ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ИНСТИТУТ ДИНАМИКИ СИСТЕМ И ТЕОРИИ УПРАВЛЕНИЯ
имени В.М. МАТРОСОВА
СИБИРСКОГО ОТДЕЛЕНИЯ РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Ушаков Антон Владимирович

НЕЛИНЕЙНЫЙ ВАРИАНТ ЗАДАЧИ О P-МЕДИАНЕ И ПОРОГОВАЯ РОБАСТНОСТЬ ДОПУСТИМЫХ РЕШЕНИЙ В ДИСКРЕТНЫХ ЗАДАЧАХ РАЗМЕЩЕНИЯ

05.13.01 – Системный анализ, управление и обработка информации
(в технике, экологии и экономике)

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель
к.ф.-м.н., доцент
И.Л. Васильев

Иркутск – 2016

Оглавление

Введение

The final test of any theory is its capacity to solve the problems which originated it.

– George B. Dantzig

Задачи размещения являются, возможно, одними из наиболее интенсивно изучаемых моделей в области исследования операций, а также одними из старейших задач современной или «высшей» математики. Традиционно полагают, что первой задачей размещения была предложенная Пьером Ферма головоломка о размещении на плоскости одной точки относительно трех заданных таким образом, чтобы сумма расстояний от данной точки до трех остальных была настолько малой, насколько это возможно. Первое геометрическое решение этой задачи, как полагают, было получено известным математиком Э. Торичелли. Представленная задача получила множество названий и известна как задача Ферма, Ферма-Торичелли и другие. Вероятно, первой настоящей классической задачей размещения можно считать так называемую задачу Вебера, представляющую собой обобщение задачи Ферма, где каждой заданной на плоскости точке приписывался некоторый вес. Таким образом, в данной задаче точка, размещаемая на плоскости, может быть интерпретирована как предприятие, местоположение которого выбирается так, чтобы минимизировать расстояния (транспортные затраты) до уже заданных точек, представляющих собой клиентов, которые обладают некоторым спросом, выраженным весами. Другой ранней геометрической задачей, которую к настоящему времени также относят к классическим задачам размещения, является так называемая задача Сильвестра. В первоначальной постановке она заключалась в поиске наименьшего круга, включающего в себя некоторый заданный набор точек на плоскости. В современной интерпретации данную задачу называют задачей об 1-центре на плоскости, в которой необходимо разместить одно предприятие относительно заданного набора клиентов таким образом, чтобы самое большое расстояние от клиента до предприятия было минимальным. Несмотря на другие важнейшие исследования и результаты, которые на сегодняшний день также связывают с задачами размещения, и представленные именами таких экономистов и географов как Август Лёш, Уолтер Айзард, Уильям Алонсо и др., зарождение современной теории размещения, как полагают, произошло в середине 60-х годов XX века вместе с выходом в свет известных работ С. Л. Хакими [157, 158]. Дальнейшее бурное развитие данной области связано с резким ростом числа приложений задач размещения, появлением и стремительным развитием ЭВМ, а также быстрым прогрессом математического программирования в общем

и таких его подразделов как линейное и целочисленное программирование в частности. Известно, что в СССР первые исследования подобного рода задач, связанные с именами таких ученых как В.П. Чечерин и В.Р. Хачатуров, имели место начиная с середины 60-х годов XX века, т.е. фактически с зарождения современной теории задач размещения. Дальнейшему развитию данного направления исследований в нашей стране посвящены работы В.Л. Береснева, Э.Х. Гимади, Ю.Б. Гермейера, В.Т. Дементьева, Н.И. Глебова, А.А. Колоколова, а также И.Л. Васильева, Г.Г. Забудского, Ю.А. Кочетова, А.В. Плясунова, Ю.В. Шамардина и др.

Несмотря на наличие большого числа различных моделей задач размещения, всех их объединяет ряд общих базовых компонентов, а именно: некоторое пространство с заданной метрикой, задающей расстояние; заданный набор точек, а также множество возможных мест для размещения фиксированного или переменного числа новых точек [136]. Отметим, что заданные, известные точки часто называют клиентами, в то время как размещаемые — предприятиями. В литературе различают два типа задач в зависимости от пространства, в котором размещаются предприятия. Так, выделяют задачи размещения в n-мерном вещественном пространстве и задачи размещения на сетях (графах), которые в свою очередь подразделяют на непрерывные и дискретные задачи. В случае вещественного пространства в качестве метрики, задающей расстояния между предприятиями и клиентами, часто используют так называемое расстояние Минковского, либо в более общих случаях функционал Минковского (калибр). Таким образом, в непрерывной задаче размещения в пространстве (чаще всего на плоскости) для каждого предприятия необходимо фактически найти координаты его положения относительно множества клиентов в зависимости от цели задачи, что приводит к нелинейным задачам оптимизации, поскольку функции, задающие расстояния, чаще всего являются нелинейными. Отметим, что если задача ставится на графе, то в непрерывном случае предприятия могут размещаться в любой точке, т.е. как на ребрах, так и в вершинах графа. С другой стороны, в дискретных задачах размещения на плоскости или на графе помимо заданного конечного множества клиентов имеется конечное множество точек, в которых могут быть размещены предприятия, что фактически сводит задачу к принятию решения о том, стоит ли размещать предприятие в некотором заданном месте, что, например, с легкостью может быть смоделировано с использованием булевых переменных, приводя в итоге к задачам целочисленного линейного программирования.

Несмотря на общее название, дискретные и непрерывные задачи размещения имеют существенные различия в структуре и свойствах и требуют применения различных подходов к поиску решений, что выделяет их в отдельные области теории размещения. Подробные

вопросы классификации задач размещения, а также обзор различных областей их практического применения и некоторых открытых проблем представлен, например, в [54, 245, 246]. На протяжении последних десятилетий было опубликовано свыше трех с половиной тысяч работ, посвященных исследованию различных постановок задач размещения, а также смежных задач (например, о присоединении клиентов к тем или иным предприятиям), изучению их структуры и свойств, разработке различных точных и эвристических методов поиска решений. Так, например, в известной библиографии Тревора Хэйла [159] собраны ссылки на более чем 3400 работ, посвященных различным задачам теории размещения, и этот список далеко не полный.

В настоящей диссертационной работе исследуются дискретные задачи размещения, а именно такие классические, базовые модели как простейшая задача размещения, задача о p-медиане и их нелинейные обобщения. Несмотря на длительную историю и большое число полученных результатов, дискретные задачи размещения привлекают все большее внимание исследователей в последние годы. Это вызвано не только большим числом практических приложений таких задач, возникающих зачастую в областях, не связанных напрямую с размещением, но и тем фактом, что большинство таких задач NP-трудны и разработка эффективных методов их решения является актуальной проблемой. В качестве примера нетрадиционного приложения задач размещения, отличного от так называемых «геогрaфических» приложений, стоит выделить так называемую задачу позиционирования товара на рынке, в которой каждое предприятие «представляет» собой некий продукт, заданный в виде некоторого вектора в пространстве признаков, каждый компонент которого выражает некую характеристику данного продукта. Схожим образом в пространстве признаков представляется каждый клиент или группа клиентов. Тогда расстояния между клиентами и продуктами охарактеризуют, какой из продуктов каждый клиент намерен купить, позволяя таким образом сделать оценки касательно продаж данных видов продукции [245].

В настоящее время исследования в области дискретных задач размещения как на плоскости, так и на графах подразделяется на ряд направлений, зачастую тесно не связанных между собой. Среди них в первую очередь стоит выделить разработку новых более сложных моделей, все более реалистично отражающих разнообразные практические ситуации. Другими направлениями являются анализ наихудших случаев, исследование полиэдральных свойств многогранника задачи (полиэдральная комбинаторика), разработка эвристических алгоритмов и методов локального поиска, вероятностный анализ, разработка точных методов, приближенных алгоритмов с априорными оценками точности, а также различных гибридных методов и техник.

Основное внимание в диссертационной работе посвящено исследованию задачи о p-медиане и ее обобщений, которая является одной из первых дискретных задач размещения на графе и одной из первых классических задач размещения вообще. В первоначальной постановке задачи, представленной Хакими [158], предполагалось, что имеется некоторая коммуникационная сеть, например телефонная система связи, в которой, как правило, необходимо наличие некоторого числа коммутаторов. Все потоки сообщений или сигналов в сети должны быть получены одним из таких устройств, обработаны и переданы в место назначения. Причем дополнительно предполагается, что стоимостью на установление связи между коммутаторами можно пренебречь. В качестве модели такой сети можно рассмотреть взвешенный граф G, каждой вершине которого также приписан некоторый неотрицательный вес. Отметим, что вес ребра графа G в данном случае может быть интерпретирован как стоимость на единицу объема передаваемых данных, в то время как вес каждой вершины — как количество кабелей или линий связи, которые необходимо проложить между соответствующей вершиной и коммутатором для того, чтобы обработать поток информации, поступающей от данной вершины или передаваемой в нее. Тогда задача о p-медиане состоит в размещении p коммутаторов на графе G таким образом, чтобы суммарная длина необходимых линий связи была минимальна. Причем в оригинальной постановке предполагается, что каждый из коммутаторов может располагаться не только в вершинах графа, но и в любой точке каждого ребра. Другими словами, если предположить, что вершинам v_i, $i = 1, \ldots, n$ графа G приписан вес h_i, и обозначить через $X_p = \{x_1, x_2, \ldots, x_p\}$ всякое множество из p точек на графе G, то необходимо найти такое множество X_p^*, что

$$\sum_{i=1}^{n} h_i d(v_i, X_p^*) \leq \sum_{i=1}^{n} h_i d(v_i, X_p),$$

где $d(v_i, X_p) = \min\{d(v_i, x_1), d(v_i, x_2), \ldots, d(v_i, x_p)\}$, а $d(x, y)$ — кратчайшее расстояние между точками x и y на графе G. Отметим, что множество X_p^* называется p-медианой графа.

Задача поиска медианы графа во многом схожа с задачей Вебера, в которой также имеется набор точек с весами (клиенты). Однако в отличие от задачи на плоскости клиенты располагаются только в вершинах, в то время как медианы в наиболее общей постановке Хакими могут размещаться в любой точке графа, включая вершины и ребра. Тем не менее, Хакими, сперва для медианы графа [157], а затем и для p-медианы [158], был доказан важнейший теоретический результат, часто называемый свойством Хакими, который утверждает, что по крайней мере одно оптимальное решение задачи о p-медиане состоит из медиан, размещенных в вершинах графа. Таким образом, поиск оптимального решения можно ограничить лишь перебором конечного множества вершин, а не исследованием непрерывного и бесконеч-

ного множества всех точек на графе, что уменьшает количество допустимых решений задачи до $\binom{n}{p} = \frac{n!}{p!(n-p)!}$. Следует также подчеркнуть, что свойство Хакими справедливо далеко не для всех задач размещения на графе и установление его справедливости для той или иной задачи размещения является довольно распространенным направлением исследований. Так, например, в случае задачи о покрытии множеств и задачи о p-центрах размещение «предприятий» в вершинах графа дает в общем случае лишь субоптимальное решение [97, 110]. Отметим также, что целевая функция задачи о p-медиане, в которой минимизируется сумма взвешенных расстояний от вершин графа (клиентов) до ближайшей медианы (предприятия), часто называют минисумным критерием. Критерии такого типа также применяются при моделировании других задача размещения, в частности простейшей задачи размещения.

Известно, что задача о p-медиане на простом графе для произвольного p является NP-трудной, однако полиномиально разрешима при фиксированном значении p [12], а также в случае постановки на дереве [182]. Если вместо графа рассматривать задачу о p-медиане на плоскости, т.е. задачу размещения p «предприятий» относительно заданного набора «клиентов», в которой в качестве метрики, задающей расстояния, выбирается евклидова или манхеттеновская метрики, то данная задача также NP-трудна [216].

Как полагают, первый эвристический алгоритм для задачи о p-медиане был предложен Маранцаной [213]. Алгоритм представляет собой достаточно простую процедуру, в которой выбирается некоторое начальное множество из p вершин, к ближайшим из которых присоединяются все оставшиеся, формируя тем самым разбиение вершин графа. Далее, в каждом из получившихся подмножеств выбирается вершина, сумма расстояний до которой от остальных вершин в данном подмножестве минимальна, т.е. в каждом из подмножеств ищется решение задачи о 1-медиане. Если после этого изначальное положение медиан не изменилось, то алгоритм останавливается, в противном случае для каждой вершины вновь ищется ближайшая медиана. Если ни одно из назначений вершин к медианам не изменилось, то алгоритм останавливается, в противном случае в каждом из подмножеств нового разбиения вновь ищется решение задачи о 1-медиане и процесс повторяется. Данный алгоритм сходится лишь к некоторому локальному минимуму и в общем случае не дает оптимального решения задачи.

Следующим предложенным алгоритмом, также сходящимся лишь к локальному решению задачи, была известная эвристика Тейтца и Барта [267]. В рамках данного подхода вначале выбирается некоторое решение задачи, состоящее из p вершин графа. Отметим, что в качестве такого начального приближения может быть использовано решение, полученное методом Маранцаны. Затем алгоритм пытается улучшить данное решение посредством замены

одной из вершин на другую, не содержащуюся в решении. Если полученное с помощью такой модификации новое решение дает лучшее значение целевой функции, то оно выбирается в качестве следующего текущего решения и процесс повторяется. Алгоритм останавливается, если не существует никаких других возможных замен вершин, дающих лучшее значение целевой функции. Следует отметить, что представленные эвристические алгоритмы являются первыми представителями так называемых методов локального поиска. Они положили начало новому классу быстрых и одних из наиболее эффективных эвристических алгоритмов, применяемых на сегодняшний день к широкому классу задач.

Первый большой прорыв в исследовании задачи о p-медиане был совершен с выходом в свет известной работы Ревелле и Свейна [247], где была представлена первая целочисленная линейная постановка задачи, названная задачей размещения центральных предприятий. Авторами проведено исследование рассматриваемой целочисленной модели, а также, что наиболее важно, предложен метод ветвей и границ, являющийся первым точным алгоритмом поиска решений в задаче о p-медиане, за исключением полного перебора, эффективного лишь для задач небольшой размерности. Данная статья послужила отправной точкой для современных исследований задачи о p-медиане, разработки большого числа различных точных и эвристических алгоритмов, а также исследованию структуры задачи.

Будучи одной из первых моделей дискретных задач размещения, обладающая к тому же краткой и ясной формулировкой, задача о p-медиане всегда привлекала широкое внимание исследователей в области дискретной оптимизации и целочисленного программирования. На протяжении десятилетий она являлась основным объектом для разработки и тестирования различных методов, техник и подходов, впоследствии получивших широкое применение в других областях дискретной оптимизации. Наиболее подробный обзор различных точных и эвристических методов поиска решений задачи, а также некоторых теоретических результатов, может быть найден в известных статьях [222, 242], а также в монографиях [110, 114, 125, 126, 202] и других многочисленных источниках.

К настоящему моменту задача о p-медиане представляет собой естественно интерпретируемую модель размещения, применяемую в различных областях, из которых прежде всего стоит выделить так называемые «географические» приложения, т.е. непосредственно связанные с размещением предприятий и различных объектов инфраструктуры, таких как пожарные станции и пункты скорой помощи, буровые платформы, школы, пункты снабжения, датчики в системах распределения (например, в муниципальных системах водоснабжения), специализированные банковские счета для оптимизации прохождения средств и т.д. Задача о p-медиане, как и некоторые другие модели, также нашла свое применение в областях,

которые относят к так называемым «негеографическим» приложениям, среди которых стоит выделить приложения в области стандартизации и унификации [3], позиционирования товара на рынке [136], задачу об оптимальном замещении в автомобильной промышленности [5, 83], а также в кластерном анализе [165, 186, 225], предсказательной аналитике [129], интеллектуальном анализе данных [81] и других областях.

Задача кластерного анализа, являясь одним из важнейших приложений задачи о p-медиане, состоит в разделении некоторого множества объектов (наблюдений, событий) на непересекающиеся подмножества (кластеры) на основе свойств, описывающих сущность этих объектов. Объекты внутри кластера должны быть «похожими» друг на друга и отличаться от объектов, находящихся в других кластерах. Чем больше схожи между собой объекты внутри кластера и чем больше отличий между объектами из разных кластеров, тем точнее выполнена кластеризация. Кластеризация широко используется в маркетинге, биоинформатике, распознавании образов, анализе текстов и многих других областях. Подходы, основанные на применении задачи о p-медиане, являются одним из известных инструментов решения задачи кластерного анализа, для чего каждый из объектов ассоциируется с некоторой вершиной графа, а мера схожести объектов между собой с весом соединяющего их ребра. Отметим, что часто объекты задаются в виде векторов в некотором пространстве признаков, тогда мера их схожести может быть подсчитана как расстояние между соответствующими векторами с использованием подходящей метрики. Известно, что задача о p-медиане дает решение задачи кластерного анализа известное как «минимум суммы звезд» [165], поскольку каждое из решений задачи состоит из p звезд, в центре которых находятся медианы. Каждая такая звезда представляет собой кластер, в то время как медиана является его представителем. Несмотря на то, что подход к задаче кластерного анализа, основанный на задаче о p-медиане дает довольно высокое качество кластеризации и зачастую превосходит некоторые популярные методы, такие как, например, k-means, применение данной модели затруднено высокой сложностью соответствующей оптимизационной задачи и отсутствием эффективных методов поиска решений для примеров большой размерности, характерной для реальных практических задач кластеризации. В связи с этим, в настоящий момент большое внимание также уделяется разработке новых алгоритмов, в том числе с применением современных средств и методов популярной и бурно развивающейся области параллельных и распределенных вычислений, что подтверждено, в частности, все возрастающим интересом к подобного рода технологиям со стороны компаний, разрабатывающих и предлагающих коммерческие решатели.

Задача о p-медиане, как и другие классические дискретные задачи размещения, является довольно хорошо изученным объектом. Посему в настоящий момент большое число иссле-

дований посвящено различным обобщениям задач, связанным с возникновением все новых разнообразных приложений в области размещения предприятий и объектов инфраструктуры с учетом различных экономических факторов, таких, например, как эффект масштаба. Другие приложения также возникают в области кластеризации и интеллектуальной обработки информации. В последнем случае стоит выделить задачу разбиения на кластеры объектов, представленных не одним, а двумя векторами разнородных характеристик (что характерно, например, для ряда задач кластеризации в биоинформатике и интернет технологиях). Так, одни из интенсивно исследуемых в настоящий момент обобщений дискретных задач размещения связаны прежде всего с наличием естественных экономических факторов, влияющих на стоимость производства, размещения предприятий или доставку продукции от предприятий к клиентам в зависимости от объема необходимой продукции, что может приводить зачастую к нелинейным задачам дискретной оптимизации. Развитию другого класса обобщений способствуют экономические факторы, связанные с естественным изменением с течением времени различных исходных данных задачи, таких как спрос клиентов, расстояния между предприятиями и клиентами и т.д. В случае задачи размещения p коммутаторов в некоторой телекоммуникационной сети к таким обобщениям можно также отнести возможность отказа оборудования или разрыв связи между узлами. Наличие таких задач и постановок, все более точно отражающих реальные процессы и естественные экономические факторы, часто приводит к моделям, структура которых существенно отличается от классических дискретных задач размещения и требует разработки новых техник и алгоритмов их решения.

Известные методы поиска решений в задаче о p-медиане. Среди первых точных алгоритмов для задачи о p-медиане стоит выделить классический метод ветвей и границ, основанный на оригинальной комбинаторной постановке задачи, предложенный в [180], а также известный вариант метода ветвей и границ, основанный на использовании релаксации Лагранжа и субграиентного метода для получения оценок оптимального значения [228]. Дальнейшее развитие точных методов связано прежде всего с исследованием полиэдральной структуры соответствующей задачи целочисленного линейного программирования, а также различных формулировок задачи о p-медиане. В данном направлении стоит выделить прежде всего статью [58], в которой исследуются свойства многогранника задачи о p-медиане, предлагается ряд новых классов гранеобразующих правильных неравенств, а также основанный на них метод ветвей и отсечений. Также новые семейства гранеобразующих неравенств для многогранника задачи и процедуры отсечения исследовались в работах [111, 112] и были обобщенны впоследствии в [280]. Другой известной статьей, посвященной разработке точного метода для задачи о p-медиане, является [60], в которой помимо нового класса правильных

неравенств предлагается метод ветвей, отсечений и оценок для поиска оптимальных решений в задачах рекордной к тому моменту размерности в 3700 вершин. В [254] также рассматривался метод ветвей и оценок для задачи о p-медиане, однако основанный на использовании суррогатной Лагранжевой релаксации, позволившей существенно ускорить алгоритм. Наконец, наиболее эффективным точным алгоритмом для задачи о p-медиане к настоящему времени является предложенный в [140] подход, названный авторами методом «расширений и ветвей». Он представляет собой вариант метода ветвей и границ, использующий специальную процедуру ветвлений в сочетании с методом генерации строк и столбцов, основанном на новой целочисленной формулировке задачи. Предложенный алгоритм позволил превзойти известные ранее точные алгоритмы и найти оптимальные решения для примеров, не поддающихся ранее разработанным точным алгоритмам, в том числе и современным коммерческим решателям. Авторами представлены оптимальные решения для задач рекордной к настоящему моменту размерности (до 89600 вершин), полученные однако только при относительно большом значении p.

Среди эвристических алгоритмов для задачи о p-медиане, как и для других дискретных задач размещения, прежде всего стоит выделить широкий класс так называемых лагранжевых эвристик [68, 110, 275]. В основе таких алгоритмов лежит так называемый метод релаксаций Лагранжа, в рамках которого для исходной задачи строится двойственная посредством добавления функций, задающих ограничения, в целевую функцию задачи с некоторыми весами (двойственными переменными или множителями Лагранжа). Поскольку получающаяся с помощью такого преобразования двойственная задача является негладкой, то для поиска решения (наилучших значений двойственных переменных) применяются различные методы негладкой оптимизации, наиболее популярным из которых и одним из наиболее эффективных является субградиентный алгоритм. В некоторых случаях решение, соответствующее оптимальным множителям Лагранжа, является также оптимальным и в исходной задаче, однако в большинстве случаев такое решение, будучи недопустимым, может предоставить лишь нижнюю оценку оптимального значения. Посему для поиска приближенных решений разрабатываются процедуры поиска верхней или прямой оценки оптимального значения посредством «достраивания» до допустимого полученного в ходе решения двойственной задачи недопустимого решения с помощью различных эвристических техник. Также в лагранжевых эвристиках информацию, полученную в ходе решения двойственной задачи, часто используют для построения более сложных эвристических методов, позволяющих находить допустимые решения, достаточно близкие к оптимальным [56, 59, 83]. Одним из важных преимуществ лагранжевых эвристик, в полной мере справедливым и по отношению к задаче о p-медиане,

является возможность оценить полученное приближенное решение задачи в связи с наличием нижней оценки оптимального значения. Отметим, что в рамках настоящей диссертационной работы основное внимание будет уделено данному направлению разработки методов поиска приближенных решений в задаче о p-медиане и ее обобщениях, поскольку к настоящему времени лагранжевы эвристики представляют собой одни из наиболее эффективных приближенных методов. Подробнее о релаксациях Лагранжа и об одной эффективной лагранжевой эвристике для задачи о p-медиане будет рассказано в первой главе.

К другими успешным и популярным методам поиска приближенных решений в задаче о p-медиане, помимо описанных выше классических эвристик Маранцаны, Тейтца и Барта, относятся также разнообразные метаэвристические подходы, в основе которых лежат различные варианты локального поиска и процедуры выхода из локального решения. Наиболее полный обзор таких подходов для задачи о p-медиане, как и других классических эвристик, может быть найден в [222]. Среди них стоит выделить различные варианты алгоритма имитации отжига, по-видимому впервые реализованного для задачи о p-медиане в статье [146]. Различные реализации этого алгоритма и численные исследования представлены также, например, в [24, 227]. Для рассматриваемой задачи многочисленные работы посвящены различным вариантам таких методов, как локальный поиск с запретами [11, 248], локальный поиск с чередующимися окрестностями [166] (для развития которого задача о p-медиане имела ключевое значение) и его модификации [167, 168], метод муравьиных колоний [24, 187], генетические алгоритмы [50, 200], нейронные сети [115], GRASP эвристика [244] и другие. Помимо прочего к настоящему времени для задачи о p-медиане предложено большое число разнообразных экзотических метаэвристических подходов, таких, например, как ALCMA эвристика [243]. На сегодняшний день одним из наиболее эффективных алгоритмов для задачи о p-медиане большой размерности является прямо-двойственный метод локального поиска с чередующимися окрестностями, описанный в работе [164], позволяющий искать «хорошие» нижние и верхние оценки для тестовых примеров размерности до 20000 вершин, а также приближенные решения для задач размерности до 89600 вершин на стандартных персональных компьютерах. Также стоит отметить предложенный совсем недавно гибридный многоэтапный эвристический алгоритм, включающий в себя процедуры агрегирования данных и специальный метод локального поиска с чередующимися окрестностями [177, 178]. Данный алгоритм позволил успешно найти приближенные решения и обновить некоторые из рекордов как по значению целевой функции, так и по скорости работы алгоритма для тестовых задач максимальной на сегодняшний день размерности в 89600 вершин. Однако главным, серьезным недостатком алгоритма является возможность его применения только для метри-

ческих задач в двумерном пространстве, что существенно ограничивает его применение во многих приложениях задачи о p-медиане.

Для задачи о p-медиане, как для одной из фундаментальных дискретных задач размещения, наряду с точными и эвристическими методами разработан ряд приближенных алгоритмов с априорными оценками точности. В основе таких алгоритмов лежат разнообразные техники, такие как прямо-двойственные методы и релаксация Лагранжа [179], целочисленное округление [93], локальный поиск [53] и другие [199]. Отметим, что большинство таких алгоритмов имеют дело с метрическими задачами (где элементы матрицы расстояний удовлетворяют неравенству треугольника). Первый приближенный алгоритмом с точностью, ограниченной константным множителем $6\frac{2}{3}$, для метрической задачи о p-медиане был предложен в работе [93]. Совсем недавно в [199] был предложен приближенный алгоритм с точностью $1+\sqrt{3}+\varepsilon$, $\varepsilon > 0$, и временем работы $O(n^{O(1/\varepsilon^2)})$, который, однако, был улучшен в [86], где предложен приближенный алгоритм с точностью $2.611+\varepsilon$ и временем работы $O(n^{O((1/\varepsilon)\log(1/\varepsilon))})$, являющийся к настоящему моменту лучшим из известных приближенных алгоритмов. Стоит также особо отметить теоретические результаты, касающиеся вычислительной сложности алгоритмов локального поиска для задачи о p-медиане [23, 49, 53]. Так, в работе [53] проведен первый анализ метода локального поиска для задачи о p-медиане, причем было показано, что локальный поиск относительно SWAP-окрестности приводит к локальному оптимуму с относительной погрешностью не более 5, в то время как в случае k-SWAP окрестности такая погрешность составляет не более $3+\frac{2}{k}$.

Цель диссертационной работы состоит в исследовании известных и новых нелинейных вариантов задачи о p-медиане и простейшей задачи размещения, имеющих вполне определенное экономическое приложение, а также в разработке и программной реализации методов поиска оптимальных и приближенных решений, в том числе с использованием параллельных вычислений.

Объектом исследования диссертационной работы являются дискретные задачи размещения, такие как задача о p-медиане и простейшая задача размещения, а также их нелинейные и бикритериальные постановки. Предметом исследования являются методы поиска точных и приближенных решений в таких задачах, в том числе с использованием параллельных вычислений.

Научная новизна. Для известного ранее нелинейного варианта задачи о p-медиане, в котором число открываемых предприятий p является переменной величиной, а в целевой функции присутствует нелинейное слагаемое, ограничивающее число открываемых предприятий, впервые построены и исследованы два типа релаксаций Лагранжа относительно раз-

личных групп ограничений. Получены формулы вычисления значений двойственных функций Лагранжа для заданного набора двойственных переменных, а также теоретические результаты, позволяющие упростить вычисление значения двойственных функций в зависимости от свойств нелинейного слагаемого. На основе полученных теоретических результатов предложен новый алгоритм поиска приближенных решений в нелинейном варианте задачи о p-медиане большой размерности.

Впервые обобщен известный подход к определению робастности решения в непрерывных задачах размещения, называемый пороговой робастностью, на случай дискретных задач на примере задачи о p-медиане и простейшей задачи размещения. Предложены новые робастные версии таких моделей, представляющие собой бикритериальные задачи нелинейного целочисленного программирования и предполагающие поиск компромиссных решений относительно робастности и значения целевой функции. Предложен и обоснован алгоритм поиска аппроксимации множества Парето-оптимальных решений, так называемых δ-эффективных решений, основанный на известном методе ε-ограничений (или методе главного критерия), учитывающем специфику рассматриваемых задач. Проведен обширный вычислительный эксперимент на тестовых примерах из известных библиотек, показавший эффективность предложенного подхода.

Предложен новый параллельный алгоритм поиска нижних оценок оптимального значения задачи о p-медиане, основанный на методе релаксаций Лагранжа и специальном методе генерации столбцов, а также включающий в себя специальную процедуры хранения данных. Важной особенностью разработанного параллельного алгоритма является, впервые реализованная для рассматриваемой задачи, схема иерархической (каскадной) сборки данных между параллельными процессами, позволяющая существенно ускорить время работы процедуры поиска нижних оценок на задачах размерностью свыше 100000 узлов, а также при запуске алгоритма с использованием большого числа параллельных процессов.

Теоретическая и практическая значимость. Диссертационная работа носит как теоретический, так и экспериментальный характер. Полученные теоретические результаты позволили разработать алгоритм поиска приближенных решений для нелинейного варианта задачи о p-медиане, имеющей естественное практическое приложение в региональной экономике и анализе данных. Реализованный в виде программы разработанный алгоритм может успешно применяться при решении практических задач большой размерности. Предложенные бикритериальные постановки дискретных задач размещения демонстрируют новый подход к поиску решений при условии неопределенных начальных данных задачи, который также может быть успешно применен и для других более сложных постановок дискретных

задач размещения, а также непосредственно для размещения объектов инфраструктуры при неопределенности величины спроса. Предложенный параллельный алгоритм поиска нижних оценок оптимального значения может служить основой для высокоэффективных параллельных алгоритмов поиска решений в задаче о p-медиане большой размерности, а также ее обобщений. Полученные с помощью программно реализованного параллельного алгоритма нижние оценки для задач большой размерности могут быть использованы как в точных методах, так и для оценки качества решения, получаемого с помощью современных эвристических и метаэвристических алгоритмов.

Исследования по теме диссертации проводились в рамках проектов по программе СО РАН «Нелокальные методы в теории управления динамическими системами» (№ гос. регистрации 01201001345), «Информационно-вычислительные технологии в системах поддержки принятия решений на основе оптимизационных моделей и методов» (№ гос. регистрации 01201351945), междисциплинарного интеграционного проекта СО РАН № 21, а также грантов РФФИ (проекты № 12-01-31198, № 12-07-33045, № 14-07-00382).

Результаты, выносимые на защиту:

1. Для нелинейного варианта задачи о p-медиане для двух видов релаксации получены явные формулы вычисления значений целевых функций двойственных задач и прямых переменных для некоторого фиксированного набора множителей. Разработан и протестирован алгоритм поиска приближенных решений, показавший свою эффективность для задач большой размерности.

2. Предложены новые бикритериальные нелинейные постановки задачи о p-медиане и простейшей задачи размещения с дополнительным критерием, максимизирующим робастность решения. Разработан, обоснован и протестирован алгоритм поиска приближенных Парето-оптимальных (δ-эффективных) решений в данных задачах.

3. Разработан и протестирован параллельный алгоритм поиска нижних оценок оптимального значения в классической задаче о p-медиане, включающий в себя процедуру каскадной сборки и специальную модель хранения данных и показавший эффективность для задачи о p-медиане большой размерности.

Структура и объем диссертации. Диссертация состоит из введения, трех глав, заключения и библиографии. Общий объем диссертации 151 страниц, из них 131 страница текста, включая 13 рисунков и 20 таблиц. Библиография включает 280 наименований на 19 страницах.

Во введении обосновывается актуальность диссертационной работы, формулируются цель и задачи исследования, аргументируется научная новизна, отражается практическая и теоретическая значимость полученных результатов, а также их апробация. Приводится обзор известных в литературе методов поиска решений в задаче о p-медиане, а также современного состояния научных исследований в данном направлении. Представляются выносимые на защиту научные положения.

Первая глава диссертационной работы носит по большей части обзорный характер. В ней представлено подробное описание так называемого метода релаксаций Лагранжа для задач целочисленного линейного программирования, а также представлен обзор алгоритмов поиска решений в двойственной задаче Лагранжа. Далее описан известный алгоритм поиска приближенных решений в задаче о p-медиане большой размерности, заложенные в котором идеи обобщаются во второй главе работы для одной нелинейной модификации задачи о p-медиане. Предложена схема распараллеливания процедуры поиска нижних оценок оптимального значения, эффективность которой продемонстрирована в ходе вычислительного эксперимента.

Во второй главе диссертации исследуется нелинейный вариант задачи о p-медиане, в котором количество открываемых предприятий p является целочисленной переменной, а в целевой функции присутствует дополнительное слагаемое в виде некоторой нелинейной функции, ограничивающей число открываемых предприятий. Для данной модификации задачи о p-медиане предложены и исследованы два вида релаксации Лагранжа, разработан алгоритм поиска приближенных решений, являющийся по сути обобщением метода, представленного в предыдущей главе диссертационной работы.

В третьей главе рассматривается один из подходов к определению робастности решения в непрерывных задачах размещения, известный также в литературе как пороговая робастность. Исследована возможность обобщения данной концепции для случая дискретных задач размещения на примере простейшей задачи и задачи о p-медиане. Рассмотрены бикритериальные варианты этих дискретных задач, где помимо основного присутствует дополнительный критерий на робастность искомого решения, дана их постановка в виде нелинейных бикритериальных задача целочисленного программирования. Предложен и обоснован метод поиска аппроксимации множества Парето-оптимальных решений.

Публикации. Материалы диссертации полностью опубликованы в 22 печатных работах, из них 5 статей в рецензируемых журналах из списка, рекомендованного ВАК для опубликования основных результатов диссертаций [7–9, 89, 90], одна статья в прочих журналах, две статьи в сборниках трудов конференций и 14 тезисов докладов.

Личный вклад автора. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в опубликованные работы. Идея исследования нелинейного варианта задачи о p-медиане [7, 89] и пороговой робастности в дискретных задачах размещения [8, 90] принадлежит Э. Каррисоса (E. Carrizosa) и И.Л. Васильеву. Подготовка к публикации полученных результатов проводилась совместно с соавторами, причем вклад диссертанта был определяющим. Все представленные в диссертации результаты получены лично автором. Разработка и программная реализация параллельного эвристического алгоритма поиска нижних оценок оптимального значения в задаче о p-медиане осуществлялась автором лично. Из совместных публикаций [9, 161] с И.Л. Васильевым, С. Ханафи (S. Hanafi) и К. Стерле (C. Sterle) на защиту выносятся только результаты, полученные автором лично.

Степень достоверности и апробация результатов. Достоверность результатов и выводов диссертации обусловлена применением апробированных методов и подходов исследования операций и математического программирования, а также современных технологий параллельных вычислений, и подтверждается результатами обширных вычислительных экспериментов.

Основные результаты диссертации докладывались и обсуждались на следующих российских и международных конференциях:

- IV Всероссийская научная конференция «Проблемы оптимизации и экономические приложения» (29 июня–4 июля 2009 г., Омск);
- Российская конференция «Дискретная оптимизация и исследование операций» (27 июня–3 июля 2010г., Алтай);
- Научная конференция «Ляпуновские чтения & презентация информационных технологий» (20–21 декабря 2010 г., Иркутск);
- XV Байкальской международная школа-семинар «Методы оптимизации и их приложения» (23–29 июня 2011 г., пос. Листвянка);
- Международная программа Ассоциации Европейских Обществ Исследования Операций для аспирантов ORP^3-2011 (13–17 сентября 2011 г., Кадис, Испания);
- Научная конференция «Ляпуновские чтения & презентация информационных технологий» (28–30 ноября 2011 г., Иркутск);

- XII Прибайкальская школа-семинар молодых ученых «Моделирование, оптимизация и информационные технологии» (19–24 марта 2012 г., Иркутск);
- V Всероссийская научная конференция «Проблемы оптимизации и экономические приложения» (2–6 июля 2012 г., Омск);
- Научная конференция «Ляпуновские чтения» (26–28 ноября 2012 г., Иркутск);
- XIV Российская конференция с международным участием «Распределенные информационные и вычислительные ресурсы» (26–30 ноября 2012 г., Новосибирск);
- Международная научная конференция «Дискретная оптимизация и исследование операций» (24–28 июня 2013 г., Новосибирск);
- III Всероссийская научная конференция «Математическое моделирование и вычислительно-информационные технологии в междисциплинарных научных исследованиях» (23–26 июня 2013 г., Иркутск);
- II Российско-монгольская конференция молодых ученых по математическому моделированию, вычислительно-информационным технологиям и управлению (25 июня–1 июля 2013 г., Иркутск (Россия) — Ханх(Монголия));
- 26-ая Европейская конференция по исследованию операций (1–4 июля 2013 г., Рим, Италия);
- XVI Байкальская международная школа-семинар «Методы оптимизации и их приложения» (30 июня–6 июля 2014 г., о. Ольхон);
- VI Международная научная конференция «Проблемы оптимизации и экономические приложения» (28 июня–4 июля 2015 г., Омск).

Кроме того, результаты диссертационной работы неоднократно обсуждались на научных семинарах в Институте динамики систем и теории управления имени В.М. Матросова СО РАН, Институте математики им. С.Л. Соболева СО РАН, а также на семинарах в Институте математики Севильского университета (Испания).

Глава 1

Задача о p-медиане и параллельный алгоритм поиска нижних оценок оптимального значения

Настоящая глава носит по большей части обзорный характер. В главе рассматривается содержательная постановка задачи о p-медиане, а также ее постановка в виде задачи целочисленного линейного программирования. Производится подробное описание так называемого метода релаксаций Лагранжа, приводятся некоторые общие свойства релаксаций Лагранжа, а также Лагранжевой двойственной функции для задач целочисленного линейного программирования. Дается обзор некоторых распространенных методов максимизации Лагранжевой двойственной функции с целью поиска нижних оценок, а в некоторых случаях, и оптимального значения. Приводится подробное описание одного известного подхода поиска приближенных решений в задаче о p-медиане, включающего в себя процедуры поиска последовательности нижних и верхних оценок оптимального значения. Поиск нижних оценок осуществляется на основании метода релаксаций Лагранжа, в то время как для поиска последовательности верхних оценок применяется так называемая ядровая эвристика (core heuristic), использующая информацию, полученную в ходе поиска нижней оценки. В конце главы предлагается схема распараллеливания процедуры поиска нижних оценок представленного эвристического алгоритма, приводится обзор некоторых необходимых теоретических результатов, а также современных средств и методов высокопроизводительных вычислений, в том числе для задач целочисленного программирования в общем и для задачи о p-медиане в частности. В конце главы приводятся результаты численного тестирования предложенного параллельного метода для ряда известных из литературы тестовых задач, а также искусственно сгенерированных примеров очень большой размерности.

1.1. Постановка задачи

Как было упомянуто ранее, задача о p-медиане представляет собой одну из базовых моделей в теории размещения и может быть сформулирована следующим образом. Пусть задано множество возможных пунктов размещения предприятий $I = \{1, \ldots, m\}$, множество клиентов $J = \{1, \ldots, n\}$, а также величины $d_{ij} > 0$, задающие транспортные затраты (или расстояния) на обслуживание j-го клиента из предприятия, размещенного в пункте i. Задача о p-медиане состоит в выборе p пунктов размещения предприятий из множества I так, что-

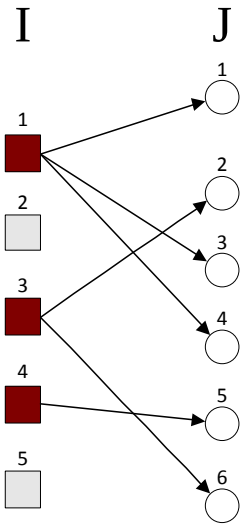

Рис. 1.1. Пример допустимого решения в задаче о p-медиане

бы суммарные затраты на обслуживание всех клиентов были минимальны. Таким образом, задача о p-медиане может быть представлена в следующем комбинаторном виде:

$$\min_{S \subseteq I} \Big\{ \sum_{j \in J} \min_{i \in S} d_{ij} : \ |S| = p \Big\}.$$

Отметим, что любое допустимое решение задачи о p-медиане может быть представлено на двудольном графе, в котором одно множество вершин соответствует возможным пунктам размещения предприятий I, а другое — множеству клиентов J, причем каждый клиент $j \in J$ соединен ребром, обладающим весом d_{ij}, только с одним из p открытых предприятий $i \in S$. На рис. 1.1 изображен пример допустимого решения задачи, в которой имеется пять возможных пунктов размещения предприятий и шесть клиентов. Как можно заметить, три предприятия открыты в пунктах 1, 3, 4, и каждый клиент обслуживается в точности одним из них.

Однако, в практических приложениях задачи часто предполагается, что множества возможных пунктов размещения предприятий и клиентов совпадают, т.е. $I = J$. Другими словами, предприятия должны быть размещены (открыты) среди клиентов. В таком случае задача о p-медиане может быть сформулирована на полном взвешенном простом орграфе с множеством узлов I и множеством дуг $A = \{(i, j) : i \in I, j \in I, i \neq j\}$ [34, 110, 113, 114, 126], причем каждой дуге $(i, j) \in A$ приписывается вес $d_{ij} > 0$, задающий расстояния между узлами. Задача о p-медиане в этом случае состоит в отыскании p вершин, называемых медианами, таких что сумма весов входящих дуг к немедианным вершинам от ближайшей медианы была минимальна. Отметим, что в такой постановке любое допустимое решение задачи состоит из

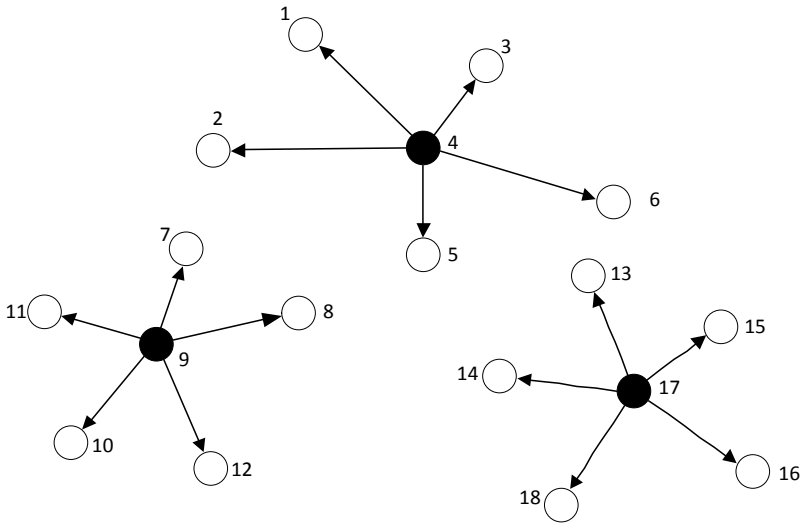

Рис. 1.2. Пример допустимого решения в задаче о p-медиане в случае $I = J$

p звезд, в центре которых находятся медианы (рис. 1.2).

Как и большинство задач комбинаторной оптимизации, задача о p-медиане может быть представлена в виде модели целочисленного линейного программирования. Классической и основной является модель, впервые предложенная Ривеллом и Свейном в 1970 г. [247]. В данной постановке вводятся две группы булевых переменных y_i и x_{ij}, $i \in I, j \in J$, соответствующих открываемым предприятиям и обслуживаемым из них клиентам:

$$y_i = \begin{cases} 1, & \text{если предприятие в пункте } i \text{ открыто;} \\ 0, & \text{в противном случае;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если клиент } j \text{ обслуживается из предприятия,} \\ & \text{размещенного в пункте } i; \\ 0, & \text{в противном случае.} \end{cases}$$

С использованием введенных переменных задача о p-медиане может быть записана в следу-

ющем виде:

$$\min_{(x,y)} \sum_{i\in I} \sum_{j\in J} d_{ij} x_{ij}, \tag{1.1}$$

$$\sum_{i\in I} x_{ij} = 1, \qquad j \in J, \tag{1.2}$$

$$x_{ij} \le y_i, \qquad i \in I, j \in J, \tag{1.3}$$

$$\sum_{i\in I} y_i = p, \tag{1.4}$$

$$y_i \in \{0,1\}, \qquad i \in I, \tag{1.5}$$

$$x_{ij} \in \{0,1\}, \qquad i \in I, j \in J. \tag{1.6}$$

В данной постановке целевая функция (1.1) минимизирует суммарные затраты на обслуживание всех клиентов. Ограничения (1.2) гарантируют, что каждый клиент обслуживается только из одного предприятия. Ограничения (1.3) гарантируют, что каждый клиент j обслуживается только из открытых предприятий. Количество открываемых предприятий задается ограничением (1.4). Условия на целочисленность переменных определяются ограничениями (1.5), (1.6).

Отметим, что ограничения (1.3), часто называемые ограничениями Балинского, являются более сильной версией ограничений, использующихся также для простейшей задачи размещения в 50–60-х годах [3, 251]:

$$\sum_{j\in J} x_{ij} \le n y_i, \qquad i \in I, \tag{1.7}$$

и задающих связь между переменными y_i и x_{ij}. Несмотря на то, что количество таких связывающих ограничений составляет m (число возможных пунктов размещения предприятий) и является существенно меньшим числа mn ограничений (1.3), линейная релаксация задачи, полученная заменой ограничений (1.3) на (1.7) и ослаблением условия на целочисленность переменных дает более слабую (меньшую) нижнюю оценку оптимального значения в задаче о p-медиане, что является существенным недостатком при разработке численных методов решения [45, 101, 110, 229, 275].

Постановка задачи о p-медиане в виде (1.1)–(1.6) оставалась основной при разработке методов решения, как точных, так и эвристических, на протяжении более тридцати лет.

В настоящей главе задача о p-медиане рассматривается в формулировке, описанной в работах [58, 60] и состоящей в следующем. Предположим, что имеется полный взвешенный простой орграф $G(I, A)$, $|I| = m$. Обозначим через $\delta^-(j) = \{i \in I |\ (i, j) \in A\}$ множество

узлов графа G, смежных с j, а через $\delta^+(i) = \{j \in I |\ (i,j) \in A\}$ множество узлов, с которыми смежна i. Также введем булевы переменные y_i и x_{ij}, соответствующие вершинам и дугам графа $G(I, A)$. Переменная y_i принимает значение 1, если узел i является медианой, 0 в противном случае. Переменная x_{ij} равна 1, если узел i представляет собой медиану, и j присоединен к i выходящей из i дугой, 0 в противном случае. Используя введенные переменные и обозначения, задача о p-медиане может быть сформулирована как следующая задача целочисленного линейного программирования:

$$\min_{(x,y)} \sum_{(i,j)\in A} d_{ij}x_{ij}, \tag{1.8}$$

$$\sum_{i\in\delta^-(j)} x_{ij} + y_j = 1, \qquad j \in I, \tag{1.9}$$

$$x_{ij} \le y_i, \qquad i \in I, j \in \delta^+(i), \tag{1.10}$$

$$\sum_{i\in I} y_i = p, \tag{1.11}$$

$$y_i \in \{0,1\}, \qquad i \in I, \tag{1.12}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A. \tag{1.13}$$

Целевая функция (1.8) минимизирует сумму весов дуг $(i,j) \in A$ от вершин до ближайших медиан. Ограничения (1.9) гарантируют, что каждая вершина i является либо медианой, либо имеет одну входящую дугу из медианной вершины. Неравенства (1.10) исключают существование выходящих дуг из немедианных вершин. Количество искомых медиан определяется уравнением (1.11).

Несмотря на то, что задача о p-медиане является довольно хорошо изученным объектом в теории размещения, и за пятьдесят лет исследований было опубликовано порядка пятисот работ, посвященных изучению ее свойств и разработке эффективных подходов и алгоритмов ее решения, на сегодняшний день поиск решений для задач большой размерности представляет собой существенную трудность. К настоящему времени в литературе не представлены точные или эвристические алгоритмы, как последовательные, так и параллельные, способные успешно находить решения для задач, размерность которых превосходила бы 100000 узлов (возможных пунктов размещения предприятий и клиентов), что вызвано прежде всего сложностью задачи, а также наличием технических ограничений вычислительных средств, в особенности недостаточным количеством оперативной памяти, поскольку в процессе работы алгоритмам необходимо хранить огромный объем информации. Наличие таких трудностей побуждает к поиску новых подходов и методов хранения и обработки больших массивов дан-

ных задачи, таких как агрегирование [137, 177, 178] и специальные модели хранения данных, позволяющие алгоритму оперировать ими в неявном виде [56].

1.2. Один метод поиска приближенных решений в задаче о p-медиане

В данном разделе приводится описание одного эвристического подхода для поиска приближенных решений в задаче о p-медиане большой размерности, являющегося вариантом широко известных лагранжевых эвристик, в основе которого лежит метод релаксаций Лагранжа для поиска последовательности нижних оценок оптимального значения задачи, а также так называемая ядровая эвристика (core heuristic) для поиска последовательности верхних оценок. В рамках настоящей главы для процедуры поиска нижних оценок оптимального значения будет предложена схема распараллеливания, в то время как аналогичный по общей схеме метод будет предложен в следующей главе для одного обобщения задачи о p-медиане. Рассматриваемый алгоритм был представлен в работе [56] и включает в себя следующие основные элементы:

1. *Метод релаксаций Лагранжа и специальный метод генерации столбцов.* Метод генерации столбцов является хорошо известным подходом для решения задач линейного программирования большой размерности [45, 74, 75, 101, 108, 275]. В основе метода лежит идея поиска решения только на некотором подмножестве переменных и динамического добавления нерассмотренных переменных с минимальными отрицательными (в случае задачи минимизации) относительными оценками. В общем случае такой подход предполагает декомпозицию исходной задачи на так называемую координирующую задачу и специального вида подзадачу для поиска нового базисного столбца. Отметим, что в случае задачи линейного и целочисленного программирования для построения координирующей задачи применяется так называемое разложение Дантцига-Вульфа [109]. Для решения получившейся декомпозиционной задачи используется модифицированный симплекс-метод с динамическим поиском нового базисного столбца для случая задачи линейного программирования или так называемый метод ветвей и оценок [65] в том случае, если переменные задачи являются целыми.

   В описываемом эвристическом подходе используется комбинированный метод поиска последовательности нижних оценок оптимального значения, включающий в себя построение и решение двойственной задачи Лагранжа с помощью эвристического субгра-

диентного алгоритма, а также использование специального метода генерации столбцов, осуществляющего динамическое добавление переменных, относительная оценка которых отрицательна. Применение такого комбинированного подхода предполагает приведение матрицы расстояний к виду, позволяющему избежать хранения в памяти ЭВМ больших массивов данных и существенно ускорить время работы алгоритма для задач большой размерности.

2. *Ядровая эвристика.* Для поиска последовательности верхних оценок оптимального значения применяется так называемая ядровая эвристика (core heuristic), активно используемая в работах [56, 57, 59, 60, 87, 212] для поиска близких к оптимальным допустимых решений в различных вариантах дискретных задач размещения. Суть метода состоит в решении исходной задачи только на некотором «перспективном» подмножестве переменных, выбор которых происходит на основе информации и относительных оценок, полученных в ходе решения двойственной задачи Лагранжа с помощью субградиентного алгоритма.

   Отметим, что сочетание метода релаксаций Лагранжа, специального метода генерации столбцов и ядровой эвристики оказывается чрезвычайно эффективным для поиска приближенных решений в задачах большой размерности с относительно большим количеством медиан p.

1.2.1. Релаксация Лагранжа. Общие понятия

В начале рассмотрим первый элемент представленного эвристического алгоритма, применяемый для поиска последовательности нижних оценок оптимального значения — метод релаксаций Лагранжа. Сама по себе релаксация является одним из наиболее важных и известных подходов для поиска так называемых двойственных (нижних в задаче на минимум) оценок оптимального значения в задачах целочисленного линейного программирования и комбинаторной оптимизации [101, 139, 196, 197, 229, 257, 275]. Ее идея состоит в редукции исходной задачи к более простой оптимизационной задаче, оптимальное значение которой, в случае поиска минимума, не превосходило бы оптимальное значение в исходной задаче. Для построения подобной релаксации имеются два очевидных способа: замена исходной целевой функции задачи минимизации другой, принимающей всюду на допустимом множестве аналогичные или меньшие значения, и расширение допустимого множества задачи.

Определение 1. [143] Задача $f(x) \downarrow \min$, $x \in Y \subseteq \mathbb{R}^n$, называется релаксацией задачи $c(x) \downarrow \min$, $x \in X \subseteq \mathbb{R}^n$, если:

1. $X \subseteq Y$;

2. $f(x) \leq c(x)\ \forall x \in X$.

Одной из наиболее важных и естественных является линейная (непрерывная) релаксация, получающаяся из исходной задачи целочисленного линейного программирования путем исключения ограничений на целочисленность переменных. Однако такой подход имеет ряд недостатков, связанных прежде всего с трудностью поиска решения в релаксированной задаче, особенно с ростом размерности. Наряду с линейной релаксацией в 70-х годах XX века был предложен еще один подход к построению эффективных релаксаций, так называемая релаксация Лагранжа, истоки и корни которой могут быть прослежены в работах [176, 203]. Впервые на практике данный подход был успешно применен к одной из задач комбинаторной оптимизации большой размерности — симметричной задаче о коммивояжере в статьях [170, 171]. Идея релаксации Лагранжа состоит в разделении набора ограничений исходной задачи на две группы: простых и сложных. Ограничения считаются сложными в том смысле, что несодержащая их исходная задача оптимизации является более простой для решения (за полиномиальное время). Отметим также, что удаление из постановки группы сложных ограничений позволяет также получить релаксацию исходной задачи за счет расширения допустимого множества, однако двойственная оценка оптимального значения задачи (нижняя в случае задачи минимизации и верхняя для задач на максимум), получаемая таким образом, является, по понятным причинам, довольно грубой, что делает применение такого подхода малоцелесообразным. Вместо этого релаксация Лагранжа предполагает добавление набора сложных ограничений в целевую функцию в виде штрафа с некоторыми весами, называемыми множителями Лагранжа (двойственными переменными). Оптимальное значение релаксированной задачи, полученной с помощью такого преобразования, дает двойственную оценку оптимального значения в исходной задаче [143, 151, 232, 257].

Предположим, что имеется задача целочисленного линейного программирования в следующей форме

$$\min\{\langle c, x\rangle : Ax \geq b, Dx \geq d, x \in \mathbb{Z}^n_+\}, \qquad (IP)$$

где $A \in \mathbb{M}^{h\times n}$, $D \in \mathbb{M}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^h$, $d \in \mathbb{R}^m$, $x \in \mathbb{Z}^n_+$. Ограничения $Ax \geq b$ будем считать простыми, а $Dx \geq d$ — сложными, т.е. исключение которых из рассмотрения делает задачу (IP) простой для решения. Например, после удаления сложных ограничений задача может быть представлена как серия независимых подзадач, решение которых может быть найдено за полиномиальное время с помощью известных алгоритмов. В качестве примера следует рассмотреть задачу о коммивояжере, в которой ослабление ограничений исключаю-

щих подмаршруты или подциклы, число которых, вообще говоря, растет экспоненциально с ростом числа городов, приводит к известной задаче о назначениях, решение которой может быть найдено за полиномиальное время. Другим интересным примером является простейшая задача размещения (задача размещения без ограничения на мощность производства), в которой ослабление условий, гарантирующих обслуживание клиентов только из одного предприятия, сводит задачу к серии независимых подзадач для каждого отдельного возможного пункта размещения, решение которых может быть найдено аналитически для всякого заданного набора множителей Лагранжа [101, 138, 150, 275].

Предположим, что задача (IP) является допустимой, тогда релаксацией Лагранжа задачи (IP) относительно ограничений $Dx \geq d$ называют следующую задачу:

$$\theta(\lambda) = \min\{\langle c, x\rangle + \lambda(d - Dx) : Ax \geq b, x \in \mathbb{Z}^n_+\}, \qquad (IP_\lambda)$$

где $\lambda \in \mathbb{R}^m_+$ — двойственные переменные задачи.

Отметим, что функцию $\theta(\lambda)$ называют двойственной функцией Лагранжа. При любом фиксированном наборе множителей $\lambda = (\lambda_1, \ldots, \lambda_m)$ значение двойственной функции дает нижнюю оценку оптимального значения $v(IP)$ исходной задачи (IP), т.е. $\theta(\lambda) \leq v(IP)$ [130, 131, 143, 151, 232].

Для поиска наилучшей нижней оценки, т.е. для выбора набора множителей $\lambda \in \mathbb{R}^m_+$, доставляющего максимальное значение функции $\theta(\lambda)$, строится задача:

$$\max\{\theta(\lambda) : \ \lambda \in \mathbb{R}^m_+\}, \qquad (D)$$

называемая двойственной задачей Лагранжа к (IP).

Отметим, что двойственная задача Лагранжа обладает рядом свойств, наличие которых позволяет применять достаточно простые и эффективные алгоритмы поиска в них оптимального решения [101, 130, 138, 153, 229, 275]. Предположим, что множество $X' = \{x \in \mathbb{Z}^n_+ : \ Ax \geq b\}$ содержит большое, но конечное число точек, т.е. может быть представлено как $X' = \{x^t : \ t = 1, \ldots, T\}$. Тогда задача (D) может быть записана в следующем виде:

$$v(D) = \max_{\lambda \geq 0} \theta(\lambda) = \max_{\lambda \geq 0} \min_{t=1,\ldots,T} \{\langle c, x^t\rangle + \lambda(d - Dx^t)\},$$

что в свою очередь может быть представлено как

$$\begin{aligned} & v(D) = \max \zeta, \\ & \zeta \leq \langle c, x^t\rangle + \lambda(d - Dx^t), \ t = 1, \ldots, T, \\ & \lambda \in \mathbb{R}^m_+, \zeta \in \mathbb{R}^1. \end{aligned} \qquad (1.14)$$

Отсюда можно заметить, что двойственная функция Лагранжа представляет собой верхнюю огибающую конечного семейства линейных функций, а следовательно является вогнутой [4], однако недифференцируемой в конечном множестве точек $\bar{\lambda} \in \mathbb{R}^m_+$, в которых задача $(IP_{\bar{\lambda}})$ имеет неединственное решение [131]. Данное свойство, несмотря на вогнутость и непрерывность, препятствует применению богатого арсенала методов выпуклой дифференцируемой оптимизации для поиска решения в двойственной задаче (D) и требует использования алгоритмов, основанных на обобщенном понятии градиента функции, о которых речь пойдет ниже. Однако, прежде чем перейти к описанию методов поиска решений в двойственной задаче Лагранжа, остановимся на вопросе о том, насколько сильна двойственная оценка, получаемая с помощью релаксации Лагранжа, а также о геометрической интерпретации Лагранжевой двойственной задачи.

В работе [143] было показано, что задача

$$\min\{\langle c, x\rangle : Dx \geq d, x \in \operatorname{conv}(Ax \geq b, x \in \mathrm{Z}^n_+)\}, \qquad (P^*)$$

где $\operatorname{conv}(\cdot)$ — выпуклая оболочка множества, эквивалентна задаче (D) в том смысле, что $v(D) = v(P^*)$. Справедливость этого утверждения, в частности, может быть показана, если построить двойственную к (1.14) задачу, имеющую следующий вид [130]:

$$\begin{aligned} \min & \sum_{t=1}^{T} \mu_t \langle c, x^t\rangle, \\ & \sum_{t=1}^{T} \mu_t (Dx^t - d) \geq 0, \\ & \sum_{t=1}^{T} \mu_t = 1, \\ & \mu_t \geq 0,\ t = 1, \ldots, T. \end{aligned} \qquad (1.15)$$

Вводя новые обозначения $x = \sum\limits_{t=1}^{T} \mu_t x^t$ и учитывая условия на двойственные переменные $\sum\limits_{t=1}^{T} \mu_t = 1$, задача (1.15) может быть переписана как (P^*).

Отметим, что из вида задачи (P^*) можно непосредственно сделать вывод, что $v(\overline{IP}) \leq v(P^*) \leq v(IP)$, где $(\overline{IP})$ представляет собой линейную релаксацию задачи (IP), а следовательно $v(\overline{IP}) \leq v(D)$. Таким образом, оценка оптимального значения, получаемая с помощью релаксации Лагранжа, всегда является по крайней мере не хуже оценки, подсчитанной с помощью линейной релаксации. Из представленной геометрической интерпретации двойственной задачи Лагранжа относительно переменных прямой задачи x можно также выделить достаточное условие, при котором $v(D) = v(\overline{IP})$.

Определение 2. [143] Говорят, что имеет место свойство целочисленности, если оптимальное значение задачи (IP_λ) не изменяется при отбрасывании условия на целочисленность переменных, т.е. $v(IP_\lambda) = v(\overline{IP_\lambda})$, где

$$v(\overline{IP_\lambda}) = \min_{x \in \mathbb{R}^n_+} \{\langle c, x\rangle + \lambda(d - Dx) : Ax \geq b\}.$$

Таким образом, если выполняется свойство целочисленности, то все угловые точки допустимого множества $\{Ax \geq b, x \in \mathbb{R}^n_+\}$ задачи $\overline{IP}_\lambda$ являются целочисленными, т.е.

$$\{Ax \geq b, x \in \mathbb{R}^n_+\} = \text{conv}\{Ax \geq b, x \in \mathbb{Z}^n_+\},$$

а следовательно двойственная задача Лагранжа не дает нижнюю оценку, лучшую чем получаемая с помощью линейной релаксации, т.е. $v(\overline{IP}) = v(P^*) = v(D)$. Такое свойство в некоторых ситуациях является довольно полезным, поскольку поиск решения в линейной релаксации, особенно для задач большой и сверхбольшой размерности, является сложной вычислительной задачей, например, при наличии экспоненциального числа ограничений, которые могут быть ослаблены с целью получения более простой оптимизационной задачи, а в большинстве случаев серии простых для решения подзадач. Примером задачи, обладающей подобной структурой, может быть симметричная задача о коммивояжере, содержащая экспоненциальное число ограничений, исключающих наличие подмаршрутов. Ослабление данной группы ограничений позволяет свести двойственную задачу Лагранжа к серии подзадач, являющихся так называемыми задачами поиска 1-дерева минимального веса [101, 151, 170, 171, 275]. С другой стороны, если не имеет место свойство целочисленности, то вполне возможно построение Лагранжевых релаксаций, дающих лучшую нижнюю оценку, чем получаемую с помощью линейной релаксации.

1.2.2. Релаксация Лагранжа для задачи о p-медиане

Считается, что в теории размещения впервые идеи Лагранжевой релаксации были применены при решении простейшей задачи размещения в работе [77], являющейся английским переводом ранее написанной статьи [139]. Для рассматриваемой задачи о p-медиане существуют разные подходы к выбору релаксации [163], однако наиболее популярный и естественный тип релаксации исходной задачи предполагает ослабление ограничений (1.9) [5, 55, 102]. Данные ограничения добавляются в целевую функцию с весами $\lambda \in \mathbb{R}^m$:

$$\theta(\lambda) = \min_{(x,y)} \left\{ \sum_{(i,j)\in A} d_{ij}x_{ij} - \sum_{j\in I} \lambda_j \left(\sum_{i\in\delta^-(j)} x_{ij} + y_j - 1 \right) : \text{при (1.10)-(1.13)} \right\} \qquad (PM_\lambda).$$

Выбор такого набора «сложных» ограничений позволяет существенно упростить релаксированную задачу и свести ее таким образом к серии подзадач, решение которых может быть найдено аналитически [101, 130, 275]. Чтобы показать это, перепишем последнее равенство в следующем виде

$$\theta(\lambda) = \min_y \Big\{ \min_x \Big\{ \sum_{i\in I} \sum_{j\in\delta^+(i)} (d_{ij} - \lambda_j) x_{ij} : \; 0 \le x_{ij} \le y_i \Big\} - \\ - \sum_{i\in I} \lambda_i y_i : \; \sum_{i\in I} y_i = p \Big\} + \sum_{j\in I} \lambda_j.$$

Предположим, что переменные y_i, $i \in I$, фиксированы, а x_{ij} являются непрерывными на отрезке $[0, 1]$. Выпишем теперь отдельно получившуюся задачу минимизации по x:

$$\min_x \Big\{ \sum_{i\in I} \sum_{j\in\delta^+(i)} (d_{ij} - \lambda_j) x_{ij} : \; 0 \le x_{ij} \le y_i \Big\}.$$

Как нетрудно видеть, в полученной задаче необходимо найти минимум сепарабельной относительно x функции при параллелепипедных ограничениях. Решение такой задачи может быть с легкостью записано следующим образом:

$$x_{ij}(\lambda) = \begin{cases} y_i, & d_{ij} - \lambda_j < 0 \\ 0, & d_{ij} - \lambda_j \ge 0. \end{cases}$$

Подставим полученное решение x_{ij} в исходную задачу. Отметим, что при подстановке решения удобно воспользоваться определением срезки: $a^- = \min\{0, a\}$, в связи с чем получаем следующую задачу:

$$\theta(\lambda) = \min_y \Big\{ \sum_{i\in I} \sum_{j\in\delta^+(i)} (d_{ij} - \lambda_j)^- y_i - \sum_{i\in I} \lambda_i y_i : \; \sum_{i\in I} y_i = p; \; 0 \le y_i \le 1 \Big\} + \sum_{j\in I} \lambda_j =$$

$$= \min_y \Big\{ \sum_{i\in I} \sum_{j\in\delta^+(i)} \left((d_{ij} - \lambda_j)^- - \lambda_i \right) y_i : \; \sum_{i\in I} y_i = p; \; 0 \le y_i \le 1 \Big\} + \sum_{j\in I} \lambda_j.$$

Далее, для каждого узла $i \in I$ вводятся величины

$$\rho_i(\lambda) = \sum_{j\in\delta^+(i)} (d_{ij} - \lambda_j)^- - \lambda_i,$$

называемые относительными оценками или оценками Лагранжа, и предполагается, что они упорядочены в неубывающем порядке, т.е.

$$\rho_{i_1}(\lambda) \le \cdots \le \rho_{i_m}(\lambda).$$

Учитывая введенные обозначения, рассмотрим следующую задачу минимизации:

$$\min_{y}\Big\{\sum_{i\in I}\rho_i(\lambda)\,y_i:\ \sum_{i\in I}y_i=p;\ 0\le y_i\le 1\Big\}.$$

В данной задаче необходимо минимизировать линейную функцию относительно одного линейного ограничения. Нетрудно видеть, что решением такой задачи будет вектор $y(\lambda)$, в котором отличными от нуля будут только p координат, соответствующие наименьшим оценкам Лагранжа. Таким образом, значение двойственной функции может быть подсчитано в явном виде по следующей формуле:

$$\theta(\lambda)=\sum_{i\in T(\lambda)}\rho_i(\lambda)+\sum_{i\in I}\lambda_i, \tag{1.16}$$

где $T(\lambda)$ — множество, состоящее из p узлов с минимальной оценкой Лагранжа. Более того, оптимальное решение $(x_{ij}(\lambda), y_i(\lambda))$ в задаче (PM_λ) может быть записано как

$$y_i(\lambda)=\begin{cases}1, & i\in T(\lambda);\\ 0, & \text{в противном случае.}\end{cases}$$

$$x_{ij}(\lambda)=\begin{cases}1, & y_i=1 \text{ и } d_{ij}-\lambda_j<0;\\ 0, & \text{в противном случае.}\end{cases}$$

Из полученного таким образом решения можно заметить, что для построенной релаксации имеет место свойство целочисленности, т.е. $v(IP_\lambda)=v(\overline{IP}_\lambda)$. Из чего можно заключить, что оптимальное значение двойственной задачи $\max\limits_{\lambda\in\mathbb{R}^m}\theta(\lambda)$ дает нижнюю оценку оптимального значения в исходной задаче о p-медиане, равную полученной с помощью линейной релаксации.

1.2.3. Методы решения Лагранжевой двойственной задачи. Субградиентные алгоритмы

Как было отмечено, двойственная функция Лагранжа $\theta(\lambda)$ является негладкой вогнутой функцией. К тому же, поскольку допустимое множество в релаксированной задаче (PM_λ) является дискретным, то $\theta(\lambda)$ представляет собой конечнопорожденную полиэдральную функцию [33, 138], т.е. ее надграфик есть многогранное множество:

$$\mathrm{epi}(\theta)=\{(u,\lambda):\ u\le\theta(\lambda)\}.$$

Другими словами, каждому решению $(\bar{x},\bar{y})$ задачи (PM_λ) для некоторого заданного $\lambda\in\mathbb{R}^m$ соответствует невертикальная опорная гиперплоскость $\sum\limits_{i\in\delta^-(j)}x_{ij}+y_j-1$ к надграфику $\mathrm{epi}(\theta)$

в точке $(\lambda, \theta(\lambda))$, являющаяся субградиентом (или точнее суперградиентом) функции θ в точке λ [33].

Для поиска решения в двойственной задаче используются различные методы, среди которых стоит выделить прямо-двойственные методы спуска, являющиеся адаптацией прямо-двойственного симплекс-метода для прямой задачи, а также методы Лагранжевого двойственного спуска, представляющие собой специализированные процедуры, использующие информацию о структуре исходной задачи для поиска оптимального решения в двойственной. На каждой итерации исследуется некоторое подмножество двойственных переменных и подсчитывается, каким образом изменение значений одного или нескольких множителей влияет на значение двойственной функции Лагранжа, так чтобы обеспечить ее монотонное убывание. Недостатком подхода является его зависимость от структуры задачи и, следовательно, невозможность его применения для решения Лагранжевой двойственной задачи в общем случае, а также трудность его применения для задач большой размерности. Отметим однако, что в некоторых случаях такой алгоритм совершает намного меньше итераций, чем субградиентный метод [150, 152, 153]. Методы Лагранжевого двойственного спуска также были успешно применены и для решения других задач дискретной и комбинаторной оптимизации [133, 150, 152], в том числе возникающих на практике [132].

С другой стороны, поскольку двойственная функция Лагранжа является негладкой, то для ее максимизации возможно также применение богатого арсенала универсальных методов недифференцируемой оптимизации, которые можно условно разделить на субградиентные алгоритмы и методы отсечений [196]. Так, для поиска решения именно в двойственной задаче Лагранжа широкое применение нашли различные варианты метода Келли [183](см. например [116, 145]), а также методы генерации строк или столбцов [138, 151]. Недостатком данных подходов безусловно является очень низкая скорость сходимости, что вызвано прежде всего слабой аппроксимацией двойственной функции Лагранжа из-за малого количества отсечений (столбцов), а следовательно, хороших значений двойственных переменных, хотя для некоторых семейств задач алгоритм генерации столбцов имеет довольно высокую скорость сходимости [151]. Несмотря на это, наибольшее распространение получили методы поиска минимума в двойственной задаче, основанные на идее обобщенного градиента. Напомним формальное определение субградиента функции [6, 21, 33, 101, 275].

Определение 3. Субградиентом выпуклой функции $f : \mathbb{R}^m \mapsto \mathbb{R}^1$ в точке $y \in \mathbb{R}^m$ называется вектор $g(y) \in \mathbb{R}^m$, такой что

$$f(x) \geq f(y) + \langle g(y), x - y \rangle \qquad \forall x \in \mathbb{R}^m.$$

Отметим, что множество всех субградиентов функции $f(x)$ в точке y называется субдифференциалом функции в точке y и обозначается $\partial f(y)$. Известно, что субдифференциал — компактное выпуклое множество.

Субградиентный алгоритм для максимизации некоторой вогнутой недифференцируемой функции $f(x)$, заданной на выпуклом множестве $X \subset \mathbb{R}^n$, представляет собой процедуру, генерирующую последовательность точек $\{x^k\}$, начиная с некоторого начального значения x^0, вдоль направления субградиента по следующему рекуррентному правилу:

$$x^{k+1} = P_X\Big(x^k + \alpha_k \frac{g^k}{\|g^k\|}\Big), \tag{1.17}$$

где α_k — размер шага метода вдоль направления субградиента $g^k \in \partial f(x^k)$ в точке x^k.

Основоположником методов субградиентной оптимизации по праву считается советский математик Н.З. Шор, впервые предложивший идею данного подхода и исследовавший его сходимость [35]. В ранних публикациях такие алгоритмы назывались методами обобщенного градиента. В своей кандидатской диссертации Н.З. Шором была доказана сходимость метода (1.17) для задачи без ограничений в конечномерном гильбертовом пространстве при выборе шага $\alpha_k = \beta$. Было доказано, что в этом случае можно найти минимум с точностью до величины порядка β.

В западной литературе часто отмечают, что корни методов субградиентной оптимизации восходят к работам С. Агмона [46], Т. Моцкина и И.Я. Шёнберга [223], а также И.И. Еремина [14–16], посвященных методу релаксаций для решения систем линейных алгебраических уравнений, а также его обобщению для поиска оптимальных в смысле Чебышева решений в несовместных системах линейных и нелинейных алгебраических уравнений. Однако непосредственно алгоритм был (независимо от Н.З. Шора) изобретен Майклом Хэлдом и Ричардом М. Карпом в их известной работе [171], посвященной разработке метода решения задачи о коммивояжере. Одними из пионерских работ, посвященных исследованию сходимости метода и правил выбора шага, являются работы Ю.М. Ермольева [18] и Б.Т. Поляка [31, 32], В.Ф. Демьянова [13], И.И. Еремина [17], Н.З. Шора [36, 40], а также М. Хэлда, Ф. Вульфа [172]. В последней работе также представлены одни из первых результатов обширного вычислительного эксперимента для различных задач дискретной и комбинаторной оптимизации: задачи о назначениях, нецелочисленной симметричной задачи о коммивояжере, а также многопродуктовой задачи о максимальном потоке.

Из литературы известны несколько различных вариантов правила выбора шага метода α_k [52, 144, 238, 260], среди которых можно выделить три наиболее популярных [39, 52, 101,

144, 275]:

$$\alpha_k \to 0, \ \sum_{k=0}^{\infty} \alpha_k = \infty \ [18, \ 31, \ 36]; \tag{1.18}$$

$$\alpha_k = C\beta_k, \quad C > 0, \ 0 < \beta < 1 \ [18] \ (\alpha_k > 0 \ [39, \ 171]); \tag{1.19}$$

$$\alpha_k = \frac{\gamma_k (f^* - f(x^k))}{\|g^k\|}, \quad 0 < \gamma_k < 2, \tag{1.20}$$

где f^* — оптимальное значение в задаче максимизации функции $f(x)$, $x \in X$ (предполагается конечным и достижимым).

Как было показано в [31, 52, 238, 261], условия выбора шага (1.18) обеспечивают довольно слабую сходимость метода. Обозначим через X^* множество оптимальных решений двойственной задачи Лагранжа, тогда $\lim\limits_{k\to\infty} \sup f(x^k) = f^*$. Более того, если X^* является компактным множеством, то $d(x^k, X^*) \to 0$, где $d(x, X^*) = \min\limits_{x^* \in X^*} \|x - x^*\|$. Однако правило выбора шага (1.18) можно усилить, дополнительно наложив условие, что $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, тогда $x^k \to x^* \in X^*$ [19, 52, 103]. При определенных дополнительных условиях можно получить сходимость алгоритма со скоростью геометрической прогрессии [32, 39].

В литературе для правила выбора шага (1.20) помимо оптимального значения f^* [17, 46, 223, 233], которое заранее не может быть найдено и посему неизвестно, исследовались также варианты, где f^* заменялась на некоторую оценку сверху $\bar{f} > f^*$ [15, 172] или некоторую достаточно близкую оценку снизу $\underline{f} < f^*$ [32, 39], поиск которой, вообще говоря, является нетривиальным. К тому же при выборе $\underline{f} < f^*$ на каждой итерации k для некоторого фиксированного $\varepsilon > 0$: $\varepsilon < \gamma_k \leq 2$ последовательность $\{f(x^k)\}$, вообще говоря, сходится или к $\underline{v}$, или к некоторой точке $\hat{x}$: $f(\hat{x}) \geq \underline{f}$ [39, 172], что является малоприменимым на практике. Следует также подчеркнуть, что методы обобщенного градиентного спуска при выборе шага (1.20) являются по сути алгоритмами фейеровских приближений и практически прямым применением процедуры Агмона-Моцкина-Шёнберга по поиску точки x, такой что $f(x) \geq \underline{f}$ [39, 172]. Стоит выделить также эвристические процедуры улучшения для правила выбора шага (1.20), предложенные в [67] и позволяющие повысить вычислительную эффективность субградиентного метода.

На практике часто применяют ненормализованный вариант правила выбора шага (1.18), (1.19), т.е.

$$\lambda^{k+1} = P_X\Big(\lambda^k + \alpha_k g(\lambda^k)\Big), \tag{1.21}$$

Использование такого рекуррентного правила, нередко применяемого на практике вместо (1.17), дает более слабые условия сходимости субградиентного алгоритма. Так, например, для простого выбора шага $\alpha_k = 1/k$ имеются примеры, когда метод, вообще говоря,

не сходится [52, 260]. Однако наложение дополнительного условия $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ помогает несколько исправить ситуацию, и тогда для последовательности (1.21) с шагом (1.18) справедливо $x^k \to x^*$ в случае, если $\{g^k\}$ — ограниченная последовательность.

Помимо непосредственно субградиентного алгоритма с правилами выбора шага (1.18)--(1.20) широко применяются также и другие, усовершенствованные методы, основанные на идее обобщенного градиента. Среди первых следует выделить субградиентные методы с растяжением пространства в направлении субградиента, разработка которых связана с анализом алгоритмов обобщенного субградиентного спуска, сходящихся со скоростью геометрической прогрессии. Поскольку такие методы неприменимы в случае овражных задач, в [37, 38] предложено использовать линейные неортогональные преобразования пространства аргументов для улучшения обусловленности задачи. Заметим, что частным случаем таких алгоритмов является предложенный в [44] метод эллипсоидов. Следующим развитием данного подхода стали субградиентные методы с растяжением пространства в направлении разности двух последовательных субградиентов — так называемые r-алгоритмы [41], особенно эффективные при решении сложных задач недифференцируемой оптимизации средней размерности, т.е. содержащих несколько сотен переменных [39, 42]. Свое развитие также получили методы сопряженных субградиентов (ε-субградиентные алгоритмы [39]), в некотором смысле формально обобщающие метод сопряженных градиентов [194, 274], также положившие начало развитию следующего поколения методов.

Среди современных популярных алгоритмов стоит прежде всего выделить различные варианты так называемых методов пучка субградиентов (bundle methods) [80, 174, 195, 198, 217, 234], основанные на использование информации о субградиентах, полученной на предыдущих итерациях метода. Алгоритм фактически начинает свою работу с некоторого набора отсекающих плоскостей, а также с выбора так называемого центра устойчивости, являющегося по сути некоторой аппроксимацией оптимального решения. Каждая итерация метода является своего рода компромиссом между получением улучшения значения целевой функции и близости к центру устойчивости. Итерации алгоритма делятся на существенные, т.е. те, в которых происходит изменение центра устойчивости, и нулевые, в которых происходит улучшение имеющейся аппроксимации целевой функции. С методами пучка субградиентов связаны также получившие распространение в области дискретной оптимизации «объемные» алгоритмы (volume algorithms), впервые предложенные в [64]. Анализ метода и его связь с методами пучка субградиентов, а также некоторое улучшение представлены в работе [62]. Отметим, что метод схож с субградиентными методами, однако его отличие состоит прежде всего в выборе шага. Так, в «объемном» алгоритме имеется три вида итераций: зеленые, жел-

тые и красные, в которых шаг метода зависит от того, насколько, и улучшилось ли вообще, значение целевой функции на предыдущей итерации. Отметим, что такой подход позволяет преодолеть недостатки субградиентного алгоритма и обеспечить монотонную сходимость метода.

1.2.4. Субградиентный алгоритм и специальный метод генерации столбцов для задачи о p-медиане

Представленные выше алгоритмы, как специализированные для решения Лагранжевой двойственной задачи, так и методы решения задач недифференцируемой оптимизации в общем виде, могут с успехом быть применены лишь к задачам, размерность которых относительно невелика, т.е. содержащих несколько сотен переменных. При возрастании размерности скорость сходимости таких алгоритмов заметно снижается и не может предоставить достаточно хорошую двойственную оценку оптимального значения. В связи с этим при решении двойственной задачи большой размерности часто используют различные эвристические способы выбора шага методов субградиентной оптимизации. Так, для рассматриваемой двойственной функции $\theta(\lambda)$ в задаче о p-медиане применяется субградиентный алгоритм с нормализованным шагом α_k, подсчитываемым по следующему эвристическому правилу [56, 68]:

$$\alpha_k = \frac{\gamma_k(1.05 \cdot BUB - \theta(\lambda^k))}{\|g(\lambda^k)\|_2^2}, \tag{1.22}$$

где BUB — верхняя оценка оптимального значения, $\|\cdot\|_2$ — евклидова норма, γ_k — параметр, причем $0 < \gamma_k \leq 2$, $g(\lambda^k)$ — субградиент в точке λ^k, имеющий следующий вид:

$$g_j(\lambda^k) = 1 - \sum_{i \in \delta^-(j)} x_{ij}(\lambda^k) - y_j(\lambda^k),\ j \in I.$$

Отметим, что данное правило выбора шага метода является схожим с описанным ранее правилом (1.20), однако в нашем случае присутствует эвристический множитель 1.05, а параметр γ_k изменяется по особому правилу, о котором речь пойдет ниже. Пока заметим лишь, что оно похоже на рекомендуемое в литературе, т.е. вначале γ_k предполагается равным 2 и уменьшается вдвое после заданного числа итераций, зависящего от размерности задачи, до некоторого порогового значения (см., например, [131, 172]).

Как было отмечено ранее, в рассматриваемом подходе используется синтез субградиентного алгоритма и специального метода генерации столбцов, основанного на приведении матрицы расстояний к специальному виду и использовании особой структуры хранения данных. Данный подход позволяет находить нижние оценки в задаче о p-медиане большой размерности, для которых применение стандартных методов, таких как непосредственный по-

иск решения в линейной релаксации или прямое применение алгоритмов субградиентной оптимизации, является затруднительным из-за большого размера задачи, но прежде всего вследствие наличия большой матрицы расстояний. Так, поскольку для вычисления значения двойственной функции Лагранжа $\theta(\lambda)$ в оперативной памяти необходимо хранить всю матрицу расстояний D, то естественным образом необходимо учитывать имеющиеся аппаратные ограничения на количество элементов в ней. Чтобы более ясно понять, насколько это важно при решении задач большой размерности, возникающих, например, из практических приложений, заметим, что для хранения всей матрицы расстояний, состоящей из элементов с плавающей запятой, для задачи с 60000 необходимо более 26Гб оперативной памяти, что зачастую превосходит возможности современных настольных ЭВМ. Отсюда можно сделать вывод, что для решения задач столь большой размерности необходимы более эффективные подходы, основанные на частичном хранении матрицы расстояний.

Для повышения эффективности работы процедуры поиска нижних оценок применяется специальный метод генерации столбцов, известный как «delayed column generation»[56]. Предположим, что матрица расстояний $D = \{d_{ij}\}$ подсчитана и каждый ее столбец упорядочен в неубывающем порядке. Таким образом, для каждого $j \in I$ имеется перестановка $u(j)$ такая, что

$$d(u_1(j), j) \leq d(u_2(j), j) \leq \cdots \leq d(u_m(j), j).$$

В этом случае значение двойственной функции Лагранжа и вектор субградиента на k-ой итерации для заданного набора множителей λ^k могут быть эффективно вычислены по следующим схемам (см. Алгоритмы 1.1, 1.2)

Отметим, что до начала основного цикла Алгоритма 1.2 подсчета субградиента в точке λ^k необходимо подсчитать вектор переменных $y(\lambda^k)$, что отражено на шаге инициализации представленного ниже алгоритма.

Как было отмечено в работе [56], для подсчета значения двойственной функции Лагранжа и вектора субградиента (при условии, что каждый столбец матрицы расстояний отсортирован по возрастанию) нет необходимости делать проход по всему j-му столбцу, но только по первым элементам, значения которых не превосходят величины λ_j^k. Таким образом, представленные процедуры 1.1, 1.2 позволяют существенно ускорить работу алгоритма, однако требуют дополнительного объема памяти при хранении упорядоченных столбцов матрицы расстояний. В данном случае для хранения всей упорядоченной матрицы расстояний, например, для задачи с 60000 узлов потребуется свыше 40Гб оперативной памяти. Для преодоления этой проблемы используется вариант метода генерации столбцов, основанный на частичном

Алгоритм 1.1 Процедура подсчета вектора оценок Лагранжа и значения двойственной функции Лагранжа

0: Инициализация: положить $\rho(\lambda^k) := -\lambda^k$, $\theta(\lambda^k) := 0$ и $j := 1$;

1: Вычислить $\theta(\lambda^k) := \theta(\lambda^k) + \lambda^k_j$ и положить $h := 1$;

2: **Если** $d(u_h(j), j) \geq \lambda^k_j$, **тогда** перейти на 5;

3: Вычислить $\rho_{u_h(j)}(\lambda^k) := \rho_{u_h(j)}(\lambda^k) + d(u_h(j), j) - \lambda^k_j$;

4: **Если** $h < m$, **тогда** положить $h := h + 1$ и перейти на 2;

5: **Если** $j < m$, **тогда** положить $j := j + 1$ и перейти на 1;

6: Найти множество $T(\lambda^k)$ и подсчитать

$$\theta(\lambda^k) := \theta L(\lambda^k) + \sum_{i \in T(\lambda^k)} \rho_i(\lambda^k).$$

Возвратить вектор оценок $\rho(\lambda^k)$ и $\theta(\lambda^k)$.

Алгоритм 1.2 Процедура подсчета вектора субградиента $g(\lambda^k)$

0: Инициализация: найти вектор $y(\lambda^k)$: $y_i := 1\ \forall i \in T(\lambda^k)$, положить $j := 1$;

1: Положить $g_j(\lambda^k) := 1 - y_j(\lambda^k)$, $h := 1$;

2: **Если** $d(u_h(j), j) - \lambda^k_j \geq 0$, **тогда** перейти на 5;

3: **Если** $y_{u_h(j)}(\lambda^k) = 1$, **тогда** положить $g_j(\lambda^k) := g_j(\lambda^k) - 1$;

4: **Если** $h < m$, **тогда** положить $h := h + 1$ и перейти на 2;

5: **Если** $j < m$, **тогда** положить $j := j + 1$ и перейти на 1, **иначе** стоп;

Возвратить вектор субградиента $g(\lambda^k)$.

хранении матрицы расстояний. Суть предложенного в [56] подхода состоит в том, что для каждого отсортированного j-го столбца матрицы расстояний алгоритм хранит в оперативной памяти лишь некоторое число $m^j_0 < m$ первых (наименьших) элементов, называемых *активными*. Если на некоторой итерации k алгоритма множитель λ^k_j превысит величину $d(u_{m^j_0}(j), j)$, то j-й столбец дополняется на некоторое фиксированное количество элементов m_1. Полагая затем $m^j_0 := m^j_0 + m_1$, алгоритм продолжает свою работу.

Как было отмечено, недостатком различных вариантов субградиентного алгоритма является их медленная, немонотонная сходимость. Для преодоления данного недостатка и контроля роста значений множителей Лагранжа, а следовательно, и числа активных элементов, применяется метод стабилизации, предложенный в работе [164]. Его суть состоит в установлении для каждого множителя верхней границы, вначале полагаемой равной первому

(наименьшему) элементу в соответствующем столбце, т.е. $\lambda_j^0 = d(u_1(j), j)$. Затем на каждой итерации верхняя граница устанавливается так, чтобы $\lambda_j^k \leq ub_j^k$, где

$$ub_j^k = \min_{k \in I}\{d(u_k(j), j) : \ d(u_k(j), j) > \lambda_j^{k-1}\}.$$

Говоря другими словами, каждый множитель Лагранжа не может превзойти сразу два элемента в столбце матрицы расстояний на одной итерации алгоритма.

Суммируя все вышесказанное, общая схема алгоритма поиска нижней оценки оптимального значения, т.е. максимизации двойственной функции Лагранжа $\theta(\lambda)$, может быть представлена в следующем виде:

Алгоритм 1.3 Процедура поиска нижних оценок оптимального значения в задаче о p-медиане

0: Инициализация: положить $BLB := -\infty$, задать начальное значение параметра γ_0, параметры θ и ε. Положить $\lambda_j^0 := d(u_1(j), j), \ j = 1 \dots m, \ k := 0$ и $\beta := 0$;

1: Запустить Алгоритм 1.1. Подсчитать вектор оценок Лагранжа $\rho(\lambda^k)$ и значение двойственной функции Лагранжа $\theta(\lambda^k)$;

2: **Если** $\theta(\lambda^k) > BLB$, **тогда** $BLB := \theta(\lambda^k)$ и $\beta := 0$;

3: **Если** $BUB - BLB < \varepsilon$, **тогда** то стоп;

4: Вычислить вектор $y(\lambda^k)$, запустить Алгоритм 1.2 и найти вектор субградиента $g(\lambda^k)$;

5: **Если** $\beta \geq \hat{\beta}$, **тогда** $\gamma_k := \frac{\gamma_k}{\theta}$ и $\beta := 0$, **иначе** $\beta := \beta + 1$;

6: **Если** $\gamma_k < 0.005$, **тогда** стоп;

7: Вычислить шаг $\alpha_k := \dfrac{\gamma_k(1.05 \cdot BUB - \theta(\lambda^k))}{\|g(\lambda^k)\|_2^2}$;

8: Положить $\lambda^{k+1} := \lambda^k + \alpha_k \, g(\lambda^k)$, $k := k + 1$ и перейти на 1.

Возвратить найденный вектор множителей $\bar{\lambda}$, нижнюю оценку BLB.

Как было замечено ранее, параметр метода γ_k выбирается как $0 < \gamma_k \leq 2$ [32]. Величина θ представляет собой параметр модификации шага метода в зависимости от числа итераций без улучшения нижней оценки. То есть, если на протяжении $\hat{\beta}$ итераций текущая нижняя оценка не изменяет своего значения, то параметр метода γ_k уменьшается в θ раз. Отметим также, что верхняя оценка BUB, применяемая для подсчета шага метода, считается известной к началу запуска алгоритма.

1.2.5. Поиск прямых оценок оптимального значения. Ядровая эвристика

С помощью релаксации Лагранжа, как известно, помимо двойственной оценки оптимального значения в некоторых случаях удается найти и оптимальное значение в исходной задаче. Предположим, что, как и прежде, имеется следующая задача целочисленного линейного программирования:

$$\min\{\langle c, x\rangle : Ax \geq b, Dx \geq d, x \in \mathbb{Z}_+^n\} \tag{IP}$$

и релаксация Лагранжа относительно ограничений $Dx \geq d$:

$$\theta(\lambda) = \min\{\langle c, x\rangle + \lambda(d - Dx) : Ax \geq b, x \in \mathbb{Z}_+^n\}. \tag{IP_λ}$$

Справедливо следующее утверждение.

Теорема ([143, 257]). *Если $\lambda \geq 0$ и, к тому же, имеют место следующие условия:*

1. *$x(\lambda)$ является оптимальным в (IP_λ),*
2. *$Dx(\lambda) \leq d$,*
3. *$(Dx(\lambda))_i - d_i = 0$ для всякого $\lambda_i > 0$,*

то решение $x(\lambda)$ также является оптимальным и в исходной задаче (IP).

В случае, когда вместо условий-неравенств ослабляются ограничения-равенства, условие 3 теоремы выполняется автоматически, поэтому решение $x(\lambda)$ будет оптимальным в задаче (IP) в случае, если оно допустимо в ней [101, 229, 275]. Также, если в случае с ослаблением ограничений-неравенств условия дополняющей нежесткости (условия комплементарности) 3 не выполнены, то полученное решение $x(\lambda)$ является по меньшей мере допустимым в исходной задаче (IP), однако не оптимальным. Оптимальность решения может быть, в частности, проверена с помощью метода ветвей и границ или другими методами [151]. В противном случае, полученное решение (IP_λ) является недопустимым в (IP), однако можно ожидать, что по мере улучшения двойственных переменных при решении Лагранжевой двойственной задачи $x(\lambda)$ будет становится «близким» к допустимому в (IP). В данном случае возможно применение простых эвристик (например, жадной [102]) для получения из $x(\lambda)$ допустимого решения, которое при этом лишь незначительно изменится по значению целевой функции [275].

В качестве примера стоит привести релаксацию задачи о p-медиане (справедливую и для случая простейшей задачи размещения) относительно ограничений на обслуживание

каждого клиента в точности одним предприятием (1.9). В данном случае, возможен вариант получения решения $(y_i(\lambda), x_{ij}(\lambda))$, в котором большинство клиентов будет обслуживаться лишь раз, однако некоторая малая часть из них не обслуживаться вовсе, делая данное решение недопустимым. Тогда простейший вариант поиска допустимого решения исходной задачи на основе известного двойственного состоит в присоединении всех клиентов к ближайшим открытым (в двойственном решении) предприятиям [68, 228]. Таким образом, лагранжева эвристика будет состоять в последовательном решении лагранжевой двойственной задачи и поиска $\theta(\lambda^k)$ с помощью, например, субградиентного алгоритма, а также в поиске на каждой итерации допустимого решения и, таким образом, постепенном улучшении верхней оценки оптимального значения. Отметим, что подсчет значения целевой функции на каждой итерации эвристики является довольно затратным с вычислительной точки зрения, поэтому зачастую обновление верхней оценки происходит с некоторой периодичностью, которая представляет собой компромисс между скорость алгоритма и качеством получаемого решения.

Еще одним вариантом подхода к поиску допустимых решений на основе информации, получаемой при решении лагранжевой двойственной задачи, являются подходы, основанные на частичном фиксировании переменных (так называемые тесты на фиксирование переменных) [47, 61, 275], а также применение метода ветвей и границ [96] для поиска лучших допустимых решений.

В данном разделе рассматривается подход, использующий информацию об оценках Лагранжа для поиска допустимых решений в прямой задачи известный как «выбор ядра» (core selection). Одним из первых данный подход был применен для задачи о покрытии множеств в работах [87, 91]. По своей сути данная идея схожа с известным методом частичной оценки для решения задач линейного программирования большой размерности [208]. Авторы предложили использовать информацию о текущих значениях двойственных переменных для построения так называемого ядра, представляющего собой исходную задачу, однако содержащую значительно меньшее количество переменных. Отбор переменных производится с использованием информации об оценках Лагранжа для столбцов матрицы инцидентности. Так, на основе обширного вычислительного эксперимента авторы предложили выбирать для ядра лишь пять столбцов, имеющих минимальные оценки Лагранжа, а также все столбцы, оценка которых меньше 0.1. Среди других работ, в которых используется поиск решения задачи, редуцированной на основании информации о двойственных оценках, стоит выделить [48], в которой рассматривались транспортные задачи с фиксированной стоимостью.

Для рассматриваемой задачи о p-медиане метод поиска верхних оценок на основе информации о двойственных переменных был предложен и исследован в работах [56, 57, 59, 60,

211, 212]. В рамках данного подхода предполагается, что по завершению процедуры поиска нижней оценки оптимального значения с помощью представленного выше алгоритма найдены близкие к оптимальным множители Лагранжа, позволяющие сделать заключение о том, какие именно переменные прямой задачи (y_i, x_{ij}) скорее всего принимают значение, равное единице.

Таким образом, для поиска верхней оценки после остановки субградиентного алгоритма 1.3 строится так называемая ядровая задача, представляющая собой редуцированную задачу о p-медиане, содержащую только некоторое подмножество «перспективных» переменных, в то время как все остальные переменные фиксируются в нуле. Отбор такого подмножества происходит на основе полученных оценок Лагранжа.

Пусть $\bar{\lambda}$ будут «оптимальными» множителями Лагранжа, полученными на выходе субградиентного алгоритма, а $\rho_i(\bar{\lambda})$ и $q_{ij}(\bar{\lambda}) = d_{ij} - \lambda_j$ — оптимальными оценками Лагранжа для переменных y_i и x_{ij} соответственно. Ядровая задача (core problem) определяется подмножеством переменных, оценки Лагранжа которых меньше некоторых заданных пороговых значений: ν для переменных y_i и μ для x_{ij}. Другими словами, необходимо решить задачу о p-медиане (1.8)–(1.13), содержащую только те переменные y_i и x_{ij}, для которых

$$i \in I(\bar{\lambda}, \nu) = \{i \in I : \ \rho_i(\bar{\lambda}) \leq \nu\},$$

$$(i,j) \in A(\bar{\lambda}, \mu) = \{(i,j) \in A : \ i \in I(\bar{\lambda}, \nu), \ q_{ij}(\bar{\lambda}) \leq \mu\}.$$

Для поиска решения в такой задаче широко используются коммерческие программные средства решения задач линейного целочисленного программирования с ограничением на время решения задачи для сокращения общего времени счета. Также вполне успешно применение различных эвристических или метавэвристических подходов, поскольку размер ядровой задачи существенно меньше исходной.

1.2.6. Лагранжева эвристика для задачи о p-медиане

Интенсивное тестирование субградиентного алгоритма и метода генерации столбцов в сочетании с ядровой эвристикой выявило некоторые особенности такого подхода. К ним относится прежде всего тот факт, что для подсчета шага (1.22) субградиентного алгоритма необходимо иметь некоторую верхнюю оценку оптимального значения, которая, вообще говоря, не известна в начале алгоритма, однако ее наличие имеет существенное влияние на сходимость и на качество получаемых в конце оценок Лагранжа. С другой стороны, наличие «хорошего» решения Лагранжевой двойственной задачи, а следовательно и оценок Лагран-

жа, является исключительно важным при выборе переменных для ядровой задачи и для получения «хорошей» верхней оценки оптимального решения.

В связи с этим для улучшения качества получаемого решения процедуры поиска верхней и нижней оценок последовательно чередуются, позволяя каждый раз улучшать найденное решение. Обозначим через $Z(x, y)$ значение целевой функции (1.8) для решения (x, y), тогда общая схема алгоритма поиска приближенных решений в задаче о p-медиане может быть представлена следующим образом:

Алгоритм 1.4 Поиск приближенных решений в задаче о p-медиане

0: Инициализация. Найти допустимую точку $(\bar{x}, \bar{y})$, задать лучшую найденную верхнюю оценку как $BUB := Z(\bar{x}, \bar{y})$, а лучшую нижнюю оценку как $BLB := -\infty$.

1: Найти с помощью Алгоритма 1.3 множители $\bar{\lambda}_1$ и $\theta(\bar{\lambda}_1)$, $BLB := \theta(\bar{\lambda}_1)$.

2: Построить $I(\bar{\lambda}_1, \nu)$, $A(\bar{\lambda}_1, \mu)$ и найти решение $(\hat{x}, \hat{y})$ ядровой задачи. Если $Z(\hat{x}, \hat{y}) < BUB$, то $BUB := Z(\hat{x}, \hat{y})$ и $(\bar{x}, \bar{y}) := (\hat{x}, \hat{y})$.

3: Запустить субградиентный алгоритм 1.3, стартуя с набора множителей $\bar{\lambda}_1$. Найти вектор множителей $\bar{\lambda}_2$, $BLB := \theta(\bar{\lambda}_2)$.

4: Построить $I(\bar{\lambda}_2, \nu)$, $A(\bar{\lambda}_2, \mu)$ и найти решение $(\hat{x}, \hat{y})$ ядровой задачи. Если $Z(\hat{x}, \hat{y}) < BUB$, то положить $BUB := Z(\hat{x}, \hat{y})$ и $(\bar{x}, \bar{y}) := (\hat{x}, \hat{y})$.

5: Запустить субградиентный алгоритм 1.3, стартуя с набора множителей $\bar{\lambda}_2$, найти вектор $\bar{\lambda}_3$, $BLB := \theta(\bar{\lambda}_3)$.

Возвратить оценки BLB, BUB и приближенное решение $(\bar{x}, \bar{y})$ задачи о p-медиане.

Отметим, что в ходе вычислительных экспериментов, представленных, например, в [56], было выявлено, что после трех последовательных повторений процедур поиска верхних и нижних оценок их значения становятся достаточно близки и последующие запуски не позволяют добиться улучшения.

1.3. Параллельный алгоритм поиска нижних оценок оптимального значения в задаче о p-медиане большой размерности

В настоящем разделе дается обзор архитектур параллельных вычислительных систем, современных интерфейсов и моделей написания параллельных программ, а также некоторых широко известных теоретических результатов касательно вычислительной сложности и производительности параллельных алгоритмов, включая различные абстрактные модели параллельных машин. Приводится также обзор современных параллельных и последователь-

ных алгоритмов поиска решений в задаче о p-медиане большой размерности, предлагается схема распараллеливания процедуры поиска нижней оценки оптимального значения в задаче о p-медиане, включающая в себя описанный ранее субградиентный алгоритм и специальный метод генерации столбцов (см. Алгоритм 1.3) [9, 161].

1.3.1. Параллельные вычисления и алгоритмы

В настоящее время наблюдается бурное, взрывное развитие области параллельных и распределенных вычислений, которое связано прежде всего со стремительным удешевлением вычислительных мощностей, памяти и средств связи, а следовательно, значительным увеличением числа вычислительных машин, поддерживающих параллельные способы выполнения программ и обработку данных, в первую очередь в сфере интенсивных высокопроизводительных вычислений. Все большее увеличение доступных, недорогих вычислительных мощностей способствует появлению большого числа вычислительных задач огромной размерности, для которых прежде не существовало возможности обработки и решения. Не исключением стала и область задач комбинаторной оптимизации и целочисленного программирования, для которых на протяжении последних лет был предложен ряд параллельных алгоритмов, а также разработаны комплексы параллельных программ для решения, например, задач смешанного целочисленного линейного программирования общего вида [94, 95, 211, 240, 259, 277].

Несмотря на сложность разработки и реализации параллельных алгоритмов и программ, затрудняющую их повсеместное применение, к настоящему времени предложен ряд инструментов написания параллельных программ, главным образом представляющих собой расширения и надстройки над широко применяемыми языками C, Fortran, Java, Pascal, Lisp и т.д., а также сравнительно обособленные языки программирования (ABCL, Adl, Linda, Occam). Данные инструменты способны предоставить высокоуровневые средства реализации параллельных программ, избавляющие программиста от необходимости использования низкоуровневых способов передачи сообщений и синхронизации различных частей параллельной программы.

Разработка и реализация эффективного параллельного алгоритма напрямую зависит от архитектуры параллельной вычислительной системы, которая согласно предложенной М. Флинном в конце 1960-х годов классификации может быть условно отнесена к одной из четырех групп в соответствии с наличием параллельного/одиночного потока (последовательности) данных или команд [134, 135]. Машины с архитектурой SISD (single instruction, single data) представляют собой стандартные системы с одним процессором, который после-

довательно выполняет команды для единственного потока данных, поэтому такая машина, вообще говоря, не является параллельной. Следующим классом являются SIMD-машины (single instruction, multiple data), в которых единственный счетчик команд последовательно выбирает одну команду, а набор процессоров выполняет ее одновременно для некоторого разделенного между ними набора данных (так называемого вектора данных). Выполняя одну и ту же команду, каждый процессор имеет свое адресное пространство. С точки зрения программиста, такая архитектура не отличается от SISD, поскольку последовательный алгоритм может быть выполнен как на SISD-машине, так и на имеющей архитектуру SIMD [237]. Отметим, что в эту категорию попадают популярные в прошлом для научных вычислений векторные и матричные процессоры, практически полностью исчезнувшие к сегодняшнему дню вследствие развития микропроцессорных технологий. Следующим выделяемым классом является MISD (multiple instruction, single data). Процессоры в машинах с такой архитектурой выполняют одновременно различные инструкции над одним потоком данных по типу конвейера, т.е. одновременно может происходить, например, расшифровка, распаковка, организация данных, и т.д. Отметим, что такая архитектура не получила распространения и к настоящему времени ее реализаций не существует. Наконец, к последнему классу архитектур относятся MIMD-машины (multiple instruction, multiple data), в которых, как следует из названия, различные процессоры выполняют одновременно различные наборы инструкций для различных наборов данных. Данный класс является наиболее общим и универсальным, к нему принято относить вычислительные кластеры, мультипроцессорные машины, а также традиционные машины на основе многоядерных процессоров [10, 43, 147, 237, 265].

Класс MIMD, в свою очередь, разделяется на подклассы по принципу взаимодействия процессоров между собой, а именно, на машины с общей (SMP) и распределенной памятью. К машинам с общей памятью относятся так называемые мультипроцессоры, в которых все процессоры имеют единое адресное пространство и общий пул памяти. Для машин с распределенной памятью, куда относят, например, вычислительные кластеры, характерно наличие своего собственного, независящего от других, адресного пространства (локальной памяти). Причем взаимодействие различных процессоров происходит посредством передачи сообщений, в то время как каждый из них выполняет свой набор команд над данными, находящимися в собственной локальной памяти. Такой подход к организации параллельных вычислений является относительно новым и позволяет строить массово-параллельные машины, состоящие из тысяч отдельных, независимых узлов, содержащих процессоры, обладающие общей памятью и связанные между собой посредством современных скоростных сетей связи.

Неоспоримым преимуществом машин с распределенной памятью является высокая степень надежности, простая расширяемость, а также высокая производительность [265].

В зависимости от имеющейся параллельной архитектуры применяются различные модели программирования. Для машин с общей памятью одной из наиболее популярных технологий является интерфейс программирования приложений OpenMP [2]. Как и другие средства программирования для машин с общей памятью, она позволяет довольно легко построить параллельный алгоритм на основе последовательного, выделив информационно-независимые операции. Напомним, что операции алгоритма называются информационно-независимыми, если результат одной из них не является аргументом другой [249]. Таким образом, OpenMP-программа состоит из последовательного алгоритма, в котором информационно-независимые участки отмечены для параллельного выполнения, после завершения которых алгоритм продолжает исполнение последовательно, реализуя таким образом параллелизм по принципу FORK-JOIN [215]. Преимуществом OpenMP является прежде всего простота и легкость написания параллельных приложений без существенного изменения программы, однако такой подход применим только в случае машины с общей памятью.

В случае вычислительных кластеров и других машин с распределенной памятью по типу MIMD общепризнанным и наиболее популярным является интерфейс передачи сообщений MPI [149]. Любая реализованная с помощью MPI параллельная программа представляет собой набор одновременно выполняющихся и взаимодействующих между собой процессов. Как было упомянуто ранее, каждый процесс имеет свое собственное адресное пространство и набор переменных и данных. Взаимодействие между процессами происходит посредством отправки сообщений, а каждый процесс идентифицируется по уникальному номеру, изменяющемуся в пределах от 0 до $n-1$, а также по так называемому коммуникатору, т.е. группе процессов, к которой он принадлежит. Сообщения, которыми обмениваются процессы, находящиеся в группе с одним коммуникатором, представляют собой некоторый набор данных определенного типа [1, 10]. Недостатком MPI является необходимость серьезного анализа последовательного алгоритма, а также существенное изменение имеющейся последовательной программы и разделение данных между процессами. К достоинствам MPI относится прежде всего высокая масштабируемость приложений.

Несмотря на то что архитектура MIMD позволяет писать для запуска на разных процессорах машины различные программы, исполняющиеся одновременно и координирующиеся между собой для достижения единой цели, на практике часто подготавливают лишь одну программу, которая запускается одновременно на всех процессорах MIMD-машины, однако включает в себя определенные условия для разделения различных участков кода между раз-

личными процессорами. Такой стиль программирования параллельных машин называется SPMD (Single Program, Multiple Data) и является довольно популярным для MIMD-компьютеров [237, 265].

Отметим, что MPI является далеко не единственным средством разработки приложений для машин с распределенной памятью. Из других инструментов стоит выделить программный пакет PVM (Parallel Virtual Machine), также основанный на обмене сообщениями между различными процессами; разработанный в Йельском университете язык Linda; DVM-подход, предложенный в Институте прикладной математики им. М.В. Келдыша РАН [22].

Теоретическому исследованию вычислительной сложности и производительности параллельных алгоритмов посвящено множество работ, количество которых стало быстро увеличиваться начиная с 1970-х. Из первых полученных теоретических результатов стоит выделить так называемую теорему или закон Брента, характеризующую то, насколько эффективно может быть использован имеющийся параллелизм задачи при построении эффективного параллельного алгоритма.

Теорема (Brent [82, 252]). *Пусть C представляет из себя алгоритм, который может быть выполнен за t параллельных шагов при условии, что временем, затраченным на коммуникацию процессоров, можно пренебречь. И пусть m_i является числом элементарных операций, совершенных на i-ом шаге, а $m = \sum_{i=1}^{t} m_i$. Предположим, что в наличии имеется машина M с p процессорами, выполняющая те же элементарные операции, причем $p \leq \max_i m_i$. Тогда если временем сообщения между операциями алгоритма C, выполняемого на машине M, можно пренебречь, то тот же алгоритм может быть выполнен за T_p шагов на машине M, причем справедливо*

$$T_p \leq \frac{m}{p} + t.$$

Таким образом, если предположить, что одна элементарная операция выполняется за одну единицу времени, то теорема Брента устанавливают границу на время выполнения параллельного алгоритма на p процессорах машины M. Отметим, что модель машины M принято называть PRAM-машиной, являющейся обобщением известной модели машины с произвольным доступом к памяти (PAM-машины).

Использование PRAM-машины при анализе параллельных алгоритмов позволяет оценить эффективность алгоритма при серьезном допущении об отсутствии времени коммуникации между процессорами, а также задержки между операциями, которое на практике может быть существенным, прежде всего для машин с распределенной памятью. В связи с этим были предложены новые вычислительные модели, например BSP (Bulk Synchronous

Parallel) [185, 270], позволяющая более точно оценить производительность параллельных алгоритмов за счет учета времени передачи данных между процессорами, а также времени на необходимую синхронизацию. Обобщениями BSP-модели являются модели LogP и PLogP, еще более точно имитирующие и оценивающие эффективность параллельного алгоритма на современных вычислительных кластерах, поскольку в данных моделях также учитывается время задержки на обработку, отправку и получение сообщений между процессорами [106, 184, 185].

Из важных базовых результатов касательно эффективности и производительности параллельных алгоритмов помимо теоремы Брента стоит также выделить так называемый закон Амдала [51]. Согласно ему имеются очевидные объективные причины, препятствующие росту производительности параллельного алгоритма. Было замечено, что время работы алгоритма делится на затраченное на выполнение части программы, которая может или выполняться только последовательно, или может быть распараллелена. Предположим, что s представляет собой долю от общего времени, затрачиваемого на выполнение последовательной части, тогда, если предположить, что параллельная часть алгоритма может быть идеально выполнена на p процессорах, т.е. за время $t_p = (1 - s)/p$, тогда ускорение параллельного алгоритма при использовании p процессоров составляет

$$S = \frac{1}{s + \frac{1-s}{p}}.$$

Следствием закона Амдала является тот факт, что даже при значительном увеличении количества процессоров для решения некоторой задачи ускорение параллельного алгоритма не может превысить величину, равную $\frac{1}{s}$. Таким образом, если имеется некий алгоритм, в котором 10% общего времени затрачивается на выполнение последовательной части, которая не может быть распараллелена, то в лучшем случае возможно получить лишь десятикратное ускорение параллельного алгоритма над последовательным, вне зависимости от числа имеющихся процессоров [147, 155, 215]. Например, если предположить, что $s = 0.001$ и $p = 2048$, то можно достичь лишь 672-х кратное ускорение параллельной программы [215]. Влияние закона Амдала было так велико, что послужило аргументом против параллельных вычислений, поскольку эффективность использования аппаратного обеспечения значительно снижается даже при небольшом значении величины s. Долгое время казалось фантастическим получение даже двухсоткратного ускорения для реальных практических задач.

Однако, в 1980-х Джон Густафсон показал, что для некоторых используемых им алгоритмов имеет место почти тысячекратное ускорение по сравнению со скоростью работы на одном процессоре, что, как казалось, расходится с законом Амдала [154]. Густафсон заметил,

что по мере развития аппаратного обеспечения и роста его производительности размерность решаемых задач также возрастает. При росте размерности задач количество вычислений, необходимых для параллельной части программы, растет значительно быстрее, чем для последовательной. Другими словами, при росте размерности задачи величина s уменьшается, а следовательно увеличивается ускорение. Данное наблюдение получило название закона Густафсона (Густафсона-Барсиса) [155, 215].

Таким образом, несмотря на очевидность закона Амдала, последовательная часть алгоритма в его формуле была сильно преувеличена, что, как полагалось, не могло обеспечить эффективного использования ресурсов параллельных вычислений. Отметим также, что закон Амдала устанавливает ограничения на ускорение параллельного алгоритма при изменении числа процессоров p, однако для одинаковой размерности задачи. Характеристику алгоритма по отношению к такому ускорению часто называют сильной масштабируемостью. С другой стороны, закон Густафсона характеризует ускорение при увеличении как числа процессоров, так и размерности задачи. Таким образом, характеристику отвечающую за то, насколько большую задачу можно решить за некоторое фиксированное время зачастую называют слабой масштабируемостью [215].

Говоря о параллельных вычислениях в целом, стоит также упомянуть и набирающую популярность в последние годы область распределенных вычислений. С развитием современных сетей связи получила популярность идея об использовании сети Интернет в качестве большой неоднородной параллельной вычислительной среды, в состав которой входят миллионы компьютеров, кластеров и других вычислительных средств по всему миру. Идея обрела своих сторонников в силу того, что большое количество практических задач допускают декомпозицию на отдельные подзадачи, для которых нет необходимости в каком-либо обмене информацией (либо такой обмен чрезвычайно низок), что позволяет свести на нет потери, вызванные пропускной способностью сетей связи. Реализацию данное направление получило в форме так называемых грид-вычислений, что в дальнейшем привело к созданию программного комплекса BOINC Университета Беркли для организации добровольных распределенных вычислений. К наиболее впечатляющим проектам, основанным на модели BOINC для распределенных добровольных вычислений, стоит отнести проекты поиска внеземной жизни SETI@HOME, новых лекарственных препаратов, сравнение и создание базы белков SIMAP@HOME, а также реализуемый в ИДСТУ СО РАН проект поиска решений для комбинаторных и дискретных задач SAT@HOME [20, 239].

Другим современным, быстроразвивающимся направлением в области разработки параллельных высокопроизводительных алгоритмов являются вычисления с использованием

графических процессоров (GPGPU). Стремительному росту интереса к данному направлению способствовало быстрое развитие индустрии видеоигр, а следовательно опережающее инвестирование и разработки в области высокопроизводительной обработки графической информации [237]. Неоспоримым преимуществом графических процессоров над центральными является более высокая производительность в задачах с интенсивными вычислениями, поскольку они изначально оптимизированы под массово-параллельную обработку больших объемов информации. Настоящий бум в данном направлении произошел с разработкой в 2007 году программно-аппаратной архитектуры CUDA, поскольку она позволила использовать широко распространенные в научных вычислениях языки программирования C/C++, Python для написания параллельных программ с использованием графических процессоров от Nvidia [273]. Отметим, что такого рода параллельные алгоритмы были предложены и для дискретных задач оптимизации [84], в частности для задач маршрутизации [253]. Обзор параллельных алгоритмов такого рода также можно найти в [236]. В работах [209, 210] для задачи о p-медиане был предложен параллельный алгоритм на графических процессорах, реализующий метод замены вершин, т.е. локальный поиск по SWAP-окрестности. Авторы приводят результаты численных экспериментов на задачах из известной библиотеки тестовых примеров OR-lib, размерность которых не превосходит 900 вершин. Для данного набора тестовых задач разработанный алгоритм показал скорость работы, превосходящую таковую для последовательного алгоритма в 10–57 раз.

1.3.2. Несколько известных параллельных и последовательных алгоритмов для задачи о p-медиане большой размерности

Как было отмечено ранее, за более чем пятидесятилетнюю историю исследований задачи о p-медиане до настоящего времени так и не предложено алгоритмов, способных находить решения в задачах очень большой размерности, т.е. для графов, содержащих сотни тысяч узлов. Очевидно, что это связано не только с большой трудностью задачи, т.е. с неспособностью алгоритмов найти решение в задах такой размерности за приемлемое время (поскольку задача о p-медиане NP-трудна в сильном смысле), но прежде всего с ограничением на объем данных, который может храниться в оперативной памяти вычислительной машины на всем протяжении работы параллельного алгоритма. Для задачи о p-медиане в настоящее время предложен ряд алгоритмов, способных находить точные или приближенные решения на графах, содержащих около 90000 узлов. Среди таких методов стоит отметить предложенный в [140] алгоритм генерации строк и столбцов, основанный на формулировке

задачи о p-медиане в терминах задачи о покрытии множеств. Данный алгоритм позволяет получить точные решения для задач размерностью до 90000 узлов, однако только в случае относительно большого числа медиан. В работах [177, 178] предложены весьма эффективные гибридные метаэвристические алгоритмы, основанные на агрегировании данных задачи и включающие в себя метод локального поиска с чередующимися окрестностями (VNS), а также использующие коммерческие программные пакеты для поиска решений в редуцированных агрегированных задачах. Однако главными недостатками данных подходов является круг их применения, который ограничивается лишь задачами на плоскости, в которых к тому же расстояния между точками задаются евклидовой метрикой, а также невозможностью оценки полученного приближенного решения, поскольку алгоритм не использует никакой информации о нижней (двойственной) оценке оптимального значения задачи. Одним из наиболее эффективных алгоритмов является предложенный в [164] прямо-двойственный метод локального поиска с чередующимися окрестностями (PDVNS), а также описанный в разделе 1.2 настоящей главы эвристический алгоритм, впервые предложенный в [56].

Для всех указанных алгоритмов были представлены результаты решения задач размерностью 90000 вершин, которая, являясь рекордной на сегодняшний день, не позволяет с успехом применять модель задачи о p-медиане в некоторых из ее практических приложений. Для преодоления данного недостатка вполне успешно и широко применяются средства и методы бурно развивающейся области параллельных вычислений. Так, для задачи о p-медиане был предложен ряд различных параллельных вариантов известных алгоритмов, по большей части метаэвристических, среди которых стоит выделить параллельный поиск с чередующимися окрестностями, предложенный в работах [104, 141], а также так называемый рассеянный поиск (scatter search) [142]. Параллельные алгоритмы в данных работах представляют собой по сути методы мультистарта (multi-search) с периодическим асинхронным обменом результатами вычислений между процессорами. Причем в первой работе авторы используют так называемую идею центральной памяти [105]. В рамках данного подхода имеется один управляющий процессор, через который остальные вычислительные процессоры производят асинхронный обмен информацией о наилучших найденных решениях. Каждый процессор фактически производит поиск решения задачи лишь относительно своей собственной окрестности, случайно выбранной при инициализации. Преимущество такого метода состоит в сохранении изначальной простоты последовательного алгоритма, легкости распараллеливания и хорошей масштабируемости, в то время как асинхронный обмен данными обеспечивает независимость каждого вычислительного процессора и отсутствие задержек, вызванных синхронизацией.

В работе [141] предлагаются три стратегии для распараллеливания локального поиска с чередующимися окрестностями. Первая состоит в разделении окрестности решения между различными процессорами и ее параллельное исследование с целью поиска наилучшего решения. При использовании второй стратегии происходит параллельный запуск алгоритма на различных процессорах и выбор наилучшего решения в конце, что является одним из простейших вариантов структуры параллельного алгоритма. Наконец, третья стратегия представляет собой классический подход, в котором один управляющий процессор запускает последовательный алгоритм, распределяет начальное решение между остальными процессорами и контролирует ход работы посредством синхронного сбора полученных решений, выбора наилучшего и его передачи остальным процессорам для перехода к следующей итерации. В [142] схожие стратегии использовались при распараллеливании алгоритма рассеянного поиска.

Отметим, что в представленных работах авторы производили тестирование разработанных параллельных алгоритмов лишь на задачах небольшой размерности, в которых число узлов графа (т.е. количество мест возможного размещения предприятий и число клиентов) составляло не более 12000, что не составляет трудности для современных последовательных методов. Отметим также, что данные параллельные алгоритмы являются метаэвристическими и, вообще говоря, не могут гарантировать нахождения глобального минимума, получая, в лучшем случае, лишь близкие к нему решения без какой-либо информации касательно оценки их качества.

Для задачи о p-медиане в литературе можно найти не так много параллельных версий специализированных точных алгоритмов для ее решения. В работе [241] исследуется обобщенная дискретная задача размещения, частным случаем которой является задача о p-медиане. Авторами предлагается лагранжева эвристика для поиска приближенных решений в данной задаче, в которой для поиска прямых оценок оптимального решения применяется метод фиксирования переменных и жадный алгоритм. Предлагается схема распараллеливания полученного метода с использованием стандарта POSIX Threads, а также приводятся результаты вычислительных экспериментов для задач малой размерности, полученных при помощи вычислительного устройства с общей памятью при небольшом количестве параллельных процессоров. В качестве стратегии распараллеливания выбирался метод разделения матрицы расстояний между различными процессорами и ее параллельная обработка (так называемое распараллеливание по данным).

Как было отмечено ранее, для задач смешанного целочисленного линейного программирования общего вида известен ряд параллельных точных алгоритмов, таких как, например,

метод ветвей и границ или метод ветвей и отсечений [240, 277], реализованных, в частности, в ряде параллельных решателей [95, 259], в том числе и на основе языка моделирования GAMS [85], которые также могут быть успешно применены и к задаче о p-медиане [211].

1.3.3. Параллельная реализация субградиентного алгоритма и метода генерации столбцов

Идея предлагаемого параллельного алгоритма состоит в следующем. Фактически на каждой итерации алгоритма подсчет новых значений множителей Лагранжа и вектора субградиента может производиться частями на разных процессорах независимо друг от друга, а следовательно, параллельным процессорам необходимо оперировать лишь некоторыми непересекающимися подгруппами столбцов матрицы расстояний. Таким образом, итерационный процесс может быть успешно распараллелен. Отметим, что такой подход к распараллеливанию называют параллелизмом по данным, который связан с моделью SPMD (Single Program, Multiple Data) для программирования MIMD-машин, являющийся к настоящему времени наиболее распространенным и общепринятым стилем для MIMD [237].

При реализации параллельного алгоритма использовался интерфейс обмена сообщениями MPI. Напомним, что приложение MPI состоит из нескольких ветвей, или процессов, выполняющихся одновременно и обменивающихся данными в виде сообщений, общих данных у процессов нет, причем каждый процесс идентифицируется по уникальному номеру и коммуникатору к которому он принадлежит. Поскольку при выполнении MPI-приложения количество параллельных процессов может, вообще говоря, не совпадать с числом имеющихся процессоров вычислительной системы, то в дальнейшем, говоря о реализации параллельного алгоритма, будем говорить о процессах, а не о процессорах вычислительной системы, имея в виду программную реализацию. В качестве модели взаимодействия между процессами использовалась модель Master-Slave [43, 147, 215, 249], в рамках которой выделяется один главный процесс Master, управляющий работой всех остальных, называемых Slave-или вычислительные процессы. Отметим, что такая модель взаимодействия процессов с использованием параллелизма по данным фактически относится к так называемому MPMD (Multiple Program, Multiple Data) стилю параллельного программирования для MIMD-машин [249]. Несмотря на название, фактически в данном случае имеются лишь две программы, одна из которых отвечает за работу управляющего, а другая — за работу всех вычислительных процессов, каждый из которых исполняет ее копию. Заметим также, что в представляемой

реализации параллельного алгоритма управляющий процесс имеет номер 0, а вычислительные — $1, \ldots, s$.

Разработанный параллельный алгоритм представлен на блок-схеме (см. рис. 1.3). Перед началом основного цикла управляющий процесс считывает входные данные, состоящие, как правило, из набора координат узлов графа, и устанавливает значения параметров алгоритма, таких как количество вершин m, медиан p, начальное количество активных элементов в каждом столбце m_0^j, а также диапазон индексов столбцов, т.е. размер блока, который каждый из Slave-процессоров будет обрабатывать, в форме (j_q, n_q), где j_q — индекс первого столбца в блоке для данного процессора, n_q — размер блока, а $q = 1, \ldots, s$ — номер процессора. По завершению он формирует и отправляет всем вычислительным процессорам сообщения, содержащие не только все считанные координаты вершин, но и индивидуальный набор параметров. На основании полученных данных каждый вычислительный процесс находит свою собственную часть матрицы расстояний $D = \{d(i,j)\}$, $i = u_1(j), \ldots, u_{m_0^j}(j)$; $j = j_q, \ldots, j_q + n_q$, а также устанавливает начальные значения множителей Лагранжа $\lambda_j^0 := d(u_1(j), j)$.

Основной цикл алгоритма начинается с подсчета вычислительными процессами части вектора оценок Лагранжа $\rho_i(\lambda^k)$, $i \in I$, на основании столбцов из своего блока. Поскольку каждый столбец матрицы расстояний отсортирован по возрастанию и хранится в памяти не полностью, то затруднительно определить заранее, какие компоненты вектора оценок Лагранжа на каком из процессов будут отличны от нуля, что вынуждает каждый Slave-процесс передавать управляющему два вектора, один из которых содержит полученные значения оценок Лагранжа, а другой — соответствующие им индексы. Другой информацией, передаваемой на этом этапе, является сумма множителей $\sum_{i=j_q}^{j_q+n_q} \lambda_i^k$ необходимая для подсчета значения двойственной функции Лагранжа $\theta(\lambda^k)$, а следовательно и нижней оценки оптимального значения BLB.

Получив эти данные, управляющий процесс собирает полный вектор оценок, сортирует его, подсчитывает $\theta(\lambda^k)$ и текущую нижнюю оценку BLB, а также находит множество $T(\lambda^k)$, проверяя затем критерий остановки. При выполнении критерия всем вычислительным процессорам отправляется сообщение о завершении работы, в противном случае каждый вычислительный процесс получает сообщение, содержащее множество индексов $T(\lambda^k)$.

На основе полученного множества $T(\lambda^k)$ вычислительные процессы находят необходимый вектор $y(\lambda^k)$. Подсчитав вектор $y(\lambda^k)$ и используя свой блок столбцов, каждый Slave-процесс вычисляет часть вектора субградиента $g_j(\lambda^k)$, $j = j_q, \ldots, j_q + n_q$, а также норму $||g(\lambda^k)||_2$, которую отсылает управляющему процессу. Просуммировав части нормы вектора субгради-

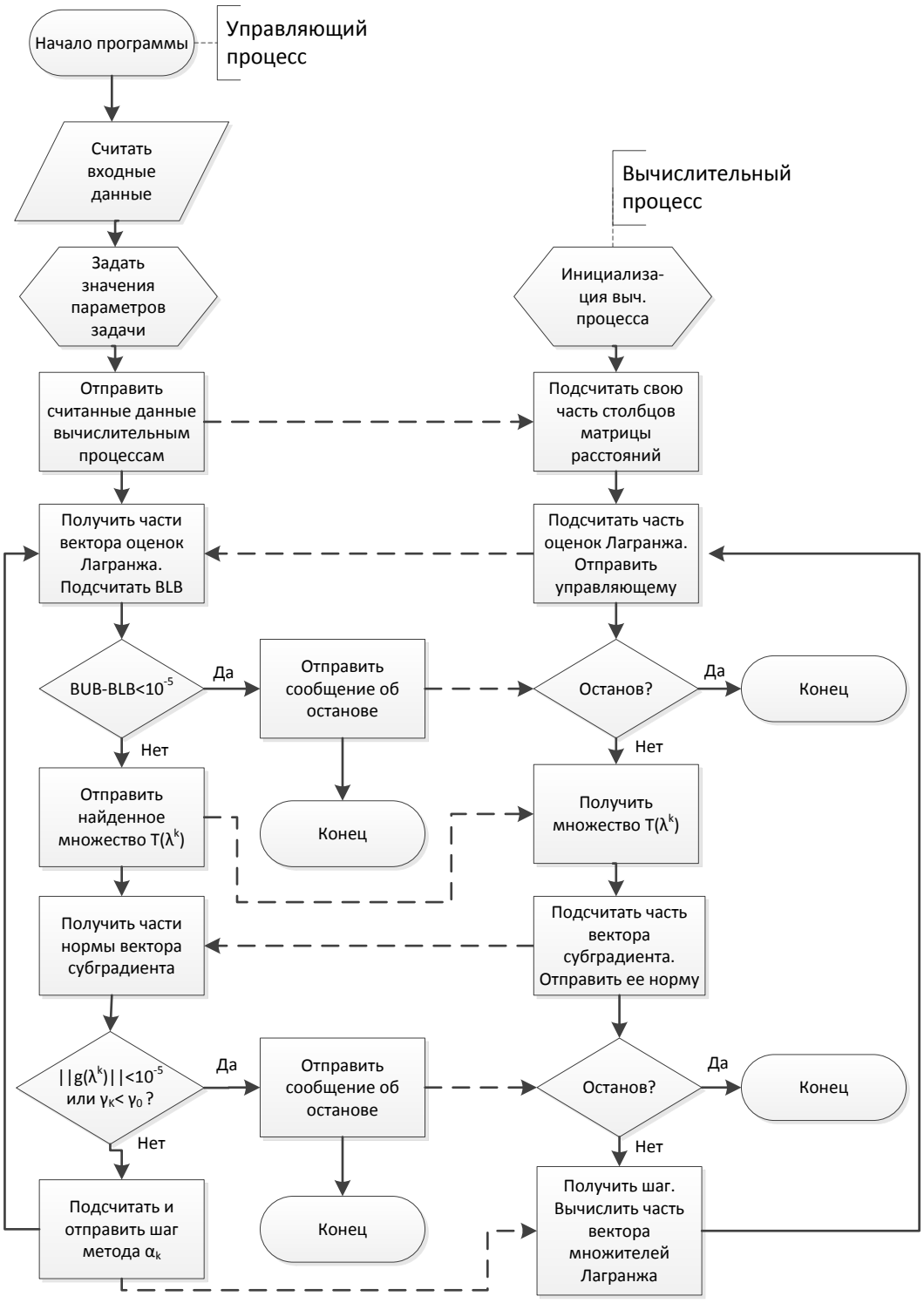

Рис. 1.3. Блок-схема параллельного алгоритма поиска нижней оценки оптимального значения в задаче о p-медиане

ента, Master-процесс вычисляет параметр γ_k, а также проверяет критерий остановки (см. последовательный алгоритм 1.3). Если критерий выполняется, то каждому вычислительному процессу отправляется сообщение о завершении работы алгоритма. В противном случае управляющий процесс подсчитывает и отправляет всем Slave-процессам подсчитанный шаг α_k.

Наконец, каждый вычислительный процесс, используя полученный шаг, может подсчитать новые значения своей части множителей Лагранжа $\lambda_j^{k+1} = \lambda_j^k + \alpha_k\, g(\lambda^k)$, $j = j_q, \ldots, j_q + n_q$ и перейти на следующую итерацию параллельного алгоритма.

Результаты интенсивных вычислительных экспериментов, полученные для представленной параллельной схемы, показали, что при увеличении размерности задачи эффективность алгоритма существенно снижается, приводя зачастую к увеличению времени вычисления даже по сравнению с последовательным алгоритмом [161]. Причиной этому служит тот факт, что на каждой итерации вычислительные процессы на этапе подсчета оценок Лагранжа вынуждены отправлять управляющему процессу существенный объем данных в виде двух массивов, размерность которых для каждого Slave-процесса, вообще говоря, неизвестна, однако в худшем случае может составлять m для каждого параллельного процесса. Соответственно, с ростом размерностей задачи и количества вычислительных процессов происходит замедление работы алгоритма в связи с необходимость Master-процесса обработать и принять все передаваемые ему сообщения, объем которых существенен. Поскольку без найденного множества $T(\lambda^k)$ Slave-процессы не могут продолжать вычисления, им приходится ожидать сначала своей очереди на передачу, а затем отправку сообщений всеми остальными процессами и их обработку управляющим.

Для снижения нагрузки на Master-процесс и ускорения работы алгоритма была разработана схема иерархической (каскадной) сборки оценок Лагранжа и суммы множителей $\sum\limits_{i=j_q}^{j_q+n_q} \lambda_i^k$, $q = 1, \ldots, s$, суть которой заключается в следующем. Все количество процессов MPI-приложения последовательно разбивается на блоки фиксированной длины. Внутри каждого блока выделяется один главный процесс, называемый Submaster, отвечающий за сбор данных внутри своего блока. В реализованном варианте схемы таким процессом назначался имеющий наименьший номер в блоке. Далее все множество блоков процессов разбивается на уровни в виде корневого дерева, листья которого представляют собой блоки самого нижнего уровня. В качестве параметров такой схемы выступает желаемое количество уровней иерархии и размер создаваемых блоков. Отметим, что вопреки классическому определению уровня вершин дерева [113] мы будет полагать для удобства изложения, что корень находится на уровне

1. Сбор данных в схеме происходит последовательно: на первой стадии каждый Submaster-процесс производит прием и обработку сообщений от вычислительных процессов из своего блока; после этого каждый Submaster отправляет собранную информацию Submaster-процессу, стоящему в дереве иерархии на один уровень выше, и так далее, пока данные не будут переданы в корень дерева управляющему процессу. На рис. 1.4 представлен пример иерархии процессов, состоящей из трех уровней с размером блока, равного 10.

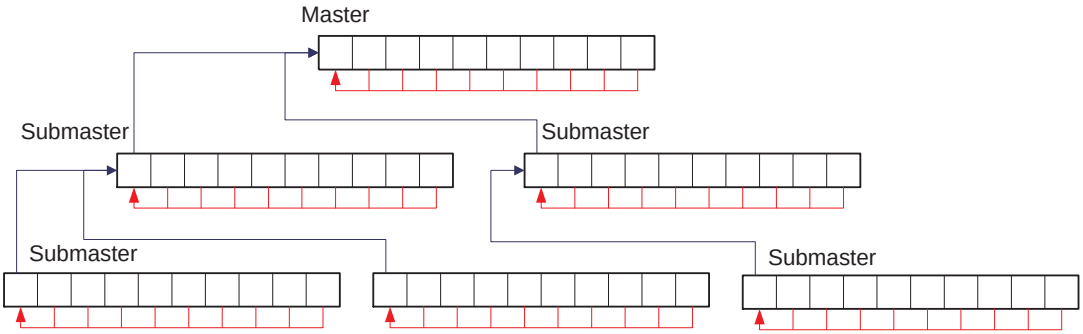

Рис. 1.4. Пример дерева иерархии

Отметим, что формирование дерева в зависимости от выбранных параметров осуществляется по следующему принципу. На вершине иерархии (в корне дерева) всегда находится единственный блок, содержащий в качестве Submaster'а управляющий процесс, чей номер, очевидно, равен 0. Количество блоков на каждом уровне иерархии, или степень вершин дерева, за исключением листьев и их родителей определяется по следующей формуле:

$$\deg(v) = \min\{k \in \mathbb{N} : \sum_{i=0}^{l-1} k^i \geq b\},$$

где l — количество уровней иерархии, а b — общее количество блоков процессов. Отметим, что степень вершин предпоследнего уровня иерархии, или родительских вершин листьев, не задается по такому правилу из-за возможной неравномерной загрузки Submaster-процессов. Действительно, может возникнуть ситуация, представленная на рисунке 1.5, в которой, несмотря на то, что построенная иерархия блоков имеет три уровня, часть вершин дерева второго уровня не имеет потомков. Для избежания таких ситуаций блоки последнего уровня распределяются между Submaster-процессорами предпоследнего уровня по возможности равномерно, таким образом, чтобы $|\deg(u) - \deg(v)| \leq 1$, где u, v — любые вершины дерева иерархии, имеющие один уровень.

Отметим, что процедура построения иерархии блоков выполняется управляющим процессом до начала основного цикла параллельного алгоритма. Для формирования иерархии каждый вычислительный процесс получает информацию касательно его места в ней, состоя-

щую только из двух параметров: количество получаемых сообщений, а также номер процесса, которому необходимо отправить полученную и обработанную информацию.

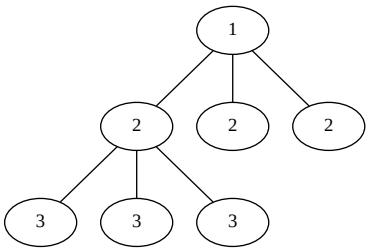

Рис. 1.5. Пример неравномерной загрузки Submaster-процессов

1.4. Вычислительный эксперимент

Представленная выше схема параллельного субградиентного алгоритма была реализована на языке C++ с использованием одной из самых известных реализаций MPI — библиотеки MPICH 1.2.7 и протестирована на вычислительном кластере «Академик В.М. Матросов» (ИДСТУ СО РАН)[1], имеющем следующие характеристики: 220 16-ядерных процессоров AMD Opteron 6276 2.3 GHz «Interlagos» на основе x86_64-микроархитектуры «Bulldozer» (всего 3520 процессорных ядра); пиковая производительность 33.7 TFlops (трлн. оп. с пл. точк. в сек.); суммарный объем оперативной памяти на узлах — 7040GB.

В качестве тестовых примеров использовались две задачи большой размерности, взятые из статьи [164], размерностью 36000 и 89000 вершин, аналогичные тем, что рассматривались в работах [56, 164, 177, 178], а также одна задача сверхбольшой размерности с 1000000 вершин (10^{12} переменных), искусственно сгенерированная в соответствии с методом, предложенным в работах [278, 279].

Верхняя оценка BUB для задач размерностью 36000 и 89000 вершин, необходимая для подсчета шага субградиентного алгоритма α_k, бралась из статьи [56]. Количество активных элементов m_0^j в j-м столбце отсортированной матрицы расстояний выбиралось таким образом, чтобы их суммарное количество в оперативной памяти для одного процессорного ядра не превосходило 800Mb. При необходимости число активных элементов в одном столбце увеличивалось на $m_1 = 100$. К было отмечено ранее, шаг алгоритма α_k в начале полагался равным 2 и уменьшался в два раза, если на протяжении тридцати итераций не происходило улучшения лучшей нижней оценки BLB.

[1] Иркутский суперкомпьютерный центр СО РАН [Электронный ресурс]: сайт. – Иркутск: ИДСТУ СО РАН. – URL: http://hpc.icc.ru (дата обращения: 03.06.2016).

Как известно, для оценки эффективности параллельного алгоритма используются различные характеристики [147, 215, 249], среди которых наиболее известными являются параллельное ускорение и эффективность параллельного алгоритма. Параллельное ускорение вычисляется по формуле $\frac{t_s}{t_i}$, где t_s — время работы последовательного алгоритма, а t_i — время, затраченное на решение задачи параллельным алгоритмом с использованием i процессоров. Отметим, что в качестве t_s берется время работы лучшего известного последовательного алгоритма, если же в качестве такового берется время работы параллельного алгоритма на одном процессоре, то говорят об относительном параллельном ускорении. Интересно также заметить, что согласно закону Амдаля наилучшим возможным параллельным ускорением является так называемое линейное ускорение, т.е. увеличивающееся пропорционально увеличению числа задействованных процессоров, т.е. $t_s/t_i \approx i$ для любого $i = 1, \ldots, s$. Однако на практике зачастую возникают аномалии, когда параллельный алгоритм достигает сверхлинейного ускорения. Такие ситуации, например, характерны для алгоритмов обхода дерева, в частности, параллельных реализаций метода ветвей и границ [147, 215] или рандомизированных алгоритмов [173]. Эффективностью параллельного алгоритма называют величину, вычисляемую по формуле $\frac{t_s}{it_i}$, т.е. определяющую параллельное ускорение относительно числа процессоров.

Результаты вычислительных экспериментов касательно ускорения и эффективности параллельного алгоритма при различном количестве процессорных ядер для задач из библиотеки TSPLIB представлены в таблицах 1.1, 1.2. Отметим, что в столбце «Время» представлено полное время работы параллельного алгоритма, включая процедуру считывания входных данных, а также подсчет матрицы расстояний. Время работы параллельного алгоритма в зависимости от количества процессорных ядер проиллюстрировано на графиках (см рис. 1.6, 1.7), в то время как параллельное ускорение отражено на рис. 1.8, 1.9.

Анализируя полученные результаты, можно заметить, что достаточно хорошее параллельное ускорение в обоих задачах, то есть близкое к линейному, достигнуто лишь при небольшом увеличении числа используемых процессорных ядер: для восьми в случае задачи с 36000 (табл. 1.1) узлов и для шестнадцати для примера с 89600 (табл. 1.2) узлами. При увеличении числа процессорных ядер при запуске параллельной программы рост параллельного ускорения замедляется и практически останавливается при 88 процессорных ядрах для меньшей задачи и 120 для задачи размерности 89600. Максимальное параллельное ускорение, как можно видеть из таблиц, оказалось двадцатикратным для меньшей задачи и более чем тридцатикратным для большего примера. Таким образом можно констатировать, что согласно

Таблица 1.1. Параллельное ускорение и эффективность алгоритма для задачи с 36000 вершинами

Количество ядер	Время, с	Ускорение	Эффективность
1	658.06	1	1
8	150.36	4.377	0.547
16	84.82	7.758	0.485
24	66.41	9.909	0.413
32	53.40	12.323	0.385
40	47.03	13.992	0.350
48	43.33	15.187	0.316
56	40.43	16.277	0.291
64	38.48	17.101	0.267
72	37.17	17.704	0.246
80	35.72	18.423	0.230
88	34.51	19.069	0.217

закону Амдаля разработанный параллельный алгоритм содержит относительно небольшую часть, выполняющуюся последовательно, из чего можно сделать вывод о достаточно хорошей сильной масштабируемости алгоритма. Сравнивая результаты для обеих тестовых задач, можно также отметить явный рост эффективности алгоритма и соответственно параллельного ускорения с ростом размерности: почти десятикратное ускорение для одного и того же числа процессорных ядер. Таким образом, можно говорить и о хорошей слабой масштабируемости разработанного параллельного алгоритма, а также о возможности его применения для решения задач сверхбольшой размерности.

Столь высокая эффективность разработанного параллельного алгоритма для задач размерности в несколько десятков тысяч узлов дала основание для проведения тестирования на примере значительно большей размерности, содержащем один миллион вершин (10^{12} переменных) и сгенерированным по правилу, предложенному в работах [278, 279]. Вычислительный эксперимент проводился на 80-ти процессорных ядрах вычислительного кластера. Относительная погрешность известной верхней оценки относительно нижней составило порядка 0.346 процента. Общее время вычисления, включая подсчет матрицы расстояний, составило 22101 секунд (чуть более шести часов).

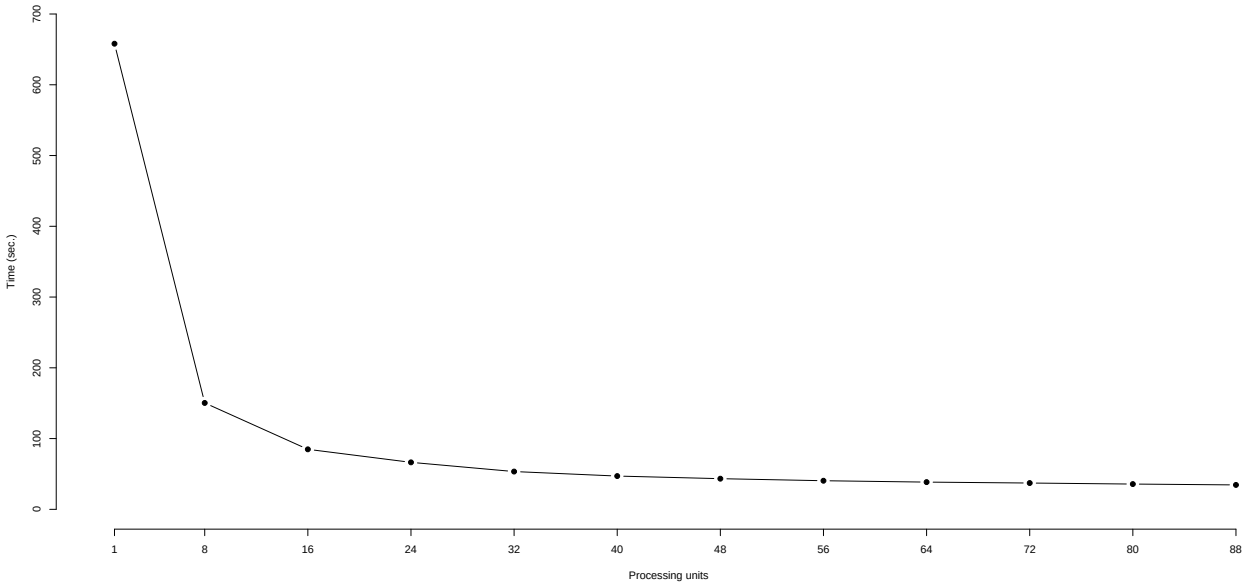

Рис. 1.6. Время работы алгоритма в зависимости от числа вычислительных ядер для задачи с 36000 вершин и двухуровневой иерархией с размером блока 8

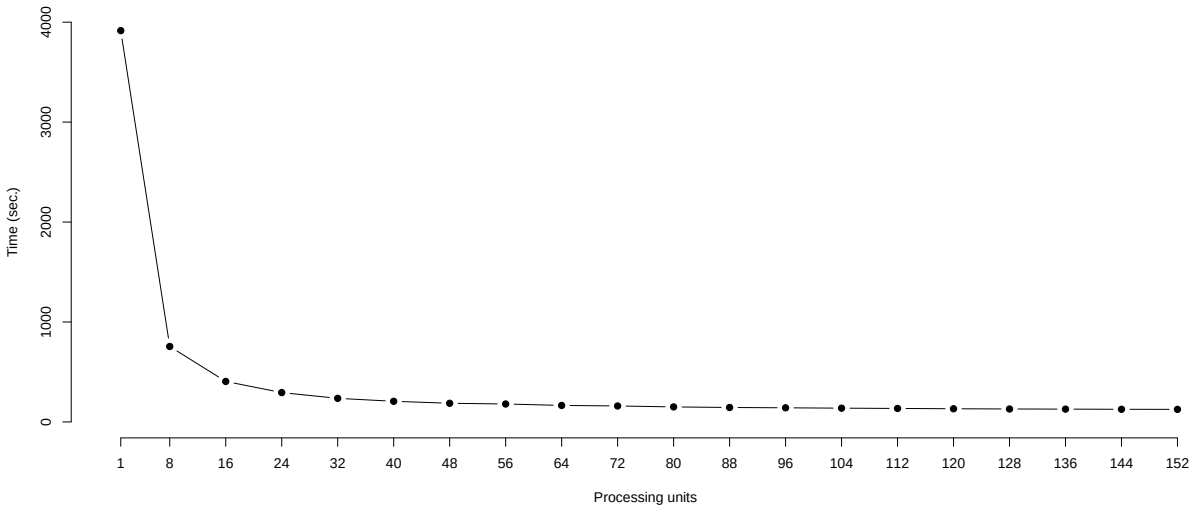

Рис. 1.7. Время работы алгоритма в зависимости от числа вычислительных ядер для задачи с 89600 вершин и двухуровневой иерархией с размером блока 8

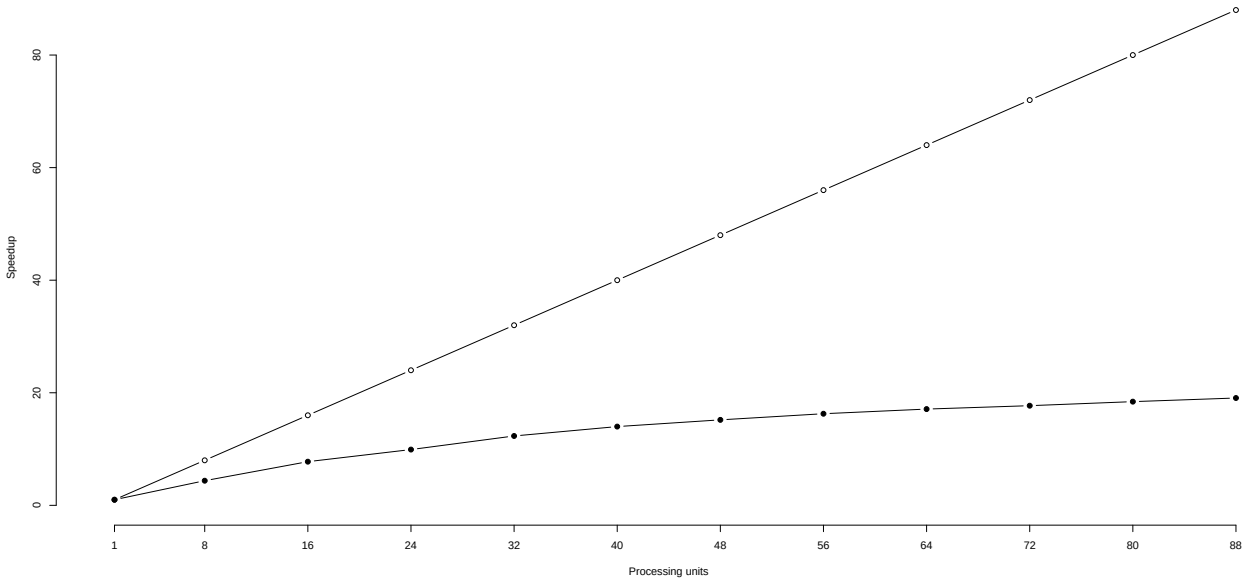

Рис. 1.8. Параллельное ускорение для задачи с 36000 вершин и двухуровневой иерархией с размером блока 8

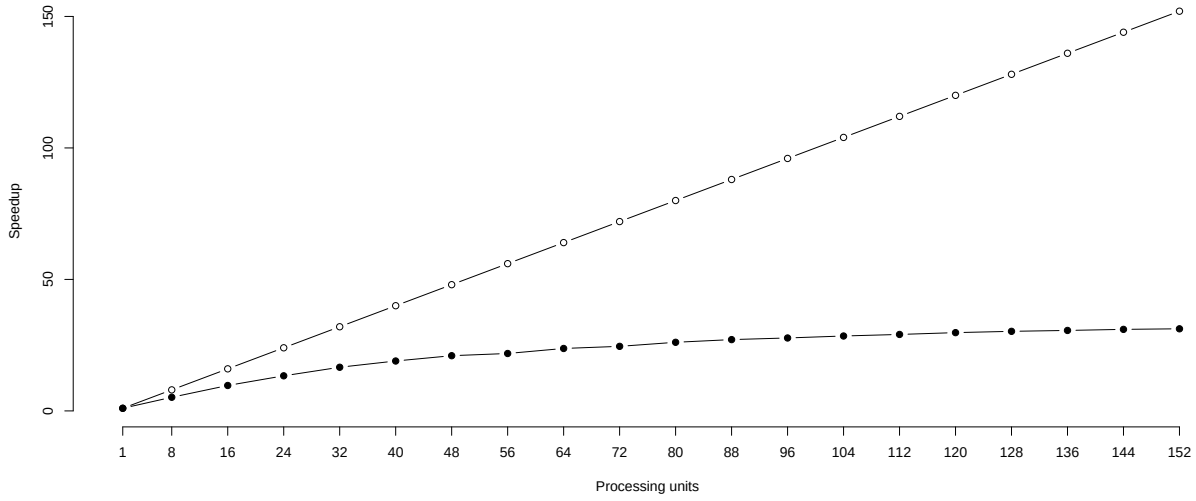

Рис. 1.9. Параллельное ускорение для задачи с 89600 вершин и двухуровневой иерархией с размером блока 8

Таблица 1.2. Параллельное ускорение и эффективность алгоритма для задачи с 89600 вершинами

Количество ядер	Время, с	Ускорение	Эффективность
1	3915.67	1	1
8	754.83	5.187	0.648
16	405.02	9.668	0.604
24	293.43	13.344	0.556
32	235.88	16.600	0.519
40	206.32	18.979	0.474
48	186.52	20.993	0.437
56	179.34	21.834	0.390
64	164.90	23.746	0.371
72	159.49	24.551	0.341
80	150.08	26.091	0.326
88	144.41	27.115	0.308
96	141.18	27.735	0.289
104	137.46	28.486	0.274
112	134.67	29.076	0.260
120	131.55	29.766	0.248
128	129.39	30.263	0.236
136	127.96	30.601	0.225
144	126.29	31.005	0.215
152	125.41	31.223	0.205

1.5. Основные результаты первой главы

В данной главе приведен достаточно подробный обзор теоретических результатов, связанных с исследованием свойств релаксаций Лагранжа для задач целочисленного линейного программирования в общем виде, а также краткий обзор методов решения лагранжевой двойственной задачи с целью поиска двойственной (нижней) оценки оптимального значения. Дана постановка задачи о p-медиане в виде задачи целочисленного линейного программирования, рассмотрен один из известных способов построения релаксации Лагаранжа в такой задаче. Также дано описание одного известного метода поиска приближенных решений в задаче о p-медиане [56], относящегося к так называемым лагранжевым эвристикам. Данный подход основан на комбинации метода релаксаций Лагранжа для поиска последовательности

нижних оценок оптимального значения задачи, включающего в себя построение релаксации и поиск решения в соответствующей двойственной задаче с помощью специального субградиентного алгоритма, а также ядровой эвристики для поиска верхних оценок оптимального значения. Приведенные в настоящей главе теоретические результаты будут использованы в дальнейшем изложении при исследовании нелинейного варианта задачи о p-медиане.

Также в настоящей главе для задачи о p-медиане разработана схема распараллеливания известного алгоритма для решения Лагранжевой двойственной задачи. Произведена реализация и тестирование разработанного параллельного алгоритма. На основании проведенного вычислительного эксперимента сделан вывод, что предлагаемый параллельный эвристический метод является достаточно эффективным и масштабируемым при увеличении размерности задачи, позволяя успешно оперировать примерами большой размерности, содержащих зачастую триллионы переменных. Более того, процедура иерархической или каскадной сборки оценок Лагранжа обеспечивает дополнительную производительность алгоритма для больших тестовых задач, существенно ускоряя время его работы.

Глава 2

Нелинейный вариант задачи о p-медиане

В настоящей главе рассматривается одно нелинейное обобщение задачи о p-медиане, где количество открываемых предприятий p является целочисленной переменной, а в целевой функции присутствует дополнительное слагаемое, ограничивающее число открываемых предприятий, которое задается некоторой, вообще говоря, нелинейной функцией. Для данной модификации задачи о p-медиане предлагаются и исследуются два вида релаксации Лагранжа, а также доказывается ряд теоретических результатов, касающихся повышения эффективности подсчета значения двойственной функции Лагранжа с учетом свойств указанной нелинейной функции. Для поиска приближенных решений в нелинейной задаче о p-медиане предлагается подход, являющийся обобщением представленного в первой главе алгоритма 1.4, включающего в себя метод релаксаций Лагранжа и так называемую ядровую эвристику. В конце главы приводятся результаты вычислительного эксперимента для серии известных тестовых задач, демонстрирующие эффективность предложенного алгоритма для задач большой размерности.

2.1. Постановка задачи

Рассматриваемая в первой главе постановка задачи о p-медиане является классической. Ее главное отличие от схожей с ней простейшей задачи размещения состоит в наличии фиксированного числа открываемых предприятий, а также в отсутствии заданной стоимости на открытие предприятия в каждом из имеющихся возможных пунктов.

Представим теперь, что в рассматриваемой задаче о p-медиане количество открываемых предприятий не фиксировано, что в некоторых случаях является более оправданным с практической точки зрения. Действительно, такая ситуация может возникнуть при размещении относительно дешевых объектов, таких как контейнеры для мусора, почтовые ящики, таксофоны, газетные киоски и т.д. В данном случае при увеличении числа предприятий стоимость их обслуживания сокращается. Так, например, при размещении двух почтовых ящиков стоимость их обслуживания увеличится по сравнению с одним, но можно ожидать, что такое увеличение не будет двукратным. Другим примером может быть одно из приложений задачи о p-медиане в области кластеризации данных, в котором количество искомых кластеров зачастую неизвестно [266], или в так называемой задаче оптимального замеще-

ния, возникающей в автомобилестроении при выборе набора электропроводки автомобиля для установки дополнительных опций [5, 55, 83], в то время как количество конфигураций электропроводки также не всегда является фиксированным. В таких случаях имеет смысл рассматривать модифицированную задачу о p-медиане, где p является целочисленной переменной, принимающей значения из множества $P \triangleq \{1, \ldots, m\}$, а в целевой функции, помимо суммарных затрат на обслуживание всех клиентов, присутствует дополнительное слагаемое $\phi(p)$, задающее штраф на количество открываемых предприятий. Таким образом, в данной главе будет рассматриваться следующая задача комбинаторной оптимизации:

$$\min_{S \subseteq I,\, p \in P} \Big\{ \sum_{j \in J} \min_{i \in S} d_{ij} + \phi(p) : \ |S| = p \Big\}, \tag{2.1}$$

где, напомним, I и J — множества возможных пунктов размещения предприятий и клиентов соответственно, d_{ij} — стоимость доставки единицы продукции из предприятия в пункте i клиенту j (транспортные расходы), а функция $\phi(p)$ в общем случае предполагается нелинейной.

Задачи размещения с нелинейными компонентами в целевой функции исследуются довольно давно. Фактически такие задачи рассматривались с периода самого зарождения теории дискретных задач размещения [63]. Интерес к данной области вызван прежде всего тем, что классические линейные постановки дискретных задач размещения являются довольно общими и зачастую не могут быть в полной мере использованы при моделировании реальных экономических ситуаций. Наверное, важнейшей задачей в теории размещения, фактически являющейся базовой при моделировании различных обобщений, является простейшая задача, которая в самом общем виде может быть представлена в виде следующей задачи комбинаторной оптимизации:

$$\min_{S \subseteq I} \Big\{ \sum_{j \in J} \min_{i \in S} (g_i + d_{ij}) + \sum_{i \in N} f_i \Big\},$$

где f_i — стоимость размещения предприятия в пункте i, а g_i — цена производства одной единицы продукции на предприятии, открытом в пункте i (стоимость эксплуатации).

В литературе известно достаточно много работ, посвященных исследованию и разработке методов решения различных вариантов представленной задачи размещения, в которых стоимость открытия предприятия, производства или доставки продукции клиенту является не фиксированной, а задается в виде некоторой, в общем случае, нелинейной функции. Необходимость такого рода модификаций следует из логических соображений о наличии таких реальных экономических факторов как эффект масштаба (т.е. экономия, обусловленная ростом масштабов производства), различия в стоимости размещения предприятий, от-

рицательный эффект роста масштабов производства и т.д. В частности, если в качестве производственной или эксплуатационной стоимости выбирать некоторую вогнутую (квазивыпуклую) функцию, что соответствует постепенному уменьшению затрат на производство единицы продукции с увеличением выхода, то возможно моделирование известного экономического эффекта, называемого положительным эффектом масштаба [25]. Среди первых работ, посвященных задачам размещения с учетом положительного эффекта масштаба, следует выделить [128], где рассматривалась задача размещения складов снабжения, затраты на установку и обслуживание которых задавались некоторыми вогнутыми функциями, зависящими от размера самого склада. Похожая постановка рассматривалась в работе [276], в которой исследовалась задача размещения с ограничениями на мощность производства, а также с возможностью размещения более одного предприятия в каждом из пунктов, причем стоимость размещения каждого предприятия выражалось некоторой нелинейной функцией от его размера. Для поиска приближенных решений в такой задаче была предложена лагранжева эвристика. Варианты простейшей задачи размещения, в которых стоимость эксплуатации или размещения каждого предприятия представлялась в виде кусочно-линейной вогнутой функции или вогнутой функции общего вида, исследовались в работах [118] и [117] соответственно. Для поиска решения в таких постановках авторами был разработан метод ветвей и границ. В работе [263] рассматривалась более сложная задача размещения предприятий, в которой, помимо производственной стоимости, транспортные расходы также выражались в виде некоторой вогнутой функции от объема доставляемой клиенту продукции. Стоит отметить задачу об оптимальной загрузке оборудования предприятия, рассматриваемую в [99], и единую задачу размещения предприятий и планирования производства [258], в которых стоимость производства задавалась в виде вогнутых функций. Для данных задач авторами были предложены методы генерации строк, а также ветвей и оценок соответственно. Наконец, в статье [201] авторы рассматривали задачу создания системы распределения продукции с учетом положительного эффекта масштаба транспортных затрат, для решения которой применялась жадная эвристика.

С другой стороны, в литературе, посвященной исследованию дискретных задач размещения, встречается достаточно работ, в которых вместо вогнутых целевых функций рассматриваются случаи выпуклых. Отметим, что с помощью такого рода постановок возможно моделирование отрицательного эффекта масштаба, т.е. увеличения издержек на производство, транспортировку или размещение новых предприятий по мере роста их количества (масштабов). Такой эффект может возникать, например, при чрезмерном использовании ресурсов (закон убывающей отдачи) или быть обусловлен спецификой отрасли, проблемой контроля

и координации большого числа предприятий и т.д. [25, 207]. Известно также [207], что задачи размещения, часто возникающие в области дизайна сетей связи и проектирования систем массового обслуживания, в которых учитывается время ожидания и накопление, обычно имеют выпуклую целевую функцию. Так, например, в работе [71] исследовалась мультипродуктная задача производства, хранения и обслуживания клиентов из нескольких предприятий с выпуклой целевой функцией, для решения которой авторами предложен метод ветвей и границ на основе линейной и лагранжевой релаксаций. В работе [169] рассматривалась задача размещения с ограничением на мощность производства, в которой стоимость производства задается кусочно-линейной выпуклой функцией. Предложены четыре формулировки рассматриваемой нелинейной задачи размещения, для поиска решений в которых был использован метод ветвей и границ.

Еще один класс нелинейных задач размещения составляют те, в которых целевые функции не являются ни вогнутыми, ни выпуклыми. Так, в статье [175] рассматривается задача размещения с ограничениями на мощность производства, в которой цена производства в каждом из возможных пунктов имеет лестничную структуру, т.е. является фиксированной или линейной для разных уровней производства. Предложен метод решения, включающий в себя выпуклую кусочную линеаризацию и декомпозицию Бендерса. Другим интересным примером является работа [269], в которой рассматривалась задача размещения скотобоен, имеющая S-образную целевую функцию, т.е. вначале вогнутую, а затем выпуклую.

В настоящей главе рассматривается задача о p-медиане, в целевой функции которой имеется дополнительное нелинейное слагаемое, задающее суммарную стоимость открытия p предприятий. Впервые подобного рода формулировка, судя по всему, была предложена в работе [218]. В данной статье рассматривалась простейшая задача размещения, в которой суммарная стоимость размещения предприятий включала в себя выпуклую неубывающую функцию от их числа. Авторами был предложен алгоритм поиска решений в такой задаче, представляющий из себя по сути метод бисекции по p, в котором на каждом шаге производится решение параметризованной простейшей задачи размещения с помощью двойственного алгоритма DUALOC [124] или обобщенной задачи о p-медиане посредством так называемого NDA-алгоритма [220] с целью нахождения подотрезка, содержащего оптимальное значение p^*, для следующей итерации метода. В качестве области приложения такой нелинейной формулировки авторы рассматривали задачу оптимального размещения копий файлов данных в некоторой компьютерной сети. Если предположить, что каждое предприятие представляет собой копию некоторого файла данных в компьютерной сети, то цена обновления данных в копиях файла в общем случае растет быстро и нелинейно относительно числа копий.

В дальнейшем такая постановка задачи исследовалась в работе [189], где также использовался метод декомпозиции исходной задачи на серию простейших задач размещения с использованием метода постепенного сокращения интервала, содержащего оптимальное число предприятий p^*, для одномерной релаксации Лагранжа [148, 176]. Разработанный алгоритм позволил получить точные решения для модифицированной нелинейной простейшей задачи размещения, в которой на функцию от числа открытых предприятий не накладывается дополнительных условий о выпуклости и монотонности. Численные результаты показали, что данный алгоритм, ко всему прочему, позволяет находить решения в рассматриваемой задаче в два раза быстрее для случая выпуклой функции, чем представленный в работе [218]. В данной работе было предложено также одно приложение задачи в области телекоммуникаций, а именно в задаче оптимального размещения пунктов управления услугами в так называемой интеллектуальной сети. Администрирование доступа каждого пользователя к клиентским данным на каждом пункте управления услугами может быть выражено с помощью упомянутого ранее нелинейного компонента в суммарной стоимости размещения пунктов.

Таким образом, рассматриваемая в настоящей главе нелинейная задача о p-медиане (2.1) является схожей с уже исследованным в работах [189, 218] нелинейным вариантом простейшей задачи размещения. Однако, в исследуемой задаче в отношении функции $\phi(\cdot)$ в общем случае не делается никаких дополнительных предположений. Таким образом, в нелинейном варианте задачи о p-медиане ищется некоторое компромиссное решение между количеством открываемых предприятий и ценой обслуживания клиентов. Отметим, что если функция $\phi(\cdot)$ является линейной, то рассматриваемая задача является частным случаем простейшей задачи размещения с общей стоимостью открытия предприятий. Более того, для некоторых частных случаев выбора функции $\phi(\cdot)$ задача (2.1) сводится к классической задаче о p-медиане при фиксированном $p = p^*$. Так, если предположить, что $Z(p^*)$ задает оптимальное значение в задаче о p-медиане, в которой количество открываемых предприятий равно p^*, то выбирая в качестве функции $\phi(\cdot)$ любую функцию, такую что $\phi(p) = 0$ для всех $p \leq p^*$ и $\phi(p) > Z(p^*)$ для всех $p > p^*$, можно заметить, что поскольку величины d_{ij} неотрицательны, то оптимальное решение задачи (2.1) состоит из в точности p^* предприятий, причем оно также является решением и в задаче о p-медиане при $p = p^*$.

Как было отмечено ранее, с помощью нелинейной неубывающей функции $\phi(\cdot)$ естественным образом моделируется положительный или отрицательный эффект масштаба при размещении предприятий. Такого рода функции могут возникать и в задаче оптимального замещения в автомобильной промышленности в случае быстрого роста цены производства, хранения и установки различных конфигураций электропроводки по сравнению с количеством возмож-

ных вариантов. Также стоит заметить, что представленные в [189, 218] задачи размещения в сфере телекоммуникаций также являются естественными приложениями рассматриваемого нелинейного варианта задачи о p-медиане.

Нелинейная задача о p-медиане может быть представлена в виде задачи целочисленного программирования с помощью набора булевых переменных y_i, x_{ij},введенных в первой главе. Напомним, что y_i принимает значение 1, если узел i является медианой, 0 в противном случае. Переменная x_{ij} равна 1, если узел i представляет собой медиану, и j присоединен к i выходящей из i дугой, 0 в противном случае. Тогда нелинейная задача о p-медиане может быть записана следующим образом:

$$\min_{(x,y,p)} \sum_{(i,j)\in A} d_{ij}x_{ij} + \phi(p), \tag{2.2}$$

$$\sum_{i\in\delta^-(j)} x_{ij} + y_j = 1, \qquad j \in I, \tag{2.3}$$

$$x_{ij} \le y_i, \qquad i \in I, j \in \delta^+(i), \tag{2.4}$$

$$\sum_{i\in I} y_i = p, \tag{2.5}$$

$$y_i \in \{0,1\}, \qquad i \in I, \tag{2.6}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \tag{2.7}$$

$$p \in P \triangleq \{p \in \mathbb{Z} :\ 1 \le p \le m\}. \tag{2.8}$$

Ограничения (2.3)–(2.7) имеют тот же смысл что и в классической задаче о p-медиане. Отметим, что в представленной модифицированной задаче p также является переменной, принимающей целые значения из множества P. Обозначим также $\Psi_* = \min\limits_{(x,y,p)\in X_2} \Psi(x,y,p)$.

Поскольку p принимает только конечное число значений, то для комбинаторной задачи (2.1) возможно построение эквивалентной целочисленной постановки, полученной посредством линеаризации функции $\phi(\cdot)$. Действительно, пусть для каждого $k \in P$ $\phi_k = \phi(k)$ и пусть z_k представляет собой булеву переменную, принимающую значение 1, если открыто в точности k предприятий, и 0 в противном случае. Тогда с использованием введенных обозна-

чений, задача (2.1) может быть записана в следующей эквивалентной форме:

$$\min_{(x,y,z)} \sum_{(i,j)\in A} d_{ij}x_{ij} + \sum_{k=1}^{n} \phi_k z_k, \tag{2.9}$$

$$\sum_{i\in\delta^-(j)} x_{ij} + y_i = 1, \qquad j \in J, \tag{2.10}$$

$$x_{ij} \le y_i, \qquad i \in I, j \in \delta^+(i), \tag{2.11}$$

$$\sum_{i\in I} y_i = \sum_{k=1}^{n} k z_k, \tag{2.12}$$

$$\sum_{k=1}^{m} z_k = 1, \tag{2.13}$$

$$y_i \in \{0,1\}, \qquad i \in I, \tag{2.14}$$

$$x_{ij} \in \{0,1\}, \qquad (i,j) \in A, \tag{2.15}$$

$$z_k \in \{0,1\}, \qquad k = 1,\dots,m. \tag{2.16}$$

Помимо дополнительной переменной, в данной формулировке присутствует линейное ограничение (2.12), которое гарантирует, что число открытых предприятий равно k, тогда как ограничение (2.13) налагает условие о том, что может быть выбрано только одно значение $k = 1,\dots,m$ в качестве числа открытых предприятий.

2.2. Релаксации Лагранжа для нелинейного варианта задачи о p-медиане

Как и в предыдущей главе, вначале рассмотрим метод релаксаций Лагранжа для поиска нижних оценок оптимального значения. Для этого будем строить два вида релаксаций, обобщающие уже известные типы релаксаций на случай, когда число предприятий p является переменной величиной. Первая получается за счет ослабления ограничений (2.3), а вторая — группы ограничений (2.3) и (2.5).

Вначале рассмотрим первый тип, называемый в дальнейшем $\mathcal{R}_1$-релаксацией, в котором ослабляются две группы ограничений. Для этого добавим (2.3) и (2.5) в целевую функцию (2.2) с множителями (λ,π), где $\lambda = (\lambda_1,\dots,\lambda_m)^T \in \mathbb{R}^m$ соответствуют ограничениям (2.3), а $\pi \in \mathbb{R}$ — ограничению (2.5). В этом случае получающаяся двойственная функция

Лагранжа $\theta_1(\lambda, \pi)$ может быть записана в следующем виде:

$$\theta_1(\lambda, \pi) = \min_{(x,y,p)} \Big\{ \sum_{(i,j)\in A} d_{ij} x_{ij} + \sum_{j\in I} \lambda_j \Big(1 - y_j - \sum_{i\in\delta^-(j)} x_{ij}\Big) + \\ + \pi(p - \sum_{i\in I} y_i) + \phi(p) : \; x_{ij} \le y_i, i \in I, j \in \delta^+(i) \Big\}. \tag{2.17}$$

Для краткости записи здесь и далее, если не оговорено противное, будем считать переменные $x_{ij}, y_i,\ i \in I,\ (i,j) \in A$ булевыми, а переменную p — целой величиной из множества P. Будем также полагать, что функция $\phi(\cdot)$ определена и принимает конечные значения на множестве $\operatorname{conv} P$. Как и в первой главе, в дальнейшем будем пользоваться определением срезки $a^- \triangleq \min\{0, a\}$.

Отметим, что поскольку функция $\phi(p)$ является дискретной и принимает значения лишь в конечном числе точек, то, вообще говоря, к ней не может быть применено определение выпуклости. Одним из выходов из подобной ситуации является применение теории дискретного выпуклого анализа [226], однако чаще всего под термином выпуклой смешанной задачи нелинейного целочисленного программирования понимается задача, непрерывная релаксация которой является выпуклой [221]. Также зачастую полагают, что некоторая дискретная функция является выпуклой, если в каждой точке ее области определения существует субградиент [189, 220]. Можно также заметить, что $\phi(\cdot)$ является выпуклой (вогнутой) на целочисленной сетке $K = \{k_1, k_1 + 1, \ldots, k_2 - 1, k_2\}$ тогда и только тогда, когда кусочно-линейная функция на узлах сетки K, равно как и интерполирующая функция на интерполяционной сетке K, являются выпуклыми (вогнутыми) на интервале $[k_1, k_2]$.

Как и в случае рассмотренной в первой главе релаксации для классической линейной постановки задачи о p-медиане, значения двойственной функции $\theta_1(\lambda, \pi)$ и прямых переменных для некоторого заданного набора множителей (λ, π) может быть подсчитано в явном виде. Для этого, введем для каждого узла $i \in I$ орграфа $G(I, A)$, как и в первой главе для линейной задачи о p-медиане, величины $\zeta_i(\lambda, \pi) = \sum\limits_{j\in\delta^+(i)} (d_{ij} - \lambda_j)^- - \lambda_i - \pi$, являющиеся оценками Лагранжа для переменных y_i. С использованием введенных обозначений справедлив следующий результат.

Теорема 1. *Для любых множителей $(\lambda, \pi) \in \mathbb{R}^m \times \mathbb{R}$ справедливо*

$$\theta_1(\lambda, \pi) = \sum_{i\in I} \zeta_i^-(\lambda, \pi) + \min_{p\in P} \Big\{ \pi p + \phi(p) \Big\} + \sum_{j\in I} \lambda_j. \tag{2.18}$$

Причем оптимальные значения прямых переменных в релаксированной задаче могут быть подсчитаны в явном виде:

$$p(\pi) \in \operatorname*{Argmin}_{q\in P} \{\pi q + \phi(q)\};$$

$$y_i(\lambda,\pi)=\begin{cases}1, & \zeta_i(\lambda,\pi)<0,\\ 0, & \textit{в противном случае};\end{cases}$$

$$x_{ij}(\lambda,\pi)=\begin{cases}1, & y_i(\lambda,\pi)=1 \ u\ d_{ij}-\lambda_j<0,\\ 0, & \textit{в противном случае.}\end{cases}$$

Доказательство. Перепишем (2.17) следующим образом:

$$\begin{aligned}\theta_1(\lambda,\pi)=\min_{(y,p)}\Big\{\min_x\Big\{\sum_{i\in I}\sum_{j\in\delta^+(i)}(d_{ij}-\lambda_j)x_{ij}:\ x_{ij}\le y_i\Big\}-\\ -\sum_{i\in I}\lambda_i y_i+\pi(p-\sum_{i\in I}y_i)+\phi(p)\Big\}+\sum_{j\in I}\lambda_j.\end{aligned}\tag{2.19}$$

Зафиксируем переменные y_i и предположим, что переменные x_{ij} являются непрерывными, принимающими значения на отрезке $[0,1]$. Рассмотрим релаксированную внутреннюю задачу минимизации по x отдельно:

$$\min_x\Big\{\sum_{i\in I}\sum_{j\in\delta^+(i)}(d_{ij}-\lambda_j)x_{ij}:\ 0\le x_{ij}\le y_i\Big\}.$$

Как нетрудно видеть, в данной задаче необходимо найти минимум линейной при каждом фиксированном y_j функции при параллелепипедных ограничениях. Очевидно, что решение данной задачи может быть записано следующим образом:

$$x_{ij}(\lambda,\pi)=\begin{cases}y_i, & d_{ij}-\lambda_j<0,\\ 0, & d_{ij}-\lambda_j\ge 0.\end{cases}$$

Переменные $x_{ij}(\lambda,\pi)$ принимают только целочисленные значения, поэтому они являются решением и в исходной задаче (2.17) с условием целочисленности x. Теперь подставим полученное решение $x_{ij}(\lambda,\pi)$ в (2.19). Так как нас интересуют только значения $d_{ij}-\lambda_j<0$, уместно воспользоваться введенным ранее определением срезки. В результате получаем следующее выражение:

$$\begin{aligned}\theta_1(\lambda,\pi)&=\min_{(y,p)}\Big\{\sum_{i\in I}\sum_{j\in\delta^+(i)}(d_{ij}-\lambda_j)^-y_i-\sum_{i\in I}\lambda_i y_i+\pi(p-\sum_{i\in I}y_i)+\phi(p)\Big\}+\sum_{j\in I}\lambda_i\\ &=\min_y\Big\{\sum_{i\in I}\Big[\sum_{j\in\delta^+(i)}(d_{ij}-\lambda_j)^--\lambda_i-\pi\Big]y_i\Big\}+\min_p\Big\{\pi p+\phi(p)\Big\}+\sum_{i\in I}\lambda_i.\end{aligned}$$

Рассмотрим отдельно задачу минимизации по y, предполагая, что переменные y_i являются непрерывными величинами на отрезке $[0,1]$:

$$\min_y\Big\{\sum_{i\in I}\Big[\sum_{j\in\delta^+(i)}(d_{ij}-\lambda_j)^--\lambda_i-\pi\Big]y_i:\ 0\le y_i\le 1\Big\}.$$

В данной задаче необходимо минимизировать линейную функцию при параллелепипедных ограничениях. Решение может быть представлено как

$$y_i(\lambda,\pi)=\begin{cases}1, & (d_{ij}-\lambda_j)^- - \lambda_i - \pi < 0,\\ 0, & \text{в противном случае.}\end{cases}$$

Аналогичным образом, подставим $y_i(\lambda,\pi)$ в формулу вычисления значения функции Лагранжа $\theta_1(\lambda,\pi)$, используя определение срезки. В результате получим необходимое выражение (2.18), а также оптимальные значения переменных $(y_i(\lambda,\pi), x_{ij}(\lambda,\pi), p(\pi))$. □

Из полученной формулы можно сделать вывод, что сложность вычисления значения двойственной функции Лагранжа $\theta_1(\lambda,\pi)$ составляет $O(n^2)$, если сложность вычисления функции $\phi(\cdot)$ является константой.

В теореме 1 показано, как при некоторых фиксированных множителях $\lambda\in\mathbb{R}^m$, $\pi\in\mathbb{R}$ может быть найдено оптимальное решение $x(\lambda,\pi)$, $y(\lambda,\pi)$. Однако для того, чтобы найти оптимальное значение $p(\pi)$, равно как и значение двойственной функции Лагранжа, необходимо решить задачу одномерной минимизации

$$\min_{p\in P}\{\pi p+\phi(p)\}. \tag{2.20}$$

Отметим, что оптимальное решение подзадачи (2.20) может быть найдено посредством полного перебора элементов множества P. Однако в некоторых случаях есть возможность избежать этого. Сперва введем следующие обозначения. Для некоторого заданного множителя $\pi\in\mathbb{R}$ определим функцию $\phi_\pi : P \longmapsto \mathbb{R}$, такую что $\phi_\pi(p)=\phi(p)+\pi p$. Пусть $\Delta\phi(p)\triangleq\phi(p+1)-\phi(p)$ обозначает приращение функции дискретной переменной $\phi(\cdot)$ в точке p, тогда справедлива следующая лемма.

Лемма 1. *Если функция $\phi(p)$ выпукла на* $\mathrm{conv}(P)$, *то* $\Delta\phi(p+1)\geq\Delta\phi(p)$
$\forall p\in\{1,\dots,m-2\}$.

Доказательство. Перепишем неравенство $\Delta\phi(p+1)\geq\Delta\phi(p)$ в следующем эквивалентном виде

$$\phi(p+2)-2\phi(p+1)+\phi(p)\geq 0. \tag{2.21}$$

По определению выпуклой функции справедливо следующее неравенство

$$\phi(ax+(1-a)y)\leq a\phi(x)+(1-a)\phi(y)\quad \forall x,y\in\mathrm{conv}(P), a\in[0,1]$$

Перепишем это неравенство для точек p и $p+2$, полагая $a = \frac{1}{2}$, тогда получим

$$\begin{aligned} &\phi(\frac{p}{2} + \frac{p+2}{2}) \leq \frac{1}{2}\phi(p) + \frac{1}{2}\phi(p+2) \Leftrightarrow \\ &\phi(p+1) \leq \frac{1}{2}\phi(p) + \frac{1}{2}\phi(p+2), \end{aligned} \tag{2.22}$$

которое справедливо для всех $p \in \{1, m-2\}$. Отсюда, после умножения обеих частей неравенства на 2, следует справедливость (2.21). □

С использованием доказанного результата можно показать, что в зависимости от вида и свойств функции $\phi(\cdot)$ поиск минимума в задаче одномерной минимизации (2.20) может быть упрощен, что наряду с некоторыми другими свойствами оптимального решения задачи (2.20) доказано в следующем предложении.

Предложение 1. *Справедливы следующие утверждения:*

1. *Если $\phi(\cdot)$ вогнута на $\mathrm{conv}(P)$, то $p(\pi) \in \{1, m\}$, т.е. оптимальное решение задачи (2.20) находится на одном из концов отрезка.*

2. *Если $\phi(\cdot)$ выпукла на $\mathrm{conv}(P)$, то*

$$\text{а) } p(\pi) = 1 \Leftrightarrow \pi + \Delta\phi(1) \geq 0; \tag{2.23}$$

$$\text{б) } p(\pi) \in \{2, \ldots, m-1\} \Leftrightarrow \Delta\phi(p(\pi)-1) \leq -\pi \leq \Delta\phi(p(\pi)); \tag{2.24}$$

$$\text{в) } p(\pi) = m \Leftrightarrow \pi + \Delta\phi(m-1) \leq 0. \tag{2.25}$$

3. *Если $\pi \geq \max\limits_{k=1,\ldots,m-1}\{-\Delta\phi(k)\}$, тогда $p(\pi) = 1$.*

4. *Если $\pi \leq \min\limits_{k=1,\ldots,m-1}\{-\Delta\phi(k)\}$, тогда $p(\pi) = m$.*

Доказательство. 1. Очевидное утверждение, следующее из того факта, что функция $\phi_\pi(p)$ также будет вогнутой на $\mathrm{conv}(P)$.

2. Сначала докажем утверждения а), в), затем б).

а) Необходимость данного условия является очевидной. Предположим, что $p(\pi) = 1$ является оптимальным решением задачи (2.20), т.е. $\pi + \phi(1) \leq \pi p + \phi(p)\ \forall p \in P$. Соответственно выполнено $\pi + \phi(1) \leq 2\pi + \phi(2)$, из чего, после необходимых преобразований, следует справедливость утверждения.

Для доказательства достаточности заметим, что $\pi + \Delta\phi(1) \geq 0$ равносильно $\pi + \phi(1) \leq 2\pi + \phi(2)$, т.е.

$$\Delta\phi_\pi(1) \triangleq \pi + \phi(2) - \phi(1) \geq 0.$$

В силу выпуклости функции $\phi(\cdot)$ и с учетом леммы 1 справедливо $\Delta\phi_\pi(2) \triangleq \pi + \phi(3) - \phi(2) \geq \Delta\phi_\pi(1) \geq 0$. Аналогично можно показать, что

$$0 \leq \Delta\phi_\pi(1) \leq \Delta\phi_\pi(2) \leq, \ldots, \Delta\phi_\pi(m-1),$$

т.е. функция $\phi_\pi(\cdot)$ является неубывающей на P, откуда следует справедливость утверждения (2.23).

в) Необходимость очевидна и доказывается аналогично. Пусть теперь справедливо неравенство $\pi + \Delta\phi(m-1) \leq 0$, из чего следует что $\pi \leq 0$, поскольку функция $\phi(\cdot)$ является неубывающей на множестве P. Неравенство $\pi + \Delta\phi(m-1) \leq 0$ может быть переписано как $\pi m + \phi(m) \leq \pi(m-1) + \phi(m-1)$, т.е.

$$\Delta\phi_\pi(m-1) \triangleq \pi + \phi(m) - \phi(m-1) \leq 0.$$

В силу выпуклости функции $\phi(\cdot)$ справедливо

$$\Delta\phi_\pi(m-2) \triangleq \pi + \phi(m-3) - \phi(m-2) \leq \Delta\phi_\pi(m-1) \leq 0.$$

Аналогично можно показать, что

$$0 \geq \Delta\phi_\pi(m-1) \geq \Delta\phi_\pi(m-2) \geq, \ldots, \Delta\phi_\pi(1),$$

т.е. функция $\phi_\pi(\cdot)$ является невозрастающей на P, откуда следует справедливость утверждения (2.25).

б) Необходимость. Предположим, что $p(\pi) \in \{2, \ldots, m-1\}$ является оптимальным решением в подзадаче (2.20), т.е. справедливо $\phi_\pi(p(\pi)) \leq \phi_\pi(k)$, $\forall k \in P$. Тогда, в частности, выполняются следующие неравенства $\phi_\pi(p(\pi)) \leq \phi_\pi(p(\pi)-1)$ и $\phi_\pi(p(\pi)) \leq \phi_\pi(p(\pi)+1)$. По определению функции $\phi_\pi(\cdot)$ данные неравенства могут быть переписаны как:

$$\pi p(\pi) + \phi(p(\pi)) \leq \pi(p(\pi)-1) + \phi(p(\pi)-1),$$
$$\pi p(\pi) + \phi(p(\pi)) \leq \pi(p(\pi)+1) + \phi(p(\pi)+1).$$

Отсюда, проводя необходимые преобразования и учитывая определение $\Delta\phi(\cdot)$, получаем требуемые соотношения (2.24).

Для доказательство достаточности необходимо заметить, что неравенства (2.24) могут быть переписаны в следующем виде:

$$\pi + \Delta\phi(p-1) \leq 0,$$
$$\pi + \Delta\phi(p) \geq 0.$$

Далее доказательство проводится по аналогии с пунктами а), в).

3. Предположим, что справедливо $\pi \geq \max\limits_{k<m}(-\Delta\phi(k))$, тогда по определению максимума выполняется $\pi \geq \max\limits_{k<m}(-\Delta\phi(k)) \geq (-\Delta\phi(k))\ \forall k = 1, \ldots, m-1$. Отсюда по определению приращения получаем $\pi \geq -\phi(k+1)+\phi(k)\ \forall k = 1, \ldots, m-1$, что может быть переписано как $\phi_\pi(k+1)-\phi_\pi(k) \geq 0\ \forall k = 1, \ldots, m-1$, т.е. функция $\phi_\pi(\cdot)$ является неубывающей на множестве P. А следовательно $\phi_\pi(1) \leq \phi_\pi(k)\ \forall k = 2, \ldots, m$, что и требовалось доказать.

4. Доказывается аналогично утверждению 3.

□

Таким образом, при заданном наборе множителей в некоторых случаях есть возможность избежать полного перебора элементов множества P при поиске решения в задаче одномерной минимизации (2.20), что является существенным при увеличении размерности исходной задачи, поскольку при росте числа возможных пунктов, т.е. $|I|$, пропорционально растет и число элементов в множестве P.

Построим теперь релаксацию исходной задачи (2.2)–(2.8), в которой ослабляется лишь один тип ограничений (2.3). В этом случае полученную двойственную функцию Лагранжа будем обозначать $\theta_2(\lambda)$, а соответствующую релаксацию будем в дальнейшем называть $\mathcal{R}_2$-релаксацией:

$$\begin{aligned}\theta_2(\lambda) = \min_{(x,y,p)}\Big\{ \sum_{(i,j)\in A} d_{ij}x_{ij} + \sum_{j\in I}\lambda_j\Big(1 - y_j - \sum_{i\in\delta^-(j)} x_{ij}\Big) + \phi(p) : \\ x_{ij} \leq y_i,\ i \in I, j \in \delta^+(i);\ \sum_{j\in I} y_j = p\Big\}.\end{aligned} \tag{2.26}$$

Для некоторого фиксированного набора множителей $\lambda \in \mathbb{R}^m$ значение двойственной функции Лагранжа $\theta_2(\lambda)$ и переменных $x_{ij}(\lambda)$, $y_i(\lambda)$ может быть подсчитано в явном виде. Для этого введем для каждого узла $i \in I$, как и в первой главе для линейной задачи о p-медиане, величины $\rho_i(\lambda) = \sum\limits_{j\in\delta^+(i)} (d_{ij}-\lambda_j)^- - \lambda_i$, представляющие собой оценки Лагранжа для переменных y_i. Предположим, что существует такая перестановка $i_1, i_2, \ldots, i_m$ узлов орграфа $G(I, A)$, что соответствующие им оценки Лагранжа оказываются упорядоченными в неубывающем порядке, т.е.

$$\rho_{i_1}(\lambda) \leq \rho_{i_2}(\lambda) \leq \cdots \leq \rho_{i_m}(\lambda).$$

Для отсортированных таким образом оценок введем функцию $\psi(p, \lambda)$, такую что

$$\psi(p, \lambda) \triangleq \sum_{k=1}^{p} \rho_{i_k}(\lambda).$$

Тогда с использованием введенных обозначений справедливы следующие утверждения.

Теорема 2. *Для любого вектора $\lambda \in \mathbb{R}^m$ справедливо:*

$$\theta_2(\lambda) = \min_{p \in P} \Big\{ \sum_{k=1}^{p} \rho_{i_k}(\lambda) + \phi(p) \Big\} + \sum_{j \in J} \lambda_j. \tag{2.27}$$

Причем оптимальные значения прямых переменных в релаксированной задаче могут быть подсчитаны в явном виде:

$$p(\lambda) \in \operatorname*{Argmin}_{q \in P} \Big\{ \sum_{k=1}^{q} \rho_{i_k}(\lambda) + \phi(q) \Big\},$$

$$y_i(\lambda) = \begin{cases} 1, & i \in \{i_1, \ldots, i_{p(\lambda)}\}; \\ 0, & \textit{в противном случае.} \end{cases}$$

$$x_{ij}(\lambda) = \begin{cases} 1, & y_i(\lambda) = 1 \ \textit{и} \ d_{ij} - \lambda_j < 0; \\ 0, & \textit{в противном случае.} \end{cases}$$

Доказательство. Перепишем (2.26) в следующем виде:

$$\begin{aligned} \theta_2(\lambda) = \min_{(y,p)} \Big\{ \min_{x} \Big\{ \sum_{i \in I} \sum_{j \in \delta^+(i)} (d_{ij} - \lambda_j) x_{ij} : \ 0 \le x_{ij} \le y_i \Big\} - \\ - \sum_{i \in I} \lambda_i y_i + \phi(p) : \ \sum_{i \in I} y_i = p \Big\} + \sum_{j \in I} \lambda_j. \end{aligned} \tag{2.28}$$

Далее проведем рассуждения, аналогичные рассуждениям в теореме 1, т.е. выпишем следующую задачу минимизации:

$$\min_{x} \Big\{ \sum_{i \in I} \sum_{j \in \delta^+(i)} (d_{ij} - \lambda_j) x_{ij} : \ 0 \le x_{ij} \le y_i \Big\},$$

решение которой может быть записано как

$$x_{ij}(\lambda) = \begin{cases} y_i, & d_{ij} - \lambda_j < 0, \\ 0, & d_{ij} - \lambda_j \ge 0. \end{cases}$$

Далее, подставим полученное решение в (2.28):

$$\theta_2(\lambda) = \min_{(y,p)} \Big\{ \sum_{i \in I} \Big[\sum_{j \in \delta^+(i)} (d_{ij} - \lambda_j)^- - \lambda_i \Big] y_i + \phi(p) : \ \sum_{i \in I} y_i = p \Big\} + \sum_{j \in I} \lambda_j.$$

Используя введенное определение оценок Лагранжа $\rho_i(\lambda)$, последнее может быть записано следующим образом:

$$\theta_2(\lambda) = \min_{p} \Big\{ \min_{y} \Big\{ \sum_{i \in I} \rho_i(\lambda) y_i : \ \sum_{i \in I} y_i = p \Big\} + \phi(p) \Big\} + \sum_{j \in I} \lambda_j. \tag{2.29}$$

Зафиксируем переменную p и рассмотрим отдельно задачу минимизации по y

$$\min_y \Big\{ \sum_{i \in I} \rho_i(\lambda) y_i : \sum_{i \in I} y_i = p \Big\}.$$

В данной задаче необходимо найти минимум линейной функции при одном линейном ограничении. Учитывая тот факт, что для узлов имеется перестановка $i_1, i_2, \ldots, i_m$, согласно которой все оценки Лагранжа $\rho_i(\lambda)$ упорядочены по возрастанию, нетрудно видеть, что решением будет вектор $y(\lambda)$, в котором отличными от 0 будут ровно p координат, соответствующих наименьшим значениям оценок Лагранжа $\rho_i(\lambda)$. Далее, подставляя найденное решение в (2.29), получим требуемое соотношение (2.27). □

Замечание 1. Сложность вычисления значения двойственной функции Лагранжа $\theta_2(\lambda)$, как и функции $\theta_1(\lambda)$, составляет $O(n^2)$, поскольку выражение $\psi(p, \lambda) + \phi(p)$ может быть вычислено рекурсивно по следующим формулам:

$$\psi(1, \lambda) + \phi(1) = \rho_{i_1}(\lambda) + \phi(1),$$

$$\psi(p, \lambda) + \phi(p) = \psi(p-1, \lambda) + \rho_{i_p}(\lambda) + \phi(p) \ \forall p = 2, \ldots, m.$$

Таким образом, для подсчета значения двойственной функции Лагранжа $\theta_2(\lambda)$ необходимо найти оптимальное решение в задаче одномерной минимизации

$$\min_{p \in P} \{\psi(p, \lambda) + \phi(p)\},$$

которая в общем случае может быть решена полным перебором по p от 1 до m. Однако, как и в случае первой релаксации, в зависимости от вида функции $\phi(\cdot)$ процедура поиска минимума может быть упрощена. Так, в случае, если функция $\phi(\cdot)$ — строго возрастающая на P, то минимум достигается при $p = 1$, если $\rho_{i_1}(\lambda) \geq 0$. В противном же случае необходимо выполнить перебор только по тем узлам графа, которые соответствуют переменным y_i с отрицательными оценками Лагранжа. В случае, если функция $\phi(\cdot)$ — не только строго возрастающая, но и выпуклая на $\mathrm{conv}(P)$, существует единственный минимум функции $\psi(p, \lambda) + \phi(p)$ при $p \in P$, что доказывается в следующем предложении.

Предложение 2. *Пусть $\phi(\cdot)$ — строго возрастающая функция на P, тогда справедливы следующие утверждения:*

1. *Если $\rho_{i_1}(\lambda) \geq 0$, то $p(\lambda) = 1$.*
2. *Если $\rho_{i_1}(\lambda) < 0$, то $p(\lambda) \in \{k \in P : \rho_{i_k}(\lambda) < 0\}$.*

3. Если $\phi(\cdot)$ к тому же выпукла на $\mathrm{conv}(P)$*, то справедливо:*

$$\text{а) } p(\lambda) = 1 \Leftrightarrow \Delta\phi(1) + \rho_{i_2}(\lambda) \geq 0; \tag{2.30}$$

$$\text{б) } p(\lambda) \in \{2, \ldots, m-1\} \Leftrightarrow \Delta\phi(\bar{p}-1) + \rho_{i_{\bar{p}}}(\lambda) \leq 0, \tag{2.31}$$

$$\Delta\phi(\bar{p}) + \rho_{i_{\bar{p}+1}}(\lambda) \geq 0; \tag{2.32}$$

$$\text{в) } p(\lambda) \neq m. \tag{2.33}$$

Доказательство. 1. Если $\rho_{v_1(\lambda)}(\lambda) \geq 0$, то $\psi(p, \lambda) \leq \psi(p+1, \lambda)$, $p \in P$. А поскольку $\phi(\cdot)$ предполагается строго возрастающей на P, то таковой является и функция $p \longmapsto \psi(p, \lambda) + \phi(p)$, из чего следует справедливость первого утверждения. Более того, значение двойственной функции Лагранжа может быть подсчитано как

$$\theta_2(\lambda) = \rho_{i_1}(\lambda) + \phi(1) + \sum_{j \in I} \lambda_j.$$

2. Предположим теперь, что $\rho_{i_1}(\lambda) < 0$. Обозначим $P_0(\lambda) \triangleq \{k \in P : \rho_{i_k}(\lambda) < 0\}$, тогда $P_0(\lambda) \neq \varnothing$. Если $P_0(\lambda) = P$, то утверждение 2 предложения очевидно. Рассмотрим случай, когда $P_0(\lambda) \neq P$. Тогда по определению оценок Лагранжа и перестановки $i_1, i_2, \ldots, i_m$, а также с учетом условий предложения, $\exists \bar{p} \in P$:

$$\rho_{i_k}(\lambda) < 0 \qquad \forall k = 1, \ldots, \bar{p};$$

$$\rho_{i_k}(\lambda) \geq 0 \qquad \forall k = \bar{p}+1, \ldots, m.$$

Покажем, что

$$\psi(\bar{p}, \lambda) + \phi(\bar{p}) \leq \psi(p, \lambda) + \phi(p) \ \forall p \in P \backslash P_0(\lambda).$$

Для этого запишем

$$\psi(p, \lambda) = \sum_{k=1}^{\bar{p}} \rho_{i_k}(\lambda) + \sum_{k=\bar{p}+1}^{p} \rho_{i_k}(\lambda) = \psi(\bar{p}, \lambda) + \sum_{k=\bar{p}+1}^{p} \rho_{i_k}(\lambda) \geq \psi(\bar{p}, \lambda).$$

Учитывая, что функция $\phi(\cdot)$ — возрастающая на P, получим требуемое $p(\lambda) \in \{k \in P : \rho_{i_k}(\lambda) < 0\}$.

3. а) Необходимость очевидна, поскольку неравенство (2.30) может быть переписано в следующем виде:

$$\psi(1, \lambda) + \phi(1) \leq \psi(2, \lambda) + \phi(2).$$

Достаточность. Рассмотрим приращения функций

$$\Delta\phi(p) = \phi(p+1) - \phi(p),$$

$$\Delta_p\psi(p,\lambda) = \psi(p+1,\lambda) - \psi(p,\lambda).$$

Тогда из формулы (2.30) следует, что

$$\psi(2,\lambda) - \psi(1,\lambda) + \phi(2) - \phi(1) \geq 0,$$

то есть

$$\Delta_p\psi(1,\lambda) + \Delta\phi(1) \geq 0.$$

По определению функции $\psi(\cdot,\cdot)$ и в силу выпуклости $\phi(\cdot)$ получаем

$$\Delta_p\psi(2,\lambda) \geq \Delta_p\psi(1,\lambda),$$

$$\Delta\phi(2) \geq \Delta\phi(1).$$

Сложив эти неравенства, получим

$$\Delta_p\psi(2,\lambda) + \Delta\phi(2) \geq \Delta_p\psi(1,\lambda) + \Delta\phi(1) \geq 0.$$

Распишем левую часть неравенств

$$\psi(3,\lambda) - \psi(2,\lambda) + \phi(3) - \phi(2) \geq 0,$$

$$\psi(3,\lambda) + \phi(3) \geq \psi(2,\lambda) + \phi(2).$$

Проведя аналогичные рассуждения, получим

$$\psi(1,\lambda) + \phi(1) \leq \psi(2,\lambda) + \phi(2) \leq \ldots \leq \psi(n,\lambda) + \phi(n).$$

б), в). Аналогичным образом можно показать, что

$$\psi(\bar{p}+1,\lambda) + \phi(\bar{p}+1) \leq \psi(\bar{p}+2,\lambda) + \phi(\bar{p}+2) \leq \ldots \leq \psi(n,\lambda) + \phi(n). \tag{2.34}$$

Далее рассмотрим неравенства (2.31), которые могут быть переписаны следующим образом:

$$\psi(\bar{p},\lambda) - \psi(\bar{p}-1,\lambda) + \phi(\bar{p}) - \phi(\bar{p}-1) \leq 0,$$

то есть

$$\Delta_p\psi(\bar{p}-1,\lambda) + \Delta\phi(\bar{p}-1) \leq 0.$$

По определению функций $\psi(\cdot,\cdot)$ и в силу выпуклости функции $\phi(\cdot)$

$$\Delta_p\psi(\bar{p}-2,\lambda) \leq \Delta_p\psi(\bar{p}-1,\lambda),$$

$$\Delta\phi(\bar{p}-2) \leq \Delta\phi(\bar{p}-1).$$

Тогда

$$\Delta\psi(\bar{p}-2,\lambda)+\Delta\phi(\bar{p}-2)\leq\Delta\psi(\bar{p}-1,\lambda)+\Delta\phi(\bar{p}-1)\leq 0.$$

Отсюда следует, что

$$\psi(\bar{p}-1,\lambda)-\psi(\bar{p}-2,\lambda)+\phi(\bar{p}-1)-\phi(\bar{p}-2)\leq 0,$$

$$\psi(\bar{p}-1,\lambda)+\phi(\bar{p}-1)\leq\psi(\bar{p}-2,\lambda)+\phi(\bar{p}-2).$$

Проведя аналогичные рассуждения, можно получить

$$\psi(\bar{p},\lambda)+\phi(\bar{p})\leq\psi(\bar{p}-1,\lambda)+\phi(\bar{p}-1)\leq\ldots\leq\psi(1,\lambda)+\phi(1). \tag{2.35}$$

Из (2.34),(2.35) можно сделать вывод, что точка $\bar{p}\in P$ является точкой минимума функции $\psi(p,\lambda)+\phi(p)$ при $p\in P$, откуда следует справедливость утверждения 3 предложения.

□

Как было отмечено в первой главе, значения двойственных функций Лагранжа $\theta_1(\lambda,\pi)$ и $\theta_2(\lambda)$ для всякого фиксированного набора множителей предоставляют нижнюю оценку оптимального значения в модифицированной задаче о p-медиане (2.2)–(2.8).

Замечание 2. Для любого $\lambda\in\mathbb{R}^m$ и $\pi\in\mathbb{R}$ справедливо следующее неравенство:

$$\theta_1(\lambda,\pi)\leq\theta_2(\lambda). \tag{2.36}$$

2.3. Метод поиска приближенных решений

Для представленной нелинейной задачи о p-медиане предлагается эвристический алгоритм поиска приближенных решений, являющийся обобщением метода, представленного в первой главе для задачи о p-медиане, учитывающим специфику рассматриваемой задачи. Напомним, что идея метода состоит в последовательном поиске нижних и верхних оценок оптимального значения задачи с помощью метода релаксаций Лагранжа и субградиентного алгоритма, а также так называемой ядровой эвристики, использующей информацию об оценках Лагранжа, получаемую по окончанию процедуры поиска нижних оценок оптимального значения.

2.3.1. Субградиентный алгоритм и метод генерации строк

Для поиска наилучшей нижней оценки, получаемой с помощью представленных в разделе 2.2 релаксаций, необходимо найти оптимальное решение в следующих двойственных

задачах $\max\limits_{(\lambda,\pi)\in\mathbb{R}^m\times\mathbb{R}} \theta_1(\lambda,\pi)$ и $\max\limits_{\lambda\in\mathbb{R}} \theta_2(\lambda)$ соответственно. Как и в первой главе, для этих целей применяется субградиентный метод с эвристическим выбором шага, а также специальный метод генерации столбцов, адаптированный к рассматриваемой задачи. Как и прежде, вначале применяется процедура препроцессинга, позволяющая существенно ускорить время работы алгоритма, а также сделать возможным применение алгоритма для задач большой размерности. Поскольку дальнейшие рассуждения во многом идентичны для обоих типов релаксаций, то остановимся подробно лишь на $\mathcal{R}_2$-релаксации. Так, если для каждого столбца $j = 1,\dots,m$ имеется перестановка $u(j) = u_1(j),\dots,u_m(j)$, такая что j-ый столбец оказывается отсортированным по возрастанию, то значение двойственной функции Лагранжа для некоторого заданного набора множителей может быть «быстро» подсчитано по следующему алгоритму:

Алгоритм 2.1 Процедура подсчета вектора оценок Лагранжа и значения двойственной функции Лагранжа $\theta_2(\lambda)$

0: Инициализация: $\rho(\lambda) := -\lambda,\ \theta_2(\lambda) := 0$ и $j := 1$;

1: Вычислить $\theta_2(\lambda) := \theta_2(\lambda) + \lambda_j$ и положить $i := 1$;

2: **Если** $d(u_i(j), j) - \lambda_j >= 0$, **тогда** перейти на 5;

3: Вычислить $\rho_{u_i(j)}(\lambda) := \rho_{u_i(j)}(\lambda) + d(u_i(j), j) - \lambda_j$;

4: **Если** $i < m$, **тогда** $i := i + 1$ и перейти на 2;

5: **Если** $j < m$, **тогда** $j := j + 1$ и перейти на 1;

6: Подсчитать $\theta_2^* := \min\limits_{p\in P}\{\psi(\lambda, p) + \phi(p)\}$;

7: Вычислить $\theta_2(\lambda) := \theta_2(\lambda) + \theta_2^*$.

Возвратить вектор оценок $\rho(\lambda)$ и значение $\theta_2(\lambda)$.

Заметим, что на шаге 6 при поиске решения в задаче минимизации могут быть использованы как теоретические результаты, представленные выше, так и различные методы одномерной минимизации, например, метод деления отрезка пополам.

Остальные компоненты метода поиска нижних оценок в рассмотренной нелинейной задаче о p-медиане, включая правило выбора шага субградиентного алгоритма, а также стабилизацию множителей Лагранжа для улучшения его сходимости, аналогичны представленным в первой главе диссертации.

2.3.2. Ядровая эвристика

Для поиска верхних оценок применяется так называемая ядровая эвристика. Пусть дан вектор оценок Лагранжа λ (а в случае $\mathcal{R}_1$-релаксации набор множителей (λ,π)), полученный

по завершению процедуры поиска нижних оценок оптимального значения задачи, т.е. решения лагранжевой двойственной задачи специальным субградиентным алгоритмом, а также оценки Лагранжа $\rho_i(\lambda) = \sum\limits_{j \in \delta^+(i)} (d_{ij} - \lambda_j)^- - \lambda_i$ и $q_{ij}(\lambda) = d_{ij} - \lambda_j$ для переменных y_i и x_{ij} соответственно. Отметим, что в случае релаксации $\mathcal{R}_1$ в качестве оценок Лагранжа для переменных y_i выбираются величины $\zeta_i(\lambda, \pi) = \sum\limits_{j \in \delta^+(i)} (d_{ij} - \lambda_j)^- - \lambda_i - \pi$. Тогда ядровая задача формируется с использованием полученного по завершению субградиентного алгоритма числа медиан (открытых предприятий) $p(\lambda)$, а также подмножества переменных, величины оценок Лагранжа которых меньше некоторых наперед заданных пороговых значений: ν для переменных y_i и μ для переменных x_{ij}.

Напомним, что при построении ядровой задачи в качестве модели используется линеаризованная формулировка (2.9)–(2.16), что позволяет применять для ее решения коммерческие решатели задач целочисленного линейного программирования (IBM ILOG CPLEX, FICO Xpress, Gurobi Optimizer). Использование линеаризованной формулировки также мотивировано отсутствием необходимости учитывать свойства функции $\phi(\cdot)$ при решении ядровой задачи с помощью коммерческих пакетов. Таким образом, в качестве ядровой строится задача вида (2.9)–(2.16), в которой не фиксированными в нуле будут только те переменные y_i и x_{ij}, для которых выполняются следующие условия:

$$
\begin{aligned}
& i \in I(\lambda, \nu) \triangleq \{i \in I : \ \rho_i(\lambda) \leq \nu\}, \\
& (i,j) \in W(\lambda, \mu) \triangleq \{(i,j) : \ i \in I(\lambda, \nu), j \in \delta^+(i), \ q_{ij} \leq \mu\}, \\
& p \in [p(\lambda) - \delta, p(\lambda) + \delta].
\end{aligned}
$$

Отметим, что в отличие от первой главы, при построении ядровой задачи необходимо учитывать, что p является переменной, поэтому при формировании редуцированной задачи $p(\lambda)$ выбирается не фиксированной, а из некоторого отрезка с параметром $\delta \in \mathbb{N}$. Такой параметр дает методу дополнительную свободу в выборе оптимального количества медиан (открываемых предприятий). Также заметим, что размер ядровой задачи напрямую зависит от значения параметров ν, μ: чем больше их величина, тем больше элементов в множествах $I(\lambda, \nu), W(\lambda, \mu)$, и, соответственно, тем больше размерность задачи целочисленного программирования, точное решение которой ищется с помощью коммерческого решателя. Таким образом, выбор значений параметров ν, μ подразумевает некий компромисс между желаемой точностью получаемой в итоге верхней оценки и временем работы алгоритма.

2.3.3. Метод поиска приближенных решений нелинейной задачи о p-медиане

При применении ядровой эвристики, как было отмечено ранее, важным является нахождение набора множителей Лагранжа, соответствующего «хорошей» нижней оценке оптимального значения, поскольку от этого зависит построение ядровой задачи и, соответственно, качество получаемой верхней оценки. С другой стороны, для подсчета шага субградиентного алгоритма важным является выбор верхней оценки оптимального значения, что существенно влияет на сходимость метода и качество получаемой нижней оценки оптимального значения. В главе 1 для использования данной особенности предлагалось естественная идея последовательного повторения процедур поиска верхних и нижних оценок с целью уменьшения относительной погрешности между ними. В данной главе применяется схожий подход, однако последовательные повторения происходят либо заданное число раз, либо до тех пор, пока погрешность, определяемая величиной $Err = (BUB - BLB)/BUB \cdot 100\%$, не будет достаточно малой.

Обозначим через $Z(x, y, p)$ значение целевой функции (2.2) на некотором допустимом решении (x, y, p), тогда общая схема предлагаемого метода поиска приближенных решений в нелинейной задаче о p-медиане может быть представлена следующим образом (см. алгоритм 2.2). Отметим, что данный алгоритм схожим образом может быть записан и для $\mathcal{R}_1$-релаксации. Первоначальное допустимое решение, предоставляющее верхнюю оценку для запуска субградиентного алгоритма, находится посредством достраивания решения $(y_i(\lambda^0), x_{ij}(\lambda^0))$, полученного на первой итерации, до допустимого.

Другой особенностью представленной реализации ядровой эвристики является динамическое изменение размера ядровой задачи. Если после трех запусков субградиентного алгоритма и двух запусков процедуры решения ядровой задачи не достигнута малая относительная погрешность между найденными верхней BUB и нижней BLB оценками, то пороговые значения ν, μ увеличиваются и происходит поиск обновленных BLB, BUB и $(\bar{x}, \bar{y}, \bar{p})$. Затем, если относительная погрешность между BUB и BLB все еще велика, то процедуры субградиентной оптимизации и решения ядровой задачи запускаются вновь при тех же значениях пороговых величин (шаги 9 и 10 алгоритма 2.2).

Алгоритм 2.2 Поиск приближенных решений нелинейной задачи о p-медиане

0: Инициализация. Построить допустимое решение $(\bar{x}, \bar{y}, \bar{p})$, положить наилучшие найденные верхние и нижние оценки как $BUB := Z(\bar{x}, \bar{y}, \bar{p})$ и $BLB := -\infty$, $h := 1$.

1: Найти оптимальное решение двойственной задачи Лагранжа λ с помощью субградиентного алгоритма, положить $BLB := \theta_2(\lambda)$.

2: Построить $I(\lambda, \nu)$, $W(\lambda, \mu)$ и найти решение $(\widehat{x}, \widehat{y}, \widehat{p})$ соответствующей ядровой задачи. Если $Z(\widehat{x}, \widehat{y}, \widehat{p}) < BUB$, то положить $BUB := Z(\widehat{x}, \widehat{y}, \widehat{p})$ и $(\bar{x}, \bar{y}, \bar{p}) := (\widehat{x}, \widehat{y}, \widehat{p})$.

3: Запустить процедуру субградиентной оптимизации, выбирая в качестве начального значения оценок Лагранжа те, которые соответствуют BLB. Найти новые значения вектора λ и положить $BLB := \theta_2(\lambda)$.

4: **Если** $h < 2$, **тогда** $h := h + 1$ и перейти на 2.

5: **Если** $Err > \varepsilon$, **тогда** увеличить размер ядровой задачи (т.е. увеличить пороговые значения ν и μ), **иначе** stop.

6: Построить $I(\lambda, \nu)$, $W(\lambda, \mu)$ и найти решение $(\widehat{x}, \widehat{y}, \widehat{p})$ соответствующей ядровой задачи. Если $Z(\widehat{x}, \widehat{y}, \widehat{p}) < BUB$, то положить $BUB := Z(\widehat{x}, \widehat{y}, \widehat{p})$ и $(\bar{x}, \bar{y}, \bar{p}) := (\widehat{x}, \widehat{y}, \widehat{p})$.

7: Продолжить процедуру субградиентной оптимизации, начиная с множителей, полученных после предыдущего запуска, найти новый вектор оценок λ и положить $BLB := \theta_2(\lambda)$.

8: **Если** $Err > \varepsilon$, **тогда** перейти на 9, **иначе** stop.

9: Построить $I(\lambda, \nu)$, $W(\lambda, \mu)$ и найти решение $(\widehat{x}, \widehat{y}, \widehat{p})$ ядровой задачи. Если $Z(\widehat{x}, \widehat{y}, \widehat{p}) < BUB$, то положить $BUB := Z(\widehat{x}, \widehat{y}, \widehat{p})$ и $(\bar{x}, \bar{y}, \bar{p}) := (\widehat{x}, \widehat{y}, \widehat{p})$.

10: Продолжить процедуру субградиентной оптимизации, начиная с множителей, полученных после предыдущего запуска, найти новый вектор оценок λ и положить $BLB := \theta_2(\lambda)$.

Возвратить значения BLB, BUB и $(\bar{x}, \bar{y}, \bar{p})$.

2.4. Вычислительный эксперимент

Представленный алгоритм реализован на языке C++ и протестирован на ЭВМ со следующими характеристиками: процессор Intel Core 2 Duo CPU 2.20GHz и 3Gb RAM, ОС Windows Vista 32-bit. В качестве коммерческого решателя задач целочисленного программирования использовался IBM ILOG CPLEX Optimizer 12.1.0[1]. Отметим, что хотя реализованный алгоритм не является параллельным, в вычислительных экспериментах CPLEX Optimizer использовал оба процессорных ядра. В качестве тестовых использовались искусственно сгенерированные задачи, а также ряд примеров из известной библиотеки тестовых задач TSPLIB.[2]

- *Искусственно сгенерированные задачи.* Первый набор примеров для тестирования разработанного алгоритма был получен с помощью генерации по схеме, предложенной в работах [278, 279]. Отметим, что задачи, полученные с помощью такого метода, в частности, использовались в работе [164] для тестирования прямо-двойственного локального поиска с чередующимися окрестностями для задачи о p-медиане большой размерности. В рамках данного подхода каждая задача класса состоит из p групп (кластеров) точек в двумерном пространстве. Каждой такая группа определяется числом содержащихся в ней точек (m), радиусом (r) и центром (c), причем $m \in [m_l, m_h]$ и $r \in [r_l, r_h]$. При выборе центра групп точек имеются три шаблона параметров, задающих центры каждой группы, а именно: *grid, sine* и *random*. При использовании шаблона *grid* центры всех кластеров размещаются в узлах сетки $\sqrt{p} \times \sqrt{p}$. Расстояние между центрами групп, соседних по вертикали или горизонтали, контролируется с помощью параметра $k_g \triangleq \dfrac{r_l + r_h}{2}$. При использовании шаблона *sine* центры групп точек располагаются на кривой функции синуса. Шаблон *random* предполагает размещение центров всех кластеров случайным образом в точках из множества $[0, p] \times [0, p]$. Отметим, что в представленных вычислительных экспериментах использовались примеры только двух типов, сгенерированных по шаблону *grid* (Type I) и *random* (Type III). Все сгенерированные задачи каждого из возможных типов содержат от 1000 до 10000 точек на плоскости. В общей сложности были сгенерированы по 8 тестовых примеров каждой из представленных размерностей с шагом 1000, содержащих различное количество медиан. Так, например, задача с 2000 точек сгенерирована для 9, 16, 25, 36, 49, 64, 81, 100 кластеров, или в нашем случае медиан. Отметим, что задачи Типа I, т.е. с центрами групп,

[1] http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

расположенными на сетке, являются более простыми для решения, чем примеры Типа III, в которых центры задаются случайно [164].

- *Задачи из TSPLIB*. Второй набор примеров представляет собой задачи различной размерности, взятые из библиотеки TSPLIB. Числа в названии каждого примера характеризуют размерность задачи, т.е. количество точек в них. Отметим, что в вычислительном эксперименте использовались только задачи, содержащие не более 10000 точек за исключением примеров $usa13509$ и $rl11849$.

Поскольку рассматриваемые задачи имеют значительно большую размерность, чем известная до сегодняшнего дня в литературе [189, 218], матрицы расстояний для таких задач, вообще говоря, не могут полностью храниться в оперативной памяти компьютера. В связи с чем применяется техника, описанная, в частности, в параграфах 1.2.4, 2.3.1, при которой в оперативной памяти хранится только некоторое число активных элементов, количество которых в нашем случае предполагается таким, что вся матрица расстояний занимает не более 800Мб.

Начальные параметры реализованного метода поиска приближенных решений выбирались таким же образом, как и в работе [56], а именно:

- На шаге 2 (см. алгоритм 2.2) параметры ν и μ задавались таким образом, чтобы ограничить размерность ядровой задачи так, что $|I(\lambda,\nu)| = 3p(\lambda)$, $|W(\lambda,\mu)| = 5m$. Лимит времени для поиска решения ядровой задачи коммерческим решателем устанавливался равным 300 секундам.

- На шагах 6 и 9 множества $I(\lambda,\nu)$ и $W(\lambda,\mu)$ увеличивались таким образом, что $|I(\lambda,\nu)| = 6p(\lambda)$, $|W(\lambda,\mu)| = 10m$. Лимит времени решения ядровой задачи ограничивался 400 секундами на шаге 6 и 600-ми секундами на шаге 9.

- Параметр ε, задающий погрешность Err, полагался равным 1%.

- Для того чтобы избежать случаев недопустимости ядровой задачи, для каждой отобранной для ядровой задачи переменной y_i, $i \in I(\lambda,\nu)$ контролируется минимальное число отбираемых переменных x_{ij}, $(i,j) \in W(\lambda,\mu)$, предполагаемое равным трем.

- Для параметра δ устанавливалось значение, равное 5.

- На шаге 1 параметр β полагался равным 1.5 и делился на 1.01 на каждой итерации, в которой не происходило улучшения лучшей нижней оценки.

- На шаге 3 ($h = 1$) параметр β устанавливался равным 0.1 и обновлялся по предыдущему правилу.

- На шаге 3 ($h = 2$) параметр β полагался равным 0.005 и делился на 1.01 всякий раз, когда нижняя оценка не улучшалась после двух последовательных итераций.

2.4.1. Результаты численного сравнения разработанного алгоритма с коммерческим решателем IBM ILOG CPLEX

Вначале представим результаты сравнения времени работы с современным коммерческим решателем IBM ILOG CPLEX 12.6.1. В качестве тестовых задач использовались искусственно сгенерированные примеры двух типов размерности от 100 до 1000 точек (вершин графа), полученные по описанной выше схеме и разбитые на 4, 9 и 16 групп (кластеров). Сразу отметим, что во всех случаях с помощью представленного в главе метода удалось найти нижние оценки оптимального значения задачи, совпадающие с верхними, т.е. удалось получить оптимальные решения задачи. Результаты вычислений представлены в таблице 2.1, где в первом столбце дана размерность тестовых примеров, во втором — время работы решателя для непрерывной релаксации исследуемого нелинейного варианта задачи о p-медиане, в третьем — время работы решателя непосредственно для нелинейного варианта задачи о p-медиане, в четвертом столбце представлено время работы разработанного в диссертации алгоритма. Отметим, что каждая строка таблицы содержит средние результаты для 9 задач, имеющих одинаковую размерность, но сгенерированных при различном числе кластеров p, равного 4, 9, 16, и функций $\phi(p) = p^2$, $\phi(p) = 5p^2$, $\phi(p) = 10p^2$. Напомним, что разработанный алгоритм не является параллельным, однако при поиске решений ядровой задачи с использованием коммерческого решателя IBM ILOG CPLEX использовались все доступные процессорные ядра. При численном сравнении результаты решателя IBM ILOG CPLEX 12.6.1 также получены при использовании всех процессорных ядер.

Как можно видеть, во всех случаях среднее время работы разработанного алгоритма существенно превосходит как время решения релаксированной, так и изначальной нелинейной задачи о p-медиане современным коммерческим решателем. Из чего можно сделать вывод о высокой эффективности разработанного метода даже на примерах небольшой размерности, а также возможности поиска решений для существенно больших задач, не поддающейся современным программным средствам.

Таблица 2.1. Сравнение времени работы с коммерческим решателем IBM ILOG CPLEX

m	$Relaxed(CPLEX)$	$Opt(CPLEX)$	$CoreHeur$
Type I			
100	1.34	1.01	0.07
200	2.91	5.35	0.14
400	18.95	27.43	0.29
800	2194.84	2258.62	0.93
1000	7109.97	7081.05	1.13
Type III			
100	0.96	0.92	0.08
200	4.41	7.97	0.15
400	31.00	49.18	0.39
800	2605.61	2788.02	1.04
1000	8817.68	11262.78	1.92

2.4.2. Выпуклая функция $\phi(\cdot)$

Вначале рассмотрим результаты вычислительных экспериментов, полученные для случая, когда $\phi(p)$ представляется в виде выпуклых функций $\phi(p) = cp^2$ с различными значениями параметра c, а именно: $\phi(p) = p^2$, $\phi(p) = 5p^2$ и $\phi(p) = 10p^2$.

Результаты расчетов представлены в таблицах с использованием следующих обозначений: столбец $Err(\%)$ содержит значения относительной погрешности между наилучшими верхними и нижними оценками, вычисляемой по формуле $Err = (BUB - BLB)/BUB \cdot 100\%$, в столбце $Time$ представлено общее время работы алгоритма, включающее в себя также время подсчета матрицы расстояний, а также поиск решения в ядровой задаче на каждом шаге метода с помощью решателя IBM ILOG CPLEX. Важно заметить, что для искусственно сгенерированных задач столбцы в таблицах с результатами содержат средние значения для восьми примеров одинаковой размерности, сгенерированных с различным числом медиан.

В таблицах 2.2, 2.3 и 2.4 представлены результаты для двух рассматриваемых видов тестовых задач, полученные с помощью представленных выпуклых функций $\phi(\cdot)$. Результаты для искусственно сгенерированных задач собраны в таблице 2.2, в то время как таблицы 2.3 и 2.4 содержат результаты вычислительного эксперимента для примеров из библиотеки TSPLIB, полученные с помощью релаксаций $\mathcal{R}_1$ и $\mathcal{R}_2$, соответственно.

Как можно заметить, для искусственно сгенерированных задач результаты хорошего качества получены во всех случаях, в которых используется релаксация Лагранжа второго

типа, т.е. $\mathcal{R}_2$. При использовании $\mathcal{R}_1$-релаксации приемлемые результаты получены только для задач с относительно небольшим количеством точек и для функций $\phi(p) = p^2$, $\phi(p) = 5p^2$. Что касается времени работы алгоритма, то при использовании первого типа релаксации оно незначительно превосходит таковое при использовании $\mathcal{R}_2$-релаксации. Для задач из библиотеки TSPLIB наблюдается аналогичная ситуация в случаях, когда используется релаксация $\mathcal{R}_2$. С использованием $\mathcal{R}_1$-релаксации хорошие результаты получены только для задач с относительно небольшим количеством точек (см. таблицу 2.3).

Анализируя полученные результаты, можно сделать вывод о том, что использование второго типа релаксации дает лучшие результаты как по качеству решения, так и по общему времени работы алгоритма, чем полученные с помощью $\mathcal{R}_1$-релаксации. В связи с чем, в дальнейшем при тестировании алгоритма на задачах более высокой сложности, используется и анализируется второй тип релаксации Лагранжа как показавший лучшие результаты.

2.4.3. Кусочно-линейная вогнутая функция $\phi(\cdot)$

С другой стороны, предварительные вычислительные эксперименты для случая вогнутой функции $\phi(\cdot)$ произвольного вида показали, что разработанный эвристический алгоритм не дает столь же оптимистичных результатов, поскольку относительная погрешность между полученными верхней и нижней оценками становится значительно большей, чем имевшаяся для случая выпуклой функции $\phi(\cdot)$, из-за наличия большого разрыва целочисленности.

Однако, как будет показано ниже, предложенный метод может быть успешно применен для одного важного частного класса вогнутых функций, а именно для кусочно-линейных вогнутых функций. Действительно, в этом случае задача может быть разбита на серию подзадач, где переменная p изменяется в пределах меньшего отрезка, в пределах которого функция $\phi(\cdot)$ линейна, а следовательно выпукла, что позволяет ожидать хороших вычислительных результатов для каждой из таких подзадач.

Рассмотрим кусочно-линейную вогнутую функцию $\phi(\cdot)$, такую что $\phi(p) = \min\limits_{1 \le i \le s} \{k_i p + b_i\}$. Отметим, что такая функция может быть представлена в следующей форме:

$$\phi(p) = \begin{cases} k_1 p + b_1, & p \in [l_0, l_1], \\ k_2 p + b_2, & p \in [l_1, l_2], \\ \dots \\ k_s p + b_s, & p \in [l_{s-1}, l_s], \end{cases} \tag{2.37}$$

где $k_1 > k_2 > \ldots > k_s$, и $l_0 = 1$, $l_s = m$.

Для поиска решения в нелинейной задаче о p-медиане (2.2)–(2.8) с кусочно-линейной

функцией $\phi(\cdot)$, определенной по правилу (2.37), разработанный эвристический алгоритм запускается последовательно для каждого отрезка кусочно-линейной функции, в пределах которого p выбирается из интервала $[l_{j-1}, l_j]$. В качестве верхних и нижних оценок оптимального значения задачи (2.2)–(2.8) выбираются наименьшие из оценок, полученных при решении подзадач для разных отрезков кусочно-линейной функции $\phi(\cdot)$.

В представленных ниже экспериментах рассматривались кусочно-линейные функции с $s = 4$ и $s = 6$ отрезками. Коэффициенты b_i выбирались таким образом, чтобы $b_0 = 0$, функция $\phi(\cdot)$ была непрерывной, и к тому же узлы l_i в (2.37) удовлетворяли соотношению $2(l_i - l_{i-1}) = l_{i+1} - l_i$. В качестве множества $K_s = \{k_i \in \mathbb{R}, i = 1, \ldots, s\}$ для искусственно сгенерированных тестовых задач использовались $K_4 = \{25, 15, 8, 1\}$ и $K_6 = \{30, 25, 18, 11, 6, 1\}$. Поскольку в задачах из тестовой библиотеки TSPLIB величины d_{ij}, задающие расстояния между узлами орграфа (между возможным пунктом размещения предприятий i и клиентом j), принимают значительно большие значения, то в качестве множеств K_s выбирались $K_4 = \{300, 250, 225, 200\}$ и $K_6 = \{170, 140, 120, 100, 75, 50\}$. Напомним, что для тестирования алгоритма при таком типе функции $\phi(\cdot)$ используется только второй тип релаксации Лагранжа $\mathcal{R}_2$ как показавший лучшие результаты в предыдущих экспериментах.

В таблице 2.5 представлены результаты для искусственно сгенерированных примеров. Как и в случае выпуклой функции $\phi(\cdot)$, столбцы 4 и 6 содержат среднее значение относительной погрешности между верхней и нижней оценками в процентах, а также среднее время вычислений для кусочно-линейной функции с 4 и 6 отрезками соответственно. Результаты для задач из тестовой библиотеки TSPLIB представлены в таблице 2.6.

Как можно заметить, для искусственно сгенерированных примеров малая относительная погрешность, меньшая одного процента, была получена для всех задач меньшей размерности (до 8000 точек для задач первого типа и до 10000 для задач второго типа). Для задач из библиотеки TSPLIB относительная погрешность Err также невелика в большинстве случаев (за исключением задач $d1291$ и $u2319$), однако время работы алгоритма значительно больше в случае использования кусочно-линейной функции с шестью отрезками.

2.5. Основные результаты второй главы

В главе рассмотрен нелинейный вариант задачи о p-медиане, в котором количество медиан, или, что то же самое, заданного количества открываемых предприятий, является переменной величиной. Дана постановка такой задачи в виде задачи целочисленного программирования, а также рассмотрены и исследованы два вида релаксаций Лагранжа, полу-

чающиеся за счет ослабления различных групп ограничений задачи. Для рассматриваемой задачи обобщен алгоритм поиска приближенных решений, представленный в первой главе диссертационной работы, в рамках которого для поиска нижних оценок оптимального значения использован метод релаксаций Лагранжа и специальный субградиентный метод, в то время как для поиска верхних оценок применена так называемая ядровая эвристика, основанная на линеаризованной постановке задачи. В главе также получены теоретические результаты, позволяющие повысить эффективность и скорость подсчета нижних оценок оптимального значения в методе релаксаций Лагранжа с учетом свойств нелинейной задачи. Разработанный эвристический алгоритм протестирован на двух наборах тестовых задач, часто используемых в литературе, для случая выпуклой и кусочно-линейной вогнутой функции $\phi(\cdot)$, ограничивающей число медиан. Полученные результаты вычислительных экспериментов свидетельствуют о высокой эффективности разработанного алгоритма как относительно времени вычислений, так и качества получаемых решений для задач рекордной размерности, которая не рассматривалась до этого в литературе.

Таблица 2.2. Численные результаты для искусственно сгенерированных примеров, полученные для выпуклой функции $\phi(\cdot)$

	θ_2^*						θ_1^*					
	Err(%)			Time			Err(%)			Time		
$\phi(p)$	p^2	$5p^2$	$10p^2$	p^2	$5p^2$	$10p^2$	p^2	$5p^2$	$10p^2$	p^2	$5p^2$	$10p^2$
n	Type I											
1000	0.007	0.056	0.005	4.02	7.13	5.66	0.009	0.382	1.132	5.00	7.98	11.18
2000	0.004	0.063	0.111	12.30	27.27	29.20	0.049	1.191	3.145	13.64	36.54	54.01
3000	0.128	0.291	0.069	26.56	40.64	58.41	0.139	1.767	4.162	29.95	69.41	91.76
4000	0.109	0.127	0.162	60.41	135.81	107.32	0.448	2.723	6.723	63.51	114.74	159.31
5000	0.024	0.078	0.047	82.55	118.86	159.58	0.853	5.242	21.254	100.70	193.21	228.50
6000	0.017	0.033	0.056	175.09	210.50	216.95	1.070	7.284	17.268	157.75	248.04	371.95
7000	0.103	0.012	0.015	167.65	243.61	239.18	1.802	16.545	17.186	362.65	350.74	497.09
8000	0.068	0.034	0.182	259.44	225.98	327.11	2.313	16.300	25.310	485.61	353.16	558.47
9000	0.107	0.057	0.067	391.52	351.75	368.07	4.222	17.382	24.753	432.46	466.22	633.04
10000	0.112	0.003	0.281	445.77	350.74	442.09	3.202	18.214	22.574	563.49	544.92	717.20
	Type III											
1000	0.001	0.000	0.000	3.74	4.72	5.16	0.005	0.240	10.799	4.31	5.99	9.26
2000	0.004	0.001	0.002	13.60	15.51	18.37	0.155	0.844	12.429	17.16	22.76	28.85
3000	0.040	0.001	0.001	31.11	31.79	37.07	0.342	1.586	4.178	38.83	44.68	61.70
4000	0.106	0.001	0.001	51.16	59.32	64.91	0.547	4.460	5.347	69.98	80.72	115.69
5000	0.029	0.017	0.014	109.67	109.55	101.90	1.428	6.462	7.859	114.12	138.59	165.97
6000	0.099	0.001	0.001	138.95	135.85	157.23	1.554	8.854	10.973	227.88	188.76	232.34
7000	0.154	0.002	0.100	252.22	195.20	196.17	1.739	7.722	11.765	303.14	286.07	301.95
8000	0.240	0.009	0.002	221.32	248.42	275.46	3.107	9.168	17.278	396.71	370.84	351.51
9000	0.113	0.017	0.001	410.50	329.41	329.48	8.039	12.040	18.453	552.69	375.27	449.92
10000	0.144	0.271	0.029	470.84	342.38	414.66	5.594	18.094	25.842	736.62	486.75	499.22

Таблица 2.3. Численные результаты для задач из тестовой библиотеки TSPLIB, полученные с использованием $\mathcal{R}_1$-релаксации и выпуклой функции $\phi(\cdot)$

Instance	Err(%)			Time		
$\phi(p)$	p^2	$5p^2$	$10p^2$	p^2	$5p^2$	$10p^2$
d1291	9.640	10.124	0.035	86.14	8.74	6.21
d1655	0.249	24.150	13.112	6.56	18.06	19.25
d2103	4.316	28.576	3.530	54.29	827.87	1064.16
dsj1000	11.300	0.013	1.769	5.62	2.27	5.17
fl417	0.029	0.001	0.002	0.85	1.05	1.05
fl1400	18.647	17.472	8.995	22.86	18.31	14.24
fl1577	28.744	24.552	2.665	13.57	11.89	14.12
fl3795	15.182	72.579	88.516	53.01	51.27	50.34
fnl4461	2.303	8.314	9.043	1050.77	791.87	1068.66
nrw1379	9.311	2.796	2.277	31.72	49.84	190.16
pcb1173	0.356	0.040	0.193	3.32	11.09	6.29
pr1002	4.628	0.131	0.746	7.16	6.83	2.58
pr2392	0.737	7.190	23.298	10.39	55.26	21.17
rl1889	0.800	2.243	10.060	9.17	36.47	14.88
rl5915	8.696	8.763	59.895	135.77	299.38	193.91
rl5934	0.905	7.777	5.657	141.05	1082.89	349.98
rl11849	3.557	54.951	18.055	1215.72	1321.63	1270.33
u1060	0.783	0.001	7.177	6.69	2.40	13.66
u1432	4.143	5.622	6.510	28.69	15.33	32.15
u1817	4.536	6.125	7.144	114.42	17.24	15.08
u2152	0.125	1.776	3.422	29.16	30.31	35.10
u2319	22.225	18.681	9.387	1118.55	1027.03	1019.65
vm1084	2.167	6.959	1.204	7.75	7.11	8.17
vm1748	1.426	3.580	2.077	13.76	14.87	13.63
pcb3038	2.672	10.226	7.756	91.96	86.55	51.19
usa13509	4.520	2.654	18.881	515.35	1247.71	1567.86

Таблица 2.4. Численные результаты для задач из тестовой библиотеки TSPLIB, полученные с использованием $\mathcal{R}_2$-релаксации и выпуклой функции $\phi(\cdot)$

Instance	Err(%)			Time		
$\phi(p)$	p^2	$5p^2$	$10p^2$	p^2	$5p^2$	$10p^2$
d1291	0.030	0.020	0.034	5.09	6.04	5.76
d1655	0.062	0.015	0.004	28.30	8.13	7.64
d2103	0.015	0.077	0.355	379.23	413.29	413.88
dsj1000	0.001	0.001	0.009	2.79	2.65	3.01
fl417	0.001	0.002	0.001	1.04	1.00	1.09
fl1400	0.010	0.001	0.001	6.06	5.97	7.05
fl1577	0.012	0.002	0.004	6.44	5.79	8.28
fl3795	0.755	0.005	0.005	30.58	40.31	45.54
fnl4461	0.050	0.760	0.070	362.37	58.98	447.52
nrw1379	0.017	0.020	0.087	5.59	6.43	11.98
pcb1173	0.008	0.013	0.064	4.24	4.34	8.33
pr1002	0.010	0.000	0.000	3.77	3.13	2.61
pr2392	0.001	0.020	0.000	13.77	11.81	11.59
rl1889	0.006	0.018	0.008	9.48	8.06	9.27
rl5915	0.007	0.019	0.018	70.60	70.12	73.87
rl5934	0.012	0.011	0.030	159.62	61.34	405.07
rl11849	0.020	0.043	0.971	506.90	523.82	394.69
u1060	0.013	0.001	0.003	4.58	3.40	2.96
u1432	0.039	0.023	0.041	9.32	6.32	7.74
u1817	0.049	0.005	0.001	22.95	8.69	10.83
u2152	0.033	0.019	0.012	12.09	12.98	15.26
u2319	0.219	0.891	0.811	50.61	415.52	817.46
vm1084	0.001	0.001	0.001	3.42	2.81	3.04
vm1748	0.001	0.002	0.001	6.44	6.19	7.52
pcb3038	0.021	0.018	0.042	31.22	19.91	28.31
usa13509	0.015	0.013	0.024	260.05	292.97	560.59

Таблица 2.5. Численные результаты для искусственно сгенерированных примеров, полученные с использованием $\mathcal{R}_2$-релаксации и кусочно-линейной вогнутой функции $\phi(\cdot)$

	Err(%)		Time	
k	$s = 4$	$s = 6$	$s = 4$	$s = 6$
n	Type I			
1000	0.001	0.000	3.20	10.03
2000	0.008	0.004	8.30	17.33
3000	0.082	0.102	17.32	29.06
4000	0.099	0.161	31.96	36.50
5000	0.093	0.057	64.67	121.33
6000	0.490	0.705	93.29	133.88
7000	0.461	0.454	293.65	194.24
8000	0.921	0.052	230.44	198.59
9000	0.871	0.859	426.12	448.25
10000	2.035	1.406	439.14	559.90
Type III				
1000	0.114	0.085	3.29	5.69
2000	0.004	0.161	10.30	15.98
3000	0.014	0.018	44.62	36.33
4000	0.070	0.202	69.92	77.52
5000	0.122	0.211	91.89	142.08
6000	0.571	0.587	205.16	124.38
7000	0.893	0.323	338.12	185.75
8000	0.412	1.366	276.46	416.15
9000	2.396	0.651	421.77	570.32
10000	1.583	0.497	630.57	558.83

Таблица 2.6. Численные результаты для искусственно сгенерированных примеров, полученные с использованием $\mathcal{R}_2$-релаксации и кусочно-линейной вогнутой функции $\phi(\cdot)$

Instance	Err(%)		Time	
k	$s = 4$	$s = 6$	$s = 4$	$s = 6$
d1291	0.380	1.265	6.69	143.40
d1655	0.074	0.008	40.06	134.55
d2103	0.548	0.620	519.52	1221.68
dsj1000	0.000	0.000	6.18	26.71
fl417	0.001	0.002	1.50	7.86
fl1400	0.001	0.002	7.48	38.44
fl1577	0.521	0.050	12.86	344.45
fl3795	0.529	0.085	79.95	717.35
fnl4461	0.086	0.524	432.93	2417.17
nrw1379	0.019	0.560	10.27	44.91
pcb1173	0.020	0.012	7.72	50.45
pr1002	0.082	0.000	14.74	36.49
pr2392	0.008	0.001	20.54	110.11
rl1889	0.004	0.001	18.33	76.11
rl5915	0.002	0.073	122.23	2127.14
rl5934	0.004	0.014	192.76	1739.04
rl11849	0.611	0.023	622.83	5418.43
u1060	0.068	0.000	6.98	32.31
u1432	0.609	0.000	12.24	368.13
u1817	0.170	0.818	33.27	73.04
u2152	0.014	0.034	26.89	76.27
u2319	1.097	0.000	4700.02	4786.08
vm1084	0.683	0.058	8.42	36.41
vm1748	0.176	0.001	16.20	69.09
pcb3038	0.008	0.031	71.11	714.17
usa13509	0.001	0.007	769.51	4140.02

Глава 3

Пороговая робастность в дискретных задачах размещения

Essentially, all models are wrong, but some are useful.

– George E. P. Box

В настоящей главе рассматривается один из подходов к определению робастности решения в непрерывных задачах размещения, известный также в литературе как пороговая робастность. Исследуется возможность обобщения данной концепции для случая дискретных задач размещения на примере простейшей задачи и задачи о p-медиане. В главе рассматриваются бикритериальные варианты данных дискретных задач, в которых, помимо основного, присутствует дополнительный критерий на робастность искомого решения, дается их постановка в виде нелинейных бикритериальных задач целочисленного программирования. Предлагается метод поиска приближенных Парето-оптимальных решений, так называемых δ-эффективных решений, в основе которого лежит известный метод ε-ограничений (метод главного критерия), учитывающий специфику рассматриваемых бикритериальных дискретных задач размещения. В конце главы приводятся результаты вычислительных экспериментов, а также проводится анализ и интерпретация полученных численных результатов.

3.1. Постановка задачи

В настоящей главе две классические модели из области дискретных задач размещения — задачу о p-медиане и простейшую задачу размещения — будем рассматривать в следующих постановках. Пусть, как и ранее, даны два множества: множество возможных пунктов размещения предприятий $I = \{1, \ldots, m\}$, производящих определенный продукт или оказывающих некоторую услугу, либо являющиеся некоторыми складами или пунктами распределения; а также множество клиентов $J = \{1, \ldots, n\}$. Пусть дополнительно для каждого клиента $j \in J$ имеется величина ω_j, определяющая спрос каждого клиента на продукт или услугу, а также даны величины $d_{ij} > 0$, задающие затраты на доставку одной единицы продукта или однократного оказания услуги из предприятия $i \in I$ клиенту $j \in J$. Тогда задача о p-медиане состоит в поиске таких p мест для открытия предприятий, чтобы суммарные затраты на

удовлетворение общего спроса всех клиентов были минимальны, т.е.

$$\min_{S \subseteq I} \Big\{ \sum_{j \in J} \omega_j \min_{i \in S} d_{ij} : \ |S| = p \Big\}. \tag{3.1}$$

Простейшая задача размещения (или задача размещения без ограничения на мощность производства) формулируется похожим образом [63, 191]. Однако, в отличие от задачи о p-медиане, предполагается, что количество открываемых предприятий не фиксировано. Вместо этого даны величины f_i, определяющие стоимость открытия предприятия в пункте $i \in I$. Тогда простейшая задача размещения может быть представлена в следующей комбинаторной постановке:

$$\min_{N \subseteq I} \Big\{ \sum_{j \in J} \omega_j \min_{i \in N} d_{ij} + \sum_{i \in N} f_i \Big\}. \tag{3.2}$$

В отличие от задачи о p-медиане из предыдущих глав, в приведенной выше постановке предполагается, что каждый клиент обладает спросом, т.е. в целевой функции задачи минимизируется взвешенная сумма расстояний от клиента $j \in J$ до предприятия, размещенного в пункте $i \in I$.

3.1.1. Неопределенности в дискретных задачах размещения

Многие задачи размещения требуют принятия стратегических решений, от правильности которых зависят вопросы обслуживания в течение, как правило, длительного периода времени, в пределах которого исходные условия могут измениться. Среди причин такого изменения могут быть, в случае постановок на сети, различные сбои или нарушения, вызванные отказами; или существенное отклонение параметров задачи (например, спроса клиентов, стоимости обслуживания, расстояний до предприятий) от изначальных значений; изменение внешних факторов, таких как действия конкурентов, демография потребителей и т.д. Также стоит заметить, что размещение некоторой группы предприятий, вообще говоря, связано с большими затратами и, как правило, производится для функционирования в течение долгого промежутка времени, а посему изменение уже имеющегося положения при изменении тех или иных условий зачастую является невозможным [262]. Более того, получение некоторых усредненных оценок неопределенных параметров задачи далеко не всегда является выходом из подобного рода ситуаций ввиду неточности данных или измерений. К тому же, как было отмечено в работе [70], в реальных практических приложениях задач линейного программирования игнорирование возможности наличия даже самой малой неточности или изменчивости данных может сделать оптимальное решение задачи абсолютно бессмысленным с практической точки зрения. В подобного рода ситуациях лицо, принимающее решения, мо-

жет предпочесть получить решение задачи размещения, в котором наличие тех или иных неопределенностей будет учитываться заранее. В литературе предложено достаточно много разнообразных подходов к различным оптимизационным задачам с разного рода неопределенностями, в том числе и для дискретных задач размещения [202, 204, 235, 262].

К основным параметрам задач размещения, в которых возможно возникновение неопределенностей, стоит отнести уровень спроса, время поездки от предприятия до клиента или стоимость обслуживания клиента предприятием, местоположение клиентов, их присутствие или отсутствие, стоимость товара или услуги [202]. При наличии неопределенностей для составления математической модели задачи критически важным является определение их типа. Например, неопределенные параметры могут принимать целые или непрерывные значения, быть выражены в виде случайных переменных с известными законами распределения, либо принадлежать некоторому известному отрезку или множеству. Отметим, что согласно известной классификации Розенхеда [250, 262] среда принятия решений может условно подразделяться на три категории: определенность, неопределенность, риск. В первую категорию попадают ситуации, в которых все параметры определены (детерминированы) и известны. В ситуациях, относящихся к категории риска, имеются неопределенные параметры, для которых, однако, известен закон распределения вероятностей. Наконец, в неопределенных ситуациях параметры не только не определены, но для них также не имеется никакой информации о вероятностях. Задачи, отвечающие ситуациям из категории риска, известны как задачи стохастического программирования [78], в которых, как правило, максимизируется или минимизируется математическое ожидание некоторой целевой функции. Задачи для ситуаций из категории неопределенности, в свою очередь, относятся к задачам так называемой робастной оптимизации, в рамках которой обычно оптимизируется эффективность системы в наихудшем случае. Таким образом, как в стохастическом программировании, так и в задачах робастной оптимизации ищется допустимое решение, являющееся в некотором смысле оптимальным для всех возможных вариантов неопределенных параметров. Причем при наличии какой-либо вероятностной информации неопределенные параметры описываются с помощью непрерывных или дискретных случайных величин. Если таковой информации не известно, то часто полагают, что неопределенные параметры принимают значения из некоторого определенного заранее интервала [262].

При моделировании дискретных задач размещения с наличием неопределенностей ключевым моментом является определение того, какие решения принимаются на этапе, предшествующем реализации того или иного сценария, а какие — как ответ или реакция на те или иные варианты развития событий, соответствующие некоторому осуществившемуся сце-

нарию. Отметим, что задачи размещения привлекли большое внимание исследователей в области теории принятия решений в условиях неопределенностей, в частности, из-за наличия ярко выраженной двухэтапной структуры таких задач. Так, в задачах размещения чаще всего полагают, что выбор мест расположения предприятий относится к первому этапу (ex ante), когда ничего не известно о сценарии или о реализации параметров. В то время как присоединение клиентов к тому или иному предприятию определяется на втором этапе (ex post) как ответ на полученные значения параметров. Отметим, что такая структура принятия решений в задачах размещения является распространенной, но не единственной [202]. В ряде задач возможно присоединение клиентов к предприятиям на первом этапе, а также наличие более сложной многоэтапной структуры принятия решений [230].

В настоящий момент в робастной оптимизации базовым подходом к моделированию неопределенностей, в том числе и в задачах дискретной оптимизации, является так называемый сценарный поход. Под сценарием понимается некоторый заданный набор значений неопределенных параметров вне зависимости от того, имеется ли какая-либо информация о вероятностях тех или иных значений. Стоит заметить, что если неопределенные параметры представляют собой некоторые случайные переменные, то для того или иного сценария можно определить вероятность его возникновения. В зависимости от задачи количество возможных сценариев может сильно варьироваться, в том числе и представлять бесконечное множество. Отметим, что недостатком сценарного подхода является сложность определения набора возможных сценариев, а в случае стохастических задач размещения — вероятности их реализаций. Другой недостаток состоит в том, что для более точного моделирования возможных исходов необходимо учитывать большое число возможных сценариев, что существенно сказывается на сложности итоговой оптимизационной задачи. Также стоит отметить, что при сценарном подходе определенную трудность вызывает выбор сценариев в случае непрерывных случайных величин, определяющих неопределенности задачи. Однако несомненным достоинством является легкость описания таких моделей, которые зачастую представляют собой исходные задачи размещения, содержащие большее число переменных и ограничений, что позволяет применять для их решения все многообразие известных алгоритмов. Из всего сказанного можно заметить, что сценарный подход может быть успешно применен как для задач стохастического программирования, так и для задач робастной оптимизации [190].

Одной из первых работ, посвященных исследованию неопределенностей в задачах линейного программирования, закладывающих основы новой тогда дисциплины — стохастического программирования, является работа Дж. Данцига [107], в которой рассматривался сценарный подход к описанию неопределенностей параметров, где те или иные сценарии возникали

с различной вероятностью. Основными недостатками такого подхода является то, что законы распределения вероятностей для тех или иных параметров задачи, а следовательно и для возникновения различных сценариев, редко удается определить на практике. К тому же, размерность задачи стремительно растет вместе с числом возможных сценариев, что вызывает значительную трудность и с вычислительной точки зрения.

Среди первых работ, посвященных исследованию неопределенностей данных в дискретных задачах размещения в общем и в задаче о p-медиане в частности с использованием сценарного подхода, стоит выделить [219], где авторами исследовалась задача поиска одной и двух медиан на дереве со случайными длинами ребер, которым соответствовало некоторое конечное множество сценариев. В качестве мотивации для исследования такой задачи авторами выделялись три фактора, влияющих на затрачиваемое транспортным средством время на путь от предприятия до клиента, а именно: сезонные, недельные или часовые циклы изменения средних значений объема трафика, случайные колебания интенсивности трафика, а также технические аварии или погодные условия. Решение задачи о 1-медиане было сведено к детерминированной задаче, в то время как для задачи о 2-медиане авторами предложен алгоритм выборочного перебора. Следующими известными статьями в данном направлении стали [219, 220, 271], в которых задача о поиске медиан обобщалась на случай сети со случайными длинами ребер и/или случайным спросом в вершинах, представляющим собой случайную величины с известным дискретным законом распределения вероятностей. Первая работа [219] носит по большей части теоретический характер и посвящена обобщению свойств задачи о p-медиане (например, свойства Хакими) на случай стохастической сети, а также на случай, если такая сеть, ко всему прочему, является ориентированной. Приведены математические формулировки таких задач, а также кратко описаны некоторые приложения. В работе [271] показано, что данная стохастической задача с использованием сценарного подхода может быть представлена в виде стандартной детерминированной задачи о p-медиане большей размерности. Отметим, что такой подход применим и к другим дискретным задачам размещения с неопределенностью в данных, определяемой в виде конечного набора сценариев. Для представленной формулировки задачи авторами предложен алгоритм на основе распространенного типа релаксации Лагранжа, рассматриваемого, в частности, в первой главе диссертационной работы. Наконец, в статье [220] для той же формулировки задачи был разработан алгоритм поиска решений, также основанный на методе релаксаций Лагранжа, но относительно ограничения, задающего количество искомых медиан. Полученная в результате такого преобразования подзадача (имеющая вид простейшей задачи размещения, в которой все предприятия обладают фиксированной стоимостью открытия, равной единствен-

ному множителю Лагранжа) решалась с помощью известного алгоритма DUALOC [124], в то время как для обновления значения множителя Лагранжа применялся субградиентный алгоритм.

Двухстадийная стохастическая задача о p-медиане и простейшая задача размещения, в которых присутствуют ограничения на мощность производства, исследовались в работах [204–206]. В данных задачах предполагалось, что стоимость производства, спрос, а также отпускная цена выражены случайными переменными. Для поиска решений в упомянутой двухстадийной простейшей задаче размещения в статье [206] обобщена эвристика DUALOC с учетом фиксированного набора сценариев.

Отметим, что в представленных выше работах неопределенные параметры выражались в виде случайных переменных с известными законами распределения вероятности, что, как было отмечено ранее, относит все такие задачи к области стохастического программирования. С другой стороны, если не известно никакой информации о вероятностях возникновения того или иного сценария (в случае дискретных параметров) или, в случае непрерывных величин, известны лишь интервалы значений неопределенных параметров, то в ход идут подходы так называемой робастной оптимизации.

Классическими критериями для такого рода задач в случае наличия конечного набора сценариев являются так называемые критерии минимаксной стоимости (minimax cost) и минимаксных потерь (minimax regret) [190, 202, 262]. В первом случае ищется решение задачи, минимизирующее максимальную стоимость решения для всех имеющихся сценариев. Во втором случае в качестве решения робастной задачи выбирается то, которое минимизирует максимальную разницу между значением целевой функции при возникновении данного сценария и оптимальным значением задачи для того же сценария.

Часто отмечаемым недостатком критерия минимаксной стоимости является излишне «консервативное» решение, поскольку фактически оно соответствует наихудшему возможному сценарию, который во многих случаях может быть маловероятен. Отметим, что такой критерий может быть вполне применим в тех случаях, когда критически важно, чтобы система приемлемо функционировала даже при наступлении наихудшего возможного сценария, что справедливо, например, при размещении пунктов аварийно-спасательных служб. В противном случае наиболее подходящим и используемым в литературе критерием робастности является критерий минимаксных потерь.

Одной из первых известных работ, посвященных применению данных критериев для дискретных задач размещения, является статья [255], в которой модель задачи о p-медиане использовалась для размещения пожарных станций в пределах городской черты. Авторы

отмечали, что неопределенности в исходных параметрах задачи могут часто возникать при размещении предприятий в пределах густонаселенной городской среды (агломераций и мегаполисов). Зачастую в такой ситуации время пути от предприятия до клиента может значительно различаться в зависимости от времени суток, что связано, например, с таким эффектом как маятниковая миграция. Очевидно, что в середине дня (рабочее время) основная масса населения сосредоточена в деловых районах города, в то время как в вечерние часы они пустеют, а большинство людей перемещается в спальные районы. В этом случае оптимальное размещение предприятий в рабочие и вечерние часы может сильно отличаться. Данная ситуация имеет особое значение, например, при размещении в пределах городской черты станций аварийно-спасательных служб (пунктов скорой помощи, пожарных станций и т.п.). Авторы предложили робастные постановки задачи о p-медиане относительно обоих представленных выше критериев. Для поиска решений был предложен эвристический алгоритм, основанный на классической эвристике Тейтса и Барта [267].

Также стоит выделить работу [256], в которой исследовалась задача размещения p предприятий в условиях высококонкурентной среды, т.е. в случае, когда предприятия некоторых конкурентов уже размещены. Целью задачи было максимизировать минимальную долю рынка или максимизировать потери при условии, что спрос клиентов точно не известен, однако задан в виде конечного набора сценариев. Предложены робастные постановки такой задачи с использованием двух представленных критериев, для поиска решений в которых также применялась эвристика [267].

Отметим, что зачастую на практике неопределенными являются непрерывные параметры задачи, для которых не всегда есть возможность определить конечный набор сценариев. Вместо этого известно, что данные параметры могут принимать значения из заданного множества, например, некоторого интервала. В общем случае в качестве таких множеств возможных значений неопределенных параметров выбирают параллелепипед или эллипсоидальные множества, которые, в частности, зачастую используют для аппроксимации более сложных множеств значений неопределенных параметров [73]. Отметим, что основным недостатком классических подходов робастной оптимизации к задачам линейного программирования [69] является прежде всего то, что робастные версии таких задач относятся к классу конических задач квадратичного программирования, которые, несмотря на выпуклость, представляют собой более сложные задачи. К тому же такие подходы не могут быть напрямую применены в случае задач целочисленного линейного программирования. Для преодоления указанных недостатков в работах [72, 73] был предложен другой подход, позволяющий получить целочисленную линейную робастную версию исходной задачи, сохраняющую ко всему прочему

вычислительную сложность, и избежать излишне «консервативных» решений, характерных для случая интервальных ограничений. В работе [72] эффективность данного подхода иллюстрируется на примерах известных задач дискретной оптимизации, таких как задача о рюкзаке, о потоке минимальной стоимости на сети, задачи управления портфелем.

В статье [231] рассматривается задача о p-медиане, в которой длины дуг графа не определены, но выбираются из некоторого заданного интервала. Авторами предлагается робастная постановка такой задачи, а также проводится численное сравнение найденного с помощью такой модели решения с результатами, полученными при использовании критерия минимаксных потерь.

В работе [66] исследуется мультипериодная задача размещения предприятий с ограничением на мощности производства. Предполагается, что спрос клиентов не задан и может изменяться от периода к периоду в пределах некоторого заданного множества. На первом этапе происходит размещение предприятий и выбираются их мощности, в то время как на последующих происходит распределение клиентов между предприятиями и определение действующей производственной мощности того или иного предприятия в зависимости от возникающего спроса. Авторами предлагаются две робастные версии такой задачи, в которых на возможные значения спроса клиентов наложены параллелепипедные ограничения или спрос может принимать значения из некоторого замкнутого эллипсоида.

3.1.2. Критерий пороговой робастности в дискретных задачах размещения

В настоящей главе рассматривается и исследуется иной подход к определению робастности решения в дискретных задачах размещения, также предполагающий сведение исходной задачи к нелинейной, а именно бикритериальной нелинейной задаче целочисленного программирования. В качестве неопределенных заранее параметров задачи будем рассматривать спрос клиентов ω_j, $j \in J$, для которого, как было отмечено, при стратегическом, долгосрочном размещении предприятий зачастую не удается получить не только точных оценок, но и какой-либо информации относительно распределения вероятностей. Более того, в классической теории робастной дискретной оптимизации предполагается, что для неопределенного непрерывного параметра задачи имеется некоторое множество (задаваемое параллелепипедными или эллипсоидными ограничениями) или интервал допустимых значения. Однако во многих случаях наличие таких, по сути, верхних и нижних оценок спроса клиентов также не всегда является возможным. Помимо долгосрочного размещения такая ситуация может возникнуть, например, в случае реализации совершенно нового товара или услуг, также при

планировании каких-либо уникальных или важных событий [88], для которых невозможно каким-либо образом спрогнозировать изменение спроса.

Пусть спрос каждого клиента ω_j точно не известен, однако выражен некоторой экспертной оценкой $\widehat{\omega}_j$. Отметим, что при замене вектора спроса его оценками погрешности могут привести к значительному росту значения целевой функции задачи, поэтому для контролирования затрат на обслуживание клиентов вводится положительная величина τ, ограничивающая максимальное возможное значение целевой функции в задаче. Величину τ можно интерпретировать как имеющийся общий бюджет на обслуживание клиентов и, в случае простейшей задачи, размещения набора предприятий. Тогда робастностью $\rho(S)$ допустимого решения $S \subseteq I$ в комбинаторной постановке задачи о p-медиане (3.1) будем называть минимальное отклонение вектора спроса ω относительно его оценок $\widehat{\omega}$, такое что суммарные затраты на обслуживание всех клиентов превысят бюджет τ, т.е.

$$\rho_{pM}(S) = \inf_{\omega \in \mathbb{R}^n_+} \Big\{ \|\omega - \widehat{\omega}\| : \sum_{j \in J} \omega_j \min_{i \in S} d_{ij} > \tau \Big\}, S \subseteq I, |S| = p. \tag{3.3}$$

Если вместо задачи о p-медиане рассматривать простейшую задачу размещения (3.2), то робастность допустимого решения $S \subseteq I$ может быть аналогичным образом выражена как

$$\rho_{UFLP}(S) = \inf_{\omega \in \mathbb{R}^n_+} \Big\{ \|\omega - \widehat{\omega}\| : \sum_{j \in J} \omega_j \min_{i \in N} d_{ij} + \sum_{i \in S} f_i > \tau \Big\}, S \subseteq I. \tag{3.4}$$

Отметим, что $\|\cdot\|$ представляет собой некоторую норму, от выбора которой, вообще говоря, напрямую зависит вид получаемой модели.

Одной из первых работ, посвященных исследованию такого определения робастности решения в непрерывных задачах размещения, называемой также пороговой робастностью, была статья [88]. В работе рассматривалась непрерывная задача поиска наиболее робастных решений в случае размещения единственного предприятия на плоскости относительно уже существующего множества клиентов — так называемая задача Вебера. Предложен метод поиска наиболее робастных решений на основе алгоритма Динкельбаха, а также разработаны процедуры поиска решений для одного частного случая выбора нормы, задающей отклонение спроса от его оценок, а именно манхэтеновского расстояния (расстояние городских кварталов). В [79] исследовались свойства такого подхода при применении к непрерывной задаче размещения с использованием конкурентной модели Хаффа. Для поиска наиболее робастного решения в такой задаче авторами применялся вариант метода ветвей и границ, известный как BSSS (big square small square). Работа [98] посвящена исследованию робастных свойств оптимальных решений двух известных непрерывных задач размещения с минисумным и

минимаксным критерием. Авторами показано, что решения таких задач являются наиболее робастными с точки зрения рассматриваемого критерия. Наконец отметим, что идея, близкая к рассматриваемой пороговой робастности, впервые исследовалась в работе [192] для оценки устойчивости решения в задаче размещения единственной медианы на сети с неопределенными весами узлов. Обозначаемая авторами как $\alpha^*(\tau)$, данная величина характеризовала максимальную разницу между значением целевой функции возмущенной задачи для оптимального решения, полученного с использованием оценок $\widehat{\omega}$, и оптимальным значением в возмущенной задаче для всех возможных возмущений, величина которых по норме не превосходит некоторого заданного порогового значения τ. В работе исследованы свойства $\alpha^*(\tau)$, а также предложена процедура ее подсчета для случая древовидной сети и задачи размещения на плоскости с функцией расстояния, порожденной некоторой полиэдральной нормой.

Исследуемые в настоящей главе задачи могут быть представлены в виде задач целочисленного линейного программирования с использованием стандартного набора переменных y_i, x_{ij}, $i \in I$, $j \in J$ (см. главу 1, (1.1)–(1.6)), $|I| = m$, $|J| = n$. Единственным отличием постановки простейшей задачи размещения от задачи о p-медиане является отсутствие ограничения, задающего количество открываемых предприятий, в то время как в целевой функции присутствует дополнительное слагаемое $\sum_{i \in I} f_i y_i$, задающее суммарную стоимость открытия предприятий. В дальнейшем для краткости будем обозначать множество допустимых решений в задаче о p-медиане как X_{pM}, а для простейшей задачи размещения как X_{UFLP}. Тогда в случае целочисленной постановки определение робастности некоторого допустимого решения (x, y) может быть записано как

$$\rho_{pM}(x, y) = \inf_{\omega \in \mathbb{R}^n_+} \left\{ \|\omega - \widehat{\omega}\| : \sum_{i \in I} \sum_{j \in J} \omega_j d_{ij} x_{ij} > \tau \right\} \tag{3.5}$$

в случае задачи о p-медиане и

$$\rho_{UFLP}(x, y) = \inf_{\omega \in \mathbb{R}^n_+} \left\{ \|\omega - \widehat{\omega}\| : \sum_{i \in I} \sum_{j \in J} \omega_j d_{ij} x_{ij} + \sum_{i \in I} f_i y_i > \tau \right\} \tag{3.6}$$

для простейшей задачи размещения.

Отметим, что для подсчета робастности решения необходимо решить нелинейную задачу минимизации, однако при некоторых дополнительных условиях значение робастности допустимого решения может быть вычислено в явном виде.

Теорема. *[79, 88] Для всяких допустимых решений $(x, y) \in X_{pM}$ и $(x', y') \in X_{UFLP}$ справед-*

либо

$$\rho_{pM}(x,y) = \max\left\{0, \frac{\tau - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}}{\|\sum_{i\in I} d_{ij}x_{ij}\|^\circ}\right\},$$
$$\rho_{UFLP}(x,y) = \max\left\{0, \frac{\tau - \sum_{i\in I} f_i y_i - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}}{\|\sum_{i\in I} d_{ij}x_{ij}\|^\circ}\right\}, \tag{3.7}$$

где $\|\cdot\|^\circ$ *обозначает двойственную относительно* $\|\cdot\|$ *норму.*

Из данной теоремы непосредственно следует следующий теоретический результат.

Теорема. *[88] Пусть* Z^* *является оптимальным решением в задаче о p-медиане, т.е.* $Z^* = \min\sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij},\ (x,y)\in X_{pM}$, *тогда*

1. *если* $Z^* \geq \tau$, *то* $\rho_{pM}(x,y) = 0\ \forall (x,y)\in X_{pM}$. *В частности, любое решение* $(x,y)\in X_{pM}$ *обладает максимальной робастнотью.*

2. *Если* $Z^* < \tau$, *то* $\max\limits_{(x,y)}\rho_{pM} > 0$. *Более того, допустимое решение* $(x',y')\in X_{pM}$ *является наиболее робастным (т.е. обладает максимально возможным значением робастности) тогда и только тогда, когда оно является оптимальным решением следующей задачи*
$$\max_{(x,y)} \frac{\tau - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}}{\|\sum_{i\in I} d_{ij}x_{ij}\|^\circ}, \qquad (x,y)\in X_{pM}. \tag{3.8}$$

Отметим, что данная теорема также справедлива и в случае, когда вместо задачи о p-медиане рассматривается простейшая задача размещения. Поскольку случай $Z^* \geq \tau$ тривиален, то в дальнейшем будем полагать, что имеющийся бюджет τ превосходит оптимальное значение в задаче, т.е. $Z^* < \tau$.

3.1.3. Бикритериальные робастные дискретные задачи размещения

В отличие от упомянутых выше работ, посвященных исследованию пороговой робастности, в настоящей главе этот подход применяется для случая дискретных задач размещения. Более того, для двух представленных моделей, рассматривается не задача поиска наиболее робастного решения, т.е. не оптимизационная задача (3.8), а бикритериальная постановка, в которой помимо основного критерия, минимизирующего суммарные затраты на обслуживание всех клиентов (и размещение предприятий), присутствует также критерий, максимизирующий робастность получаемых решений. Таким образом, робастные версии представленных дискретных задач размещения могут быть записаны в виде следующих бикритериальных

нелинейных задач целочисленного программирования:

$$\begin{array}{ll} \min\limits_{(x,y)} & \sum\limits_{i\in I}\sum\limits_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}, \\ \max\limits_{(x,y)} & \rho_{pM}(x,y), \\ & (x,y)\in X_{pM} \end{array} \qquad (BpM)$$

в случае задачи о p-медиане и

$$\begin{array}{ll} \min\limits_{(x,y)} & \sum\limits_{i\in I}\sum\limits_{j\in J}\widehat{\omega}_j d_{ij}x_{ij} + \sum\limits_{i\in I} f_i y_i, \\ \max\limits_{(x,y)} & \rho_{UFLP}(x,y), \\ & (x,y)\in X_{UFLP} \end{array} \qquad (BUFLP)$$

если рассматривается простейшая задача размещения. Другими словами, необходимо так разместить предприятия, чтобы не только минимизировать затраты на обслуживание клиентов, но и обеспечить максимально возможную «устойчивость» решения по отношению к изменениям спроса. Отметим, что многокритериальные задачи размещения исследуются уже на протяжении пятидесяти лет. Среди первых обзоров, посвященных такого рода постановкам, стоит выделить [100, 162]. Довольно подробный обзор можно найти в [224], а также в недавних статьях, посвященных непосредственно многокритериальным задачам размещения [127] и комбинаторной оптимизации [268]. В [181] производится теоретическое исследование многокритериальной задачи о p-медиане на сети с положительными и отрицательными весами узлов. Авторами предлагается эффективный алгоритм поиска Парето-оптимальных решений, полиномиальный в случае, когда количество медиан и критериев фиксировано. Различные варианты многокритериальной задачи размещения, в которой каждый из критериев минимизировал сумму расстояний от вершин до одной медианы, исследовались в [160]. В работе предлагаются полиномиальные алгоритмы поиска Парето-оптимальных решений, которые к тому же могут быть значительно улучшены в случае, если задача рассматривается на дереве.

3.2. Метод поиска аппроксимации множества Парето-оптимальных решений

Как известно, в задачах векторной оптимизации редко удается найти решение, минимизирующее одновременно все критерии, поэтому обычно происходит поиск некоторого компромисса. Данное компромиссное решение может быть определено различными способами. Поскольку дальнейший анализ идентичен для обеих исследуемых задач, то остановимся более подробно на случае задачи о p-медиане.

Определение 4. 1. Решение $(x^*, y^*) \in X_{pM}$ называется оптимальным по Парето (эффективным) в задаче (BpM), если не существует такого решения $(x, y) \in X_{pM}$, что

$$\begin{aligned} \sum_{i\in I}\sum_{j\in J} \widehat{\omega}_j d_{ij} x_{ij} &\leq \sum_{i\in I}\sum_{j\in J} \widehat{\omega}_j d_{ij} x^*_{ij}, \\ \rho_{pM}(x, y) &\geq \rho_{pM}(x^*, y^*), \end{aligned} \tag{3.9}$$

причем по крайней мере одно из неравенств выполняется строго.

2. Решение $(x^*, y^*) \in X_{pM}$ называется оптимальным по Слейтеру (слабо эффективным) в задаче (BpM), если не существует решения $(x, y) \in X_{pM}$, строго удовлетворяющего неравенствам (3.9).

Таким образом, необходимо найти такие решения $(x, y) \in X_{pM}$, что если лицо, принимающее решение, предполагает найти другое допустимое решение, обладающее большей робастностью или соответствующее меньшим суммарным затратам на обслуживание клиентов, то значение другого критерия ухудшится. Напомним, что множество всех Парето-оптимальных решений задачи называют множеством Парето, а образ эффективных решений в пространстве критериев называют Парето-фронтом или Парето-границей.

Для поиска решений, оптимальных по Парето, в многокритериальных задачах оптимизации могут быть применены различные точные и приближенные методы [119, 224]. Одним из наиболее популярных точных апостериорных алгоритмов для задач комбинаторной оптимизации и целочисленного программирования является метода ε-ограничений, также известный как метод главного критерия, впервые предложенный в [156]. Идея метода состоит в сведении исходной многокритериальной задачи к задаче с единственным выбранным главным критерием, в то время как все остальные критерии добавляются в формулировку задачи в виде ограничений с некоторыми параметрами ε_k, принимающими значения в пространстве критериев. Изменяя последовательно параметры ε_k, теоретически можно найти все Парето-множество. Отметим, что основной задачей, возникающей при применении метода ε-ограничений, является выбор значений параметров ε_k, для чего в общем случае необходимо применять некоторые процедуры поиска. Так, в случае бикритериальной задачи классическая схема метода предполагает увеличивать на каждом шаге метода значение единственного параметра ε на некоторую заданную величину δ, определяющую сетку в пространстве критериев. Вопрос выбора такого параметра является существенным. Поскольку на каждой итерации метода ε-ограничений возможно найти только одно решение, то выбор значения δ должен быть таким, чтобы не «пропустить» какое-либо Парето-оптимальное решение. С другой стороны, выбор слишком малого значения параметра δ может привести к

избыточным запускам процедуры поиска решений в подзадаче с одним критерием. Отметим, что для задач бикритериального целочисленного линейного программирования с целыми коэффициентами правило выбора значения параметра δ может быть легко реализовано, что делает метод главного критерия особенно эффективным и простым в реализации для такого рода задач, однако достаточно сложным с вычислительной точки зрения.

Одним из несомненных достоинств метода ε-ограничений является тот факт, что для его применения фактически нет необходимости накладывать дополнительные условия на критерии задачи или допустимое множество, например, условие выпуклости [92]. Другим преимуществом данного алгоритма является его теоретическая способность находить все Парето-оптимальные решения, в том числе и так называемые неопорные, которые невозможно получить с помощью распространенного метода линейной свертки критериев [119]. Главным недостатком, помимо отмеченного выше, является необходимость решения последовательности задач с одним критерием, что во многих случаях является довольно существенной проблемой в связи с тем, что большинство задач комбинаторной оптимизации являются NP-трудными. Кроме того, добавление дополнительных ограничений может, вообще говоря, нарушить структуру задачи, на использовании которой построены многие алгоритмы. Например, полиномиально разрешимая задача о кратчайшем пути становится задачей с ограничением на ресурсы, являющейся NP-трудной [121]. Обзор других методов скаляризации для задач целочисленного линейного программирования может быть найден в [120].

Метод главного критерия является довольно популярным алгоритмом поиска эффективных решений в задачах многокритериальной комбинаторной оптимизации. Так, в [122] данный метод успешно применялся для задачи составления расписаний полетов для экипажей самолетов, в основе которой лежала известная задача о разбиении множества. В дополнение к основному постановка содержала дополнительный критерий робастности получаемых решений, минимизирующий задержки между рейсами. Стоит отметить и одну из первых работ, посвященных применению метода, в которой исследовалась транспортная задача с дополнительным критерием, минимизирующим максимальное время доставки продукции. В качестве альтернативного критерия рассматривался также критерий, минимизирующий количество доставляемой продукции по маршрутам с максимальным временем доставки. Предложены несколько вариантов метода ε-ограничений для такой задачи [264]. В отечественной литературе метод ε-ограничений также использовался для решения бикритериальной задачи о назначениях, о покрывающем дереве и о 1-дереве с минисумными и минимаксными критериями [26, 27], задачи о назначениях и о покрывающем дереве с тремя минимаксны-

ми критериями [29], несимметричной бикритериальной задачи о коммивояжере [28], а также бикритериальной задачи о рюкзаке [30].

Большое число современных публикаций посвящено различным модификациям метода ε-ограничений, позволяющим преодолеть ряд его недостатков. Так, например, в [76] подробно исследован вариант алгоритма для случая бикритериальных задач комбинаторной оптимизации, целевые функции которых принимают только целые значения. Авторами показано, что с помощью метода главного критерия можно найти все Парето-оптимальные решения задачи, а также предложены некоторые эвристические процедуры, позволяющие избегать поиска слабоэффективных решений и существенно сократить время работы метода. Проведено численное тестирование алгоритма для бикритериальной задачи о коммивояжере с прибылью (TSPP). Другие модификации метода, связанные прежде всего с поиском решений, эффективных по Джеоффриону, представлены в [121]. В статье [214] предлагается новый улучшенный вариант метода, названный AUGMECON, позволяющий не только сократить время работы алгоритма, но и усилить метод, избегая поиска решений, оптимальных по Слейтеру. Вопросы выбора шага метода ε-ограничений для поиска всех эффективных решений для случая задач с более чем двумя критериями исследованы в [193].

При применении метода ε-ограничений для рассматриваемой задачи (BpM) возможны два варианта выбора параметрических подзадач, отличие которых заключается лишь в выборе основного критерия, а именно:

$$\begin{gathered}\min_{(x,y)} \sum_{i\in I}\sum_{j\in J} \widehat{\omega}_j d_{ij} x_{ij}, \\ \rho_{pM}(x,y) \geq \varepsilon, \\ (x,y) \in X_{pM};\end{gathered} \tag{3.10}$$

$$\begin{gathered}\max_{(x,y)} \rho_{pM}(x,y), \\ \sum_{i\in I}\sum_{j\in J} \widehat{\omega}_j d_{ij} x_{ij} \leq \varepsilon, \\ (x,y) \in X_{pM}.\end{gathered} \tag{3.11}$$

Основной трудностью в поиске решений таких задач целочисленного программирования является то, что функция $\rho_{pM}(x,y)$ — нелинейная. Однако, для некоторого частного выбора нормы, задающей отклонение спроса клиентов от его оценок, подзадача (3.10) может быть сведена к линейной задаче целочисленного программирования. Действительно, наиболее естественным выбором такой нормы является ℓ_∞ (норма Чебышева)[192], задающая максимальное координатное отклонение. Двойственной к ней нормой будет норма ℓ_1. В этом случае

подзадача (3.10) может быть переписана в следующем линейном виде:

$$\begin{aligned} \min_{(x,y)} & \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}, \\ & \sum_{i\in I}\sum_{j\in J}(\widehat{\omega}_j+\varepsilon)d_{ij}x_{ij}\le\tau, \\ & (x,y)\in X_{pM}. \end{aligned} \qquad (P_\varepsilon)$$

Отметим, что при значении $\varepsilon = 0$ дополнительное ограничение становится избыточным и задача P_0 является задачей о p-медиане, оптимальное решений которой в случае единственности является Парето-оптимальным в задаче (BpM), имеющим к тому же минимально возможное значение по обоим рассматриваемым критериям, и задает таким образом нижнюю оценку для Парето-оптимальных решений в пространстве критериев.

Поскольку входные данные задачи, такие как расстояние между клиентом и предприятиями d_{ij}, а также оценки спроса $\widehat{\omega}$ могут принимать не только целые значения, то и значение критериев бикритериальной задачи также не является целочисленным. Таким образом, применение известных подходов к выбору шага метода ε-ограничений в случае бикритериальных задач дискретной оптимизации для поиска всего Парето-множества не представляется возможным. Вместо этого в настоящей главе метод главного критерия применяется для поиска аппроксимации Парето-множества, а именно так называемых $\delta = (\delta_1, \delta_2)$-эффективных решений.

Определение 5. Для $\delta = (\delta_1, \delta_2) \in \mathbb{R}^2_+$ решение $(x^*, y^*) \in X_{pM}$ называется δ-эффективным в задаче (BpM), если не существует такого решения $(x, y) \in X_{pM}$, что

$$\begin{aligned} \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j c_{ij}x^*_{ij} &\ge \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j c_{ij}x_{ij}+\delta_1, \\ \rho_1(x^*,y^*) &\le \rho_1(x,y)-\delta_2. \end{aligned}$$

Причем по крайней мере одно из неравенств выполняется строго.

Таким образом, если при решении подзадач (P_ε) вместо точного решения находится приближенное с некоторой заданной точностью δ_1, то оно является δ-эффективным для бикритериальной задачи (BpM) при условии, что оно единственно. В связи с этим, для поиска таких решений вместо точных алгоритмов возможно применение различных эвристических подходов, что особенно актуально для случая бикритериальных дискретных задач оптимизации, поскольку большинство из них NP-трудны.

Для каждой подзадачи (P_ε) обозначим допустимое множество через $X_{pM}(\varepsilon)$, а множество всех δ_1-оптимальных решений — через $Sol_{pM}(\varepsilon, \delta_1)$. Как было отмечено ранее, главной

сложностью реализации метода ε-ограничений является выбор правила изменения параметра ε таким образом, чтобы не «пропускать» решения, оптимальные по Парето, и в то же время, чтобы каждая итерация метода давала по крайней мере одно новое эффективное решение. Идея предлагаемой схемы метода, во многом схожая с [264], а также близкая к упоминаемой в [193], предполагает достаточно простое правило варьирования параметра ε, учитывающее специфику рассматриваемой задачи. Представленный далее алгоритм генерирует последовательность $\delta = (\delta_1, \delta_2)$-эффективных решений для задачи (BpM), но может быть также легко модифицирован для простейшей задачи размещения. Алгоритм начинает работу с поиска решения в исходной задаче о p-медиане, т.е. в задаче P_0 ($\varepsilon = 0$). В качестве начального значения параметра ε_0 выбирается значение робастности оптимального решения (x^0, y^0) в задаче о p-медиане. Далее, вместо постепенного увеличение параметра ε_k с каждой итерацией метода его значение меняется каждый раз при нахождении нового оптимального решения в подзадаче (P_{ε_k}) и присваивается значению робастности данного решения с добавлением некоторого малого заданного значения $\delta_2' > \delta_2$.

Таким образом, общая схема алгоритма выглядит следующим образом:

Алгоритм 3.1 Поиск δ-эффективных решений в задаче (BpM)

0: Инициализация: Найти $(x^0, y^0) \in Sol_{pM}(0, \delta_1)$, положить $\varepsilon_0 := \dfrac{\tau - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij} x^0_{ij}}{\sum_{i\in I}\sum_{j\in J} d_{ij} x^0_{ij}}$, $k := 0$;

1: Положить $\bar{\varepsilon}_k := \varepsilon_k + \delta_2'$;

2: Решить подзадачу $(P_{\bar{\varepsilon}_k})$ с точностью δ_1;

3: **Если** $X_{pM}(\bar{\varepsilon}_k) = \varnothing$, **тогда** то stop, **иначе** положить $(\bar{x}, \bar{y}) \in Sol_{pM}(\bar{\varepsilon}_k, \delta_1)$;

4: **Если** $\sum\limits_{i\in I}\sum\limits_{j\in J}\widehat{\omega}_j d_{ij}\bar{x}_{ij} - \sum\limits_{i\in I}\sum\limits_{j\in J}\widehat{\omega}_j d_{ij} x^k_{ij} > \delta_1$, **тогда** $k := k + 1$.

5: Положить $(x^k, y^k) := (\bar{x}, \bar{y})$ и $\varepsilon_k := \dfrac{\tau - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij} x^k_{ij}}{\sum_{i\in I}\sum_{j\in J} d_{ij} x^k_{ij}}$ и перейти на 1.

Отметим, что в качестве критерия остановки представленного алгоритма ε-ограничений используется условие того, что допустимое множество задачи $(P_{\bar{\varepsilon}_k})$ для некоторого параметра $\bar{\varepsilon}_k$ пусто. Выполнимость такого критерия гарантируется конечностью множества X_{pM} и монотонностью последовательности ε_k. Другими словами, если для некоторого значения параметра $\bar{\varepsilon}_k$ множество $X(\bar{\varepsilon}_k)$ пусто, то алгоритм останавливается, поскольку не существует других точек, допустимых в задаче (BpM), робастность которых превосходила бы $\bar{\varepsilon}_k$. Это свойство сформулировано в виде следующего результата.

Предложение 3. *Справедливы следующие утверждения:*

1. *Если* $\varepsilon_1 \geq \varepsilon_2$, *то* $X_{pM}(\varepsilon_1) \subseteq X_{pM}(\varepsilon_2)$.

2. *Если* $X_{pM}(\varepsilon) = \varnothing$, *то* $\nexists (x, y) \in X_{pM} : \rho_{pM}(x, y) > \varepsilon$.

Доказательство. Справедливость первого утверждения следует из очевидного факта, что если $\varepsilon_1 \geq \varepsilon_2$, то ограничение $\sum\limits_{i\in I}\sum\limits_{j\in J}(\widehat{\omega}_j + \varepsilon_1)d_{ij}x_{ij} \leq \tau$ доминирует ограничение $\sum\limits_{i\in I}\sum\limits_{j\in J}(\widehat{\omega}_j + +\varepsilon_2)d_{ij}x_{ij} \leq \tau$, а следовательно $X_{pM}(\varepsilon_1) \subseteq X_{pM}(\varepsilon_2)$.

Справедливость второго утверждения следует из того очевидного факта, что если $X_{pM}(\varepsilon) = \varnothing$, то $\sum\limits_{i\in I}\sum\limits_{j\in J}(\widehat{\omega}_j + \varepsilon)d_{ij}x_{ij} > \tau \ \forall (x, y) \in X_{pM}$, а следовательно

$$\rho_{pM}(x, y) = \frac{\tau - \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}}{\sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij}} < \varepsilon \qquad \forall (x, y) \in X_{pM}.$$

□

Более того, для последовательности решений, генерируемых алгоритмом 3.1, справедлив следующий результат, аналогичный классическому в теории многокритериальной оптимизации (например, см. [92, 123, 272]).

Предложение 4. *Генерируемая алгоритмом 3.1 последовательность*

$$\{(x^0, y^0), (x^1, y^1), \ldots, (x^k, y^k)\}$$

конечна и для любого $\delta_2' > \delta_2$ *представляет собой множество* δ*-эффективных решений в задаче* (BpM). *Более того, если* $\delta_1 = 0$, *т.е. каждая подзадача* $(P_{\bar{\varepsilon}_k})$ *решается точно и найденное решение единственно, то* (x^j, y^j), $j = 0, \ldots, k$ *является оптимальными по Парето в задаче* (BpM).

Доказательство. Докажем только первое утверждение, поскольку справедливость второго показывается аналогично. Предположим, что некоторое решение $(x^j, y^j) \in Sol_{pM}(\bar{\varepsilon}_j, \delta_1)$, $j = = 1, \ldots, k$ не является δ-эффективным в задаче (BpM), тогда $\exists (x', y') \in X_{pM}$:

$$\begin{aligned} &\sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}^j > \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}' + \delta_1, \\ &\rho(x^j, y^j) < \rho(x', y') - \delta_2; \end{aligned} \tag{3.12}$$

либо

$$\begin{aligned} &\sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}^j = \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x_{ij}' + \delta_1, \\ &\rho(x^j, y^j) < \rho(x', y') - \delta_2; \end{aligned} \tag{3.13}$$

либо

$$\begin{aligned} &\sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x^*_{ij} > \sum_{i\in I}\sum_{j\in J}\widehat{\omega}_j d_{ij}x'_{ij} + \delta_1, \\ &\rho(x^j, y^j) = \rho(x', y') - \delta_2. \end{aligned} \tag{3.14}$$

Условия (3.12), (3.14) противоречат тому факту, что $(x^j, y^j) \in Sol_{pM}(\bar{\varepsilon}_j, \delta_1)$, а (3.13) — условию шага 4 алгоритма 3.1. Отсюда и из построения задачи $(P_{\bar{\varepsilon}_j})$, а также условия $\delta'_2 > \delta_2$, следует, что (x^j, y^j) является $\delta = (\delta_1, \delta_2)$-эффективным в задаче (BpM). □

Отметим, что аналогичный алгоритм и теоретические результаты можно получить и для бикритериальной простейшей задачи размещения $(BUFLP)$ с дополнительным критерием на робастность решения.

3.3. Вычислительный эксперимент

Представленная схема метода ε-ограничений была реализована на языке C++ и протестирована на компьютере с процессором Intel Core 2 Duo 2.2GHz и 3Gb оперативной памяти. В качестве решателя подзадач (P_{ε_k}), представляющих собой задачи целочисленного линейного программирования, использовался IBM ILOG CPLEX Optimizer 12.1.0[1]. Тестирование алгоритма производилось на всех метрических примерах из библиотеки TSPLIB с числом возможных пунктов размещения предприятий и клиентов от 50 до 500 (в общей сложности 37 задач), а также на примерах из тестовой библиотеки «Дискретные задачи размещения» (ДЗР)[2] [188], размерность которых составляет $|I| = |J| = 100$, а именно на задачах классов $Euclidean$, $Uniform$, $PCodes$, $Chess$ и FPP. В каждом классе имеется 30 тестовых задач. В случае бикритериальной задачи о p-медиане для примеров, взятых из библиотеки TSPLIB, результаты вычислительных экспериментов получены при различном выборе числа открываемых предприятий p, которое варьировалось от 5 до 50 в зависимости от размера задачи. Как для задачи о p-медиане, так и для простейшей задачи размещения бюджет τ задавался трех типов, а именно $\tau = 1.05Z^*$, $\tau = 1.1Z^*$, $\tau = 1.3Z^*$, где Z^* — оптимальное значение в соответствующей задаче. Оценки спроса $\hat{\omega}_j$ клиентов выбирались двух видов случайным образом с равномерным распределением из интервала от 10 до 100 и от 1000 до 10000. Величина шага δ'_2 принималась равной 0.01, в то время как параметр δ_1 предполагался равным 0, т.е. решение каждой подзадачи находилось точно.

[1] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[2] http://math.nsc.ru/AP/benchmarks/index.html

Для каждого рассматриваемого примера построена аппроксимация множества Парето-оптимальных решений, а именно множество δ-эффективных решений. Отметим, что такое множество, представленное в пространстве критериев, имеет вид, подобный изображенному на рис. 3.1, на котором представлены найденные δ-эффективные решения в случае бикритериальной задачи о p-медиане (BpM) для тестового примера *KroA200.tsp* при $p = 50$ и $\tau = 1.3Z^*$, или подобный изображенному на рис. 3.2, на котором представлена аппроксимация Парето-множества задачи $(BUFLP)$ для тестового примера *411Eucl* из библиотеки ДЗР, относящийся к классу *Euclidean*, при том же выборе бюджета τ. Из графиков можно заметить, что робастность решений изменяется незначительно относительно робастности оптимального решения задачи, являющегося первым найденным δ-эффективным решением, поэтому для существенного увеличения робастности лицу, принимающему решения, возможно, потребуется выбирать точки с большим значением суммарных затрат на обслуживание клиентов или увеличивать имеющийся в распоряжении бюджет. Данные наблюдения также подтверждаются полученными вычислительными результатами.

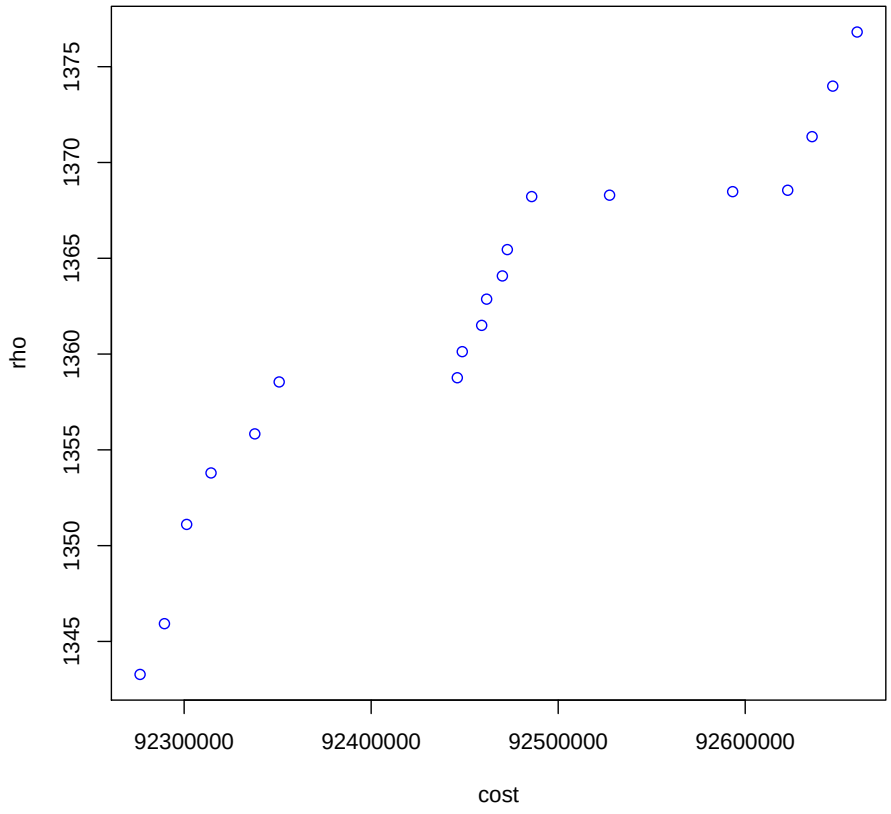

Рис. 3.1. Множество найденных δ-эффективных решений для (BpM) в задаче KroA200.tsp, $p = 50$ и $\tau = 1.3Z^*$

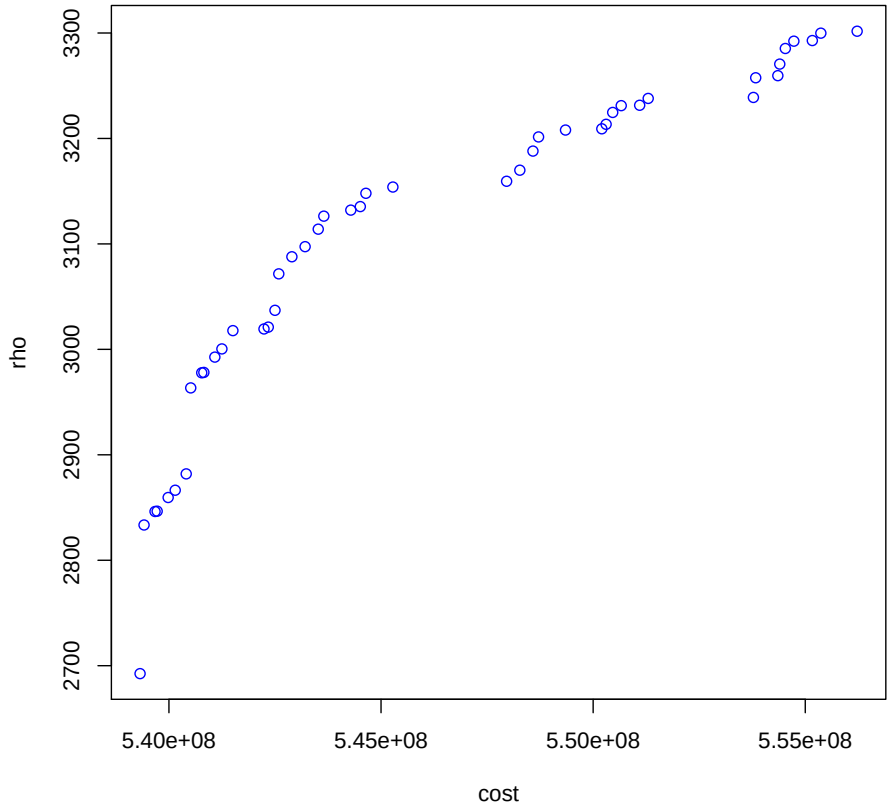

Рис. 3.2. Множество найденных δ-эффективных решений для $(BUFLP)$ в задаче 411Eucl, $\tau = 1.3Z^*$

Проанализируем теперь, как изменяется количество найденных δ-эффективных решений при изменении различных параметров, а также величины бюджета τ. Рассмотрим сначала случай бикритериальной задачи о p-медиане (BpM). Результаты вычислительного эксперимента для 37 тестовых примеров из библиотеки TSPLIB представлены в таблицах 3.1, 3.2 для случая $\widehat{\omega}_j \in [10, 1000]$ и таблицах 3.3, 3.4 для случая $\widehat{\omega}_j \in [1000, 10000]$ с использованием следующих обозначений: в столбце p представлено количество медиан, для которого производилось тестирование, в столбцах 1, 2, ... и т.д. показано количество примеров, в которых было найдено соответствующее число δ-эффективных решений при заданном выборе бюджета τ, т.е. в столбце 1 количество задач из 37, в которых найдено одно решение, в столбце 2 — два решения и т.д. Например, при бюджете $\tau = 1.05Z^*$ и при $p = 5$ в 32 из 37 тестовых примерах была найдено только одно δ-эффективное решение. Отметим, что в случае, когда количество медиан было равно или превышало 30, тестирование метода проводилось только для 31 задачи, поскольку из рассмотрения исключались примеры, в которых количество возможных пунктов размещения предприятий и клиентов не превосходит 100.

Таблица 3.1. Результаты для задач из библиотеки TSPLIB при $\widehat{\omega}_j \in [10, 100]$ и бюджетом τ, отличным на 5% и 10% от оптимального значения задачи о p-медиане (BpM)

	$\lvert S\rvert$						
	$\tau = 1.05Z^*$			$\tau = 1.1Z^*$			
p	1	2	3	1	2	3	4
5	34	2	1	31	5	1	0
10	34	3	0	33	4	0	0
15	32	5	0	27	9	1	0
20	29	8	0	22	14	1	0
30	30	1	0	21	9	0	1
40	24	6	1	16	11	2	2
50	28	3	0	20	8	2	1

Таблица 3.2. Результаты для задач из библиотеки TSPLIB при $\widehat{\omega}_j \in [10, 100]$ и бюджетом τ, отличным на 30% от оптимального значения задачи о p-медиане (BpM)

	$\lvert S\rvert$												
	$\tau = 1.3Z^*$												
p	1	2	3	4	5	6	7	8	9	10	12	15	17
5	22	12	3	0	0	0	0	0	0	0	0	0	0
10	17	13	4	1	2	0	0	0	0	0	0	0	0
15	10	16	3	5	1	0	2	0	0	0	0	0	0
20	8	11	7	5	4	1	0	1	0	0	0	0	0
30	6	3	8	4	4	3	1	0	1	0	0	0	1
40	2	5	4	5	2	1	3	4	1	1	1	2	0
50	1	5	6	5	5	1	3	2	1	1	0	1	0

Распределение количества найденных δ-эффективных решений при трех различных типах бюджета и семи рассмотренных значениях количества медиан p также представлены на стандартной статистической диаграмме типа «ящика с усами» (рис. 3.3) для случая $\widehat{\omega}_j \in [1000, 10000]$.

Результаты для тестовых примеров из библиотеки ДЗР для пяти классов задач при различных вариантах выбора спроса клиентов собраны в таблицах 3.5, 3.6, где первый столбец представляет класс рассматриваемых задач, в то время как другие обозначения аналогичны обозначениям в таблицах 3.1–3.4. Распределение количества найденных δ-эффективных решений при трех различных типах выбранного бюджета для пяти классов тестовых задач представлены на диаграмме типа «ящика с усами» (рис. 3.4) для случая $\widehat{\omega}_j \in [1000, 10000]$.

Таблица 3.3. Результаты для задач из библиотеки TSPLIB при $\widehat{\omega}_j \in [1000, 10000]$ и бюджетом τ, отличным на 5% и 10% от оптимального значения задачи о p-медиане (BpM)

	$\lvert S\rvert$													
	$\tau = 1.05Z^*$						$\tau = 1.1Z^*$							
p	1	2	3	4	5	6	1	2	3	4	5	6	7	10
5	32	5	0	0	0	0	31	6	0	0	0	0	0	0
10	29	8	0	0	0	0	21	15	0	1	0	0	0	0
15	26	10	1	0	0	0	18	11	3	3	2	0	0	0
20	28	8	1	0	0	0	15	16	3	2	0	0	1	0
30	18	10	1	1	0	1	12	7	6	2	1	0	2	1
40	19	8	3	1	0	0	10	8	7	2	3	0	1	0
50	19	10	0	0	1	1	9	12	1	3	4	1	1	0

Таблица 3.4. Результаты для задач из библиотеки TSPLIB при $\widehat{\omega}_j \in [1000, 10000]$ и бюджетом τ, отличным на 30% от оптимального значения задачи о p-медиане (BpM)

	$\lvert S\rvert$																
	$\tau = 1.3Z^*$																
p	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	18	19
5	23	13	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	13	13	3	3	3	2	0	0	0	0	0	0	0	0	0	0	0
15	6	11	6	6	4	1	0	0	1	1	0	1	0	0	0	0	0
20	4	10	5	8	2	3	1	1	1	1	0	1	0	0	0	0	0
30	4	5	4	2	2	5	2	4	0	1	0	1	1	0	0	0	0
40	0	1	7	3	4	3	3	3	2	1	0	2	1	0	1	0	0
50	2	2	3	6	6	2	0	1	3	0	2	0	0	1	1	1	1

Анализируя полученные результаты, можно заметить, что для подавляющего большинства примеров из библиотеки ДЗР множество найденных δ-эффективных решений состоит из единственного элемента, что особенно ярко выражается в случае относительно небольшого бюджета τ, равного $1.05Z^*$ или $1.1Z^*$, т.е. большего минимально возможного на 5% и 10% соответственно, и небольших значений оценок спроса клиентов, т.е. $\widehat{\omega}_j \in [10, 100]$. Максимальное количество найденных δ-эффективных решений в таком случае не превосходит 4 и получено только для одного примера класса *Euclidean* при бюджете $1.1Z^*$. При увеличении бюджета ситуация несколько меняется, а именно увеличивается количество задач, в которых были найдены 2 и 3 δ-эффективных решения. Для классов *Euclidean* и *Uniform* выявлены в общей сложности 4 примера, в которых найдены 5 и 6 δ-эффективных решений. Если оценки спроса принимают большие значения, т.е. из отрезка $[1000, 10000]$, то количество задач, в

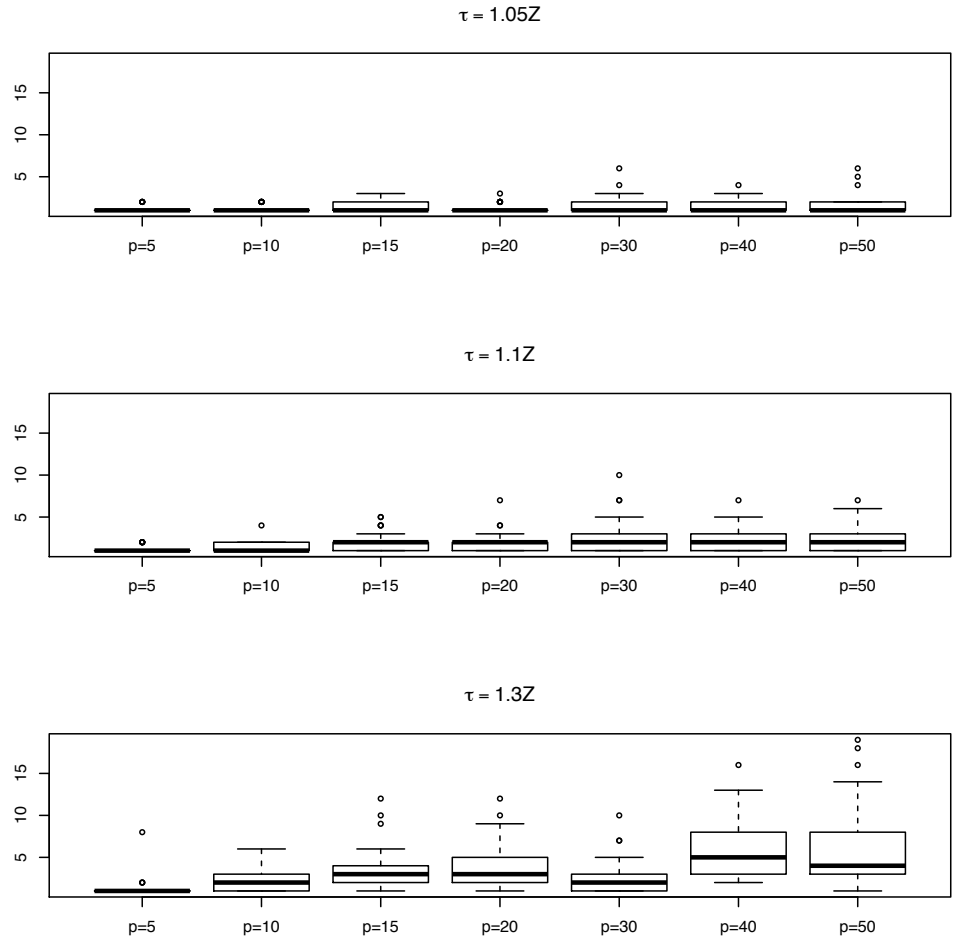

Рис. 3.3. Число найденных δ-эффективных решений для (BpM). Примеры из библиотеки TSPLIB, $\widehat{\omega}_j \in [1000, 10000]$

которых найдено только одно или два решения также сокращается, в том числе и с ростом бюджета. Так, наибольшее число найденных решений равно 6 и было получено только для одной задачи из класса $Euclidean$. Отметим, что наибольшее количество задач, в которых найдено два и более δ-эффективных решений, относятся к классу $Euclidean$ и, в меньшей степени, к классу $Uniform$.

Для примеров из библиотеки TSPLIB особенно ярко прослеживается тенденция к увеличению количества найденных δ-эффективных решений с увеличением бюджета τ и оценок спроса клиентов $\widehat{\omega}$. Так, максимальное их количество для задач с бюджетом $1.05Z^*$ и $1.1Z^*$ не превосходит 3 и 4, соответственно, при $\widehat{\omega}_j \in [10, 100]$, $j \in J$, а также 6 и 10 при $\widehat{\omega}_j \in [1000, 10000]$, в то время как максимальное количество найденных δ-эффективных решений при бюджете $1.3Z^*$ равно 17 и 19 соответственно. Для примеров из библиотеки TSPLIB также можно заметить тенденцию к увеличению числа δ-эффективных решений при увеличе-

Таблица 3.5. Результаты для 30 задач каждого из классов библиотеки ДЗР при $\widehat{\omega}_j \in [10, 100]$ и бюджетом τ, отличным на 5%, 10% и 30% от оптимального значения задачи о p-медиане (BpM)

	$\|S\|$											
	$\tau = 1.05Z^*$		$\tau = 1.1Z^*$				$\tau = 1.3Z^*$					
Class	1	2	1	2	3	4	1	2	3	4	5	6
Euclidean	27	3	22	6	1	1	9	12	5	1	1	2
Uniform	28	2	24	6	0	0	19	7	2	1	1	0
Chess	28	2	28	2	0	0	26	4	0	0	0	0
PCodes	30	0	29	1	0	0	26	3	1	0	0	0
FPP	30	0	28	2	0	0	25	5	0	0	0	0

Таблица 3.6. Результаты для 30 задач каждого из классов библиотеки ДЗР при $\widehat{\omega}_j \in [1000, 10000]$ и бюджетом τ, отличным на 5%, 10% и 30% от оптимального значения задачи о p-медиане (BpM)

	$\|S\|$											
	$\tau = 1.05Z^*$			$\tau = 1.1Z^*$			$\tau = 1.3Z^*$					
Class	1	2	3	1	2	3	1	2	3	4	5	6
Euclidean	24	4	2	19	10	1	9	16	1	1	2	1
Uniform	26	4	0	24	6	0	18	10	2	0	0	0
Chess	30	0	0	29	1	0	22	7	1	0	0	0
PCodes	29	1	0	28	2	0	26	3	1	0	0	0
FPP	29	1	0	28	2	0	25	5	0	0	0	0

нии числа медиан, что может быть вызвано большей гибкостью при присоединении клиентов, а следовательно возможностью обеспечения большей робастности решений.

Вычислительные эксперименты для бикритериальной робастной простейшей задачи размещения ($BUFLP$) во многом схожи с полученными для задачи (BpM), а посему не представлены здесь в полной мере. Отметим, что, как и для задачи о p-медиане, в большинстве случаев для обоих типов тестовых задач множество найденных δ-эффективных решений состоит из одного или двух элементов в случае малого бюджета $\tau = 1.05Z^*$. Однако при увеличении бюджета для простейшей задачи размещения наблюдается несколько большее число задач, в которых найдено более одного δ-эффективного решения. Так, например,максимальное число найденных δ-эффективных решений составляет 43 и получено для примера $411Eucl$.

В заключение проанализируем как изменяется значение критериев рассматриваемых бикритериальных задач для первого найденного δ-эффективного решения, являющегося оптимальным значением в соответствующей задачи, по сравнению с последним, обладающим максимальной возможной робастностью. В таблицах 3.7, 3.8, 3.9, а также 3.10, 3.11, 3.12 пред-

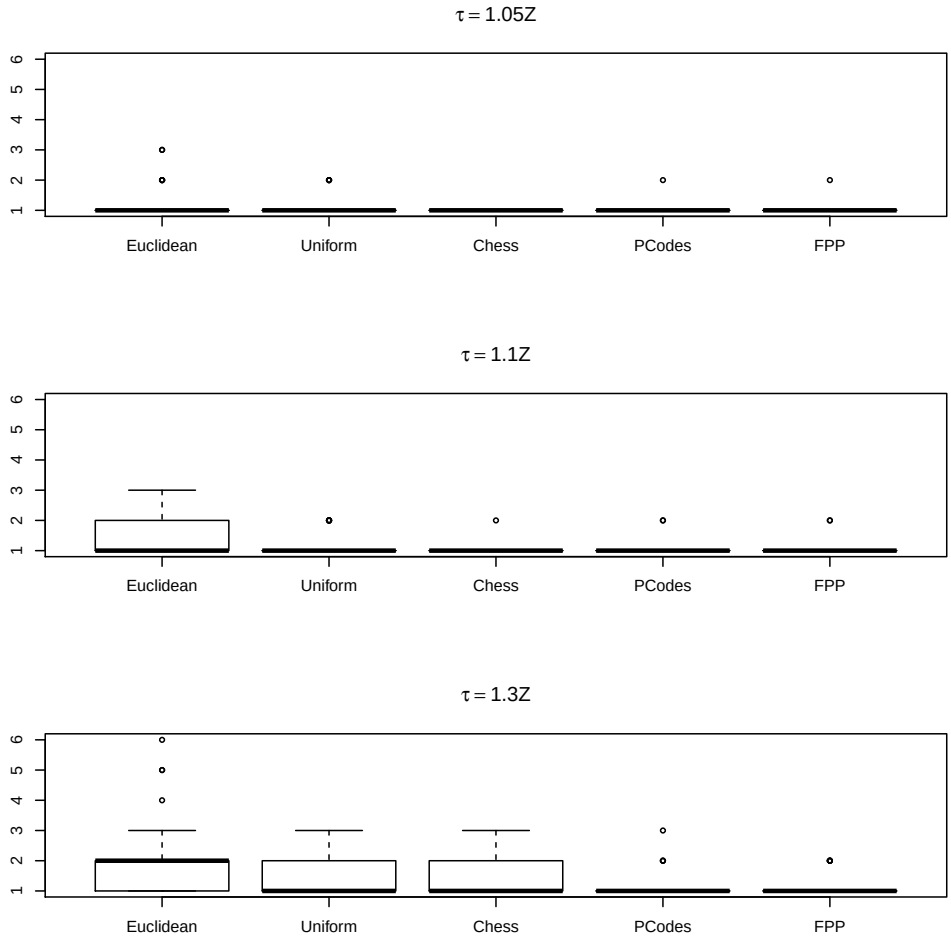

Рис. 3.4. Число найденных δ-эффективных решений для (BpM). Примеры из библиотеки ДЗР, $\widehat{\omega}_j \in [1000, 10000]$

ставлены результаты для бикритериальной робастной простейшей задачи размещения и задачи о p-медиане соответственно, полученные для 30 тестовых примеров из класса $Euclidean$. В первых двух столбцах таблиц даны названия каждого из примеров и количество открываемых предприятий. Столбцы $Obj.$ и $Obj.\ rob.$ содержат оптимальное значение задачи и соответствующее ему значение критерия робастности (для первого δ-эффективного решения) соответственно. В следующих двух столбцах $Max.\ obj.(\%)$ и $Max.\ rob.(\%)$ представлено увеличение значения основного критерия и критерия робастности для последнего найденного δ-эффективного решения по сравнению с первым решением, выраженное в процентах. Наконец, в столбцах N_δ и $Time(sec.)$ можно видеть количество элементов в полученной аппроксимации Парето-множества и время вычислений в секундах. Отметим, что поскольку в большом числе тестовых примеров было найдено только одно δ-эффективное решение, результаты для данных задач не представлены в таблицах 3.10, 3.11 и 3.12.

Проанализировав полученные результаты для $(BUFLP)$ можно заметить, что в случае относительно небольшого бюджета, т.е. $\tau = 1.05Z^*$ или $\tau = 1.10Z^*$, разница в значении основного критерия для первого и последнего δ-эффективного решения во всех случаях составляет не более 1%, в то время как значение робастности меняется более, чем на 10% (задача 2611 в таблице 3.8). Аналогичные результаты можно наблюдать и для случая $\tau = 1.3Z^*$ (таблица 3.9), где максимальное изменение значения робастности составляет более 25% для задачи 2611, в то время как изменение основного критерия простейшей задачи размещения для данного примера находится в пределах 3%. Схожие результаты можно наблюдать и для случая задачи (BpM), однако разница в значениях по обоим критериям не так велика как для бикритериальной простейшей задачи размещения.

На основании результатов можно сделать вывод, что если для лица, принимающего решения, робастность имеет первостепенное значение, то более предпочтительными являются решения, обладающие робастностью, близкой к максимально возможной, однако дающие большее значение суммарной стоимости обслуживания клиентов (и размещения предприятий). В этом случае увеличение расходов на обслуживание спроса клиентов может составлять всего несколько процентов, в то время как рост значения робастности такого решения может быть значительным. Также стоит отметить, что количество найденных δ-эффективных решений, в особенности для бикритериальной задачи о p-медиане (BpM), является относительно небольшим, даже для сравнительно большого бюджета τ. В связи с этим, у лица, принимающего решения, имеется не так много возможностей для выбора более эффективных путей обслуживания клиентов на протяжении длительного промежутка времени, и, возможно, в этом случае более предпочтительным является выбор оптимального решения задачи о p-медиане, робастность которого относительно исследуемого критерия также достаточно велика.

Таблица 3.7. Вычислительный эксперимент. Задача ($UFLP$), примеры класса $Euclidean$, бюджет $\tau = 1.05Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

$Prob.$	p	$Obj.$	$Obj.\ rob.$	$Max.\ obj.(\%)$	$Max.\ rob.(\%)$	N_δ	$Time(sec.)$
1011	12	504369705	392.57	0.26	9.03	3	2.68
1111	13	494317552	396.74	0.34	0.38	2	1.83
111	12	504985897	406.18	0.18	0.26	4	2.70
1211	16	531543135	504.07	0.03	3.34	2	1.51
1311	16	555805265	503.00	0.10	5.77	3	2.09
1411	16	536716881	527.25	0.15	2.51	3	2.05
1011	12	504369705	392.57	0.26	9.03	3	2.49
1111	13	494317552	396.74	0.34	0.38	2	1.73
111	12	504985897	406.18	0.18	0.26	4	2.62
1211	16	531543135	504.07	0.03	3.34	2	1.50
1311	16	555805265	503.00	0.10	5.77	3	2.28
1411	16	536716881	527.25	0.15	2.51	3	2.15
1511	14	547352584	456.21	0.14	1.94	3	2.49
1611	16	527511332	506.95	–	–	1	1.13
1711	13	508186287	415.75	–	–	1	1.07
1811	15	541531310	462.72	0.06	2.74	2	1.58
1911	14	515686282	428.95	–	–	1	1.01
2011	16	522756729	505.68	–	–	1	1.04
2111	12	510483434	424.50	0.15	0.98	2	1.95
211	14	483758444	424.25	0.14	5.39	5	3.85
2211	16	531927970	505.39	–	–	1	0.96
2311	13	508395882	420.43	0.27	4.94	3	2.22
2411	13	521012649	439.29	0.16	0.78	3	2.25
2511	14	530371536	429.76	0.20	2.46	3	2.30
2611	12	542521681	425.40	0.11	8.20	4	2.86
2711	14	499536471	471.97	–	–	1	0.93
2811	16	553757141	512.10	0.14	3.08	4	2.64
2911	14	534040677	454.19	0.04	3.36	2	1.49
3011	13	496903018	407.89	0.20	0.21	2	1.62
311	15	507895781	483.74	–	–	1	0.98
411	15	539319652	448.75	0.22	5.63	3	2.16
511	14	503902198	433.60	0.03	0.12	2	1.44
611	14	521253101	440.64	–	–	1	1.01
711	16	553653000	514.84	–	–	1	0.91
811	15	519653291	455.18	0.03	0.49	2	1.50
911	13	519455489	409.74	–	–	1	0.91

Таблица 3.8. Вычислительный эксперимент. Задача ($UFLP$), примеры класса *Euclidean*, бюджет $\tau = 1.1Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

Prob.	p	*Obj.*	*Obj. rob.*	*Max. obj.*(%)	*Max. rob.*(%)	N_δ	*Time*(*sec.*)
1011	12	504369705	785.13	0.73	12.03	5	3.99
1111	13	494317552	793.47	0.83	6.23	6	5.25
111	12	504985897	812.36	0.55	4.14	9	6.82
1211	16	531543135	1008.14	0.05	3.68	4	2.69
1311	16	555805265	1006.00	0.10	6.73	3	2.31
1411	16	536716881	1054.50	0.52	5.56	7	4.99
1511	14	547352584	912.42	0.67	6.36	10	10.10
1611	16	527511332	1013.90	–	–	1	0.92
1711	13	508186287	831.50	–	–	1	0.88
1811	15	541531310	925.44	0.33	4.95	3	2.32
1911	14	515686282	857.89	–	–	1	0.93
2011	16	522756729	1011.37	0.50	1.92	6	4.53
2111	12	510483434	848.99	0.41	3.46	3	4.09
211	14	483758444	848.51	0.14	6.72	5	3.66
2211	16	531927970	1010.79	0.47	0.59	4	2.87
2311	13	508395882	840.85	0.27	7.60	3	2.45
2411	13	521012649	878.59	0.90	4.90	6	5.36
2511	14	530371536	859.53	0.63	7.38	6	5.34
2611	12	542521681	850.80	1.08	10.90	8	7.13
2711	14	499536471	943.95	–	–	1	0.93
2811	16	553757141	1024.21	0.14	4.50	4	2.70
2911	14	534040677	908.39	0.13	3.78	3	2.13
3011	13	496903018	815.79	0.57	3.90	4	3.36
311	15	507895781	967.48	–	–	1	0.97
411	15	539319652	897.50	0.80	8.86	15	13.46
511	14	503902198	867.20	0.03	0.45	2	1.65
611	14	521253101	881.28	0.36	2.82	3	2.48
711	16	553653000	1029.69	0.30	1.58	2	1.66
811	15	519653291	910.36	0.03	0.75	2	1.48
911	13	519455489	819.49	–	–	1	0.95

Таблица 3.9. Вычислительный эксперимент. Задача ($UFLP$), примеры класса *Euclidean*, бюджет $\tau = 1.3Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

Prob.	p	*Obj.*	*Obj. rob.*	*Max. obj.*(%)	*Max. rob.*(%)	N_δ	*Time*(*sec.*)
1011	12	504369705	2355.40	3.73	22.56	21	27.28
1111	13	494317552	2380.42	3.37	12.25	30	39.21
111	12	504985897	2437.07	3.12	11.08	32	40.54
1211	16	531543135	3024.43	0.88	5.90	14	16.98
1311	16	555805265	3018.00	0.96	9.25	8	8.76
1411	16	536716881	3163.49	2.34	12.54	20	23.34
1511	14	547352584	2737.26	3.69	19.24	34	67.10
1611	16	527511332	3041.70	3.15	5.69	14	18.36
1711	13	508186287	2494.49	1.35	6.86	10	18.50
1811	15	541531310	2776.32	5.10	13.22	22	33.05
1911	14	515686282	2573.67	3.11	6.69	14	20.17
2011	16	522756729	3034.11	2.72	7.95	17	18.00
2111	12	510483434	2546.98	4.44	15.54	35	120.72
211	14	483758444	2545.52	1.84	9.75	20	24.85
2211	16	531927970	3032.37	1.77	5.69	43	62.82
2311	13	508395882	2522.56	2.32	13.76	14	19.14
2411	13	521012649	2635.77	2.88	13.34	21	33.67
2511	14	530371536	2578.58	1.85	15.35	23	38.78
2611	12	542521681	2552.40	2.84	24.87	30	63.39
2711	14	499536471	2831.84	3.21	7.76	19	33.40
2811	16	553757141	3072.62	2.17	11.10	14	17.51
2911	14	534040677	2725.16	2.44	10.67	12	20.66
3011	13	496903018	2447.36	2.21	10.29	10	16.66
311	15	507895781	2902.43	3.55	10.33	19	39.79
411	15	539319652	2692.51	3.04	18.45	45	88.15
511	14	503902198	2601.59	3.07	8.02	15	22.79
611	14	521253101	2643.85	3.85	10.85	36	65.15
711	16	553653000	3089.06	3.39	9.33	18	33.57
811	15	519653291	2731.09	2.55	9.36	13	24.79
911	13	519455489	2458.46	3.88	7.53	32	61.92

Таблица 3.10. Вычислительный эксперимент. Задача (BpM), примеры класса $Euclidean$, бюджет $\tau = 1.05Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

$Prob.$	p	$Obj.$	$Obj.\ rob.$	$Max.\ obj.(\%)$	$Max.\ rob.(\%)$	N_δ	$Time(sec.)$
1511	13	335430412	262.70	0.01	1.16	2	0.70
1611	14	301530848	245.89	0.05	4.23	3	1.03
1811	14	310803856	247.80	0.01	2.85	2	0.66
2511	16	269600407	249.19	0.05	0.07	2	0.67
2811	14	325194266	269.61	0.01	0.06	2	0.68
811	16	258747810	240.37	0.05	0.03	2	0.70

Таблица 3.11. Вычислительный эксперимент. Задача (BpM), примеры класса $Euclidean$, бюджет $\tau = 1.1Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

$Prob.$	p	$Obj.$	$Obj.\ rob.$	$Max.\ obj.(\%)$	$Max.\ rob.(\%)$	N_δ	$Time(sec.)$
111	12	306985897	493.84	0.02	0.12	2	0.68
1511	13	335430412	525.41	0.01	1.24	2	0.68
1611	14	301530848	491.77	0.05	4.73	3	1.01
1711	15	266924062	486.46	0.01	0.05	2	0.65
1811	14	310803856	495.60	0.01	2.94	2	0.64
2411	14	290629595	504.64	0.07	0.36	2	0.77
2511	16	269600407	498.37	0.05	0.56	2	0.65
2611	12	344521681	540.29	0.03	0.19	2	0.66
2811	14	325194266	539.21	0.01	0.18	2	0.76
511	14	272902198	469.65	0.06	0.17	2	0.64
811	16	258747810	480.74	0.05	0.55	2	0.67

Таблица 3.12. Вычислительный эксперимент. Задача (BpM), примеры класса $Euclidean$, бюджет $\tau = 1.3Z^*$, $\widehat{\omega}_j \in [1000, 10000]$

$Prob.$	p	$Obj.$	$Obj.\ rob.$	$Max.\ obj.(\%)$	$Max.\ rob.(\%)$	N_δ	$Time(sec.)$
1011	13	290083242	1442.63	0.38	0.82	4	1.42
1111	14	265014044	1375.48	0.15	0.40	2	0.70
111	12	306985897	1481.52	0.02	0.24	2	0.70
1311	14	329017402	1644.73	0.17	0.27	2	0.86
1411	13	328785222	1655.88	0.87	0.48	3	1.06
1511	13	335430412	1576.22	0.01	1.29	2	0.63
1611	14	301530848	1475.32	0.36	5.95	5	1.69
1711	15	266924062	1459.37	0.01	0.09	2	0.62
1811	14	310803856	1486.79	0.01	3.00	2	0.59
1911	13	301190886	1450.24	0.47	0.18	2	0.78
2111	12	312483434	1559.09	0.25	0.52	2	0.83
2211	15	284549916	1529.26	0.29	0.53	6	2.08
2411	14	290629595	1513.93	0.07	0.85	2	0.63
2511	16	269600407	1495.12	0.05	0.89	2	0.55
2611	12	344521681	1620.87	0.03	0.36	2	0.57
2811	14	325194266	1617.64	0.01	0.26	2	0.66
2911	14	303040677	1546.39	0.17	0.44	2	0.64
411	15	291819652	1456.89	0.52	0.78	5	1.74
511	14	272902198	1408.96	0.06	0.57	2	0.66
611	14	290253101	1472.20	0.01	0.01	2	0.81
811	16	258747810	1442.22	0.05	0.89	2	0.59

3.4. Основные результаты третьей главы

В данной главе один из известных ранее подходов к определению робастности решения для непрерывных задач размещения был обобщен на случай дискретных задач и успешно применен для двух известных моделей: задачи о p-медиане и простейшей задачи размещения. На основании данного подхода были предложены новые бикритериальные нелинейные постановки задачи о p-медиане и простейшей задачи размещения, в которых помимо критерия, минимизирующего суммарные затраты на обслуживание клиентов, присутствует дополнительный критерий, максимизирующий робастность решения. Для одного частного случая критерия робастности, полученного с использованием нормы Чебышева, был разработан алгоритм поиска приближенных решений (так называемых δ-эффективных решений) в рассматриваемых бикритериальных задачах, основанный на известном, классическом методе главного критерия (или методе ε-ограничений) с учетом специфики рассматриваемых моделей. Исследуемый подход к определению робастности и разработанный алгоритм поиска δ-эффективных решений были протестированы на широком наборе известных в литературе тестовых задач. На основе полученных данных сделан вывод об эффективности рассматриваемого подхода, а также возможных случаях выбора того или иного найденного δ-эффективного решения лицом, принимающим решения.

Заключение

Основные результаты диссертационной работы состоят в следующем.

1. Для нелинейного варианта задачи о p-медиане для двух видов релаксации получены явные формулы вычисления значений целевых функций двойственных задач и прямых переменных для некоторого фиксированного набора множителей. Разработан и протестирован алгоритм поиска приближенных решений, показавший свою эффективность для задач большой размерности.

2. Предложены новые бикритериальные нелинейные постановки задачи о p-медиане и простейшей задачи размещения с дополнительным критерием, максимизирующим робастность решения. Разработан, обоснован и протестирован алгоритм поиска приближенных Парето-оптимальных (δ-эффективных) решений в данных задачах.

3. Разработан и протестирован параллельный алгоритм поиска нижних оценок оптимального значения в классической задаче о p-медиане, включающий в себя процедуру каскадной сборки и специальную модель хранения данных и показавший эффективность для задачи о p-медиане большой размерности.

Список литературы

1. Антонов А. С. Параллельное программирование с использованием технологии MPI. М.: Изд-во МГУ, 2004. 71 с.
2. Антонов А. С. Параллельное программирование с использованием технологии OpenMP. М.: Изд-во МГУ, 2009. 77 с.
3. Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стандартизации. Новосибирск: Наука, 1978. 336 с.
4. Васильев Ф. П. Численные методы решения экстремальных задач. М.: Наука, 1988. 520 с.
5. Васильев И. Л. Метод декомпозиции для задачи о p-медиане на несвязном графе // Дискретн. анализ и исслед. опер. 2007. Т. 14, № 1. С. 43–58.
6. Васильев И. Л. Метод отсечений для многогранника задачи о рюкзаке // Известия РАН. Теория и системы управления. 2009. № 1. С. 74–81.
7. Васильев И. Л., Ушаков А. В. Релаксации Лагранжа для нелинейной задачи о p-медиане // Изв. Иркутского гос. ун-та. Сер. Математика. 2011. Т. 4, № 2. С. 45–59.
8. Васильев И. Л., Ушаков А. В. Об одном подходе к робастности решения в задаче о p-медиане // Изв. Иркутского гос. ун-та. Сер. Математика. 2012. Т. 5, № 4. С. 2–15.
9. Васильев И. Л., Ушаков А. В. Параллельная реализация субградиентного алгоритма для максимизации двойственной функции Лагранжа в задаче о p-медиане // Вычислительные методы и программирование. 2013. Т. 14. С. 9–16.
10. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб: БХВ-Петербург, 2002. 608 с.
11. Гончаров Е. Н., Кочетов Ю. А. Вероятностный поиск с запретами для дискретных задач безусловной оптимизации // Дискретн. анализ и исслед. опер., сер. 2. 2002. Т. 9, № 2. С. 13–30.
12. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. М.: Мир, 1982. 416 с.
13. Демьянов В. Ф., Малоземов В. Н. К теории нелинейных минимаксных задач // Успехи матем. наук. 1971. Т. 26, № 3(159). С. 53–104.
14. Еремин И. И. О несовместных системах линейных неравенств // Доклады АН СССР. 1961. Т. 138, № 6. С. 1280—1283.
15. Еремин И. И. Итеративный метод для чебышевских приближений несовместных систем линейных неравенств // Доклады АН СССР. 1962. Т. 143, № 6. С. 1254–1256.

16. Еремин И. И. Обобщение релаксационного метода Моцкина-Агмона // Успехи матем. наук. 1965. Т. 20, № 2. С. 183–187.
17. Еремин И. И. Методы фейеровских приближений в выпуклом программировании // Матем. заметки. 1968. Т. 3, № 2. С. 217–234.
18. Ермольев Ю. М. Методы решения нелинейных экстремальных задач // Кибернетика. 1966. № 4. С. 1–17.
19. Ермольев Ю. М. Методы стохастического программирования. М.: Наука, 1976. 240 с.
20. Заикин О. С., Посыпкин М. А., Семёнов А. А., Храпов Н. П. Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home // Вестник ННГУ. 2012. № 5(2). С. 340–347.
21. Измаилов А. Ф., Солодов М. В. Численные методы оптимизации. М.: ФИЗМАТЛИТ, 2005. 304 с.
22. Коновалов Н. А., Крюков В. Д., Погребцов А. Д., Сазанов Ю. Л. C-DVM язык разработки мобильных параллельных программ // Программирование. 1999. № 1. С. 20–28.
23. Кочетов Ю. А., Пащенко М. Г., Плясунов А. В. О сложности локального поиска в задаче о p-медиане // Дискретн. анализ и исслед. опер., сер. 2. 2005. Т. 12, № 2. С. 44–71.
24. Леванова Т. В., Лореш М. А. Алгоритмы муравьиной колонии и имитации отжига для задачи о p-медиане // Автоматика и телемеханика. 2004. № 3. С. 80–88.
25. Макконел К. Р., Брю С. Л., Флинн Ш. М. Экономикс. Принципы, проблемы и политика: Пер. с 19-го англ. изд. М.: Инфра-М, 2013. 1028 с.
26. Меламед И. И., Сигал И. Х. Исследование линейной свертки критериев в многокритериальном дискретном программировании // Ж. вычисл. матем. и матем. физ. 1995. Т. 35, № 8. С. 1260–1270.
27. Меламед И. И., Сигал И. Х. Вычислительное исследование линейной параметризации критериев в многокритериальном дискретном программировании // Ж. вычисл. матем. и матем. физ. 1996. Т. 36, № 10. С. 23–25.
28. Меламед И. И., Сигал И. Х. Исследование линейной свертки критериев в бикритериальной задаче коммивояжера // Ж. вычисл. матем. и матем. физ. 1997. Т. 37, № 8. С. 933–936.
29. Меламед И. И., Сигал И. Х. Вычислительное исследование трехкритериальных задач о деревьях и назначениях // Ж. вычисл. матем. и матем. физ. 1998. Т. 38, № 10. С. 1780–1787.
30. Меламед И. И., Сигал Н. Х., Владимирова Н. Ю. Исследование линейной свертки критериев в бикритериальной задаче о ранце // Ж. вычисл. матем. и матем. физ. 1999.

Т. 39, № 5. С. 753–758.

31. Поляк Б. Т. Один общий метод решения экстремальных задач // Докл. АН СССР. 1967. Т. 174, № 1. С. 33–36.
32. Поляк Б. Т. Минимизация негладких функционалов // Ж. вычисл. матем. и матем. физ. 1969. Т. 9, № 3. С. 509–521.
33. Рокафеллар Р. Выпуклый анализ. М.: Мир, 1973. 465 с.
34. Харари Ф. Теория графов. 2-е изд. М.: Едиториал УРСС, 2003. 296 с.
35. Шор Н. З. Применение метода градиентного спуска для решения сетевой транспортной задачи // Материалы науч. семинара по теорет. и прикл. вопр. кибернетики и исследования операций. Вып. 1, № 6. Киев: Ин-т кибернетики АН УССР, 1962. С. 9–17.
36. Шор Н. З. О скорости сходимости обобщенного градиентного спуска // Кибернетика. 1968. Т. 4, № 3. С. 98–99.
37. Шор Н. З. Использование операций растяжения пространства в задачах минимизации выпуклых функций // Кибернетика. 1970. № 1. С. 6–12.
38. Шор Н. З. О скорости сходимости обобщенного градиентного спуска с растяжением пространства // Кибернетика. 1970. № 2. С. 80–85.
39. Шор Н. З. Методы минимизации недифференцируемых функций и их приложения. Киев: Наукова думка, 1979. 200 с.
40. Шор Н. З., Гамбурд П. Р. Некоторые вопросы сходимости обобщенного градиентного спуска // Кибернетика. 1971. № 6. С. 82–84.
41. Шор Н. З., Журбенко Н. Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов // Кибернетика. 1971. № 3. С. 51–59.
42. Шор Н. З., Журбенко Н. Г., Лиховид А. П., Стецюк П. И. Развитие алгоритмов недифференцируемой оптимизации и их приложения // Кибернетика и системный анализ. 2003. № 4. С. 82–93.
43. Эндрюз Г. Р. Основы многопоточного, параллельного и распределенного программирования. : Пер. с англ. М.: Издательский дом "Вильямс 2003. 512 с.
44. Юдин Д. В., Немировский А. С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач // Экономика и математические методы. 1976. Т. 12, № 2. С. 357–369.
45. 50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art / Ed. by M. Jünger, T. Liebling, D. Naddef et al. Berlin: Springer, 2010. 824 p.
46. Agmon S. The relaxation method for linear inequalities // Canad. J. Math. 1954. Vol. 6.

P. 382–392.

47. Agra A., Cardoso D. M., Cerdeira J. O. et al. Solving huge size instances of the optimal diversity management problem // J. of Math. Science. 2009. Vol. 161, no. 6. P. 956–960.
48. Aguado J. S. Fixed Charge Transportation Problems: a new heuristic approach based on Lagrangean relaxation and the solving of core problems // Ann. Oper. Res. 2009. Vol. 172, no. 1. P. 45–69.
49. Alekseeva E., Kochetov Y., Plyasunov A. Complexity of local search for the p-median problem // Eur. J. Oper. Res. 2008. Vol. 191, no. 3. P. 736–752.
50. Alp O., Erkut E., Drezner Z. An Efficient Genetic Algorithm for the p-Median Problem // Annals of Operations Research. 2003. Vol. 122, no. 1-4. P. 21–42.
51. Amdahl G. M. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities // Proceedings of the Spring Joint Computer Conference. AFIPS '67 (Spring). New York: ACM, 1967. P. 483–485.
52. Anstreicher K. M., Wolsey L. A. Two "well-known" properties of subgradient optimization // Math. Program. 2009. Vol. 120, no. 1. P. 213–220.
53. Arya V., Garg N., Khandekar R. et al. Local Search Heuristics for k-Median and Facility Location Problems // SIAM J. Comput. 2004. Vol. 33, no. 3. P. 544–562.
54. Avella P., Benati S., Martinez L. C. et al. Some personal views on the current state and the future of locational analysis // Eur. J. Oper. Res. 1998. Vol. 104, no. 2. P. 269–287.
55. Avella P., Boccia M., Martino C. D. et al. A decomposition approach for a very large scale optimal diversity management problem // 4OR. 2005. Vol. 3, no. 1. P. 23–37.
56. Avella P., Boccia M., Salerno S., Vasilyev I. An aggregation heuristic for large scale p-median problem // Comput. Oper. Res. 2012. Vol. 39, no. 7. P. 1625–1632.
57. Avella P., Boccia M., Sforza A., Vasilyev I. An effective heuristic for large-scale capacitated facility location problems // J. Heuristics. 2008. Vol. 15, no. 6. P. 597–615.
58. Avella P., Sassano A. On the p-Median polytope // Math. Program. 2001. Vol. 89, no. 3. P. 395–411.
59. Avella P., Sassano A., Vasil'ev I. A heuristic for large-scale p-median instances // Electron. Notes in Discrete Math. 2003. Vol. 13, no. 0. P. 14–17. 2nd Cologne-Twente Workshop on Graphs and Combinatorial Optimization.
60. Avella P., Sassano A., Vasilyev I. Computational study of large-scale p-median problems // Math. Program. 2007. Vol. 109, no. 1. P. 89–114.
61. Avella P., Sforza A. Logical reduction tests for the p-problem // Ann. Oper. Res. 1999. Vol. 86, no. 0. P. 105–115.

62. Bahiense L., Maculan N., Sagastizábal C. The volume algorithm revisited: relation with bundle methods // Math. Program. 2002. Vol. 94, no. 1. P. 41–69.
63. Balinski M. L. Integer Programming: Methods, Uses, Computations // Manage. Sci. 1965. Vol. 12, no. 3. P. 253–313.
64. Barahona F., Anbil R. The volume algorithm: producing primal solutions with a subgradient method // Math. Program. 2000. Vol. 87, no. 3. P. 385–399.
65. Barnhart C., Johnson E. L., Nemhauser G. L. et al. Branch-and-Price: Column Generation for Solving Huge Integer Programs // Oper. Res. 1998. Vol. 46, no. 3. P. 316–329.
66. Baron O., Milner J., Naseraldin H. Facility Location: A Robust Optimization Approach // Prod. Oper. Manag. 2011. Vol. 20, no. 5. P. 772–785.
67. Bazaraa M. S., Sherali H. D. On the choice of step size in subgradient optimization // Eur. J. Oper. Res. 1981. Vol. 7, no. 4. P. 380–388.
68. Beasley J. E. Lagrangean heuristics for location problems // Eur. J. Oper. Res. 1993. Vol. 65, no. 3. P. 383–399.
69. Ben-Tal A., Ghaoui L. E., Nemirovski A. Robust Optimization. Princeton Series in Applied Mathematics. Princeton: Princeton University Press, 2009. 576 p.
70. Ben-Tal A., Nemirovski A. Robust solutions of linear programming problems contaminated with uncertain data // Math. Program. 2000. Vol. 88, no. 3. P. 411–424.
71. Benjaafar S., ElHafsi M., de Véricourt F. Demand Allocation in Multiple-Product, Multiple-Facility, Make-to-Stock Systems // Manage. Sci. 2004. Vol. 50, no. 10. P. 1431–1448.
72. Bertsimas D., Sim M. Robust discrete optimization and network flows // Math. Program. 2003. Vol. 98, no. 1-3. P. 49–71.
73. Bertsimas D., Sim M. The Price of Robustness // Oper. Res. 2004. Vol. 52, no. 1. P. 35–53.
74. Bertsimas D., Tsitsiklis J. N. Introduction to Linear Optimization. 3rd edition. Belmont: Athena Scientific, 1997. Vol. 6 of Athena Scientific Series in Optimization and Neural Computation. 608 p.
75. Bertsimas D., Weismantel R. Optimization over Integers. Belmont: Dynamic Ideas, 2005. 602 p.
76. Bérubé J.-F., Gendreau M., Potvin J.-Y. An exact ε-constraint method for bi-objective combinatorial optimization problems: Application to the Traveling Salesman Problem with Profits // Eur. J. Oper. Res. 2009. Vol. 194, no. 1. P. 39–50.
77. Bilde O., Krarup J. Sharp Lower Bounds and Efficient Algorithms for the Simple Plant Location Problem // Studies in Integer Programming / Ed. by P. L. Hammer, E. L. Johnson, B. H. Korte, G. L. Nemhauser. Elsevier, 1977. Vol. 1 of Annals of Discrete Mathematics.

P. 79–97.

78. Birge J. R., Louveaux F. Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering. 2nd edition. New York: Springer, 2011. 485 p.
79. Blanquero R., Carrizosa E., Hendrix E. M. T. Locating a competitive facility in the plane with a robustness criterion // Eur. J. Oper. Res. 2011. Vol. 215, no. 1. P. 21–24.
80. Bonnans J. F., Gilbert J. C., Lemaréchal C., Sagastizábal C. A. Numerical Optimization: Theoretical and Practical Aspects. Universitext. 2nd edition. Berlin: Springer, 2006. 494 p.
81. Bradley P. S., Fayyad U. M., Mangasarian O. L. Mathematical Programming for Data Mining: Formulations and Challenges // INFORMS J. Comput. 1999. Vol. 11, no. 3. P. 217–238.
82. Brent R. P. The Parallel Evaluation of General Arithmetic Expressions // J. ACM. 1974. Vol. 21, no. 2. P. 201–206.
83. Briant O., Naddef D. The optimal diversity management problem // Oper. Res. 2004. Vol. 52, no. 4. P. 515–526.
84. Brodtkorb A. R., Hagen T. R., Schulz C., Hasle G. GPU computing in discrete optimization. Part I: Introduction to the GPU // EURO J. Transp. Logist. 2013. Vol. 2, no. 1-2. P. 129–157.
85. Bussieck M. R., Ferris M. C., Meeraus A. Grid-Enabled Optimization with GAMS // INFORMS J. Comput. 2009. Vol. 21, no. 3. P. 349–362.
86. Byrka J., Pensyl T., Rybicki B. et al. An Improved Approximation for K-median, and Positive Correlation in Budgeted Optimization // Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms / Ed. by P. Indyk. SODA '15. Philadelphia: SIAM, 2015. P. 737–756.
87. Caprara A., Fischetti M., Toth P. A heuristic method for the set covering problem // Oper. Res. 1999. Vol. 47, no. 5. P. 730–743.
88. Carrizosa E., Nickel S. Robust facility location // Math. Methods Oper. Res. 2003. Vol. 58, no. 2. P. 331–349.
89. Carrizosa E., Ushakov A., Vasilyev I. A computational study of a nonlinear minsum facility location problem // Comput. Oper. Res. 2012. Vol. 39, no. 11. P. 2625–2633.
90. Carrizosa E., Ushakov A., Vasilyev I. Threshold robustness in discrete facility location problems. A bi-objective approach // Optim. Lett. 2015. Vol. 9, no. 7. P. 1297–1314.
91. Ceria S., Nobili P., Sassano A. A Lagrangian-based heuristic for large-scale set covering problems // Math. Program. 1998. Vol. 81, no. 2. P. 215–228.
92. Chankong V., Haimes Y. Multiobjective Decision Making: Theory and Methodology. New York: North-Holland, 1983. Vol. 8 of North-Holland Series in System Science and Engineering. 424 p.

93. Charikar M., Guha S., Tardos É., Shmoys D. B. A Constant-Factor Approximation Algorithm for the k-Median Problem // J. Comput. Syst. Sci. 2002. Vol. 65, no. 1. P. 129–149.
94. Chen Q., Ferris M. C. FATCOP: A Fault Tolerant Condor-PVM Mixed Integer Programming Solver // SIAM J. Optim. 2001. Vol. 11, no. 4. P. 1019–1036.
95. Chen Q., Ferris M. C., Linderoth J. FATCOP 2.0: Advanced Features in an Opportunistic Mixed Integer Programming Solver // Ann. Oper. Res. 2001. Vol. 103, no. 1-4. P. 17–32.
96. Christofides N., Beasley J. E. A tree search algorithm for the p-median problem // Eur. J. Oper. Res. 1982. Vol. 10, no. 2. P. 196–204.
97. Church R. L., Meadows M. Location Modeling Utilizing Maximum Service Distance Criteria // Geographical Analysis. 1979. Vol. 11, no. 4. P. 358–379.
98. Ciligot-Travain M., Traoré S. On a robustness property in single-facility location in continuous space // TOP. 2014. Vol. 22, no. 1. P. 321–330.
99. Cohen M. A., Moon S. An integrated plant loading model with economies of scale and scope // Eur. J. Oper. Res. 1991. Vol. 50, no. 3. P. 266–279.
100. Cohon J. L. Multiobjective Programming and Planning. New York: Academic Press, 1978. Vol. 140 of Mathematics in Science and Engineering. 333 p.
101. Conforti M., Cornuéjols G., Zambelli G. Integer Programming. Cham: Springer, 2014. Vol. 271 of Graduate Texts in Mathematics. 456 p.
102. Cornuejols G., Fisher M. L., Nemhauser G. L. Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms // Manage. Sci. 1977. Vol. 23, no. 8. P. 789–810.
103. Correa R., Lemaréchal C. Convergence of some algorithms for convex minimization // Math. Program. 1993. Vol. 62, no. 1–3. P. 261–275.
104. Crainic T. G., Gendreau M., Hansen P., Mladenović N. Cooperative parallel variable neighborhood search for the p-median // J. Heuristics. 2004. Vol. 10, no. 3. P. 293–314.
105. Crainic T. G., Toulouse M. Parallel Meta-heuristics // Handbook of Metaheuristics / Ed. by M. Gendreau, J.-Y. Potvin. New York: Springer, 2010. Vol. 146 of International Series in Operations Research & Management Science. P. 497–541.
106. Culler D., Karp R., Patterson D. et al. LogP: Towards a Realistic Model of Parallel Computation // Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. PPOPP '93. New York: ACM, 1993. P. 1–12.
107. Dantzig G. B. Linear Programming under Uncertainty // Manage. Sci. 1955. Vol. 1, no. 3-4. P. 197–206.
108. Dantzig G. B., Thapa M. N. Linear Programming 2: Theory and Extensions. New York:

Springer-Verlag, Inc., 1997. P. 520.

109. Dantzig G. B., Wolfe P. Decomposition Principle for Linear Programs // Oper. Res. 1960. Vol. 8, no. 1. P. 101–111.
110. Daskin M. S. Network and Discrete Location: Models, Algorithms, and Applications. 2nd edition. Hoboken: Wiley, 2013. 536 p.
111. de Farias Jr. I. R. A family of facets for the uncapacitated p-median polytope // Oper. Res. Lett. 2001. Vol. 28, no. 4. P. 161–167.
112. de Vries S., Posner M. E., Vohra R. V. Polyhedral Properties of the K-median Problem on a Tree // Math. Program. 2007. Vol. 110, no. 2. P. 261–285.
113. Diestel R. Graph Theory. 4th edition. Berlin: Springer, 2010. Vol. 173 of Graduate Texts in Mathematics. 410 p.
114. Discrete Location Theory / Ed. by P. B. Mirchandani, R. L. Francis. New York: John Wiley & Sons, Inc., 1990. 555 p.
115. Domínguez E., Muñoz J. A neural model for the p-median problem // Comput. Oper. Res. 2008. Vol. 35, no. 2. P. 404–416. Part Special Issue: Location Modeling Dedicated to the memory of Charles S. ReVelle.
116. du Merle O., Goffin J. L., Vial J. P. On Improvements to the Analytic Center Cutting Plane Method // Comput. Optim. Appl. 1998. Vol. 11, no. 1. P. 37–52.
117. Dupont L. Branch and bound algorithm for a facility location problem with concave site dependent costs // International Journal of Production Economics. 2008. Vol. 112, no. 1. P. 245–254. Special Section on Recent Developments in the Design, Control, Planning and Scheduling of Productive Systems.
118. Efroymson M. A., Ray T. L. A Branch-Bound Algorithm for Plant Location // Oper. Res. 1966. Vol. 14, no. 3. P. 361–368.
119. Ehrgott M. Multicriteria Optimization. 2nd edition. Berlin: Springer, 2005. 323 p.
120. Ehrgott M. A discussion of scalarization techniques for multiple objective integer programming // Ann. Oper. Res. 2006. Vol. 147, no. 1. P. 343–360.
121. Ehrgott M., Ruzika S. Improved ε-Constraint Method for Multiobjective Programming // Journal of Optimization Theory and Applications. 2008. Vol. 138, no. 3. P. 375–396.
122. Ehrgott M., Ryan D. M. Bicriteria robustness versus cost optimisation in tour of duty planning at Air New Zealand // Proceedings of the 35th Annual Conference of the Operational Research Society of New Zealand / Victoria University of Wellington. 2000. P. 31–39.
123. Engau A., Wiecek M. M. Generating ε-efficient solutions in multiobjective programming // Eur. J. Oper. Res. 2007. Vol. 177, no. 3. P. 1566–1579.

124. Erlenkotter D. A dual-based procedure for uncapacitated facility location // Oper. Res. 1978. Vol. 26, no. 6. P. 992–1009.
125. Facility Location: Applications and Theory / Ed. by Z. Drezner, H. W. Hamacher. 2nd edition. Berlin: Springer, 2004. 460 p.
126. Facility Location: Concepts, Models, Algorithms and Case Studies / Ed. by R. Z. Farahani, M. Hekmatfar. Contributions to Management Science. Heidelberg: Physica, 2009. 549 p.
127. Farahani R. Z., SteadieSeifi M., Asgari N. Multiple criteria facility location problems: A survey // Appl. Math. Model. 2010. Vol. 34, no. 7. P. 1689–1709.
128. Feldman E., Lehrer F., Ray T. Warehouse location under continuous economies of scale // Manage. Sci. 1966. Vol. 12, no. 3. P. 670–684.
129. Fersini E., Messina E., Archetti F. A p-Median approach for predicting drug response in tumour cells // BMC Bioinformatics. 2014. Vol. 15, no. 1. P. 1–19.
130. Fisher M. L. The Lagrangian Relaxation Method for solving integer programming problems // Manage. Sci. 1981. Vol. 27, no. 1. P. 1–18.
131. Fisher M. L. An applications oriented guide to Lagrangian Relaxation // Interfaces. 1985. Vol. 15, no. 2. P. 10–21.
132. Fisher M. L., Hochbaum D. S. Database Location in Computer Networks // J. ACM. 1980. Vol. 27, no. 4. P. 718–735.
133. Fisher M. L., Jaikumar R., Wassenhove L. N. V. A Multiplier Adjustment Method for the Generalized Assignment Problem // Manage. Sci. 1986. Vol. 32, no. 9. P. 1095–1103.
134. Flynn M. J. Very high-speed computing systems // Proc. IEEE. 1966. Vol. 54, no. 12. P. 1901–1909.
135. Flynn M. J. Some Computer Organizations and Their Effectiveness // IEEE Trans. Comput. 1972. Vol. 21, no. 9. P. 948–960.
136. Foundations of Location Analysis / Ed. by H. A. Eiselt, V. Marianov. International Series in Operations Research & Management Science. New York: Springer, 2011. 520 p.
137. Francis R., Lowe T., Rayco M., Tamir A. Aggregation error for location models: survey and analysis // Ann Oper Res. 2009. Vol. 167, no. 1. P. 171–208.
138. Frangioni A. About Lagrangian Methods in Integer Optimization // Ann. Oper. Res. 2005. Vol. 139, no. 1. P. 163–193.
139. Galvão R. D., Marianov V. Lagrangean Relaxation-Based Techniques for Solving Facility Location Problems // Foundations of Location Analysis, Ed. by H. A. Eiselt, V. Marianov. New York: Springer, 2011. P. 391–420.
140. García S., Labbé M., Marín A. Solving large p-median problems with a radius formulation //

INFORMS J. Comput. 2011. Vol. 23, no. 4. P. 546–556.

141. Garcia-López F., Melián-Batista B., Moreno-Pérez J. A., Moreno-Vega J. M. The Parallel Variable Neighborhood Search for the p-Median Problem // J. of Heuristics. 2002. Vol. 8, no. 3. P. 375–388.
142. Garcia-López F., Melián-Batista B., Moreno-Pérez J. A., Moreno-Vega J. M. Parallelization of the scatter search for the p-median problem // Parallel Comput. 2003. Vol. 29, no. 5. P. 575–589. Parallel computing in logistics.
143. Geoffrion A. M. Lagrangean relaxation for integer programming // Approaches to Integer Programming / Ed. by M. L. Balinski. Berlin: Springer, 1974. Vol. 2 of Mathematical Programming Studies. P. 82–114.
144. Goffin J. On the convergence rates of subgradient optimization methods // Math. Program. 1977. Vol. 13, no. 1. P. 329–347.
145. Goffin J. L. Decomposition and nondifferentiable optimization with the projective algorithm // Manage. Sci. 1992. Vol. 38, no. 2. P. 284–302.
146. Golden B. L., Skiscim C. C. Using simulated annealing to solve routing and location problems // Nav. Res. Log. Quart. 1986. Vol. 33, no. 2. P. 261–279.
147. Grama A., Gupta A., Kumar G. K. V. Introduction to Parallel Computing. 2nd edition. Boston: Addison-Wesley, 2003. 656 p.
148. Greenberg H. J. The One-Dimensional Generalized Lagrange Multiplier Problem // Oper. Res. 1977. Vol. 25, no. 2. P. 338–345.
149. Gropp W., Lusk E., Thakur R. Using MPI-2: Advanced Features of the Message Passing Interface. Scientific and Engineering Computation. Cambridge, Massachusets: MIT Press, 1999. 406 p.
150. Guignard M. A Lagrangean dual ascent algorithm for simple plant location problems // Eur. J. Oper. Res. 1988. Vol. 35, no. 2. P. 193–200.
151. Guignard M. Lagrangean relaxation // TOP. 2003. Vol. 11, no. 2. P. 151–228.
152. Guignard M., Opaswongkarn K. Lagrangean dual ascent algorithms for computing bounds in capacitated plant location problems // Eur. J. Oper. Res. 1990. Vol. 46, no. 1. P. 73–83.
153. Guignard M., Rosenwein M. B. An application-oriented guide for designing Lagrangean dual ascent algorithms // Eur. J. Oper. Res. 1989. Vol. 48, no. 2. P. 197–205.
154. Gustafson J. L. Reevaluating Amdahl's Law // Commun. ACM. 1988. Vol. 31, no. 5. P. 532–533.
155. Gustafson J. L. Amdahl's Law // Encyclopedia of Parallel Computing / Ed. by D. Padua. New York: Springer, 2011. P. 53–60.

156. Haimes Y. Y., Lasdon L. S., Wismer D. A. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization // IEEE Trans. Syst., Man, Cybern., Syst. 1971. Vol. SMC-1, no. 3. P. 296–297.
157. Hakimi S. L. Optimal location of switching centers and the absolute centers and medians of a graph // Oper. Res. 1964. Vol. 12, no. 3. P. 450–459.
158. Hakimi S. L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems // Oper. Res. 1965. Vol. 13, no. 3. P. 462–475.
159. Hale T. Trevor Hale's Location Science References. 2015. URL: http://gator.uhd.edu/~halet/ (дата обращения: 14.07.2015).
160. Hamacher H. W., Labbé M., Nickel S. Multicriteria network location problems with sum objectives // Networks. 1999. Vol. 33, no. 2. P. 79–92.
161. Hanafi S., Sterle C., Ushakov A., Vasilyev I. A parallel subgradient algorithm for Lagrangean dual function of the p-median problem // Studia Informatica Universalis. 2011. Vol. 9, no. 3. P. 105–124.
162. Handler G. Y., Mirchandani P. B. Multiobjective and other location problems on network // Location on Networks: Theory and Algorithms. Signal Processing, Optimization, and Control. Cambridge, MA: MIT Press, 1979. P. 165–195.
163. Hanjoul P., Peeters D. A comparison of two dual-based procedures for solving the p-median problem // Eur. J. Oper. Res. 1985. Vol. 20, no. 3. P. 387–396.
164. Hansen P., Brimberg J., Urosević D., Mladenović N. Solving large p-median clustering problems by primal-dual variable neighborhood search // Data Min. Knowl. Discov. 2009. Vol. 19, no. 3. P. 351–375.
165. Hansen P., Jaumard B. Cluster analysis and mathematical programming // Math. Program. 1997. Vol. 79, no. 1-3. P. 191–215.
166. Hansen P., Mladenović N. Variable neighborhood search for the p-median // Locat. Sci. 1997. Vol. 5, no. 4. P. 207–226.
167. Hansen P., Mladenović N. Variable neighborhood search: Principles and applications // Eur. J. Oper. Res. 2001. Vol. 130, no. 3. P. 449–467.
168. Hansen P., Mladenović N., Perez-Britos D. Variable neighbourhood decomposition search // J. Heuristics. 2001. Vol. 7, no. 4. P. 335–350.
169. Harkness J., ReVelle C. Facility location with increasing production costs // Eur. J. Oper. Res. 2003. Vol. 145, no. 1. P. 1–13.
170. Held M., Karp R. M. The Traveling-Salesman Problem and Minimum Spanning Trees // Oper. Res. 1970. Vol. 18, no. 9. P. 1138–1162.

171. Held M., Karp R. M. The Traveling-Salesman Problem and Minimum Spanning Trees: Part II // Math. Program. 1971. Vol. 1, no. 1. P. 6–25.

172. Held M., Wolfe P., Crowder H. Validation of subgradient optimization // Math. Program. 1974. Vol. 6, no. 1. P. 62–88.

173. Helmbold D. P., McDowell C. E. Modeling Speedup(N) Greater Than N // IEEE Trans. Parallel Distrib. Syst. 1990. Vol. 1, no. 2. P. 250–256.

174. Hiriart-Urruty J. B., Lemaréchal C. Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods. Grundlehren der mathematischen Wissenschaften 306. Berlin: Springer, 1993. 348 p.

175. Holmberg K. Solving the staircase cost facility location problem with decomposition and piecewise linearization // Eur. J. Oper. Res. 1994. Vol. 75, no. 1. P. 41–61.

176. Everett III H. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources // Oper. Res. 1963. Vol. 11, no. 3. P. 399–417.

177. Irawan C. A., Salhi S. Solving large p-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search // J. Glob. Optim. 2013. Vol. DOI: 10.1007/s10898-013-0080-z.

178. Irawan C. A., Salhi S., Scaparra M. P. An adaptive multiphase approach for large unconditional and conditional p-median problems // Eur. J. Oper. Res. 2014. Vol. 237, no. 2. P. 590–605.

179. Jain K., Vazirani V. V. Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-dual Schema and Lagrangian Relaxation // J. ACM. 2001. Vol. 48, no. 2. P. 274–296.

180. Järvinen P., Rajala J., Sinervo H. A Branch-and-Bound Algorithm for Seeking the p-Median // Oper. Res. 1972. Vol. 20, no. 1. P. 173–178.

181. Kalcsics J., Nickel S., Pozo M. A. et al. The multicriteria p-facility median location problem on networks // Eur. J. Oper. Res. 2014. Vol. 235, no. 3. P. 484–493.

182. Kariv O., Hakimi S. An algorithmic approach to network location problems. I: The p-centers // SIAM J. Appl. Math. 1979. Vol. 37, no. 3. P. 513–538.

183. Kelley J. H. The cutting plane method for solving convex programs // J. Soc. Industr. and Appl. Math. 1960. Vol. 8, no. 4. P. 703–712.

184. Kielmann T., Bal H. E., Verstoep K. Fast Measurement of LogP Parameters for Message Passing Platforms // Parallel and Distributed Processing / Ed. by J. Rolim. Berlin: Springer, 2000. Vol. 1800 of Lecture Notes in Computer Science. P. 1176–1183.

185. Kielmann T., Gorlatch S. Bandwidth-Latency Models (BSP, LogP) // Encyclopedia of Par-

allel Computing / Ed. by D. Padua. New York: Springer, 2011. P. 107–112.

186. Klastorin T. The p-median problem for cluster analysis: A comparative test using the mixture model approach // Manage. Sci. 1985. Vol. 31, no. 1. P. 84–95.
187. Kochetov Y., Alekseeva E., Levanova T., Loresh M. Large neighborhood local search for the p-median problem // Yugoslav Journal of Oper. Res. 2005. Vol. 15, no. 1. P. 53–63.
188. Kochetov Y., Ivanenko D. Computationally difficult instances for the uncapacitated facility location problem // Metaheuristics: progress as real problem solvers / Ed. by T. Ibaraki, K. Nonobe, M. Yagiura. New York: Springer, 2005. Vol. 32 of Operations Research/Computer Science Interfaces Series. P. 351–367.
189. Körkel M. Discrete facility location with nonlinear facility costs // RAIRO-Oper. Res. 1991. Vol. 25, no. 1. P. 31–43.
190. Kouvelis P., Yu G. Robust Discrete Optimization and Its Applications. New York: Springer, 1997. Vol. 14 of Nonconvex Optimization and Its Applications. 358 p.
191. Krarup J., Pruzan P. M. The simple plant location problem: Survey and synthesis // Eur. J. Oper. Res. 1983. Vol. 12, no. 1. P. 36–81.
192. Labbé M., Thisse J.-F., Wendell R. E. Sensitivity Analysis in Minisum Facility Location Problems // Oper. Res. 1991. Vol. 39, no. 6. P. 961–969.
193. Laumanns M., Thiele L., Zitzler E. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method // Eur. J. Oper. Res. 2006. Vol. 169, no. 3. P. 932–942.
194. Lemaréchal C. An extension of Davidon methods to non differentiable problems // Nondifferentiable Optimization / Ed. by M. L. Balinski, P. Wolfe. Berlin: Springer Berlin Heidelberg, 1975. Vol. 3 of Mathematical Programming Studies. P. 95–109.
195. Lemaréchal C. Nonsmooth optimization and descent methods: Research report rr-78-004. Laxenburg, Austria: IIASA, 1978.
196. Lemaréchal C. Lagrangian Relaxation // Computational Combinatorial Optimization / Ed. by M. Jünger, D. Naddef. Berlin: Springer, 2001. Vol. 2241 of Lecture Notes in Computer Science. P. 112–156.
197. Lemaréchal C. The omnipresence of Lagrange // Ann. Oper. Res. 2007. Vol. 153, no. 1. P. 9–27.
198. Lemaréchal C., Nemirovskii A., Nesterov Y. New variants of bundle methods // Math. Program. 1995. Vol. 69, no. 1-3. P. 111–147.
199. Li S., Svensson O. Approximating K-median via Pseudo-approximation // Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. STOC '13. New York:

ACM, 2013. P. 901–910.

200. Li X., Xiao N., Claramunt C., Lin H. Initialization strategies to enhancing the performance of genetic algorithms for the p-median problem // Comput. Ind. Eng. 2011. Vol. 61, no. 4. P. 1024–1034.
201. Lin J.-R., Nozick L. K., Turnquist M. A. Strategic design of distribution systems with economies of scale in transportation // Ann. Oper. Res. 2006. Vol. 144, no. 1. P. 161–180.
202. Location Science / Ed. by G. Laporte, S. Nickel, F. S. da Gama. Cham: Springer, 2015. 644 p.
203. Lorie J., Savage L. J. Three problems in capital rationing // J. of Bus. 1955. Vol. 28, no. 4. P. 229–239.
204. Louveaux F. V. Discrete stochastic location models // Ann. Oper. Res. 1986. Vol. 6, no. 2. P. 21–34.
205. Louveaux F. V. Stochastic location analysis // Locat. Sci. 1993. Vol. 1. P. 127–154.
206. Louveaux F. V., Peeters D. A Dual-Based Procedure for Stochastic Facility Location // Oper. Res. 1992. Vol. 40, no. 3. P. 564–573.
207. Lu D., Gzara F., Elhedhli S. Facility location with economies and diseconomies of scale: models and column generation heuristics // IIE Trans. 2014. Vol. 46, no. 6. P. 585–600.
208. Lübbecke M. E., Desrosiers J. Selected Topics in Column Generation // Oper. Res. 2005. Vol. 53, no. 6. P. 1007–1023.
209. Ma L., Lim G. GPU-Based Parallel Computational Algorithms for Solving p-Median Problem // Proceedings of the IIE Annual Conference and Expo, ID; 1110, IERC 2011. 2011.
210. Ma L., Lim G. J. GPU-based parallel vertex substitution algorithm for the p-median problem // Comput. Ind. Eng. 2013. Vol. 64, no. 1. P. 381–388.
211. Mancini E. P., Marcarelli S., Ritrovato P. et al. A Grid-Aware Branch, Cut and Price Implementation // Recent Advances in Parallel Virtual Machine and Message Passing Interface / Ed. by B. D. Martino, D. Kranzlmüller, J. Dongarra. Springer, 2005. Vol. 3666 of Lecture Notes in Computer Science. P. 38–47.
212. Mancini E. P., Marcarelli S., Vasilyev I., Villano U. A grid-aware MIP solver: Implementation and case studies // Futur. Gener. Comp. Syst. 2008. Vol. 24, no. 2. P. 133 – 141.
213. Maranzana F. E. On the Location of Supply Points to Minimize Transport Costs // Oper. Res. Quart. 1964. Vol. 15, no. 3. P. 261–270.
214. Mavrotas G. Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems // Appl. Math. Comput. 2009. Vol. 213, no. 2. P. 455–465.
215. McCool M., Reinders J., Robison A. Structured Parallel Programming: Patterns for Efficient

Computation. Waltham: Morgan Kaufmann, 2012. 432 p.

216. Megiddo N., Supowit K. J. On the Complexity of Some Common Geometric Location Problems // SIAM J. Comput. 1984. Vol. 13, no. 1. P. 182–196.
217. Mifflin R. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization // Nondifferential and Variational Techniques in Optimization / Ed. by D. C. Sorensen, R. J.-B. Wets. Berlin: Springer, 1982. Vol. 17 of Mathematical Programming Studies. P. 77–90.
218. Mirchandani P., Jagannathan R. Discrete facility location with nonlinear diseconomies in fixed costs // Ann. Oper. Res. 1989. Vol. 18, no. 1. P. 213–224.
219. Mirchandani P. B., Odoni A. R. Locations of Medians on Stochastic Networks // Transp. Sci. 1979. Vol. 13, no. 2. P. 85–97.
220. Mirchandani P. B., Oudjit A., Wong R. T. 'Multidimensional' extensions and a nested dual approach for the m-median problem // Eur. J. Oper. Res. 1985. Vol. 21, no. 1. P. 121–137.
221. Mixed Integer Nonlinear Programming / Ed. by J. Lee, S. Leyffer. New York: Springer, 2012. Vol. 154 of The IMA Volumes in Mathematics and its Applications. 692 p.
222. Mladenović N., Brimberg J., Hansen P., Moreno-Pérez J. The p-median problem: A survey of metaheuristic approaches // Eur. J. Oper. Res. 2007. Vol. 179, no. 3. P. 927–939.
223. Motzkin T., Schoenberg I. The relaxation method for linear inequalities // Canad. J. Math. 1954. Vol. 6, no. 1. P. 393–404.
224. Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys / Ed. by M. Ehrgott, X. Gandibleux. International Series in Operations Research & Management Science. Kluwer Academic Publishers, 2003. 520 p.
225. Mulvey J. M., Crowder H. P. Cluster Analysis: An Application of Lagrangian Relaxation // Manage. Sci. 1979. Vol. 25, no. 4. P. 329–340.
226. Murota K. Discrete Convex Analysis. Discrete Mathematics and Applications. Philadelphia: SIAM, 2003. 389 p.
227. Murray A. T., Church R. L. Applying simulated annealing to location-planning models // J. Heuristics. 1996. Vol. 2, no. 1. P. 31–53.
228. Narula S. C., Ogbu U. I., Samuelsson H. M. An Algorithm for the p-Median Problem // Oper. Res. 1977. Vol. 25, no. 4. P. 709–713.
229. Nemhauser G. N., Wolsey L. A. Integer and Combinatorial Optimization. Wiley-Interscience series in discrete mathematics and optimization. New York: Wiley-Interscience, 1999. 763 p.
230. Nickel S., da Gama F. S., Ziegler H.-P. A multi-stage stochastic supply network design problem with financial decisions and risk management // Omega. 2012. Vol. 40, no. 5.

P. 511–524.

231. Nikoofal M. E., Sadjadi S. J. A robust optimization model for p-median problem with uncertain edge lengths // Int. J. Adv. Manuf. Technol. 2010. Vol. 50, no. 1-4. P. 391–397.
232. Nowak I. Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming. Basel, Switzerland: Birkhäuser Verlag, 2005. 213 p.
233. Oettli W. An iterative method, having linear rate of convergence, for solving a pair of dual linear programs // Math. Program. 1972. Vol. 3, no. 1. P. 302–311.
234. Oliveira W. L., Sagastizábal C. Bundle methods in the XXIst century: A bird's-eye view // Pesquisa Operacional. 2014. Vol. 34, no. 2. P. 647–670.
235. Owen S. H., Daskin M. S. Strategic facility location: A review // Eur. J. Oper. Res. 1998. Vol. 111, no. 3. P. 423–447.
236. Owens J. D., Houston M., Luebke D. et al. GPU Computing // Proc. IEEE. 2008. Vol. 96, no. 5. P. 879–899.
237. Patterson D. A., Hennessy J. L. Computer Organization and Design. The Morgan Kaufmann Series in Computer Architecture and Design. 5th edition. Waltham: Morgan Kaufmann, 2013. 800 p.
238. Poljak B. T. Subgradient Methods: A Survey of Soviet Research // Nonsmooth optimization, Proceedings of IIASA Workshop / Ed. by C. Lemarechal, R. Mifflin. Vol. 3 of IIASA Proceedings Series. Pergamon Press, 1977. P. 5–31.
239. Posypkin M., Semenov A., Zaikin O. Using BOINC Desktop GRID to solve large scale SAT problems // Computer Science. 2012. Vol. 13, no. 1. P. 25–34.
240. Ralphs T. K., Ladányi L., Saltzman M. J. Parallel branch, cut, and price for large-scale discrete optimization // Math. Program. 2003. Vol. 98, no. 1-3. P. 253–280.
241. Redondo J. L., Marín A., Ortigosa P. M. A parallelized Lagrangean relaxation approach for the discrete ordered median problem // Ann. Oper. Res. 2014. P. 1–20.
242. Reese J. Solution Methods for the p-Median Problem: An Annotated Bibliography // Networks. 2006. Vol. 28, no. 3. P. 125–142.
243. Ren Z., Jiang H., Xuan J., Luo Z. An Accelerated-Limit-Crossing-Based Multilevel Algorithm for the p-Median Problem // IEEE Trans. Syst. Man Cybern. B Cybern. 2012. Vol. 42, no. 4. P. 1187–1202.
244. Resende M. G. C., Werneck R. F. A hybrid heuristic for the p-median problem // J. Heuristics. 2004. Vol. 10, no. 1. P. 59–88.
245. ReVelle C. S., Eiselt H. A. Location analysis: A synthesis and survey // Eur. J. Oper. Res. 2005. Vol. 165, no. 1. P. 1–19.

246. ReVelle C. S., Eiselt H. A., Daskin M. S. A bibliography for some fundamental problem categories in discrete location science // Eur. J. Oper. Res. 2008. Vol. 184, no. 3. P. 817–848.

247. ReVelle C. S., Swain R. W. Central facilities location // Geographical Analysis. 1970. Vol. 2, no. 1. P. 30–42.

248. Rolland E., Schilling D. A., Current J. R. An efficient tabu search procedure for the p-Median Problem // Eur. J. Oper. Res. 1997. Vol. 96, no. 2. P. 329–342.

249. Roosta S. H. Parallel Processing and Parallel Algorithms: Theory and Computation. New York: Springer, 2000. 566 p.

250. Rosenhead J., Elton M., Gupta S. K. Robustness and Optimality as Criteria for Strategic Decisions // Oper. Res. Quart. 1972. Vol. 23, no. 4. P. 413–431.

251. Rosing K. E., ReVelle C. S., Rosing-Vogelaar H. The p-median and its linear programming relaxation: an approach to large problems // J. Oper. Res. Soc. 1979. Vol. 30, no. 9. P. 815–823.

252. Savage J. E. Parallel Computation // Models of computation: Exploring the power of computing. Boston: Addison-Wesley, 1998. P. 281–325.

253. Schulz C., Hasle G., Brodtkorb A. R., Hagen T. R. GPU computing in discrete optimization. Part II: Survey focused on routing problems // EURO J. Transp. Logist. 2013. Vol. 2, no. 1-2. P. 159–186.

254. Senne E., Lorena L., Pereira M. A branch-and-price approach to p-median location problems // Comput. Oper. Res. 2005. Vol. 32, no. 6. P. 1655–1664.

255. Serra D., Marianov V. The p-median problem in a changing network: the case of Barcelona // Locat. Sci. 1998. Vol. 6, no. 1-4. P. 383–394.

256. Serra D., Ratick S., ReVelle C. The maximum capture problem with uncertainty // Environ. Plann. B. 1996. Vol. 23, no. 1. P. 49–59.

257. Shapiro J. F. A Survey of Lagrangean Techniques for Discrete Optimization // Discrete Optimization II. Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver / Ed. by P. L. Hammer, E. L. Johnson, B. H. Korte. Elsevier, 1979. Vol. 5 of Annals of Discrete Mathematics. P. 113–138.

258. Sharkey T. C., Geunes J., Romeijn H. E., Shen Z.-J. M. Exact algorithms for integrated facility location and production planning problems // Nav. Res. Logist. 2011. Vol. 58, no. 5. P. 419–436.

259. Shinano Y., Achterberg T., Berthold T. et al. ParaSCIP: A Parallel Extension of SCIP //

Competence in High Performance Computing 2010 / Ed. by C. Bischof, H.-G. Hegering, W. E. Nagel, G. Wittum. Berlin: Springer, 2012. P. 135–148.

260. Shor N. Z. Minimization Methods for Non-Differentiable Functions. Berlin: Springer, 1985. Vol. 3 of Springer Series in Computational Mathematics. 162 p.
261. Shor N. Z. Nondifferentiable Optimization and Polynomial Problems. Nonconvex Optimization and Its Applications, Vol. 24. Dordrecht: Kluwer Academic Publishers, 1998. 396 p.
262. Snyder L. V. Facility location under uncertainty: a review // IIE Trans. 2006. Vol. 38, no. 7. P. 547–564.
263. Soland R. M. Optimal Facility Location with Concave Costs // Oper. Res. 1974. Vol. 22, no. 2. P. 373–382.
264. Srinivasan V., Thompson G. L. Algorithms for minimizing total cost, bottleneck time and bottleneck shipment in transportation problems // Nav. Res. Log. Quart. 1976. Vol. 23, no. 4. P. 567–595.
265. Stallings W. Computer Organization and Architecture. William Stallings Books on Computer and Data Communications. 8th edition. New Jersey: Prentice Hall, 2012. 792 p.
266. Sugar C. A., James G. M. Finding the Number of Clusters in a Dataset: An Information-Theoretic Approach // J. Am. Stat. Assoc. 2003. Vol. 98, no. 463. P. 750–763.
267. Teitz M., Bart P. Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph // Oper. Res. 1968. Vol. 16, no. 5. P. 955–961.
268. Ulungu E. L., Teghem J. Multi-objective combinatorial optimization problems: A survey // J. Multi-Crit. Decis. Anal. 1994. Vol. 3, no. 2. P. 83–104.
269. v. d. Broek J., Schütz P., Stougie L., Tomasgard A. Location of slaughterhouses under economies of scale // Eur. J. Oper. Res. 2006. Vol. 175, no. 2. P. 740–750.
270. Valiant L. G. A Bridging Model for Parallel Computation // Commun. ACM. 1990. Vol. 33, no. 8. P. 103–111.
271. Weaver J. R., Church R. L. Computational Procedures for Location Problems on Stochastic Networks // Transp. Sci. 1983. Vol. 17, no. 2. P. 168–180.
272. White D. J. Epsilon efficiency // J. Optim. Theory Appl. 1986. Vol. 49, no. 2. P. 319–337.
273. Wilt N. CUDA Handbook: A Comprehensive Guide to GPU Programming. Crawfordsville: Addison-Wesley Professional, 2013. 528 p.
274. Wolfe P. A method of conjugate subgradients for minimizing nondifferentiable functions // Nondifferentiable Optimization / Ed. by M. L. Balinski, P. Wolfe. Berlin: Springer, 1975. Vol. 3 of Mathematical Programming Studies. P. 145–173.
275. Wolsey L. A. Integer Programming. Wiley-Interscience series in discrete mathematics and

optimization. New York: Wiley-Interscience, 1998. 288 p.

276. Wu L. Y., Zhang X. S., Zhang J. L. Capacitated facility location problem with general setup cost // Comput. Oper. Res. 2006. Vol. 33, no. 5. P. 1226–1241.
277. Xu Y., Ralphs T. K., Ladányi L., Saltzman M. J. Computational Experience with a Software Framework for Parallel Integer Programming // INFORMS J. Comput. 2009. Vol. 21, no. 3. P. 383–397.
278. Zhang T., Ramakrishnan R., Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases // Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. SIGMOD '96. New York: ACM, 1996. P. 103–114.
279. Zhang T., Ramakrishnan R., Livny M. BIRCH: a new data clustering algorithm and its applications // Data Min. Knowl. Discov. 1997. Vol. 1, no. 2. P. 141–182.
280. Zhao W., Posner M. E. A large class of facets for the K-median polytope // Math. Program. 2011. Vol. 128, no. 1-2. P. 171–203.