энергетикой. Речь идет, в первую очередь, о вопросах, связанных со снижением облучаемости персонала АЭС и специалистов аварийно-спасательных формирований при ликвидации аварийных ситуаций, возникающих на АЭС, в которых существенную роль имеет маршрутизация выполняемых заданий. Дело в том, что снижать облучаемость можно не только с помощью различных защитных сооружений, но и за счет разумной организации последовательности выполняемых операций (в работе приведен соответсвующий пример). В этой связи задачи маршрутизации настоящей работы могут представлять не только теоретический, но и серьезный практический интерес в плане снижения радиационного воздействия на исполнителей операций по демонтажу радиационно опасных объектов.

В то же время сама реализация упомянутых возможностей по снижению радиационного воздействия связана с решением очень сложных задач маршрутизации, имеющих лишь весьма отдалённое сходство со своим прототипом – задачей коммивояжера или TSP, хотя известные вопросы, связанные с трудно-решаемостью TSP (см. [38]), в полной мере сохраняют свое значение и в задачах, рассматриваемых в настоящей работе. Речь идет о комбинаторных задачах с ограничениями и усложненными функциями стоимости, которые допускают зависимость от списка заданий, не выполненных на текущий момент. Кроме того, сами упомянутые зависимости в типичных случаях включают при своем построении интегрирование нелинейных зависимостей и комбинирование интегральных эффектов, создаваемых отдельными излучателями.

Возникающая при должной формализации постановка выходит за пределы круга задач, исследуемых в дискретной оптимизации и включает объективно элементы задач управления с дискретным временем. В этой связи отметим,

что используемый в работе вариант ДП, восходящий к схеме Р. Беллмана [7], соответствует в идейном отношении процедурам на основе ДП, применяемым в теории управления; в этой связи отметим [39]. В дискретной оптимизации при решении TSP и задач типа TSP чаще используется вариант Хелда-Карпа; см. [8] (стоит отметить, что при решении самой исходной TSP и многих слабо осложненных задач, ДП используется крайне редко и в основном теоретически – для оценки вычислительной сложности).

Методология и методы исследования. Выбор ДП в качестве основного метода исследования (что нетипично для дискретной оптимизации) связан с определённой "всеядностью" процедур на основе ДП, что и позволило встраивать в эти процедуры элементы соблюдения ограничений и разнообразные условия функций стоимости. Таким образом, именно ДП позволяет (по крайней мере на теоретическом уровне) учитывать реальные интересы решения прикладных задач, что достигается, конечно, при соответствующей формализации исходной задачи с использованием ее разнообразных преобразований и должной теоретической проработки основных этапов исследования.

Целью диссертационной работы является исследование применения параллельных методов для решения ряда прикладных задач, возникающих в атомной энергетике.

Для достижения поставленной цели потребовалось решить следующие **задачи**:

1. Выполнить анализ модели перемещений с условиями предшествования и функциями стоимости, зависящими от списка невыполненных заданий, применительно к различным инженерным задачам.
2. Построить параллельный алгоритм поиска оптимального решения на основе ДП для задач маршрутизации и распределения заданий.
3. Разработать и реализовать комплекс программ, провести обширный вычислительный эксперимент.
4. Оценить влияние условий предшествования на степень распараллеливания предложенных алгоритмов ДП.

Положения, выносимые на защиту:

1. Предложен алгоритм "вертикального" распараллеливания (схема независимых вычислений [40]) ДП в системах с распределенной памятью

с применением программного интерфейса MPI, использован принцип разбиения задачи верхнего уровня на подзадачи меньшей размерности.

2. Реализован программный комплекс для точного решения задачи маршрутной оптимизации с ограничениями в виде условий предшествования и функциями стоимости, учитывающими возможную зависимость от списка невыполненных заданий.
3. Реализован эвристический алгоритм для решения "больших" маршрутных задач, учитывающий ограничения в виде условий предшествования и функции стоимости, зависящие от списка заданий.
4. Предложен метод улучшения эвристического алгоритма на основе оптимизирующих мультивставок с использованием параллельных вычислений. В результате решения этой задачи реализован программный комплекс на вычислительном кластере.
5. Построен параллельный алгоритм и реализован программный комплекс для решения задачи распределения заданий между участниками.

Научная новизна:

1. Разработан алгоритмический аппарат для решения задач маршрутизации с ограничениями и усложнёнными функциями стоимости. Впервые реализована схема распараллеливания динамического программирования на вычислительном кластере, при которой вычислительные узлы выполняют "сквозные" вычисления, не использующие обмен данными между узлами при последовательном расчёте слоев функции Беллмана. Данная реализация позволила существенно повысить размерность решаемых задач и сократить время вычислений.
2. Для решения прикладной задачи минимизации дозовой нагрузки работников атомных электростанции при проведении работ по демонтажу выводимого из эксплуатации оборудования или персонала аварийно-спасательных формирований при ликвидации чрезвычайных ситуаций на АЭС, построен и реализован в виде программы для МВС параллельный алгоритм. Данная прикладная задача предполагает учет ряда ограничений в виде условий предшествования и возможную зависимость функций стоимости от списка невыполненных заданий.
3. Для решения задач маршрутизации, имеющих большую размерность, построен параллельный алгоритм, реализующий оптимизирующие

мультивставки в эвристические решения с целью улучшения качества исследуемого процесса.

4. Построен параллельный алгоритм на основе ДП для решения задачи распределения заданий.

Теоретическая и практическая значимость.

Построены новые параллельные алгоритмы решения задач маршрутизации перемещений и распределения заданий между участниками; в основе данных алгоритмов находится аппарат ДП и схемы независимых вычислений в условиях возможного перекрытия потоков данных вычислительной процедуры.

Построенные в работе алгоритмы могут быть использованы при решении таких актуальных инженерных задач, как проблема демонтажа радиационно опасных объектов при авариях на АЭС, подобных Чернобылю и Фукусиме, плановом выводе из эксплуатации отработавшего радиационно опасного оборудования на АЭС и управления инструментом при листовой резке деталей на машинах с ЧПУ (задача, связанная с раскроем). Другие возможные применения связаны с транспортными задачами, авиапожарным патрулированием лесов, задачей о сборе космического "мусора".

Степень достоверности полученных результатов обеспечивается строгими выводами и математическими доказательствами, воспроизводимостью прогнозируемых результатов при вычислительных экспериментах с использованием МВС, сопоставлением известным из литературы результатам для аналогичных моделей в тех случаях, когда упомянутые аналогичные результаты имеются.

Апробация работы. Основные результаты работы докладывались на:

– Григорьев А.М., Иванко Е.Е., Ченцов А.Г. *К вопросу о применении параллельных алгоритмов для решения задач маршрутизации по методу динамического программирования* // Анализ моделирование, развитие экономических систем : V междунар. шк.-симп. АМУР-2011 (Украина, Севастополь, 2011)

– Григорьев, А. М. *Решение минимаксной распределительной задачи методом динамического программирования с применением параллельных вычислений* // Научный сервис в сети Интернет: экзафлопсное будущее (Россия, Новоросийск, 2011)

- Григорьев А.М., Иванко Е.Е., Ченцов А.Г., Ченцов П.А. *Параллельная реализация метода динамического программирования в обобщённой задаче курьера* // Научный сервис в сети Интернет: поиск новых решений (Россия, Новоросийск, 2012)
- Григорьев А.М., Иванко Е.Е. *Об одной реализации метода динамического программирования для задачи коммивояжера осложнённой условиями предшествования* // 47-я молодежная школа-конференция "Современные проблемы математики и ее приложений" (Россия, Екатеринбург, 2016)
- Chentsov A. G., Grigoryev A.M. *Scheme of Independent Calculations in a Precedence Constrained Routing Problem* // Discrete Optimization and Operations Research - (DOOR 2016): 9th International Conference (Russia, Vladivostok, 2016)
- Chentsov A. G., Grigoryev A.M., Chentsov A. A. *An Extremal Problem in Minimizing Staff Exposure to Radiation During Tasks Connected with Dismantlement of Radiation Sources* // VIII Intern. Conf. “OPTIMIZATION AND APPLICATIONS” (OPTIMA-2017) (Montenegro, Petrovac, 2017)
- Grigoryev A.M., Tashlykov O.L. *Solving a routing optimization of works in radiation fields with using a supercomputer* // Physics, Technologies and Innovation (PTI-2018) (Russia, Ekaterinburg, 2018)
- Ченцов А.Г., Григорьев А.М., Ченцов А.А. *Экстремальная маршрутизация с выбором точки старта и ее применение в задаче о демонтаже излучающих элементов* // Анализ, моделирование, управление, развитие социально-экономических систем: сборник научных трудов XII Международной школы-симпозиума АМУР-2018 (Россия, Симферополь-Судак, 2018)
- Grigoryev A.M., Tashlykov O.L. *Route optimization during works in non-stationary radiation fields with obstacles* // Physics, Technologies and Innovation (PTI-2019) (Russia, Ekaterinburg, 2019)

Основные результаты по теме диссертации изложены в 23 печатных и электронных изданиях, 4 из которых изданы в журналах и в трудах конференций, рекомендованных ВАК, 9 – в журналах и в трудах конференций, индексируемых системами WoS и Scopus, 10 – в тезисах докладов.

Личный вклад. Основные результаты диссертации опубликованы в [41–53]. В работах, написанных в соавторстве с научным руководителем, А.Г. Ченцову принадлежит постановка задачи и теоретические конструкции, связанные с процедурами на основе ДП, а А.М. Григорьеву эффективные параллельные алгоритмы, реализованные на суперкомпьютере; кроме того, им проведен обширный вычислительный эксперимент, позволивший выявить целый ряд полезных качественных зависимостей. В статьях, написанных в соавторстве с А.А. Ченцовым, соавтору А.А. Ченцову принадлежат выводы формул, оценивающих радиационное воздействие на различных этапах перемещений исполнителя, а также построение алгоритмов и программ для ПЭВМ, которые использовались в экспериментах при сравнении производительности с алгоритмами для МВС. В работах, написанных в соавторстве с О.Л. Ташлыковым, соавтору О.Л. Ташлыкову принадлежит инженерная постановка и описание физических явлений, связанных с задачей дозиметриста; А.М. Григорьевым была предложена математическая постановка и построен параллельный алгоритм, реализованный на супервычислителе. В работах, написанных в соавторстве с Е.Е. Иванко, соавтору Е.Е. Иванко принадлежит разработка алгоритмов и оценка их трудоемкости, А.М. Григорьеву принадлежит построение параллельного алгоритма, программная реализация и проведение вычислительных экспериментов. В статье «Григорьев А.М., Иванко Е.Е., Ченцов А.Г., Сесекин А.Н., Ташлыков О.Л., Щеклеин С.Е. Решение задач маршрутизации применительно в радиационно опасным объектам с использованием суперкомпьютера "Уран"// Безопасность АЭС и подготовка кадров: тез. докл. 12-й Междунар. конф.- Обнинск, 2011.– Т.2.– С.103–105» соавторам А.Н. Сесекину и С.Е. Щеклейну принадлежит содержательная постановка рассматриваемой инженерной задачи, а диссертанту – построение параллельного алгоритма. В работе «Григорьев А.М., Иванко Е.Е., Князев С.Т., Ченцов А.Г. Динамическое программирование в обобщенной задаче курьера, осложненной внутренними работами // Мехатроника. Автоматизация. Управление.– 2012.– № 7.– С. 14–21.» соавтору Князеву С.Т. принадлежит некоторые элементы содержательной постановки задачи об авиапожарном патрулировании лесов. В статье «Григорьев А.М., Иванко Е.Е., Ченцов А.Г., Ченцов П.А. Параллельная реализация метода динамического программирования в обобщенной задаче курьера // Междунар. суперкомпьютер. конф. "Научный сервис с сети Интернет: поиск новых решений Абрау-Дюрсо,– 2012: труды.

С.315–319.» соавтору П.А. Ченцову принадлежит алгоритм решения обобщеной задачи курьера для ПЭВМ (результаты решения на ПЭВМ использовались в целях сравнения с аналогичными результатами для МВС).

Объем и структура работы. Диссертация состоит из введения, трёх глав, заключения и списка литературы. Полный объём диссертации составляет 114 страниц с 13 рисунками Список литературы содержит 113 наименований.

Введение дает общую характеристику работы, как того требует ГОСТ Р 7.0.11—2011 "Диссертация и автореферат диссертации. Структура и правила оформления".

Глава 1 посвящена вопросам параллельного решения задачи последовательного обхода мегаполисов с условиями предшествования и внутренними работами, связанными с посещением мегаполисов; данная задача является обобщением GTSP-PC и включает, на уровне постановки, функции стоимости, зависящие от списка невыполненных (на текущий момент) заданий. На основе метода ДП построен параллельный алгоритм для нахождения оптимального решения. Кроме того, построен эффективный эвристический (жадный) алгоритм и проведено сравнение последнего с оптимальным в смысле достигаемого результата: выполнен вычислительный эксперимент на МВС решения точным и эвристическим алгоритмами.

Глава 2 посвящена вопросу локального улучшения эвристических решений в задаче последовательного обхода мегаполисов с условиями предшествования и функциями стоимости, допускающими зависимость от списка заданий. Данное улучшение реализуется параллельным алгоритмом, реализуемым в рамках построения специальной оптимизирующей мультивставки (имеется в виду система оптимизирующих вставок в эвристическое решение). Приведен вычислительный эксперимент на супервычислителе, показывающий значительное улучшение исходного эвристического решения.

Глава 3 посвящена исследованию задачи оптимизации маршрута перемещения персонала при проведении работ в нестационарных радиационных полях с учётом обхода возможных препятствий. Описывается метод построения карты радиационного фона на плоскости по заранее измеренным значениям уровня радиации в ряде точек на этой плоскости. Рас-

сматривается параллельная реализация метода ДП в задачах об оптимальном распределении заданий.

Заключение представляет изложение итогов исследования, рекомендаций и перспектив дальнейшей разработки темы.

Глава 1. Реализация схемы независимых вычислений в обобщённой задаче курьера

1.1 Некоторые прикладные задачи с элементами маршрутизации

Задача коммивояжера (Traveling Salesman Problem, TSP) является одной из самых известных и популярных задач дискретной оптимизации. Постановка задачи заключается в поиске маршрута наименьшей стоимости, проходящего через все города $1 \dots n$, каждый город нужно посетить ровно один раз. Стоимость перемещений между городами может задаваться какой-то функцией (в простейшем случае матрицей попарных расстояний). Задача коммивояжера занимает особое место в комбинаторной оптимизации и принадлежит к числу типовых NP-трудных задач; см. [38, гл.3]. Напомним некоторые исследования в области TSP: [1–8;12]. Большое число важных в прикладном отношении вариантов задач типа TSP обуславливает интерес к развитию методов эффективного (и, в частности, параллельного) решения этой задачи.

Задачи маршрутизации возникают во многих инженерных приложениях. В частности, элементы маршрутной оптимизации могут возникать при рассмотрении некоторых прикладных задач, актуальных для атомной энергетики в области минимизации дозовой нагрузки работников при выполнении работ в нестационарных радиационных полях (демонтаж оборудования энергоблока АЭС при выводе из эксплуатации, ликвидация последствий радиационных аварий). Важным фактором при проведении таких комплексов работ является минимизация дозовой нагрузки как на ремонтный персонал АЭС,так и на специалистов аварийно-спасательных формирований, занимающихся устранением последствий аварий на объектах использования атомной энергии. Как показывают исследования, главным дозообразующим фактором на АЭС является ремонтно-профилактическое обслуживание оборудования ядерной энергетической установки. Для ее минимизации предлагается построение оптимального маршрута для рабочей смены работника, позволяющего уменьшить «бесполезную дозу» радиации (доза, полученная при переходе от одного объекта к другому). Так же важным моментом является построение модельных примеров

устранения чрезвычайных ситуаций при аварии энергоблока АЭС (в качестве входных параметров обычно берутся результаты исследований аварии на Чернобыльской АЭС). Такие примеры позволят построить маршруты бригады спасателей и смоделировать действия этой бригады при возникновении ЧС, что впоследствии может пригодиться в реальных условиях. Существенной особенностью данной постановки задачи является зависимость радиационного поля от объектов, не демонтированных на момент прибытия к объекту. Речь идет об утилизации источников радиоактивного излучения, осуществляемой последовательно во времени; в этом случае исполнитель находится под воздействием источников, которые не были демонтированы на момент соответствующего перемещения. Кроме того, могут возникать дополнительные ограничения в виде условий предшествования, обусловленные необходимостью посетить один объект раньше другого. Наконец, в отличие от TSP, в прикладных задачах объектами посещения нередко являются не “города” (как в TSP), а мегаполисы, что отвечает возможной многовариантности самих перемещений. Последнее обстоятельство приводит к “двухуровневым” решениям задачи: выделяется этап нумерации мегаполисов посредством назначения перестановки индексов, то есть собственно маршрута, и этап выбора трассы движения по маршруту, то есть варианта перемещений по занумерованным мегаполисам. Данная логика, приводящая к двухуровневой задаче оптимизации, изложена в [54], а применение в атомной энергетике методов, разработанных в [54], обсуждается в [55; 56]. Существуют и другие приложения [36; 57; 58].

Отметим также, что подобные зависимости могут возникать в задачах, связанных с машиностроением; имеются в виду вопросы управления инструментом при листовой резке деталей на машинах с ЧПУ; см. [59–63]. Введение в этих задачах упомянутых зависимостей может быть связано с учётом технологических ограничений (жесткость листа, тепловые допуски) посредством введения соответствующих штрафов, а также необходимостью реза вложенных контуров раньше внешних.

Задача оптимизации управлением режущим инструментом при листовой резке на станках с ЧПУ в случае использования технологии резки по замкнутому контуру состоит в построении маршрута, который начинается из стартовой точки, затем соединяет все эквидистанты контуров (кривая, отстоящая от контура на заданное расстояние) и заканчивается маршрут в финишной точке.

Для решения данной задачи заранее определяется конечное множество точек на эквидистанте. Если количество таких точек достаточно велико (несколько десятков), то аппроксимация получается удовлетворительной, и существенного ухудшения результата по сравнению с непрерывной задачей не будет. Стоит заметить, что дискретная задача эквивалентна известной обобщенной задаче коммивояжера с дополнительными ограничениями. Первое из этих ограничений состоит в использовании условий предшествования, возникающих из вложенности контуров (сначала нужно вырезать внутренние). Второе ограничение касается качества реза, и состоит в обеспечении достаточного количества металла возле финишного участка реза контура. В то же время от него в значительной степени зависит и соблюдение геометрических размеров вырезаемых заготовок. Целью оптимизации является не только сокращение времени работы машины, но и повышение качества заготовок. Таким образом, рассматриваемая задача сводится к обобщенной задаче коммивояжера с условиями предшествования, функциями стоимости, зависящими от списка уже вырезанных контуров и учётом направления реза. Имеются и другие приложения развиваемой теории: транспортные задачи, задачи авиапожарного патрулирования.

В настоящей работе мы сосредоточимся на задаче, связанной со снижением дозовой нагрузки персонала АЭС и специалистов аварийно-спасательных формирований, хотя многие конструкции, используемые ниже, сохраняют свою применимость и в других прикладных задачах (в том числе в задаче управления инструментом при листовой резке на машинах с ЧПУ).

1.2 Математическая постановка задачи

Общие сведения и обозначения.

Отметим сначала некоторые общие понятия и обозначения, используемые ниже. Через $\stackrel{\triangle}{=}$ обозначаем равенство по определению. Семейством называем множество, элементами которого являются множества. Через $\mathbb{R}$ обозначаем вещественную прямую, $\mathbb{R}_+ \stackrel{\triangle}{=} \{\xi \in \mathbb{R} \,|\, 0 \leqslant \xi\}$, $\mathbb{N} \stackrel{\triangle}{=} \{1; 2; \ldots\}$ и $\mathbb{N}_0 \stackrel{\triangle}{=} \mathbb{N} \cup \{0\} = \{0; 1; 2; \ldots\}$, $\mathbb{N} \subset \mathbb{N}_0 \subset \mathbb{R}$. Для $p \in \mathbb{N}_0$ и $q \in \mathbb{N}_0$ полагаем $\overline{p,q} \stackrel{\triangle}{=} \{t \in \mathbb{N}_0 \,|\, (p \leqslant t) \,\&\, (t \leqslant q)\}$. Для каждой упорядоченной пары УП

$z = (a,b)$ произвольных объектов a и b через $\mathrm{pr}_1(z)$ и $\mathrm{pr}_2(z)$ обозначаем соответственно первый и второй элементы z: $\mathrm{pr}_1(z) = a,\ \mathrm{pr}_2(z) = b$. Если x — объект, то $\{x\}$ есть одноэлементное множество, содержащее x. Для произвольных объектов a,b и c, как обычно [64, с. 17], полагаем $(a,b,c) \stackrel{\triangle}{=} ((a,b),c)$, получая триплет в виде УП специального вида. Для трех произвольных множеств A,B и C, следуя [64, с. 17], имеем $A \times B \times C \stackrel{\triangle}{=} (A \times B) \times C$, а потому при $\mu \in A \times B$ и $\nu \in C$ реализуется $(\mu, \nu) \in A \times B \times C$.

Через $\mathcal{R}_+[S]$ обозначаем множество всех функций, действующих из непустого множества S в $\mathbb{R}_+$, то есть множество всех неотрицательных вещественнозначных (в/з) функций на S.

Если H — множество, то через $\mathcal{P}(H)$ обозначаем семейство всех подмножеств (п/м) H, то есть булеан H; $\mathcal{P}'(H) \stackrel{\triangle}{=} \mathcal{P}(H) \setminus \{\varnothing\}$, а $\mathrm{Fin}(H)$ есть семейство всех конечных множеств из $\mathcal{P}'(H)$. Если H — конечное множество, то $\mathrm{Fin}(H) = \mathcal{P}'(H)$. Непустому конечному множеству K сопоставляется его мощность $|K| \in \mathbb{N}$ (количество элементов) и непустое множество $(\mathrm{bi})[K]$ всех биекций [65, с. 87] множества $\overline{1,|K|} = \{j \in \mathbb{N} \,|\, j \leqslant |K|\}$ на K; кроме того, $|\varnothing| \stackrel{\triangle}{=} 0$. Перестановкой непустого множества T называется [65, с. 87] всякая биекция T на себя; если α — перестановка T, то определена перестановка α^{-1} множества T, обратная к α: $\alpha(\alpha^{-1}(t)) = \alpha^{-1}(\alpha(t)) = t \quad \forall t \in T$. Используемые ниже перестановки индексов будут рассматриваться в качестве маршрутов посещения целевых множеств – мегаполисов.

Специальные понятия.

Фиксируем непустое множество X, точку $x^\circ \in X$, число $N \in \mathbb{N}, N \geqslant 2$, (непустые конечные) множества $M_1 \in \mathrm{Fin}(X), \ldots, M_N \in \mathrm{Fin}(X)$, а также отношения

$$\mathbb{M}_1 \in \mathcal{P}'(M_1 \times M_1), \ldots, \mathbb{M}_N \in \mathcal{P}'(M_N \times M_N).$$

Полагаем в дальнейшем, что

$$(x^\circ \notin M_j \quad \forall j \in \overline{1,N}) \,\&\, (M_p \cap M_q = \varnothing \quad \forall p \in \overline{1,N} \quad \forall q \in \overline{1,N} \setminus \{p\}). \qquad (1.2.1)$$

Рассматриваем $M_1, \ldots, M_N$ в качестве мегаполисов, подлежащих посещению. При $j \in \overline{1,N}$ УП из $\mathbb{M}_j$ определяют фактически возможные варианты выполнения работ, связанных с посещением M_j: первый элемент каждой такой УП есть пункт прибытия, а второй элемент – пункт отправления. Конкретная интер-

претация множеств $\mathbb{M}_j$ может быть разной, но мы будем рассматривать вышеупомянутый вариант инженерной задачи, поэтому точки множеств $\mathbb{M}_j$ считаем как входы-выходы в помещение с номером $j \in \overline{1,N}$. Работник использует эти входы-выходы для того, чтобы войти в соответствующее помещение, выполнить в нем необходимые работы, и выйти из помещения. Мы рассматриваем далее процессы следующего вида [66, (3.3)]:

$$\begin{aligned} x^\circ \to (\mathrm{pr}_1(z_1) \in M_{\alpha(1)} \rightsquigarrow \mathrm{pr}_2(z_1) \in M_{\alpha(1)} \to \ldots \\ \to (\mathrm{pr}_1(z_N) \in M_{\alpha(N)} \rightsquigarrow \mathrm{pr}_z(z_N) \in M_{\alpha(N)}, \end{aligned} \tag{1.2.2}$$

где $z_1 \in \mathbb{M}_{\alpha(1)}, \ldots, z_N \in \mathbb{M}_{\alpha(N)}$ и α — перестановка индексов из $\overline{1,N}$, именуемая далее маршрутом. Полагаем, что в (1.2.2) прямые стрелки отвечают внешним перемещениям (перемещениям между мегаполисами и из x° в мегаполисы), а волнистые — перемещениям, связанным с выполнением (внутренних) работ. Объектами нашего выбора являются $\alpha, z_1, \ldots, z_N$. Следуя [66, (3.4)], полагаем

$$\mathbf{M}_j \stackrel{\triangle}{=} \{\mathrm{pr}_2(z) : \ z \in \mathbb{M}_j\} \in \mathrm{Fin}(M_j) \ \ \forall j \in \overline{1,N}. \tag{1.2.3}$$

С учётом этого мы получаем непустые конечные множества

$$\mathbb{X} \stackrel{\triangle}{=} \{x^\circ\} \cup \left(\bigcup_{i=1}^{N} M_i \right) \in \mathrm{Fin}(X), \mathbf{X} \stackrel{\triangle}{=} \{x^\circ\} \cup \left(\bigcup_{i=1}^{N} \mathbf{M}_i \right) \in \mathrm{Fin}(\mathbb{X}).$$

Множество $\mathbb{X}$ играет объективно роль фазового пространства процессов (1.2.2), а множество $\mathbf{X}$ будет использовано для более экономного варианта построения слоев функции Беллмана.

В связи с (1.2.2) заметим, что внешние перемещения и работы, выполняемые при посещении мегаполисов и именуемые далее внутренними, оцениваются посредством заданных функций, а результаты этих оцениваний агрегируются аддитивно, что естественно для многих приложений: в частности, такой способ агрегирования соответствует упомянутой задаче о снижении дозовой нагрузки.

В [48;67–69] рассматривались постановки, в которой x° может выбираться (и при этом оптимизироваться)из заданного непустого множества $X^\circ, X^\circ \subset X$. В диссертации данное обобщение не рассматривается.

Полагаем, что выбор α может быть стеснен дополнительными ограничениями, так называемыми условиями предшествования. Элементами $\mathbf{K}$ являются упорядоченные пары; будем условно именовать первую компоненту упорядоченной пары отправителем, а вторую – получателем; сами же элементы $\mathbf{K}$ называем адресными парами. Допустимость $\alpha \in \mathbb{P}$, где $\mathbb{P} \stackrel{\triangle}{=} (\mathrm{bi})[\overline{1,N}]$ сводится к требованию: при $z = (i,j) \in \mathbf{K}$ посещение M_i должно предшествовать посещению M_j. Полагаем в дальнейшем, что:

$$\forall \mathbf{K}_0 \in \mathcal{P}'(\mathbf{K}) \exists z_0 \in \mathbf{K}_0 : \ \mathrm{pr}_1(z_0) \neq \mathrm{pr}_2(z) \ \ \forall z \in \mathbf{K}_0. \tag{1.2.4}$$

Данное неограничительное условие (см. [54]) как правило выполняется в прикладных задачах. При этом множество $\mathbf{A}$ всех допустимых (по предшествованию) маршрутов из $\mathbb{P}$ имеет вид [66, (3.12)]:

$$\mathbf{A} \stackrel{\triangle}{=} \{\alpha \in \mathbb{P} | \alpha^{-1}(\mathrm{pr}_1(z)) < \alpha^{-1}(\mathrm{pr}_2(z)) \ \ \forall z \in \mathbf{K}\} \in \mathcal{P}'(\mathbb{P}). \tag{1.2.5}$$

Итак, нас удовлетворяют только такие маршруты, в которых для каждой адресной пары отправитель будет посещаться раньше получателя.

В связи с (1.2.4) отметим, что $\mathrm{pr}_1(z) \neq \mathrm{pr}_2(z) \ \ \forall z \in \mathbf{K}$. Как видно из (1.2.2) выбор маршрута $\alpha \in \mathbf{A}$ еще не определяет конкретную реализацию процесса; следует еще рассматривать трассу $z_1, \ldots, z_N$. Возможности выбор трассы (посещения мегаполисов) зависят от указанного маршрута; совокупное решение определяется далее в виде УП маршрут–трасса, где маршрут должен быть элементом $\mathbf{A}$.

Итак, если $\alpha \in \mathbb{P}$, то через $\mathcal{Z}_\alpha$ обозначаем множество всех кортежей $(z_i)_{i \in \overline{1,N}} : \overline{0,N} \to \mathbb{X} \times \mathbb{X}$, для каждого из которых

$$(z_0 = (x^0, x^0)) \,\&\, (\, z_t \in \mathbb{M}_{\alpha(t)} \ \ \forall t \in \overline{1,N});$$

$\mathcal{Z}_\alpha$ — непустое конечное множество. В качестве допустимых решений (ДР) будем использовать УП $(\alpha, (z_i)_{i \in \overline{0,N}})$, где $\alpha \in \mathbf{A}$ и $(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha$ Пусть $\mathfrak{N} \stackrel{\triangle}{=} \mathcal{P}'(\overline{1,N})$. Множества-элементы $\mathfrak{N}$ – будем называть списками (заданий). В дальнейших построениях элементы множества $K \in \mathfrak{N}$ будут играть роль заданий, не выполненных на текущий момент времени.

Полагаем далее заданными функции стоимости

$$\begin{gathered}\mathbf{c} \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}], c_1 \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}], \ldots, \\ c_N \in \mathcal{R}_+[\mathbb{X} \times \mathbb{X} \times \mathfrak{N}],\ f \in \mathcal{R}_+[\mathbb{X}].\end{gathered} \tag{1.2.6}$$

Из (1.2.6) следует, что используемые функции стоимости допускают зависимость от списка заданий. Отметим в этой связи работу [70], где построено эвристическое решение TSP с объёмной матрицей затрат (коробкой). В терминах этих функций определяем аддитивный критерий, полагая при $\alpha \in \mathbf{A}$ и $(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha$

$$\begin{gathered}\mathfrak{G}_\alpha[(z_i)_{i \in \overline{0,N}}] \stackrel{\triangle}{=} \sum_{t=1}^{N}[\mathbf{c}(\mathrm{pr}_2(z_{t-1}), \mathrm{pr}_1(z_t), \{\alpha(j) : j \in \overline{t,N}\}) + \\ + c_{\alpha(t)}(z_t, \{\alpha(j) : j \in \overline{t,N}\})] + f(\mathrm{pr}_2(z_N)).\end{gathered} \tag{1.2.7}$$

Итак, функция $\mathbf{c}$ оценивает внешние перемещения, функции $c_1, \ldots, c_N$ - внутренние работы, а f - терминальное состояние процесса. В качестве основной рассматриваем задачу минимизации аддитивного критерия:

$$\mathfrak{G}_\alpha[(z_i)_{i \in \overline{0,N}}] \to \min,\ \alpha \in \mathbf{A}, (z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha. \tag{1.2.8}$$

Поскольку $\mathbf{A}$ – непустое конечное множество, то задаче (1.2.8) сопоставляется значение (экстремум)

$$V \stackrel{\triangle}{=} \min_{\alpha \in \mathbf{A}} \min_{(z_i)_{i \in \overline{0,N}} \in \mathcal{Z}_\alpha} \mathfrak{G}_\alpha[(z_i)_{i \in \overline{0,N}}] \in \mathbb{R}_+ \tag{1.2.9}$$

Наша цель состоит в нахождении (глобального) экстремума V (1.2.9) и какого–либо оптимального ДР $(\alpha^0, (z_i^0)_{i \in \overline{0,N}})$, где $\alpha^0 \in \mathbf{A}$, $(z_i^0)_{i \in \overline{0,N}} \in \mathcal{Z}_{\alpha^0}$ и при этом $\mathfrak{G}_{\alpha^0}[(z_i^0)_{i \in \overline{0,N}}] = V$.

В случае, когда задача (1.2.8) имеет достаточно большую размерность (прежде всего, при большом количестве мегаполисов) нахождение V и ДР, доставляющих V (1.2.9), крайне сложно реализуемо с вычислительной точки зрения, хотя общие принципы здесь известны (см. в частности, процедуры на основе широко понимаемого ДП [6;71;72]). В этой связи возникает необходимость в использовании эвристик с локальными улучшениями в виде вставок и мультивставок на основе ДП (см. [51;73–75]). Предполагается, что в каждой вставке

используется процедура оптимизации на основе широко понимаемого динамического программирования. Предлагаемая конструкция допускает вычисления каждой вставки в отдельности, независимо друг от друга, в этой связи можно отметить целесообразным использование параллельной процедуры реализации мультвставок. В такой реализации, каждая вставка обрабатывается отдельным вычислительным узлом, что позволяет реализовать параллельную процедуру с распределённой памятью.

1.3 Вспомогательные конструкции для процедуры решения по динамическому программированию

В дальнейшем изложении используется вариант метода ДП, восходящий к работе [7]. Данный вариант допускает естественные аналоги с конструкциями, применяемыми в теории управления и теории дифференциальных игр. Речь идет об использовании попятной процедуры при построении функции Беллмана, либо ее слоя. В задачах дискретной оптимизации получила широкое распространение известная "прямая"схема Хелда и Карпа [8]. Мы, однако, придерживаемся конструкций, развивающих [7], имея в виду то, что при их использовании удаётся без какого-либо существенного усложнения процедуры реализовать решение задачи оптимизации начального состояния процесса. Это обстоятельство будет использоваться в дальнейших построениях.

Для решения задачи (1.2.8) используется вариант ДП [66; 71; 72], излагаемый здесь в краткой форме. Мы используем принятую в [66; 71; 72] конструкцию на основе слоев функции Беллмана, для построения которых сейчас излагается алгоритм на функциональном уровне. Общие вопросы, связанные с ДП и касающиеся вывода уравнения Беллмана, изложены, в частности, в [40; 54; 66; 71; 76; 77]. В настоящем исследовании мы ограничиваемся изложением алгоритмических конструкций. Для использования ДП в задаче с возможными ограничениями в виде условий предшествования, определяемых множеством $\mathbf{K}$, мы прежде всего осуществляем редукцию соответствующих ограничений: допустимость по предшествованию заменяется допустимостью по вычеркиванию.

Рассмотрим в этой связи построение слоев пространства позиций, используя оператор вычеркивания (заданий из списка) [54, часть 2]

$$\mathbf{I} : \mathfrak{N} \to \mathfrak{N}, \tag{1.3.1}$$

а именно: полагаем при $K \in \mathfrak{N}$, что $\Xi(K) \stackrel{\triangle}{=} \{z \in \mathbf{K} \,|\, (\mathrm{pr}_1(z) \in K) \,\&\, (\mathrm{pr}_2(z) \in K)\}$ и

$$\mathbf{I}(K) \stackrel{\triangle}{=} K \setminus \{\mathrm{pr}_2(z) : \ z \in \Xi(K)\}. \tag{1.3.2}$$

Оператор (1.3.2) позволяет соблюдать условия предшествования в процессе рекуррентного построения слоев функции Беллмана. Отметим, что $\mathbf{I}(\{t\}) = \{t\}$ при $t \in \overline{1,N}$. Это свойство вытекает из (1.2.4) (см. также [66, замечание 3.2]). Мы используем оператор (1.3.1), (1.3.2) для построения слоев пространства позиций. В этой связи введем (непустое) множество

$$\mathbf{G} \stackrel{\triangle}{=} \{K \in \mathfrak{N} \,|\, \forall z \in \mathbf{K} \ (\mathrm{pr}_1(z) \in K) \Rightarrow (\mathrm{pr}_2(z) \in K)\}, \tag{1.3.3}$$

элементы которого называем существенными списками заданий; ясно, что $\overline{1,N} \in \mathbf{G}$. Упомянутые списки ранжируем по мощности, полагая

$$\mathbf{G}_s \stackrel{\triangle}{=} \{K \in \mathbf{G} \,|\, s = |K|\} \ \forall s \in \overline{1,N}. \tag{1.3.4}$$

Тогда семейство $\{\mathbf{G}_j : \ j \in \overline{1,N}\}$ (см. (1.3.4)) определяет разбиение исходного множества (1.3.3). При этом

$$\mathbf{G}_1 \stackrel{\triangle}{=} \{\{t\} : \ t \in \overline{1,N} \setminus \mathbf{K}_1\},$$

где $\mathbf{K}_1 \stackrel{\triangle}{=} \{\mathrm{pr}_1(z) : z \in \mathbf{K}\}$. Кроме того [66; 71; 72],

$$\mathbf{G}_{s-1} = \{K \setminus \{j\} : \ K \in \mathbf{G}_s, \, j \in \mathbf{I}(K)\} \ \forall s \in \overline{2,N}. \tag{1.3.5}$$

Следовательно, у нас реализуется рекуррентная процедура

$$\mathbf{G}_N \to \mathbf{G}_{N-1} \to \ldots \to \mathbf{G}_1. \tag{1.3.6}$$

На основе этой процедуры конструируются слои пространства позиций, обозначаемые через $D_0, D_1, \ldots, D_N$. При этом

$$D_0 = \{(x, \varnothing) : \ x \in \mathfrak{M}\}, \tag{1.3.7}$$

где $\mathfrak{M}$ есть def объединение всех множеств $\mathbf{M}_i$, $i \in \overline{1,N} \setminus \mathbf{K}_1$. Кроме того, $D_N \stackrel{\triangle}{=} \{(x^0, \overline{1,N})\}$ (синглетон, содержащий УП $(x^0,\overline{1,N})$). Если $s \in \overline{1,N-1}$ и $K \in \mathbf{G}_s$, то последовательно полагаем, что [71]

$$J_s(K) \stackrel{\triangle}{=} \{t \in \overline{1,N} \setminus K \,|\, \{t\} \cup K \in \mathbf{G}_{s+1}\}, \mathcal{M}_s[K] \stackrel{\triangle}{=} \bigcup_{j \in J_s(K)} \mathbf{M}_j, \ \mathbb{D}_s[K] \stackrel{\triangle}{=}$$
$$\stackrel{\triangle}{=} \{(x,K) : x \in \mathcal{M}_s[K]\} \in \mathcal{P}'(\mathbf{X} \times \mathbf{G}_s).$$

Тогда при $s \in \overline{1,N-1}$ слой D_s определяем правилом

$$D_s \stackrel{\triangle}{=} \bigcup_{K \in \mathbf{G}_s} \mathbb{D}_s[K]. \tag{1.3.8}$$

Легко видеть, что $D_0 \neq \varnothing, D_1 \neq \varnothing, \ldots, D_N \neq \varnothing$. Если $s \in \overline{1,N}, (x,K) \in D_s, j \in \mathbf{I}(K)$ и $z \in \mathbb{M}_j$, то

$$(\mathrm{pr}_2(z), K \setminus \{j\}) \in D_{s-1}. \tag{1.3.9}$$

Посредством (1.3.7)–(1.3.9) определена конструкция слоев, логически связанная с процедурой (1.3.6).

1.4 Рекуррентная процедура построения слоев функции Беллмана

Последовательно определяем систему функций $v_0 \in \mathcal{R}_+[D_0], v_1 \in \mathcal{R}_+[D_1], \ldots, v_N \in \mathcal{R}_+[D_N]$, являющихся [40; 71; 72; 77] слоями функции Беллмана основной задачи маршрутизации (см. (1.2.8)). Полагаем сначала, что $v_0 \in \mathcal{R}_+[D_0]$ определяется (см. (1.3.7)) условием

$$v_0(x, \varnothing) \stackrel{\triangle}{=} f(x) \ \ \forall x \in \mathfrak{M}. \tag{1.4.1}$$

Дальнейшее построение определяется рекуррентной схемой: если $s \in \overline{1,N}$ и $v_{s-1} \in \mathcal{R}_+[D_{s-1}]$ уже построена, то полагаем с учетом (1.3.9), что $v_s \in \mathcal{R}_+[D_s]$ определяется правилом [66]

$$v_s(x,K) \stackrel{\triangle}{=} \min_{j \in \mathbf{I}(K)} \min_{z \in \mathbb{M}_j} [\mathbf{c}(x,\mathrm{pr}_1(z),K) + c_j(z,K) + v_{s-1}(\mathrm{pr}_2(z),K \setminus \{j\})| \; \forall (x,K) \in D_s. \tag{1.4.2}$$

После этого при $s < N$ может осуществляться замена v_{s-1} функцией v_s в памяти вычислителя, т. е. может осуществляться перезапись текущего слоя. Если же $s = N$, то осуществляем остановку процедуры, получая $v_N(x^0, \overline{1,N})$, что отвечает представлению D_N.

Из общих построений [66; 71; 72], использующих уравнение Беллмана, вытекает равенство

$$V = v_N(x^0, \overline{1,N}), \tag{1.4.3}$$

определяющее глобальный экстремум (см. (1.4.3)). Процедуру

$$v_0 \to v_1 \to \ldots \to v_N, \to V, \tag{1.4.4}$$

полностью определяемую посредством (1.4.1), (1.4.2) можно (в силу (1.4.2)) рассматривать как алгоритм определения V, при котором в памяти вычислителя находится только один слой функции Беллмана (подробнее см. в [78]).

Итак, для определения V может использоваться следующий алгоритм (допускающий аналогию с [32]):

1. определяем функцию v_0, используя (1.4.1)
2. организуем рекуррентную процедуру на основе (1.4.2). При этом в памяти вычислителя находится один слой функции Беллмана (см. [78], где рассматривался вариант задачи с 34 мегаполисами для полного решения, и 37 мегаполисами для построения V)
3. определяем V, применяя формулу $V = v_N(x^0, \overline{1,N}) = \min\limits_{j \in \mathbf{I}(\overline{1,N})} \min\limits_{z \in \mathbb{M}_j} [\mathbf{c}(x^0, \mathrm{pr}_1(z), \overline{1,N}) + c_j(z,\overline{1,N}) + v_{N-1}(\mathrm{pr}_2(z),\overline{1,N} \setminus \{j\})]$. Значение экстремума V без определения оптимального маршрута может использоваться для тестирования эвристик.

1.5 Построение оптимального маршрута

Рассмотрим вариант, являющийся частным случаем построений [71, § 7]. Здесь требуется сохранение в памяти всех функций $v_0, v_1, \ldots, v_N$. Полагаем $\mathbf{z}^{(0)} \stackrel{\triangle}{=} (x^0, x^0), \mathbf{z}^{(0)} \in \mathbb{X} \times \mathbf{X}$. Мы конструируем решение в виде некоторой УП (α, z), где $\alpha \in \mathbf{A}$ и $\mathbf{z}^{(t)}{}_{t \in \overline{0,N}} : \overline{0,N} \to \mathbb{X} \times \mathbf{X}$. При этом $\mathbf{z}^{(0)}$ играет роль стартового состояния $z(0)$. Поскольку $(x^0, \overline{1,N}) \in D_N$, то в силу (1.3.9) при $j \in \mathbf{I}(\overline{1,N})$ и $z \in \mathbb{M}_j$ имеем $(\mathrm{pr}_2(z), \overline{1,N} \setminus \{j\}) \in D_{N-1}$, а потому определено значение $v_{N-1}(\mathrm{pr}_2(z), \overline{1,N} \setminus \{j\}) \in \mathbb{R}_+$. При этом согласно (1.4.2) и (1.4.3)

$$V = \min_{j \in \mathbf{I}(\overline{1,N})} \min_{z \in \mathbb{M}_j} [\mathbf{c}(x^0, \mathrm{pr}_1(z), \overline{1,N}) + c_j(z, \overline{1,N}) + v_{N-1}(\mathrm{pr}_2(z), \overline{1,N} \setminus \{j\})] \quad (1.5.1)$$

С учетом (1.5.1) находим (в результате решения локальной экстремальной задачи, связанной с (1.5.1)) $\mathbf{j}_1 \in \mathbf{I}(\overline{1,N})$ и $\mathbf{z}^{(1)} \in \mathbb{M}_{\mathbf{j}_1}$, для которых

$$V = \mathbf{c}(x^0, \mathrm{pr}_1(\mathbf{z}^{(1)}), \overline{1,N}) + c_{\mathbf{j}_1}(\mathbf{z}^{(1)}, \overline{1,N}) + v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \overline{1,N} \setminus \{\mathbf{j}_1\}), \quad (1.5.2)$$

где согласно (1.3.9) $(\mathrm{pr}_2(\mathbf{z}^{(1)}), \overline{1,N} \setminus \{\mathbf{j}_1\}) \in D_{N-1}$. Тогда в силу (1.4.2)

$$\begin{aligned} v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \overline{1,N} \setminus \{\mathbf{j}_1\}) = \min_{j \in \mathbf{I}(\overline{1,N} \setminus \{\mathbf{j}_1\})} \min_{z \in \mathbb{M}_j} [\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \mathrm{pr}_1(z), \overline{1,N} \setminus \{\mathbf{j}_1\}) + \\ + c_j(z, \overline{1,N} \setminus \{\mathbf{j}_1\}) + v_{N-2}(\mathrm{pr}_2(z), \overline{1,N} \setminus \{\mathbf{j}_1; j\})]. \end{aligned} \quad (1.5.3)$$

С учетом (1.5.3) находим $\mathbf{j}_2 \in \mathbf{I}(\overline{1,N} \setminus \{\mathbf{j}_1\})$ и $\mathbf{z}^{(2)} \in \mathbb{M}_{\mathbf{j}_2}$ со свойством

$$\begin{aligned} v_{N-1}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \overline{1,N} \setminus \{\mathbf{j}_1\}) = \mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \mathrm{pr}_1(\mathbf{z}^{(2)})), \overline{1,N} \setminus \{\mathbf{j}_1\}) + \\ + c_{\mathbf{j}_2}(\mathbf{z}^{(2)}, \overline{1,N} \setminus \{\mathbf{j}_1\}) + v_{N-2}(\mathrm{pr}_2(\mathbf{z}^{(2)}), \overline{1,N} \setminus \{\mathbf{j}_1; \mathbf{j}_2\}), \end{aligned} \quad (1.5.4)$$

где согласно (1.3.9) $(\mathrm{pr}_2(\mathbf{z}^{(2)}), \overline{1,N} \setminus \{\mathbf{j}_1; \mathbf{j}_2\}) \in D_{N-2}$. Заметим, что в силу (1.5.2) и (1.5.4)

$$\begin{aligned} V = \mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(0)}), \mathrm{pr}_1(\mathbf{z}^{(1)}), \overline{1,N}) + \mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}), \mathrm{pr}_1(\mathbf{z}^{(2)}), \overline{1,N} \setminus \{\mathbf{j}_1\}) + \\ + c_{\mathbf{j}_1}(\mathbf{z}^{(1)}, \overline{1,N}) + c_{\mathbf{j}_2}(\mathbf{z}^{(2)}, \overline{1,N} \setminus \{\mathbf{j}_1\}) + v_{N-2}(\mathrm{pr}_2(\mathbf{z}^{(2)}), \overline{1,N} \setminus \{\mathbf{j}_1; \mathbf{j}_2\}) \end{aligned} \quad (1.5.5)$$

(при $N = 2$ имеем в силу (1.4.1) и (1.5.5), что кортежи

$$(\mathbf{j}_k)_{k\in\overline{1,2}} : \overline{1,2} \to \overline{1,N}, (\mathbf{z}^{(k)})_{k\in\overline{0,2}} : \overline{0,2} \to \mathbb{X} \times \mathbf{X}$$

образуют оптимальное ДР). При $N > 2$ процедуру на основе соотношений, подобных (1.5.2), (1.5.4), следует продолжать вплоть до исчерпывания индексного множества $\overline{1,N}$. В результате будут получены кортежи $\mathfrak{Z} \stackrel{\triangle}{=} (\mathbf{j}_k)_{k\in\overline{1,N}}, (\mathbf{z}^{(k)})_{k\in\overline{0,N}}$,

$$\mathfrak{Z} : \overline{1,N} \to \overline{1,N}, (\mathbf{z}^{(k)})_{k\in\overline{0,N}} : \overline{0,N} \to \mathbb{X} \times \mathbf{X},$$

для которых $\mathfrak{Z} \in \mathbf{A}, (\mathbf{z}^{(k)})_{k\in\overline{0,N}} \in \mathcal{Z}_{\mathfrak{Z}}$ и при этом

$$\mathfrak{G}_{\mathfrak{Z}}[(\mathbf{z}^{(k)})_{k\in\overline{0,N}}] = V. \tag{1.5.6}$$

Следовательно (см. (1.5.6)), в виде $(\mathfrak{Z}, (\mathbf{z}^{(k)})_{k\in\overline{0,N}})$ имеем оптимальное ДР.

1.6 Схема независимых вычислений

В последующем изложении используется вариант общей конструкции [40; 71; 79], связанной с организацией независимых вычислений. В связи с этим напомним (1.5.1), где приведена "оконечная" формула для нахождения V в терминах v_{N-1}. С учетом этого для дальнейших построений существенно определение v_{N-1}, что и реализуется в излагаемой ниже процедуре (вычисление V (1.5.1) при известной функции v_{N-1} затруднений не вызывает). Функция v_{N-1} определена на множестве D_{N-1}, которое, в свою очередь, определяется в терминах $\mathbf{G}_{N-1}$ (см. (1.3.8)). Полагая в дальнейшем $N \geqslant 3$, имеем из (1.3.5) равенство

$$\mathbf{G}_{N-1} = \{\overline{1,N} \setminus \{j\} : j \in \mathbf{I}(\overline{1,N})\} \tag{1.6.1}$$

(учитываем, что $\mathbf{G}_N = \{\overline{1,N}\}$, что позволяет определить $\mathbf{G}_{N-1}$).

Мы предполагаем, что построение всех функций-слоев будет осуществляться $\mathbf{n}$ узлами, где (здесь и ниже) $\mathbf{n} \stackrel{\triangle}{=} |\mathbf{G}_{N-1}| = |\mathbf{I}(\overline{1,N})| \in \mathbb{N}$. С этой целью множества — слои $D_0, D_1, \ldots, D_N$ — будут рассматриваться каждое в виде объединения $\mathbf{n}$ п/м; каждое из этих п/м будет выделяться соответствующему узлу

для построения фрагментов функций-слоев $v_1, \ldots, v_N$. Для построения упомянутых п/м множеств-слоев будут использоваться вспомогательные дискретные динамические системы (ДДС), траектории которых будут определены посредством системы включений, с дискретным временем.

При данном построении каждому узлу выделяется его система (частных) слоев пространства позиций. Данные (частные) слои могут пересекаться. После этого для каждого узла создается своя система (частных) слоев функции Беллмана. В то же время возникает совокупная «горизонтальная» структура, которую составляют упомянутые функции-слои при фиксации мощности множеств, отвечающих спискам заданий данного уровня. Упомянутый уровень как раз и задается индексом, определяющим используемые списки заданной мощности.

Построения на основе ДДС. Итак, пусть $K \in \mathbf{G}_{N-1}$ и $\mathbb{T}[K]$ (пучок траекторий) есть def множество всех кортежей

$$(K_t)_{t\in\overline{0,N-2}} : \overline{0,N-2} \to \mathbf{G} \tag{1.6.2}$$

таких, что $(K_0 \triangleq K)$ & $(\forall \tau \in \overline{1,N-2} \;\; \exists s \in \mathbf{I}(K_{\tau-1}) : K_\tau = K_{\tau-1} \setminus \{s\})$. Множество $\mathbf{G}$ играет роль фазового пространства. Кортежи (1.6.2) — траектории системы — определяются при заданном K не единственным образом, а потому каждому "моменту времени" $t_* \in \overline{0,N-2}$ соответствует область достижимости (ОД) $\widetilde{\mathbb{T}}[K;t_*]$, определяемая в виде множества всех списков K_{t_*} (см. (1.6.2)) при условии, что $(K_\tau)_{\tau\in\overline{0,N-2}}$ пробегает множество $\mathbb{T}[K]$:

$$\widetilde{\mathbb{T}}[K;t] \triangleq \{K_t : (K_i)_{i\in\overline{0,N-2}} \in \mathbb{T}[K]\},$$

при этом [40; 79] $\widetilde{\mathbb{T}}[K;t_*] \in \mathcal{P}'(\mathbf{G}_{N-(t_*+1)})$. Как легко видеть, для всякого $K \in \mathbf{G}_{N-1}$

$$(\widetilde{\mathbb{T}}[K;0] = \{K\}) \;\&\; (\widetilde{\mathbb{T}}[K;N-2] \subset \mathbf{G}_1).$$

В терминах, упомянутых ОД, определяются семейства существенных списков заданной мощности, а именно

$$\mathbf{G}_{N-(t+1)} = \bigcup_{K\in\mathbf{G}_{N-1}} \widetilde{\mathbb{T}}[K;t] \;\; \forall t \in \overline{0,N-2} \;\; \forall K \in \mathbf{G}_{N-1}.$$

Отметим, наконец, что вышеупомянутые ОД можно [79, предложение 16], при фиксированном начальном условии, строить рекуррентно: если $K \in \mathbf{G}_{N-1}$ и $t \in \overline{0,N-3}$, то

$$\widetilde{\mathbb{T}}[K;t+1] = \{P \setminus \{h\} : P \in \widetilde{\mathbb{T}}[K;t],\, h \in \mathbf{I}(P)\}. \tag{1.6.3}$$

В последующих построениях используются только ОД; траектории (1.6.2), их определяющие, далее не используются. Итак, на основе (1.6.3) определяется конечная пошаговая процедура построения ОД.

$$\widetilde{\mathbb{T}}[K;0] \to \widetilde{\mathbb{T}}[K;1] \to \ldots \to \widetilde{\mathbb{T}}[K;N-2]$$

Элементы параллельной структуры алгоритма. С помощью ОД конструируются "индивидуальные" (для каждого вычислительного узла свои) слои пространства позиций: при $K \in \mathbf{G}_{N-1}$ и $s \in \overline{1,N-1}$ полагаем с учетом (1.3.8)

$$\mathcal{D}_s[K] \triangleq \bigcup_{P \in \widetilde{\mathbb{T}}[K;N-(s+1)]} \mathbb{D}_s[P] \in \mathcal{P}'(D_s). \tag{1.6.4}$$

После построения слоев (1.6.4) информация об ОД удаляется из памяти каждого вычислительного узла кластера (это позволяет экономить ресурсы памяти вычислителя); таким образом, роль ОД в построении общего решения сводится к процедуре (1.6.4). Данные слои (1.6.4) обладают свойством, подобным в идейном отношении (1.3.9):

$$\begin{gathered}(\mathrm{pr}_2(z), Q \setminus \{s\}) \in \mathcal{D}_l[K] \quad \forall K \in \mathbf{G}_{N-1} \quad \forall l \in \overline{1,N-2} \\ \forall (x,Q) \in \mathcal{D}_{l+1}[K] \quad \forall s \in \mathbf{I}(Q) \quad \forall z \in \mathbb{M}_s.\end{gathered} \tag{1.6.5}$$

На каждом из множеств (1.6.4) задается своя (частная) функция, являющаяся сужением функции Беллмана. Тем самым реализуется распараллеливание ключевого этапа решения по методу ДП.

Итак, с учетом (1.6.4) полагаем, что при $K \in \mathbf{G}_{N-1}$ и $s \in \overline{1,N-1}$

$$W_s[K] \triangleq (v_s(x,P))_{(x,P) \in \mathcal{D}_s[K]} = (v(x,P))_{(x,P) \in \mathcal{D}_s[K]} \in \mathcal{R}_+[\mathcal{D}_s[K]]. \tag{1.6.6}$$

В частности, при $K \in \mathbf{G}_{N-1}$ определены $\mathcal{D}_1[K] \in \mathcal{P}'(D_1)$ и $W_1[K] \in \mathcal{R}_+[D_1[K]]$. Для целей конкретизации $\mathcal{D}_1[K]$ при $K \in \mathbf{G}_{N-1}$ заметим сначала в связи с

(1.6.4), что $\widetilde{\mathbb{T}}[K;N-2]\subset\mathbf{G}_1$, а потому

$$\forall P\in\widetilde{\mathbb{T}}[K;N-2]\ \ \exists t\in\overline{1,N}\setminus\mathbf{K}_1:P=\{t\}.$$

Как следствие, реализуется [66, (10.4)] свойство

$$\forall K\in\mathbf{G}_{N-1}\ \ \forall(x,P)\in\mathcal{D}_1[K]\ \ \exists t\in\overline{1,N}\setminus\mathbf{K}_1:P=\{t\}.$$

С учетом (1.4.1), (1.4.2) и (1.6.6) получаем, что при $K\in\mathbf{G}_{N-1}$ и $(x,P)\in\mathcal{D}_1[K]$

$$W_1[K](x,P)=v_1(x,P)=\min_{j\in\mathbf{I}(P)}\min_{z\in\mathbb{M}_j}[\mathbf{c}(x,\mathrm{pr}_1(z),P)+c_j(z,P)+v_0(\mathrm{pr}_2(z),P\setminus\{j\})],$$

где $P=\{t\}$ для некоторого $t\in\overline{1,N}\setminus\mathbf{K}_1$ и, как следствие, $\mathbf{I}(P)=\mathbf{I}(\{t\})=\{t\}=P$; тогда

$$W_1[K](x,P)=\min_{j\in P}\min_{z\in\mathbb{M}_j}[\mathbf{c}(x,\mathrm{pr}_1(z),P)+c_j(z,P)+f(\mathrm{pr}_2(z))]\tag{1.6.7}$$

(в (1.6.7) учитываем, что $\mathbf{M}_t\subset\mathfrak{M}$ и $\mathrm{pr}_2(z)\in\mathbf{M}_t$ при $z\in\mathbb{M}_t$, поскольку при $j\in P$ непременно $j=t$; см. также (1.2.3)). Коль скоро в (1.6.7) P — синглетон, то данную формулу можно упростить: следуя [44, (9.7)] введем при $K\in\mathbf{G}_{N-1}$ (см. [79, (62), предложение 7]) непустое множество

$$\mathbb{M}_0[K]\stackrel{\triangle}{=}\{h\in\overline{1,N}\setminus\mathbf{K}_1|\{h\}\in\widetilde{\mathbb{T}}[K;N-2]\}$$

при $K\in\mathbf{G}_{N-1}$. С учетом (1.6.4) получаем, что

$$\mathcal{D}_1[K]=\bigcup_{P\in\widetilde{\mathbb{T}}[K;N-2]}\mathbb{D}_1[P]=\bigcup_{P\in\widetilde{\mathbb{T}}[K;N-2]}\{(x,P):x\in\mathcal{M}_1[P]\},\tag{1.6.8}$$

где $\widetilde{\mathbb{T}}[K;N-2]=\{\{h\}:h\in\mathbb{M}_0[K]\}$ [44, замечание 5.1]; поэтому из (1.6.8) имеем

$$\mathcal{D}_1[K]=\bigcup_{h\in\mathbb{M}_0[K]}\{(x,\{h\}):x\in\mathcal{M}_1[\{h\}]\}=\{(x,\{h\}):h\in\mathbb{M}_0[K],\,x\in\mathcal{M}_1[\{h\}]\}.\tag{1.6.9}$$

В свою очередь, при $K \in \mathbf{G}_{N-1}$ и $h \in \mathbb{M}_0[K]$ имеем $\{h\} \in \widetilde{\mathbb{T}}[K; N-2]$, где $h \in \overline{1,N} \setminus \mathbf{K}_1$; тогда $\{h\} \in \mathbf{G}_1$ и

$$J_1(\{h\}) = \{t \in \overline{1,N} \setminus \{h\} | \{t; h\} \in \mathbf{G}_2\}, \tag{1.6.10}$$

$$\mathcal{M}_1[\{h\}] = \bigcup_{j \in J_1(\{h\})} \mathbf{M}_j. \tag{1.6.11}$$

С учетом (1.6.7) и (1.6.9) имеем при $K \in \mathbf{G}_{N-1}$, $h \in \mathbb{M}_0[K]$ и $x \in \mathcal{M}_1[\{h\}]$ свойство $(x, \{h\}) \in \mathcal{D}_1[K]$ и

$$W_1[K](x, \{h\}) = \min_{z \in \mathbb{M}_h} [\mathbf{c}(x, \mathrm{pr}_1(z), \{h\}) + c_h(z, \{h\}) + f(\mathrm{pr}_2(z))]. \tag{1.6.12}$$

В силу (1.6.9) мы получаем, что посредством (1.6.12) полностью определена функция $W_1[K]$. В свою очередь, для применения (1.6.12) мы должны, располагая $K \in \mathbf{G}_{N-1}$ и $h \in \mathbb{M}_0[K]$, построить $J_1(\{h\})$ с использованием (1.6.10), а затем определить $\mathcal{M}_1[\{h\}]$ посредством (1.6.11), после чего можно воспользоваться формулой (1.6.12) для $x \in \mathcal{M}_1[\{h\}]$. Таким образом, определяются все функции $W_1[K]$, $K \in \mathbf{G}_{N-1}$. Дальнейшее построение будем рассматривать применительно к одному отдельно взятому вычислительному узлу.

Преобразование функции $\mathcal{W}_l[K]$ в $\mathcal{W}_{l+1}[K]$, где $K \in \mathbf{G}_{N-1}$ и $l \in \overline{1,N-2}$, определяется соотношением, подобным (1.4.2) и использующим (1.6.5); в самом деле, согласно (1.6.5) и (1.6.6) при $(x, Q) \in \mathcal{D}_{l+1}[K]$, $s \in \mathbf{I}(Q)$ и $z \in \mathbb{M}_s$ определено значение

$$\mathcal{W}_l[K](\mathrm{pr}_2(z), Q \setminus \{s\}) \in \mathbb{R}_+,$$

которое может использоваться при подсчете $\mathcal{W}_{l+1}[K](x,Q)$. Итак (см. (1.4.2), (1.6.6)), при $K \in \in \mathbf{G}_{N-1}$, $l \in \overline{1,N-2}$ и $(x,Q) \in \mathcal{D}_{l+1}[K]$ имеем [66, (10.17)]

$$\mathcal{W}_{l+1}[K](x,Q) = \min_{j \in \mathbf{I}(Q)} \min_{z \in \mathbb{M}_j} [\mathbf{c}(x, \mathrm{pr}_1(z), Q) + c_j(z,Q) + \mathcal{W}_l[K](\mathrm{pr}_2(z), Q \setminus \{j\})]. \tag{1.6.13}$$

Итак, фиксируем $\mathbb{K} \in \mathbf{G}_{N-1}$. Данному множеству отвечают слои $\mathcal{D}_s[\mathbb{K}]$ (1.6.4), $s \in \overline{1,N-1}$, являющиеся каждый непустым множеством в пространстве позиций. При $s \in \overline{1,N-1}$ определена функция $\mathcal{W}_s[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_s[\mathbb{K}]]$. В частности, определена функция $\mathcal{W}_1[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_1[\mathbb{K}]]$, для конкретного построения кото-

рой следует использовать (1.6.12) и учитывать при этом представление $\mathcal{D}_1[\mathbb{K}]$, определяемое в (1.6.9). Дальнейшее построение функций $\mathcal{W}_1[\mathbb{K}], \ldots, \mathcal{W}_{N-1}[\mathbb{K}]$ осуществляется посредством рекуррентной процедуры на основе (1.6.13). А именно: при $l \in \overline{1,N-2}$ преобразование $\mathcal{W}_l[\mathbb{K}]$ в $\mathcal{W}_{l+1}[\mathbb{K}]$ определяется посредством выражения

$$\mathcal{W}_{l+1}[\mathbb{K}](x,Q) = \min_{j \in \mathbf{I}(Q)} \min_{z \in \mathbb{M}_j} [\mathbf{c}(x, \mathrm{pr}_1(z), Q) + c_j(z,Q) + \mathcal{W}_l[\mathbb{K}](\mathrm{pr}_2(z), Q \setminus \{j\})]$$
$$\forall (x,Q) \in \mathcal{D}_{l+1}[\mathbb{K}]. \tag{1.6.14}$$

Правило (1.6.14) является конкретизацией (1.6.13). Итак, посредством (1.6.14) осуществляется преобразование

$$\mathcal{W}_l[\mathbb{K}] \to \mathcal{W}_{l+1}[\mathbb{K}].$$

Поскольку l выбиралось произвольно, мы получили рекуррентную процедуру

$$\mathcal{W}_1[\mathbb{K}] \to \mathcal{W}_2[\mathbb{K}] \to \ \ldots \ \to \mathcal{W}_{N-1}[\mathbb{K}]. \tag{1.6.15}$$

Осуществление всех расчетов, связанных с (1.6.15), возлагается на один вычислительный узел и проводится независимо от вычислений, выполняемых другими узлами. Мы полагаем, что каждый из узлов реализует вычисления по схеме, аналогичной (1.6.15).

1.7 Построение слоев функции Беллмана в параллельной реализации

Перейдем к построению слоев $v_0 \to v_1 \to \ldots \to V$ Используя процедуру (1.6.15), мы при каждом $s \in \overline{1,N-1}$ получаем множества $\mathcal{D}_s[K], K \in \mathbf{G}_{N-1}$, которые в объединении реализуют D_s,

$$D_s = \bigcup_{K \in \mathbf{G}_{N-1}} \mathcal{D}_s[K] \ \ \forall s \in \overline{1,N-1}; \tag{1.7.1}$$

на каждом из этих множеств определена (после реализации процедуры (1.6.15)) соответствующая функция $\mathcal{W}_s[K] \in \mathcal{R}_+[\mathcal{D}_s[K]]$.

Равенства (1.7.1) позволяют "объединить" процессы, реализуемые отдельными вычислительными узлами. Мы стремимся при этом к построению слоев $v_1, \ldots, v_{N-1}$. При каждом $s \in \overline{1,N-1}$ мы располагаем функциями

$$\mathcal{W}_s[K],\ K \in \mathbf{G}_{N-1}, \tag{1.7.2}$$

области определения которых (то есть множества $\mathcal{D}_s[K]$, $K \in \mathbf{G}_{N-1}$) образуют в силу (1.7.1) покрытие D_s. С учетом (1.6.6) имеем следующее свойство: функция v_s полностью определяется набором (1.7.2). Функции этого набора оказываются в силу (1.6.6) согласованными: при $K_1 \in \mathbf{G}_{N-1}$, $K_2 \in \mathbf{G}_{N-1}$, $(x,P) \in \mathcal{D}_s[K_1] \cap \mathcal{D}_s[K_2]$ непременно

$$\mathcal{W}_s[K_1](x,P) = v_s(x,P) = \mathcal{W}_s[K_2](x,P). \tag{1.7.3}$$

С учетом (1.7.1), (1.7.3) мы получаем следующее простое правило определения функции v_s, где $s \in \overline{1,N-1}$.

Итак, располагая "частными" функциями (1.7.2), мы для определения значения $v_s(x_0, K_0)$, где $(x_0, K_0) \in D_s$, отыскиваем сначала с учетом (1.7.1) множество $\mathbb{K}_0 \in \mathbf{G}_{N-1}$ со свойством $(x_0, K_0) \in \mathcal{D}_s[\mathbb{K}_0]$. Тогда согласно (1.6.6)

$$v_s(x_0, K_0) = \mathcal{W}_s[\mathbb{K}_0](x_0, K_0). \tag{1.7.4}$$

Свойство (1.7.3) говорит о том, что конкретный выбор $\mathbb{K}_0 \in \mathbf{G}_{N-1}$ со свойством $(x_0, K_0) \in \mathcal{D}_s[\mathbb{K}_0]$ может быть произвольным. На основе правила, использующего (1.7.4), находятся все значения функции v_s, а следовательно, и сама эта функция.

Таким образом, располагая функциями (1.6.6), мы находим все слои $v_0, v_1, \ldots, v_N$ функции Беллмана. На основе данных слоев, используя процедуры решения локальных задач, подобные (1.5.2) и (1.5.4), определяем оптимальное ДР; для построения данного ДР в памяти вычислителя следует сохранять все слои $v_1, \ldots, v_N$ функции Беллмана, что видно уже из (1.5.2), (1.5.4).

Алгоритм определения глобального экстремума. Рассмотрим процедуру определения V (1.4.3), при реализации которой в памяти каждого вы-

числительного узла находятся всегда одна "частная" функция вида (1.6.6), то есть один "индивидуальный" слой функции Беллмана.

Рассмотрим для определенности работу одного вычислительного узла, отвечающего множеству $\mathbb{K} \in \mathbf{G}_{N-1}$. Конечной целью процедуры, выполняемой данным узлом, объявляется построение функции

$$\mathcal{W}_{N-1}[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_{N-1}[\mathbb{K}]].$$

Основные этапы итерационной процедуры.

1. Определяем $\mathcal{W}_1[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_1[\mathbb{K}]]$, используя (1.6.9), (1.6.12).
2. Пусть $s \in \overline{1,N-2}$ и "частная" функция $\mathcal{W}_s[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_s[\mathbb{K}]]$ уже построена. Тогда посредством (1.6.13) осуществляется насчитывание значений функции $\mathcal{W}_{s+1}[\mathbb{K}]$, использующее только значения функции $\mathcal{W}_s[\mathbb{K}]$ (в (1.6.13) следует при этом полагать $l = s$). В результате реализуется "частная" функция
$$\mathcal{W}_{s+1}[\mathbb{K}] \in \mathcal{R}_+[\mathcal{D}_{s+1}[\mathbb{K}]].$$
После этого массив значений $\mathcal{W}_s[\mathbb{K}]$ уничтожается и заменяется массивом значений $\mathcal{W}_{s+1}[\mathbb{K}]$: осуществляется перезапись "частных" функций.
3. После последовательного повторения этапов вида 2 получаем частную функцию $\mathcal{W}_{N-1}[\mathbb{K}]$.
Пусть этапы 1–3 завершены каждым из вычислительных узлов, в результате чего найдены все функции
$$\mathcal{W}_{N-1}[K],\ K \in \mathbf{G}_{N-1}. \tag{1.7.5}$$
4. Теперь конструируем v_{N-1}, используя равенство
$$D_{N-1} = \bigcup_{K \in \mathbf{G}_{N-1}} \mathcal{D}_{N-1}[K]. \tag{1.7.6}$$

Для этого при каждом выбранном значении $(\hat{x}, \hat{P}) \in D_{N-1}$ определяем, используя (1.7.6), множество $\hat{K} \in \mathbf{G}_{N-1}$ со свойством $(\hat{x}, \hat{P}) \in \mathcal{D}_{N-1}[\hat{K}]$. Тогда (см. (1.7.5)) мы располагаем значением $\mathcal{W}_{N-1}[\hat{K}](\hat{x}, \hat{P}) \in \mathbb{R}_+$, для которого в силу

(1.6.6) справедливо равенство

$$v_{N-1}(\hat{x},\hat{P}) = \mathcal{W}_{N-1}[\hat{K}](\hat{x},\hat{P}), \tag{1.7.7}$$

завершающее определение $v_{N-1}(\hat{x},\hat{P})$.

Итак, мы находим все значения функции v_{N-1} (логика рассуждений здесь подобна используемой в случае (1.7.4)).

Теперь, располагая функцией v_{N-1} (финальный слой функции Беллмана), мы посредством (1.5.1) определяем глобальный экстремум V. В связи с данной процедурой отметим работу [78], где показано, что определение значения V без построения оптимального ДР может осуществлятся для задачи существенно большей размерности.

1.8 Параллельный алгоритм

Для "вертикального"распараллеливания задачи в системах с распределенной памятью с применением программного интерфейса MPI использован принцип разбиения задачи верхнего уровня на подзадачи меньшей размерности. Для этого главный вычислительный узел формирует семейство $\mathbf{G}_{N-1}$ (1.6.1), элементами которого являются множества мощности $N-1$, а именно, из семейства всех множеств индексов $\overline{1,N}$ строится набор множеств $K \in \mathbf{G}_{N-1}$ (меньшего по мощности на единицу значения N, определяющего количество мегаполисов), т.е. первое множество содержит все номера мегаполисов, за исключением первого в $\mathbf{I}(\overline{1,N})$, второе множество – номера всех мегаполисов за исключением второго в $\mathbf{I}(\overline{1,N})$ мегаполиса и т.д. (таким образом учитываются условия предшествования).

При старте параллельного алгоритма эти множества $K \in \mathbf{G}_{N-1}$ распределяются по работающим MPI-процессам, каждый такой процесс обслуживается отдельным вычислительным узлом, который обсчитывает данное ему множество позиций и вычисляет “индивидуальные” слои функции Беллмана $\mathcal{W}_s[K]$, $s \in \overline{1,N-1}$ (см. раздел 1.6). Поскольку оперативная память для каждого узла является общей и обмена данными между узлами не происходит, то имеет смысл использовать дополнительное распараллеливание для расчётов “индиви-

дуальных" слоев функции Беллмана на уровне каждого вычислительного узла. Для этого будем применять библиотеку OpenMP. Данное обстоятельство позволяет реализовать двойное распараллеливание: на уровне кластера (MPI) и на уровне узла (OpenMP); это позволяет более эффективно использовать вычислительные ресурсы и ускоряет сами вычисления.

После того, как последовательно были построены все "индивидуальные" слои функции Беллмана $\mathcal{W}_s[K]$, $s \in \overline{1,N-1}$, результаты расчётов передаются с узлов, связанных со списками $K \in \mathbf{G}_{N-1}$, и собираются на главном MPI-процессоре, где на их основе, путём склейки (см. раздел 1.7) находится слой функции Беллмана v_{N-1}, (см. (1.7.6) и (1.7.7)).

Далее вычисляем значение глобального экстремума V посредством (1.4.3). Затем главный процессор реализует построение оптимального маршрута путём решения локальных экстремальных задач подобных (1.5.1) и (1.5.3). Применяемая схема независимых вычислений представлена на Рисунке 1.1.

Выделенная при распараллеливании подзадача нахождения оптимальных путей (маршрутов и трасс) по семейству мегаполисов и их городов решается построением, использующим соотношения, подобные (1.5.2) и (1.5.4), с естественным применением динамического программирования. В процессе данного построения перебираются мегаполисы и соответствующие им пункты прибытия и отправления.

На верхнем уровне алгоритм перебирает мегаполисы текущего семейства и соответствующие пути, начинающиеся в них, а для вычисления стоимостей этих путей данный алгоритм рекурсивно вызывает себя же для вычисление оптимальных путей через семейства мегаполисов на единицу меньшего "размера", а точнее, без того мегаполиса, который является текущим в осуществляемом переборе.

Такие рекурсивные вызовы будут заканчиваться на этапе вычисления оптимальных путей через семейства мегаполисов единичного "размера", т.е. через единственный мегаполис, здесь уже путь будет только один. При нахождении оптимальных путей их стоимости будут сохраняться в хэш-таблицу. Оптимальные значения стоимостей будут вычисляться и сохраняться, начиная от меньших размерностей семейств к большим и при вычислении стоимостей по семействам мегаполисов некоторой размерности $s \in \overline{1,N}$ будут использоваться оптимальные значения стоимосте для семейств мегаполисов размерности $s-1$,

уже ранее вычисленные и сохраненные в хэш-таблице. Это сокращает перебор и экономит тем самым время вычисления за счет увеличения потребности в оперативной памяти. Ключ данной хэш-таблицы – семейство мегаполисов, для которого требуется узнать оптимальные стоимости. Значение, получаемое из хэш-таблицы по этому ключу – набор оптимальных значений данного семейства для случаев различных начальных мегаполисов.

Для каждого оптимального решения (пути) в хэш-таблице сохраняются его стоимость и его первый в рабочей нумерации мегаполис. Здесь для сокращения объема потребляемой хэш-таблицей памяти для каждого оптимального значения не хранится вся определяющая его последовательность мегаполисов, потому что необходимость знать эту последовательность возникает только по завершению алгоритма для выдачи соответствующего результата на выход. Тогда эта последовательность легко восстанавливается получением из хэш-таблицы оптимальных значений по соответствующим семействам уменьшающихся размерностей и сбором их "первых"мегаполисов.

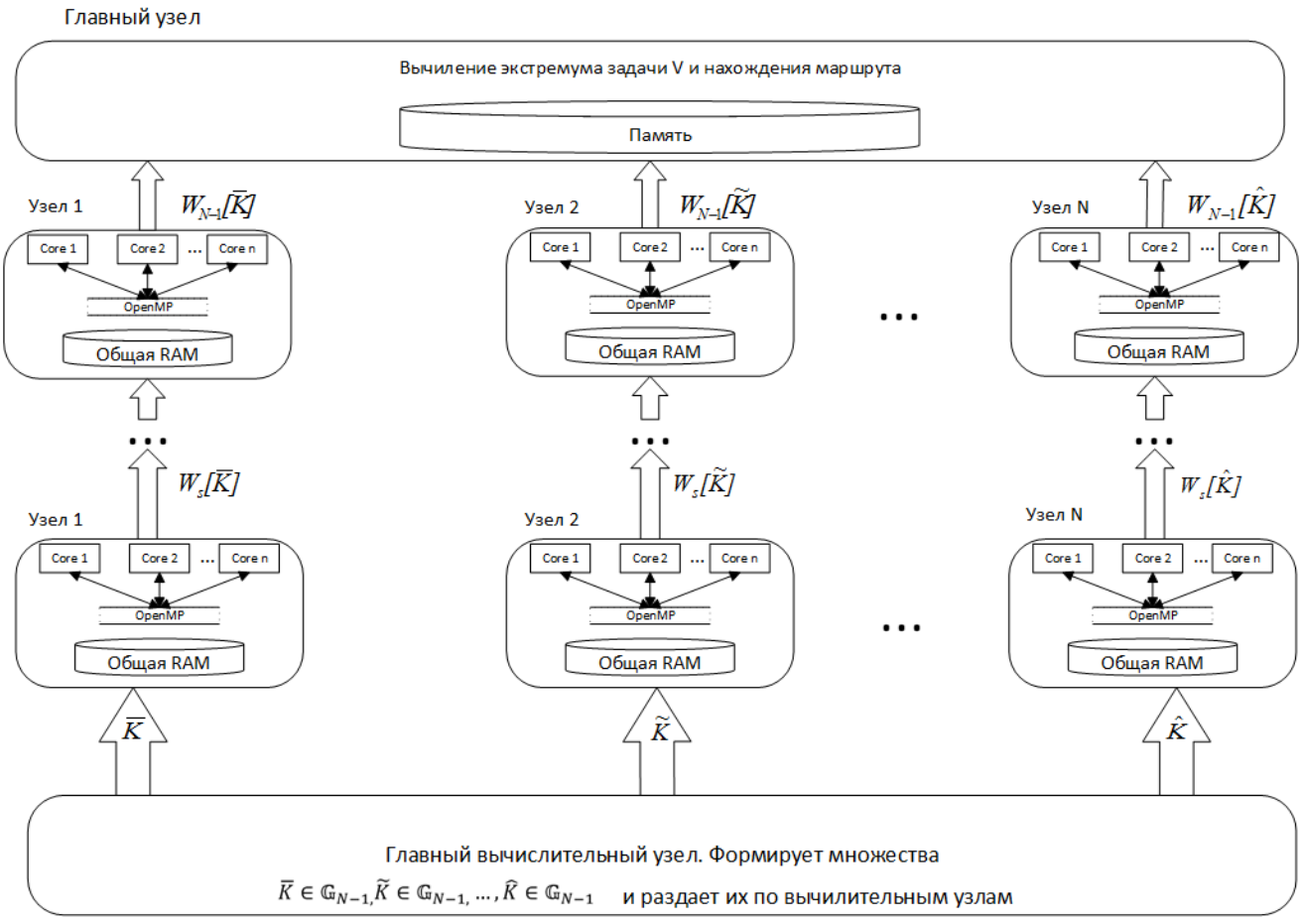

Рисунок 1.1 — Схема независимых вычислений.

Алгоритм определения экстремума с перезаписью.

В тех случаях, когда построение маршрута и трассы не требуется, можно применить алгоритм нахождения глобального экстремума с последовательной перезаписью слоев функции Беллмана, который позволяет сэкономить используемую оперативную память. Такой подход может применяться для тестирования эвристических алгоритмов, для которых важно сравнение получаемого результата (на сколько отличается найденное значение от оптимального). Как можно увидеть из формулы (1.6.13), для расчёта "индивидуального"слоя функции Беллмана $\mathcal{W}_{l+1}[K]$ необходимо знать лишь предыдущий слой $\mathcal{W}_l[K]$, поэтому в памяти вычислительного узла требуется хранить только один этот слой. Из хэш-таблицы удаляются все записи, для которых ключ хэш-таблицы имеет размер меньше номера слоя, который надо оставить в хэш-таблице. Таким образом, на каждом этапе вычисления слоев функции Беллмана происходит перезапись слоя l на слой $l+1$ для $\forall l \in \overline{1,N-2}$. После того, как "индивидуальные"слои $\mathcal{W}_{N-1}[K]$ для каждого множества $K \in \mathbf{G}_{N-1}$ будут найдены узлами кластера, эти слои передаются на главный MPI-узел, где при помощи склейки строится слой основной функции Беллмана v_{N-1}, после этого вычисляется значение глобального экстремума V. Такой подход позволяет существенно сэкономить используемую оперативную память вычислительных узлов и решать задачи большей размерности.

1.9 Функции стоимости в задачах АЭС

Рассмотрим прикладную задачу, в которой функции стоимости изначально зависят от списка невыполненных заданий. Примером такой постановки может является инженерная задача утилизации источников излучения на объектах атомной энергетики. В связи с этим рассмотрим конкретизацию общих построений предыдущих разделов для упомянутой инженерной задачи. Исследуется постановка, в которой необходимо посещение каждого мегаполиса с целью демонтажа одного излучающего элемента "внутри"мегаполиса. Сами мегаполисы получаются всякий раз посредством дискретизации границы ближней зоны соответствующего источника. Другим вариантом роли городов в мегаполисе яв-

ляется представление в виде входов-выходов в помещение, где находится источник радиации. Пусть системе мегаполисов поставлена в соответствие система точечных излучающих объектов $(\mathbf{z}_i)_{i\in\overline{1,N}} : \overline{1,N} \to X$ со свойством: $\mathbf{z}_i \notin M_i$ $\forall i \in \overline{1,N}$. Демонтаж объектов $\mathbf{z}_1, \ldots, \mathbf{z}_N$ является целью посещения мегаполисов, а именно: при посещении мегаполиса $M_{\alpha(i)}$ требуется придти в точку входа $\mathrm{pr}_1(z_i)$, переместиться в точку $\mathbf{z}_{\alpha(i)}$, выполнить работы по демонтажу источника излучения с номером $\alpha(i)$ и затем переместиться в точку выхода $\mathrm{pr}_2(z_i)$. Тем самым реализуется схема (1.2.2). Представляется полезным выделить основные варианты радиационного воздействия на исполнителя а процессе последовательного демонтажа источников.

Прохождение мимо источника излучения s для демонтажа другого источника излучения.

В настоящем изложении ограничимся, следуя конструкциям [80], описанием воздействия одного источника в процессе перемещения из заданной точки плоскости к другой точке (как уже отмечалось, агрегирование таких воздействий при построении функции $\mathbf{c}$ сводится к суммированию результатов этих воздействий) в «регулярном» случае, отвечающем ситуации, когда на траектории перемещения отсутствует упомянутый (активный) источник. Итак, при перемещении из пункта i в пункт j величина затрат (в той модели, которой мы придерживаемся), а, точнее, доза облучения, получаемая при данном перемещении за счет не демонтированного источника излучения s, равна

$$c_{i,j}[\{s\}] = \int_0^T \frac{\gamma_s}{\rho^2_{s,t}(t)} dt = \\ = 4\rho^2_{i,j}\frac{\gamma_s}{v}\int_0^{\rho_{i,j}} \frac{d\rho}{\left(2\rho_{i,j}\rho+\rho^2_{j,s}-\rho^2_{i,s}-\rho^2_{i,j}\right)^2+\left(4\rho^2_{i,j}\rho^2_{i,s}-(\rho^2_{j,s}-\rho^2_{i,s}-\rho^2_{i,j})^2\right)}, \tag{1.9.1}$$

где ρ используется для обозначения евклидова расстояния (при необходимости упомянутые расстояния между точками снабжаются индексами); γ_s определяет интенсивность источника s, а v скорость перемещения исполнителя. Полагаем, что $R(\rho) \stackrel{\triangle}{=} 2\rho_{i,j}\rho + \rho^2_{j,s} - \rho^2_{i,s} - \rho^2_{i,j}$ при $\rho \in [0,\rho_{i,j}]$. Для расчета используется одна из следующих табличных интегральных формул [81, с. 367]:

$$\int \frac{dR}{A^2+R^2} = \frac{1}{A}\mathrm{arctg}\frac{R}{A} + C, \qquad \int \frac{dR}{R^2-A^2} = \frac{1}{2A}\ln\left|\frac{R-A}{R+A}\right| + C, \tag{1.9.2}$$

в зависимости от знака выражения $4\rho_{i,j}^2\rho_{i,s}^2-(\rho_{j,s}^2-\rho_{i,s}^2-\rho_{i,j}^2)^2$. Именно, возможен один из следующих двух случаев:

1. Пусть $4\rho_{i,j}^2\rho_{i,s}^2-(\rho_{j,s}^2-\rho_{i,s}^2-\rho_{i,j}^2)^2\geq 0$. Тогда полагаем, что

$$A=\sqrt{4\rho_{i,j}^2\rho_{i,s}^2-(\rho_{j,s}^2-\rho_{i,s}^2-\rho_{i,j}^2)^2},$$

получая неотрицательное число. Имеем цепочку равенств

$$c_{i,j}[\{s\}]=\frac{2\rho_{i,j}\gamma_s}{v}\int_{R(0)}^{R(\rho_{i,j})}\frac{dR}{R^2+A^2}=\frac{2\rho_{i,j}\gamma_s}{vA}\operatorname{arctg}\frac{R}{A}\Big|_{R(0)}^{R(\rho_{i,j})}=$$
$$=\frac{2\rho_{i,j}\gamma_s}{vA}\left(\operatorname{arctg}\frac{R(\rho_{i,j})}{A}-\operatorname{arctg}\frac{R(0)}{A}\right).$$

2. Пусть $4\rho_{i,j}^2\rho_{i,s}^2-(\rho_{j,s}^2-\rho_{i,s}^2-\rho_{i,j}^2)^2<0$. Тогда полагаем, что

$$A=\sqrt{(\rho_{j,s}^2-\rho_{i,s}^2-\rho_{i,j}^2)^2-4\rho_{i,j}^2\rho_{i,s}^2};$$

получаем, что $A>0$. Используя вторую в (1.9.2) табличную формулу, получаем, что

$$c_{i,j}[\{s\}]=\frac{2\rho_{i,j}\gamma_s}{v}\int_{R(0)}^{R(\rho_{i,j})}\frac{dR}{R^2-A^2}=\frac{\rho_{i,j}\gamma_s}{vA}\ln\left|\frac{R-A}{R+A}\right|\Big|_{R(0)}^{R(\rho_{i,j})}=$$
$$=\frac{\rho_{i,j}\gamma_s}{vA}\left(\ln\left|\frac{R(\rho_{i,j})-A}{R(\rho_{i,j})+A}\right|-\ln\left|\frac{R(0)-A}{R(0)+A}\right|\right).$$

Мы ограничиваемся сейчас рассмотрением таких вариантов задачи (1.2.8), для которых при всяком выборе пунктов i,j,s вышеупомянутого типа значения

$$\ln\left|\frac{R(\rho_{i,j})-A}{R(\rho_{i,j})+A}\right|,\quad \ln\left|\frac{R(0)-A}{R(0)+A}\right|$$

конечны, т. е. определены и являются вещественными числами. Напомним, что в рассматриваемом случае источник не находится на одной прямой между пунктами i и j. Если же источник s лежит на траектории перемещения из пункта i в пункт j, то $c_{i,j}[\{s\}]$ отождествляется с очень большим числом (грубо говоря, в этом случае $c_{i,j}[\{s\}]=\infty$; на самом же деле в вычислительной процедуре достаточно числа, которое заведомо больше в разы, чем самое «затратное» перемещение).

Прохождение к источнику излучения s для его демонтажа (оценивание внутренних работ).

В данном случае полагаем, что пункт j совпадает с s и нас будет интересовать доза облучения, получаемая исполнителем в процессе приближения к источнику излучения, подлежащему демонтажу.

Здесь мы по-прежнему придерживаемся модели, в которой получаемая доза обратно пропорциональна квадрату расстояния до источника, но для исключения некорректности в момент прихода в источник, способной привести к делению на 0, мы в рассматриваемой здесь модели знаменатель подынтегрального выражения в (1.9.1) увеличиваем на единицу. Для учета более интенсивного радиационного воздействия в ближней зоне используем дополнительный множитель 3 (содержательно это может отвечать случаю, когда действие источника не ослабляется препятствиями, что может иметь место на этапе внешних перемещений). Величина затрат (дозы облучения) в данном случае вычисляется по формуле [80]:

$$c_{i,j}[\{s\}] = c_{i,s}[\{s\}] = 3\big(\frac{\gamma_s}{v}\big) \int_0^{\rho_{i,j}} \frac{d\rho}{\rho^2+1} = 3\big(\frac{\gamma_s}{v}\big)\,\mathrm{arctg}(\rho_{i,j}).$$

1.10 Вычислительный эксперимент

В настоящем разделе приведены описание программы и результаты вычислительного эксперимента для решения задачи минимизации дозы облучения работника при демонтаже радиоактивных источников. Рассматривается процедура ДП для ее решения. Данный алгоритм был реализован на $C++11$ и обеспечивает точное решение описанной выше задачи. Программа может выполняться в 64-битных ОС Windows и Linux, и компилироваться с помощью MS Visual Studio выпуска не ниже 2013 и gcc/g++ версии не ниже 4.8 соответственно. Переносимость исходного кода программы между платформами для случаев использования ОС-зависимого API достигается применением директив условной компиляции. Вычисления распараллелены с помощью программного интерфейса MPI в реализациях Microsoft MPI и MPICH2. Программа замеряет и выводит затраченное на нахождение решения время и потребляемый объём

оперативной памяти с помощью переносимого API (для времени) и ОС-зависимого API (для объёма оперативной памяти).

Рассмотрим модельные примеры решения задачи маршрутизациии процесса демонтажа радиоактивных источников на плоскости. Пусть мегаполисы, имитирующие возможные входы/выходы помещений с источниками излучения, получены дискретизацией окружностей: на каждой окружности на равном угловом расстоянии, начиная с точки с 0-й угловой координатой, располагаются (в зависимости от выполняемого вычислительного эксперимента) от 30 до 40 точек. Каждому мегаполису соответствует точечный объект, имитирующий источник излучения в помещении (в ближней зоне). Пусть исходная точка (она же база) процесса демонтажа совпадает с началом координат, т. е. $x^0 = (0{,}0)$; по завершении демонтажа всего оборудования осуществляется возврат на базу (этот этап не оценивается и не участвует в формировании совокупного критерия: это соответствует случаю $f \equiv 0$). Напомним, что функция ρ — суть евклидово расстояния. Пусть скорость движения исполнителя, выполняющего демонтаж, вне помещений в 4 раза больше, чем внутри, что призвано моделировать сложность перемещения внутри каждого мегаполиса, обусловленную наличием тех или иных конструкций и механизмов, мешающих упомянутому быстрому перемещению внутри помещения. Интенсивность излучения источников $(\gamma_j)_{j\in\overline{1,N}}$ генерируется случайным образом в диапазоне от 0,8 до 1.

Далее приведены некоторые результаты решения основной задачи с помощью данной вычислительной программы, реализованой на супервычислителе УРАН. Для модельного примера, заданного 49 мегаполисами, 30-ю городами в каждом мегаполисе и 45 адресными парами, соответствующими условиям предшествования, были получены следующие результаты: суммарная величина дозы облучения – 2.710485, общее время вычислений составило 2 часа 16 мин. 2 сек, максимальный объём используемой оперативной памяти 34292 МБ. График обхода 49 мегаполисов приведен на рис. 1.2.

Для примера решения задачи обхода 51 мегаполиса, 20-ю городами в каждом мегаполисе и при наличии 45 адресных пар, получены следующие результаты: суммарная величина дозы облучения – 2.836942, время счета составило 9 часов 57 мин. 11 сек, максимальный объём используемой оперативной памяти 145400 МБ. График обхода 51 мегаполисов приведен на рис. 1.3.

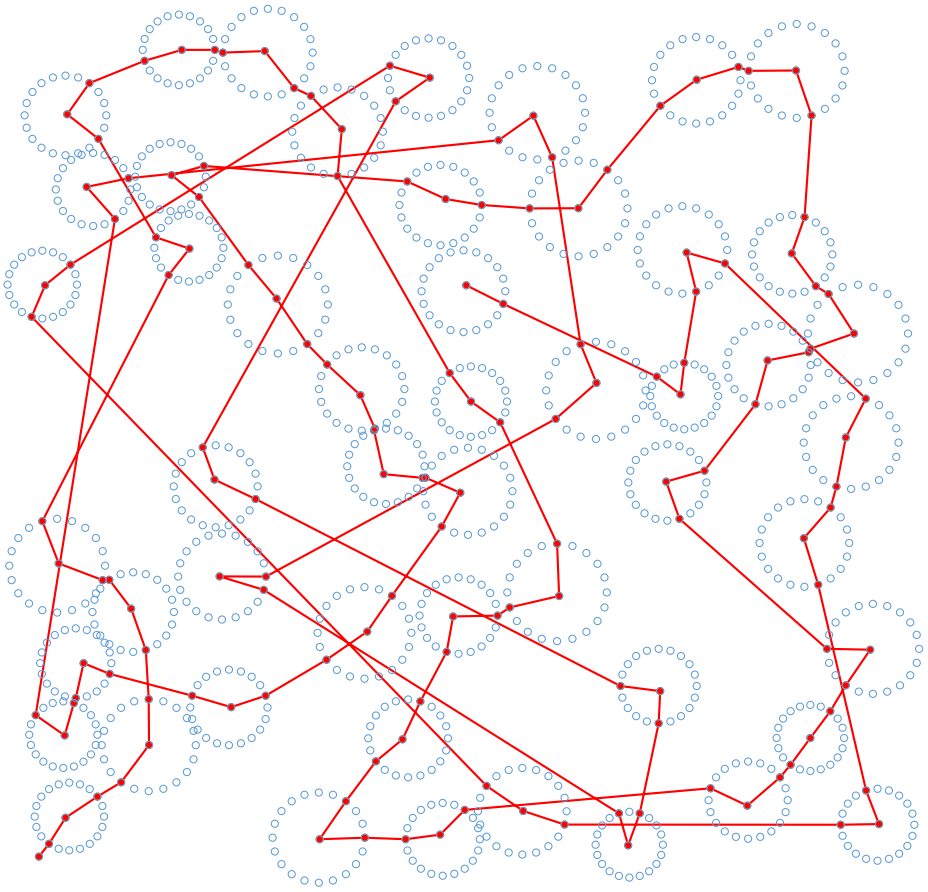

Рисунок 1.2 — Пример вычисления для случая $N = 51$, $|M_i| = 20$, $|\mathbf{K}| = 45$

Для определения зависимости времени расчётов основной задачи от количества условий предшествования был проведен ряд вычислительных экспериментов. Данные эксперименты выполнялись для задачи с 45 мегаполисами, 20-ю городами в каждом мегаполисе, количество условий предшествования изменялось от 25 до 40. График зависимости времени вычисления от количества условий предшествования представлен на рис. 1.4. Как можно заметить из приведённого графика, при увеличении количества условий предшествования, время счета значительно уменьшается, это связано с тем, что мы обрабатываем не весь массив значений функции Беллмана, а лишь ту часть, которая удовлетворяет условиям существенных списков (см. раздел 1.3).

Также немаловажным является определение зависимости получаемого значения экстремума задачи V в зависимости от количества городов в мегаполисе. Данное значение изменяется, так как от изменения количества городов меняется трасса. Для выявления такой зависимости был выполнен ряд экспериментов для варианта с 45-ю магаполисами, 30-ю условиями предшествования,

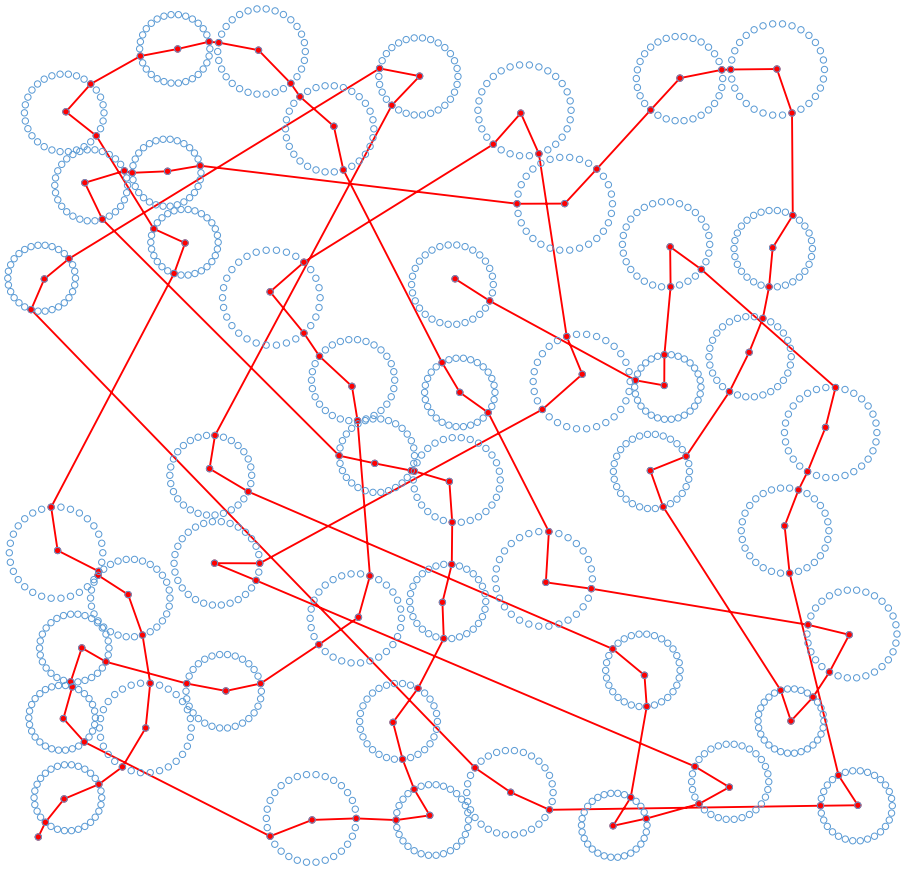

Рисунок 1.3 — Пример вычисления для случая $N = 49$, $|M_i| = 30$, $|\mathbf{K}| = 45$

количество городов изменялось от 2 до 40. На рисунке 1.5 представлена данная зависимость. На нем показано улучшение результата в процентах относительно варианта с 2-мя городами. На графике видно, что для "больших"мегаполисов зависимость стабилизируется, поскольку расстояние между городами становится достаточно близким и их увеличение не приводит к значимому изменению результата.

1.11 Апробация параллельного алгоритма на экземплярах задач TSPLIB SOP

Для тестирования параллельного алгоритма на основе ДП, используемого для поиска оптимального маршрута, мы использовали экземпляры SOP из библиотеки TSPLIB (имеются в виду варианты задачи курьера [3]). К сожалению,

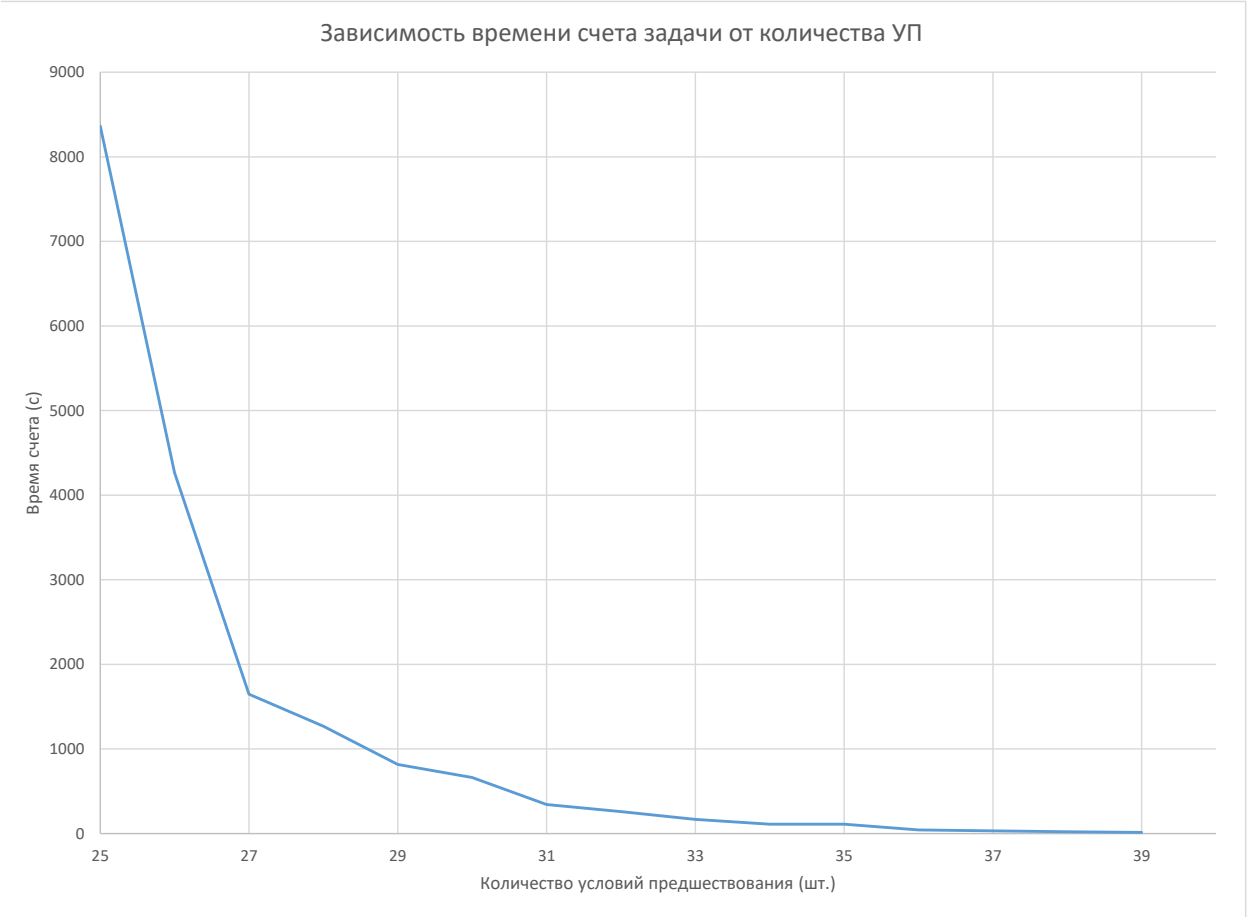

Рисунок 1.4 — Зависимость времени вычисления от количества условий предшествования

для постановки задачи, описанной в текущей главе (GTSP с условиями предшествования и функциями стоимости зависящими от списка невыполненных заданий) автору диссертации не известна открытая библиотека экземпляров задач. Существуют библиотеки экземпляров задач для SOP (sequential ordering problem - проблема последовательного упорядочения), которая аналогична TSP-PC (TSP с условиями предшествования). Также автору известна библиотека GTSP (обобщенный TSP), поддерживаемая Карапетяном [1]. Обе эти библиотеки лишь частично соответствуют представленной постановке задачи. Приведённые в них задачи являются существенно более простыми в сравнении с рассматриваемыми в диссертации. Поскольку в диссертации наибольшее внимание уделяется задаче с условиями предшествования, было решено протестировать алгоритм на экземплярах SOP из TSPLIB.

[1] see url http://www.cs.nott.ac.uk/ pszdk/gtsp.html

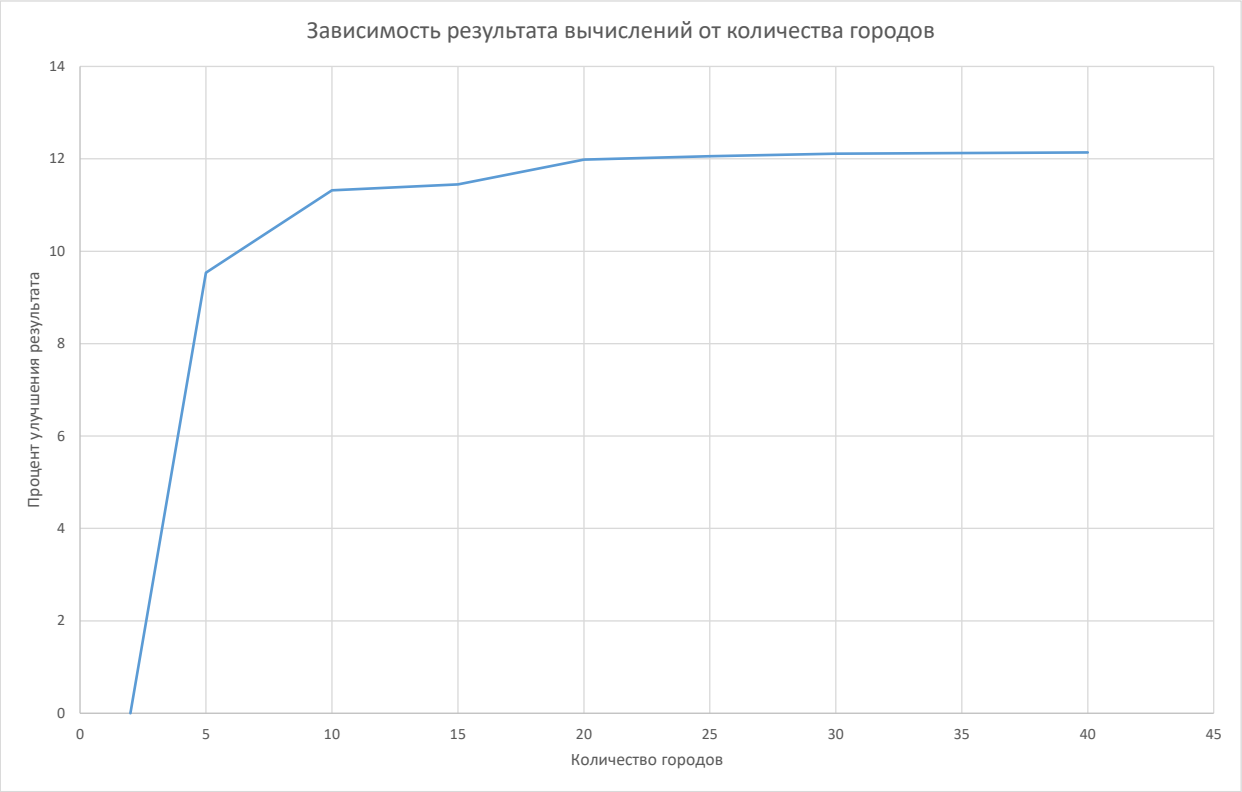

Рисунок 1.5 — Зависимость V от количества городов в мегаполисе

Чтобы адаптировать библиотеку к нашему решателю (и постановке задачи), мы рассматривали города SOP как одноэлементный мегаполис и убрали зависимость функции стоимости от списка невыполненных заданий. Эти предположения значительно упростили нашу первоначальную формулировку, но позволили нам протестировать производительность предложенного алгоритма для нахождения оптимального маршрута в упомянутых известных и более простых задачах.

В результате вычислений было решено значительное количество экземпляров TSPLIB SOP (ESC07.sop, ESC11.sop, br17.12.sop, ESC12.sop, br17.10.sop, ry48p.4.sop, rbg109a .sop, p43.4.sop, ft53.4.sop, rbg150a.sop, ft70.4.sop, ESC25.sop, p43.3.sop, ft53.3.sop). Для всех этих проблем было найдено оптимальное решение, которое сравнивалось с результатами, доступными на веб-сайте TSPLIB. Помимо этого удалось найти оптимальные решения для двух экземпляров задач, которые были отмечены как нерешённые на сайте, а также эти задачи не решены в признанной статье по SOP [20], ry48p.3 (которая, по-видимому, впервые была решена на оптимальность в [22]) и kro124p.4.

Для экземпляра задачи ry48p.3 (48 города и 179 условия предшествования), были получены следующие результаты:

- Значение оптимального решения $V = 19894$.
- Найденный оптимальный маршрут 21,10,14,16,42,29,36,5,43,30,39,15,40, 4,41,47,38,20,46,32,45,35,26,18,27,22,2,13,24,12,31,23,9,44,34,3,25,1,28,33, 11,19,6,17,37,8,7,48.
- Время счёта составило 2331 секунда.
- Используемый объём оперативной памяти 12003 MB.

Для задачи из SOP TSPLIB kro124p.4 (100 городов и 2404 условия предшествования), были получены следующие результаты:

- Значение оптимального решения задачи $V = 76103$.
- Оптимальный маршрут 5,89,87,9,62,27,57,39,1,81,32,4,84,56,76,7,88,96, 65,25,64,74,48,66,68,49,47,13,54,26,11,31,20,16,44,82,83,14,10,92,29,86,50, 42,59,71,35, 98,52,15,21,69,93,17,79,30,55,91,78,61,34,6,8,60,95,77,12,75,94, 51,40,99,70,2,28,33,85,22,90,73,24,67,80,97,58,37,23,18,3,41,53,63,72,43,36, 38,45,19,46,100.
- Время вычисления составило 84603 секунды.
- Объём используемой оперативной памяти 128248 MB.

Алгоритм продемонстрировал свою работоспособность и эффективность, тем самым доказав применимость для определённого круга задач, где другие алгоритмы не способы находить оптимальное решение (имеются в виду вышеупомянутые нерешеные задачи из TSPLIB SOP). В то же время построенный на основе ДП оптимальный алгоритм "справляется"с еще более сложными и имеющими практический смысл задачами.

1.12 Жадный алгоритм в задаче с АЭС

В предыдущих разделах рассматривался алгоритм точного решения задачи маршрутизации, но порой возникают случаи, когда ДП не может справиться с поставленной задачей в силу трудностей вычислительной реализации (имеются в виду задачи существенной размерности). В таких случаях возникает необходимость применения эвристических алгоритмов.

В данном разделе рассматривается решение нашей основной задачи маршрутизации жадным алгоритмом (учитывающим условия предшествования и зависимость функции стоимости от списка невыполненных заданий), аналогичным [82, Раздел 6]. Поэтому, (в связи с построениями [82, Раздел 6]) отметим здесь лишь краткую схему реализации жадного алгоритма. Жадный алгоритм (greedy algorithm) — это алгоритм, который на каждом шаге делает локально оптимальный выбор. Решение, найденное таким образом, не всегда оказывается оптимальным, но в ряде случаев все-таки доставляет приемлемое качество процесса.

Фиксируем базу x^0 предположим $\mathbf{z}^{(0)} \stackrel{\triangle}{=} (x^0,x^0)$, и рассмотрим задачу

$$\mathbf{c}(x^0,\mathrm{pr}_1(z),\overline{1,N}) + c_j(z,\overline{1,N}) \longrightarrow \min, j \in \mathbf{I}(\overline{1,N}), z \in \mathbb{M}_j. \tag{1.12.1}$$

Далее мы выбираем $\mathbf{j}_1 \in \mathbf{I}(\overline{1,N})$ и $\mathbf{z}^{(1)} \in \mathbb{M}_{\mathbf{j}_1}$ для которого

$$\begin{gathered}\mathbf{c}(x^0,\mathrm{pr}_1(\mathbf{z}^{(1)}),\overline{1,N}) + c_{\mathbf{j}_1}(\mathbf{z}^{(1)},\overline{1,N}) = \\ = \min_{j\in\mathbf{I}(\overline{1,N})}\min_{z\in\mathbb{M}_j}\big[\mathbf{c}\big(x^0,\mathrm{pr}_1(z),\overline{1,N}\big)+c_j(z,\overline{1,N})\big].\end{gathered} \tag{1.12.2}$$

Таким образом получаем, что

$$(\mathrm{pr}_2(\mathbf{z}^{(1)}), \overline{1,N} \setminus \{\mathbf{j}_1\}) \in D_{N-1}.$$

Теперь у нас найдена вышеупомянутая позиция и мы знаем стоимость первого перемещения. Далее рассмотрим следующий шаг

$$\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(z),\overline{1,N}\setminus\{\mathbf{j}_1\}) + c_j(z,\overline{1,N}\setminus\{\mathbf{j}_1\}) \longrightarrow \min, j \in \mathbf{I}(\overline{1,N}\setminus\{\mathbf{j}_1\}), z \in \mathbb{M}_j.$$

Выбираем $\mathbf{j}_2 \in \mathbf{I}(\overline{1,N} \setminus \{\mathbf{j}_1\})$ и $\mathbf{z}^{(2)} \in \mathbb{M}_{\mathbf{j}_2}$ для которого

$$\begin{gathered}\mathbf{c}(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(\mathbf{z}^{(2)}),\overline{1,N}\setminus\{\mathbf{j}_1\}) + c_{\mathbf{j}_2}(\mathbf{z}^{(2)},\overline{1,N}\setminus\{\mathbf{j}_1\}) = \\ = \min_{j\in\mathbf{I}(\overline{1,N}\setminus\{\mathbf{j}_1\})}\min_{z\in\mathbb{M}_j}\big[\mathbf{c}\big(\mathrm{pr}_2(\mathbf{z}^{(1)}),\mathrm{pr}_1(z),\overline{1,N}\setminus\{\mathbf{j}_1\}\big)+c_j(z,\overline{1,N}\setminus\{\mathbf{j}_1\})\big].\end{gathered} \tag{1.12.3}$$

Получаем следующее выражение

$$(\mathrm{pr}_2(\mathbf{z}^{(2)}), \overline{1,N} \setminus \{\mathbf{j}_1;\mathbf{j}_2\}) \in D_{N-2}$$

Дальнейшая конструкция реализуется аналогично (1.12.2) и (1.12.3) до исчерпания всего списка $\overline{1,N}$. Мы получаем следующие конечные процессы

$$(\mathbf{j}_k)_{k\in\overline{1,N}} : \overline{1,N} \longrightarrow \overline{1,N},$$

$$(\mathbf{z}^{(k)})_{k\in\overline{0,N}} : \overline{0,N} \longrightarrow \mathbb{X} \times \mathbb{X}.$$

К тому же, $\mathbf{i}[x^0] \triangleq (\mathbf{j}_k)_{k\in\overline{1,N}} \in \mathbf{A}$ и $(\mathbf{z}^{(k)})_{k\in\overline{0,N}} \in \mathcal{Z}_{\mathbf{i}[x^0]}[x^0]$. Конечно, значение

$$\mathfrak{C}_{\mathbf{i}[x^0]}[(\mathbf{z}^{(k)})_{k\in\overline{0,N}}] \in \mathbb{R}_+ \tag{1.12.4}$$

соответствует нашему начальному состоянию x^0. Поэтому мы вводим обозначение

$$w[x^0] \triangleq \mathfrak{C}_{\mathbf{i}[x^0]}[(\mathbf{z}^{(k)})_{k\in\overline{0,N}}]. \tag{1.12.5}$$

Мы рассматриваем (1.12.5) как верхнюю оценку для V и используем $\mathbf{i}[x_0]$ как решение, соответствующее жадному алгоритму.

Для сравнения жадного алгоритма с оптимальным решением методом ДП был выполнен вычислительный эксперимент. На рисунках 1.6 и 1.7 представлены результаты решения одной задачи методом ДП и жадным алгоритмом. Расчёты выполнялись для задачи N=25 (количество мегаполисов), в каждом мегаполисе 10 городов. Количество условий предшествования равно 0. Для точного решения задачи были получены следующие результаты:

- Значение оптимального решения $V = 0{,}681090$.
- Время счёта составило 31155 секунд.
- Используемый объём оперативной памяти 2056 MB.

Для жадного алгоритма были получены следующие результаты:

- Значение $w[x^0] = 0{,}837732$.
- Время счёта составило 5 секунд.
- Используемый объём оперативной памяти 50 MB.

Из приведённых данных можно увидеть, что выигрыш точного решения относительно жадного алгоритма составляет 23%. Но жадный алгоритм выполняет вычисления практически мгновенно. Эту особенность можно использовать в тех случаях, когда нам нужно быстро выполнить вычисления, для того чтобы оценить верхнюю границу для оптимального решения или в экстренных

случаях, когда время вычисления критично и необходимо получить решение немедленно.

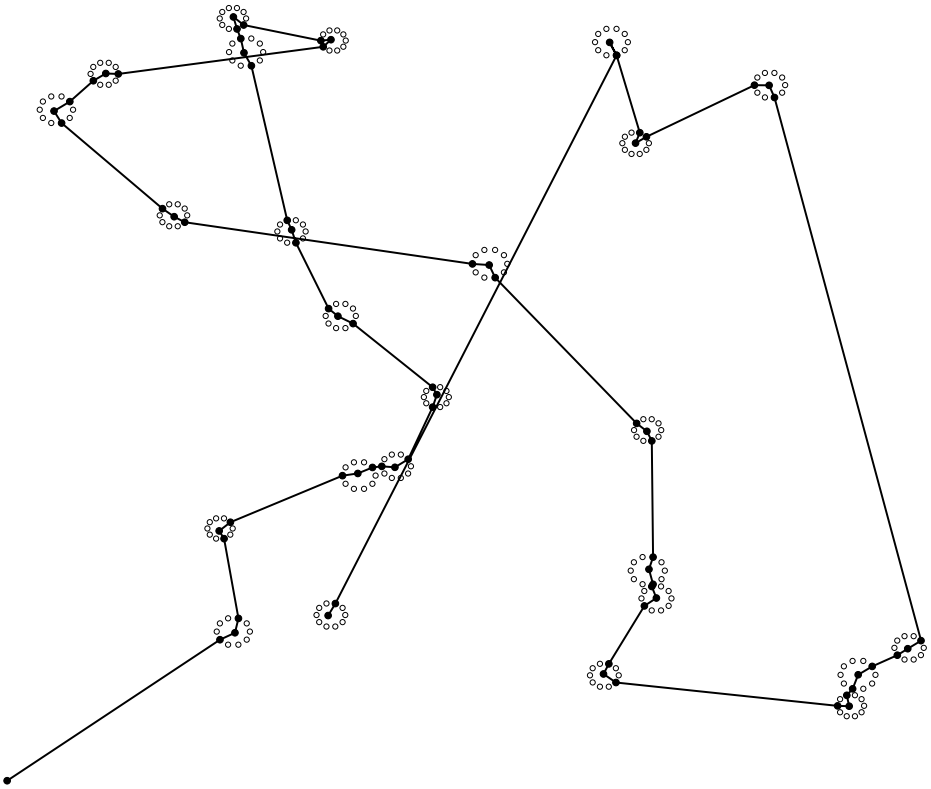

Рисунок 1.6 — Результат точного решения задачи методом ДП на 25 мегаполисов

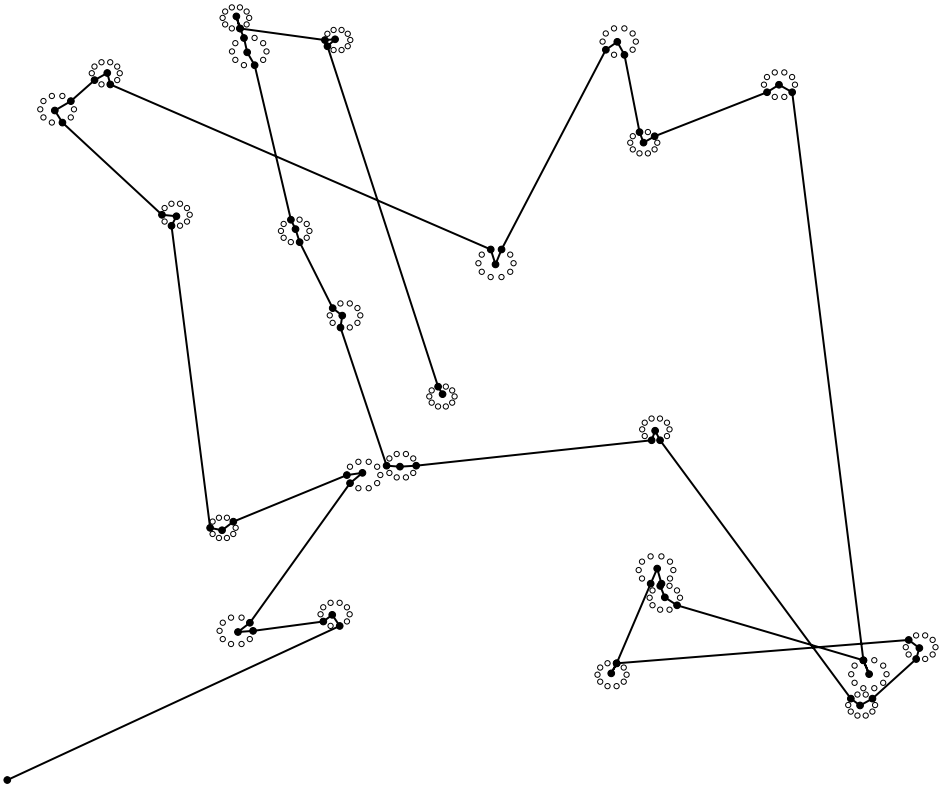

Рисунок 1.7 — Результат решения задачи жадным алгоритмом на 25 мегаполисов

Глава 2. Мультивставка в маршрутных задачах

2.1 Краткое введение

Настоящее исследование продолжает [73–75; 83], где рассматривались процедуры локального улучшения эвристических решений в задаче последовательного обхода мегаполисов с условиями предшествования и функциями стоимости, допускающими зависимость от списка заданий. Прототипом упомянутой задачи является известная труднорешаемая ЗК; см. [1–5; 7; 8] и др. Трудности вычислительной реализации, присущие ЗК, в упомянутой более общей задаче только усугубляются. Все это делает актуальным исследование методов работы с эвристиками и возможностей улучшения результатов, доставляемых этими эвристиками. Такая возможность может, в частности, осуществляться посредством применения оптимизирующих вставок, при построении которых можно использовать широко понимаемое ДП. Однократное применение оптимизирующей вставки умеренной размерности приводит, как правило, лишь к несущественному улучшению достигаемого результата. Можно, однако (см. [73; 74; 83]), использовать итерационные процедуры с последовательным изменением локализации вставки. В данной работе предлагается другой прием: для улучшения качества эвристики планируется использовать сразу конечный набор оптимизирующих вставок, то есть мультивставку. Это приводит к более сильному преобразованию исходной эвристики. Вставки (точнее, элементарные вставки), составляющие мультивставку, могут обслуживаться, как показано в работе, различными вычислителями, что открывает возможность применения параллельных алгоритмов.

В связи с применением параллельных алгоритмов для решения задач маршрутизации с ограничениями отметим [40; 44; 79; 84]. Однако схема распараллеливания, связанная с мультивставками, существенно отличается от [40; 44; 79; 84], где объектами построения были слои функции Беллмана. В настоящей работе распараллеливание связано с распределением вычислительных ресурсов по каналам, связанным каждый с "элементарной" вставкой, являющейся фрагментом мультивставки.

2.2 Общие понятия и обозначения

В данном разделе вводятся понятия и определения, которые не были введены в предыдущей главе. Если X и Y — непустые множества, то через Y^X обозначаем (непустое) множество всех отображений (функций) из X в Y, что соответствует [85, с.77]; если $\mathbf{f} \in Y^X$ и $Z \in \mathcal{P}(X)$, то $\mathbf{f}^1(Z) \stackrel{\triangle}{=} \{\mathbf{f}(z) : z \in Z\} \in \mathcal{P}(Y)$ (при $x \in X$ в виде $\mathbf{f}(x) \in Y$ имеем значение $\mathbf{f}$ в точке x) есть образ Z при действии $\mathbf{f}$; ясно, что $(Z \neq \varnothing) \Rightarrow (\mathbf{f}^1(Z) \neq \varnothing)$. Если A, B, C и D — непустые множества, $\varphi \in D^{A \times B \times C}$, $x \in A \times B$ и $y \in C$, то определено $\varphi(x,y) = \varphi(x_1, x_2, y) \in D$, где $x_1 \stackrel{\triangle}{=} \mathrm{pr}_1(x)$ и $x_2 \stackrel{\triangle}{=} \mathrm{pr}_2(x)$.

Если P и Q — непустые множества, то через $(\mathrm{Bi})[P;Q]$ обозначаем множество всех биекций [65, с.87] множества P на Q. Для любых непустых множеств S, T и биекции $\psi \in (\mathrm{Bi})[S;T]$ определена биекция $\psi^{-1} \in (\mathrm{Bi})[T;S]$ обратная к ψ, для которой

$$(\psi(\psi^{-1}(t)) = t \ \ \forall t \in T) \& (\psi^{-1}(\psi(s)) = s \ \ \forall s \in S).$$

Перестановка непустого множества A определяется как биекция A на себя; итак, $(\mathrm{Bi})[A;A]$ есть множество всех перестановок A. Для произвольных непустых множеств P,Q и R, отображений $g \in Q^P$ и $h \in R^Q$ полагаем, как обычно, что $h \circ g \in R^P$ есть композиция g и h (итак, $(h \circ g)(x) \stackrel{\triangle}{=} h(g(x)) \ \ \forall x \in X$); при $g \in (\mathrm{Bi})[P;Q]$ и $h \in (\mathrm{Bi})[Q;R]$ непременно $h \circ g \in (\mathrm{Bi})[P;R]$.

2.3 Постановка основной задачи

В ряде случаев в прикладных задачах возникает необходимость решать задачи "большой" размерности (при $\mathbf{n} >> 50$). В таких случаях практическое решение по схеме ДП не может быть реализовано ввиду известных ограничений, характерных для современной вычислительной техники (недостаток оперативной памяти для хранения значений функции Беллмана), хотя теоретические конструкции для нахождения точного решения, описанные в предыдущей главе, применимы и для задач любой размерности, в том смысле, что они

всегда определяют структуру оптимальных решений. В данном разделе описывается постановка задачи "большой" размерности, которая практически может решаться только при помощи эвристик. Результаты, доставленные этими эвристиками, мы будем улучшать при помощи оптимизирующей мультивставки.

Всюду в дальнейшем фиксируем непустое множество X, точку $x_0 \in X$, число $\mathbf{n} \in \mathbb{N}$, $\mathbf{n} \geqslant 3$, множества

$$\mathbf{L}_1 \in \mathrm{Fin}(X), \ldots, \mathbf{L}_n \in \mathrm{Fin}(X), \tag{2.3.1}$$

а также (непустые) отношения

$$\mathbb{L}_1 \in \mathcal{P}'(\mathbf{L}_1 \times \mathbf{L}_1), \ldots, \mathbb{L}_{\mathbf{n}} \in \mathcal{P}'(\mathbf{L}_{\mathbf{n}} \times \mathbf{L}_{\mathbf{n}}). \tag{2.3.2}$$

Полагаем в дальнейшем, что выполнены следующие условия:

$$(\mathbf{x}_0 \notin \mathbf{L}_j \;\; \forall j \in \overline{1,\mathbf{n}}) \& (\mathbf{L}_p \cap \mathbf{L}_q = \varnothing \;\; \forall p \in \overline{1,\mathbf{n}} \;\; \forall q \in \overline{1,\mathbf{n}} \setminus \{p\}).$$

Множества (2.3.1) именуем мегаполисами; с каждым из мегаполисов связываем варианты выполнения работ, именуемых внутренними. Рассматриваем процессы следующего вида

$$\mathbf{x}_0 \to (x_{1,1} \in \mathbf{L}_{\alpha(1)} \rightsquigarrow x_{1,2} \in \mathbf{L}_{\alpha(1)}) \to \ldots \to (x_{\mathbf{n},1} \in \mathbf{L}_{\alpha(\mathbf{n})} \rightsquigarrow x_{\mathbf{n},2} \in \mathbf{L}_{\alpha(\mathbf{n})}) \tag{2.3.3}$$

при ограничениях $(x_{1,1}, x_{1,2}) \in \mathbb{L}_{\alpha(1)}, \ldots, (x_{\mathbf{n},1}, x_{\mathbf{n},2}) \in \mathbb{L}_{\alpha(\mathbf{n})}$. Здесь α — перестановка в $\overline{1,\mathbf{n}}$, также удовлетворяющая ограничениям. Кортеж $(\alpha,(x_{1,1},x_{1,2}),\ldots,(x_{\mathbf{n},1},x_{\mathbf{n},2}))$ выбирается, как и в [73–75; 83], с целью минимизации аддитивного критерия, Прямые стрелки в (2.3.3) отвечают внешним (по смыслу) перемещениям, а волнистые — выполнению внутренних работ; $\mathbf{x}_0$ — база процесса, а отношения (2.3.2) определяют возможные варианты выполнения внутренних работ. Основные элементы постановки соответствуют разделу 1.2, но все же их имеет смысл напомнить и ввести дополнительные обозначения.

Пусть $\mathbf{P} \triangleq (\mathrm{bi})[\overline{1,\mathbf{n}}]$ (множество всех перестановок индексного множества $\overline{1,\mathbf{n}}$, именуемых (полными) маршрутами); итак, в (2.3.3) $\alpha \in \mathbf{P}$. Возможны, однако, ограничения и на выбор α. Имеются в виду условия предшествования. Для целей их формализации введем множество $\mathcal{K} \in \mathcal{P}(\overline{1,\mathbf{n}} \times \overline{1,\mathbf{n}})$, элементами

которого являются УП индексов, именуемые адресными. Тогда [74, (2.6)]

$$\mathcal{A} \stackrel{\triangle}{=} \{\alpha \in \mathbf{P}|\ \forall z \in \mathcal{K}\ \ \forall t_1 \in \overline{1,\mathbf{n}}\ \ \forall t_2 \in \overline{1,\mathbf{n}}\ \ ((\mathrm{pr}_1(z) = \alpha(t_1))\&(\mathrm{pr}_2(z) = \alpha(t_2))) \Rightarrow (t_1 < t_2)\} = \{\alpha \in \mathbf{P}|\alpha^{-1}(\mathrm{pr}_1(z)) < \alpha^{-1}(\mathrm{pr}_2(z))\ \ \forall z \in \mathcal{K}\} \tag{2.3.4}$$

есть множество всех маршрутов (исходной задачи), допустимых по предшествованию или, короче, $\mathcal{K}$-допустимых. Следуя [74, (2.7)], полагаем, что

$$\forall \mathcal{K}_0 \in \mathcal{P}'(\mathcal{K})\ \exists z_0 \in \mathcal{K}_0 : \mathrm{pr}_1(z_0) \neq \mathrm{pr}_2(z)\ \ \forall z \in \mathcal{K}_0. \tag{2.3.5}$$

Тогда [54, часть 2] $\mathcal{A} \neq \varnothing$, то есть $\mathcal{A} \in \mathcal{P}'(\mathbf{P})$. Отметим, что

$$\mathfrak{X} \stackrel{\triangle}{=} \{x_0\} \cup \left(\bigcup_{i=1}^{n} \mathbf{L}_i\right) \in \mathrm{Fin}(X),$$

$\tilde{\mathfrak{Z}} \stackrel{\triangle}{=} (\mathfrak{X} \times \mathfrak{X})^{\overline{0,\mathbf{n}}} \in \mathrm{Fin}((X \times X)^{\overline{0,\mathbf{n}}})$. Если $\alpha \in \mathbf{P}$, то в виде

$$\mathfrak{Z}_\alpha \stackrel{\triangle}{=} \{(z_i)_{i \in \overline{0,\mathbf{n}}} \in \tilde{\mathfrak{Z}} | (z_0 = (\mathbf{x}_0, \mathbf{x}_0))\&(z_t \in \mathbb{L}_{\alpha(t)}\ \ \forall t \in \overline{1,\mathbf{n}}\} \in \mathrm{Fin}(\tilde{\mathfrak{Z}}) \tag{2.3.6}$$

имеем множество всех трасс или траекторий, согласованных с маршрутом α; если $\alpha \in \mathcal{A}$ и $\mathbf{z} \in \mathfrak{Z}_\alpha$, то УП $(\alpha, \mathbf{z})$ рассматриваем как допустимое решение (ДР) "большой" задачи. Тогда

$$\mathbf{D} \stackrel{\triangle}{=} \{(\alpha,\mathbf{z}) \in \mathcal{A} \times \tilde{\mathfrak{Z}} | \mathbf{z} \in \mathfrak{Z}_\alpha\} \in \mathrm{Fin}(\mathcal{A} \times \tilde{\mathfrak{Z}}) \tag{2.3.7}$$

есть множество всех ДР исходной задачи. Каждое ДР имеет иерархическую структуру: выбор трассы подчинен выбору маршрута.

В части определения функций стоимости следуем [74, (2.12)]: при $\mathbf{N} \stackrel{\triangle}{=} \mathcal{P}'(\overline{1,\mathbf{n}})$ полагаем заданными функции

$$\mathbf{c}^\natural \in \mathcal{R}_+[\mathfrak{X} \times \mathfrak{X} \times \mathbf{N}],\ c_1^\natural \in \mathcal{R}_+[\mathfrak{X} \times \mathfrak{X} \times \mathbf{N}], \ldots, c_N^\natural \in \mathcal{R}_+[\mathfrak{X} \times \mathfrak{X} \times \mathbf{N}],\ f^\natural \in \mathcal{R}_+[\mathfrak{X}]. \tag{2.3.8}$$

В (2.3.8) предполагается, что функция $\mathbf{c}^\natural$ оценивает в существенной части перемещения между мегаполисами, а также из точки $\mathbf{x_0}$ к мегаполисам.

Функции $c_1^\natural, \ldots, c_N^\natural$ оценивают (в существенной части) работы, выполняемые при посещении мегаполисов и называемые внутренними. Функция $f^\natural$ оценивает терминальное состояние процесса.

В прикладной задаче, связанной с демонтажем излучающих элементов, функции $\mathbf{c}^\natural, c_1^\natural, \ldots, c_N^\natural$ характеризуют дозы радиации, полученные при внешних перемещениях и выполнении внутренних работ. Конкретные формулы, определяющие $\mathbf{c}^\natural, c_1^\natural, \ldots, c_N^\natural$, приведены в [80] и определяются интегрированием нелинейных функций на промежутках времени, определяющих каждое конкретное перемещение (при определении $\mathbf{c}^\natural, c_1^\natural, \ldots, c_N^\natural$ возникают при этом некоторые особенности, связанные с возможностью неограниченного приближения к источнику излучения). Функция $f^\natural$ может характеризовать уровень остаточного радиационного фона после демонтажа всех излучающих элементов. В простейшем случае можно полагать функцию $f^\natural$ тождественно равной нулю.

В терминах (2.3.8) определяется аддитивный критерий: если $\alpha \in \mathbf{P}$ и $\mathbf{z} \in \mathfrak{Z}_\alpha$, то качество УП $(\alpha, \mathbf{z})$ оцениваем значением

$$\hat{\mathfrak{G}}_\alpha[\mathbf{z}] \triangleq \sum_{t=1}^{n} [\mathbf{c}^\natural(\mathrm{pr}_2(\mathbf{z}(t-1)), \mathrm{pr}_1(\mathbf{z}(t)), \alpha^1(\overline{t,\mathbf{n}})) + c^\natural_{\alpha(t)}(\mathbf{z}(t), \alpha^1(\overline{t,\mathbf{n}}))] + f^\natural(\mathrm{pr}_2(z(\mathbf{n}))). \tag{2.3.9}$$

Выражение (2.3.9) определяет аддитивный критерий исследуемой задачи. Основная (аддитивная "большая") задача имеет следующий вид:

$$\hat{\mathfrak{G}}_\alpha[\mathbf{z}] \to \min\,,\ \alpha \in \mathcal{A}, \mathbf{z} \in \mathfrak{Z}_\alpha. \tag{2.3.10}$$

Она характеризуется значением (экстремумом)

$$V \triangleq \min_{\alpha \in \mathcal{A}} \min_{\mathbf{z} \in \mathfrak{Z}_\alpha} \hat{\mathfrak{G}}_\alpha[\mathbf{z}] = \min_{(\alpha, \mathbf{z}) \in \mathbf{D}} \hat{\mathfrak{G}}_\alpha[\mathbf{z}] \in \mathbb{R}_+ \tag{2.3.11}$$

и непустым множеством оптимальных решений. Однако в случае, когда задача (2.3.10) имеет достаточно большую размерность (прежде всего, при большом значении $\mathbf{n}$), нахождение V и ДР, доставляющих V (2.3.11), крайне затруднено, хотя общие принципы здесь известны (см. в частности, процедуры на основе широко понимаемого ДП [71;74;77]). Стоит отметить, что структура оптимального решения понятна и характеризуется в схеме ДП, но не выполнима практически

из-за трудностей с вычислительной реализацией. В этой связи основное внимание в дальнейшем изложении уделяется вставкам и мультивставкам на основе ДП (см. [73–75; 83]).

2.4 Оптимизирующие вставки: общие свойства

Всюду в дальнейшем фиксируем $N \in \overline{2,\mathbf{n}-1}$ в качестве "длины" возможной (однократной) вставки; в конкретных построениях полагается, конечно, что значение N является "умеренным"; последнее связано с возможностью применения ДП в пределах вставки. Локализация вставки определяется [73–75; 83] значением $\nu \in \overline{0,\mathbf{n}-N}$. Фиксируя ν, мы заменяем фрагмент ДР "большой" задачи локально оптимальным. В настоящем разделе ограничимся конструированием однократных вставок, как и в [73–75; 83]. В этой связи совсем кратко напомним построения [73–75; 83], несколько изменяя обозначения с тем, чтобы в дальнейшем применять данные построения уже для исследования мультивставок.

Приводимые ниже конструкции с использованием однократных вставок важны для последующих (основных для данной работы) построений с применением мультивставок, т.е. конечных наборов (однократных) вставок. В связи с использованием последних имеет смысл отметить особо осложняющие обстоятельства.

Так, в частности, условия предшествования в своем целостном виде относятся к "большой" задаче. При построении вставки приходится "вырезать" фрагмент упомянутых условий, локализующийся в пределах вставки. Если же используется мультивставка, то нам потребуется уже конечный набор таких фрагментов. Варьируя допустимый по предшествованию "глобальный" маршрут в пределах вставки, мы должны сохранить упомянутое свойство (также "глобальной") допустимости. Это требует, конечно, достаточно специального способа варьирования маршрута; подробное описание упомянутого способа см. в [73–75]. В данной работе приводится это описание в краткой форме.

Функции стоимости "большой" задачи допускают зависимость от (полного) списка заданий. Устраивая варьирование исходного решения (маршрута и

трассы) в пределах вставки, с целью улучшения качества данной вставки, мы, строго говоря, влияем на "совокупные" списки заданий. Поэтому нужна конструкция, при которой оптимизирующая вставка (и, тем более, мультвставка) была бы улучшающей не только локально, в пределах фрагмента, но и глобально, т.е. в смысле критерия "большой" задачи. Данное свойство удалось реализовать в конструкциях [74;75]. Упомянутые конструкции в краткой форме приводятся ниже; они играют важную роль в построении параллельного алгоритма на основе мультвставок. Данные построения касаются вариации как маршрутов (перестановка индексов), так и трасс, или траекторий; эти вариации, особенно в случае мультивставок, имеют достаточно специальный характер, что и отражено в последующем изложении; при этом требуется серьезная формализация используемых процедур.

Итак, при $\alpha \in \mathcal{A}$ и $\nu \in \overline{0,\mathbf{n}-N}$ полагаем $\Lambda_\nu[\alpha] \stackrel{\triangle}{=} (\alpha(\nu+s))_{s\in\overline{1,N}}$, получая инъективное отображение $\overline{1,N}$ в $\overline{1,\mathbf{n}}$, а также конечное множество – образ ("окно" на маршруте α)

$$\Gamma_\nu[\alpha] \stackrel{\triangle}{=} \Lambda_\nu[\alpha]^1(\overline{1,N}) \in \mathcal{P}'(\overline{1,\mathbf{n}}), \tag{2.4.12}$$

для которого, конечно, $\Lambda_\nu[\alpha] \in (\mathrm{bi})[\Gamma_\nu[\alpha]]$. Подчеркнём, что $\overline{1,N}$ является интервалом индексов, используемых в пределах вставки. Для такого индексного множества будут введены свои (локальные) условия предшествования, касающиеся маршрутов определяемых в виде перестановок $\overline{1,N}$. Данные локальные маршруты будут, однако, вклеиваться в глобальные; отображение $\Lambda_\nu[\alpha]$ будет при этом вклеивании играть важную роль. С (2.4.12) связываем "окно" условий предшествования: при $\alpha \in \mathcal{A}$ и $\nu \in \overline{0,\mathbf{n}-N}$ множество

$$Q_\nu[\alpha] \stackrel{\triangle}{=} \{z \in \mathcal{K} | (\mathrm{pr}_1(z) \in \Gamma_\nu[\alpha]) \& (\mathrm{pr}_2(z) \in \Gamma_\nu[\alpha])\} \in \mathcal{P}(\mathcal{K}) \tag{2.4.13}$$

индексных УП "большой" задачи порождает локализацию (адресных пар)

$$\mathbf{K}_\nu[\alpha] \stackrel{\triangle}{=} \{(\Lambda_\nu[\alpha]^{-1}(\mathrm{pr}_1(z)), \Lambda_\nu[\alpha]^{-1}(\mathrm{pr}_2(z))) : z \in Q_\nu[\alpha]\} \in \mathcal{P}(\overline{1,N} \times \overline{1,N}) \tag{2.4.14}$$

(случай $\mathbf{K}_\nu[\alpha] = \varnothing$ не исключается) со свойством [74, с.127]

$$\forall \mathbf{K}_0 \in \mathcal{P}'(\mathbf{K}_\nu[\alpha])\ \exists z_0 \in \mathbf{K}_0 : \mathrm{pr}_1(z_0) \neq \mathrm{pr}_2(z)\ \forall z \in \mathbf{K}_0. \tag{2.4.15}$$

В виде (2.4.14) имеем новое множество адресных пар задачи маршрутизации, возникающей при построении вставки. Свойство (2.4.15) наследуется от (2.3.5) и доставляет факт существования допустимых по предшествованию (в новом толковании) маршрутов вставки: полагая далее $\mathbb{P} \stackrel{\triangle}{=} (\mathrm{bi})[\overline{1,N}]$, из (2.4.15) извлекается [54, часть 2] свойство

$$\mathbf{A}_\nu[\alpha] \stackrel{\triangle}{=} \{\beta \in \mathbb{P} | \beta^{-1}(\mathrm{pr}_1(z)) < \beta^{-1}(\mathrm{pr}_2(z))\ \ \forall z \in \mathbf{K}_\nu[\alpha]\} \neq \varnothing, \tag{2.4.16}$$

определяющее существование $\mathbf{K}_\nu[\alpha]$ - допустимых маршрутов вставки; итак, (2.4.16) — множество всех допустимых маршрутов вставки.

Вклеивание локального маршрута. Если $\alpha \in \mathcal{A}$, $\nu \in \overline{0,\mathbf{n}-N}$ и $\beta \in \mathbb{P}$, то склеенный маршрут (перестановка индексов) $(\beta - \mathrm{sew})[\alpha;\nu] \in \mathbf{P}$ [74, (3.9)] определяется условиями

$$\begin{gathered}((\beta - \mathrm{sew})[\alpha;\nu](t) \stackrel{\triangle}{=} \alpha(t)\ \ \forall t \in \overline{1,\mathbf{n}} \setminus \overline{\nu+1,\nu+N})\&((\beta - \mathrm{sew})[\alpha;\nu](t) \stackrel{\triangle}{=} \\ \stackrel{\triangle}{=} (\Lambda_\nu[\alpha] \circ \beta)(t-\nu)\ \ \forall t \in \overline{\nu+1,\nu+N}).\end{gathered} \tag{2.4.17}$$

Вклеивание (2.4.17) сохраняет [74, предложение 3.3] допустимость по предшествованию:

$$(\beta - \mathrm{sew})[\alpha;\nu] \in \mathcal{A}\ \ \forall \alpha \in \mathcal{A}\ \ \forall \nu \in \overline{0,\mathbf{n}-N}\ \ \forall \beta \in \mathbf{A}_\nu[\alpha]. \tag{2.4.18}$$

Пусть $\mathbf{e} \in \mathbb{P}$ — тождественная перестановка $\overline{1,N}$: $\mathbf{e}(s) \stackrel{\triangle}{=} s\ \ \forall s \in \overline{1,N}$. Ясно, что $\mathbf{e}$ при вклеивании по схеме (2.4.17) не изменяет [74, (3.10)] исходный маршрут:

$$(\mathbf{e} - \mathrm{sew})[\alpha;\nu] = \alpha\ \ \forall \alpha \in \mathcal{A}\ \ \forall \nu \in \overline{0,\mathbf{n}-N}. \tag{2.4.19}$$

Свойство (2.4.19) дополняется очевидной допустимостью $\mathbf{e}$:

$$\mathbf{e} \in \mathbf{A}_\nu[\alpha]\ \ \forall \alpha \in \mathcal{A}\ \ \forall \nu \in \overline{0,\mathbf{n}-N}. \tag{2.4.20}$$

В связи с (2.4.19) и (2.4.20) отметим построения [74; 75].

Приступая к определению локальных трасс, отметим естественную редукцию $\mathfrak{X}$: для этого сначала введем $\forall \alpha \in \mathcal{A}\ \ \forall \nu \in \overline{0,\mathbf{n}-N}\ \ \forall s \in \overline{1,N}$

$$(M_s[\alpha;\nu] \stackrel{\triangle}{=} \mathbf{L}_{\Lambda_\nu[\alpha](s)}) \& (\mathbb{M}_s[\alpha;\nu] \stackrel{\triangle}{=} \mathbb{L}_{\Lambda_\nu[\alpha](s)}). \tag{2.4.21}$$

Вышеупомянутая редукция множества $\mathfrak{X}$ сводится при $\alpha \in \mathcal{A}, \mathbf{z} \in \mathfrak{Z}_\alpha$ и $\nu \in \overline{0,\mathbf{n}-N}$ к замене $\mathfrak{X}$ множеством

$$\mathbb{X}[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \{\mathrm{pr}_2(\mathbf{z}(\nu))\} \cup \Big(\bigcup_{s=1}^{N} M_s[\alpha;\nu]\Big) \in \mathrm{Fin}(\mathfrak{X}), \tag{2.4.22}$$

где $\mathrm{pr}_2(\mathbf{z}(\nu))$ играет роль локальной базы (стартовой точки локальной задачи); полагаем также

$$\mathbb{Z}[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} (\mathbb{X}[\alpha;\mathbf{z};\nu] \times \mathbb{X}[\alpha;\mathbf{z};\nu])^{\overline{0,N}},$$

получая множество всех отображений из $\overline{0,N}$ в $\mathbb{X}[\alpha;\mathbf{z};\nu] \times \mathbb{X}[\alpha;\mathbf{z};\nu]$, т.е. множество всех кортежей $(z_i)_{i\in\overline{0,N}}$, где $z_j \in \mathbb{X}[\alpha;\mathbf{z};\nu] \times \mathbb{X}[\alpha;\mathbf{z};\nu]$ при $j \in \overline{0,N}$. Если $\alpha \in \mathcal{A}, z \in \mathfrak{Z}_\alpha$, $\nu \in \overline{0,\mathbf{n}-N}$ и $\beta \in \mathbb{P}$, то множество всех локальных трасс, согласованных с маршрутом β, имеет следующий вид:

$$\begin{aligned}\mathcal{Z}_\beta[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \{h \in \mathbb{Z}[\alpha;\mathbf{z};\nu] | (h(0) = (\mathrm{pr}_2(\mathbf{z}(\nu)), \mathrm{pr}_2(\mathbf{z}(\nu)))) \& (h(t) \in \\ \in \mathbb{M}_{\beta(t)}[\alpha;\nu]\ \ \forall t \in \overline{1,N})\} \in \mathrm{Fin}(\mathbb{Z}[\alpha;\mathbf{z};\nu]).\end{aligned} \tag{2.4.23}$$

Решение каждой локальной задачи является по сути иерархическим: выбор трассы подчинен выбору маршрута. Поэтому каждому маршруту (перестановке индексов из $\overline{1,N}$) сопоставляется пучок трасс (траекторий) вида (2.4.23). При $\alpha \in \mathcal{A}$, $\mathbf{z} \in \mathfrak{Z}_\alpha$ и $\nu \in \overline{0,\mathbf{n}-N}$ в виде

$$\widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \{(\beta,h) \in \mathbf{A}_\nu[\alpha] \times \mathbb{Z}[\alpha;\mathbf{z};\nu] | h \in \mathcal{Z}_\beta[\alpha;\mathbf{z};\nu]\} \in \mathrm{Fin}(\mathbf{A}_\nu[\alpha] \times \mathbb{Z}[\alpha;\mathbf{z};\nu]) \tag{2.4.24}$$

имеем множество всех ДР конструируемой ниже локальной задачи.

Всюду в дальнейшем $\mathfrak{N} \stackrel{\triangle}{=} \mathcal{P}'(\overline{1,N})$. Итак, $\mathfrak{N}$ есть семейство всех непустых п/м $\overline{1,N}$, т.е. семейство списков заданий. С каждым триплетом $(\alpha;\mathbf{z};\nu) \in \mathbf{D} \times$

$\overline{0,\mathbf{n}-N}$ связываем конкретный набор функций стоимости:

$$\begin{gathered}\mathbf{c}[\alpha;\mathbf{z};\nu]\in\mathcal{R}_+[\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathfrak{N}],c_1[\alpha;\mathbf{z};\nu]\in\mathcal{R}_+[\mathbb{X}[\alpha;\mathbf{z};\nu]\times\\ \times\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathfrak{N}],\ldots,c_N[\alpha;\mathbf{z};\nu]\in\mathcal{R}_+[\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathfrak{N}],\\ f[\alpha;\mathbf{z};\nu]\in\mathcal{R}_+[\mathbb{X}[\alpha;\mathbf{z};\nu]].\end{gathered}\tag{2.4.25}$$

Здесь функция $\mathbf{c}[\alpha;\mathbf{z};\nu]$ используется для оценивания (внешних) перемещений между мегаполисами, а также из стартовой точки (локальной базы) к мегаполисам. Функции $c_1[\alpha;\mathbf{z};\nu],c_2[\alpha;\mathbf{z};\nu],\ldots,c_N[\alpha;\mathbf{z};\nu]$ используются для оценивания (внутренних) работ, выполняемых при посещении мегаполисов. Наконец, функция $f[\alpha;\mathbf{z};\nu]$ оценивает терминальное состояние процесса в локальной задаче. Из (2.4.25) видно, что функции $\mathbf{c}[\alpha;\mathbf{z};\nu],c_1[\alpha;\mathbf{z};\nu],\ldots,c_N[\alpha;\mathbf{z};\nu]$ допускают зависимость от списка (невыполненных) заданий, что как раз типично для прикладной задачи о демонтаже радиационно опасных элементов.

Упомянутые функции определяем отдельно для случаев $\nu<\mathbf{n}-N$ и $\nu=\mathbf{n}-N$. Итак, если $(\alpha;\mathbf{z};\nu)\in\mathbf{D}\times\overline{0,\mathbf{n}-(N+1)}$ (а это объективно более сложный случай), то полагаем, что при $h\in\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathbb{X}[\alpha;\mathbf{z};\nu]$ и $K\in\mathfrak{N}$

$$\begin{gathered}(\mathbf{c}[\alpha;\mathbf{z};\nu](h,K)\triangleq\mathbf{c}^\natural(h,\Lambda_\nu[\alpha]^1(K)\cup\alpha^1(\overline{\nu+N+1,\mathbf{n}})))\&(c_j[\alpha;\mathbf{z};\nu](h,K)\triangleq\\ \triangleq c^\natural_{\Lambda_\nu[\alpha](j)}(h,\Lambda_\nu[\alpha]^1(K)\cup\alpha^1(\overline{\nu+N+1,\mathbf{n}}))\ \ \forall j\in\overline{1,N});\end{gathered}\tag{2.4.26}$$

кроме того, при $x\in\mathbb{X}[\alpha;\mathbf{z};\nu]$ полагаем, что

$$f[\alpha;\mathbf{z};\nu](x)\triangleq\mathbf{c}^\natural(x,\mathrm{pr}_1(\mathbf{z}(\nu+N+1)),\alpha^1(\overline{\nu+N+1,\mathbf{n}})),\tag{2.4.27}$$

получая оценку терминального состояния x.

Итак, посредством (2.4.26) и (2.4.27) конкретизированы функции (2.4.25) в случае $\nu<\mathbf{n}-N$, т.е. в случае, когда вставка не является финальной.

Если $(\alpha,\mathbf{z},\nu)\in\mathbf{D}\times\{\mathbf{n}-N\}$ (случай финальной вставки), то полагаем при $h\in\mathbb{X}[\alpha;\mathbf{z};\nu]\times\mathbb{X}[\alpha;\mathbf{z};\nu]$ и $K\in\mathfrak{N}$. что

$$\begin{gathered}(\mathbf{c}[\alpha;\mathbf{z};\nu](h,K)\triangleq\mathbf{c}^\natural(h,\Lambda_\nu[\alpha]^1(K)))\ \&\ (c_j[\alpha;\mathbf{z};\nu](h,K)\triangleq\\ \triangleq c^\natural_{\Lambda_\nu[\alpha](j)}(h,\Lambda_\nu[\alpha]^1(K))\ \ \forall j\in\overline{1,N});\end{gathered}\tag{2.4.28}$$

кроме того, полагаем при $x \in \mathbb{X}[\alpha;z;\nu]$, что

$$f[\alpha;\mathbf{z};\nu](x) \stackrel{\triangle}{=} f^{\natural}(x). \tag{2.4.29}$$

Тем самым конкретезированы функции стоимости для случая, когда терминальное состояние процесса, реализуемого в пределах вставки, является также и терминальным состоянием "большой" задачи. Посредством (2.4.28), (2.4.29) конкретизированы функции (2.4.25) для случая $\nu = \mathbf{n} - N$; в данном (более понятном) случае имеется особенность в определении этих функций, отмеченная в (2.4.28), (2.4.29).

Сейчас, располагая во всех возможных случаях функциями (2.4.25), введем аддитивные критерии для используемых в дальнейшем локальных задач: при $(\alpha;\mathbf{z};\nu) \in \mathbf{D} \times \overline{0,\mathbf{n}-N}$, $\beta \in \mathbf{A}_{\nu}[\alpha]$ и $h \in \mathcal{Z}_{\beta}[\alpha;\mathbf{z};\nu]$ полагаем в дальнейшем, что

$$\begin{aligned}\mathfrak{B}_{\beta}[h|\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \sum_{s=1}^{N}[\mathbf{c}[\alpha;z;\nu](\mathrm{pr}_2(h(s-1)),\mathrm{pr}_1(h(s)),\beta^1(\overline{s,N})) + \\ + c_{\beta(s)}[\alpha;z;\nu](h(s),\beta^1(\overline{s,N}))] + f[\alpha;z;\nu](\mathrm{pr}_2(h(N))).\end{aligned} \tag{2.4.30}$$

Теперь каждому триплету $(\alpha,\mathbf{z},\nu) \in \mathbf{D} \times \overline{0,\mathbf{n}-N}$, т.е. каждому ДР $(\alpha,\mathbf{z})$ большой задачи и "моменту" начала вставки, сопоставляем локальную задачу (задачу оптимизации процессов, реализуемых в пределах вставки)

$$\mathfrak{B}_{\beta}[h|\alpha;\mathbf{z};\nu] \to \min, (\beta,h) \in \widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu]. \tag{2.4.31}$$

С каждой локальной задачей (2.4.31) связываем значение (экстремум) и непустое множество ее оптимальных решений: если $(\alpha,\mathbf{z},\nu) \in \mathbf{D} \times \overline{0,\mathbf{n}-N}$, то

$$\mathbb{V}[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \min_{(\beta,h)\in\widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu]} \mathfrak{B}_{\beta}[h|\alpha;\mathbf{z};\nu] \in \mathbb{R}_{+} \tag{2.4.32}$$

(экстремум локальной задачи) и при этом

$$(\mathrm{SOL})[\alpha;\mathbf{z};\nu] \stackrel{\triangle}{=} \{(\beta_0,h_0) \in \widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu] | \mathfrak{B}_{\beta_0}[h_0|\alpha;\mathbf{z};\nu] = \mathbb{V}[\alpha;\mathbf{z};\nu]\} \in \mathrm{Fin}(\widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu]) \tag{2.4.33}$$

(множество всех оптимальных ДР задачи (2.4.31). Решение каждой локальной задачи связываем с определением (2.4.32) и какого — либо элемента множества (2.4.33). В этой связи отметим процедуры [75, §§3 – 6], использующие аппарат ДП.

Вклеивание локальных трасс. Рассмотрим процедуру вклеивания трасс или траекторий, реализуемых в пределах вставки, в исходном ДР "большой" задачи. Если $\alpha \in \mathcal{A}, \mathbf{z} \in \mathfrak{Z}_\alpha$, $\nu \in \overline{0,\mathbf{n}-N}$ и $h \in \mathbb{Z}[\alpha;\mathbf{z};\nu]$, то полагаем, что $(\mathrm{sew})[h|\alpha;\mathbf{z};\nu] \in \tilde{\mathfrak{Z}}$ определяется условиями

$$((\mathrm{sew})[h|\alpha;\mathbf{z};\nu](t) \stackrel{\triangle}{=} h(t-\nu) \ \ \forall t \in \overline{\nu+1,\nu+N})\&((\mathrm{sew})[h|\alpha;\mathbf{z};\nu](t) \stackrel{\triangle}{=} \stackrel{\triangle}{=} \mathbf{z}(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \overline{\nu+1,\nu+N}). \tag{2.4.34}$$

Разумеется, в (2.4.34), $(\alpha,\mathbf{z})$ есть ДР исходной задачи, ν - "момент" начала вставки и h - (по смыслу) трасса, реализуемая в пределах вставки. В частности, (2.4.34) применимо в случае, когда $\beta \in \mathbb{P}$ и $h \in \mathcal{Z}_\beta[\alpha;\mathbf{z};\nu]$, т.е. в случае вклеивания "настоящих" трасс.

Предложение 1. *См. [51, Предложение 1]. Если $\alpha \in \mathcal{A}$, $\mathbf{z} \in \mathfrak{Z}_\alpha$, $\nu \in \overline{0,\mathbf{n}-N}$, $\beta \in \mathbb{P}$ и $h \in \mathcal{Z}_\beta[\alpha;\mathbf{z};\nu]$, то в результате вклеивания на основе (2.4.34) реализуется трасса исходной задачи, согласованная со склеенным маршрутом:*

$$(\mathrm{sew})[h|\alpha;\mathbf{z};\nu] \in \mathfrak{Z}_{(\beta-\mathrm{sew})[\alpha;\nu]}. \tag{2.4.35}$$

Доказательство. Фиксируем $\alpha,\mathbf{z},\nu,\beta$ и h в соответствии с условиями предложения; полагаем для краткости $\mathbf{y} \stackrel{\triangle}{=} (\mathrm{sew})[h|\alpha;\mathbf{z};\nu]$. В силу (2.4.21), (2.4.23) и (2.4.34) при $t \in \overline{\nu+1,\nu+N}$

$$\mathbf{y}(t) = h(t-\nu) \in \mathbb{M}_{\beta(t-\nu)}[\alpha;\nu],$$

где $\mathbb{M}_{\beta(t-\nu)} = \mathbb{L}_{(\Lambda_\nu[\alpha]\circ\beta)(t-\nu)}$ и согласно (2.4.17) $(\Lambda_\nu[\alpha]\circ\beta)(t-\nu) = (\beta-\mathrm{sew})[\alpha;\nu](t)$, то есть

$$\mathbf{y}(t) \in \mathbb{L}_{(\beta-\mathrm{sew})[\alpha;\nu](t)}. \tag{2.4.36}$$

Если же $\tau \in \overline{1,\mathbf{n}} \setminus \overline{\nu+1,\nu+N}$, то согласно (2.3.6) и (2.4.34) $\mathbf{y}(\tau) = \mathbf{z}(\tau) \in \mathbb{L}_{\alpha(\tau)}$, где согласно (2.4.17) $\alpha(\tau) = (\beta - \mathrm{sew})[\alpha;\nu](\tau)$, а потому

$$\mathbf{y}(\tau) \in \mathbb{L}_{(\beta-\mathrm{sew})[\alpha;\nu](\tau)}. \tag{2.4.37}$$

Поскольку выбор t и τ был произвольным, установлено (см. (2.4.34),(2.4.36),(2.4.37)), что во всех возможных случаях

$$\mathbf{y}(s) \in \mathbb{L}_{(\beta-\mathrm{sew})[\alpha;\nu](s)} \quad \forall s \in \overline{1,\mathbf{n}}. \tag{2.4.38}$$

Наконец, из (2.3.6) и (2.4.34) получаем следующую цепочку равенств: $\mathbf{y}(0) = \mathbf{z}(0) = (\mathbf{x}_0,\mathbf{x}_0)$. Далее используя (2.3.6),(2.4.18) и (2.4.38), получаем, что $\mathbf{y} \in \mathfrak{Z}_{(\beta-\mathrm{sew})[\alpha;\nu]}$. С учетом определения $\mathbf{y}$ получаем (2.4.35), т.е. требуемое утверждение. □

Из (2.4.18) и предложения 1 вытекает (см. (2.3.7),(2.4.18)) следующие свойство допустимости получающейся после склеивания пары маршрут-трасса:

$$\begin{gathered}((\beta - \mathrm{sew})[\alpha;\nu],(\mathrm{sew})[h|\alpha;\mathbf{z};\nu]) \in \mathbf{D} \quad \forall \alpha \in \mathcal{A} \\ \forall \mathbf{z} \in \mathfrak{Z}_\alpha \quad \forall \nu \in \overline{0,\mathbf{n}-N} \quad \forall (\beta,h) \in \widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu].\end{gathered} \tag{2.4.39}$$

Итак, посредством (2.4.39) определена процедура вклеивания локального ДР в глобальное (в обоих случаях ДР определяется в виде пары маршрут–трасса). В частности (см. (2.4.33)), таким образом определено вклеивание локально оптимальных ДР в глобальные, и, вообще говоря, не оптимальные ДР "большой" задачи.

При $\alpha \in \mathcal{A}$, $\mathbf{z} \in \mathfrak{Z}_\alpha$ и $\nu \in \overline{0,\mathbf{n}-N}$ имеем [75, §§4,5] локальную трассу

$$(\mathrm{nar})[\mathbf{z}|\alpha;\nu] \in \mathcal{Z}_{\mathbf{e}}[\alpha;\mathbf{z};\nu], \tag{2.4.40}$$

являющуюся, по сути дела, фрагментом исходной варьируемой трассы, для которой (т.е. для трассы (2.4.40))

$$((\mathrm{nar})[\mathbf{z}|\alpha;\nu](0) \stackrel{\triangle}{=} (\mathrm{pr}_2(\mathbf{z}(\nu)),\mathrm{pr}_2(\mathbf{z}(\nu))))\&((\mathrm{nar})[\mathbf{z}|\alpha;\nu](t) \stackrel{\triangle}{=} \mathbf{z}(t+\nu) \quad \forall t \in \overline{1,N}). \tag{2.4.41}$$

Конечно, трасса (2.4.40),(2.4.41) является по сути дела сужением $\mathbf{z}$ на $\overline{\nu,\nu+N}$ с естественным сдвигом, отвечающим перемещению данного сужения во вставку. Из (2.4.20), (2.4.24) и (2.4.40) имеем очевидное свойство допустимости:

$$(\mathbf{e},(\mathrm{nar})[\mathbf{z}|\alpha;\nu])\in\widetilde{\mathbf{D}}[\alpha;\mathbf{z};\nu].$$

Поэтому при $\alpha\in\mathcal{A}$, $\mathbf{z}\in\mathfrak{Z}_\alpha$ и $\nu\in\overline{0,\mathbf{n}-N}$ определено следующие значение

$$\varkappa[\alpha;\mathbf{z};\nu]\stackrel{\triangle}{=}\mathfrak{B}_{\mathbf{e}}[(\mathrm{nar})[\mathbf{z}|\alpha;\nu]|\alpha;\mathbf{z};\nu]-\mathbb{V}[\alpha;\mathbf{z};\nu]\in\mathbb{R}_+. \tag{2.4.42}$$

С учётом (2.4.30) и (2.4.40) мы в виде (2.4.42) получаем оценку улучшения (а, точнее, неухудшения) фрагмента исходного ДР $(\alpha,\mathbf{z})$, реализуемого за счёт оптимизирующей процедуры вставки. Таким образом, с учётом (2.4.40), (2.4.41) получаем, что (2.4.42) определяет потенциальные возможности улучшения фрагмента исходного ДР за счёт применения однократной оптимизирующей вставки. При этом получающееся значение задачи (2.3.10) имеет (см. (2.3.9), (2.4.42)) следующий вид:

$$\begin{gathered}\hat{\mathfrak{G}}_{(\beta_0-\mathrm{sew})[\alpha;\nu]}[(\mathrm{sew})[h_0|\alpha;\mathbf{z};\nu]]=\hat{\mathfrak{G}}_\alpha[\mathbf{z}]-\varkappa[\alpha;\mathbf{z};\nu]\ \ \forall\alpha\in\mathcal{A}\ \ \forall\mathbf{z}\in\mathfrak{Z}_\alpha\\ \forall\nu\in\overline{0,\mathbf{n}-N}\ \ \forall(\beta_0,h_0)\in(\mathrm{SOL})[\alpha;\mathbf{z};\nu].\end{gathered} \tag{2.4.43}$$

Как следствие, получаем оценку экстремума "большой" задачи в терминах "улучшения" , доставляемого вставкой:

$$V\leqslant\hat{\mathfrak{G}}_\alpha[\mathbf{z}]-\varkappa[\alpha;\mathbf{z};\nu]\ \ \forall\alpha\in\mathcal{A}\ \ \forall\mathbf{z}\in\mathfrak{Z}_\alpha\ \ \forall\nu\in\overline{0,\mathbf{n}-N}. \tag{2.4.44}$$

Нашей целью является, в частности, распространение (2.4.44) на случай, когда используется несколько оптимизирующих вставок. Поскольку рассматривается задача, осложнённая ограничениями и возможной зависимостью функций стоимости от списка заданий, необходимы дополнительные конструкции, связанные со вставками, которые изложены в [51, §5]. Заметим, что последовательное применение оптимизирующих вставок в режиме итераций позволяет ощутимым образом улучшить оценки типа (2.4.44); см. в этой связи [73; 86]. В настоящей работе "усиление" оценок типа (2.4.44) связывается с "одновременным" применением системы вставок, каждая из которых подобна рассмотренной ранее. В

итоге реализуется мультивставка, использование которой обсуждается в следующем разделе.

2.5 Мультивставка

Рассмотрим конструкцию, направленную на улучшение эвристик посредством применения мультивставки, компонентами которой являются оптимизирующие вставки в смысле [73–75; 83; 86]. Данная конструкция ориентирована на применение параллельных алгоритмов с реализацией на МВС и многоядерных ПЭВМ. В принципе, на основе перестраиваемых мультивставок также могут быть построены итерационные процедуры.

Всюду в дальнейшем фиксируем ДР исходной "большой" задачи:

$$(\lambda, \mathbf{h}) \in \mathbf{D}; \tag{2.5.45}$$

тогда $\lambda \in \mathcal{A}$ и $\mathbf{h} \in \mathfrak{Z}_\lambda$. Кроме того, фиксируем $m \in \mathbb{N}$, $m \geqslant 2$, и кортеж индексов

$$(\nu_i)_{i \in \overline{1,m}} : \overline{1,m} \to \overline{0,\mathbf{n}}; \tag{2.5.46}$$

здесь m - количество индивидуальных вставок, а $\nu_1, \ldots, \nu_m$ определяют всякий раз начало каждой такой вставки. Полагаем, что $\nu_m \leqslant \mathbf{n} - N$, а индексы в (2.5.46) упорядочены и, более того,

$$\nu_j + N < \nu_{j+1} \quad \forall j \in \overline{1,m-1}; \tag{2.5.47}$$

в дальнейшем, дизъюнктные "промежутки" $\overline{\nu_1 + 1,\nu_1 + N}, \ldots, \overline{\nu_m + 1,\nu_m + N}$ натурального ряда отвечают применению однократных вставок, подобных рассматриваемой в предыдущем разделе; совокупность этих вставок рассматривается в качестве мультивставки. Заметим, что из (2.5.47) вытекает очевидная цепочка неравенств

$$0 \leqslant \nu_1 < \ldots < \nu_m \leqslant \mathbf{n} - N. \tag{2.5.48}$$

При этом, конечно, $\nu_j \in \overline{0,\mathbf{n}-N} \quad \forall j \in \overline{1,m}$. Итак, мы можем использовать построения предыдущих разделов применительно к каждой отдельно взятой вставке, определяемой для ДР (2.5.45). С учётом этого полагаем сейчас, что найдена конечная система оптимальных решений (в виде пары маршрут-трасса) для индивидуальных "однократных" вставок

$$((\rho_j, u_j))_{j\in\overline{1,m}} \in \prod_{j=1}^{m} (\mathrm{SOL})[\lambda; \mathbf{h}; \nu_j], \tag{2.5.49}$$

то есть имеет свойства $(\rho_1, u_1) \in (\mathrm{SOL})[\lambda; \mathbf{h}; \nu_1], \ldots, (\rho_m, u_m) \in (\mathrm{SOL})[\lambda; \mathbf{h}; \nu_m]$. Конкретное построение данных УП — оптимальных решений локальных задач — может быть реализовано по схеме на основе ДП; см. [75, §3]. Впрочем, для некоторых наших целей достаточным будет лишь сам факт существования кортежа (2.5.49) (см. в этой связи алгоритм 1^0 работы [75]). При этом, в частности, имеем систему включений

$$(\rho_j, u_j) \in \widetilde{\mathbf{D}}[\lambda; \mathbf{h}; \nu_j] \quad \forall j \in \overline{1,m}. \tag{2.5.50}$$

Следовательно (см. (2.4.24), (2.5.49)), при $j \in \overline{1,m}$ реализуется $\rho_j \in \mathbf{A}_{\nu_j}[\lambda]$ и $u_j \in \mathcal{Z}_{\rho_j}[\lambda; \mathbf{h}; \nu_j]$, причём (всякий раз) имеет место следующее равенство:

$$\mathfrak{B}_{\rho_j}[u_j|\lambda; \mathbf{h}; \nu_j] = \mathbb{V}[\lambda; \mathbf{h}; \nu_j].$$

В терминах кортежа (2.5.49) конструируем далее два специальных "протяженных" кортежа, один из которых (кортеж индексов) будет допустимым маршрутом "большой" задачи, а второй – траекторией, согласованной с упомянутым маршрутом; подробнее см. [51]. Итак, пусть сначала кортеж$\eta : \overline{1,\mathbf{n}} \to \overline{1,\mathbf{n}}$ определяется условиями

$$\Big(\eta(t) \stackrel{\triangle}{=} \lambda(t) \quad \forall t \in \overline{1,\mathbf{n}} \setminus \Big(\bigcup_{j=1}^{m} \overline{\nu_j + 1, \nu_j + N}\Big)\Big) \& (\eta(t) \stackrel{\triangle}{=} (\Lambda_{\nu_s}[\lambda] \circ \rho_s)(t - \nu_s)$$
$$\forall s \in \overline{1,m} \quad \forall t \in \overline{\nu_s + 1, \nu_s + N}) \tag{2.5.51}$$

Для установления требуемых свойств кортежа (2.5.51) введем также систему вспомогательных кортежей. Именно, по аналогии с (2.5.51) введем при $k \in \overline{1,m}$

кортеж $\eta_k : \overline{1,\mathbf{n}} \to \overline{1,\mathbf{n}}$ по следующему правилу

$$\Big(\eta_k(t) \stackrel{\triangle}{=} \lambda(t) \;\; \forall t \in \overline{1,\mathbf{n}} \setminus \Big(\bigcup_{j=1}^{k} \overline{\nu_j + 1,\nu_j + N}\Big)\Big)\&(\eta_k(t) \stackrel{\triangle}{=} (\Lambda_{\nu_s}[\lambda] \circ \rho_s)(t - \nu_s)$$
$$\forall s \in \overline{1,k} \;\; \forall t \in \overline{\nu_s + 1,\nu_s + N}). \tag{2.5.52}$$

Итак, мы варьируем здесь исходный маршрут, используя не все, вообще говоря, индивидуальные вставки. Посредством (2.5.52) определены вспомогательные кортежи $\eta_1, \ldots, \eta_m$, причем (см. (2.4.17), (2.5.51), (2.5.52)) крайние кортежи определяются условиями

$$(\eta_1 = (\rho - \mathrm{sew})[\lambda; \nu_1])\&(\eta_m = \eta). \tag{2.5.53}$$

Последующее построение, использующее индукцию, характеризует переход от η_1 к η_2. Следующее предложение характеризует кортежи $\eta_1, \ldots, \eta_m$ как допустимые маршруты в "большой" задаче, а именно: справедливо

Предложение 2. *Каждый из кортежей, определяемых в* (2.5.52), *является* $\mathfrak{K}$*–допустимым маршрутом в исходной задаче:* $\eta_k \in \mathcal{A} \;\; \forall k \in \overline{1,m}$.

Доказательство. Пусть $\Omega \stackrel{\triangle}{=} \{k \in \overline{1,m} | \eta_k \in \mathcal{A}\}$. Тогда $1 \in \Omega$ в силу (2.4.18) и (2.5.53). Покажем, что $\Omega = \overline{1,m}$. Допустим противное: $\Omega \neq \overline{1,m}$, а тогда $\overline{1,m} \setminus \Omega \neq \varnothing$ и наименьший элемент множества в нем содержится:

$$\theta \stackrel{\triangle}{=} \inf(\overline{1,m} \setminus \Omega) \in \overline{1,m} \setminus \Omega;$$

в частности, $\theta \neq 1$ (см (2.4.18),(2.5.53)), а потому $\theta \in \overline{2,m}$ и $\theta - 1 \in \overline{1,m-1}$, причём $\eta_{\theta-1} \in \mathcal{A}$ по определению множества Ω. Легко видеть, (см. (2.5.52)), что $\rho_\theta \in \mathbf{A}_{\nu_\theta}[\eta_{\theta-1}]$ и

$$\eta_\theta = (\rho_\theta - \mathrm{sew})[\eta_{\theta-1}; \nu_\theta],$$

а потому согласно (2.4.18) $\eta_\theta \in \mathcal{A}$ вопреки предположению. □

Предложение 2 дополняется рекуррентной процедурой $\eta_1 \to \eta_2 \to \ldots \to (\eta_m = \eta)$, которая в существенной части определяется следующим положением.

Следствие 1. *Если* $k \in \overline{1,m-1}$, *то* $\eta_{k+1} = (\rho_{k+1} - \mathrm{sew})[\eta_k; \nu_{k+1}]$.

Доказательство. Вышеупомянутая рекуррентная процедура сводится к достаточно понятной комбинации (2.5.52) и следствия 1. С учётом (2.5.53), предложения 2 и следствия 1 получаем,в частности, что

$$\eta = (\rho_m - \mathrm{sew})[\eta_{m-1}; \nu_m] \in \mathcal{A}. \tag{2.5.54}$$

Итак, посредством (2.5.51) определяется некоторый $\mathfrak{K}$–допустимый (допустимый по предшествованию) маршрут "большой" задачи (имеется в виду первый из двух "протяженных" кортежей).

Рассмотрим теперь построение второго "протяженного" кортежа, который, как устанавливается ниже, будет траекторией согласованной с маршрутом η. Отметим, что при $j \in \overline{1,m}$ имеем, в частности, что $u_j : \overline{0,N} \to \mathfrak{X} \times \mathfrak{X}$. С учетом этого полагаем, что $w : \overline{0,\mathbf{n}} \to \mathfrak{X} \times \mathfrak{X}$ есть такая функция, что

$$\begin{gathered}(w(t) \stackrel{\triangle}{=} \mathbf{h}(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \Big(\bigcup_{j=1}^{m} \overline{\nu_j + 1,\nu_j + N}\Big)\Big)\&(w(t) = u_s(t - \nu_s) \\ \forall s \in \overline{1,m} \ \ \forall t \in \overline{\nu_s + 1, \nu_s + N}).\end{gathered} \tag{2.5.55}$$

Кроме того, как и в случае построения η, введем вспомогательную систему отображений $w_1, \ldots, w_m$. Итак, если $p \in \overline{1,m}$, то

$$w_p : \overline{0,\mathbf{n}} \to \mathfrak{X} \times \mathfrak{X}$$

определяемся следующими условиями

$$\begin{gathered}\Big(w_p(t) \stackrel{\triangle}{=} \mathbf{h}(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \Big(\bigcup_{j=1}^{p} \overline{\nu_j + 1,\nu_j + N}\Big)\Big)\&(w_p(t) \stackrel{\triangle}{=} \\ \stackrel{\triangle}{=} u_s(t - \nu_s) \ \ \forall s \in \overline{1,p} \ \ \forall t \in \overline{\nu_s + 1, \nu_s + N}).\end{gathered} \tag{2.5.56}$$

Таким образом, для этих отображений мы допускаем варьирование исходной траектории $\mathbf{h}$, используя не все, вообще говоря, индивидуальные вставки. Легко видеть, что (см.(2.4.34)) справедливы свойства

$$(w_1 = (\mathrm{sew})[u_1|\lambda; \mathbf{h}; \nu_1])\&(w = w_m), \tag{2.5.57}$$

характеризующие крайние кортежи (траектории). Как следствие, имеем с учётом (2.5.53) и (2.5.57), что (см. предложение 1)

$$\eta_1 \in \mathcal{A} : w_1 \in \mathfrak{Z}_{\eta_1}; \tag{2.5.58}$$

тогда получаем ДР "большой" задачи $(\eta_1,w_1) \in \mathbf{D}$. На самом деле и другие кортежи-траектории согласованы с соответствующими вспомогательными маршрутами: справедливо

Предложение 3. *Если* $k \in \overline{1,m}$, *то* $w_k \in \mathfrak{Z}_{\eta_k}$.

Доказательство. Полагаем, что $\Omega \stackrel{\triangle}{=} \{k \in \overline{1,m} | w_k \in \mathfrak{Z}_{\eta_k}\}$. Тогда в силу (2.5.58) $1 \in \Omega$, а потому $\Omega \neq \varnothing$. Покажем что $\Omega = \overline{1,m}$. Допустим противное: $\Omega \neq \overline{1,m}$. Тогда множество $\overline{1,m} \setminus \Omega \neq \varnothing$ содержит наименьший элемент

$$\theta \stackrel{\triangle}{=} \inf(\overline{1,m} \setminus \Omega) \in \overline{1,m} \setminus \Omega,$$

в то время как $\theta - 1 \in \Omega$. Поэтому по определению Ω имеем свойства

$$\eta_{\theta-1} \in \mathcal{A} : w_{\theta-1} \in \mathfrak{Z}_{\eta_{\theta-1}}; \tag{2.5.59}$$

поскольку $\rho_\theta \in \mathbb{P}$, определен склеенный маршрут "большой" задачи $(\rho_\theta - \mathrm{sew})[\eta_{\theta-1};\nu_\theta] \in \mathbf{P}$ (см. раздел 2.3) и при этом согласно следствию 1 этот маршрут допустим по предшествованию:

$$\eta_\theta = (\rho_\theta - \mathrm{sew})[\eta_{\theta-1};\nu_\theta] \in \mathcal{A}.$$

С учетом (2.5.52) и (2.5.59) нетрудно показать, что $w_\theta(t) \in \mathbb{L}_{\eta_\theta(t)} \ \ \forall t \in \overline{1,\mathbf{n}}$. Кроме того, с учетом (2.5.56) получаем равенство $w_\theta(0) = (\mathbf{x}_0, \mathbf{x}_0)$. Согласно (2.3.6) $w_\theta \in \mathfrak{Z}_{\eta_\theta}$ и, как следствие, $\theta \in \Omega$, что невозможно. Требуемое противоречие получено.

Это означает, что $(\eta_k, w_k) \in \mathbf{D} \ \ \forall k \in \overline{1,m}$. Вместе с тем с учетом (2.5.49), (2.5.52) и предложения 2 получаем, что (см. [51]) справедливо пошаговое представление

$$(\rho_{k+1}, u_{k+1}) \in (\mathrm{SOL})[\eta_k; w_k; \nu_{k+1}] \ \ \forall k \in \overline{1,m-1} \tag{2.5.60}$$

((2.5.60) следует из определений; при проверке (2.5.60), однако, следует отдельно рассматривать при $k \in \overline{1,m-1}$ случаи $\nu_{k+1} < \mathbf{n} - N$ и $\nu_{k+1} = \mathbf{n} - N$). Отметим теперь, что в силу предложения 1, следствия 1 и (2.5.60) имеют место свойства

$$(\mathrm{sew})[u_{k+1}|\eta_k; w_k; \nu_{k+1}] \in \mathfrak{Z}_{\eta_{k+1}} \ \ \forall k \in \overline{1,m-1}. \tag{2.5.61}$$

При этом (см.(2.4.33),(2.4.42), (2.5.60)) определены значения $\varkappa[\eta_k; w_k; \nu_{k+1}] \in \mathbb{R}_+$ при $k \in \overline{1,m-1}$, характеризующие "улучшения", доставляемые индивидуальными вставками.

Предложение 4. *Если* $k \in \overline{1,m-1}$, *то справедливо равенство* $w_{k+1} = (\mathrm{sew})[u_{k+1}|\eta_k; w_k; \nu_{k+1}]$.

Доказательство. Фиксируем $k \in \overline{1,m-1}$ и рассматриваем склеенную трассу $\tilde{w}_k \stackrel{\triangle}{=} (\mathrm{sew})[u_{k+1}|\eta_k; w_k; \nu_{k+1}]$; $\tilde{w}_k \in \tilde{\mathfrak{Z}}$. Тогда (см. (2.4.34))

$$\begin{gathered}(\tilde{w}_k(t) = u_{k+1}(t-\nu_{k+1}) \ \ \forall t \in \overline{\nu_{k+1}+1,\nu_{k+1}+N}) \& (\tilde{w}_k(t) = \\ = w_k(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \overline{\nu_{k+1}+1,\nu_{k+1}+N}).\end{gathered} \tag{2.5.62}$$

С другой стороны, в силу (2.5.56) кортеж $w_{k+1} \in \tilde{\mathfrak{Z}}$ таков, что

$$\begin{gathered}\left(w_{k+1}(t) = \mathbf{h}(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \left(\bigcup_{j=1}^{k+1} \overline{\nu_j+1,\nu_j+N}\right)\right) \& (w_{k+1}(t) = \\ = u_s(t-\nu_s) \ \ \forall s \in \overline{1,k+1} \ \ \forall t \in \overline{\nu_s+1,\nu_s+N}).\end{gathered} \tag{2.5.63}$$

Наконец, "предваряющая" трасса $w_k \in \tilde{\mathfrak{Z}}$ имеет следующий вид

$$\begin{gathered}\left(w_k(t) = \mathbf{h}(t) \ \ \forall t \in \overline{0,\mathbf{n}} \setminus \left(\bigcup_{j=1}^{k} \overline{\nu_j+1,\nu_j+N}\right)\right) \& (w_k(t) = \\ = u_s(t-\nu_s) \ \ \forall s \in \overline{1,k} \ \ \forall t \in \overline{\nu_s+1,\nu_s+N}).\end{gathered} \tag{2.5.64}$$

Из (2.5.63) и (2.5.64) получаем, в частности, что

$$w_{k+1}(t) = \mathbf{h}(t) = w_k(t) \ \ \forall t \in \left(\overline{0,\mathbf{n}} \setminus \overline{\nu_{k+1}+1,\nu_{k+1}+N}\right) \setminus \left(\bigcup_{j=1}^{k} \overline{\nu_j+1,\nu_j+N}\right). \tag{2.5.65}$$

Здесь мы учитываем очевидное равенство $\bigcup\limits_{j=1}^{k+1}\overline{\nu_j+1,\nu_j+N}=\left(\bigcup\limits_{j=1}^{k}\overline{\nu_j+1,\nu_j+N}\right)\cup\overline{\nu_{k+1}+1,\nu_{k+1}+N}$. С другой стороны, из (2.5.63) и (2.5.64) следует, что

$$w_{k+1}(t)=w_k(t)\ \ \forall t\in\bigcup_{j=1}^{k}\overline{\nu_j+1,\nu_j+N}. \tag{2.5.66}$$

Поэтому (см.(2.5.65), (2.5.66)) имеем следующее представление

$$w_{k+1}(t)=w_k(t)\ \ \forall t\in\overline{0,\mathbf{n}}\setminus\overline{\nu_{k+1}+1,\nu_{k+1}+N}. \tag{2.5.67}$$

Из (2.5.62), (2.5.67) получаем теперь, что

$$\tilde{w}_k(t)=w_{k+1}(t)\ \ \forall t\in\overline{0,\mathbf{n}}\setminus\overline{\nu_{k+1}+1,\nu_{k+1}+N}. \tag{2.5.68}$$

Вместе с тем, согласно (2.5.62)–(2.5.64)

$$\tilde{w}_k(t)=u_{k+1}(t-\nu_{k+1})=w_{k+1}(t)\ \ \forall t\in\overline{\nu_{k+1}+1,\nu_{k+1}+N}. \tag{2.5.69}$$

Из (2.5.68) и (2.5.69) получаем требуемое равенство $\tilde{w}_k=w_{k+1}$ (имеется в виду равенство кортежей, определённых на $\overline{0,m}$).

Из предложения 6 получаем после простых преобразований (см.(2.4.43)), что реализуется следующее представление, отвечающее применению однократной вставки:

$$\hat{\mathfrak{G}}_{\eta_{k+1}}[w_{k+1}]=\hat{\mathfrak{G}}_{\eta_k}[w_k]-\varkappa[\eta_k;w_k;\nu_{k+1}]\ \ \forall k\in\overline{1,m-1}. \tag{2.5.70}$$

В то же время с учетом (2.5.53) и (2.5.57)

$$\mathfrak{G}_{\eta_1}[w_1]=\mathfrak{G}_{\lambda}[\mathbf{h}]-\varkappa[\lambda;\mathbf{h};\nu_1]. \tag{2.5.71}$$

Предложение 5. *Если* $k\in\overline{1,m-1}$, *то* $\varkappa[\eta_k;w_k;\nu_{k+1}]=\varkappa[\lambda;\mathbf{h};\nu_{k+1}]$.

Доказательство. Из (2.5.70) получаем, что

$$\hat{\mathfrak{G}}_{\eta_{k+1}}[w_{k+1}]=\hat{\mathfrak{G}}_{\eta_k}[w_k]-\varkappa[\lambda;\mathbf{h};\nu_{k+1}]\ \ \forall k\in\overline{1,m-1}.$$

Учитывая (2.5.53), (2.5.57) и (2.5.71), получаем с использованием индукции следующее положение:

Теорема 1. *Справедливо равенство*

$$\hat{\mathfrak{G}}_{\eta}[w] = \hat{\mathfrak{G}}_{\lambda}[\mathbf{h}] - \sum_{i=1}^{m} \varkappa[\lambda; \mathbf{h}; \nu_i].$$

Смысл теоремы состоит в том, чтобы улучшения, достигаемые в индивидуальных вставках, сделать независимыми, а общее улучшение качества свести к сумме частных. Из теоремы 1 вытекает очевидная оценка

$$V \leqslant \hat{\mathfrak{G}}_{\lambda}[\mathbf{h}] - \sum_{i=1}^{m} \varkappa[\lambda; \mathbf{h}; \nu_i], \tag{2.5.72}$$

подобная по смыслу (2.4.44). Для получения оценки (2.5.72) достаточно найти (см. (2.4.42)) значения

$$\mathbb{V}[\lambda; \mathbf{h}; \nu_1] \in \mathbb{R}_+, \ldots, \mathbb{V}[\lambda; \mathbf{h}; \nu_m] \in \mathbb{R}_+, \tag{2.5.73}$$

которые следует вычитать из величин, определяемых исходным ДР (2.4.44). Для определения упомянутых значений (2.5.73) (экстремумов локальных задач) можно использовать алгоритм 1° работы [75], что приводит к некоторой экономии ресурсов памяти соответствующего вычислителя. Такой подход может быть, в частности, использован в интересах оптимизации кортежа (2.5.46), что полезно на этапе поиска локализации мультивставки. На этапе построения склеенного ДР большой задачи, то есть УП (η, w), в каждой из "элементарных" вставок, составляющих мультивставку, следует уже использовать алгоритм 2° работы [75] для поиска кортежа (2.5.49). При этом обслуживание "элементарных" вставок может осуществляться независимо работающими процессорами (узлами). Итак, как видно из (2.5.51) и (2.5.55), для построения нового ДР (η, w) нам уже требуется оптимальные ДР для вставок (см. (2.5.49)), а построение последних по методу ДП требует сохранения в памяти вычислителя всех слоев функций Беллмана для вставок, что и подразумевает алгоритм 2° работы [75].

2.6 Вычислительный эксперимент

В данном разделе описывается практическая реализация решения маршрутной задачи, осложненной условиями предшествования и функциями стоимости, зависящими от списка невыполненных заданий. Рассматривается задача, моделирующая процесс демонтажа системы излучающих элементов; см. [80]. Для решения упомянутой задачи была реализована схема на основе вышеупомянутой конструкции с использованием мультивставки (см. раздел 2.5). Данный метод позволяет решать маршрутные задачи существенной размерности.

Далее приведен алгоритм решения задачи (2.3.10) с использованием оптимизирующей мультивставки.

1. На начальном этапе выполняется считывание исходных данных, таких как:
   - координаты городов мегаполисов и точки старта,
   - координаты излучающих элементов,
   - мощность излучающих элементов,
   - скорость перемещения работника,
   - условия предшествования (адресные пары).

   Затем выполняется расчет функций стоимости (2.3.8) по формулам, приведенным в [80]. Так как процесс вычисления этих функций является достаточно трудоемким при большом количестве мегаполисов, то для ускорения вычислений используется параллельный алгоритм на базе библиотеки OpenMP с общей памятью.
2. На данном этапе выполняется построение эвристического решения при помощи алгоритма [82, §6], который учитывает ограничения в виде условий предшествования и функции стоимости, зависящие от списка невыполненных заданий.
3. Далее выполняем улучшение найденного на предыдущем этапе маршрута при помощи мультивставки. Для этого выделяем из найденного ранее решения (точнее, из ДР "большой" задачи) фрагменты по k мегаполисов, при этом оставляя всякий раз между фрагментами "перемычку" – в виде одного мегаполиса. Каждый такой фрагмент передаем при помощи протокола MPI отдельному вычислительному узлу. Затем каж-

дый вычислительный узел, по отдельности и независимо, выполняет построение оптимального решения (в виде пары маршрут-трасса) для выделенного ему фрагмента. В качестве инструмента для вычисления оптимального маршрута используется ДП. Для ускорения вычислений используется параллельный алгоритм с общей памятью на базе библиотеки OpenMP. Так как изначально переданные фрагменты маршрута были просчитаны приближённо, после вычислений в типичном случае неоптимальной эвристики происходит улучшение результатов.

4. После того, как каждый вычислительный узел обработал свой фрагмент первоначального решения, он передает получившийся результат на главную управляющую машину, где происходит склеивание улучшенных фрагментов в одно большое решение, которое не является, вообще говоря, оптимальным, но является по смыслу "более близким" к таковому за счет оптимизирующих фрагментов, из которых оно состоит. Большая "близость" понимается в смысле улучшения качества.

Рассмотрим теперь модельный пример решения задачи маршрутизациии процесса демонтажа радиоактивного оборудования на плоскости; см. [80]. Пусть мегаполисы, имитирующие возможные входы/выходы помещений с источниками излучения, получены дискретизацией окружностей: на каждой окружности на равном угловом расстоянии, начиная с точки с 0-й угловой координатой, располагаются 30 точек. Каждому мегаполису соответствует точечный объект, имитирующий источник излучения в помещении. Пусть стартовая точка (она же база процесса демонтажа) совпадает с началом координат, т. е. $x^0 = (0{,}0)$. Полагаем, что скорость движения исполнителя, выполняющего демонтаж, вне помещений в 4 раза больше, чем внутри, что призвано моделировать сложность перемещения внутри каждого мегаполиса, обусловленную наличием тех или иных конструкций и механизмов, мешающих быстрому движению внутри соответствующего помещения.

Далее приведен результат расчета одной модельной задачи на супервычислителе УРАН. Для модельного примера использовался вариант с 255 мегаполисами и 45 адресными парами, соответствующими условиям предшествования, размер фрагмента вставки 25 мегаполисов. Были получены следующие результаты:

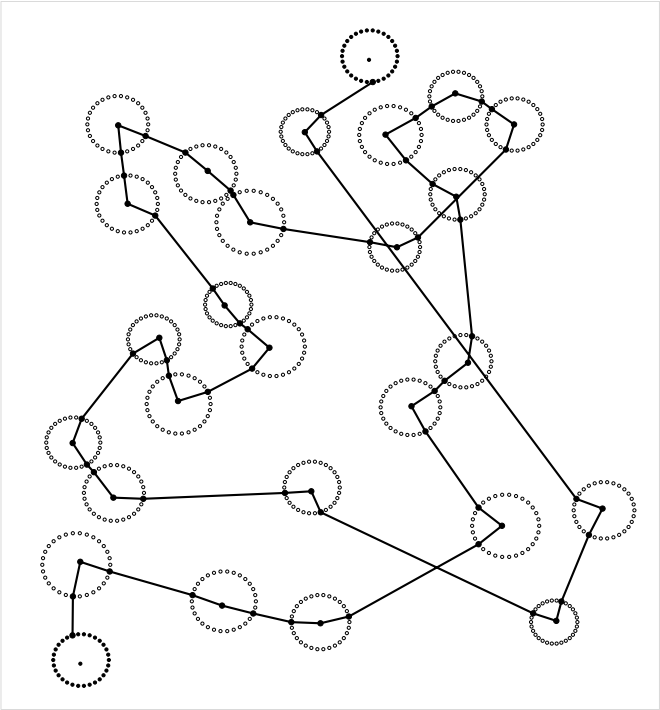

Рисунок 2.1 — Графическая иллюстрация одной элементарной вставки(до улучшения).

1. Суммарная величина дозы облучения, полученная при помощи эвристического (жадного) алгоритма составила – 3,61487.
2. Суммарная величина дозы облучения, полученная после применения мультиставок составила – 2,99925.
3. Степень улучшения результата составила 17,03%.
4. Время вычислений составило 22 мин. 35 сек.

Также был выполнен ряд экспериментов с целью определения зависимости улучшения значения экстремума V "большой" задачи от размера фрагментов мультивставки. Размер фрагмента варьировался от 15 до 25 мегаполисов. Для каждого размера фрагмента мультивставки было проведено 10 экспериментов, и было рассчитано среднее значение процента улучшения экстремума V "большой" задачи. На рисунке 2.3 показана зависимость улучшения от размера фрагмента мультивставки. Из него следует, что процент улучшения увеличивается с размером фрагмента мультивставки.

На основании вышеизложенного эксперимента можно сделать вывод, что чем больше размер фрагмента мультивставки, тем лучше выполняется корректировка значения экстремума V "большой" задачи. В связи с этим, для задач, в которых не нужно строить маршрут и трассу, целесообразным являет-

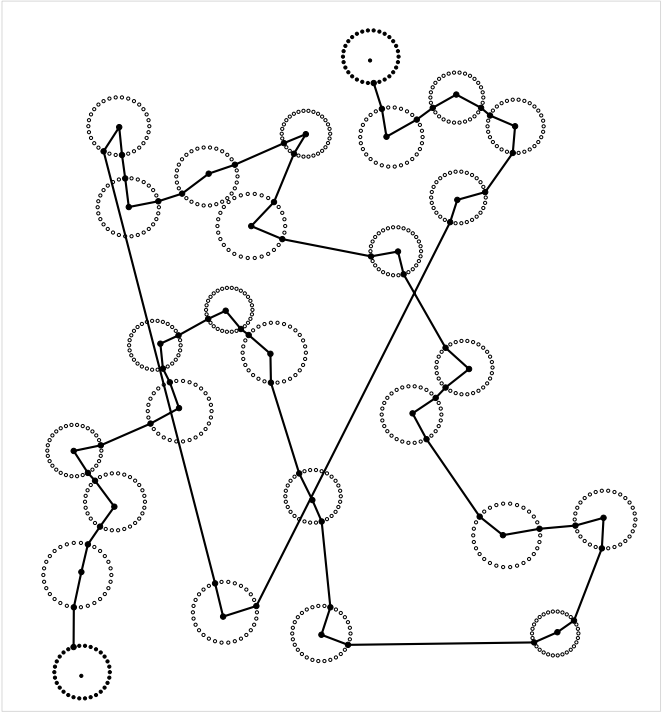

Рисунок 2.2 — Графическая иллюстрация действия одной элементарной вставки(после улучшения).

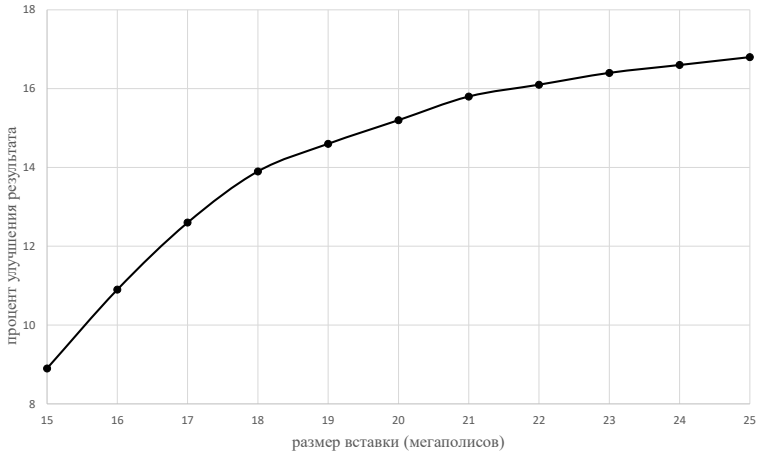

Рисунок 2.3 — Зависимость улучшения значения экстремума "большой" задачи от размера фрагмента мультивставки

ся использование алгоритма 1° работы [75]. Данный алгоритм предназначен для нахождения только значения экстремумов "локальных" задач, что позволяет осуществлять экономию ресурсов оперативной память соответствующих вычислительных узлов (требуется хранение только текущего и предыдущего слоя значений функции Беллмана). За счёт этого можно выполнять вычисления для размеров фрагментов мультивставки большего размера, что в свою очередь приводит к улучшению конечного результата. Для проверки данной гипотезы был выполнен ряд экспериментов. Было выполнено по 10 расчётов для двух случаев:

1. Вариант с 255 мегаполисами и 45 адресными парами, соответствующими условиям предшествования, размер фрагмента вставки 25 мегаполисов (размер мультивставки, который можно посчитать в случае необходимости построения маршрута и трассы).
2. Вариант с 255 мегаполисами и 45 адресными парами, размер фрагмента вставки 30 мегаполисов (размер мультивставки, который можно посчитать в случае определения только значение экстремума "локальной" задачи).

Для первого случая средний процент улучшения значения экстремума "большой" задачи составил 16,68%. Для второго случая соответствующее улучшение составило 17,85%. В задачах атомной энергетики, даже небольшое уменьшение получаемой персоналом дозы радиации, может принести существенную пользу. Поэтому, использование алгоритма 1° работы [75] является оправданным, в тех случаях, когда нам нужно знать только значение экстремума задачи.

Глава 3. Разные оптимизационные задачи

3.1 Введение

Современный высокотехнологичный мир предъявляет повышенные требования к безопасности окружающих нас сложных систем таких, например, как атомные электростанции и химическое производство. В случае непредвиденных ситуаций нередко слаженная и оперативная работа аварийно-спасательных формирований позволяет оперативно восстановить критические функции безопасности АЭС, существенно снизить вредные последствия и предотвратить возможную техногенную катастрофу. Оптимизация действий бригады работников в чрезвычайных ситуациях позволяет обеспечить скорейшее решение поставленной задачи при минимальном риске для здоровья спасателей и максимальной экономии средств.

Среди возможных вариантов снижения дозовых нагрузок облучения персонала АЭС, подлежащих оптимизации, несомненный интерес представляет оптимизация дозовых затрат, получаемых персоналом при перемещении от объекта к объекту, поскольку данный вопрос практически не рассматривался в отечественной и зарубежной практике. Снижение "транзитных доз" , полученных в пути на рабочее место и от рабочего места до выхода из зоны контролируемого доступа, представляет собой важную задачу в общем процессе оптимизации облучения ремонтного и обслуживающего персонала. Для сокращения доз, получаемых при перемещении, на ряде зарубежных станций используются подробные карты, которые можно получить при входе в реакторное здание и на различные отметки внутри здания. Однако, по имеющимся данным, на предприятиях атомной энергетики и промышленности не используется оптимизация пути перемещения работников с целью сокращения доз, получаемых при перемещении [55]. В предыдущих главах такая проблема рассматривалась и, как уже отмечалось, ее решение доведено до реализуемых алгоритмов. При выполнении ремонтных работ выбор маршрута перемещения с минимальной дозой облучения не представляет сложности, так как количество обслуживаемых объектов ограничено одним–двумя. При обслуживании значительного количества

объектов в радиационно–опасных зонах требуется использование эффективных вычислительных программ, так как число возможных маршрутов перемещения составляет N!, где N — количество посещаемых объектов. Среди вопросов такого рода возникает, в частности, необходимость в построении радиационных карт местности; такая задача рассматривается в следующем разделе. Также в данной главе описывается построение параллельного алгоритма решения задачи «на узкие места», связанное с поиском разбиения конечного множества заданий на конечное число исполнителей (работников). Показан алгоритм нахождения оптимального разбиения заданий с использованием метода динамического программирования с элементами параллельных вычислений при построении массива значений функции Беллмана.

3.2 Задача дозиметриста

Дозы облучения персонала при проведении радиационно-опасных работ снижают посредством уменьшения времени пребывания в радиационных полях, увеличения расстояния от источника излучения до человека, снижением мощности дозы на рабочих местах и в рабочих помещениях посредством экранирования, дезактивации и т.д. [56]. Эффективным способом сокращения дозовых затрат персонала (на 25-40 %), не требующим значительных материальных затрат, является оптимизация маршрутов выполнения работ в нестационарных радиационных полях [87]. Упомянутая оптимизация маршрута перемещений при проведении работ в нестационарных радиационных полях является эффективным способом сокращения дозовых затрат персонала [55]. Радиационная безопасность персонала считается обеспеченной, если соблюдаются основные принципы радиационной безопасности (обоснование, оптимизация, нормирование) и требования радиационной защиты, установленные Федеральным законом «О радиационной безопасности населения» № З - ФЗ от 09.01.96 (Собрание законодательства Российской Федерации, 1996, N 3, ст.141), НРБ-99/2009 и ОСПОРБ-99/2010 [88–90]. Принцип оптимизации предусматривает поддержание на возможно низком и достижимом уровне как индивидуальных (ниже пределов, установленных НРБ-99/2009), так и коллективных доз облучения

(доз, получаемых группой исполнителей), с учётом социальных и экономических факторов. Принцип оптимизации имеет важное практическое значение для обеспечения радиационной безопасности на всех этапах жизнедеятельности радиационно-опасного объекта. В условиях нормальной эксплуатации источника излучения или условий облучения оптимизация (совершенствование защиты) осуществляется при уровнях облучения в диапазоне от соответствующих пределов доз до достижения пренебрежимо малого уровня – 10 мкЗв в год индивидуальной дозы [91]. Индивидуальная доза является обобщенным показателем уровня безопасности установок, технологий, организации производства и культуры безопасности применительно к конкретным людям. Она зависит от мощности дозы внешнего гамма-нейтронного излучения, объемной активности радионуклидов в воздухе на рабочих местах и в рабочих помещениях и времени облучения. Коллективная эффективная доза определяется как сумма индивидуальных доз персонала (т.е. зависит как от величины индивидуальных доз, так и от численности персонала) и является ключевым параметром оптимизации защиты персонала.

Для оптимизации маршрута перемещения персонала при проведении работ в нестационарных полях строится рабочий план смены для каждого исполнителя. Для того, чтобы построить такой план, дозиметристу необходимо выполнить ряд предварительных работ:

1. Дозиметрист должен предварительно выполнить ряд замеров в помещении, в котором планируется выполнение работ бригадой исполнителей или одним исполнителем.
2. Выполняется построение радиационной карты помещения, в котором были выполнены предварительные измерения.
3. Назаначаются объекты, которые необходимо обслуживать (посещать) работникам АЭС.
4. На основе данных, полученных на 2-ом шаге, осуществляется расчёт функций стоимости перемещений между выделенными ранее объектами с учётом обхода возможных препятствий, а также терминальная функция.
5. Строится оптимальный маршрут посещения всех обозначенных объектов для исполнителя, с учётом возможных ограничений в виде условий предшествования.

После этих операций к работе приступает бригада, которая использует для перемещений маршрут, найденный дозиметристом (либо с помощью дозиметриста). Условимся в этой связи называть задачей дозиметриста всю систему 1-5.

3.2.1 Постановка задачи о выборе маршрута посещения заданных точек

Фиксируем число $N \in \mathbb{N}, N \geq 2$. Нашей целью является исследование перемещений следующего вида:

$$0 \to \alpha(1) \to \cdots \to \alpha(N) \tag{3.2.1}$$

где α – перестановка индексов из $\overline{1,N}$. Мы должны посетить каждый пункт ровно один раз и (в случае необходимости) вернуться на базу. В дальнейшем упомянутые перестановки, как и ранее, именуем маршрутами. Полагаем, что выбор α может быть стеснен дополнительными ограничениями, а именно: так называемыми условиями предшествования. Эти условия задаются посредством множества $\mathbf{K}$ адресных пар, см. (1.2). Элементами $\mathbf{K}$ являются упорядоченные пары; как и в первой главе, будем условно именовать первую компоненту упорядоченной пары отправителем, а вторую – получателем. Полагаем в дальнейшем, что справедливо условие (1.2.4). Тогда множество всех допустимых маршрутов можно записать следующим образом.

$$\mathbf{A} \stackrel{\triangle}{=} \{\alpha \in \mathbb{P} | \forall z \in \mathbf{K}\ \forall t_1 \in \overline{1,N}\ \forall t_2 \in \overline{1,N} (z = (\alpha(t_1),\alpha(t_2)) \Rightarrow (t_1 < t_2)\}$$

где $\mathbb{P} \stackrel{\triangle}{=} (\mathrm{bi})[\overline{1,N}]$. Данное условие соответствует главе 1.

Итак, нас удовлетворяют только такие маршруты, в которых для каждой адресной пары отправитель будет посещаться раньше получателя. Таким образом, имеем множество всех допустимых маршрутов в смысле условий предшествования, определяемых посредством множества $\mathbf{K}$. Итак, $\mathbf{A}$ определяет множество всех допустимых решений формулируемой ниже задачи. Теперь введем функции стоимости. Полагаем заданными две неотрицательные функции с областями определения $\overline{1,N} \times \overline{1,N}$ и $\overline{1,N}$ соответственно, имеются в виду функ-

ции, участвующие в формировании аддитивного критерия: $\mathbf{c}$ это функция (а, фактически, матрица) стоимостей перемещений, а f – терминальная функция, оценивающая, например, возврат на базу. Следовательно, даны функции

$$\mathbf{c} \in \mathcal{R}_+[\overline{1,N} \times \overline{1,N}], f \in \mathcal{R}_+[\overline{1,N}],$$

Тогда аддитивный критерий будет задаваться соотношением:

$$\mathfrak{G}_\alpha \stackrel{\triangle}{=} \mathbf{c}(0,\alpha(1)) + \sum_{t=1}^{N-1} \mathbf{c}(\alpha(t),\alpha(t+1)) + f(\alpha(N)). \tag{3.2.2}$$

В качестве основной рассматриваем здесь задачу минимизации аддитивного критерия (3.2.2), т.е. задачу

$$\mathfrak{G}_\alpha \to \min, \quad \alpha \in \mathbf{A}. \tag{3.2.3}$$

Поскольку $\mathbf{A}$ – непустое конечное множество, то задаче (3.2.3) сопоставляется значение (экстремум)

$$V \stackrel{\triangle}{=} \min_{\alpha \in \mathbf{A}} \mathfrak{G}_\alpha \tag{3.2.4}$$

и (непустое) множество оптимальных допустимых маршрутов:

$$\mathbf{A}_{opt} \stackrel{\triangle}{=} \{\alpha_0 \in \mathbf{A} |\ \mathfrak{G}_{\alpha_0} = V\} \in \mathcal{P}'(\mathbf{A}). \tag{3.2.5}$$

Нашей целью является определение значения экстремума V (3.2.4) и какого-либо элемента множества (3.2.5). Для решения задачи (3.2.3) будем использовать аппарат ДП, действуя в духе [54;92;93], а также главы 1 настоящей диссертации. Исследуемая задача (3.2.3) может рассматриваться как вариант TSP-PC; см [17;19;94–97]. В настоящей главе основное внимание уделяется прикладной задаче – задаче дозиметриста, который находится под воздействием радиации и, в этих условиях, производит комплекс необходимых измерений.

Задача (3.2.3) является математической моделью инженерной задачи дозиметриста. Для решения используется схема на основе ДП, восходящая к [7] в отличие от многих других работ по решению TSP-PC, где используется схема [8]. Кроме того, самостоятельный интерес представляет проблема построения функций $\mathbf{c}$ и f, отвечающих исходной прикладной задаче.

3.2.2 Построение радиационной карты помещения

В данном разделе описывается метод построения карты радиационного фона на плоскости по заранее измеренным значениям уровня радиации в ряде точек на этой плоскости. Упомянутые измерения выполнены дозиметристом. Итак, в рассматриваемой постановке имеется набор точек на плоскости, в которых были произведены замеры уровня радиации. По имеющимся данным необходимо произвести интерполяцию и построить такую "часть" плоскости, для которой известно значение уровня радиации уже в каждой ее точке; это требуется для дальнейшего нахождения функций стоимостей с учётом возможных препятствий. Поскольку в данной постановке невозможно выделить точечные источники радиации, построить функции стоимости подобно [80] не представляется возможным.

Для построения фрагмента плоскости с известными значениями радиации в каждой точке сетки будем использовать интерполяцию на основе радиальных базисных функций (РБФ) [98]. Радиальные базисные функции представляют собой набор методов жесткой интерполяции; это означает, что поверхность, определяемая значениями конструируемой функции, должна проходить через каждое измеренное опорное значение; РБФ часто используются для создания сглаженных поверхностей из большого количества расчётных данных. Также с их помощью можно создавать слабо изменяющиеся поверхности. Данный метод напоминает размещение резиновой мембраны между опорными точками и одновременно уменьшение общей кривизны поверхности. Выбор конкретной базисной функции определяет поведение этой мембраны между измеренными значениями.

В связи с тем, что получение данных в местах с повышенным значением радиационного поля сопряжено с риском для здоровья работников, в нашей постановке задачи имеется не так много полученных опорных значений. В связи с этим был выбран метод интерполяции РБФ [98], так как этот метод наиболее удобен для построения медленно меняющихся поверхностей при наличии достаточного количества опорных точек. Уменьшение же числа опорных точек приводит лишь к изменению формы изолиний, но общий характер поверхности изучаемого пространства (положение и интенсивность экстремумов и т.д.)

сохраняется или изменяется в незначительной степени. Далее полагаем, что $m \in \mathbb{N}$, хотя в рассматриваемой задаче достаточно полагать $m = 2$.

Схема решения на основе РБФ. Пусть задано некоторое множество опорных точек $x_1,x_2,\ldots,x_n \in \mathbb{R}^m$. Радиальная базисная функция – это такая функция вещественной прямой, значение которой зависит только от расстояния от опорной точки x до некоторой другой точки x_i. Через $\|\cdot\|$ обозначаем евклидову норму в $\mathbb{R}^m$. Другими словами, это функция удовлетворяющая следующему условию $\varphi(x,x_i) = \varphi_i(x) = \varphi(\|x - x_i\|), i = 1,\ldots,n$, рассматривается как радиальная базисная функция, а ее модуль $|\varphi(x,x_i)|$,будет определяться величиной евклидова расстояния $\|x - x_i\|$. Пусть определены значения $y_1,y_2,\ldots,y_n \in \mathbb{R}$, на соответствующем множестве опорных точек $x_1,x_2,\ldots,x_n \subset \mathbb{R}^m$. Итак, каждой точке x_i сопоставлено число $y_i; i \in \overline{1,n}$. В качестве интерполирующей функции выбираем линейную комбинацию радиальных базисных функций

$$F(x) = \sum_{i=1}^{n} \alpha_i \varphi_i(x) + \alpha_{n+1}. \tag{3.2.6}$$

Реализуя процедуру интерполяции с использованием (3.2.6), получаем систему из $n+1$ линейных уравнений с $n+1$ неизвестными коэффициентами α_i

$$\begin{cases} \sum\limits_{j=1}^{n} \alpha_j \varphi_j(x) + \alpha_{n+1} = y_i = F(x_i),\ i = 1,\ldots,n; \\ \sum\limits_{j=1}^{n} \alpha_j = 0 \end{cases}$$

где y_i – значение неизвестной интерполирующей функции в точке x_i при всяком $i \in \overline{1,n+1}$. Выделяют ряд наиболее часто используемых на практике видов РБФ.

1. Гауссиан (gaussian): $\varphi(r) = e^{-(\varepsilon r)^2}$.
2. Инверсный мультиквадрик (inverse multiquadric): $\varphi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}}$.
3. Мультиквадрик (multiquadric): $\varphi(r) = \sqrt{1 + (\varepsilon r)^2}$.
4. Плоский сплайн (thin-plate spline): $\varphi(r) = r^2 \ln r$.

В данной работе для построения радиационной карты помещения в качестве радиальной базисной функции был выбран плоский сплайн (см. случай 4), но вопрос выбора наиболее подходящей для интерполяции РБФ является предметом отдельного исследования.

3.2.3 Вычисление функций стоимости (на основе измерений и метода РБФ)

Для нахождения функций стоимостей, учитывающих возможность обхода препятствий, рассматривается применение метода Дейкстры для нахождения кратчайшего пути в графе (см. [99;100]). Для этого предполагается задание сетки на плоскости и построение связного графа, из которого удаляются все ребра и вершины, попадающие в область препятствий. Полагается, что радиационное поле неоднородно, поэтому ребра графа могут иметь существенно различный вес. Далее приведем подробный алгоритм нахождения функций стоимостей, учитывающих возможность обхода препятствий.

1. На первом этапе берем план местности с повышенным уровнем радиации или помещения АЭС. Отмечаем на нем объекты, являющиеся препятствиями, сквозь которые проходить нельзя. Также выделяем точки, которые необходимо посетить. Далее осуществляем построение (достаточно "густой") сетки на этом плане, которая будет полностью его покрывать. В тех местах, где сетка покрывает препятствия, узлы этой сетки отбрасываются. Шаг сетки зависит от точности, с которой мы хотим получить значения функции стоимости. Чем меньше шаг сетки, тем точнее результат мы получаем, но при этом увеличивается время счета.
2. Далее переходим к построению неориентированного графа. В качестве вершин графа выступают узлы построенной сетки, а в качестве ребер – грани.
3. Задача поиска кратчайшего пути на графе может быть определена для неориентированного, ориентированного или смешанного графа. В нашем случае рассматривается постановка задачи в самом простом виде для неориентированного графа. Для смешанного и ориентированного графа дополнительно должны учитываться направления ребер.

Граф представляет собой (см. [101–103]) совокупность непустого множества вершин и ребер (наборов пар вершин). Две вершины на графе смежны, если они соединяются общим ребром. Путь в неориентированном графе представляет собой последовательность вершин $P = (k_1,k_2,\ldots,k_{\mathbf{n}}) \in K \times K \times \ldots \times K$, таких

что k_i смежна с k_{i+1} для $1 \leq i \leq \mathbf{n}$, где $\mathbf{n}$ число вершин графа. Такой путь P называется путём длиной $\mathbf{n}$ из вершины k_1 в $k_{\mathbf{n}}$ (i указывает на номер вершины пути и не имеет никакого отношения к нумерации вершин на графе). Пусть $e_{i,j}$ — ребро соединяющее две вершины: k_i и k_j . Дана весовая функция $f : E \to \mathbf{R}$, которая отображает ребра на их веса, значения которых выражаются действительными числами, и неориентированный граф G. Тогда кратчайшим путём из вершины k в вершину k' будет называться путь $P = (k_1,k_2,\ldots,k_{\mathbf{n}})$ (где $k_1 = k$ и $k_{\mathbf{n}} = k'$), который имеет минимальное значение суммы $\sum\limits_{i=1}^{\mathbf{n}-1} f(e_{i,i+1})$.

Таким образом, находим стоимости перемещений (значения $\mathbf{c}$) между каждой парой точек , которые необходимо посетить исполнителю. После этого можем переходить к решению задачи (3.2.3). Для решения данной задачи используется метод ДП. Отличие данной (более простой) реализации ДП от той, которая описана в главе 1, в том, что мегаполисы представляются одноэлементным множеством и нет зависимости от списка невыполненных заданий.

Далее опишем кратко схему ДП для решения "упрощённой" задачи (3.2.3).

Как и в главе 1 данной диссертации, для использования ДП в задаче с возможными ограничениями в виде условий предшествования, определяемых множеством $\mathbf{K}$, осуществляется редукция соответствующих ограничений: допустимость по предшествованию заменяется допустимостью по вычеркиванию. С этой целью введем в рассмотрение оператор вычеркивания $\mathbf{I}$, действующий в семействе $\mathfrak{N}$ ($\mathfrak{N} \triangleq \mathcal{P}'(\overline{1,N})$ множество всех не пустых подмножеств $\overline{1,N}$) по следующему правилу. Данный оператор позволяет соблюдать условия предшествования в процессе рекуррентного построения функции Беллмана. Он нам понадобится для построения существенных списков.

$$\mathbf{I}(K) \stackrel{\triangle}{=} K \setminus \{\mathrm{pr}_2(z) : \ z \in \Xi(K)\}.$$

где $\Xi(K) \stackrel{\triangle}{=} \{z \in \mathbf{K} \,|\, (\mathrm{pr}_1(z) \in K)\,\&\,(\mathrm{pr}_2(z) \in K)\}$ при $K \in \mathfrak{N}$.

Далее напомним определение существенных списков. Речь идет о том, чтобы (при условиях предшествования) использовать только «часть» массива значений функции Беллмана, а потому при реализации ДП оказывается возможным ограничиться лишь этой "частью". Существенные списки удовлетворяет следующему условию: если в множестве присутствует отправитель, то в нем

обязательно должен присутствовать и получатель.

$$\mathbf{G} \triangleq \{K \in \mathfrak{N} \,|\, \forall z \in \mathbf{K}\ (\mathrm{pr}_1(z) \in K) \Rightarrow (\mathrm{pr}_2(z) \in K)\}.$$

Семейство $\mathbf{G}$ можно разбить по мощности $\mathbf{G}_s \triangleq \{K \in \mathbf{G} \,|\, s = |K|\}\ \forall s \in \overline{1,N}$. Тогда в виде $\mathbf{G}_1, \ldots, \mathbf{G}_N$ имеем разбиение $\mathbf{G}$ в конечную сумму семейств. Легко заметить, что $\mathbf{G}_N = \{\overline{1,N}\}$ (синглетон, содержащий множество $\overline{1,N}$). Кроме того $\mathbf{G}_1$ состоит из одноэлементных множеств, из которых исключены все отправители,

$$\mathbf{G}_1 \triangleq \{\{t\} :\ t \in \overline{1,N} \setminus \mathbf{K}_1\},$$

где $\mathbf{K}_1 \triangleq \{\mathrm{pr}_1(z) : z \in \mathbf{K}\}$. Слой $\mathbf{G}_{s-1}$ можно определить из слоя $\mathbf{G}_s$ по следующему правилу:

$$\mathbf{G}_{s-1} = \{K \setminus \{j\} :\ K \in \mathbf{G}_s,\, j \in \mathbf{I}(K)\}\ \forall s \in \overline{2,N}.$$

Вышеупомянутые свойства естественно связать с рекуррентной процедурой

$$\mathbf{G}_N \to \mathbf{G}_{N-1} \to \ldots \to \mathbf{G}_1,$$

где крайние слои определены явным образом. Далее определяем слои пространства позиций. Наиболее просто задаются крайние значения, они имеют следующий вид:

$$D_0 \triangleq \{(x, \varnothing) :\ s \in \overline{1,N} \setminus \mathbf{K}_1\}$$

$$D_N \triangleq \{(0, \overline{1,N})\}.$$

Для того, чтобы построить промежуточные слои пространства позиций, потребуются некоторые вспомогательные понятия, а именно, при $K \in \mathbf{G}_s$ последовательно определяем множества

$$J_s(K) \triangleq \{j \in \overline{1,N} \setminus K \,|\, \{j\} \cup K \in \mathbf{G}_{s+1}\}.$$

Это свойство позволяет при добавлении элемента к множеству K сохранять существенность списков. Таким образом, $\mathbb{D}_s[K]$ задаётся позициями, удовлетворяющих следующему свойству:

$$\mathbb{D}_s[K] \triangleq \{(j,K) :\ j \in J_s(K)\}.$$

Объединение таких списков по K из $\mathbf{G}_s$ даёт нам слой s пространства позиций

$$D_s \triangleq \bigcup_{K \in \mathbf{G}_s} \mathbb{D}_s[K].$$

Далее переходим к построению слоев функции Беллмана. Выделим следующее свойство. Это свойство позволяет связать между собой два соседних слоя пространства позиций, не нарушая при этом условия предшествования. Если $s \in \overline{1,N}$ и $(l,K) \in D_s$, то $(t,K \setminus \{t\}) \in D_{s-1}$ при $t \in \mathbf{I}(K)$. Теперь можем выписать рекуррентную формулу преобразования функции v_{s-1} в v_s

$$v_s(l,K) \triangleq \min_{t \in \mathbf{I}(K)} [\mathbf{c}(l,t,K) + v_{s-1}(t,K \setminus \{t\})] \quad \forall (l,K) \in D_s$$

Нулевой слой задан.

$$v_0(t,\varnothing) \triangleq f(t) \quad \forall t \in \overline{1,N} \setminus \mathbf{K}_1.$$

Таким образом, благодаря рекурсивной процедуре определяются последовательно все слои функции Беллмана:

$$v_0 \to v_1 \to \ldots \to v_N$$

На финальном этапе этой процедуры находим значение функции v_N. Из уравнения Беллмана следует, что это значение и является экстремумом нашей задачи.

$$V = v_N(x^0, \overline{1,N}).$$

Таким образом, экстремум задачи найден, далее мы можем строить оптимальный маршрут, следуя [84, раздел 4]. Построение оптимального маршрута осуществляется традиционным для ДП способом.

3.2.4 Вычислительный эксперимент

Вычислительные эксперименты проводились для реальных помещений Нововоронежской АЭС и Белоярской АЭС. Данные для опорных точек были

взяты из протоколов радиационного контроля, выполненных в вышеуказанных помещениях. Также были указаны точки на планах помещений, которые необходимо посещать для оперативных действий и для планового обслуживания оборудования. Необходимо на основе этих данных построить радиационную карту помещения и рассчитать оптимальный маршрут посещения указанных точек с учетом обхода возможных препятствий. На Рисунке 3.1 представлен результат вычислительного эксперимента для помещения Белоярской АЭС, размер помещения 8 на 10 метров. Белым обозначены области, сквозь которые проходить нельзя. Для представленного помещения было задано 10 опорных точек с измеренными ранее значениями радиации (значения указаны на рисунке, минимальное значение радиации 0,3 мкЗв/ч, максимальное – 1,1 мкЗв/ч). По рабочему плану необходимо посетить в помещении 12 точек, в которых необходимо выполнить какие-либо работы. Начальная и конечная точки были зафиксированы (предположим, что это были точки входа и выхода на объект с радиационной опасностью). Номером 1 обозначена точка входа, номером 14 – точка выхода. По опорным данным была построена радиационная карта помещения. Для визуализации радиационного фона использовалась цветная раскраска: от красного – наиболее высокого фона, до зеленого – наиболее низкого фона. Линией обозначен оптимальный маршрут бригады исполнителей, который позволяет минимизировать дозовую нагрузку на персонал при посещении необходимых пунктов. Скорость передвижения исполнителя считается постоянной и равна 1 км/ч. Время простоя в посещаемых пунктах не учитывается, так как оптимизируется маршрут, а доза радиации, получаемая при выполнении каких-либо работ в посещаемом пункте, не зависит от порядка посещения этих пунктов, и в сумме является величиной постоянной (источники излучения здесь "не выключаются").

В результате вычислений был получен маршрут: $i_1 = 1, i_2 = 2, i_3 = 13, i_4 = 10, i_5 = 11, i_6 = 7, i_7 = 12, i_8 = 9, i_9 = 3, i_{10} = 6, i_{11} = 4, i_{12} = 8, i_{13} = 5, i_{14} = 14$ (см. Рисунок 3.1). Были получены следующие результаты: суммарный путь, пройденный исполнителем, составляет 42,882 метра, доза полученной радиации за время прохождения найденного маршрута составляет V=0,00609 мкЗв (экстремум задачи). Время счёта составило 612 секунд.

На Рисунке 3.2 представлен результат работы программы для помещения Нововоронежской АЭС, размер помещения 5 на 17 метров. Белым обозначе-

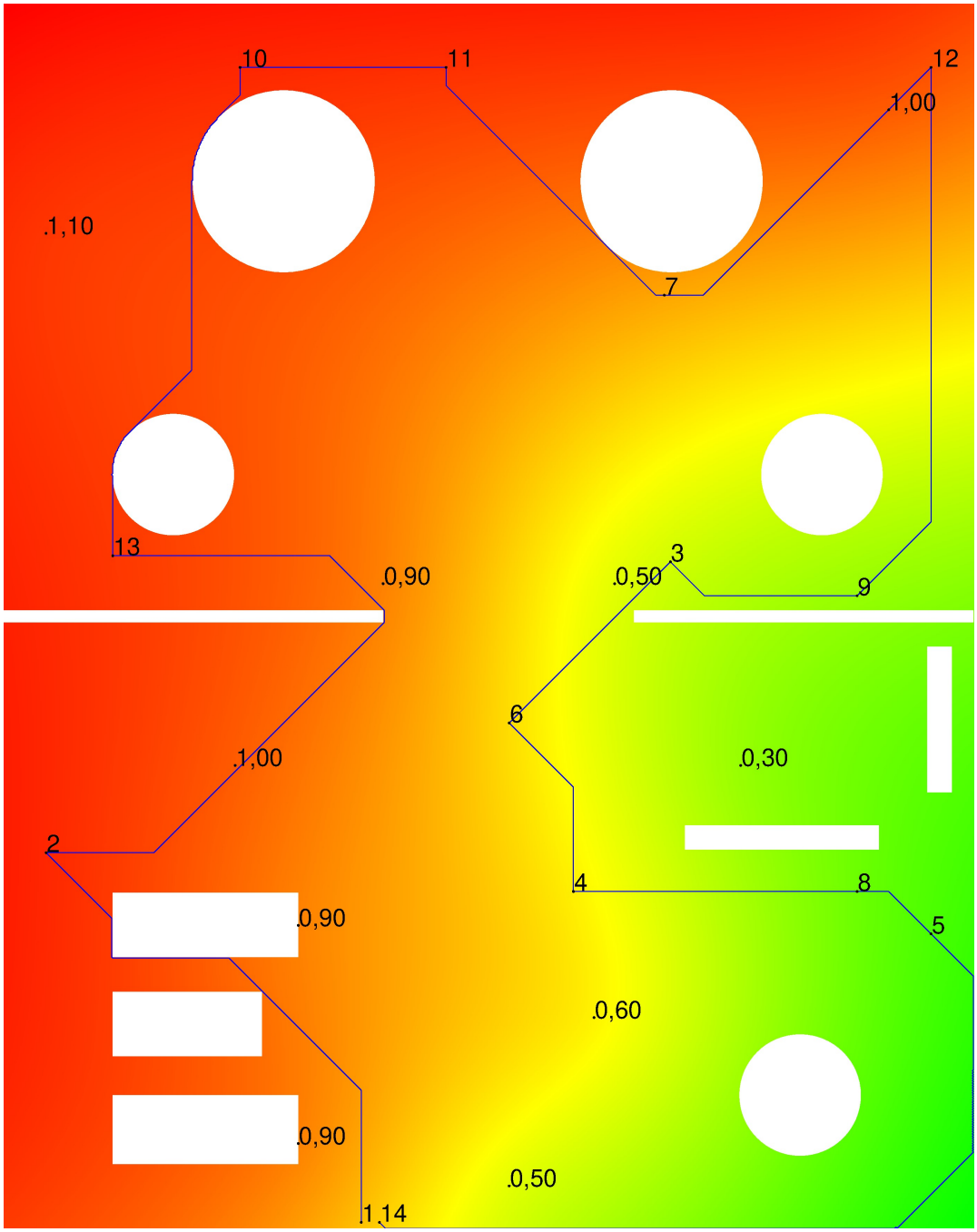

Рисунок 3.1 — Оптимальный маршрут, рассчитанный для помещения Белоярской АЭС.

ны объекты, сквозь которые проходить нельзя. Для данного помещения было задано 12 опорных точек с измеренными ранее значениями радиации (значения указаны на рисунке, минимальное значение радиации 2,8 мкЗв/ч, максимальное – 62 мкЗв/ч). По рабочему плану необходимо посетить 6 точек в помещении, в которых необходимо выполнить какие-либо работы. Начальная и конечная точки были зафиксированы (предположим, что это были точки входа и выхода на объект с радиационной опасностью). Номером 1 обозначена точка входа, номером 8 – точка выхода. Скорость передвижения исполнителя считается постоянной и равна 1 км/ч. Время простоя в посещаемых пунктах не учитывается, как и в случае вычислительного эксперимента для помещения Белоярской АЭС. В результате вычислений был получен маршрут: $i_1 = 1, i_2 = 2, i_3 = 7, i_4 = 6, i_5 = 5, i_6 = 4, i_7 = 3, i_8 = 8$ (см. Рисунок 3.2). Получены следующие результаты: суммарный путь, пройденный исполнителем, составляет 48,57 метра, доза полученной радиации за время прохождения найденного маршрута составляет V=0,66476 мкЗв (экстремум задачи). Время вычисления составило 523 секунды. Интересной особенностью этого случая является то, что в средней части помещения в ряд стоят цистерны с радиоактивной жидкостью на подставках, под ними наблюдается высокий радиоактивный фон. По краям помещения, радиоактивный фон значительно ниже, чем под баками. В результате вычислений мы видим, что исполнителю выгоднее, с точки зрения получаемого облучения, идти вдоль стен и подходить к точкам посещения не по кратчайшему пути. Таким образом, он проходит более длинный путь, но получает меньшую дозу радиации, чем если бы он шёл по наиболее короткому пути между этими точками.

Для сравнения результатов оптимального алгоритма, полученных для представленных помещений, были проведены расчеты для наиболее простой эвристики (жадный алгоритм), а также были сгенерированы 10 случайных маршрутов посещения всех необходимых пунктов. В результате эксперимента, жадный алгоритм выбирает маршрут в среднем на 46,85% хуже оптимального, выбор случайных маршрутов дает результат в среднем на 88,04% хуже оптимального. Это говорит о том, что если бы маршрут выбирался случайным образом, то доза маршрутной составляющей получаемой радиации работником, при выполнении необходимых работ, будет почти в 2 раза выше, чем для случая, когда работник передвигается по заранее рассчитанному оптимальному маршруту.

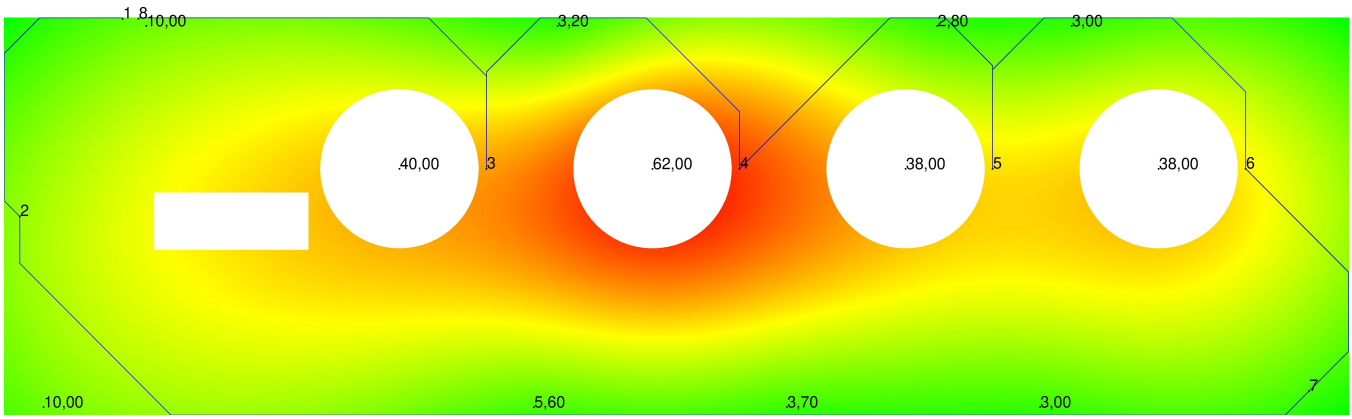

Рисунок 3.2 — Оптимальный маршрут, рассчитанный для помещения Нововоронежской АЭС.

3.3 Параллельная реализация динамического программирования в задачах об оптимальном распределении заданий

Рассматриваемая в данном разделе задача об оптимальном распределении заданий является составной частью маршрутно-распределительной задачи. В первой главе описано решение маршрутных задач. Совокупное же изучение маршрутно-расперределительной задачи позволяет решать такие сложные задачи, как оптимизация перемещений бригады исполнителей при демонтаже энергоблоков АЭС или устранение последствий радиационной аварии группами специалистов аварийно-спасательных формирований. При рассмотрении такой прикладной задачи требуется минимизировать наибольшее по всем исполнителям время нахождения в радиоактивной среде. Формально задачу оптимизации действий бригады, выполняющей работы в нестационарных радиационных полях, можно представить, как задачу распределения конечного числа заданий среди конечного числа исполнителей. В случае равенства числа исполнителей и заданий имеет место известная задача о назначениях (assignment problem) [104], успешно решаемая с помощью методов линейного программирования за полиномиальное время (см., например, венгерский алгоритм) [105]. Обобщенная постановка задачи о назначениях [106], в которой множество заданий и множество работников могут не совпадать по мощности, а также присутствует ограничение на ресурсы работников, уже не только NP – трудна, но и APX – трудна [107], то есть не допускает существования полиномиального приближенного алгорит-

ма, решающего задачу с заданной наперед точностью. Обе приведенные задачи рассматриваются обычно в двух постановках: на минимизацию совокупных затрат (максимизацию совокупного дохода) и на минимизацию наибольших по всем работникам затрат (максимизацию наименьшего дохода). Второй вариант (связываемый обычно с добавкой в названии проблемы “узкое горлышко” или “bottleneck”) рассматривается в печати несколько реже. Именно этот вариант рассматривается в настоящей работе, имея в виду, что бригаде исполнителей необходимо закончить работы в кратчайшие сроки как по причине скорейшей нейтрализации вредных последствий аварии, так и из соображений минимизации вредных воздействий на каждого из исполнителей. Далее мы считаем, что затрачиваемый исполнителями ресурс и получаемый доход — это один параметр. (Иными словами экономию мы и считаем выигрышем, будь то экономия времени, затрачиваемого на ликвидацию аварии, минимизация вреда для здоровья спасателей или уменьшение ущерба для техники.) В таких условиях рассмотренная выше обобщенная задача о назначениях теряет свою “рюкзачную” компоненту и становится близка к классической NP–полной задаче о разбиении (partition problem) в оптимизационной постановке, с тем лишь отличием, что в данном случае разбиение производится не на два, а на произвольное целое число (количество исполнителей) подмножеств. Итак, распределительная компонента формулируемой в настоящей работе оптимизационной задачи сочетает в себе элементы двух известных сложных проблем: обобщенной задачи о назначениях и задачи о разбиении.

Обычно, в задачах распределения заданий, рассматриваются аддитивные функции агрегирования затрат. В данной работе предполагаемая постановка допускает использование произвольных функций оценки “качества” распределения, имеющая место, например, в случае минимаксной маршрутно-распределительной задачи, где группе исполнителей требуется распределить между собой множество позиций таким образом, чтобы при посещении исполнителями соответствующих подмножеств длина максимального по всем исполнителям пути была минимальна.

В связи с тем, что рассматриваемая задача распределения является трудоемкой в вычислительном отношении (NP– трудной), появляется необходимость разработки параллельных алгоритмов для ее решения. В диссертации описыва-

ется параллельный алгоритм (см. [47]) нахождения оптимального разбиения N – заданий на n–участников (работников) при помощи метода ДП.

3.3.1 Формальная постановка задачи

Напомним, что через $\stackrel{\triangle}{=}$ обозначаем равенство по определению, $\mathbb{N} \stackrel{\triangle}{=} \{1; 2; ...\}$, $\mathbb{N}_0 \stackrel{\triangle}{=} \mathbb{N} \cup \{0\}$, $\mathbb{R}$ — вещественная прямая.

Задано число $N \in \mathbb{N}, N \geqslant 2$, индексы $i \in \overline{1,N}$ — определяют номера задач. Кроме того, задано число $n \in \mathbb{N}, n \geqslant 2$, индексы $j \in \overline{1,n}$ — определяют номера участников (работников).

Итак, предполагается, что у нас имеется N заданий и n работников (исполнителей). Требуется распределить все задания между работниками. При этом предполагается, что каждое задание выделяется ровно одному участнику.

Разбиением множества $\overline{1,N}$ на $n \in N$ подмножеств будем называть всякую совокупность (кортеж) n непустых подмножеств $(K_1, \ldots, K_n)$ множества $\overline{1,N}$ таких, что:

1. $\bigcup\limits_{i=1}^{n} K_i = \overline{1,N}$;
2. $\forall i \in \overline{1,n}\ \forall j \in \overline{1,n}\ (i \neq j) \Rightarrow (K_i \cap K_j = \varnothing)$.

Символ $\bigsqcup$ будет использоваться вместо $\bigcup$ в случае дизъюнктного объединения семейства множеств или их конечного набора.

Множество таких разбиений будем обозначать $M_n(\overline{1,N})$. Разбиение множества $K \subset \overline{1,N}$ на два подмножества K_1 и K_2 будем обозначать как $K = K_1 \bigsqcup K_2$. Пусть задана функция стоимости $\mathcal{D}$ выполнения конечных наборов заданий; $\mathcal{D} : \mathcal{P}(\overline{1,N}) \to \mathbb{R}$, где, как отмечалось ранее, $\mathcal{P}(X)$ — есть семейство всех подмножеств множества X.

Стоимостью распределения $\alpha = (K_1, \ldots, K_n) \in M_n(\overline{1,N})$ называется значение функции $D : M_n(\overline{1,N}) \to \mathbb{R}^+$, определяемое как $D(\alpha) = \max\limits_{i=\overline{1,n}} \mathcal{D}(K_i)$. Далее $\alpha^0 \in M_n(\overline{1,N})$ называется оптимальным на $M_n(\overline{1,N})$, если $\alpha^0 \in \operatorname{argmin}\limits_{\alpha \in M_n(\overline{1,N})} D(\alpha)$. В настоящей работе исследуется задача нахождения оптимального распределения множества N работ между n исполнителями, решаемая ниже

с помощью параллельных вычислений:

$$V = \max_{i=\overline{1,N}} D(K_i) \to \min, \bigsqcup_{i=1}^{n} K_i = \overline{1,N}. \tag{3.3.7}$$

Строгая постановка задачи (3.3.7) имеет вид:

$$V = \max_{i=\overline{1,n}} D(K_i) \to \min, (K_i)_{i\in\overline{1,n}} \in M_n(\overline{1,N}).$$

3.3.2 Метод динамического программирования

Предлагаемый ниже алгоритм содержит особенности, связанные с использование параллельной структуры для расчета на супервычислителе (см. раздел 3.3.3). Ниже используются конструкции [54; 92; 93].

Алгоритм 3.1:

1. Инициализация функции Беллмана (1-слой), для всякого подмножества $K \subset \overline{1,N}$ выполняем $V_1(K) := D(K)$. Имеется в виду "распределение" работ из K для случая одного исполнителя.

2. Последовательно увеличивая j от 2 до $n-1$, выполняем расчёт промежуточных значений функции Беллмана, опираясь на j-том шаге на результат вычислений, полученный на предыдущем шаге. Для всякого $K \subset \overline{1,N}$ (т.е. всякого $K \in \mathcal{P}(\overline{1,N}) : V_j(K) := \min\limits_{\widetilde{K}\subset K} \left\{\max\{D(\widetilde{K}), V_{i-1}(K \setminus \widetilde{K})\}\right\}$, где j – количество исполнителей.

3. Вычисляем последнее значение функции Беллмана, совпадающее со значением (экстремумом) задачи (3.3.7):

$$V_n(\overline{1,N}) := \min_{\widetilde{K}\subset\overline{1,N}} \left\{\max\{D(\widetilde{K}), V_{n-1}(\overline{1,N} \setminus \widetilde{K})\}\right\}.$$

Рассмотрим краткое доказательство корректности алгоритма (см. более подробно [54; 92; 93; 108]).

Утверждение 1. *Значение $V_m(K)$, вычисляемое в алгоритме, — есть экстремум задачи оптимального разбиения множества K среди m исполнителей.*

Д о к а з а т е л ь с т в о. Покажем по индукции на число исполнителей m, что

$$V_m(K) \leqslant \min_{\bigsqcup_{j=1}^{m} K_j = K} \left\{ \max_{j=\overline{1,m}} \{D(K_j)\} \right\}.$$

База индукции: $m = 1$. Очевидно, так как $V_1(K) = D(K)$, при $K \in \overline{1,N}$.

Шаг индукции: пусть утверждение доказано при $j < m$, покажем утверждение для $j = m$. Для любого распределения $\{L_1, \ldots, L_m\} : \bigsqcup_{j=1}^{m} L_j = K$ по определению

$$\begin{aligned} \max_{j=\overline{1,m}} D(L_j) &= \max\left\{ D(L_m), \max_{j=\overline{1,m-1}} \{L_1, \ldots, L_{m-1}\} \right\} \geqslant \\ &\geqslant \max\left\{ D(L_m), \left\{ V_{m-1}\left(\bigcup_{k=1}^{m-1} L_k \right) \right\} \right\} \geqslant \\ &\geqslant \min_{L \subseteq K} \{\max\{D(L), V_{m-1}(K \setminus L)\}\} = V_m(K). \end{aligned}$$

Очевидно, стоимость $V_m(K)$ достигается на разбиении $\{K_1^0, \ldots, K_m^0\}$, получающемся при движении "вверх" по алгоритму 3.1:

$$K_m^0 : V_m(K) = \max\{D(K_m^0), V_{m-1}(K \setminus K_m^0)\},$$

$$K_{m-1}^0 : V_{m-1}(K \setminus K_m^0) = \max\left\{ D(K_{m-1}^0), V_{m-2}\left((K \setminus K_m^0) \setminus K_{m-1}^0 \right) \right\}$$

и так далее, до K_1^0.

В итоге имеем $V_m(K) = \max\{D(K_m^0), D(K_{m-1}^0), \ldots, D(K_1^0)\} = \mathcal{D}((K_1^0, \ldots, K_m^0))$.

Общие вопросы теории динамического программирования рассматривались в [109]; см. также подробное изложение абстрактных вариантов метод ДП в [110].

Приведенный рекурсивный алгоритм обладает высокой вычислительной сложностью, поэтому помимо данной схемы точного решения, актуальны "быстрые" эвристические алгоритмы решения задачи распределения, которые были построены , например, в [111].

В следующем разделе рассматривается параллельный алгоритм построения точного решения этой задачи, реализованный на суперкомпьютере. В настоящей работе конструируется алгоритм, реализованный на суперкомпьютере,

что отличает ее от [54; 92; 93; 108], где использовались подобные алгоритмы для решения на ПК. Результаты настоящей работы анонсированы в [112].

3.3.3 Параллельная реализация алгоритма

В данном разделе рассматривается параллельная реализация алгоритма для решения задачи нахождения оптимального распределения.

В наших вычислениях участвует k процессоров.

– На первом шаге алгоритма корневой процессор рассылает начальные данные, всем участникам вычислительного процесса. Для этого используется функция MPI Bcast.

– На втором шаге алгоритма выполняется построение функции Беллмана

$$V_i(K) := \min_{\widetilde{K} \subset K} \left\{ \max\{D(\widetilde{K}), V_{i-1}(K \setminus \widetilde{K})\} \right\}, K \in \mathcal{P}(\overline{1,N})$$

На данном этапе для каждого слоя выполняется оптимальное распределение подмножеств $K \subset \overline{1,N}$ между j участниками. Распараллеливаем наиболее трудоемкий процесс перебора подмножеств $\overline{1,N}$. Для этого необходимо распределить подмножества $K \subset \overline{1,N}$ так , чтобы все процессоры были загружены равномерно. Каждый процессор, участвующий в вычислении, находит оптимальное разбиение любого из выделенных ему подмножеств. Подмножества $K, K \subset \overline{1,N}$, упорядочиваем по возрастанию их мощности: от одноэлементных до подмножества, содержащего все элементы множества $\overline{1,N}$, и тогда оно совпадает с $\overline{1,N}$. Если мы разделим число подмножеств на количество процессоров и распределим их поровну группами, упорядоченными по возрастанию индексов, то первому процессору достанутся подмножества малой мощности, в то время как последнему достанутся наиболее трудоемкие задания. В такой ситуации первые процессоры закончат работу раньше последних и будут вынуждены простаивать в ожидании обмена полученными результатами. Для того, чтобы избежать подобной ситуации, будем распределять подмножества на процессоры по “модулю k”.

– Проиндексируем все подмножества $K \subset \overline{1,N}$ в порядке возрастания мощности. Поскольку количество процессоров равно k, первый процессор получит подмножества с индексами $1, 1+k, 1+2k, \ldots$ второй процессор получит подмножества $2, 2+k, 2+2k, \ldots$ и так далее. В результате такой процедуры подмножества различной мощности относительно равномерно (по мощности) распределяются по процессорам.
– После того, как все процессоры выполнили вычисления на определенном слое, им необходимо обменяться полученными результатами для расчетов на следующем слое $V_i, i \in \overline{1,n}$. Для этого используется функция MPI Allgather.

3.3.4 Оценка вычислительной сложности

В настоящем разделе проводится оценка трудоемкости описанного выше алгоритма как в исходной, так и в параллельной реализации.

1. Оценка трудоемкости алгоритма без использования параллельной реализации.

– На шаге 1 алгоритма выполняется считывание начальных данных. Трудоемкостью данной процедуры можно пренебречь.
– На втором шаге алгоритма для каждого $j \in \overline{2,n-1}$ идет построение слоев функции Беллмана, при этом выполняется полный перебор всевозможных разбиений $K = K_1 \amalg K_2$ для каждого подмножества K множества $\overline{1,N}$, $V_j(K) := \min\limits_{K_1 \subseteq K} \{\max\{D(K_1), V_{j-1}(K_2)\}\}$. Полагаем $i \triangleq |K|$, где $i \in \overline{1,N}$. Количество подмножеств $K \subset \overline{1,N}$ таких, что $|K| = i$ равно $\dfrac{N!}{(N-i)! \cdot i!}$, количество вариантов выбора $\widetilde{K} \subset K$ операций по перебору данного подмножества равно 2^i, в результате для каждого $j \in \overline{1,n-1}$ получаем оценку числа операций j-ого слоя

$$\sum_{i=0}^{N} \left(\frac{N!}{(N-i)! \cdot i!} \cdot 2^i \right).$$

– Для $j = n$ выполняется перебор подмножеств $\overline{1,N}$, получаем оценку числа операций 2^N.
– Итоговая трудоемкость алгоритма без использования параллельной реализации получается путем суммирования трудоемкостей каждого слоя:

$$(n-1) \cdot \sum_{i=0}^{N} \left(\frac{N!}{(N-i)! \cdot i!} \cdot 2^i \right) + 2^N.$$

2. Оценка трудоемкости алгоритма с использованием параллельной реализации.

– На шаге 1 алгоритма выполняется считывание начальных данных и их пересылка процессорам, участвующим в вычислении. Трудоемкостью данной процедуры вновь пренебрегаем.
– На втором шаге алгоритма для $j \in \overline{1,n-1}$ трудоемкость рассчитывается аналогично трудоемкости алгоритма без использования параллельной реализации. Учитывая, что расчёт оптимального разбиения подмножеств выполняется на k процессорах, получаем оценку числа операций j-ого слоя в виде:

$$\frac{\sum_{i=0}^{N} \left(\frac{N!}{(N-i)! \cdot i!} \cdot 2^i \right)}{k}.$$

– Для $j = n$ выполняется перебор подмножеств $\overline{1,N}$; данная процедура рассчитывается на одном процессоре, а значит получаем оценку числа операций 2^N.
– Число операций по обмену полученных данных между процессорами рассчитывается следующим образом, В конце j-ого слоя каждый процессор должен отправить 2^N значений (V_j, $K \subset \overline{1,N}$) каждому из $k-1$ оставшихся процессоров. Учитывая, что такой обмен в конце слоя осуществляется одновременно всеми процессорами, а число слоев равно n, общее число операций будет $n \cdot 2^N$. Для приведения времен выполнения операций передачи данных и операций сравнения к единым единицам измерения необходимо ввести коэффициент M, $t_{\text{пер}} \approx M \cdot t_{\text{срав}}$, где, исходя из характеристик вычислителя, M может варьироваться в пределах $200 - 1000$. В нашем случае, исходя из вычислительного эксперимента, $M = 500$.

– Итоговая оценка числа операций сравнения алгоритма с использованием параллельной реализации может быть выражена как:

$$\frac{n-1}{k} \cdot \sum_{i=0}^{N} \left(\frac{N!}{(N-i)! \cdot i!} \cdot 2^i \right) + (\frac{n}{M} + 1) \cdot 2^N.$$

3.3.5 Вычислительный эксперимент

Рассматривается решение задачи нахождения оптимального разбиения N– заданий на n– участников (работников) методом динамического программирования с использованием супервычислителя “УРАН”. Также выполнен сравнительный анализ зависимости времени счета задачи от количества вычислительных ядер кластера.

Расчеты были произведены для $n = 3{,}4{,}5$ работников, количество заданий N варьируется от 20 до 25. Стоимость всех заданий формируется случайным образом в интервале от 1 до 100. Для сравнения времени вычисления для разного количества ядер супервычислителя берутся идентичные начальные данные.

На основании ранее описанного алгоритма была написана программа на языке программирования C^{++} с использованием библиотеки MPI [113]. Программа работает в среде 64–разрядной операционной системы семейства Linux. Вычислительный эксперимент проводился на супервычислителе “УРАН” с процессорной мощностью 1664 вычислительных ядра и 3584 Гбайт оперативной памяти. На данном супервычислителе используются узлы на основе двухпроцессорных блейд–серверов HP ProLiant BL460c CTO Blade со следующими характеристиками:

– два 4–х ядерных процессора Intel® Xeon® E5450 (3.0 GHz, 1333 FSB, 80W);
– кэш память 2×6 MB Level 2 cache (5400 Sequence);
– оперативная память 2 GB PC2–5300, Registered DDR2–667 на ядро;
– сетевой адаптер Dual 1GbE NC326i plus (1) additional 10/100 NIC dedicated to iLO 2;
– контроллер дисков Embedded SATA;

– жесткие диски HDD 90GB Non–Hot Plug SFF SATA.

Для анализа зависимости времени счета задачи от количества вычислительных ядер кластера был выполнен вычислительный эксперимент, в котором количество работников $n = 5$, количество заданий $N = 20$, количество ядер k варьируется от 1 до 256. Полученные данные представлены на рис. 3.3. Можно заметить, что при увеличении количества ядер до 32, идет существенное уменьшение времени счета. Далее наблюдается более плавное уменьшение времени, это объясняется тем, что при большом числе вычислительных ядер необходимо выполнять много операций по обмену данными между каждым процессором, так как эти данные являются большими по объёму, и на это затрачивается много времени. Это уменьшает выигрыш по времени счета задачи от добавления дополнительного количества вычислительных ядер.

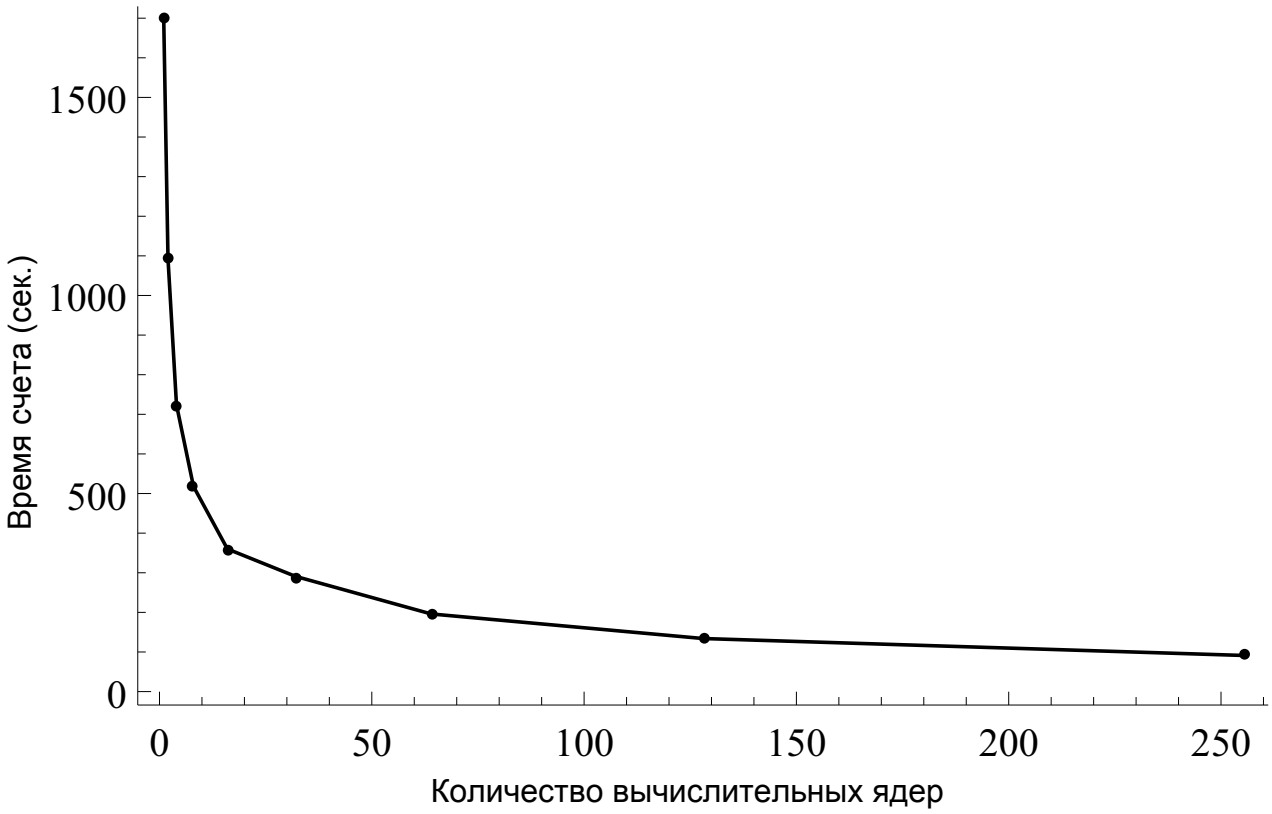

Рисунок 3.3 — Диаграмма зависимости времени счета задачи на супервычислителе от количества вычислительных ядер.

В качестве примера приводится результат одного из вычислительных экспериментов, выполненных на супервычислителе “УРАН”. В данном примере количество работников $n = 3$, количество заданий $N = 25$. Функция стоимости

$D(K)$, $K \subset \overline{1,N}$, получается суммированием стоимостей отдельных заданий. При этом стоимости выполнения упомянутых заданий равны:

$$d_1 = 36, d_2 = 68, d_3 = 52, d_4 = 28, d_5 = 99, d_6 = 79, d_7 = 39, d_8 = 40, d_9 = 5,$$

$$d_{10} = 24, d_{11} = 11, d_{12} = 39, d_{13} = 98, d_{14} = 74, d_{15} = 33, d_{16} = 35, d_{17} = 67,$$

$$d_{18} = 41, d_{19} = 84, d_{20} = 87, d_{21} = 52, d_{22} = 100, d_{23} = 97, d_{24} = 99, d_{26} = 36.$$

$$D(\oslash) \stackrel{\triangle}{=} 0, \;\; D(K) = \sum_{i \in K} d_i.$$

В результате выполнения программы были получены следующие результаты. Первый работник получил работы с индексами $K_1 = \{1,2,3,4,5,6,7,10,11,12\}$. Стоимость распределения равна $D(K_1) = 475$. Второй работник получил работы с индексами $K_2 = \{9,13,14,15,18,19,20,21\}$. Стоимость распределения равна $D(K_2) = 474$. Третий работник получил работы с индексами $K_3 = \{8,16,17,22,23,24,25\}$. Стоимость распределения равна $D(K_3) = 474$. Значение задачи V равно 475.

Время счета программы составило 3 часа 56 минут.

В результате проделанной работы был реализован алгоритм нахождения оптимального разбиения N– заданий на n–участников (работников) при помощи метода динамического программирования как с использованием, так и без использования параллельной структуры. Приведен вычислительный эксперимент для сравнения этих двух алгоритмов. Полученные результаты показали, что алгоритм с использованием параллельной структуры дает существенный выигрыш по времени вычисления для 32 вычислительных ядер. Далее идет более плавное сокращение времени счета программы. При вычислении слоев функции Беллмана процессоры обмениваются промежуточными данными, вследствие чего возникают накладные расходы по времени, что приводит к уменьшению эффекта от использования параллельного алгоритма для большого количества вычислительных ядер, участвующих в расчетах.

Заключение

В диссертации исследовались важные для теории и инженерных приложений задачи маршрутизации, связанные с процедурами последовательного демонтажа системы радиационно-опасных объектов. Возникающая при этом математическая модель является достаточно сложной и требует разработки серьезных теоретических конструкций для своего исследования. В настоящей работе принята модель мегаполисов, которые подлежат последовательному посещению с целью демонтажа источников излучения при соблюдении определённых ограничений, связанных с учётом особенностей исходной инженерной задачи. Среди упомянутых ограничений особую роль играют условия предшествования, которые удаётся использовать "в положительном направлении"в смысле снижения вычислительной сложности.

В диссертации построены методы и алгоритмы решения упомянутых задач маршрутизации, включая задачи ощутимой размерности. В основе используемых конструкций находится метод ДП, а конструируемые на его основе параллельные алгоритмы реализованы в виде стандартных программ на языке C++ для МВС. Эти алгоритмы используют идеи, заложенные в схемах с независимыми вычислениями слоев функции Беллмана. Они реализованы в двух вариантах:

1. точные алгоритмы, доставляющие глобальные экстремум и оптимальные маршруты;
2. алгоритмы, улучшающие качество эвристик за счет применения оптимизирующих мультивставок (оптимизация "в окнах").

В обоих случаях получено существенное продвижение в плане увеличения размерности решаемых задач. Эту цель удалось достичь, в частности, за счет теоретически обоснованного распараллеливания вычислительных процедур. Представляется, что данный подход к решению очень сложных по постановке инженерных задач имеет смысл развивать и в дальнейшем, распространяя упомянутые конструкции на задачи другой природы. Среди таких задач можно отметить задачу управления инструментом при листовой резке на машинах с ЧПУ, задачу авиапожарного патрулирования лесов, транспортные задачи.

Список литературы

1. *Gutin G., Punnen A. P.* The Traveling Salesman Problem and Its Variations. — Dordrecht: Springer, 2002.

2. *Cook William J.* In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. — Princeton University Press, 2012. — URL: http://www.jstor.org/stable/j.ctt7t8kc.

3. *Меламед И. И., Сергеев С. И., Сигал И. Х.* Задача коммивояжера. Вопросы теории // *Автоматика и телемеханика.* — 1989. — № 9. — С. 3–34.

4. *Меламед И. И., Сергеев С. И., Сигал И. Х.* Задача коммивояжера. Точные алгоритмы // *Автоматика и телемеханика.* — 1989. — № 10. — С. 3–29.

5. *Меламед И. И., Сергеев С. И., Сигал И. Х.* Задача коммивояжера. Приближенные алгоритмы // *Автоматика и телемеханика.* — 1989. — № 11. — С. 3–26.

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

7. *Bellman R.* Dynamic Programming Treatment of the Travelling Salesman Problem // *J. Assoc. Comput. Mach.* — 1962. — no. 9. — Pp. 61–63.

8. *Held M., Karp R. M.* A Dynamic Programming Approach to Sequencing Problems // *Journal of the Society for Industrial and Applied Mathematics.* — 1962. — Vol. 10, no. 1. — Pp. 196–210.

9. *И. Сергеев С.* Использование методов теории оптимального управления для решения некоторых задач дискретной оптимизации. I. Сепарабельная задача // *Автоматика и телемеханика.* — 2006. — № 4. — С. 42–52.

10. *И. Сергеев С.* Использование методов теории оптимального управления для решения некоторых задач дискретной оптимизации. II. Статическая задача коммивояжера // *Автоматика и телемеханика.* — 2006. — № 6. — С. 106–112.

11. *И. Сергеев С.* Использование методов теории оптимального управления для решения некоторых задач дискретной оптимизации. III. Динамическая задача коммивояжера // *Автоматика и телемеханика.* — 2006. — № 7. — С. 27–40.

12. An Algorithm for the Traveling Salesman Problem / John D. C. Little, Katta G. Murty, Dura W. Sweeney, Caroline Karel // *Operations Research.* — 1963. — December. — Vol. 11, no. 6. — Pp. 972–989. — URL: https://ideas.repec.org/a/inm/oropre/v11y1963i6p972-989.html.

13. *Chisman James A.* The clustered traveling salesman problem // *Computers & Operations Research.* — 1975. — Vol. 2, no. 2. — Pp. 115–119.

14. *Ю. Хачай М., Д. Незнахина Е.* Приближенные схемы для обобщенной задачи коммивояжера // *Труды ИММ УрО РАН.* — 2016. — Т. 22, № 3. — С. 283–292.

15. *Ю. Хачай М., Д. Незнахина Е.* Разрешимость обобщенной задачи коммивояжера в классе квази- и псевдопирамидальных маршрутов // *Труды ИММ УрО РАН.* — 2017. — Т. 23, № 3. — С. 280–291.

16. *Balas E.* New classes of efficiently solvable generalized Traveling Salesman Problems // *Annals of Operations Research.* — 1999. — January. — Vol. 86, no. 0. — Pp. 529–558.

17. The Traveling Salesman Problem with Precedence Constraints / Lucio Bianco, Aristide Mingozzi, Salvatore Ricciardelli, Massimo Spadoni // Papers of the 19th Annual Meeting/Vorträge der 19. Jahrestagung / Springer. — 1992. — Pp. 299–306.

18. *Escudero Laureano F.* A production planning problem in FMS // *Annals of Operations Research.* — 1989. — Vol. 17, no. 1. — Pp. 69–103.

19. *Kubo Mikio, Kasugai Hiroshi.* The precedence constrained traveling salesman problem // *Journal of the Operations Research Society of Japan.* — 1991. — Vol. 34, no. 2. — Pp. 152–172.

20. *Gouveia Luis, Ruthmair Mario.* Load-dependent and precedence-based models for pickup and delivery problems // *Computers & Operations Research.* — 2015. — 04. — Vol. 63. — Pp. 56–71.

21. *В. Салий Я.* Влияние условий предшествования на вычислительную сложность решения маршрутных задач методом динамического программирования // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки.*

22. *Salii Yaroslav.* Revisiting Dynamic Programming for Precedence-Constrained Traveling Salesman Problem and Its Time-Dependent Generalization // *European Journal of Operational Research.* — 2017. — 04. — Pp. 32–42.

23. *М. Плотинский Ю.* Обобщенная задача развозки // *Автомат. и телемех.*

24. *Fiala Timlin Marie T, Pulleyblank William R.* Precedence constrained routing and helicopter scheduling: heuristic design // *Interfaces.* — 1992. — Vol. 22, no. 3. — Pp. 100–111.

25. *Escudero Laureano F.* A production planning problem in FMS // *Annals of Operations Research.* — 1989. — Vol. 17, no. 1. — Pp. 69–103.

26. *Dubowsky Steven, Blubaugh TD.* Planning time-optimal robotic manipulator motions and work places for point-to-point tasks // *Robotics and Automation, IEEE Transactions on.* — 1989. — Vol. 5, no. 3. — Pp. 377–381.

27. *Spieckermann Sven, Gutenschwager Kai, Voß Stefan.* A sequential ordering problem in automotive paint shops // *International journal of production research.* — 2004. — Vol. 42, no. 9. — Pp. 1865–1878.

28. *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. — Vol. 13, no. 2. — Pp. 280–286.

29. *Ascheuer Norbert.* Hamiltonian path problems in the on-line optimization of flexible manufacturing systems: Ph.D. thesis / Technische Universität Berlin Germany. — 1995.

30. *Kalantari Bahman, Hill Arthur V, Arora Sant R.* An algorithm for the traveling salesman problem with pickup and delivery customers // *European Journal of Operational Research.* — 1985. — Vol. 22, no. 3. — Pp. 377–386.

31. *Shobaki Ghassan, Jamal Jafar.* 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. — Pp. 343–372.

32. *Lawler E.L.* Efficient implementation of dynamic programming algorithms for sequencing problems. CWI Technical report // *BW 106/79: Stichting Mathematisch Centrum.* — 1979. — Pp. 1–16.

33. *Mitten L G.* Composition principles for synthesis of optimal multistage processes // *Operations Research.* — 1964. — Vol. 12, no. 4. — Pp. 610–619.

34. *Morin Thomas L.* Monotonicity and the principle of optimality // *Journal of Mathematical Analysis and Applications.* — 1982. — Vol. 88, no. 2. — Pp. 665–674.

35. *А.Г. Ченцов, А.А. Ченцов, А.Н. Сесекин.* Задачи маршрутизации перемещений с аддитивным агрегированием затрат. — Москва: Ленанд, 2020.

36. *Alkaya Ali, Duman Ekrem.* A new generalization of the Traveling salesman problem // *Applied and Computational Mathematics.* — 2010. — 01. — Vol. 9. — Pp. 162–175.

37. *Lokin FCJ.* Procedures for travelling salesman problems with additional constraints // *European Journal of Operational Research.* — 1979. — Vol. 3, no. 2. — Pp. 135–141.

38. *Garey M. R., Johnson D. S.* Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). — First Edition edition. — W. H. Freeman, 1979. — URL: http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455.

39. *Р. Беллман, Р. Калаба.* Динамическое программирование и современная теория управления. — Москва: Наука, 1969.

40. *Ченцов А. Г.* Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами // *Автоматика и телемеханика.* — 2012. — № 3. — С. 134–149.

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

42. Динамическое программирование в обобщенной задаче курьера, осложненной внутренними работами / Григорьев А.М., Иванко Е.Е., Князев С.Т., Ченцов А.Г. // *Мехатроника, автоматизация, управление.* — 2012. — № 7. — С. 14–21.

43. *A.G. Chentsov, A.M. Grigoryev.* Scheme of independent calculations in a precedence constrained routing problem // *Lecture Notes in Computer Science.* — 2016. — Vol. 9869. — Pp. 121–135.

44. *Ченцов А. Г., Григорьев А.М.* Динамическое программирование в задаче маршрутизации: схема независимых вычислений // *Мехатроника, Автоматизация, Управление.* — 2016. — Т. 17, № 12. — С. 834–846.

45. *A.G. Chentsov, A.M. Grigoryev, A.A. Chentsov.* Scheme of independent calculations in a precedence constrained routing problem // *CEUR-WS.* — 2017. — Vol. 1987. — Pp. 123–130.

46. *А.Г. Ченцов, А.А. Ченцов, А.М. Григорьев.* Об одной задаче маршрутизации, моделирующей перемещения в радиационных полях // *Вестн. УдГУ. Математика. Механика. Компьютерные науки.* — 2017. — Т. 27, № 4. — С. 540–557.

47. *Григорьев А.М.* Решение задачи об оптимальном распределении заданий методом динамического программирования с применением параллельных вычислений // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки.* — 2017. — Т. 27, № 1. — С. 129–137.

48. *Chentsov A.G., Grigoryev A.M., Chentsov A.A.* Optimizing the starting point in a precedence constrained routing problem with complicated travel cost functions // *Ural Mathematical Journal.* — 2018. — Vol. 4, no. 2. — Pp. 43–55.

49. *A.M. Grigoryev, L. Tashlykov O.* Solving a routing optimization of works in radiation fields with using a supercomputer // *AIP Conference Proceedings.* — 2018. — Vol. 2015. — P. 6.

50. *A.G. Chentsov, A.M. Grigoryev, A.A. Chentsov.* Solving a Routing Problem with the Aid of an Independent Computations Scheme // *Bulletin South Ural State Univ. Ser. Math. Modelling, Programming & Computer Software.* — 2018. — Vol. 11, no. 1. — Pp. 60–74.

51. *Ченцов А. Г., Григорьев А. М.* Оптимизирующие мультивставки в задачах маршрутизации с ограничениями // *Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки.* — 2018. — Т. 28, № 4. — С. 513–530.

52. *A.G. Chentsov, A.M. Grigoryev, A.A. Chentsov.* Optimization "In Windows" for Routing Problems with Constraints // *Communications in Computer and Information Science.* — 2019. — Vol. 1090. — Pp. 470–485.

53. *A.M. Grigoryev, L. Tashlykov O.* Route optimization during works in nonstationary radiation fields with obstacles // *AIP Conference Proceedings.* — 2019. — Vol. 2174. — P. 6.

54. *Ченцов А. Г.* Экстремальные задачи маршрутизации и распределения заданий: вопросы теории. — Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2008.

55. Методы маршрутизации и их приложения в задачах повышения безопасности и эффективности эксплуатации атомных станций / В.В. Коробкин, А.Н. Сесекин, О.Л Ташлыков, А.Г. Ченцов. — Москва: Новые технологии, 2012.

56. *Ташлыков О. Л.* Дозовые затраты персонала в атомной энергетике. Анализ. Пути снижения. Оптимизация. — LAP LAMBERT Academic Publishing, 2011.

57. *Leon V. Jorge, Peters Brett A.* Replanning and analysis of partial setup strategies in printed circuit board assembly systems // *International Journal of Flexible Manufacturing Systems.* — 1996. — Vol. 8, no. 4. — Pp. 389–411. — URL: https://doi.org/10.1007/BF00170019.

58. *Kinable J., Cire A.A., van Hoeve W.J.* Hybrid optimization methods for time-dependent sequencing problems // *European Journal of Operational Research.* — 2017. — 6. — Vol. 259, no. 3. — Pp. 887–897.

59. *Петунин А.А.* О некоторых стратегиях формирования маршрута инструмента при разработке управляющих программ для машин термической резки материала // *Вестник УГАТУ. Серия: Управление, вычислительная техника и информатика.* — 2009. — Т. 13, № 2 (35). — С. 280—286.

60. *Петунин А.А., Ченцов А.Г., Ченцов П.А.* К вопросу о маршрутизации движения инструмента в машинах листовой резки с числовым программным управлением // *Науч.-техн. ведомости СПбГПУ. Серия: Информатика. Телекоммуникации. Управление.* — 2013. — № 2 (169). — С. 103–111.

61. *Фроловский В.Д.* Автоматизация проектирования управляющих программ тепловой резки металла на оборудовании с ЧПУ // *Информационные технологии в проектировании и производстве.* — 2005. — № 4. — С. 63—66.

62. *Wang Gary, XIEz S.* Optimal process planning for a combined punch-and-laser cutting machine using ant colony optimization // *International Journal of Production Research - INT J PROD RES.* — 2005. — 06. — Vol. 43. — Pp. 2195–2216.

63. *Dewil Reginald, Vansteenwegen Pieter, Cattrysse Dirk.* Construction heuristics for generating tool paths for laser cutters // *International Journal of Production Research.* — 2014. — Vol. 52, no. 20. — Pp. 5965–5984. — URL: https://doi.org/10.1080/00207543.2014.895064.

64. *Дьедонне Ж.* Основы современного анализа. — Москва: Мир, 1964.

65. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: Построение и анализ. — Москва: МЦНМО, 2000.

66. *Ченцов А. Г.* К вопросу о маршрутизации комплексов работ // *Вестник Удм. ун-та. Математика. Механика. Комп. науки.* — 2013. — № 1. — С. 59–82.

67. *Ченцов А. Г., Ченцов П. А.* Оптимизация точки старта в задаче последовательного обхода мегаполисов при наличии условий предшествования // *Вестн. ЮУрГУ. Сер. Матем. моделирование и программирование.* — 2018. — Vol. 11, no. 2. — P. 83–95.

68. *Ченцов А. Г., Ченцов П. А.* Об одной задаче маршрутизации с оптимизацией точки старта–финиша // *Изв. ИМИ УдГУ.* — 2018. — Vol. 52. — P. 103–115.

69. *Ченцов А. Г., Ченцов П. А.* Маршрутная задача с оптимизацией стартовой точки: динамическое программирование // *Изв. ИМИ УдГУ.* — 2019. — Т. 54. — С. 102–121.

70. *Alkaya Ali Fuat, Duman Ekrem.* Combining and solving sequence dependent traveling salesman and quadratic assignment problems in PCB assembly // *Discrete Applied Mathematics.* — 2015. — Vol. 192. — Pp. 2–16.

71. *Ченцов А. Г., Ченцов А.А.* Задача маршрутизации с ограничениями, зависящими от списка заданий // *Доклады Академии наук.* — 2015. — Т. 465, № 2. — С. 154–158.

72. *Ченцов А. Г., Ченцов П.А.* Маршрутизация в условиях ограничений: задача о посещении мегаполисов // *Автоматика и телемеханика.* — 2016. — № 11. — С. 96–117.

73. *Петунин А. А., Ченцов А. Г., Ченцов П.А.* Локальные вставки на основе динамического программирования в задаче маршрутизации с ограничениями // *Вестник Удмуртского университета.* — 2014. — № 2. — С. 56–75.

74. *Ченцов А. Г.* Беллмановские вставки в задаче маршрутизации с ограничениями и усложненной функцией стоимости // *Вестник Удмуртского университета.* — 2014. — № 4. — С. 122–141.

75. *Ченцов А. Г.* Оптимизирующие вставки в задачах маршрутизации и их реализация на основе динамического программирования // *Вестник Удмуртского университета.* — 2016. — № 4. — С. 565–578.

76. *Ченцов А. Г.* Об оптимальной маршрутизации в условиях ограничений // *Доклады Академии наук.* — 2008. — Т. 423, № 3. — С. 303–307.

77. *Ченцов А. Г.* Задача последовательного обхода мегаполисов с условиями предшествования // *Автоматика и телемеханика.* — 2014. — № 4. — С. 170–190.

78. *Ченцов А. Г., Ченцов А.А.* К вопросу о нахождении значения маршрутной задачи с ограничениями // *Проблемы управления и информатики.* — 2016. — № 1. — С. 41–54.

79. *Ченцов А. Г.* Одна параллельная процедура построения функции Беллмана в обобщенной задаче курьера с внутренними работами // *Вестник ЮУрГУ. Сер. Мат. моделирование и программирование.* — 2012. — Т. 12, № 18. — С. 53–76.

80. *Ченцов А. Г., Ченцов А. А.* Модельный вариант задачи о последовательной утилизации источников излучения (итерации на основе оптимизирующих вставок) // *Изв. ИМИ УдГУ.* — 2017. — Т. 50. — С. 83–109.

81. *Александров П.С., Маркушевич А.И., Хинчин А.Я.* Энциклопедия элементарной математики. Книга 3. Функции и пределы. — М.-Л.: ГИТТЛ, 1952.

82. *Ченцов А. А., Ченцов А. Г., Ченцов П. А.* Экстремальная задача маршрутизации с внутренними потерями // *Тр. ИММ УрО РАН.* — 2008. — Т. 14, № 3. — С. 183–201.

83. *Ченцов А. А., Ченцов А. Г.* Задача последовательного обхода мегаполисов // *Вестник Тамбовского университета. Сер. Естественные и технические науки.* — 2014. — Т. 19, № 2. — С. 454–475.

84. *Ченцов А. Г., Кошелева М. С.* Динамическое программирование в задаче курьера со стоимостями, зависящими от списка заданий // *Мехатроника, автоматизация, управление.* — 2015. — Vol. 16, no. 4. — Pp. 232–244.

85. *Куратовский К., Мостовский А.* Теория множеств. — Москва: Мир, 1970.

86. Элементы динамического программирования в конструкциях локального улучшения эвристических решений задач маршрутизации с ограничениями / А. А. Петунин, А. А. Ченцов, А. Г. Ченцов, П. А. Ченцов // *Автоматика и телемеханика.* — 2017. — № 4. — С. 106–125.

87. Возможности математических методов моделирования в решении проблемы снижения облучаемости персонала / Ташлыков О.Л., Сесекин А.Н., Щеклеин С.Е. et al. // *Вопросы радиационной безопасности.* — 2009). — no. 4. — P. 47–57.

88. *Собрание законодательства Российской Федерации № 3 ст.141.* Федеральный закон «О радиационной безопасности населения» № З - ФЗ от 09.01.96. — Москва, 1996.

89. *2.6.1.2523-09 СанПиН.* Нормы радиационной безопасности (НРБ-99/2009). — Москва, 2009.

90. *2.6.1.2612-10 СП.* Основные санитарные правила обеспечения радиационной безопасности (ОСПОРБ – 99/2010). — Москва, 2010.

91. *2.6.5.054-2017 МУ.* Оптимизация радиационной защиты персонала предприятий Госкорпорации «Росатом». Методические указания. — Москва: Федеральное медико-биологическое агентство, 2017.

92. *Чнецов А.Г., Ченцов П.А.* К вопросу о построении процедуры разбиения конечного множества на основе метода динамического программирования // *Автоматика и телемеханика.* — 2000. — № 4. — С. 129–142.

93. *Ченцов А. Г., Ченцов П. А.* Динамическое программирование в задаче оптимизации разбиений // *Автоматика и телемеханика.* — 2002. — № 5. — С. 133–146.

94. *Schrage Linus, Baker Kenneth R.* Dynamic programming solution of sequencing problems with precedence constraints // *Operations Research.* — 1978. — Vol. 26, no. 3. — Pp. 444–449.

95. *Ченцов А. Г., Ченцов П. А.* Маршрутная задача с условиями предшествования (задача курьера): метод динамического программирования // *Вестн. УГТУ-УПИ.* — 2004. — no. 15. — Pp. 148–151.

96. *Ченцов А. Г.* О структуре одной экстремальной задачи маршрутизации с ограничениями в виде условий предшествования // *Вестн. Удмуртского ун-та. Математика.* — 2006. — no. 1. — Pp. 127–150.

97. *Ченцов А. Г.* Экстремальные задачи маршрутизации с ограничениями // *Изв. института матем. и информатики.* — 2006. — no. 3. — Pp. 163–166.

98. *Buhmann Martin D.* Radial Basis Functions: Theory and Implementations. — Cambridge University Press, 2003.

99. *Dijkstra E. W.* A Note on Two Problems in Connexion with Graphs // *NUMERISCHE MATHEMATIK.* — 1959. — Vol. 1, no. 1. — Pp. 269–271.

100. *Левитин А.В.* Алгоритмы: введение в разработку и анализ. — Москва: Издательский дом "Вильямс 2006.

101. *Омельченко А. В.* Теория графов. — Москва: МЦНМО, 2018.

102. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация: Алгоритмы и сложность. — Москва: Мир, 1984.

103. *Оре О.* Теория графов. — Москва: Наука, 1980.

104. *Burkard Rainer., Dell'Amico Mauro., Martello Silvano.* Assignment Problems. — Society for Industrial and Applied Mathematics, 2012. — URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972238.

105. *Kuhn H. W.* The Hungarian method for the assignment problem // *Naval Research Logistics Quarterly.* — 1955. — Vol. 2, no. 1-2. — Pp. 83–97. — URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

106. *Kellerer Hans, Pferschy Ulrich, Pisinger David.* Knapsack Problems. — 2004. — 01.

107. *Papadimitriou Christos H., Yannakakis Mihalis.* Optimization, approximation, and complexity classes // *Journal of Computer and System Sciences.* — 1991. — Vol. 43, no. 3. — Pp. 425 – 440. — URL: http://www.sciencedirect.com/science/article/pii/002200009190023X.

108. *Коротаева Л.Н., Назаров Э.М., Ченцов А.Г.* Об одной задаче о назначениях // *Журн. вычисл. математики и мат. физики.* — 1993. — Т. 33, № 4. — С. 483–494.

109. *Беллман Р.* Динамическое программирование. — Москва: Издательство иностранной литературы, 1960.

110. *Мину М.* Математическое программирование. — Москва: Наука, 1990.

111. *Ченцов П. А.* О некоторых алгоритмах распределения заданий между участниками // *Автоматика и телемеханика.* — 2006. — № 8. — С. 77–91.

112. *Григорьев А.М.* Решение минимаксной распределительной задачи методом динамического программирования с применением параллельных вычислений // *Научный сервис в сети Интернет: экзафлопсное будущее: Труды Международной суперкомпьютерной конференции.* — 2011. — С. 580–586.

113. *Антонов А.С.* Параллельное программирование с импользованием техгологии MPI: Учебное пособие. — Москва: МГУ, 2004.