

УЧРЕЖДЕНИЕ РОССИЙСКОЙ АКАДЕМИИ НАУК
ИНСТИТУТ ДИНАМИКИ СИСТЕМ И ТЕОРИИ УПРАВЛЕНИЯ
СИБИРСКОГО ОТДЕЛЕНИЯ РАН

На правах рукописи

Мальшев Антон Валентинович

МЕТОДЫ РЕШЕНИЯ КВАДРАТИЧНО-ЛИНЕЙНЫХ ЗАДАЧ
ДВУХУРОВНЕВОЙ ОПТИМИЗАЦИИ

Специальность 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:
доктор физико-математических
наук, профессор А.С. Стрекаловский

Иркутск-2011

Оглавление

Введение	3
1 Глобальный поиск оптимистических решений в двухуровневых задачах	21
1.1 Постановка задачи и ее редукция	22
1.2 Локальный поиск	29
1.3 Тестирование процедур локального поиска	37
1.4 Алгоритм глобального поиска	44
1.5 Тестирование алгоритма глобального поиска	50
1.6 Заключительные замечания	59
2 Теоретические основы поиска гарантированных решений	60
2.1 Постановка задачи и ее взаимосвязь с задачей поиска оптимистического решения специальной двухуровневой задачи	61
2.2 Свойства задачи нижнего уровня	68
2.3 Редукция к задачам d.c. оптимизации	74
2.4 Процедуры локального поиска	77
2.5 Алгоритм глобального поиска	82
2.6 Заключительные замечания	86
3 Численный поиск гарантированных решений	87
3.1 Генерация тестовых задач	87
3.2 Тестирование локального поиска	102
3.3 Численный поиск гарантированных решений в сгенерированных задачах	110
3.4 Заключительные замечания	114
Заключение	115
Литература	116

Введение

На протяжении последних десятилетий внимание специалистов все больше привлекают иерархические задачи с конфликтами, возникающие в различных приложениях (в экономике, экологии, энергетике, технике и других областях). Эти задачи характеризуются неравноправным положением участников (игроков) и называются играми с иерархической структурой [7, 8, 11, 60, 74].

Игры с иерархической структурой, как правило, возникают при моделировании сложноорганизованных систем управления (социальных, экономических, эколого-экономических и др.) [7, 8, 11, 60, 74, 118]. Неизбежность возникновения иерархической структуры в “достаточно сложных” системах подчеркивал акад. Н.Н. Моисеев в своих работах по теории оптимальных систем [33]. Иерархия в таких системах возникает по причине высокой трудоемкости своевременного сбора и переработки большого объема информации о контролируемых процессах единым управляющим центром, что приводит к принятию решений по неполной или устаревшей информации (т.е. в условиях неопределенности). Обработка информации и принятие решений отдельными элементами системы (подсистемами) позволяет обычно уменьшить влияние такого рода неопределенностей. Однако введение частичной децентрализации управления приводит к появлению другого вида неопределенности, связанной с самостоятельными действиями подсистем, определяемыми их интересами [11].

Первые исследования многоуровневых иерархических систем показали их непреодолимую на данном этапе сложность [8, 11]. Поэтому естественным был переход к рассмотрению наиболее простых объектов с иерархией — игр двух лиц с фиксированным порядком ходов и передачей информации о первом ходе. К таким играм сводится исследование многих двухуровневых иерархических систем управления [7, 8, 11, 60, 74].

Впервые иерархические игры двух лиц были рассмотрены Генрихом фон Штакель-

бергом в монографии [118] при исследовании практических моделей рыночной экономики. В играх такого рода имеются два игрока, осуществляющие выбор своих стратегий x и y из множеств X и Y . При этом считается, что интересы игроков полностью описываются их желанием минимизировать соответствующие критерии эффективности $F(x, y)$ и $G(x, y)$, которые, вообще говоря, могут быть различными. Игрок 1 (называемый игроком верхнего уровня или лидером) обладает определенным приоритетом, выражающимся в праве сделать свой ход первым. Ход игрока 1 состоит в выборе стратегии $x \in X$, которая может быть, вообще говоря, функцией от информации об ожидающемся поведении игрока 2, и передаче сообщения о выбранной стратегии игроку 2 (игроку нижнего уровня). Поскольку после сообщения игроком 1 своей стратегии выигрыш игрока 2 зависит только от его выбора, естественно считать, что поведение игрока 2 определяется стремлением минимизировать функцию $G(x, \cdot)$. Однако, это не устраняет полностью неопределенность для игрока 1, даже если он знает функцию $G(x, \cdot)$, поскольку ее минимум может быть неединственным (если эта функция не является, скажем, строго выпуклой) [11, 74].

Описанная выше игра (с позиции игрока верхнего уровня) может быть сформулирована в виде следующей экстремальной задачи:

$$\begin{cases} F(x, y) \downarrow \underset{x}{\text{min}}, & x \in X, \\ y \in \underset{z}{\text{Argmin}}\{G(x, z) \mid z \in Y\}, \end{cases} \quad (\mathcal{S})$$

которая называется *задачей или игрой Штакельберга* [74].

Так называемые *двухуровневые задачи оптимизации* [60, 74] обобщают задачу Штакельберга в том смысле, что множество допустимых стратегий игрока нижнего уровня (игрока 2) считается зависящим от стратегии игрока верхнего уровня (игрока 1) [74], и формулируются следующим образом:

$$\begin{cases} F(x, y) \downarrow \underset{x}{\text{min}}, & x \in X, \\ y \in \underset{z}{\text{Argmin}}\{G(x, z) \mid z \in Y(x)\}. \end{cases} \quad (\mathcal{BP})$$

Подчеркнем, что кавычки в постановках задач (\mathcal{S}) и (\mathcal{BP}) отражают отмеченную выше неопределенность выбора конкретной стратегии y игроком нижнего уровня из множества своих оптимальных стратегий. Для снятия этой неопределенности известны два подхода — оптимистический (кооперативный) и гарантированный (пессимистический),

которые приводят к соответствующим определениям решений в задачах двухуровневой оптимизации [7, 8, 11, 60, 74].

В первом случае считается, что интересы игрока верхнего уровня могут быть согласованы с действиями игрока нижнего уровня, и, следовательно, игрок 2 из множества своих оптимальных стратегий выбирает стратегию наиболее благоприятствующую игроку 1. В этом случае задача (\mathcal{BP}) принимает следующий вид:

$$\begin{cases} F(x, y) \downarrow \min_{x, y}, & x \in X, \\ y \in \underset{z}{\text{Argmin}}\{G(x, z) \mid z \in Y(z)\}. \end{cases} \quad (\mathcal{BP}_o)$$

Решение задачи (\mathcal{BP}_o) называется *оптимистическим* решением задачи (\mathcal{BP}) [60, 74]. Отметим, что в случае оптимистического решения зачастую рассматривают более общую задачу, где ограничение на выбор допустимой стратегии игроком верхнего уровня $x \in X$ заменяется более общим ограничением на совокупность стратегий игроков $(x, y) \in D$ [74].

Во втором случае предполагается, что игрок 2 предпочитает действовать независимо, и игрок 1 вынужден это принять во внимание, исходя из обобщенного принципа гарантированного результата [8, 11, 74]. Здесь считается, что игрок нижнего уровня может действовать наихудшим по отношению к игроку верхнего уровня образом, вследствие чего возникает задача:

$$\begin{cases} W(x) \triangleq \sup_y \{F(x, y) \mid y \in Y_*(x)\} \downarrow \min_x, & x \in X, \\ Y_*(x) \triangleq \underset{z}{\text{Argmin}}\{G(x, z) \mid z \in Y(x)\}, \end{cases} \quad (\mathcal{BP}_g)$$

где $W(x)$ — оценка эффективности стратегий игрока верхнего уровня [8].

Решение задачи (\mathcal{BP}_g) называется *гарантированным* или *пессимистическим* решением задачи (\mathcal{BP}) [7, 8, 11, 60, 74]. Также будем далее называть задачу (\mathcal{BP}_o) двухуровневой задачей в оптимистической (кооперативной) постановке, а задачу (\mathcal{BP}_g) — двухуровневой задачей в гарантированной (пессимистической, некооперативной) постановке.

Сразу же отметим, что двухуровневая задача в гарантированной постановке является более сложной по сравнению с задачей в оптимистической постановке [79]. Здесь целевая функция верхнего уровня (оценка эффективности лидера) включает в себя операцию взятия точной верхней грани. Поэтому целевая функция лидера оказывается, вообще говоря, негладкой даже в том случае, когда функции $F(\cdot)$, $G(\cdot)$ — гладкие.

Кроме того, как будет показано ниже, даже двухуровневая задача в оптимистической постановке обладает структурной невыпуклостью, причем даже в случае выпуклости функций $F(\cdot)$, $G(\cdot)$ и множеств допустимых стратегий игроков.

Близким к двухуровневым задачам классом игровых задач являются задачи равновесного программирования [1,98,113,117], отличающиеся тем, что игроки в них являются равноправными. Кроме того, к двухуровневым задачам тесно примыкают так называемые задачи с равновесными ограничениями, задачи нижнего уровня в которых могут иметь вид вариационного неравенства, некоторой задачи равновесного программирования или дополненности [19, 101, 104, 112, 113].

К настоящему времени опубликовано огромное количество работ, в которых рассматриваются задачи двухуровневой оптимизации (см., например, обзоры [67,72]). Приведем ссылки лишь на работы, наиболее близкие к тематике диссертации. Прежде всего необходимо отметить две монографии Дж.Ф. Барда [60] и С. Демпе [74], ставшие уже классическими в этой области. В них рассматриваются вопросы существования и устойчивости решений различных классов двухуровневых задач, приводятся различные необходимые и достаточные условия оптимальности, предлагаются подходы к решению таких задач и описаны некоторые их практические приложения.

Большое количество публикаций посвящено теоретическому исследованию непрерывных двухуровневых задач. В них рассматриваются вопросы существования решения и алгоритмической трудности двухуровневой задачи [52, 61], устойчивости решения [95,99], получения условий оптимальности [73, 76–79, 95, 124, 131].

Известные методы отыскания оптимистических решений в непрерывных двухуровневых задачах можно разделить на следующие группы.

1) Методы, в которых двухуровневая задача сводится к невыпуклой (одноуровневой) задаче оптимизации [13, 53, 55, 81, 82, 90, 92, 100, 109, 127, 128]. Обычно это сведение осуществляется при помощи замены задачи нижнего уровня ее необходимыми и достаточными условиями типа Каруша-Куна-Таккера [74]. К полученной в результате такого сведения невыпуклой задаче затем применяется один из алгоритмов глобальной оптимизации [43, 93, 132].

2) Методы штрафов [70,91,102,115,130] являются родственными методам из пункта 1. Здесь, как и в методах из п. 1, осуществляется редукция задачи к некоторой одноуровне-

вой. Решение последней осуществляется с применением метода штрафов [6]. Чаще всего в качестве штрафной функции выступает сумма целевой функции верхнего уровня и невязки двойственности для задачи нижнего уровня со штрафным множителем [74].

3) Методы, в которых двухуровневая задача сводится к одноуровневой задаче оптимизации, часть переменных которой являются целочисленными [58, 66].

4) Методы решения двухуровневых задач с линейными ограничениями на обоих уровнях перебором вершин многогранника, определяющего допустимую область задачи [65].

5) Методы допустимых направлений для задачи верхнего уровня с информацией о градиенте, получаемой из задачи нижнего уровня [84, 97, 116, 125]. Такие методы обеспечивают лишь получение стационарной точки в задаче, так что их можно считать алгоритмами локального поиска.

6) Методы доверительной области [54, 68, 71, 75]. Здесь рассматриваемая двухуровневая задача заменяется более простой модельной задачей и производится ее решение в некоторой заданной (доверительной) области. Затем полученное решение оценивается с точки зрения исходной задачи, на основании чего принимается решение об изменении текущей доверительной области.

7) Методы, основанные на редукции двухуровневой задачи к задаче векторной оптимизации [17, 86, 123].

8) Методы “параметрического программирования” [83], в которых допустимое множество задачи верхнего уровня представляется в явном виде, путем нахождения явного вида множеств оптимальных стратегий игрока нижнего уровня для каждой из допустимых стратегий игрока верхнего уровня.

9) Метод ветвей и границ [59, 94, 108, 115, 122], генетические алгоритмы [133], поиск с запретами [114] и другие алгоритмы глобальной оптимизации, пришедшие из дискретной оптимизации, применяемые к двухуровневым задачам напрямую (в отличие от методов из п.п. 1–3).

С точки зрения численного поиска оптимистического решения сравнительно эффективными себя показывают, как правило, только специальные численные методы для линейно-линейных двухуровневых задач, в которых критерии эффективности игроков и функции, задающие ограничения, являются линейными функциями. Так, в [58, 115]

приведены результаты решения таких задач с суммарным числом переменных до 200.

При численном поиске оптимистических решений в нелинейных двухуровневых задачах чаще всего авторы ограничиваются рассмотрением иллюстративных примеров размерности до 10 (см. [66, 83, 84, 90, 100, 108, 109, 114, 116, 122, 127, 128, 133]) и только в [59, 68, 81, 82, 88] представлены результаты решения нелинейных двухуровневых задач размерности до 50, а в [94] — результаты решения таких задач размерности до 100 с разреженными матрицами. В работе [92] представлены результаты численного решения нелинейных двухуровневых задач размерности до 220, в которых целью игрока нижнего уровня является только лишь отыскание точки Каруша-Куна-Таккера в невыпуклой квадратичной задаче оптимизации. Входящая в решение такой двухуровневой задачи стратегия игрока нижнего уровня, вообще говоря, не является решением задачи нижнего уровня в силу невыпуклости этой задачи. Следовательно, отыскиваемое в [92] решение двухуровневой задачи не будет даже оптимистическим решением в смысле данного выше определения.

Что касается подходов к поиску гарантированного решения в двухуровневых задачах, этот вопрос рассматривался группой исследователей под руководством Ю.Б. Гермейера [8, 11, 34, 35]. В работах [34, 35] было получено сведение неантагонистической игры двух лиц с передачей информации (задачи Штакельберга) в гарантированной постановке к семейству вспомогательных задач Штакельберга в оптимистической постановке. К сожалению, в 90-е годы XX в. исследование таких задач в этой группе было приостановлено по объективным причинам. В более поздней работе [103] результаты из [34] были несколько обобщены. А именно, были ослаблены предположения на функции, входящие в постановку редуцируемой двухуровневой задачи.

Единственным известным нам (и авторам публикации [121]) алгоритмом численного решения непрерывной задачи двухуровневой оптимизации в гарантированной постановке является алгоритм поиска гарантированного решения нелинейной задачи Штакельберга, предложенный и протестированный на задачах размерности до 4 в [121]. Этот алгоритм основан на сведении рассматриваемой двухуровневой задачи к так называемой задаче полубесконечного программирования.

Также в литературе имеются публикации, посвященные решению отдельных классов дискретных двухуровневых задач, часть переменных в которых может принимать лишь

значения из конечного набора [3, 9, 10, 15, 51, 80, 88, 96, 126]. Некоторые из таких задач могут быть сведены к непрерывным двухуровневым задачам [88].

Объектом исследования диссертационной работы являются так называемые квадратично-линейные задачи двухуровневой оптимизации, в которых критерий эффективности игрока верхнего уровня — квадратичная функция, критерий эффективности игрока нижнего уровня — линейная функция, и множества допустимых стратегий игроков описываются линейными неравенствами.

Основная цель диссертационного исследования состоит в разработке методов поиска оптимистических и гарантированных решений в (непрерывных) квадратично-линейных задачах двухуровневой оптимизации, эффективных с вычислительной точки зрения.

Актуальность темы диссертационной работы обусловлена прежде всего широким полем практических приложений двухуровневых задач, а также сложностью исследуемых двухуровневых задач (которые могут быть сведены к невыпуклым задачам оптимизации, трудным с точки зрения поиска в них глобального решения). Приведем примеры практических двухуровневых задач.

1) Определение размеров квот и дотаций производителям сельхозпродукции [69, 111].

Игроком верхнего уровня в данной задаче является некий орган управления, игроком нижнего уровня — агрегированный производитель сельскохозяйственной продукции. Задача игрока верхнего уровня состоит в оптимальном выборе уровня квот и дотаций с учетом прогнозируемого рационального ответа игрока нижнего уровня на этот выбор.

Пусть m — количество видов квот и дотаций, n — количество видов продукции, тогда стратегия игрока верхнего уровня определяется вектором $x = (x_1, \dots, x_m)$, где x_i — размер финансирования квоты или дотации вида $i = 1, \dots, m$, в свою очередь стратегия игрока нижнего уровня определяется вектором $y = (y_1, \dots, y_n)$, где y_j — объем производства продукции вида $j = 1, \dots, n$. Известны ограничения x_i^+ на максимальный размер финансирования квоты или дотации вида i , а также общие ограничения $x_{m_1 m_2}^+$ на максимальный суммарный размер финансирования квот или дотаций видов m_1, \dots, m_2 . Критерием эффективности игрока верхнего уровня является свертка $m + 1$ критериев с

весовыми коэффициентами $w_i \geq 0$, $i = 1, \dots, m+1$. При этом каждый из первых m критериев отражает стремление к достижению рекомендуемого уровня \hat{x}_i квоты или дотации вида i (например, по экологическим соображениям может быть определен желаемый размер посевных площадей), а максимизация $(m+1)$ -го критерия $\langle b, x \rangle + \langle c, y \rangle$ означает стремление увеличить трудовую занятость населения, где c_j — размер заработной платы, соответствующий количеству рабочих мест, требуемых для производства единицы продукции вида j , b_i — размер заработной платы, соответствует количеству рабочих мест, дополнительно возникающих в случае увеличения дотации вида i (такие рабочие места возникают, например, при дотациях на постройку новых ирригационных систем, которые позволяют увеличить посевную площадь; в том случае, когда дополнительных рабочих мест не возникает, $b_i = 0$).

Целью игрока нижнего уровня является максимизация прибыли $\langle d, y \rangle$, где d_j — стоимость единицы вида продукции $j = 1, \dots, n$, с учетом балансовых ограничений $By \leq a + Ax$, где B_{kj} — количество ресурса вида $k = 1, \dots, q$, требуемого для производства единицы продукции вида j , a_k — запас ресурса вида k у производителя, A_{ki} — количество ресурса вида k , соответствующее единице финансирования квоты или дотации вида $i = 1, \dots, m$. Кроме того, на нижнем уровне учитываются ограничения y_j^+ на максимальный спрос на продукцию вида j и ограничения $y_{n_1 n_2}^+$ на максимальный общий спрос на продукцию видов n_1, \dots, n_2 .

Таким образом, возникает следующая двухуровневая задача.

$$\left. \begin{aligned} & \sum_{i=1}^m [w_i(x_i - \hat{x}_i)^2] - w_{m+1}(\langle b, x \rangle + \langle c, y \rangle) \downarrow \text{“min”}_x, \\ & 0 \leq x_i \leq x_i^+, \quad i = 1, \dots, m, \quad \sum_{i=m_1}^{m_2} x_i \leq x_{m_1 m_2}^+, \quad (m_1, m_2) \in \mathcal{M}, \\ & y \in \underset{y}{\text{Argmin}} \{ -\langle d, y \rangle \mid By \leq a + Ax, \\ & 0 \leq y_j \leq y_j^+, \quad j = 1, \dots, n, \quad \sum_{j=n_1}^{n_2} y_j \leq y_{n_1 n_2}^+, \quad (n_1, n_2) \in \mathcal{N} \}, \end{aligned} \right\}$$

где $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $\mathcal{M} \triangleq \{(m_1, m_2) \mid 0 \leq m_1 < m_2 \leq m, m_1, m_2 \text{ — целые}\}$, $\mathcal{N} \triangleq \{(n_1, n_2) \mid 0 \leq n_1 < n_2 \leq n, n_1, n_2 \text{ — целые}\}$.

Отметим, что эта задача принадлежит классу квадратично-линейных двухуровневых задач, которые исследуются в диссертационной работе. #

2) Задача оптимального выбора тарифов телекоммуникационным оператором [110].

Игроком верхнего уровня здесь является одна из нескольких конкурирующих телекоммуникационных компаний, а игроком нижнего уровня — клиент, которому необходимо обеспечить связь между несколькими узлами.

Предположим, что определен граф $G = (V, U, A)$, где V — множество вершин графа, соответствующих узлам телекоммуникационной сети, U — множество ребер, соответствующих доступным каналам связи, а отношение инцидентности определяется матрицей A . Пусть $|V| = m$, $|U| = n$. Множество U разобьем на два непересекающихся подмножества: $U = U_1 \cup U_2$, где ребра, входящие в U_1 , соответствуют каналам связи, которые обслуживает игрок верхнего уровня, а ребра, входящие в U_2 , соответствуют каналам, принадлежащим конкурирующим компаниям. Пусть $|U_1| = n_1$, $|U_2| = n_2$.

Каждому ребру графа G , входящему в U_1 , сопоставим два числа: c_i^1 — некоторая фиксированная цена передачи данных по соответствующей линии, и x_i — тариф, определяемый игроком верхнего уровня. Стоимость передачи данных по каналу связи, соответствующему произвольному ребру из U_1 , равна $x_i + c_i^1, i = 1, 2, \dots, n_1$. Каждому ребру из U_2 сопоставлена некоторая известная стоимость передачи данных по соответствующей линии $c_i^2, i = 1, 2, \dots, n_2$. Таким образом, $x = (x_1, x_2, \dots, x_{n_1})$ — вектор стратегии игрока верхнего уровня. Определены допустимые границы изменения тарифов: $x \in X = \prod_{i=1}^{n_1} [x_i, \bar{x}_i] \triangleq \{x \in \mathbb{R}^{n_1} \mid x_i \in [x_i, \bar{x}_i], i = 1, 2, \dots, n_1\}$.

Стратегия игрока нижнего уровня определяется вектором $y = (y^k) = ((y_1^k, y_2^k)), k = 1, 2, \dots, K$, каждый y^k соответствует объему трафика, который необходимо передать из узла s^k в узел r^k . При этом $y_1^k = (y_{1_1}^k, y_{1_2}^k, \dots, y_{1_{n_1}}^k)$ — поток трафика по каналам связи, соответствующим ребрам из U_1 , а $y_2^k = (y_{2_1}^k, y_{2_2}^k, \dots, y_{2_{n_2}}^k)$ — поток трафика по каналам, соответствующим ребрам из U_2 . Накладываются ограничения на пропускную способность каждой линии:

$$0 \leq y_j^k \leq \bar{y}_j^k, \quad j = 1, 2, \dots, n,$$

и ограничение сохранения потока:

$$Ay^k = \delta^k d^k,$$

где δ^k — запрашиваемый объем трафика, $d^k \in \mathbb{R}^m$ — вектор, у которого только компо-

ненты с индексами s^k и r^k отличны от нуля, причем $d_{s^k} = 1$, $d_{r^k} = -1$.

Целью игрока нижнего уровня является минимизация стоимости передачи данных, а целью игрока верхнего уровня — максимизация дохода. Таким образом, возникает следующая двухуровневая задача:

$$y \in \underset{y}{\operatorname{Argmin}} \left\{ \begin{array}{l} \sum_{k=1}^K \langle x, y_1^k \rangle \uparrow \underset{x}{\operatorname{max}} \text{ "max"}, \quad x \in \prod_{i=1}^{n_1} [\underline{x}_i, \bar{x}_i], \\ \sum_{k=1}^K (\langle c^1 + x, y_1^k \rangle + \langle c^2, y_2^k \rangle) \mid Ay^k = \delta^k d^k, \quad 0 \leq y_j^k \leq \bar{y}_j^k, \\ j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K \end{array} \right\}$$

Полученная задача является билинейно-билинейной задачей Штакельберга, поскольку критерии эффективности игроков являются билинейными функциями и ограничения на верхнем и на нижнем уровне накладываются только на переменные соответствующего уровня. #

3) Проектирование сети шоссейных дорог [60, 62].

Рассматривается задача улучшения сети шоссейных дорог, соединяющей города некоторой развивающейся страны. При этом страна считается развивающейся, если все ее шоссейные дороги состоят не более чем из двух полос (или, по крайней мере, большинство из них). Высокая стоимость расширения имеющихся и строительства новых шоссейных дорог приводит к необходимости тщательного планирования таких операций.

Поскольку двухполосные шоссейные дороги предполагают, что встречная полоса используется, в частности, для того, чтобы более медленные транспортные средства могли уступать дорогу более быстрым, когда это возможно, считается уместным оперировать суммарным транспортным потоком в обоих направлениях во всех функциях, оценивающих время в дороге.

Обозначим множество всех шоссейных дорог в транспортной сети через R , множество городов — через S .

Каждая компонента $x_i \geq 0$ стратегии игрока верхнего уровня (проектировщика дорожной сети) играет роль планируемого увеличения пропускной способности i -той дороги, входящей в дорожную сеть.

Стратегии игрока нижнего уровня (агрегированного “пользователя” дорог) включают в себя транспортный поток $y_i \geq 0$, $i \in R$, на каждой из дорог, транспортные

потоки $z_{ij} \geq 0$, $i \in R$, $j \in C$, по дороге i в город j (с выполнением условия согласования $\sum_{j \in C} z_{ij} = y_i \forall i \in R$), а также вспомогательные переменные u_i, v_i, w_{li} , $i \in R$, $j \in C$, $l \in \{1, 2, \dots, J_i\}$.

Значения переменных $u_i \geq 0$ и $v_i \geq 0$ аппроксимируют значения функций стоимости перевозки по дороге $i \in R$ для проектировщика и “пользователя” дорог:

$$u_i \geq r_{li}y_i - g_{li}x_i + s_{li}, \quad l = 1, \dots, I_i,$$

$$v_i \geq \hat{r}_{li}y_i - \hat{g}_{li}x_i + \hat{s}_{li}, \quad l = 1, \dots, J_i,$$

где I_i, J_i — количество сегментов в кусочно-линейной аппроксимации соответствующей функции стоимости перевозки по дороге $i \in R$, r_{li}, g_{li}, s_{li} и $\hat{r}_{li}, \hat{g}_{li}, \hat{s}_{li}$ — коэффициенты уравнений прямых, составляющих эти аппроксимации.

Дополнительная минимизация функции $\sum_{l=1}^{K_i} w_{li}$ при ограничениях

$$w_{li} \geq x_i - q_{li}, \quad w_{li} \geq 0, \quad l = 1, \dots, K_i,$$

на нижнем уровне обеспечивает то, что значение выражения

$$a_{1i}x_i + b_{1i} + \sum_{l=1}^{K_i-1} (a_{l+1,i} - a_{li})w_{li}$$

совпадает со значением кусочно-линейной аппроксимации некоторой функции затрат на расширение дороги $i \in R$

$$f(x_i) = a_{li}x_i + b_{li}, \quad x_i \in [q_{l-1,i}, q_{li}], \quad l \in \{1, 2, \dots, K_i\}.$$

Фиксированные объемы перевозок e_{kj} из города $k \in C$ в город $j \in C$, которые осуществляет игрок нижнего уровня, накладывают ограничения на транспортные потоки z_{ij} :

$$\sum_{i \in A_k} z_{ij} - \sum_{i \in B_k} z_{ij} = e_{kj} \quad \forall k, j \in C,$$

где множества A_k и B_k , $k \in C$, содержат начальные и конечные пункты маршрутов соответственно.

Цель игрока верхнего уровня — оптимальное распределение инвестиций на улучшение дорог при минимизации суммарной функции стоимости перевозок с учетом предпочтений игрока нижнего уровня, который стремится минимизировать свою функцию стоимости перевозок.

В итоге, возникает следующая двухуровневая задача оптимизации:

$$\left. \begin{aligned}
 & \sum_{i \in R} \left[u_i + \tau \left(a_{1i} x_i + b_{1i} + \sum_{l=1}^{K_i-1} (a_{l+1,i} - a_{li}) w_{li} \right) \right] \downarrow \text{“min”}_x, \\
 & (y, z, u, v, w) \in \operatorname{Argmin}_{y, z, u, v, w} \left\{ \sum_{i \in R} \left(v_i + \varepsilon u_i + \sum_{l=1}^{K_i-1} w_{li} \right) \mid \right. \\
 & \quad u_i \geq r_{li} y_i - g_{li} x_i + s_{li}, \quad l = 1, \dots, I_i, \\
 & \quad v_i \geq \hat{r}_{li} y_i - \hat{g}_{li} x_i + \hat{s}_{li}, \quad l = 1, \dots, J_i, \\
 & \quad w_{li} \geq x_i - q_{li}, \quad l = 1, \dots, K_i, \\
 & \quad \sum_{j \in C} z_{ij} = y_i, \quad \sum_{i \in A_k} z_{ij} - \sum_{i \in B_k} z_{ij} = e_{kj}, \\
 & \quad k, j \in C, \quad i \in R, \\
 & \quad \left. x \geq 0, y \geq 0, z \geq 0, u \geq 0, v \geq 0, w \geq 0 \right\},
 \end{aligned} \right\}$$

где $\tau > 0$ — множитель, позволяющий конвертировать стоимость затрат на расширение дорог в единицы, в которых измеряется стоимость перевозки, $\varepsilon > 0$ — малый вещественный параметр, который необходимо выбирать так, чтобы компоненты y, z, v, w решения задачи нижнего уровня совпадали с таковыми при $\varepsilon = 0$ при любом выборе стратегии x игроком верхнего уровня.

Отметим, что полученная задача является линейно-линейной двухуровневой задачей, причем ограничений в задаче верхнего уровня нет (кроме экстремального), а допустимое множество задачи нижнего уровня зависит от выбора стратегии игроком верхнего уровня. #

Методика исследования. При выполнении работы использовались элементы выпуклого анализа, аппарат математического программирования, теория многозначных отображений, а также новейшие достижения теории глобального экстремума.

Предложенный в диссертации подход к разработке методов решения квадратично-линейных двухуровневых задач состоит из двух этапов:

- 1) Редукция двухуровневой задачи к семейству одноуровневых задач оптимизации;
- 2) Разработка алгоритмов локального и глобального поиска в редуцированных задачах, основанных на теории глобального поиска, предложенной А.С. Стрекаловским [43].

Так для поиска оптимистических решений в квадратично-линейных двухуровневых задачах в диссертации предлагается метод, который можно отнести к группе методов 2 согласно предложенной выше классификации. При этом вспомогательные задачи

со штрафом, полученные при редукции исходной двухуровневой задачи, оказываются невыпуклыми, что демонстрирует скрытую невыпуклость исследуемых задач.

Достаточно сложно привести полную классификацию задач невыпуклой (или глобальной) оптимизации. В частности, в литературе [43, 93] выделяют следующие важные классы невыпуклых задач, содержащие d.c. функции (такие функции, которые могут быть представлены в виде разности двух выпуклых функций):

1) задача d.c. минимизации:

$$g(x) - f(x) \downarrow \min_x, x \in S, \quad (\mathcal{DC})$$

где функции $f(\cdot)$, $g(\cdot)$ и множество S выпуклы;

2) задача оптимизации с одним d.c. ограничением:

$$\begin{cases} h(x) \downarrow \min_x, x \in S, \\ g(x) - f(x) \leq (=) 0, \end{cases} \quad (\mathcal{DCC})$$

где $f(\cdot)$, $g(\cdot)$, $h(\cdot)$, S — выпуклы.

В диссертационной работе осуществляется редукция квадратично-линейной задачи в оптимистической постановке к задаче класса (\mathcal{DCC}) , которая затем сводится к серии задач класса (\mathcal{DC}) , для решения которых используется теория глобального поиска в задачах d.c. минимизации [43]. Основными этапами алгоритмов глобального поиска, основанных на данной теории, являются:

1) Алгоритм локального поиска, учитывающий специфику невыпуклости конкретной задачи;

2) Процедуры выхода из точки, найденной алгоритмом локального поиска, основанные на условиях глобальной оптимальности.

Алгоритмы, основанные на теории глобального поиска, неоднократно демонстрировали свою вычислительную эффективность при решении многих задач исследования операций (см., например, [5, 36, 38, 43, 47, 119]). Необходимо отметить, что внутри алгоритмов локального поиска, составляющих один из основных этапов стратегии глобального поиска, работают классические методы выпуклой оптимизации [2, 4, 6, 14, 39].

Для решения квадратично-линейных задач в гарантированной постановке в работе прежде всего осуществляется их редукция к семейству задач поиска оптимистического решения вспомогательных квадратично-квадратичных двухуровневых задач (в которых

критерий эффективности игрока нижнего уровня — тоже квадратичная функция). Тем самым осуществляется развитие подхода к поиску гарантированных решений в задачах Штакельберга, предложенного Д.А. Молодцовым и В.В. Федоровым [34, 35, 103]. Дальнейшее сведение к задачам d.c. минимизации осуществляется на базе описанного выше подхода к поиску оптимистических решений квадратично-линейных двухуровневых задачах. При разработке методов локального и глобального поиска в полученных при таком сведении задачах, как и ранее, используется теория глобального поиска в задачах d.c. минимизации.

Далее опишем **структуру диссертации**. Диссертационная работа состоит из введения, трех глав, заключения и списка литературы из 134 наименований.

Первая глава посвящена разработке и численному тестированию метода поиска оптимистических решений в квадратично-линейных двухуровневых задачах вида

$$\left\{ \begin{array}{l} F_1(x, y) \downarrow \min_{x, y}, \quad Ax + By \leq a, \\ y \in \underset{z}{\text{Argmin}}\{\langle d, z \rangle \mid A_1x + B_1z \leq b\}, \end{array} \right. \quad (\mathcal{P}_1)$$

где $F_1(x, y) = \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle$, $C = C^T \geq 0$, $C_1 = C_1^T \geq 0$ (т.е. матрицы C, C_1 являются неотрицательно определенными или, что то же самое, положительно полуопределенными). Отметим, что в случае, когда $C = 0$, $C_1 = 0$, задача (\mathcal{P}_1) представляет собой линейно-линейную задачу двухуровневой оптимизации.

В §1.1 осуществляется постановка задачи и ее сведение к серии невыпуклых одноуровневых задач оптимизации. В §1.2 разработан алгоритм локального поиска для полученных при такой редукции невыпуклых одноуровневых задач. §1.3 посвящен численному тестированию алгоритма локального поиска из §1.2 на серии случайно сгенерированных тестовых двухуровневых задач. Далее в §1.4 предлагается алгоритм глобального поиска в редуцированных одноуровневых задачах. Наконец, в §1.5 осуществляется тестирование двух программ, реализующих алгоритм глобального поиска из §1.4 на сериях случайно сгенерированных тестовых задач, включающих в себя задачи из §1.3, а также численное сравнение глобального поиска с одним известным коммерческим пакетом для решения задач нелинейной оптимизации, ориентированным, в частности, на задачи, исследуемые в данной главе.

Во **второй главе** предлагается новый метод численного поиска гарантированных решений в квадратично-линейных двухуровневых задачах вида

$$\begin{cases} \sup_y \{F_2(x, y) \mid y \in Y_*(x)\} \downarrow \min_x, & Ax \leq a, \\ Y_*(x) \triangleq \underset{z}{\text{Argmin}} \{ \langle d, z \rangle \mid A_1x + B_1z \leq b \}, \end{cases} \quad (\mathcal{P}_2)$$

где $F_2(x, y) = \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle - \frac{1}{2} \langle y, C_1y \rangle + \langle c_1, y \rangle$, $C = C^T \geq 0$, $C_1 = C_1^T \geq 0$.

В §2.1 осуществляется постановка задачи и доказаны некоторые ее свойства. В §2.2 отдельно рассматривается задача нижнего уровня исследуемой двухуровневой задачи и изучаются свойства ее оптимального значения и множества ее решений. Затем, с использованием полученных в §2.1 и §2.2 результатов, в §2.3 осуществляется редукция исследуемой двухуровневой задачи в гарантированной постановке к семейству двухуровневых задач в оптимистической постановке, которые затем сводятся к семейству невыпуклых одноуровневых задач. Для редуцированных одноуровневых задач из §2.3 в §2.4 разрабатываются два варианта алгоритма локального поиска. §2.5 содержит описание нового алгоритма глобального поиска в редуцированных одноуровневых задачах.

Третья глава диссертации посвящена численному поиску гарантированных решений в тестовых двухуровневых задачах класса (\mathcal{P}_2) .

Проведен вычислительный эксперимент по тестированию разработанных в главе 2 алгоритмов локального и глобального поиска гарантированных решений. В связи с отсутствием общедоступных коллекций тестовых двухуровневых задач в гарантированной постановке и методов их построения, предлагается новый метод генерации таких задач. Этот метод развивает на случай гарантированного решения известную методику генерации тестовых двухуровневых задач с оптимистическим решением [63, 64]. Согласно этой методике построение тестовых задач двухуровневой оптимизации осуществляется путем объединения произвольного количества задач-ядер нескольких видов с известными локальными и глобальными решениями в одну сепарабельную задачу. При этом выбор конкретных видов задач-ядер влияет на количество локальных и глобальных решений в построенной двухуровневой задаче, определяя тем самым ее сложность с точки зрения поиска глобального решения. Затем, чтобы избавиться от сепарабельности построенной двухуровневой задачи специальным образом производится ее преобразование и доказываются утверждения о конкретном виде локальных и глобальных решений в

преобразованной задаче.

Описанный новый метод генерации тестовых квадратично-линейных двухуровневых задач в гарантированной постановке представлен в §3.1. В §3.2 производится тестирование алгоритма локального поиска из §2.3 на серии задач, сгенерированной описанным методом. Наконец, в §3.3 осуществляется тестирование алгоритма глобального поиска, предложенного в §2.4, на серии тестовых задач из §3.2 и на сериях сгенерированных двухуровневых задач высокой размерности.

Научная новизна. Для поиска оптимистических и гарантированных решений в двухуровневых задачах высокой размерности (которые являются невыпуклыми структурами в неявной форме) разработан и обоснован новый подход, заключающийся

- во-первых, в редукции исходной задачи к серии невыпуклых одноуровневых задач оптимизации;
- во-вторых, в применении теории глобального поиска для отыскания глобальных решений в полученных невыпуклых задачах.

При этом для квадратично-линейной двухуровневой задачи в оптимистической постановке получены следующие результаты:

- а) построен и обоснован специальный метод локального поиска;
- б) разработан алгоритм глобального поиска;
- в) разработаны и протестированы программы, реализующие предложенные алгоритмы локального и глобального поиска.

Кроме того, для квадратично-линейной двухуровневой задачи в гарантированной постановке получены следующие результаты:

- а) обосновано ее сведение к серии задач двухуровневой оптимизации специального вида в оптимистической постановке;
- б) предложен и обоснован специальный метод локального поиска;
- в) построен алгоритм глобального поиска;
- г) разработан метод генерации тестовых двухуровневых задач произвольной размерности с известными локальными и глобальными гарантированными решениями;
- д) разработаны и протестированы программы, реализующие предложенные алгоритмы локального и глобального поиска.

Отметим, что разработка численного метода для поиска гарантированных решений

в задачах двухуровневой оптимизации, в которых допустимое множество задачи нижнего уровня зависит от стратегий игрока верхнего уровня, осуществляется, насколько известно, впервые.

Научные положения выносимые на защиту.

1) Предложены и обоснованы методы локального и глобального поисков оптимистических решений в квадратично-линейных двухуровневых задачах, основанные на теории глобального поиска. Осуществлен вычислительный эксперимент, продемонстрировавший эффективность этих методов.

2) Обоснована редукция задачи поиска гарантированного решения квадратично-линейной двухуровневой задачи к серии специальных квадратично-квадратичных двухуровневых задач в оптимистической постановке.

3) Разработаны методы локального и глобального поисков гарантированных решений в квадратично-линейных двухуровневых задачах, основанные на теории глобального поиска. Проведен вычислительный эксперимент, показавший эффективность этих методов на сериях новых тестовых примеров, построенных разработанным методом генерации тестовых задач.

Полученные научные результаты соответствуют пунктам 1, 4, 5 паспорта специальности 05.13.01 — “Системный анализ, управление и обработка информации (в технике, экологии, экономике)”.

Достоверность научных положений обусловлена прежде всего строгостью математических доказательств и использованием апробированных научных методов и средств. Эффективность разработанных алгоритмов подтверждается результатами проведенных вычислительных экспериментов.

Список публикаций. Полученные в работе результаты опубликованы в [20–32, 37, 44–46, 48, 49, 105, 120], в том числе в четырех статьях в журналах из списка ВАК [31, 32, 48, 49] и в одной статье в журнале, включенном в базу Web of science [120].

Апробация результатов. Результаты, полученные в диссертации, докладывались и обсуждались на IX школе-семинаре молодых ученых “Математическое моделирование и информационные технологии” (Иркутск–Ангасолка, 2007), ежегодных конференциях “Ляпуновские чтения и презентация информационных технологий” в ИДСТУ СО РАН (Иркутск, 2007–2009), XIV Байкальской международной школе-семинаре “Мето-

ды оптимизации и их приложения” (Иркутск–Северобайкальск, 2008), Всероссийской конференции “Математическое моделирование и вычислительно-информационные технологии в междисциплинарных научных исследованиях” (Иркутск, 2009), IV Всероссийской конференции “Проблемы оптимизации и экономические приложения” (Омск, 2009), XI Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям (Иркутск–Байкал, 2010), II Международной школе-семинаре “Нелинейный анализ и экстремальные задачи” (Иркутск, 2010), VI Московской международной конференции по исследованию операций (Москва, 2010), XIV Всероссийской конференции “Математическое программирование и приложения” (Екатеринбург, 2011).

Кроме того, результаты диссертационной работы обсуждались на семинаре кафедры исследования операций факультета ВМК МГУ, семинаре отдела математического программирования ИММ УрО РАН и неоднократно обсуждались на семинарах в ИДСТУ СО РАН.

Глава 1

Глобальный поиск оптимистических решений в двухуровневых задачах

В настоящей главе диссертации рассматривается один класс квадратично-линейных задач двухуровневой оптимизации в оптимистической постановке. Вначале осуществляется редукция таких двухуровневых задач к серии задач d.c. минимизации, затем — их численное решение. Для решения редуцированных задач используется теория глобального поиска в задачах d.c. минимизации [43]. Одним из основных элементов алгоритмов глобального поиска, базирующихся на стратегии глобального поиска из [43], является специализированный локальный поиск, который принимает во внимание структуру исследуемой задачи. В данной главе разработаны, обоснованы и протестированы два варианта специального алгоритма локального поиска, а также алгоритм глобального поиска, их использующий.

В начале данной главы производится постановка квадратично-линейной двухуровневой задачи и проводится редукция задачи поиска ее оптимистического решения к серии невыпуклых задач оптимизации с квадратично-билинейной структурой. Далее предлагается и обосновывается метод локального поиска, описываются результаты численного его тестирования. Затем предложен и протестирован на специально сгенерированных сериях задач алгоритм глобального поиска.

1.1 Постановка задачи и ее редукция

Рассмотрим следующую квадратично-линейную задачу двухуровневой оптимизации в оптимистической постановке:

$$\left. \begin{aligned} F(x, y) &\triangleq \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle \downarrow \min_{x, y}, \\ (x, y) &\in X \triangleq \{(x, y) \in \mathbb{R}^m \times \mathbb{R}^n \mid Ax + By \leq a, \ x \geq 0\}, \\ y &\in Y_*(x) \triangleq \underset{z}{\operatorname{Argmin}}\{\langle d, z \rangle \mid z \in Y(x)\}, \\ Y(x) &\triangleq \{y \in \mathbb{R}^n \mid A_1x + B_1y \leq b, \ y \geq 0\}, \end{aligned} \right\} \quad (\mathcal{BP})$$

где $A \in \mathbb{R}^{p \times m}$ (матрица $p \times m$), $B \in \mathbb{R}^{p \times n}$, $A_1 \in \mathbb{R}^{q \times m}$, $B_1 \in \mathbb{R}^{q \times n}$, $C \in \mathbb{R}^{m \times m}$, $C_1 \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^m$, $c_1, d \in \mathbb{R}^n$, $a \in \mathbb{R}^p$, $b \in \mathbb{R}^q$, кроме того, $C \geq 0$, $C_1 \geq 0$.

Целевая функция верхнего уровня $F(x, y)$ задачи (\mathcal{BP}) является выпуклой квадратичной, с несвязанными переменными. Целевая функция нижнего уровня является линейной. Отметим, что слагаемое $\langle d_1, x \rangle$ в целевой функции задачи нижнего уровня опущено, поскольку является константой в этой задаче и на ее решение не влияет.

Подчеркнем, что двухуровневая задача (\mathcal{BP}) отличается от задачи Штакельберга тем, что в ограничения на верхнем уровне $((x, y) \in X)$ и на нижнем уровне $(y \in Y(x))$ входят переменные обеих уровней иерархии.

Существенным предположением в приведенной постановке задачи (\mathcal{BP}) является кооперативность поведения игроков верхнего и нижнего уровня. Предполагается, что в случае неединственности решения задачи нижнего уровня при фиксированной стратегии игрока верхнего уровня игрок нижнего уровня выберет (из множества своих оптимальных стратегий) стратегию, наиболее благоприятствующую игроку верхнего уровня, что и отражено в требовании минимизации целевой функции игрока верхнего уровня по переменным обеих уровней. Поэтому постановка задачи (\mathcal{BP}) и названа оптимистической, а ее решение — оптимистическим (или кооперативным) [60, 74].

Кроме того, предположим, что функция $F(\cdot)$ ограничена снизу на множестве X , так что

$$(\mathcal{H}) : \quad \exists F_- > -\infty : F(x, y) \geq F_- \quad \forall (x, y) \in X, \quad (1.1.1)$$

а функция $\langle d, \cdot \rangle$ ограничена снизу на множестве $Y(x)$ для всех

$x \in Pr(X) \triangleq \{x \in \mathbb{R}^m \mid \exists y : (x, y) \in X\}$, так что

$$(\mathcal{H}) : \quad \inf_x \inf_y \{\langle d, y \rangle \mid y \in Y(x), x \in Pr(X)\} > -\infty. \quad (1.1.2)$$

Как упоминалось во введении, одним из общепринятых подходов к поиску оптимального решения в задаче (\mathcal{BP}) является ее сведение к одноуровневой задаче математического программирования [67, 74]. Для осуществления такой редукции отдельно рассмотрим задачу нижнего уровня, зависящую от x как от параметра:

$$\left. \begin{array}{l} \langle d, y \rangle \downarrow \min_y, \\ A_1 x + B_1 y \leq b, \quad y \geq 0. \end{array} \right\} \quad (\mathcal{FP}(x))$$

Поскольку ограничения этой задачи линейны, для нее автоматически выполняется условие регулярности Абади (см., например, [2]). Значит условия Каруша-Куна-Таккера (ККТ) могут быть сформулированы здесь следующим образом.

Теорема 1.1.1 [2, 6] *Для того, чтобы точка $y \in \mathbb{R}^n$ была решением задачи $(\mathcal{FP}(x))$, необходимо и достаточно существования вектора множителей Лагранжа $v \in \mathbb{R}^q$ такого, чтобы пара (y, v) удовлетворяла системе*

$$\left. \begin{array}{l} A_1 x + B_1 y \leq b, \quad y \geq 0, \quad d + v B_1 \geq 0, \quad v \geq 0, \\ \langle d, y \rangle - \langle A_1 x - b, v \rangle = 0. \end{array} \right\} \quad (1.1.3)$$

Заменив в (\mathcal{BP}) задачу нижнего уровня равносильными ей условиями ККТ (1.1.3) и добавив в постановку минимизацию целевой функции по переменной v , получим следующую задачу:

$$\left. \begin{array}{l} F(x, y) \triangleq \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2} \langle y, C_1 y \rangle + \langle c_1, y \rangle \downarrow \min_{x, y, v}, \\ Ax + By \leq a, \quad A_1 x + B_1 y \leq b, \quad d + v B_1 \geq 0, \\ \langle d, y \rangle - \langle A_1 x - b, v \rangle = 0, \quad x \geq 0, \quad y \geq 0, \quad v \geq 0. \end{array} \right\} \quad (\mathcal{P})$$

Отметим, что в результате перехода от двухуровневой задачи (\mathcal{BP}) к одноуровневой задаче (\mathcal{P}) повышается размерность задачи. А именно, появляется новая переменная — вектор множителей Лагранжа $v \in \mathbb{R}^q$.

Эквивалентность задач (\mathcal{BP}) и (\mathcal{P}) устанавливается следующим результатом (см. также [60, 67, 74, 115]).

Теорема 1.1.2 Для того, чтобы пара (x^*, y^*) была глобальным решением задачи (\mathcal{BP}) , необходимо и достаточно существования вектора v^* , такого чтобы тройка (x^*, y^*, v^*) являлась глобальным решением задачи (\mathcal{P}) .

Доказательство. Достаточность. Пусть $(x^*, y^*, v^*) \in \text{Sol}(\mathcal{P})$ (является решением задачи (\mathcal{P})), тогда нетрудно видеть, что пара (x^*, y^*) является допустимой в задаче (\mathcal{BP}) , поскольку в задачу (\mathcal{P}) входят ограничения верхнего и нижнего уровня задачи (\mathcal{BP}) . Кроме того, $y^* \in Y_*(x^*)$, поскольку выполнены необходимые и достаточные условия ККТ (1.1.3).

Предположим теперь, что $(x^*, y^*) \notin \text{Sol}(\mathcal{BP})$, так что $\exists(\hat{x}, \hat{y}) : F(\hat{x}, \hat{y}) < F(x^*, y^*)$, $A\hat{x} + B\hat{y} \leq a$, $\hat{x} \geq 0$, $\hat{y} \in Y_*(\hat{x})$. Поскольку \hat{y} является решением задачи $(\mathcal{FP}(\hat{x}))$, то по теореме 1.1.1 (Каруша-Куна-Таккера) найдется вектор \hat{v} , такой что $(\hat{x}, \hat{y}, \hat{v})$ — допустимая точка в (\mathcal{P}) , и $F(\hat{x}, \hat{y}) < F(x^*, y^*)$. Это противоречит тому, что $(x^*, y^*, v^*) \in \text{Sol}(\mathcal{P})$.

Необходимость. Пусть $(x^*, y^*) \in \text{Sol}(\mathcal{BP})$, тогда снова по теореме 1.1.1 найдется v^* , такой что тройка (x^*, y^*, v^*) является допустимой в (\mathcal{P}) .

Далее, предположим, что $(x^*, y^*, v^*) \notin \text{Sol}(\mathcal{P})$, так что найдется тройка $(\hat{x}, \hat{y}, \hat{v})$ — допустимая в задаче (\mathcal{P}) и такая что $F(\hat{x}, \hat{y}) < F(x^*, y^*)$. Поскольку (\hat{x}, \hat{y}) является допустимой в (\mathcal{BP}) и, при этом, выполнено $F(\hat{x}, \hat{y}) < F(x^*, y^*)$, получаем противоречие с тем, что $(x^*, y^*) \in \text{Sol}(\mathcal{BP})$. #

Отметим, что задача (\mathcal{P}) оказывается задачей оптимизации с d.c. ограничением равенством (поскольку билинейная функция может быть представлена в виде разности двух выпуклых функций [47]). Тем самым поиск оптимистического решения исходной квадратично-линейной задачи двухуровневой оптимизации эквивалентен решению невыпуклой задачи оптимизации.

Отдельно подчеркнем, что можно говорить об эквивалентности задач (\mathcal{BP}) и (\mathcal{P}) только с точки зрения глобального решения, поскольку известно [60, 74], что совпадение локальных решений в этих задачах, вообще говоря, не имеет места.

Таким образом, для решения задачи (\mathcal{BP}) достаточно решить задачу (\mathcal{P}) напрямую с использованием любого из известных методов невыпуклой оптимизации [43, 93], например, стратегии глобального поиска для задач с d.c. ограничением из [12, 13, 40, 41]. Однако существует и другой подход [67, 74], основанный на сведениях задачи (\mathcal{BP}) к

серии невыпуклых задач оптимизации с выпуклым допустимым множеством, который будет использован далее.

Обозначим $h(x, y, v) \triangleq \langle d, y \rangle - \langle A_1x - b, v \rangle$ и введем μ -параметрическое семейство задач:

$$\left. \begin{aligned} & F(x, y) + \mu h(x, y, v) \downarrow \min_{x, y, v}, \\ & (x, y, v) \in D \triangleq \{(x, y, v) \mid Ax + By \leq a, \quad A_1x + B_1y \leq b, \quad d + vB_1 \geq 0, \\ & \quad x \geq 0, \quad y \geq 0, \quad v \geq 0\}, \end{aligned} \right\} \quad (\mathcal{P}(\mu))$$

где $\mu > 0$ — штрафной множитель. Переписав функцию $h(\cdot)$ в эквивалентном виде $h(x, y, v) \triangleq \langle d + vB_1, y \rangle - \langle A_1x + B_1y - b, v \rangle$, нетрудно видеть, что

$$h(x, y, v) \geq 0 \quad \forall (x, y, v) \in D, \quad (1.1.4)$$

или $h(x, y, v)$ неотрицательна на произвольных допустимых в задаче $(\mathcal{P}(\mu))$ векторах x, y, v . При фиксированном μ задача $(\mathcal{P}(\mu))$ является выпуклой и квадратичной по паре переменных (x, y) и билинейной по переменным x и v . Будем далее называть ее квадратично-билинейной.

Далее рассмотрим связь между решением задачи $(\mathcal{P}(\mu))$ и решением задачи (\mathcal{P}) . Пусть $(x(\mu), y(\mu), v(\mu))$ — решение задачи $(\mathcal{P}(\mu))$ при некотором значении μ . Обозначим $h[\mu] \triangleq h(x(\mu), y(\mu), v(\mu))$. Тогда справедливо (см. также лемму 9.2.1 и теорему 9.2.2 из [2])

Предложение 1.1.1 *i) Пусть при некотором $\mu = \hat{\mu}$ на решении $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ задачи $(\mathcal{P}(\mu))$ справедливо равенство: $h(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu})) = 0$. Тогда тройка $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ является решением задачи (\mathcal{P}) .*

ii) Для всех значений параметра $\mu > \hat{\mu}$ функция $h[\mu]$ обращается в ноль, т.е. $h(x(\mu), y(\mu), v(\mu)) = 0$, и $(x(\mu), y(\mu), v(\mu))$ является решением задачи (\mathcal{P}) .

Доказательство. *i)* Пусть тройка $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ является решением задачи $(\mathcal{P}(\hat{\mu}))$ и, кроме того, $h(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu})) = 0$. Тогда с учетом определения функции $h(\cdot)$ точка $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ является допустимой в задаче (\mathcal{P}) . Далее, пусть найдется допустимая в задаче (\mathcal{P}) тройка $(\bar{x}, \bar{y}, \bar{v})$, такая что $F(\bar{x}, \bar{y}) < F(x(\hat{\mu}), y(\hat{\mu}))$. Тогда эта тройка допустима и в задаче $(\mathcal{P}(\hat{\mu}))$, имеющей более широкое допустимое множество. Кроме того, имеем $h(\bar{x}, \bar{y}, \bar{v}) = 0$. Таким образом, выполняется неравенство

$$F(\bar{x}, \bar{y}) + \mu h(\bar{x}, \bar{y}, \bar{v}) < F(x(\hat{\mu}), y(\hat{\mu})) + \mu h(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu})),$$

которое, с учетом допустимости тройки $(\bar{x}, \bar{y}, \bar{v})$ в задаче $(\mathcal{P}(\hat{\mu}))$, противоречит тому, что точка $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ является решением задачи $(\mathcal{P}(\hat{\mu}))$.

ii) Покажем сначала, что функция $h[\mu]$ является монотонно невозрастающей. Пусть $0 < \mu_1 < \mu_2$. По определению решения задачи $(\mathcal{P}(\mu))$, имеют место следующие неравенства:

$$F(x(\mu_1), y(\mu_1), v(\mu_1)) + \mu_1 h[\mu_1] \leq F(x(\mu_2), y(\mu_2), v(\mu_2)) + \mu_1 h[\mu_2],$$

$$F(x(\mu_2), y(\mu_2), v(\mu_2)) + \mu_2 h[\mu_2] \leq F(x(\mu_1), y(\mu_1), v(\mu_1)) + \mu_2 h[\mu_1].$$

Складывая эти неравенства, получаем: $(\mu_2 - \mu_1)(h[\mu_1] - h[\mu_2]) \geq 0$. Поскольку $\mu_2 > \mu_1$, из последнего неравенства следует, что $h[\mu_1] \geq h[\mu_2]$.

Пусть теперь $\mu = \check{\mu} > \hat{\mu}$, тогда с учетом монотонности функции $h[\cdot]$ и неравенства (1.1.4) заключаем, что $h(x(\check{\mu}), y(\check{\mu}), v(\check{\mu})) = 0$. В силу доказанного выше, решение $(x(\check{\mu}), y(\check{\mu}), v(\check{\mu}))$ задачи $(\mathcal{P}(\check{\mu}))$ является решением задачи (\mathcal{P}) . #

Таким образом, при выполнении равенства $h[\mu] = 0$ решение задачи $(\mathcal{P}(\mu))$ является решением задачи (\mathcal{P}) , и с ростом значения μ данная ситуация сохраняется. Изучим поведение функции $h[\mu]$ при неограниченном увеличении μ .

Предложение 1.1.2 Пусть $\exists \hat{\mu} \geq 0 : \forall \mu \geq \hat{\mu} \exists (x(\mu), y(\mu), v(\mu)) \in \text{Sol}(\mathcal{P}(\mu))$. Тогда $h[\mu] = h(x(\mu), y(\mu), v(\mu)) \rightarrow 0$ при $\mu \rightarrow +\infty$.

Доказательство. Предположим, что при некотором значении $\mu = \check{\mu} \geq \hat{\mu}$ выполнилось равенство $h(x(\mu), y(\mu), v(\mu)) = 0$. В этом случае утверждение следует из пункта ii) предложения 1.1.1.

Остается рассмотреть случай, когда $h(x(\mu), y(\mu), v(\mu)) > 0 \forall \mu \geq 0$. По определению решения задачи $(\mathcal{P}(\mu))$:

$$F(x(\mu), y(\mu)) + \mu h(x(\mu), y(\mu), v(\mu)) \leq F(x, y) + \mu h(x, y, v) \quad \forall (x, y, v) \in D.$$

Поскольку неравенство верно для всех допустимых троек (x, y, v) , оно будет справедливо и для тех из них, на которых $h(x, y, v) = 0$:

$$F(x(\mu), y(\mu)) + \mu h(x(\mu), y(\mu), v(\mu)) \leq F(x, y) \quad \forall (x, y, v) \in D : h(x, y, v) = 0.$$

Отсюда, с учетом предположения (\mathcal{H}) –(1.1.1), получаем:

$$\mu h(x(\mu), y(\mu), v(\mu)) \leq F(x, y) - F(x(\mu), y(\mu)) \leq F(x, y) - F_- \forall (x, y, v) \in D : h(x, y, v) = 0.$$

Зафиксируем $x = \bar{x}, y = \bar{y}$:

$$\mu \leq \frac{F(\bar{x}, \bar{y}) - F_-}{h(x(\mu), y(\mu), v(\mu))}.$$

Устремляя $\mu \rightarrow +\infty$, получаем $h[\mu] \triangleq h(x(\mu), y(\mu), v(\mu)) \rightarrow 0$. #

Следующее предложение поясняет ситуацию, если для некоторого μ мы получили, что $0 < h[\mu] < \tau$.

Предложение 1.1.3 Пусть $(x(\mu), y(\mu), v(\mu)) \in D$ является τ_1 -решением (приближенным решением с точностью τ_1) задачи $(\mathcal{P}(\mu))$ и при этом $h(x(\mu), y(\mu), v(\mu)) \leq \tau_2$. Тогда *i)* $y(\mu)$ является τ_2 -решением задачи нижнего уровня в (\mathcal{BP}) при $x = x(\mu)$, *ii)* $(x(\mu), y(\mu))$ является приближенным τ_1 -решением задачи (\mathcal{BP}) .

Доказательство. *i)* Для произвольной допустимой в задаче нижнего уровня точки $y \in Y(x(\mu)) \triangleq \{y \geq 0 \mid A_1 x(\mu) + B_1 y \leq b\}$ запишем очевидные равенства:

$$\begin{aligned} \langle d, y(\mu) - y \rangle &= \langle d, y(\mu) \rangle - \langle d, y \rangle \pm \langle A_1 x(\mu) - b, v(\mu) \rangle = \\ &= h(x(\mu), y(\mu), v(\mu)) - \langle d, y \rangle + \langle A_1 x(\mu) - b, v(\mu) \rangle = \\ &= h(x(\mu), y(\mu), v(\mu)) - h(x(\mu), y, v(\mu)). \end{aligned} \quad (1.1.5)$$

Далее, в силу включения $(x(\mu), y(\mu), v(\mu)) \in D$ имеет место включение

$$v(\mu) \in V \triangleq \{v \mid d + vB_1 \geq 0, v \geq 0\}. \quad (1.1.6)$$

Используя включения (1.1.6) и $y \in Y(x(\mu)) = \{y \mid A_1 x(\mu) + B_1 y - b \leq 0, y \geq 0\}$, получаем неравенство:

$$h(x(\mu), y, v(\mu)) \triangleq \langle d + v(\mu)B_1, y \rangle - \langle A_1 x(\mu) + B_1 y - b, v(\mu) \rangle \geq 0 \quad \forall y \in Y(x(\mu)). \quad (1.1.7)$$

Продолжим теперь цепочку (1.1.5), с учетом неравенств $h(x(\mu), y(\mu), v(\mu)) \leq \tau_2$ и (1.1.7):

$$h(x(\mu), y(\mu), v(\mu)) - h(x(\mu), y, v(\mu)) \leq \tau_2. \quad (1.1.5')$$

В результате из (1.1.5)–(1.1.5'), получаем

$$\langle d, y(\mu) \rangle \leq \langle d, y \rangle + \tau_2 \quad \forall y \in Y(x(\mu)),$$

так что $y(\mu)$ является τ_2 -решением задачи нижнего уровня в (\mathcal{BP}) при $x = x(\mu)$.

ii) С учетом неравенства (1.1.4) справедлива цепочка

$$\begin{aligned} F(x(\mu), y(\mu)) &\leq F(x(\mu), y(\mu)) + \mu h(x(\mu), y(\mu), v(\mu)) \leq \\ &\leq F(x, y) + \mu h(x, y, v) + \tau_1 \quad \forall (x, y, v) \in D. \end{aligned}$$

В частности,

$$F(x(\mu), y(\mu)) \leq F(x, y) + \tau_1 \quad \forall (x, y, v) \in D : h(x, y, v) = 0. \quad (1.1.8)$$

Выполнение равенства $h(x, y, v) = 0$ при $(x, y, v) \in D$ означает, что точка (x, y, v) является точкой ККТ для задачи $(\mathcal{FP}(x))$, а значит, по теореме 1.1.1, и ее решением, так что (1.1.8) равносильно следующему соотношению:

$$F(x(\mu), y(\mu)) \leq F(x, y) + \tau_1 \quad \forall (x, y) \in X : y \in Y_*(x).$$

Таким образом, пара $(x(\mu), y(\mu))$ является приближенным τ_1 -решением задачи (\mathcal{BP}) в том смысле, что точка $y(\mu)$ есть τ_2 -решение задачи нижнего уровня в (\mathcal{BP}) . #

В итоге, объединяя предложения 1.1.1–1.1.3 и теорему 1.1.2, для приближенного решения задачи (\mathcal{BP}) можно предложить использовать метод направленного перебора по μ , который заключается в поиске (глобального) решения в серии задач $(\mathcal{P}(\mu))$, соответствующих увеличивающимся значениям параметра μ . В том случае, когда при некотором μ функция $h[\mu]$ обращается в 0, мы получаем точное решение задачи (\mathcal{BP}) в силу предложения 1.1.1 и теоремы 1.1.2. Если же функция $h[\mu]$ принимает положительные значения для всех выбранных значений параметра μ , то по предложению 1.1.3 мы получаем некоторое приближенное решение задачи (\mathcal{BP}) с известной точностью решения задач верхнего и нижнего уровня.

Поскольку целевая функция задачи $(\mathcal{P}(\mu))$ является д.с. функцией, в качестве приближенного метода решения невыпуклой задачи $(\mathcal{P}(\mu))$ при фиксированном μ предлагается использовать стратегию глобального поиска для задач д.с. оптимизации, разработанную в [42, 43, 119]. Одним из основных элементов этой стратегии является локальный поиск, теоретическому описанию которого посвящен следующий параграф.

1.2 Локальный поиск

Перепишем задачу $(\mathcal{P}(\mu))$ при фиксированном значении параметра $\mu > 0$ в более детальном виде:

$$\left. \begin{aligned} \Phi(x, y, v) &\triangleq \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle + \\ &\quad + \mu(\langle d, y \rangle - \langle A_1x - b, v \rangle) \downarrow \min_{x, y, v}, \\ (x, y) &\in Z \triangleq \{(x, y) \mid Ax + By \leq a, A_1x + B_1y \leq b, x \geq 0, y \geq 0\}, \\ v &\in V = \{v \mid d + vB_1 \geq 0, v \geq 0\}. \end{aligned} \right\} (\mathcal{P}(\mu))$$

Отметим, что в силу предположения (\mathcal{H}) –(1.1.1) и неравенства (1.1.4) целевая функция $\Phi(x, y, v)$ этой задачи является ограниченной снизу на множестве $D \triangleq Z \times V$, поскольку $Z = X \cap \{(x, y) \mid x \in \mathbb{R}^m, y \in Y(x)\}$.

Поскольку, как было отмечено в предыдущем параграфе, задача $(\mathcal{P}(\mu))$ является задачей d.c. минимизации, для локального поиска в задаче $(\mathcal{P}(\mu))$ можно использовать метод локального поиска в задачах d.c. минимизации из [43]. Однако, чтобы учесть билинейную специфику невыпуклости этой задачи при построении алгоритмов локального поиска, предлагается обобщить на случай квадратично-билинейной целевой функции методику последовательного решения задачи по группам переменных, которая была успешно использована ранее в задачах оптимизации с билинейными целевыми функциями (см. [5, 36, 38, 47]).

В данном случае последовательное решение будем проводить по паре (x, y) и по переменной v . Задача $(\mathcal{P}(\mu))$ при фиксированном значении переменной v становится выпуклой квадратичной задачей оптимизации, а при фиксированной паре (x, y) мы получаем задачу линейного программирования (ЛП). Эти вспомогательные задачи являются выпуклыми и для их решения уже разработаны эффективные численные методы (см., например, [2, 6]), к настоящему времени реализованные в виде стандартных пакетов прикладных программ. Таким образом, возникает специальный метод локального поиска, один из вариантов которого можно записать в следующем виде.

Пусть задана некоторая начальная точка $(x_0, y_0, v_0) \in \mathbb{R}^{m+n+q}$, которая, вообще говоря, не является допустимой в задаче $(\mathcal{P}(\mu))$.

V-процедура

Шаг 0. Положить $s := 0$, $v^s := v_0$.

Шаг 1. Некоторым методом квадратичного программирования найти $\frac{\rho_s}{2}$ -решение (x^{s+1}, y^{s+1}) задачи:

$$\left. \begin{aligned} \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle + \mu(\langle d, y \rangle - \langle A_1x, v^s \rangle) \downarrow \min_{(x,y)} \\ (x, y) \in Z, \end{aligned} \right\} \quad (\mathcal{QP}(v^s))$$

так что справедливым будет неравенство

$$\Phi(x^{s+1}, y^{s+1}, v^s) \leq \inf_{(x,y)} \{\Phi(x, y, v^s) \mid (x, y) \in Z\} + \frac{\rho_s}{2}. \quad (1.2.9)$$

Шаг 2. Найти $\frac{\rho_s}{2}$ -решение v^{s+1} задачи ЛП:

$$\langle b - A_1x^{s+1}, v \rangle \downarrow \min_v, \quad v \in V, \quad (\mathcal{LP}(x^{s+1}))$$

так что справедливым будет неравенство

$$\Phi(x^{s+1}, y^{s+1}, v^{s+1}) \leq \inf_v \{\Phi(x^{s+1}, y^{s+1}, v) \mid v \in V\} + \frac{\rho_s}{2}. \quad (1.2.10)$$

Шаг 3. Положить $s := s + 1$ и перейти на шаг 1. #

Отметим, что для начала работы V-процедуры не требуется целиком точка (x_0, y_0, v_0) , достаточно только ее части v_0 , поэтому процедура и носит такое название. Подчеркнем также, что стартовая точка для V-процедуры может быть даже недопустимой.

Справедлива следующая теорема сходимости V-процедуры.

Теорема 1.2.1 *i) При условии $\rho_s > 0, s = 0, 1, 2, \dots, \sum_{s=0}^{\infty} \rho_s < +\infty$, числовая последовательность значений функции $\Phi_s \triangleq \Phi(x^s, y^s, v^s)$, генерируемая V-процедурой, является сходящейся.*

ii) В случае, если $(x^s, y^s, v^s) \rightarrow (\hat{x}, \hat{y}, \hat{v})$, предельная точка $(\hat{x}, \hat{y}, \hat{v})$ удовлетворяет следующим неравенствам:

$$\Phi(\hat{x}, \hat{y}, \hat{v}) \leq \Phi(x, y, \hat{v}) \quad \forall (x, y) \in Z, \quad (1.2.11)$$

$$\Phi(\hat{x}, \hat{y}, \hat{v}) \leq \Phi(\hat{x}, \hat{y}, v) \quad \forall v \in V. \quad (1.2.12)$$

Доказательство. *i)* Введем обозначение: $\bar{\Phi}_s \triangleq \Phi(x^{s+1}, y^{s+1}, v^s)$. Тогда согласно неравенствам (1.2.9) и (1.2.10) справедливы следующие соотношения ($s \geq 1$):

$$\begin{aligned} \Phi_s &\geq \inf_{(x,y)} \{\Phi(x, y, v^s) \mid (x, y) \in Z\} \geq \bar{\Phi}_s - \frac{\rho_s}{2} \geq \\ &\geq \inf_v \{\Phi(x^{s+1}, y^{s+1}, v) \mid v \in V\} - \frac{\rho_s}{2} \geq \Phi_{s+1} - \rho_s. \end{aligned} \quad (1.2.13)$$

Откуда следует $\bar{\Phi}_{s+1} \leq \bar{\Phi}_s + \rho_s$, т.е. последовательность $\{\bar{\Phi}_s\}$, $s = 0, 1, 2, \dots$ является почти монотонно невозрастающей. Кроме того, данная последовательность является ограниченной снизу, поскольку функция $\Phi(\cdot)$ ограничена снизу на $D \triangleq Z \times V$ и $(x^s, y^s, v^s) \in D$, $s = 1, 2, 3, \dots$ по построению. Тогда, с учетом условия, наложенного на ρ_s , заключаем, что последовательность $\{\bar{\Phi}_s\}$ сходится (см. лемму 2.6.2 из [6]).

ii) Согласно шагу 1 V -процедуры имеет место неравенство:

$$\Phi(x^{s+1}, y^{s+1}, v^s) - \frac{\rho_s}{2} \leq \Phi(x, y, v^s) \quad \forall (x, y) \in Z.$$

Переходя здесь к пределу при $s \rightarrow +\infty$ (при этом $\rho^s \downarrow 0$) и при фиксированных (x, y) с учетом непрерывности функции $\Phi(x, y, v)$ получаем неравенство (1.2.11). Аналогично, согласно шагу 2 V -процедуры справедливо неравенство:

$$\Phi(x^{s+1}, y^{s+1}, v^{s+1}) - \frac{\rho_s}{2} \leq \Phi(x^{s+1}, y^{s+1}, v) \quad \forall v \in V.$$

Переходя к пределу, как и выше, получаем неравенство (1.2.12). #

Определение 1.2.1 *Тройку $(\hat{x}, \hat{y}, \hat{v})$, удовлетворяющую неравенствам (1.2.11) и (1.2.12), будем называть критической точкой задачи $(\mathcal{P}(\mu))$.*

Определение 1.2.2 *Тройку $(\hat{x}, \hat{y}, \hat{v})$, удовлетворяющую неравенствам (1.2.11) и (1.2.12) с точностью $\rho \geq 0$, так что:*

$$\begin{aligned} \Phi(\hat{x}, \hat{y}, \hat{v}) - \rho &\leq \Phi(x, y, \hat{v}) \quad \forall (x, y) \in Z, \\ \Phi(\hat{x}, \hat{y}, \hat{v}) - \rho &\leq \Phi(\hat{x}, \hat{y}, v) \quad \forall v \in V, \end{aligned}$$

будем называть приближенно ρ -критической точкой задачи $(\mathcal{P}(\mu))$.

Из неравенств (1.2.11) и (1.2.12) нетрудно видеть, что критическая точка является частично глобальным решением задачи $(\mathcal{P}(\mu))$ по паре (x, y) и по переменной v в отдельности. Остановимся теперь на взаимосвязи критической точки и локального решения в задаче $(\mathcal{P}(\mu))$.

Предложение 1.2.1 Если точка $(\bar{x}, \bar{y}, \bar{v})$ является локальным решением квадратично-билинейной задачи $(\mathcal{P}(\mu))$, то (\bar{x}, \bar{y}) является (глобальным) решением следующей задачи квадратичного программирования:

$$\Phi(x, y, \bar{v}) \downarrow \min_{(x, y)} \quad (x, y) \in Z, \quad (\mathcal{QP}(\bar{v}))$$

а \bar{v} является решением следующей задачи ЛП:

$$\Phi(\bar{x}, \bar{y}, v) \downarrow \min_v \quad v \in V. \quad (\mathcal{LP}(\bar{x}))$$

Другими словами, точка $(\bar{x}, \bar{y}, \bar{v})$ является критической в задаче $(\mathcal{P}(\mu))$ в смысле определения 1.2.1.

Доказательство. Из определения локального решения в задаче $(\mathcal{P}(\mu))$ при $v = \bar{v}$, в частности, следует, что

$$\Phi(\bar{x}, \bar{y}, \bar{v}) \leq \Phi(x, y, \bar{v}) \quad \forall (x, y) \in Z : \|(\bar{x} - x, \bar{y} - y)\| \leq \delta, \delta > 0.$$

Это означает, что (\bar{x}, \bar{y}) является локальным решением выпуклой задачи квадратичного программирования $(\mathcal{QP}(\bar{v}))$, а значит и ее глобальным решением.

Аналогично показывается, что \bar{v} является решением задачи ЛП $(\mathcal{LP}(\bar{x}))$. #

Дополнительно укажем связь критической точки в смысле определения 1.2.1 с классической теорией экстремума.

Предложение 1.2.2 Пусть тройка $(\hat{x}, \hat{y}, \hat{v})$ является критической точкой в задаче $(\mathcal{P}(\mu))$. Тогда в этой точке выполняется необходимое условие оптимальности первого порядка [6, 39] для задачи $(\mathcal{P}(\mu))$:

$$\langle \nabla \Phi(\hat{x}, \hat{y}, \hat{v}), (x, y, v) - (\hat{x}, \hat{y}, \hat{v}) \rangle \geq 0, \quad \forall (x, y, v) \in D. \quad (1.2.14)$$

Доказательство. По определению 1.2.1 критической точки пара (\hat{x}, \hat{y}) является решением квадратичной задачи минимизации $(\mathcal{QP}(\hat{v}))$, а точка \hat{v} — решением задачи ЛП $(\mathcal{LP}(\hat{x}, \hat{y}))$. Следовательно [6, 39], выполняется необходимое условие оптимальности для задачи $(\mathcal{QP}(\hat{v}))$ в точке (\hat{x}, \hat{y}) :

$$\langle \nabla_{x, y} \Phi(\hat{x}, \hat{y}, \hat{v}), (x, y) - (\hat{x}, \hat{y}) \rangle \geq 0, \quad \forall (x, y) \in Z, \quad (1.2.15)$$

и выполняется необходимое условие оптимальности для задачи $(\mathcal{LP}(\hat{x}, \hat{y}))$ в точке \hat{v} :

$$\langle \nabla_v \Phi(\hat{x}, \hat{y}, \hat{v}), v - \hat{v} \rangle \geq 0, \quad \forall v \in V. \quad (1.2.16)$$

Суммируя неравенства (1.2.15), (1.2.16) и учитывая, что $D = Z \times V$, получаем требуемое неравенство (1.2.14). #

Таким образом, определение критической точки 1.2.1 оказывается более сильным по сравнению с классическим необходимым условием оптимальности.

Далее исследуем важный с практической точки зрения вопрос о критериях останова предложенного метода (см. также [36, 47]).

Вычтем из всех членов цепочки неравенств (1.2.13) величину Φ_s :

$$\begin{aligned} 0 &\geq \inf_{(x,y) \in Z} \Phi(x, y, v^s) - \Phi_s \geq \bar{\Phi}_s - \Phi_s - \frac{\rho_s}{2} \geq \\ &\geq \inf_{v \in V} \Phi(x^{s+1}, y^{s+1}, v) - \Phi_s - \frac{\rho_s}{2} \geq \Phi_{s+1} - \Phi_s - \rho_s. \end{aligned} \quad (1.2.17)$$

Можно видеть, что если для некоторой итерации s и $\tau_1 > 0$ справедливо условие

$$\Phi_s - \bar{\Phi}_s \leq \tau_1, \quad (1.2.18)$$

то справедливо неравенство

$$\inf_{(x,y) \in Z} \Phi(x, y, v^s) - \Phi_s \geq -\tau_1 - \frac{\rho_s}{2}$$

или, с учетом обозначения Φ_s , неравенство

$$\Phi(x^s, y^s, v^s) - \left(\tau_1 + \frac{\rho_s}{2} \right) \leq \inf_{(x,y) \in Z} \Phi(x, y, v^s), \quad (1.2.19)$$

которое означает, что (x^s, y^s, v^s) является частично глобальным $\left(\tau_1 + \frac{\rho_s}{2} \right)$ -решением задачи $(\mathcal{P}(\mu))$ по (x, y) .

С другой стороны, если записать правое неравенство цепочки (1.2.13) в виде:

$$\inf_{v \in V} \Phi(x^s, y^s, v) + \frac{\rho_{s-1}}{2} \geq \Phi_s,$$

то, используя первое и второе неравенства цепочки (1.2.13), можно продолжить это неравенство следующим образом:

$$\inf_{v \in V} \Phi(x^s, y^s, v) + \frac{\rho_{s-1}}{2} \geq \Phi_s \geq \bar{\Phi}_s - \frac{\rho_s}{2}.$$

После вычитания отсюда Φ_s получаем цепочку, аналогичную (1.2.17):

$$\inf_{v \in V} \Phi(x^s, y^s, v) - \Phi_s + \frac{\rho_{s-1}}{2} \geq 0 \geq \bar{\Phi}_s - \Phi_s - \frac{\rho_s}{2}.$$

Если, как и ранее, выполнено условие (1.2.18), то справедливо неравенство

$$\inf_{v \in V} \Phi(x^s, y^s, v) - \Phi_s + \frac{\rho_{s-1}}{2} \geq -\tau_1 - \frac{\rho_s}{2},$$

или, что то же самое, неравенство

$$\Phi(x^s, y^s, v^s) - \left(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2} \right) \leq \inf_{v \in V} \Phi(x^s, y^s, v). \quad (1.2.20)$$

Последнее означает, что по переменной v тройка (x^s, y^s, v^s) является частично глобальным $\left(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2} \right)$ -решением задачи $(\mathcal{P}(\mu))$.

Из всего вышесказанного можно сделать вывод, что условие (1.2.18) может быть использовано в качестве критерия останова V -процедуры (после шага 2), поскольку получаемая в этом случае на шаге s точка (x^s, y^s, v^s) является приближенно критической точкой задачи $(\mathcal{P}(\mu))$. Различие точностей по разным переменным продиктовано последовательной структурой метода. Отметим, что если требуется отыскать приближенно τ -критическую точку в задаче $(\mathcal{P}(\mu))$, следует выбирать, например, $\tau_1 < \frac{\tau}{2}$, $\rho_{s-1} < \frac{\tau}{4}$, $\rho_s < \frac{\tau}{4}$.

Заметим, что другим кандидатом на критерий останова, следующим из цепочки (1.2.17), является условие

$$\Phi_s - \Phi_{s+1} \leq \tau_2. \quad (1.2.21)$$

Здесь обращает на себя внимание тот факт, что как только выполняется это условие, согласно цепочке (1.2.17) справедливым оказывается и неравенство

$$\Phi_s - \bar{\Phi}_s \leq \tau_2 + \frac{\rho_s}{2}.$$

Это означает, что условие (1.2.18) выполнено с точностью $\tau_1 = \tau_2 + \frac{\rho_s}{2}$. Таким образом, можно заключить, что неравенство (1.2.21) также может быть использовано в качестве критерия останова V -процедуры. При этом неравенства (1.2.19) и (1.2.20) выполнены с другой точностью (грубее на $\frac{\rho_s}{2}$). Поэтому применение критерия останова (1.2.18) в численных экспериментах представляется более предпочтительным, хотя, с другой стороны, критерий (1.2.21) может выполниться раньше.

Теперь вычтем из всех частей цепочки (1.2.13) величину $\bar{\Phi}_s - \frac{\rho_s}{2}$:

$$\begin{aligned} \Phi_s - \bar{\Phi}_s + \frac{\rho_s}{2} &\geq \inf_{(x,y) \in Z} \Phi(x, y, v^s) - \bar{\Phi}_s + \frac{\rho_s}{2} \geq 0 \geq \\ &\geq \inf_{v \in V} \Phi(x^{s+1}, y^{s+1}, v) - \bar{\Phi}_s \geq \Phi_{s+1} - \bar{\Phi}_s - \frac{\rho_s}{2}. \end{aligned} \quad (1.2.22)$$

Получившаяся цепочка подсказывает критерий останова

$$\bar{\Phi}_s - \Phi_{s+1} \leq \tau_3. \quad (1.2.23)$$

При выполнении этого условия, из (1.2.22) следует:

$$\inf_{(x,y) \in Z} \Phi(x, y, v^s) - \bar{\Phi}_s + \frac{\rho_s}{2} \geq -\tau_3 - \frac{\rho_s}{2},$$

или

$$\inf_{(x,y) \in Z} \Phi(x, y, v^s) + \tau_3 + \rho_s \geq \Phi(x^{s+1}, y^{s+1}, v^s),$$

т.е. (x^{s+1}, y^{s+1}, v^s) является частично глобальным $(\tau_3 + \rho_s)$ -решением по (x, y) .

С другой стороны, согласно (1.2.22),

$$\inf_{v \in V} \Phi(x^{s+1}, y^{s+1}, v) - \bar{\Phi}_s \geq -\tau_3 - \frac{\rho_s}{2}.$$

Откуда также получаем, что (x^{s+1}, y^{s+1}, v^s) является частично глобальным $(\tau_3 + \frac{\rho_s}{2})$ -решением по v .

Итак, можно сделать вывод, что условие (1.2.23) также может быть использовано в качестве критерия останова V -процедуры. При этом заметим, что приближенно критической точкой здесь оказывается тройка (x^{s+1}, y^{s+1}, v^s) , что вполне естественно, поскольку (x^s, y^s, v^s) не входит в формулировку критерия останова (1.2.23). Дополнительно отметим, что “расстояние” между частичными глобальными решениями здесь равно $\frac{\rho_s}{2}$. Так же, как и ранее, “несимметричность” метода порождает несимметричность результата.

Сравнивая данный критерий по точности получаемой критической точки с лучшим из двух предыдущих — (1.2.18), заметим, что при $\tau_1 = \tau_3$, по переменной (x, y) более точным оказывается решение, полученное с использованием критерия останова (1.2.18), а по переменной v — более точным будет решение, полученное с использованием критерия останова (1.2.23).

Поэтому при численной реализации описанного варианта метода локального поиска можно использовать оба этих критерия.

Таким образом, доказано

Предложение 1.2.3 *i)* В том случае, когда на некоторой итерации s V -процедуры после шага 2 оказывается выполненным условие (1.2.18), тройка (x^s, y^s, v^s) является приближенно $\left(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2}\right)$ -критической точкой в задаче $(\mathcal{P}(\mu))$ (более точно, (x^s, y^s, v^s) является для задачи $(\mathcal{P}(\mu))$ частично глобальным $\left(\tau_1 + \frac{\rho_s}{2}\right)$ -решением по (x, y) и частично глобальным $\left(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2}\right)$ -решением по переменной v).

ii) Если же после шага 2 V -процедуры оказывается выполненным условие (1.2.23), то тройка (x^{s+1}, y^{s+1}, v^s) оказывается приближенно $(\tau_3 + \rho_s)$ -критической точкой в задаче $(\mathcal{P}(\mu))$ (является для задачи $(\mathcal{P}(\mu))$ частично глобальным $(\tau_3 + \rho_s)$ -решением по (x, y) и частично глобальным $\left(\tau_3 + \frac{\rho_s}{2}\right)$ -решением по v).

Для осуществления локального поиска в задаче $(\mathcal{P}(\mu))$, согласно логике из [36, 47], можно предложить и другой вариант реализации, “симметричный” вышеописанному. В нем вспомогательные задачи решаются в другом порядке и для начала работы алгоритма из стартовой точки (x_0, y_0, v_0) требуется не компонента v_0 , а пара (x_0, y_0) .

XY-процедура

Шаг 0. Положить $s := 0$, $(x^s, y^s) := (x_0, y_0)$.

Шаг 1. Найти $\frac{\rho_s}{2}$ -решение v^{s+1} задачи линейного программирования:

$$\Phi(x^s, y^s, v) \downarrow \min_v, \quad v \in V, \quad (\mathcal{LP}(x^s))$$

так что справедливым будет неравенство

$$\Phi(x^s, y^s, v^{s+1}) \leq \inf_v \{\Phi(x^s, y^s, v) \mid v \in V\} + \frac{\rho_s}{2}. \quad (1.2.24)$$

Шаг 2. Найти $\frac{\rho_s}{2}$ -решение (x^{s+1}, y^{s+1}) задачи квадратичного программирования:

$$\Phi(x, y, v^{s+1}) \downarrow \min_{(x,y)}, \quad (x, y) \in Z, \quad (\mathcal{QP}(v^{s+1}))$$

так что справедливым будет неравенство

$$\Phi(x^{s+1}, y^{s+1}, v^{s+1}) \leq \inf_{(x,y)} \{\Phi(x, y, v^{s+1}) \mid (x, y) \in Z\} + \frac{\rho_s}{2}. \quad (1.2.25)$$

Шаг 3. Положить $s := s + 1$ и перейти на шаг 1. #

Для XY-версии локального поиска справедлива теорема сходимости, аналогичная теореме 1.2.1.

Теорема 1.2.2 *i)* При условии $\rho_s > 0, s = 0, 1, 2, \dots, \sum_{s=0}^{\infty} \rho_s < +\infty$, числовая последовательность значений функции $\Phi_s \triangleq \Phi(x^s, y^s, v^s)$, генерируемая XY-процедурой, является сходящейся.

ii) В случае, если $(x^s, y^s, v^s) \rightarrow (\hat{x}, \hat{y}, \hat{v})$, предельная точка $(\hat{x}, \hat{y}, \hat{v})$ является критической точкой в задаче $(\mathcal{P}(\mu))$ в смысле определения 1.2.1.

Кроме того, несложно получить и обосновать критерии останова, родственные (1.2.18) и (1.2.23).

Предложение 1.2.4 *i)* При останове XY-процедуры (после шага 2) на итерации s по критерию

$$\Phi_s - \tilde{\Phi}_s \leq \tau_1, \quad (1.2.26)$$

где $\tilde{\Phi}_s \triangleq \Phi(x^s, y^s, v^{s+1})$, полученная точка (x^s, y^s, v^s) является для задачи $(\mathcal{P}(\mu))$ частично глобальным $(\tau_1 + \frac{\rho_s}{2})$ -решением по v и $(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2})$ -решением по (x, y) .

ii) При останове XY-процедуры по критерию

$$\tilde{\Phi}_s - \Phi_{s+1} \leq \tau_2, \quad (1.2.27)$$

тройка (x^s, y^s, v^{s+1}) является для задачи $(\mathcal{P}(\mu))$ частично глобальным $(\tau_2 + \rho_s)$ -решением по v и $(\tau_2 + \frac{\rho_s}{2})$ -решением по (x, y) .

Так же, как и в случае с V -процедурой, при счете можно использовать любой из двух указанных критериев останова.

В целом, отметим, что оба варианта алгоритма локального поиска сходятся к точке с одинаковыми свойствами, но точки, получаемые с использованием критериев останова, оказываются приближенно критическими с разной точностью (по разным группам переменных).

Перейдем теперь к тестированию разработанного метода локального поиска.

1.3 Тестирование процедур локального поиска

Одним из важнейших вопросов, возникающих при тестировании новых численных методов, является выбор или построение тестовых примеров. В данном случае была использована методика генерации тестовых двухуровневых задач, предложенная в [63, 64].

Идея такой генерации заключается в построении двухуровневых задач произвольной размерности с помощью так называемых задач-ядер — одномерных двухуровневых задач с известными локальными и глобальными решениями (см. также [36]).

При генерации тестовых двухуровневых задач класса (\mathcal{BP}) были использованы задачи-ядра, сконструированные при помощи линейно-линейных задач-ядер из [63] и квадратично-квадратичных задач-ядер из [64], решения задач нижнего уровня в которых единственны (при фиксированном значении переменной верхнего уровня x):

$$\left. \begin{aligned} (x-3)^2 + y^2 \downarrow \min_{x,y}, \\ 1 \leq x \leq 3, \quad y \in Y_*(x) \triangleq \text{Sol}(1.3.29), \end{aligned} \right\} \quad (1.3.28)$$

$$\left. \begin{aligned} -y \downarrow \min_y, \\ -2x + y \leq 0, \quad y \geq 0, \quad x + y \leq t, \end{aligned} \right\} \quad (1.3.29)$$

где t — параметр, от значения которого зависит число локальных и глобальных решений в задаче. Справедливо следующее утверждение.

Предложение 1.3.1 [63, 64] *i) При $t = t_1 = 5$ задача (1.3.28)–(1.3.29) имеет два локальных решения $(\check{x}, \check{y}) = (1, 2)$ и $(\bar{x}, \bar{y}) = (3, 2)$, одно из которых — (\bar{x}, \bar{y}) — является ее глобальным решением.*

ii) При $t = t_2 = 3 + 2\sqrt{2}$ задача имеет два глобальных решения в точках $(\check{x}, \check{y}) = (1, 2)$ и $(\hat{x}, \hat{y}) = (3, 2\sqrt{2})$ и не имеет локальных решений, не являющихся глобальными.

iii) При $t = t_3 = 9$ задача имеет единственное локальное решение в точке $(\check{x}, \check{y}) = (1, 2)$, которое является ее глобальным решением.

Далее, согласно схеме генерации, производится объединение произвольного количества задач-ядер разных типов в одну большую задачу

$$\left. \begin{aligned} \sum_{i=1}^r ((x_i - 3)^2 + y_i^2) \downarrow \min_{x,y}, \\ 1 \leq x_i \leq 3, \quad i = 1, \dots, r, \\ y \in \underset{z}{\text{Argmin}} \left\{ -\sum_{i=1}^r z_i \mid z \in Y(x) \right\}, \\ Y(x) \triangleq \{y \in \mathbb{R}^r \mid -2x_i + y_i \leq 0, \quad y_i \geq 0, \quad x_i + y_i \leq \hat{t}_i, \quad i = 1, \dots, r\}, \end{aligned} \right\} \quad (SP)$$

где $\hat{t}_i \in \{t_1, t_2, t_3\}$, $i = 1, \dots, r$.

Затем, чтобы избавиться от сепарабельности, производится подстановка (невыврожденное преобразование координат) в задаче (\mathcal{SP}):

$$x = M_x z, \quad y = M_y u, \quad \det M_x \neq 0, \quad \det M_y \neq 0,$$

где матрицы $M_x, M_y \in \mathbb{R}^{r \times r}$ строятся с использованием генератора псевдослучайных чисел [36, 63, 64]. Несмотря на то, что второй и третий типы задач оказываются выпуклыми, их композиция с первым типом придает разнообразие генерируемым задачам. В результате получается двухуровневая задача класса (\mathcal{BP}), имеющая переменные верхнего и нижнего уровня одинаковой размерности, с известными локальными и глобальными решениями, которые вычисляются с использованием предложения 1.3.1.

Двухуровневые задачи с одинаковым (или близким к этому) количеством переменных на верхнем и нижнем уровне считаются специалистами [60, 74] наиболее сложными для численного решения. При использовании r задач-ядер количество переменных и ограничений в полученной задаче определяется следующим образом: $m = n = r$, $p = 2r$, $q = 3r$, где m — размерность вектора x , n — размерность вектора y , p — количество ограничений в задаче верхнего уровня, q — количество ограничений в задаче нижнего уровня. Следующее предложение говорит о сложности задач, генерируемых с помощью представленной выше методики.

Предложение 1.3.2 [63, 64] *Задача (\mathcal{BP}), полученная объединением r_1 задач-ядер вида (1.3.28)–(1.3.29), соответствующих значению $t_1 = 5$, r_2 — соответствующих значению $t_2 = 3 + 2\sqrt{2}$ и r_3 — соответствующих значению $t_3 = 9$ ($r_1 + r_2 + r_3 = r$), имеет $2^{r_1+r_2}$ локальных решений, 2^{r_2} из которых являются ее глобальными решениями.*

Теперь перейдем к описанию численного тестирования локального поиска на сериях задач, сгенерированных вышеописанным методом.

При проведении вычислительного эксперимента на языке программирования C++ были написаны программы, реализующие XY- и V-процедуры. Для решения вспомогательных задач квадратичного программирования был реализован известный метод особых точек, относящийся к классу методов направленного перебора граней [14, 39], а для решения вспомогательных задач линейного программирования использовался пакет GLPK [87], реализующий симплекс-метод [4]. Счет производился на компьютере с процессором Intel Core 2 Duo 2.4ГГц. В качестве критериев останова было выбрано

условие (1.2.18) для V -процедуры и (1.2.26) для XU -процедуры. При этом требовалось выполнение этих критериев с точностью $\tau = 10^{-4}$.

На **первом этапе** тестирования локального поиска, XU - и V -процедуры были применены к задачам вида $(\mathcal{P}(\mu))$, соответствующим исходным двухуровневым задачам при различных значениях параметра штрафа: $\mu = 1, 2, 3, 5, 10$. Это было сделано для того, чтобы определиться с выбором достаточно большого начального значения $\mu = \mu_0$ параметра штрафа (см. §1.1) для направленного перебора по μ при глобальном поиске (см. замечание в конце §1.1).

Начальной точкой для запуска алгоритма локального поиска на данном этапе тестирования была выбрана точка $(x_0, y_0, v_0) = (0, 0, \dots, 0)$.

В табл. 1 и 2, где представлены результаты тестирования XU - и V -процедур соответственно, использовались следующие обозначения:

№ — номер тестового примера;

Name — название примера в формате “ $m \times n_k$ ”, где, как и ранее, m — размерность вектора x , n — размерность вектора y , k — номер примера данной размерности;

r_1, r_2, r_3 — количество задач-ядер в тестовой задаче, соответствующих значению $t_1 = 5$, $t_2 = 3 + 2\sqrt{2}$ и $t_3 = 9$ соответственно;

$\Phi_{\mu=1,2,\dots}$ — значение целевой функции задачи $(\mathcal{P}(\mu))$ в найденной приближенно критической точке при соответствующем значении μ ;

Φ^* — значение целевой функции задачи $(\mathcal{P}(\mu))$ в глобальном решении соответствующей двухуровневой задачи.

Время счета оказалось очень небольшим (менее 0.1с) и в таблицы не включено. Также таблицы не содержат количество решенных задач ЛП и задач квадратичного программирования при работе методов локального поиска, поскольку во всех случаях этих задач было решено по 2 (было проведено всего 2 итерации локального поиска, после чего с точностью $\tau = 10^{-4}$ выполнялся критерий останова).

Анализируя результаты тестирования, следует также отметить, что в тех случаях, когда значение целевой функции задачи $(\mathcal{P}(\mu))$ в полученной приближенно критической точке лучше (меньше), чем (оптимальное) значение исходной двухуровневой задачи, найденная точка является недопустимой в этой двухуровневой задаче по экстремальному ограничению. Функция $h(\cdot)$ при этом в приближенно критической точке в нуль не

обращается. Например, при $\mu = 1$ все точки, полученные с помощью XU - и V -процедур, оказались недопустимыми.

№	Name	r_1	r_2	r_3	$\Phi_{\mu=1}$	$\Phi_{\mu=2}$	$\Phi_{\mu=3}$	$\Phi_{\mu=5}$	$\Phi_{\mu=10}$	Φ^*
1	$2 \times 2_1$	1	1	0	-11.50	-4.00	-2.50	-2.00	-2.00	-6.00
2	$2 \times 2_2$	0	1	1	-8.50	-4.00	-2.50	-2.00	-2.00	-2.00
3	$2 \times 2_3$	2	0	0	-11.50	-4.00	-2.50	-2.00	-2.00	-6.00
4	$2 \times 2_4$	0	2	0	-8.50	-4.00	-2.50	-2.00	-2.00	-2.00
5	$2 \times 2_5$	0	0	2	-8.50	-4.00	-2.50	-2.00	-2.00	-2.00
6	$4 \times 4_1$	3	1	0	-26.00	-8.00	-5.00	-4.00	-4.00	-16.00
7	$4 \times 4_2$	1	2	1	-23.00	-8.00	-5.00	-4.00	-4.00	-12.00
8	$4 \times 4_3$	4	0	0	-23.00	-8.00	-5.00	-4.00	-4.00	-12.00
9	$4 \times 4_4$	0	1	3	-20.00	-8.00	-5.00	-4.00	-4.00	-8.00
10	$4 \times 4_5$	2	1	1	-20.00	-8.00	-5.00	-4.00	-4.00	-8.00
11	$6 \times 6_1$	2	2	2	-43.50	-12.00	-7.50	-6.00	-6.00	-30.00
12	$6 \times 6_2$	0	0	6	-34.50	-12.00	-7.50	-6.00	-6.00	-18.00
13	$6 \times 6_3$	0	6	0	-25.50	-12.00	-7.50	-6.00	-6.00	-6.00
14	$6 \times 6_4$	6	0	0	-31.50	-12.00	-7.50	-6.00	-6.00	-14.00
15	$6 \times 6_5$	0	3	3	-34.50	-12.00	-7.50	-6.00	-6.00	-18.00
16	$10 \times 10_1$	7	1	2	-66.50	-20.00	-12.50	-10.00	-10.00	-42.00
17	$10 \times 10_2$	0	10	0	-63.50	-20.00	-12.50	-10.00	-10.00	-38.00
18	$10 \times 10_3$	10	0	0	-57.50	-20.00	-12.50	-10.00	-10.00	-30.00
19	$10 \times 10_4$	0	0	10	-51.50	-20.00	-12.50	-10.00	-10.00	-22.00
20	$10 \times 10_5$	3	3	4	-45.50	-20.00	-12.50	-10.00	-10.00	-14.00

Таблица 1: XU -процедура при различных μ

При $\mu = 5$ в некоторых задачах размерности до 6×6 уже локальным поиском удалось найти глобальное решение (XU -процедурой — в задачах №2, 4, 5, 13, а V -процедурой — в №11), причем, как и следовало ожидать в соответствии с развитой выше теорией, дальнейшее увеличение μ на результаты работы локального поиска в этих задачах не повлияло.

Отметим также более стабильную работу XU -процедуры на данном спектре тестовых примеров в смысле получаемых значений целевой функции в критических точках при увеличении значений параметра μ . Это дает основание для выбора начального значения μ , например, $\mu = 10$ при использовании XU -процедуры локального поиска.

№	Name	r_1	r_2	r_3	$\Phi_{\mu=1}$	$\Phi_{\mu=2}$	$\Phi_{\mu=3}$	$\Phi_{\mu=5}$	$\Phi_{\mu=10}$	Φ^*
1	$2 \times 2_1$	1	1	0	-11.50	-4.00	-2.50	-0.25	2.00	-6.00
2	$2 \times 2_2$	0	1	1	-8.50	-4.00	-2.50	9.50	14.00	-2.00
3	$2 \times 2_3$	2	0	0	-11.50	-4.00	-2.50	9.75	21.00	-6.00
4	$2 \times 2_4$	0	2	0	-8.50	-4.00	-2.50	19.50	33.00	-2.00
5	$2 \times 2_5$	0	0	2	-8.50	-4.00	-2.50	29.50	52.00	-2.00
6	$4 \times 4_1$	3	1	0	-26.00	-8.00	-5.00	-10.25	-8.00	-16.00
7	$4 \times 4_2$	1	2	1	-23.00	-8.00	-5.00	-0.50	4.00	-12.00
8	$4 \times 4_3$	4	0	0	-23.00	-8.00	-5.00	9.50	23.00	-12.00
9	$4 \times 4_4$	0	1	3	-20.00	-8.00	-5.00	19.25	35.00	-8.00
10	$4 \times 4_5$	2	1	1	-20.00	-8.00	-5.00	29.25	54.00	-8.00
11	$6 \times 6_1$	2	2	2	-43.50	-12.00	-7.50	-30.00	-30.00	-30.00
12	$6 \times 6_2$	0	0	6	-34.50	-12.00	-7.50	-0.75	6.00	-18.00
13	$6 \times 6_3$	0	6	0	-25.50	-12.00	-7.50	28.50	42.00	-6.00
14	$6 \times 6_4$	6	0	0	-31.50	-12.00	-7.50	29.00	56.00	-14.00
15	$6 \times 6_5$	0	3	3	-34.50	-12.00	-7.50	29.25	63.00	-18.00
16	$10 \times 10_1$	7	1	2	-66.50	-20.00	-12.50	-20.50	-7.00	-42.00
17	$10 \times 10_2$	0	10	0	-63.50	-20.00	-12.50	-0.75	24.00	-38.00
18	$10 \times 10_3$	10	0	0	-57.50	-20.00	-12.50	28.75	67.00	-30.00
19	$10 \times 10_4$	0	0	10	-51.50	-20.00	-12.50	58.25	110.00	-22.00
20	$10 \times 10_5$	3	3	4	-45.50	-20.00	-12.50	77.75	134.00	-14.00

Таблица 2: V -процедура при различных μ

На **втором этапе** тестирования локального поиска для тех же тестовых задач было исследовано поведение XU - и V -процедур локального поиска в задаче ($\mathcal{P}(\mu)$) при фиксированном $\mu = 10$ в зависимости от выбора стартовой точки. Для этого были выбраны следующие точки:

$$(x_0^{(1)}, y_0^{(1)}, v_0^{(1)}) = (0, 0, \dots, 0), \quad (x_0^{(2)}, y_0^{(2)}, v_0^{(2)}) = (3, 3, \dots, 3),$$

$$(x_0^{(3)}, y_0^{(3)}, v_0^{(3)}) = (3, 0, 3, 0, \dots, 3, 0).$$

В табл. 3 использованы следующие новые обозначения:

$\Phi_X^{(1)}, \Phi_X^{(2)}, \Phi_X^{(3)}$ — значение целевой функции задачи ($\mathcal{P}(\mu)$) в найденной с помощью XU -процедуры приближенно критической точке, начиная из точек $(x_0^{(1)}, y_0^{(1)}, v_0^{(1)})$, $(x_0^{(2)}, y_0^{(2)}, v_0^{(2)})$ или $(x_0^{(3)}, y_0^{(3)}, v_0^{(3)})$ соответственно;

$\Phi_V^{(1)}, \Phi_V^{(2)}, \Phi_V^{(3)}$ — аналогичные значения, полученные при использовании V -процедуры.

№	Name	r_1	r_2	r_3	$\Phi_X^{(1)}$	$\Phi_X^{(2)}$	$\Phi_X^{(3)}$	$\Phi_V^{(1)}$	$\Phi_V^{(2)}$	$\Phi_V^{(3)}$	Φ^*
1	2 × 2_1	1	1	0	-2.00	2.00	6.00	2.00	-2.00	-6.00	-6.00
2	2 × 2_2	0	1	1	-2.00	14.00	6.00	14.00	-2.00	6.00	-2.00
3	2 × 2_3	2	0	0	-2.00	22.00	25.00	22.00	-2.00	-6.00	-6.00
4	2 × 2_4	0	2	0	-2.00	34.00	25.00	34.00	-2.00	6.00	-2.00
5	2 × 2_5	0	0	2	-2.00	54.00	25.00	54.00	-2.00	25.00	-2.00
6	4 × 4_1	3	1	0	-4.00	-8.00	0.00	-8.00	-4.00	-12.00	-16.00
7	4 × 4_2	1	2	1	-4.00	4.00	0.00	4.00	-4.00	0.00	-12.00
8	4 × 4_3	4	0	0	-4.00	24.00	19.00	24.00	-4.00	0.00	-12.00
9	4 × 4_4	0	1	3	-4.00	36.00	31.00	36.00	-4.00	0.00	-8.00
10	4 × 4_5	2	1	1	-4.00	56.00	31.00	56.00	-4.00	19.00	-8.00
11	6 × 6_1	2	2	2	-6.00	-30.00	-18.00	-30.00	-6.00	-18.00	-30.00
12	6 × 6_2	0	0	6	-6.00	6.00	6.00	6.00	-6.00	-6.00	-18.00
13	6 × 6_3	0	6	0	-6.00	42.00	18.00	42.00	-6.00	18.00	-6.00
14	6 × 6_4	6	0	0	-6.00	58.00	25.00	58.00	-6.00	25.00	-14.00
15	6 × 6_5	0	3	3	-6.00	66.00	44.00	66.00	-6.00	13.00	-18.00
16	10 × 10_1	7	1	2	-10.00	-6.00	1.00	-6.00	-10.00	-18.00	-42.00
17	10 × 10_2	0	10	0	-10.00	26.00	13.00	26.00	-10.00	1.00	-38.00
18	10 × 10_3	10	0	0	-10.00	70.00	44.00	70.00	-10.00	13.00	-30.00
19	10 × 10_4	0	0	10	-10.00	114.00	56.00	114.00	-10.00	44.00	-22.00
20	10 × 10_5	3	3	4	-10.00	138.00	68.00	138.00	-10.00	56.00	-14.00

Таблица 3: Локальный поиск из различных начальных точек

По результатам тестирования можно видеть, что при использовании стартовых точек $(x_0^{(1)}, y_0^{(1)}, v_0^{(1)})$ и $(x_0^{(2)}, y_0^{(2)}, v_0^{(2)})$ посредством различных вариантов локального поиска в 5 из 20 задач удается найти глобальное решение. Однако при старте из точки $(x_0^{(3)}, y_0^{(3)}, v_0^{(3)})$ локальный поиск находит глобальное решение лишь в двух тестовых задачах размерности 2×2 . Причем, это удается сделать только с использованием V -процедуры. Таким образом, эта точка представляется подходящей для проверки эффективности глобального поиска.

Отдельно необходимо отметить высокую скорость сходимости XU - и V -процедур на данной серии сгенерированных задач: всего 2 итерации, начиная с любого начального приближения. Похожая ситуация наблюдалась при численном тестировании локального поиска в задачах билинейного программирования [36, 47]. Получаемая при этом в результате срабатывания критерия останова точка непременно является приближенно критической с известной точностью, как было показано в §1.2.

1.4 Алгоритм глобального поиска

Как было показано выше, локальный поиск не обеспечивает, вообще говоря, достижения глобального решения даже в задачах небольших размерностей. Поэтому далее предлагается алгоритм глобального поиска в задачах вида $(\mathcal{P}(\mu))$, который в соответствии с теорией глобального поиска фактически состоит из двух основных этапов: а) локального поиска и б) процедур выхода из полученной локальным поиском критической точки, основанных на условиях глобальной оптимальности [43]. Этот алгоритм базируется на стратегии глобального поиска в задачах д.с. оптимизации [36, 43, 47], поскольку, как уже было отмечено ранее, целевая функция в задаче такого вида является д.с. функцией. В сочетании с направленным перебором значений параметра μ (с его увеличением), предложенным в конце §1.1, рассматриваемый далее алгоритм глобального поиска для задачи $(\mathcal{P}(\mu))$ образует новый метод решения задач класса (\mathcal{BP}) .

Согласно методологии из [43], прежде всего необходимо получить явное д.с. разложение целевой функции задачи. Нетрудно видеть, что целевая функция $\Phi(x, y, v)$ задачи $(\mathcal{P}(\mu))$ может быть представлена в виде разности двух выпуклых функций, например, следующим образом:

$$\Phi(x, y, v) = g(x, y, v) - f(x, y, v), \quad (1.4.30)$$

где $g(x, y, v) = \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle + \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle + \mu\left(\langle v, b \rangle + \frac{1}{4}\|v - A_1x\|^2\right)$,
 $f(x, y, v) = \mu\left(\frac{1}{4}\|v + A_1x\|^2 - \langle y, d \rangle\right)$ — выпуклые функции.

Далее, как говорилось выше, процедуры выхода из критической точки, основаны на условиях глобальной оптимальности (УГО) [42, 47]. Дело в том, что УГО обладают так называемым алгоритмическим (конструктивным) свойством, позволяющим в случае нарушения этих УГО строить допустимую точку, лучшую, чем исследуемая. Действительно, для задачи $(\mathcal{P}(\mu))$ УГО имеют следующий вид.

Теорема 1.4.1 [42, 43, 47] *Если допустимая точка (x_*, y_*, v_*) является (глобальным) решением задачи $(\mathcal{P}(\mu))$, то $\forall(z, u, w, \gamma) \in \mathbb{R}^{m+n+q+1}$:*

$$f(z, u, w) = \gamma - \zeta, \quad \zeta := \Phi(x_*, y_*, v_*), \quad (1.4.31)$$

$$g(z, u, w) \leq \gamma \leq \sup_{(x, y, v)} (g, D), \quad (1.4.32)$$

$$g(x, y, v) - \gamma \geq \langle \nabla f(z, u, w), (x, y, v) - (z, u, w) \rangle \quad \forall(x, y, v) \in D. \quad (1.4.33)$$

Если при некоторых $(\hat{z}, \hat{u}, \hat{w}, \hat{\gamma})$ из (1.4.31) при $\zeta = \zeta_k \triangleq \Phi(x^k, y^k, v^k)$ и $(\hat{x}, \hat{y}, \hat{v}) \in D$ (1.4.33) нарушено:

$$g(\hat{x}, \hat{y}, \hat{v}) < \hat{\gamma} + \langle \nabla f(\hat{z}, \hat{u}, \hat{w}), (\hat{x}, \hat{y}, \hat{v}) - (\hat{z}, \hat{u}, \hat{w}) \rangle,$$

то из выпуклости $f(\cdot)$ вытекает

$$\Phi(\hat{x}, \hat{y}, \hat{v}) = g(\hat{x}, \hat{y}, \hat{v}) - f(\hat{x}, \hat{y}, \hat{v}) < f(\hat{z}, \hat{u}, \hat{w}) + \zeta - f(\hat{z}, \hat{u}, \hat{w}) = \Phi(x_*, y_*, v_*),$$

или $\Phi(\hat{x}, \hat{y}, \hat{v}) < \Phi(x_*, y_*, v_*)$. Так что $(\hat{x}, \hat{y}, \hat{v}) \in D$ лучше по значению целевой функции, чем (x_*, y_*, v_*) . Поэтому, изменяя в (1.4.31) при $\zeta = \zeta_k$ параметры (z, u, w, γ) и решая линейаризованные задачи (см. (1.4.33))

$$g(x, y, v) - \langle \nabla f(z, u, w), (x, y, v) \rangle \downarrow \min_{(x, y, v)} \text{ , } (x, y, v) \in D \quad (1.4.34)$$

(где тройка (z, u, w) не обязательно допустима), получаем целое семейство начальных точек $(x(z, u, w, \gamma), y(z, u, w, \gamma), v(z, u, w, \gamma))$ для запуска процедуры локального поиска. При этом на каждом уровне ζ не обязательно перебирать все возможные (z, u, w, γ) — достаточно получить нарушение (1.4.33) только для одной точки $(\hat{z}, \hat{u}, \hat{w}, \hat{\gamma})$. После этого нужно переходить на новый уровень $(x^{k+1}, y^{k+1}, v^{k+1}) := (\hat{x}, \hat{y}, \hat{v})$, $\zeta_{k+1} := \Phi(x^{k+1}, y^{k+1}, v^{k+1})$, и начинать перебирать параметры сначала.

Отдельно отметим, что процедуры локального поиска, предложенные в §1.2, не требуют допустимости стартовой точки. Поэтому при глобальном поиске с использованием этих процедур решать линейаризованную задачу (1.4.34) не обязательно, можно начинать локальный поиск, стартуя из точки (z, u, w) .

Далее, согласно теории глобального поиска [36, 43, 47], задача $(\mathcal{P}(\mu))$ декомпозируется на несколько более простых задач, что можно представить в виде следующей схемы.

Стратегия глобального поиска

1) Методом локального поиска найти приближенно критическую точку (x^k, y^k, v^k) со значением целевой функции $\zeta_k := \Phi(x^k, y^k, v^k)$.

2) Выбрать число $\gamma \in [\gamma_-, \gamma_+]$, где

$$\gamma_- \triangleq \inf_{(x, y, v)} (g, D), \quad \gamma_+ \triangleq \sup_{(x, y, v)} (g, D).$$

Начальное значение параметра $\gamma = \gamma_0$ можно выбирать равным, например, $g(x^k, y^k, v^k)$ [43, 47].

3) Построить некоторую конечную аппроксимацию

$$\mathcal{A}_k = \{(z^l, u^l, w^l) \mid f(z^l, u^l, w^l) = \gamma - \zeta_k, \quad l = 1, 2, \dots, N\}$$

поверхности уровня выпуклой функции $f(\cdot)$, задающей базовую невыпуклость в задаче,

$$U(\zeta_k) = \{(x, y, v) \mid f(x, y, v) = \gamma - \zeta_k\}.$$

Отметим, что эта аппроксимация может быть построена, например, с использованием некоторого множества направлений [47].

4) Проверить неравенство

$$g(z^l, u^l, w^l) \leq \gamma, \quad l = 1, 2, \dots, N, \quad (1.4.35)$$

следующее из условий глобальной оптимальности (1.4.32). Если неравенство (1.4.35) выполнено, то точка аппроксимации будет использоваться дальше. Если же нет — точка (z^l, u^l, w^l) является бесперспективной с точки зрения улучшения текущей критической точки с ее помощью.

5) Стартуя из точек (z^l, u^l, w^l) , $l = 1, 2, \dots, N$, осуществляется запуск некоторой процедуры локального поиска с целью выйти из текущей приближенно критической точки (x^k, y^k, v^k) .

Если улучшить значение ζ_k с помощью всех точек аппроксимации \mathcal{A}_k не удалось, требуется продолжить одномерный поиск по γ на отрезке $[\gamma_-, \gamma_+]$.

Стратегию глобального поиска для задачи $(\mathcal{P}(\mu))$ можно представить в следующей более алгоритмической форме.

Пусть дана некоторая точка $(x_0, y_0, v_0) \in \mathbb{R}^{m+n+q}$, числовые последовательности $\{\tau_k\}$, $\{\delta_k\}$, $\tau_k, \delta_k > 0$, $k = 0, 1, \dots$, $\tau_k \downarrow 0, \delta_k \downarrow 0$ ($k \rightarrow \infty$), множество направлений

$$Dir = \{(a^l, b^l, c^l) \in \mathbb{R}^{m+n+q} \mid (a^l, b^l, c^l) \neq 0, \quad l = 0, 1, \dots, N\},$$

числа $\gamma_- \triangleq \inf_{(x,y,v)} (g, D)$ и $\gamma_+ \triangleq \sup_{(x,y,v)} (g, D)$ и параметры алгоритма M и ν .

Алгоритм глобального поиска

Шаг 0. Положить $k := 0$, $(\bar{x}^k, \bar{y}^k, \bar{v}^k) := (x_0, y_0, v_0)$, $l := 1$, $\gamma := \gamma_-$, $\Delta\gamma := (\gamma_+ - \gamma_-)/M$.

Шаг 1. Начиная из точки $(\bar{x}^k, \bar{y}^k, \bar{v}^k)$, с помощью XY - или V -процедуры построить τ_k -критическую точку $(x^k, y^k, v^k) \in D$ в задаче $(\mathcal{P}(\mu))$. Положить $\zeta_k := \Phi(x^k, y^k, v^k)$.

Шаг 2. По точке $(a^l, b^l, c^l) \in Dir$ построить точку (z^l, u^l, w^l) из аппроксимации \mathcal{A}_k поверхности уровня функции $f(\cdot)$: $\mathcal{A}_k = \{(z^l, u^l, w^l) \mid f(z^l, u^l, w^l) = \gamma - \zeta_k\}$.

Шаг 3. Если $g(z^l, u^l, w^l) > \gamma + \nu\gamma$, то положить $l := l + 1$ и перейти на шаг 2.

Шаг 4. Начиная с точки (z^l, u^l, w^l) , специальным методом локального поиска построить δ_k -критическую точку $(\hat{x}^l, \hat{y}^l, \hat{v}^l) \in D$ в задаче $(\mathcal{P}(\mu))$.

Шаг 5. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) < \Phi(x^k, y^k, v^k)$, то положить $(\bar{x}^{k+1}, \bar{y}^{k+1}, \bar{v}^{k+1}) := (\hat{x}^l, \hat{y}^l, \hat{v}^l)$, $k := k + 1$, $l := 1$, $\gamma := \gamma_-$ и перейти на шаг 1.

Шаг 6. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l < N$, то положить $l := l + 1$ и вернуться на шаг 2.

Шаг 7. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma < \gamma_+$, то положить $\gamma := \gamma + \Delta\gamma$, $l := 1$ и вернуться на шаг 2.

Шаг 8. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma = \gamma_+$, то стоп; (x^k, y^k, v^k) — полученное решение задачи. #

Замечание 1.4.1 На шаге 3 алгоритма производится проверка точки на пригодность к дальнейшим исследованиям с помощью неравенства (1.4.32). При численной реализации необходимо использовать параметр ν , варьированием которого можно изменять точность выполнения этого неравенства (с целью уменьшить влияние машинных ошибок округления, см. [5, 36, 38, 47]). Различные значения параметра M отвечают за разбиение отрезка $[\gamma_-, \gamma_+]$ на соответствующее количество частей для реализации пассивного одномерного поиска по γ .

Нетрудно видеть, что представленный алгоритм глобального поиска в задаче $(\mathcal{P}(\mu))$ не является алгоритмом в общепринятом смысле, поскольку не все его этапы конкретизированы. Дополнительно необходимо исследовать следующие моменты: выбор параметров алгоритма M и ν ; поиск чисел γ_- и γ_+ ; выбор множества направлений Dir и

построение точек аппроксимации поверхности уровня на основе этого множества (на шаге 2 алгоритма).

Что касается первого вопроса, можно отталкиваться от вариантов, предложенных в [36, 47]: а) $M = 10$, $\nu = 0$; б) $M = 20$, $\nu = 0.05$; в) $M = 100$, $\nu = 0.15$. При этом, если требуется быстро отыскать некоторое приближение к глобальному решению задачи ($\mathcal{P}(\mu)$), можно воспользоваться вариантом а) выбора параметров. При увеличении M и ν (случаи б) и в)) происходит рост точности работы алгоритма, но при этом падает скорость его работы.

Для вычисления интервала одномерного поиска $[\gamma_-, \gamma_+]$ нужно решить две задачи: одну на минимум выпуклой квадратичной функции $g(\cdot)$, другую — на максимум этой же функции. Первая задача может быть решена любым из методов квадратичного программирования [2, 6, 14, 39]. Для решения второй можно использовать известную стратегию глобального поиска для задач выпуклой максимизации [43]. При этом отметим, что вычислительный процесс не требует точного знания этих границ — достаточно иметь сравнительно грубые оценки [43].

Построение аппроксимации поверхности уровня выпуклой функции

Остановимся теперь на построении аппроксимации поверхности уровня функции $f(\cdot)$. С одной стороны, она должна быть достаточно репрезентативной для решения вопроса о том, является ли текущая критическая точка глобальным решением. С другой стороны, если текущая критическая точка не является глобальным оптимумом, эта аппроксимация должна позволять построить точку лучшую, чем исследуемая [43].

Здесь был использован следующий прием построения аппроксимаций [43, 47]. Для заданного множества направлений

$$Dir = \{(a^l, b^l, c^l) \in \mathbb{R}^{m+n+q} \mid (a^l, b^l, c^l) \neq 0, l = 0, 1, \dots, N\},$$

множество \mathcal{A}_k строилось из векторов, коллинеарных векторам из Dir и лежащих на поверхности уровня:

$$(z^l, u^l, w^l) = \lambda_l(a^l, b^l, c^l), \quad f(z^l, u^l, w^l) = \gamma - \zeta, \quad l = 1, \dots, N. \quad (1.4.36)$$

Из вида функции $f(\cdot)$ согласно d.c. разложению (1.4.30) и из (1.4.36) вытекает уравнение для определения λ_l :

$$\frac{\mu}{4} \|c^l + A_1 a^l\|^2 \lambda_l^2 - \mu \langle b^l, d \rangle \lambda_l = \gamma - \zeta.$$

Решив это уравнение, получим

$$\lambda_l = \lambda_l(\zeta, \gamma) = 2 \frac{\mu \langle b^l, d \rangle \pm \sqrt{\mu^2 \langle b^l, d \rangle^2 + \mu(\gamma - \zeta) \|c^l + A_1 a^l\|^2}}{\mu \|c^l + A_1 a^l\|^2}. \quad (1.4.37)$$

Чтобы не увеличивать количества точек в аппроксимации (и, соответственно, объема вычислений при глобальном поиске), при численной реализации использовались только точки вида (1.4.36), где в выражении (1.4.37) для λ_l стоит знак “+”.

Наконец, последний вопрос — выбор множества направлений Dir для аппроксимации, — является проблемой, которую можно решить с использованием опыта решения подобных задач [36, 38, 42, 43, 47, 119]. При этом должна быть максимально учтена информация, заключенная в постановке задачи. В задачах с многогранным допустимым множеством (как и в задаче $(\mathcal{P}(\mu))$), хорошо зарекомендовало себя множество направлений, в котором присутствуют стандартные базисные векторы [36, 38, 47]. Кроме того, неплохие результаты для задач д.с. программирования были получены при использовании дополнительной информации, получаемой в процессе вычислений, в частности, данных о текущей критической точке [36, 47]. На основании вышесказанного, для решения задач вида $(\mathcal{P}(\mu))$ было выбрано следующее множество направлений:

$$Dir1 = \{((x, y) + e^i, v + e^j), ((x, y) - e^i, v - e^j), \quad i = 1, \dots, m + n, \quad j = 1, \dots, q\},$$

где $e^i \in \mathbb{R}^{m+n}$, $e^j \in \mathbb{R}^q$ — векторы евклидова базиса, (x, y, v) — текущая критическая точка. Число точек в аппроксимации, построенной с помощью этого множества, равно $2q(m + n)$, а значит достаточно быстро растет при увеличении размерности. Поэтому дополнительно использовался прием сокращения аппроксимации, предложенный в [36].

Рассмотрим некоторое произвольное множество направлений, состоящее из $q(m + n)$ точек:

$$Dir0 = \{(a^i, b^j) \mid (a^i, b^j) \neq 0, \quad i = 1, \dots, m + n, \quad j = 1, \dots, q\}.$$

Найдем в матрице A_1 , которая задает невыпуклость в целевой функции задачи $(\mathcal{P}(\mu))$, два столбца и две строки с максимальной суммой элементов. Обозначим их через i_1, i_2 и j_1, j_2 соответственно. Тогда новое множество направлений будет иметь следующий вид:

$$Cut(Dir0) = \{(a^{i_1}, b^j), (a^{i_2}, b^j), \quad j = 1, \dots, q; (a^i, b^{j_1}), (a^i, b^{j_2}), \quad i = 1, \dots, m + n\}.$$

Количество точек в аппроксимации, построенной с помощью множества $Cut(Dir0)$, равно $2(q + m + n)$ и, следовательно, растет с увеличением размерности медленнее, чем

количество точек в исходной аппроксимации, порожденной множеством $Dir0$, равное $q(m + n)$.

Итак, все необходимые элементы глобального поиска в задаче $(\mathcal{P}(\mu))$ описаны, перейдем к его численному тестированию.

1.5 Тестирование алгоритма глобального поиска

Вычислительный эксперимент по тестированию разработанного в предыдущем параграфе алгоритма глобального поиска в задаче $(\mathcal{P}(\mu))$ состоял из нескольких этапов. Первый этап заключался в тестировании реализации алгоритма глобального поиска, использующей для локального поиска подпрограммы из §1.3. На втором этапе был значительно расширен спектр тестовых задач. Третий этап тестирования глобального поиска был посвящен попытке увеличить эффективность глобального поиска при решении задач повышенной размерности заменой подпрограмм решения вспомогательных выпуклых задач оптимизации более производительной. На заключительном четвертом этапе тестирования глобального поиска осуществлялось его сравнение с пакетом KNITRO.

На **первом этапе** тестирования глобального поиска, был осуществлен поиск решений в построенной ранее серии задач размерности до 10×10 (см. §1.3). Как было отмечено, практически во всех таких задачах локальный поиск не обеспечивает нахождения глобального решения при использовании стартовой точки $(x_0^{(3)}, y_0^{(3)}, v_0^{(3)})$. Поэтому тестирование глобального поиска проводилось, начиная из этой стартовой точки. Стартовое значение $\mu_0 = 10$ параметра μ также было выбрано в §1.3.

Для глобального поиска с использованием XU -процедуры экспериментально были выбраны следующие значения параметров алгоритма (которые отвечают за скорость и точность его работы): $M = 10$, $\nu = 0.0$, с использованием V -процедуры — $M = 5$, $\nu = 0.15$.

Программа, реализующая алгоритм глобального поиска из §1.4, была написана на языке C++. В качестве подпрограмм, реализующих специальный метод локального поиска, использовались программы XU - и V -процедур, описанные и протестированные в §1.2 и §1.3. Условия вычислительного эксперимента не изменились (см. §1.3).

В табл. 4 используются следующие новые обозначения:

$LocSol$ — количество локальных решений, не являющихся глобальными (дает представление о сложности соответствующей тестовой задачи, см. предложение 1.3.2);

Loc_X, Loc_V — число запусков XY - и V -процедур;

T_X, T_V — время работы программы, реализующей метод глобального поиска с использованием XY - и V -процедуры соответственно;

$\bar{\Phi}$ — значение целевой функции задачи в точке, найденной методом глобального поиска с использованием XY - или V -процедуры;

Φ^* — значение целевой функции задачи ($\mathcal{P}(\mu)$) в глобальном решении соответствующей двухуровневой задачи.

№	Name	LocSol	Loc _X	T _X	Loc _V	T _V	$\bar{\Phi}$	Φ^*
1	2 × 2_1	2	340	0.38	183	0.20	-6.00	-6.00
2	2 × 2_2	0	68	0.09	120	0.14	-2.00	-2.00
3	2 × 2_3	3	346	0.38	183	0.20	-6.00	-6.00
4	2 × 2_4	0	70	0.09	125	0.14	-2.00	-2.00
5	2 × 2_5	0	72	0.10	105	0.11	-2.00	-2.00
6	4 × 4_1	14	1846	3.61	746	1.53	-16.00	-16.00
7	4 × 4_2	4	1502	2.97	1035	2.11	-12.00	-12.00
8	4 × 4_3	15	1521	3.02	1028	2.07	-12.00	-12.00
9	4 × 4_4	0	885	1.84	1001	1.94	-8.00	-8.00
10	4 × 4_5	6	900	1.86	772	1.53	-8.00	-8.00
11	6 × 6_1	12	3377	10.78	1006	3.38	-30.00	-30.00
12	6 × 6_2	0	4212	13.82	1965	7.06	-18.00	-18.00
13	6 × 6_3	0	728	2.70	2437	8.16	-6.00	-6.00
14	6 × 6_4	63	3125	10.43	1990	7.15	-14.00	-14.00
15	6 × 6_5	0	4287	13.98	2638	8.86	-18.00	-18.00
16	10 × 10_1	254	25641	2:34.93	3151	21.99	-42.00	-42.00
17	10 × 10_2	0	25360	2:31.41	4730	32.70	-38.00	-38.00
18	10 × 10_3	1023	22635	2:13.26	9421	1:03.40	-30.00	-30.00
19	10 × 10_4	0	16064	1:34.98	7896	54.78	-22.00	-22.00
20	10 × 10_5	56	6929	42.19	17679	1:58.91	-14.00	-14.00

Таблица 4: **Первый вариант программы глобального поиска (задачи небольших размерностей)**

Анализируя результаты численного эксперимента, следует отметить, что с помощью алгоритма глобального поиска удалось получить глобальное решение во всех тестовых примерах с точностью $\varepsilon = 10^{-4}$. При этом увеличения значения параметра μ не потребовалось — начального значения $\mu = \mu_0 = 10$ оказалось достаточно.

Далее, несмотря на то, что в тестовых двухуровневых задачах количество локальных решений, не являющихся глобальными ($LocSol$), невелико (в некоторых случаях даже равно нулю), для отыскания глобального решения алгоритму требуется значительно большее число запусков процедур локального поиска (Loc_X, Loc_V), которые находят критические точки в задаче ($\mathcal{P}(\mu)$). Это можно объяснить, во-первых, тем фактом, что решаемая невыпуклая задача эквивалентна двухуровневой только в глобальном смысле (т.е. в задаче ($\mathcal{P}(\mu)$) могут существовать локальные решения, не являющиеся таковыми в (\mathcal{BP}), как упоминалось выше) [60, 74], а во-вторых, используемые процедуры локального поиска гарантируют нахождение только критических точек задачи ($\mathcal{P}(\mu)$), которые оказываются слабее, чем локальные решения в задаче ($\mathcal{P}(\mu)$) (см. предложение 1.2.2). Наконец, некоторые из найденных критических точек могут совпадать.

Что касается сравнения эффективности различных вариантов метода локального поиска, на данной серии сгенерированных задач более выгодным оказалось использование V -процедуры. Используя ее в составе глобального поиска, удалось решить за меньшее время 14 задач (лучшее время выделено в таблице жирным шрифтом). В целом, по результатам первого этапа тестирования глобального поиска можно констатировать работоспособность предлагаемой методики решения двухуровневых задач класса (\mathcal{BP}).

Целью **второго этапа** тестирования глобального поиска было увеличение размерности, а также спектра решаемых задач. Для этого были сгенерированы серии тестовых задач размерности от 10×10 до 50×50 . Кроме того, был использован упомянутый в §1.4 прием сокращения аппроксимации поверхности уровня, что позволило значительно уменьшить общее время счета. В качестве метода локального поиска была использована XU -процедура, которая лучше показала себя в ходе проведения предварительного вычислительного эксперимента на задачах размерности, большей чем 10×10 .

На данном этапе был выбран также более “строгий” набор параметров счета: $M = 20$, $\nu = 0.05$ по сравнению с тем, который использовался в ходе первого этапа тестирования. Прочие условия вычислительного эксперимента не изменились.

В табл. 5 дополнительно использованы следующие обозначения:

N — количество задач в серии;

$LocSol_{avg}$ — среднее количество локальных решений, не являющихся глобальными, в задачах серии;

Loc_{avg} (Loc_{max}) — среднее (максимальное) число запусков метода локального поиска при проведении глобального поиска в серии тестовых задач;

St_{avg} (St_{max}) — среднее (максимальное) количество итераций алгоритма глобального поиска;

T_{avg} (T_{max}) — среднее (максимальное) время работы программы, реализующей алгоритм глобального поиска.

$m \times n$	$LocSol_{avg}$	Loc_{avg}	Loc_{max}	St_{avg}	St_{max}	T_{avg}	T_{max}	N
10×10	146.2	882.6	1267	2.7	5	9.35	12.69	1000
20×20	131284.1	1105.0	2831	3.2	4	47.23	2:36.12	100
30×30	$1.34 \cdot 10^8$	1571.1	3784	3.0	4	2:52.25	8:08.48	100
40×40	$1.17 \cdot 10^{11}$	2389.7	6219	2.9	5	10:38.67	34:36.86	10
50×50	$1.20 \cdot 10^{14}$	9536.2	26258	11.7	25	1:35:48.38	5:24:18.35	10

Таблица 5: **Первый вариант программы глобального поиска (задачи средних размерностей)**

По результатам вычислительного эксперимента отметим, что программой, реализующей алгоритм глобального поиска с использованием упомянутого выше метода направленного перебора граней для решения выпуклых квадратичных задач и пакета GLPK для решения задач ЛП, удалось успешно решить все сгенерированные задачи размерности до 50×50 . При этом увеличения значения штрафного параметра μ , как и ранее, не потребовалось.

Однако для данного варианта программы глобального поиска характерно быстрое возрастание времени счета с увеличением размерности задачи, и поэтому решение задач размерности, большей чем 50×50 , за разумное время провести затруднительно. Тем не менее, отметим, что среднее количество различных пройденных критических точек (St_{avg}), как и количество запусков процедуры локального поиска (Loc_{avg} , мера трудоемкости алгоритма глобального поиска), с увеличением размерности задачи растет незначительно по сравнению, например, с количеством локальных решений, не являющихся глобальными ($LocSol_{avg}$). Это дает надежду на значительное улучшение быстродействия алгоритма глобального поиска при замене подпрограмм для решения вспомогательных задач более эффективными.

Этому и посвящен **третий этап** тестирования глобального поиска. Был разработан второй вариант программы, реализующей метод глобального поиска, описанный в §1.4,

где в качестве вспомогательных модулей для решения задач квадратичного и линейного программирования были использованы подпрограммы из пакета Xpress-MP [85], который считается специалистами одним из самых эффективных для решения таких задач. Необходимо отметить, что в силу лицензионных ограничений тестирование второго варианта программы проводилось на компьютере с менее производительным процессором Intel Pentium 4 3.0ГГц.

Для проведения тестирования методом из §1.3 были сгенерированы серии двухуровневых задач размерностей от 10×10 до 150×150 (до размерности 50×50 эти серии включают в себя задачи, решавшиеся на втором этапе тестирования). Как и ранее, использовался прием сокращения аппроксимации поверхности уровня и значения параметров $M = 20$, $\nu = 0.05$.

С помощью второй реализации глобального поиска с использованием ХУ-процедуры не удалось решить некоторое количество сгенерированных задач. Однако дополнительное применение для этих задач глобального поиска с использованием V-процедуры позволило достичь в них глобального решения. Количество таких задач в серии тестовых задач каждой размерности указано в столбце V табл. 6.

$m \times n$	$LocSol_{avg}$	Loc_{avg}	Loc_{max}	St_{avg}	St_{max}	T_{avg}	T_{max}	N	V
10×10	146.2	2012.7	4118	1.9	7	22.27	44.52	1000	34
20×20	131284.1	3436.6	8671	2.1	7	53.80	2:38.23	1000	4
30×30	$1.34 \cdot 10^8$	4601.5	15296	2.0	6	1:29.08	4:54.19	100	0
40×40	$1.17 \cdot 10^{11}$	6485.1	14660	2.1	9	2:34.73	6:06.92	100	0
50×50	$1.20 \cdot 10^{14}$	9352.5	17473	2.1	7	4:20.76	8:53.17	100	2
75×75	$3.78 \cdot 10^{21}$	8050.3	13914	3.0	4	4:52.20	8:21.31	10	0
100×100	$1.27 \cdot 10^{29}$	12263.8	20199	2.8	4	10:25.81	16:59.16	10	0
125×125	$4.27 \cdot 10^{36}$	17704.3	32049	2.7	3	20:59.14	38:33.78	10	0
150×150	$3.93 \cdot 10^{44}$	72245.6	286267	17.9	80	2:09:39.95	8:06:46.38	10	4

Таблица 6: Второй вариант программы глобального поиска (задачи средних и больших размерностей)

Что касается сравнения первого и второго варианта реализации алгоритма глобального поиска, можно отметить, что первый вариант (результаты которого приведены в табл. 5) показывает лучшее быстродействие, чем второй (табл. 6) на задачах размерности до 20×20 . Причиной этому служит тот факт, что подпрограммы Xpress-MP, использованные для решения вспомогательных задач во втором случае, нацелены на решение

задач большой размерности. Начиная же с размерности 30×30 , второй вариант реализации глобального поиска демонстрирует значительное преимущество в быстродействии над первым. Причем, с увеличением размерности это преимущество растет (например, на задачах размерности 50×50 второй вариант программы оказывается быстрее первого в среднем в 22 раза), что позволило решить задачи размерности, большей чем 50×50 .

Отдельно отметим разное количество итераций глобального поиска при использовании отличающихся подпрограмм методов линейного и квадратичного программирования. По-видимому, этот факт связан с тем, что в случае неединственности решения вспомогательной задачи различные подпрограммы выбирают разные решения из множества решений такой задачи.

Нельзя не обратить внимание и на тот факт, что количество различных критических точек, доставляющих улучшение целевой функции задачи, полученных при проведении глобального поиска, в задачах размерности 150×150 достигает достаточно большого числа $St_{max} = 80$. Однако общее число запусков локального поиска при проведении глобального поиска в задачах данной размерности, достигающее в среднем значения 72245.6, несравнимо с гигантским количеством локальных решений, не являющихся глобальными, которое в среднем достигает $3.93 \cdot 10^{44}$.

Таким образом, при использовании более эффективной подпрограммы для решения задач квадратичного и линейного программирования удалось за приемлемое время решить все тестовые задачи размерности до 300 (суммарно по всем переменным), что превышает известные из литературы результаты численного решения подобного сорта задач, отмеченные во введении.

На **четвертом этапе** тестирования глобального поиска было проведено сравнение алгоритма глобального поиска с пакетом KNITRO [134], предназначенным для решения нелинейных задач оптимизации. С этой целью методом генерации тестовых задач из §1.3 была сгенерирована серия из 15 задач размерности от 5×5 до 30×30 , кроме того решались следующие три линейно-линейные двухуровневые задачи, имеющиеся в литературе.

Пример 1 [56].

$$\left. \begin{aligned} & -x - 3y \downarrow \min_{x,y}, \\ & x \geq 0, \\ & y \in \underset{z}{\text{Argmin}}\{3z \mid -x - 2z \leq -10, x - 2z \leq 6, 2x - z \leq 21, \\ & x + 2z \leq 38, -x + 2z \leq 18, z \geq 0\}. \end{aligned} \right\}$$

Пример 2 [69].

$$\left. \begin{aligned} & -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3 \downarrow \min_{x,y}, \\ & x_1, x_2 \geq 0, \\ & y \in \underset{z}{\text{Argmin}}\{z_1 + z_2 + 2z_3 \mid -z_1 + z_2 + z_3 \leq 1, 4x_1 - 2z_1 + 4z_2 - z_3 \leq 2, \\ & 4x_2 + 4z_1 - 2z_2 - z_3 \leq 2, z_1, z_2, z_3 \geq 0\}. \end{aligned} \right\}$$

Пример 3 [89].

$$\left. \begin{aligned} & -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3 \downarrow \min_{x,y}, \\ & x_1 + 2x_2 - y_3 \leq 1.3, x_1, x_2 \geq 0, \\ & y \in \underset{z}{\text{Argmin}}\{2z_1 + z_2 + 2z_3 \mid -z_1 + z_2 + z_3 \leq 1, 4x_1 - 2z_1 + 4z_2 - z_3 \leq 2, \\ & 4x_2 + 4z_1 - 2z_2 - z_3 \leq 2, z_1, z_2, z_3 \geq 0\}. \end{aligned} \right\}$$

При тестировании использовался второй вариант программы глобального поиска (с использованием пакета Xpress-MP для решения вспомогательных выпуклых задач оптимизации) и пакет KNITRO (с выключенным мультистартом при нахождении локального решения и с включенным мультистартом при попытке найти глобальное решение). Поскольку пакет KNITRO не имеет возможности работать с двухуровневыми задачами напрямую, на вход программе из этого пакета подавалась задача вида (\mathcal{P}) , эквивалентная соответствующей двухуровневой задаче (см. §1.1). Использовалось начальное приближение $(0, 0, \dots, 0)$.

Чтобы убедиться в необходимости применения глобального поиска для данной серии тестовых задач, предварительно было осуществлено сравнение пакета KNITRO без мультистарта с процедурами локального поиска, результаты которого представлены в таблице 7, где использовались следующие новые обозначения:

F_K — значение целевой функции верхнего уровня тестовой двухуровневой задачи в точке, найденной пакетом KNITRO;

F_{XY} (F_V) — значение целевой функции верхнего уровня в точке, полученной с помощью XY -процедуры (V -процедуры);

F_* — значение целевой функции задачи в ее известном решении;

T — время работы соответствующей программы.

Name	r_1	r_2	r_3	F_*	KNITRO		XY -процедура		V -процедура	
					F_K	T	F_{XY}	T	F_V	T
1	—	—	—	-49	-15	0.003	-15	0.016	-15	0.016
2	—	—	—	-29.2	-6	0.003	-29.2	0.015	-29.2	0.000
3	—	—	—	-18.4	-5.6	0.010	-5.6	0.016	-5.6	0.016
5x5_1	4	0	1	-21	-21	0.010	-13	0.031	-21	0.016
5x5_2	1	2	2	-9	63	0.008	-1	0.016	7	0.015
5x5_3	0	1	4	-5	115	0.009	3	0.015	3	0.016
10x10_1	7	1	2	-38	-30	0.131	-10	0.016	-30	0.015
10x10_2	4	1	5	-26	122	0.024	-10	0.016	-18	0.016
10x10_3	1	4	5	-14	18.031	0.181	-10	0.015	18	0.016
15x15_1	1	2	12	-19	-3	0.066	-15	0.015	-3	0.015
15x15_2	3	7	5	-27	29	1.082	-15	0.015	29	0.015
15x15_3	7	1	7	-43	161	0.049	-15	0.015	-35	0.015
20x20_1	1	4	15	-24	8	0.276	-20	0.015	8	0.015
20x20_2	7	11	2	-48	40	0.922	-20	0.016	40	0.015
20x20_3	8	6	6	-52	-4	0.294	-20	0.015	-3.998	0.015
30x30_1	28	1	1	-142	-134	0.765	-30	0.016	-134	0.016
30x30_2	7	11	12	-58	30	0.571	-30	0.016	30	0.016
30x30_3	3	7	20	-42	14.029	2.686	-30	0.016	14.002	0.016

Таблица 7: Сравнение процедур локального поиска с KNITRO

Отметим, что запуск программ, реализующих процедуры локального поиска, и программы глобального поиска осуществлялся на компьютере с процессором Intel Core 2 Duo 2.0GHz, а запуск программы из пакета KNITRO осуществлялся на компьютере с более производительным процессором Intel Core 2 Quad 2.8GHz.

Сравнивая результаты работы KNITRO, XY -процедуры и V -процедуры по значениям целевых функций задач в найденных точках, отмечаем, что в большинстве задач меньшее значение целевой функции получено в критических точках, найденных с помощью XY -процедуры. Заметим, что с помощью V -процедуры удалось найти глобальные решения в двух задачах, а с помощью XY -процедуры и KNITRO — в одной из этих задач (эти случаи выделены в таблице жирным шрифтом).

Отдельно отметим преимущество XY - и V -процедур по времени счета по сравнению с пакетом KNITRO для задач размерности большей, чем 5×5 .

Итак, найти глобальные решения в большинстве тестовых задач с помощью алгоритмов локального поиска и пакета KNITRO не удалось, перейдем к описанию тестирования вариантов алгоритма глобального поиска с XY - и V -процедурами локального поиска и программы из пакета KNITRO с мультистартом, результаты которого представлены в таблице 8, где использованы следующие новые обозначения:

F_{Kms} — значение целевой функции верхнего уровня тестовой двухуровневой задачи в точке, найденной пакетом KNITRO с мультистартом;

F_{XY-AGP} (F_{V-AGP}) — значение целевой функции верхнего уровня в точке, полученной с помощью глобального поиска с использованием XY -процедуры (V -процедуры).

Name	F_*	KNITRO multistart		Глобальный поиск			
		F_{Kms}	T	F_{XY-AGP}	T	F_{V-AGP}	T
1	-49	-49.001	0.6	-49	2.2	-49	1.7
2	-29.2	-29.2	0.8	-29.2	4.6	-29.2	1.5
3	-18.4	-18.4	0.5	-18.4	1.5	-18.4	0.8
5x5_1	-21	-21	7.7	-21	11.4	-21	7.3
5x5_2	-9	-9	8.7	-9	11.3	-9	4.7
5x5_3	-5	-5	9.9	-5	11.7	-5	4.6
10x10_1	-38	-30	1:17.2	-38	33.1	-38	23.7
10x10_2	-26	-26	1:17.0	-26	29.2	-26	12.4
10x10_3	-14	-14	1:22.4	-14	24.6	-14	16.0
15x15_1	-19	-19	11:08.5	-19	20.3	-19	19.0
15x15_2	-27	-19	5:43.6	-27	38.6	-27	31.5
15x15_3	-43	-35	7:01.2	-43	48.1	-43	35.5
20x20_1	-24	-24	19:13.7	-24	26.9	-24	45.9
20x20_2	-48	-48	30:20.5	-47.999	1:12.0	-47.999	1:10.9
20x20_3	-52	-32	29:54.4	-52	1:13.1	-52	52.0
30x30_1	-142	-134	1:34:34.9	-141.997	4:30.3	-141.997	1:46.8
30x30_2	-58	-38	1:31:23.5	-58	1:26.2	-58	1:51.3
30x30_3	-42	-29.999	2:39:15.4	-42	51.5	-42	1:08.9

Таблица 8: Сравнение глобального поиска с KNITRO

По результатам вычислительного эксперимента, прежде всего, заметим, что с помощью пакета KNITRO с использованием мультистарта удалось найти глобальные решения только в 61% задач с точностью $\varepsilon = 10^{-2}$, в то время как с помощью обоих вариантов

программы, реализующей алгоритм глобального поиска, удалось решить с данной точностью все тестовые задачи. Также заметим огромное преимущество глобального поиска по сравнению с KNITRO по времени счета для задач размерности большей, чем 5×5 , в случае глобального поиска с использованием XU -процедуры и для задач размерности не меньшей, чем 5×5 , в случае глобального поиска с V -процедурой.

Таким образом, по результатам вычислительного эксперимента можно сделать вывод, что подход к поиску оптимистических решений в квадратично-линейных двухуровневых задачах, основанный на применении стратегии глобального поиска, продемонстрировал подавляющее преимущество перед использованием пакета KNITRO для данной серии тестовых задач.

1.6 Заключительные замечания

Основные результаты, полученные в данной главе, заключаются в следующем. Разработан и протестирован алгоритм поиска оптимистических решений в квадратично-линейных двухуровневых задачах. В частности:

- обоснована редукция рассматриваемой двухуровневой задачи к семейству одноуровневых задач математического программирования;
- предложены и протестированы два варианта специального алгоритма локального поиска в редуцированных задачах, доказана их сходимость к критической точке, отвечающей билинейной структуре невыпуклости таких задач, и обоснованы критерии останова, позволяющие обнаружить приближенно критическую точку;
- разработан алгоритм глобального поиска, который показал свою эффективность при численном тестировании на сериях случайно сгенерированных задач размерности до 300 и продемонстрировал существенное преимущество по времени счета и количеству решенных задач по сравнению с пакетом KNITRO.

Глава 2

Теоретические основы поиска гарантированных решений

Глава 2 посвящена исследованию одного класса квадратично-линейных задач двухуровневой оптимизации в гарантированной (пессимистической) постановке [8, 11, 74]. Ранее в работах [8, 34, 35] была поставлена и исследована задача поиска гарантированного решения следующей двухуровневой задачи (задачи Штакельберга с приближенным решением задачи нижнего уровня):

$$\left. \begin{array}{l} \sup_y \{F(x, y) \mid y \in N(x, \varepsilon)\} \downarrow \min_x, \\ x \in X, \quad N(x, \varepsilon) \triangleq \{y \in Y \mid G(x, y) \leq \inf_z [G(x, z) \mid z \in Y] + \varepsilon\}, \end{array} \right\} \quad (\mathcal{G}(\varepsilon))$$

где $\varepsilon \geq 0$.

В [34, 35] был предложен подход к решению задачи $(\mathcal{G}(0))$ (задачи $(\mathcal{G}(\varepsilon))$ при $\varepsilon = 0$), который заключается в сведении этой задачи к серии двухуровневых задач с оштрафованной целевой функцией нижнего уровня в оптимистической постановке вида

$$\left. \begin{array}{l} F(x, y) \downarrow \min_{x, y}, \quad x \in X, \\ y \in \{y \in Y \mid G(x, y) - \nu F(x, y) \leq \inf_z [G(x, z) - \nu F(x, z) \mid z \in Y] + \delta\}, \end{array} \right\} \quad (\mathcal{G}_o(\delta, \nu))$$

где $\delta \geq 0, \nu > 0$.

Оказывается, имеет место сходимость последовательности значений задач $(\mathcal{G}_o(\delta, \nu))$, соответствующей различным значениям параметров δ, ν , к значению задачи $(\mathcal{G}(0))$.

Теорема 2.0.1 [34] Пусть $F(\cdot)$ и $G(\cdot)$ непрерывны на произведении компактов $X \times Y$ некоторых метрических пространств. Тогда

$$\lim_{\delta, \nu} \{\mathcal{V}(\mathcal{G}_o(\delta, \nu)) \mid \delta \downarrow 0, \nu \downarrow 0, \frac{\delta}{\nu} \rightarrow 0\} = \mathcal{V}(\mathcal{G}(0)).$$

С целью развить данный подход для поиска гарантированного решения квадратично-линейной двухуровневой задачи (которая отличается от задачи $(\mathcal{G}(0))$) тем, в частности, что допустимое множество задачи нижнего уровня зависит от стратегии игрока верхнего уровня, $Y = Y(x)$) в первом параграфе главы 2 исследуются взаимоотношения этой задачи со вспомогательной квадратично-квадратичной двухуровневой задачей в оптимистической постановке. Далее показана непрерывность зависимостей множества решений и (оптимального) значения задачи нижнего уровня от переменных верхнего уровня. Затем, с использованием полученных результатов производится редукция исследуемой двухуровневой задачи в гарантированной постановке к серии вспомогательных (оптимистических) двухуровневых задач.

Кроме того, осуществляется дальнейшая редукция вспомогательной двухуровневой задачи в оптимистической постановке к одноуровневым невыпуклым задачам оптимизации. Наконец, предлагаются два варианта алгоритма локального поиска в редуцированных задачах, а также алгоритм глобального поиска, их использующий.

2.1 Постановка задачи и ее взаимосвязь с задачей поиска оптимистического решения специальной двухуровневой задачи

В диссертационной работе подход из [34, 35] обобщается для решения двухуровневых задач следующего класса:

$$\left. \begin{aligned} W(x, \varepsilon) &\triangleq \sup_y \left\{ \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle - \frac{1}{2} \langle y, C_1 y \rangle + \langle c_1, y \rangle \mid y \in Y_*(x, \varepsilon) \right\} \downarrow \min_x, \\ x \in X, \quad Y_*(x, \varepsilon) &\triangleq \{y \in Y(x) \mid \langle d, y \rangle \leq \inf_z [\langle d, z \rangle \mid z \in Y(x)] + \varepsilon\}, \end{aligned} \right\} (\mathcal{BP}(\varepsilon))$$

так что задача нижнего уровня имеет вид:

$$\langle d, y \rangle \downarrow \min_y, \quad y \in Y(x), \quad (\mathcal{FP}(x))$$

где $X \triangleq \{x \in \mathbb{R}^m \mid Ax \leq a, x \geq 0\}$, $Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1 x + B_1 y \leq b, y \geq 0\}$, $A \in \mathbb{R}^{p \times m}$, $A_1 \in \mathbb{R}^{q \times m}$, $B_1 \in \mathbb{R}^{q \times n}$, $C = C^T \geq 0$, $C_1 = C_1^T \geq 0$, $c \in \mathbb{R}^m$, $c_1, d \in \mathbb{R}^n$, $a \in \mathbb{R}^p$, $b \in \mathbb{R}^q$, а $Y_*(x, \varepsilon)$ — множество ε -решений задачи $(\mathcal{FP}(x))$.

Здесь $F(x, y) \triangleq \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle - \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle$ и $G(y) \triangleq \langle d, y \rangle$ — критерии эффективности стратегий для игроков верхнего и нижнего уровней, так что $W(x, \varepsilon) = \sup_y \{F(x, y) \mid y \in Y_*(x, \varepsilon)\}$ — есть оценка эффективности стратегий игрока верхнего уровня [8, 11].

Наряду с задачей $(\mathcal{BP}) = (\mathcal{BP}(\varepsilon))$ используемый подход [34, 35] подразумевает рассмотрение следующей двухуровневой задачи в оптимистической постановке:

$$\left. \begin{aligned} & F(x, y) \downarrow \min_{x, y}, \quad x \in X, \quad y \in Y_{**}(x, \delta) \triangleq \\ & \triangleq \{y \in Y(x) \mid G(y) - \nu F(x, y) \leq \inf_z [G(z) - \nu F(x, z) \mid z \in Y(x)] + \delta\}, \end{aligned} \right\} (\mathcal{BP}_o(\delta, \nu))$$

где $\delta \geq 0$ — точность решения задачи нижнего уровня, $\nu > 0$ — штрафной множитель, с которым критерий эффективности игрока верхнего уровня исходной задачи штрафует нижний уровень. Здесь, очевидно, задача нижнего уровня имеет вид:

$$G(y) - \nu F(x, y) \downarrow \min_y, \quad y \in Y(x), \quad (\mathcal{FP}_o(x, \nu))$$

а $Y_{**}(x, \delta)$ — это множество δ -решений этой задачи, $Y_{**}(x, \delta) = \delta\text{-Sol}(\mathcal{FP}_o(x, \nu))$.

Поскольку критерий $F(x, y)$ эффективности стратегий игрока верхнего уровня задачи (\mathcal{BP}) является вогнутой по y функцией и $\nu > 0$, целевая функция задачи $(\mathcal{FP}_o(x, \nu))$ со штрафом остается выпуклой по переменной y . Напомним, что выпуклость по y задачи нижнего уровня была использована при поиске оптимистических решений двухуровневых задач в главе 1.

Введем следующие предположения на допустимые множества задач верхнего и нижнего уровня:

$$(\mathcal{H}) : X \text{ — ограниченное множество и } \exists Y : Y \supseteq Y(x) \quad \forall x \in X, \quad Y \text{ — компакт.} \quad (2.1.1)$$

Нетрудно видеть, что выполнение условий (\mathcal{H}) –(2.1.1) влечет выполнение условий:

$$\forall x \in X \quad Y(x) \text{ — компакт,} \quad (2.1.2)$$

$$\exists F_-, F_+ \in \mathbb{R} : F_- \leq F(x, y) \leq F_- + F_+ \quad \forall (x, y) : x \in X, y \in Y(x). \quad (2.1.3)$$

Далее исследуем взаимосвязи задач $(\mathcal{BP}(\varepsilon))$ и $(\mathcal{BP}_o(\delta, \nu))$ с целью свести задачу $(\mathcal{BP}(\varepsilon))$ при $\varepsilon = 0$ (точном решении задачи нижнего уровня) к серии задач $(\mathcal{BP}_o(\delta, \nu))$, соответствующих различным значениям параметров δ, ν , как это было сделано в [34] для задач $(\mathcal{G}(0))$ и $(\mathcal{G}_o(\delta, \nu))$. Для этого сначала докажем взаимосвязь решений задач нижнего уровня в $(\mathcal{BP}(\varepsilon))$ и $(\mathcal{BP}_o(\delta, \nu))$.

Предложение 2.1.1 Пусть выполнено условие (2.1.3), а пара $(\hat{x}, \hat{y}) \in X \times Y$ такая, что \hat{y} является δ -решением задачи нижнего уровня в (оптимистической) задаче $(\mathcal{BP}_o(\delta, \nu))$, т.е. $\hat{y} \in Y_{**}(\hat{x}, \delta)$.

Тогда точка \hat{y} будет $(\delta + \nu F_+)$ -решением задачи нижнего уровня в (пессимистической) задаче $(\mathcal{BP}(\delta + \nu F_+))$, т.е. $\hat{y} \in Y_*(\hat{x}, \delta + \nu F_+)$.

Доказательство. Из условия (2.1.3) следует неравенство $(\forall(x, y) : x \in X, y \in Y(x))$:

$$G(y) - \nu(F_- + F_+) \leq G(y) - \nu F(x, y). \quad (2.1.4)$$

Далее, поскольку $\hat{y} \in Y_{**}(\hat{x}, \delta) = \delta\text{-Sol}(\mathcal{FP}_o(\hat{x}, \nu))$, имеем:

$$G(\hat{y}) - \nu F(\hat{x}, \hat{y}) \leq \inf_y \{G(y) - \nu F(\hat{x}, y) \mid y \in Y(\hat{x})\} + \delta. \quad (2.1.5)$$

Из условия (2.1.3) вытекает также неравенство:

$$\inf_y \{G(y) - \nu F(\hat{x}, y) \mid y \in Y(\hat{x})\} + \delta \leq \inf_y \{G(y) - \nu F_- \mid y \in Y(\hat{x})\} + \delta. \quad (2.1.6)$$

Объединяя неравенства (2.1.4)–(2.1.6), окончательно получаем:

$$G(\hat{y}) \leq \inf_y \{G(y) \mid y \in Y(x)\} + \delta + \nu F_+.$$

Последнее означает, что $\hat{y} \in Y_*(\hat{x}, \delta + \nu F_+)$. #

Следующая лемма о взаимосвязи значений целевых функций задач $(\mathcal{BP}(\varepsilon))$ и $(\mathcal{BP}_o(\delta, \nu))$ обобщает для этих задач лемму из [34].

Лемма 2.1.1 Пусть выполнено условие (2.1.3). Тогда для любого набора $(\varepsilon, \delta, \nu, \hat{x}, \hat{y})$, такого что $\varepsilon \geq 0$, $\delta \geq 0$, $\nu > 0$, $\hat{x} \in X$, $\hat{y} \in Y_{**}(\hat{x}, \delta)$, справедливы следующие оценки:

$$\sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon)\} - \frac{\varepsilon + \delta}{\nu} \leq F(\hat{x}, \hat{y}) \leq \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \delta + \nu F_+)\}. \quad (2.1.7)$$

Доказательство. 1) Поскольку из включения $\hat{y} \in Y_{**}(\hat{x}, \delta)$ в силу предложения 2.1.1 следует включение $\hat{y} \in Y_*(\hat{x}, \delta + \nu F_+)$, то правое неравенство в (2.1.7) становится очевидным.

2) Далее, из включений $\hat{y} \in Y_{**}(\hat{x}, \delta)$ и $Y_*(\hat{x}, \varepsilon) \subseteq Y(\hat{x})$ имеем:

$$G(\hat{y}) - \nu F(\hat{x}, \hat{y}) \leq \inf_y \{G(y) - \nu F(\hat{x}, y) \mid y \in Y(\hat{x})\} + \delta \leq$$

$$\leq \inf_y \{G(y) - \nu F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon)\} + \delta. \quad (2.1.8)$$

С другой стороны, включение $y \in Y_*(\hat{x}, \varepsilon)$ означает, что

$$G(y) \leq \inf_z \{G(z) \mid z \in Y(\hat{x})\} + \varepsilon,$$

откуда следует цепочка неравенств:

$$\begin{aligned} & \inf_y \{G(y) - \nu F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon)\} + \delta \leq \\ & \leq \inf_y \{\inf_z [G(z) \mid z \in Y(\hat{x})] + \varepsilon - \nu F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon)\} + \delta \leq \\ & \leq \inf_z \{G(z) \mid z \in Y(\hat{x})\} - \nu \sup_z \{F(\hat{x}, z) \mid z \in Y_*(\hat{x}, \varepsilon)\} + \varepsilon + \delta \leq \\ & \leq G(\hat{y}) - \nu \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon)\} + \varepsilon + \delta. \end{aligned} \quad (2.1.9)$$

С учетом того, что $\nu > 0$, из (2.1.8) и (2.1.9) следует левое неравенство в (2.1.7). $\#$

Полученные оценки (2.1.7) значений целевых функций задач $(\mathcal{BP}(\varepsilon))$ и $(\mathcal{BP}_o(\delta, \nu))$ позволяют установить следующие взаимосвязи (оптимальных) значений этих двухуровневых задач при различной точности решения задачи нижнего уровня в задаче $(\mathcal{BP}(\varepsilon))$. Обозначим через $\mathcal{V}(\varepsilon) \triangleq \inf_x \{W(x, \varepsilon) \mid x \in X\}$ — значение задачи $(\mathcal{BP}(\varepsilon))$, соответствующее точности решения задачи нижнего уровня ε , а через $\mathcal{V}_o(\delta, \nu) \triangleq \inf_{x, y} \{F(x, y) \mid x \in X, y \in Y_{**}(x, \delta)\}$ — значение задачи $(\mathcal{BP}_o(\delta, \nu))$, которое зависит от точности решения оштрафованной задачи нижнего уровня δ и от параметра штрафа ν .

Следствие 2.1.1 *Значения задач $(\mathcal{BP}(\varepsilon))$ и $(\mathcal{BP}_o(\delta, \nu))$, соответствующие различной точности решения задач нижнего уровня, связаны следующей цепочкой неравенств:*

$$\mathcal{V}(\varepsilon) - \frac{\varepsilon + \delta}{\nu} \leq \mathcal{V}_o(\delta, \nu) \leq \mathcal{V}(\delta + \nu F_+). \quad (2.1.10)$$

Доказательство. Для получения (2.1.10) достаточно последовательно перейти к точной нижней грани по \hat{y} и \hat{x} в (2.1.7). $\#$

Прежде чем исследовать функцию $\mathcal{V}(\varepsilon)$, входящую в цепочку (2.1.10), докажем вспомогательные утверждения.

Отметим, что множества решений задач $(\mathcal{FP}(x))$ и $(\mathcal{FP}_o(x, \nu))$ обладают следующим свойством.

Лемма 2.1.2 Пусть выполнены предположения (\mathcal{H}) –(2.1.1). Тогда при любом фиксированном значении $\hat{x} \in X$ для произвольного $\varepsilon \geq 0$ множества $Y_*(\hat{x}, \varepsilon)$ и $Y_{**}(\hat{x}, \varepsilon)$ компактны.

Доказательство. Пусть \hat{x} — фиксированный элемент из X . Рассмотрим произвольную последовательность $\{y^k\} \subseteq Y_*(\hat{x}, \varepsilon) \subseteq Y(\hat{x})$. В силу (2.1.2) с точностью до подпоследовательности можно считать, что $y^k \rightarrow \hat{y} \in Y(\hat{x})$. Покажем, что $\hat{y} \in Y_*(\hat{x}, \varepsilon)$. Воспользуемся включением $\{y^k\} \subseteq Y_*(\hat{x}, \varepsilon)$:

$$G(y^k) \leq G(y) + \varepsilon \quad \forall y \in Y(\hat{x}).$$

Переходя в этом неравенстве к пределу при $k \rightarrow +\infty$, по непрерывности $G(\cdot)$ получаем:

$$G(\hat{y}) \leq G(y) + \varepsilon \quad \forall y \in Y(\hat{x}),$$

что и означает $\hat{y} \in Y_*(\hat{x}, \varepsilon)$.

Компактность множества $Y_{**}(\hat{x}, \varepsilon)$ доказывается аналогично. #

Теперь рассмотрим значение целевой функции $W(\hat{x}, \varepsilon)$ двухуровневой задачи $(\mathcal{BP}(\varepsilon))$ в некоторой допустимой в этой задаче точке $\hat{x} \in X$ как функцию от ε . Оказывается, эта функция обладает следующим свойством.

Лемма 2.1.3 Пусть выполнены условия леммы 2.1.2 и последовательность $\{\varepsilon_k\}$ монотонно сходится к нулю (будем обозначать этот факт следующим образом: $\varepsilon_k \downarrow 0$).

Тогда последовательность $\{W_k := W(\hat{x}, \varepsilon_k)\}$ оценок эффективности

$$W(\hat{x}, \varepsilon_k) \triangleq \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon_k)\}$$

монотонно сходится к оценке эффективности

$$W(\hat{x}, 0) \triangleq \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, 0)\}.$$

Доказательство. 1) Поскольку $\varepsilon_k \geq \varepsilon_{k+1}$, справедливы включения:

$$Y_*(\hat{x}) := Y_*(\hat{x}, 0) \subseteq Y_*(\hat{x}, \varepsilon_{k+1}) \subseteq Y_*(\hat{x}, \varepsilon_k) \subseteq Y(\hat{x}), \quad (2.1.11)$$

откуда следуют неравенства ($k = 0, 1, 2, \dots$)

$$W(\hat{x}, 0) \leq W(\hat{x}, \varepsilon_{k+1}) \leq W(\hat{x}, \varepsilon_k). \quad (2.1.12)$$

Итак, числовая последовательность $\{W_k\}$ является монотонно невозрастающей и ограничена снизу числом $W(\hat{x}, 0) > -\infty$. Поэтому $\exists \lim_{k \rightarrow +\infty} W_k = V > -\infty$.

2) В силу компактности множества $Y_*(\hat{x}, \varepsilon_k)$ (лемма 2.1.2) и непрерывности функции $y \rightarrow F(\hat{x}, y) \quad \forall k = 0, 1, 2, \dots \quad \exists y^k \in Y_*(\hat{x}, \varepsilon_k) \subseteq Y(\hat{x})$:

$$W_k \triangleq W(\hat{x}, \varepsilon_k) \triangleq \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, \varepsilon_k)\} = F(\hat{x}, y^k). \quad (2.1.13)$$

3) С другой стороны, из определения множества $Y_*(\hat{x}, \varepsilon_k)$ имеем

$$G(y^k) \leq \inf_y \{G(y) \mid y \in Y(\hat{x})\} + \varepsilon_k. \quad (2.1.14)$$

Поскольку $y^k \in Y(\hat{x}) \quad \forall k = 0, 1, 2, \dots$, а множество $Y(\hat{x})$ компактно, можно считать, что $y^k \rightarrow \hat{y} \in Y(\hat{x})$.

Тогда из (2.1.14) в пределе получаем равенство:

$$G(\hat{y}) = \inf_y \{G(y) \mid y \in Y(\hat{x})\}.$$

Поскольку $\hat{y} \in Y(\hat{x})$, это означает, что

$$\hat{y} \in Y_*(\hat{x}, 0) \triangleq Y_*(\hat{x}). \quad (2.1.15)$$

4) Далее, из (2.1.13) по непрерывности функции $y \rightarrow F(\hat{x}, y)$ получаем

$$\lim_{k \rightarrow +\infty} W(\hat{x}, \varepsilon_k) = \lim_{k \rightarrow +\infty} F(\hat{x}, y^k) = F(\hat{x}, \hat{y}). \quad (2.1.16)$$

С другой стороны, согласно (2.1.13) имеем

$$F(\hat{x}, y^k) \geq F(\hat{x}, y) \quad \forall y \in Y_*(\hat{x}) \subseteq Y_*(\hat{x}, \varepsilon_k). \quad (2.1.17)$$

Переходя к пределу в (2.1.17) при каждом фиксированном $y \in Y_*(\hat{x})$, получаем

$$F(\hat{x}, \hat{y}) \geq F(\hat{x}, y) \quad \forall y \in Y_*(\hat{x}).$$

В силу (2.1.15) последнее соотношение означает, что

$$F(\hat{x}, \hat{y}) = \sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x}, 0)\} \triangleq W(\hat{x}, 0).$$

Отсюда с учетом (2.1.16) окончательно получаем

$$\lim_{k \rightarrow +\infty} W(\hat{x}, \varepsilon_k) = \lim_{k \rightarrow +\infty} F(\hat{x}, y^k) = F(\hat{x}, \hat{y}) = W(\hat{x}, 0), \quad (2.1.16')$$

что и требовалось. При этом отметим, что сходимость $W_k = W(\hat{x}, \varepsilon_k) \rightarrow W(\hat{x}, 0)$ — монотонная в силу неравенств (2.1.12), и эта сходимость имеет место при каждом $\hat{x} \in X$. #

Функция $\mathcal{V}(\varepsilon)$ значения задачи $(\mathcal{BP}(\varepsilon))$ обладает следующим свойством, развивающим аналогичный результат для значения задачи $(\mathcal{G}(\varepsilon))$ из [35].

Предложение 2.1.2 Пусть выполнены условия (\mathcal{H}) –(2.1.1), а последовательность $\{\varepsilon_k\}$, $\varepsilon_k > 0$, $k = 0, 1, 2, \dots$, монотонно сходится к нулю ($\varepsilon_k \downarrow 0$).

Тогда числовая последовательность $\{\mathcal{V}_k := \mathcal{V}(\varepsilon_k)\}$ значений задач двухуровневого программирования $(\mathcal{BP}(\varepsilon_k))$:

$$\mathcal{V}(\varepsilon_k) \triangleq \inf_{x \in X} \sup_y \{F(x, y) \mid y \in Y_*(x, \varepsilon_k)\}, \quad (2.1.18)$$

монотонно сходится к $\mathcal{V}(0)$ ($\mathcal{V}(\varepsilon_k) \downarrow \mathcal{V}(0)$), где

$$\mathcal{V}(0) \triangleq \inf_{x \in X} \sup_y \{F(x, y) \mid y \in Y_*(x, 0)\}. \quad (2.1.18')$$

Доказательство. 1) Во-первых, заметим, что в силу предположений (\mathcal{H}) –(2.1.1), из которых следует (2.1.3), справедливы неравенства

$$\mathcal{V}_- := \inf_{x, y} \{F(x, y) \mid x \in X, y \in Y(x)\} \geq F_- > -\infty.$$

Поэтому из определения (2.1.18') получаем оценку

$$\mathcal{V}(0) \geq \inf_{x, y} \{F(x, y) \mid x \in X, y \in Y_*(x, 0)\} \geq \inf_{x, y} \{F(x, y) \mid x \in X, y \in Y(x)\} = \mathcal{V}_- > -\infty.$$

С другой стороны, из цепочки (2.1.12) имеем

$$\mathcal{V}_- \leq \mathcal{V}(0) \triangleq \inf_{x \in X} W(x, 0) \leq \inf_{x \in X} W(x, \varepsilon_{k+1}) \leq \inf_{x \in X} W(x, \varepsilon_k) \quad (2.1.19)$$

или, что то же самое, для \mathcal{V}_k ,

$$\mathcal{V}_- \leq \mathcal{V}(0) \leq \mathcal{V}_{k+1} \leq \mathcal{V}_k. \quad (2.1.20)$$

Отсюда следует, что последовательность $\{\mathcal{V}_k\}$ монотонно не возрастает и ограничена снизу. Поэтому

$$\exists \lim_{k \rightarrow +\infty} \mathcal{V}(\varepsilon_k) = \hat{\mathcal{V}} \geq \mathcal{V}(0). \quad (2.1.21)$$

2) Покажем, что $\hat{\mathcal{V}} = \mathcal{V}(0)$. По определению точной нижней грани, для любой последовательности $\{\delta_l\} : \delta_l \downarrow 0$, найдется последовательность $\{x^l\} \subseteq X$, такая что

$$W(x^l, 0) \leq \inf_{x \in X} W(x, 0) + \delta_l \triangleq \mathcal{V}(0) + \delta_l. \quad (2.1.22)$$

Из (2.1.21) следует:

$$\mathcal{V}(0) \leq \hat{\mathcal{V}} = \lim_{k \rightarrow +\infty} \mathcal{V}(\varepsilon_k) = \lim_{k \rightarrow +\infty} \inf_x \{W(x, \varepsilon_k) \mid x \in X\} \leq \lim_{k \rightarrow +\infty} W(x^l, \varepsilon_k). \quad (2.1.23)$$

Эту цепочку, с учетом леммы 2.1.3 ($\varepsilon_k \downarrow 0$) и (2.1.22), можно продолжить следующим образом:

$$\lim_{k \rightarrow +\infty} W(x^l, \varepsilon_k) = W(x^l, 0) \leq \mathcal{V}(0) + \delta_l. \quad (2.1.24)$$

Объединяя (2.1.23) и (2.1.24), получаем:

$$\mathcal{V}(0) \leq \lim_{k \rightarrow +\infty} \mathcal{V}(\varepsilon_k) \leq \mathcal{V}(0) + \delta_l.$$

Переходя в последней цепочке неравенств к пределу при $l \rightarrow +\infty$, получаем требуемое.

Отметим, что сходимость $\mathcal{V}(\varepsilon_k) \rightarrow \mathcal{V}(0)$ является монотонной в силу (2.1.20). #

Доказанное предложение позволяет перейти к пределу в цепочке (2.1.10), как это было сделано в [34] для аналогичной цепочки неравенств, и свести задачу $(\mathcal{BP}(0))$ к серии задач $(\mathcal{BP}_o(\delta, \nu))$. Необходимые для осуществления этой редукции свойства множества решений и значения задачи $(\mathcal{FP}(x))$ будут получены в следующем параграфе.

2.2 Свойства задачи нижнего уровня

Рассмотрим отдельно задачу нижнего уровня двухуровневой задачи (\mathcal{BP}) :

$$\left. \begin{array}{l} \langle d, y \rangle \downarrow \min_y, \\ y \in Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1 x + B_1 y \leq b\}, \end{array} \right\} \quad (\mathcal{FP}(x))$$

как задачу ЛП с параметром $x \in X \triangleq \{x \in \mathbb{R}^m \mid Ax \leq a\}$. Напомним, что по определению многозначное отображение $(x, \varepsilon) \rightarrow Y_*(x, \varepsilon)$ задает множество приближенных ε -решений этой задачи. Установим некоторые свойства многозначных отображений $Y_*(x, \varepsilon)$ и $Y_*(x, 0)$ (которое задает множество точных решений задачи $(\mathcal{FP}(x))$). Для этого введем следующие определения.

Определение 2.2.1 [16] Мнозначное отображение $Z(x)$ называется полунепрерывным сверху в точке \hat{x} , если $\forall\{x^k\} : x^k \rightarrow \hat{x} \quad \forall\{y^k\} : y^k \in Z(x^k), y^k \rightarrow \hat{y}$ выполняется включение $\hat{y} \in Z(\hat{x})$.

Определение 2.2.2 [16] Мнозначное отображение $Z(x)$ называется полунепрерывным снизу в точке \hat{x} , если $\forall\{x^k\} : x^k \rightarrow \hat{x} \quad \forall y \in Z(\hat{x}) \quad \exists\{y^k\} : y^k \in Z(x^k), y^k \rightarrow y$.

Определение 2.2.3 [16] Мнозначное отображение $Z(x)$ называется полунепрерывным сверху (снизу) на множестве X , если оно полунепрерывно сверху (снизу) в каждой точке $x \in X$.

Определение 2.2.4 [16] Мнозначное отображение $Z(x)$ называется K -непрерывным в точке \hat{x} (на множестве X), если оно полунепрерывно сверху и снизу в точке \hat{x} (на множестве X).

Приведем известное свойство многогранных многозначных отображений специального вида, к которым относится отображение $Y(x)$, определяющее допустимое множество задачи $(\mathcal{FP}(x))$.

Лемма 2.2.1 [107, 129] Мнозначное отображение

$$x \rightarrow \mathcal{Y}(x) \triangleq \{y \in \mathbb{R}^{r_1} \mid M_1 y \leq M_2 x + p^1\},$$

где $x \in \mathbb{R}^{r_2}$, $p^1 \in \mathbb{R}^{r_3}$, M_1, M_2 — матрицы соответствующего размера, является K -непрерывным на множестве $\{x \in \mathbb{R}^{r_2} \mid M_3 x \leq p^2\}$, $p^2 \in \mathbb{R}^{r_4}$.

Теперь исследуем свойства множества решений и значения задачи $(\mathcal{FP}(x))$.

Предложение 2.2.1 Мнозначное отображение $(x, \varepsilon) \rightarrow Y_*(x, \varepsilon)$ полунепрерывно сверху на $X \times \mathbb{R}$.

Доказательство. Рассмотрим некоторые последовательности $\{x^k\}, \{\varepsilon_k\}, \{y^k\}$: $x^k \rightarrow \hat{x}$, $\varepsilon_k \rightarrow \hat{\varepsilon}$, $y^k \in Y_*(x^k, \varepsilon_k)$, $y^k \rightarrow \hat{y}$. Необходимо показать, что $\hat{y} \in Y_*(\hat{x}, \hat{\varepsilon})$ или, что то же самое,

$$\hat{y} \in Y(\hat{x}), \quad G(\hat{y}) \leq G(y) + \hat{\varepsilon} \quad \forall y \in Y(\hat{x}). \quad (2.2.25)$$

Сразу же отметим, что первое включение в (2.2.25) выполнено в силу полунепрерывности сверху отображения $x \rightarrow Y(x)$. Для доказательства неравенства из (2.2.25) зафиксируем в нем точку $y = \check{y} \in Y(\hat{x})$:

$$G(\hat{y}) \leq G(\check{y}) + \hat{\varepsilon}. \quad (2.2.26)$$

Включение $y^k \in Y_*(x^k, \varepsilon_k)$ равносильно следующему неравенству:

$$G(y^k) \leq G(y) + \varepsilon_k \quad \forall y \in Y(x^k). \quad (2.2.27)$$

В силу полунепрерывности снизу отображения $x \rightarrow Y(x)$ существует последовательность $\{\check{y}^k\} : \check{y}^k \rightarrow \check{y}, \check{y}^k \in Y(x^k), k = 1, 2, \dots$. Фиксируя в (2.2.27) $y = \check{y}^k$ и переходя к пределу при $k \rightarrow +\infty$, по непрерывности функции $G(\cdot)$ получаем (2.2.26). Наконец, из (2.2.26) в силу произвольности выбора $\check{y} \in Y(\hat{x})$ получаем (2.2.25). #

Следствие 2.2.1 *Многозначное отображение $x \rightarrow Y_*(x, 0)$ полунепрерывно сверху на X .*

Доказательство. Для доказательства данного следствия достаточно в доказательстве предложения 2.2.1 положить $\varepsilon_k := 0, k = 1, 2, \dots, \hat{\varepsilon} := 0$. #

Следствие 2.2.2 *Пусть выполнены предположения (\mathcal{H}) –(2.1.1) и последовательность $\{x^k\}$ сходится: $x^k \rightarrow \hat{x}$. Тогда последовательность $\{\mathcal{V}(\mathcal{FP}(x^k))\}$ значений задач $(\mathcal{FP}(x^k))$ сходится к значению $\mathcal{V}(\mathcal{FP}(\hat{x}))$ задачи $(\mathcal{FP}(\hat{x}))$.*

Доказательство. Рассмотрим последовательность $\{y^k\}$ решений задач $(\mathcal{FP}(x^k))$: $y^k \in Y_*(x^k, 0), k = 1, 2, \dots$. В силу допустимости точки $\{y^k\}$ в задаче $(\mathcal{FP}(x^k))$ и предположений (\mathcal{H}) –(2.1.1) с точностью до подпоследовательности имеем

$$Y \supseteq Y(x^k) \supseteq Y_*(x^k, 0) \ni y^k \rightarrow \hat{y} \in Y. \quad (2.2.28)$$

По следствию 2.2.1 получаем $\hat{y} \in Y_*(\hat{x}, 0)$. Объединяя (2.2.28) и последнее включение, в силу непрерывности скалярного произведения получаем:

$$\mathcal{V}(\mathcal{FP}(x^k)) \triangleq \langle d, y^k \rangle \rightarrow \langle d, \hat{y} \rangle \triangleq \mathcal{V}(\mathcal{FP}(\hat{x})),$$

что и требовалось. #

Чтобы показать полунепрерывность снизу многозначного отображения $x \rightarrow Y_*(x, 0)$ рассмотрим две вспомогательные задачи оптимизации:

$$F_1(y) \triangleq \sum_{i=1}^l |H_i(y)| \downarrow \min_y, \quad y \in Q, \quad (\mathcal{P}_1)$$

$$\left. \begin{aligned} F_2(u, w) &\triangleq \sum_{i=1}^l (u_i + w_i) \downarrow \min_{y, u, w}, \quad y \in Q, \\ (u, w) &\in \{(u, w) \mid H_i(y) = u_i - w_i, \quad u_i \geq 0, \quad w_i \geq 0, \quad i = 1, 2, \dots, l\}, \end{aligned} \right\} \quad (\mathcal{P}_2)$$

где $Q \subseteq \mathbb{R}^n$, $H_i(y) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, l$.

Покажем эквивалентность задач (\mathcal{P}_1) и (\mathcal{P}_2) в смысле решения.

Лемма 2.2.2 *Вектор \hat{y} является решением задачи (\mathcal{P}_1) тогда и только тогда, когда найдутся векторы \hat{u}, \hat{w} такие, что тройка $(\hat{y}, \hat{u}, \hat{w})$ является решением задачи (\mathcal{P}_2) .*

Доказательство. *Достаточность.* Пусть $(\hat{y}, \hat{u}, \hat{w}) \in \text{Sol}(\mathcal{P}_2)$. Очевидно, точка \hat{y} допустима в задаче (\mathcal{P}_1) . Необходимо показать, что $\hat{y} \in \text{Sol}(\mathcal{P}_1)$. Для этого покажем сначала, что $F_1(\hat{y}) = F_2(\hat{u}, \hat{w})$.

Предположим, что для некоторого индекса $j \in \{1, 2, \dots, l\}$ $\hat{u}_j > 0$, $\hat{w}_j > 0$. Тогда, с учетом ограничений типа равенств в задаче (\mathcal{P}_2) , получим:

$$\hat{u}_j - H_j(\hat{y}) = \hat{w}_j > 0, \quad (2.2.29)$$

откуда следует, что $\hat{u}_j > H_j(\hat{y})$, а значит, справедливо неравенство

$$\hat{u}_j > \max\{H_j(\hat{y}), 0\}. \quad (2.2.30)$$

Положим $\check{u} := (\hat{u}_1, \dots, \hat{u}_{j-1}, \check{u}_j, \hat{u}_{j+1}, \dots, \hat{u}_l)$, $\check{w} := (\hat{w}_1, \dots, \hat{w}_{j-1}, \check{w}_j, \hat{w}_{j+1}, \dots, \hat{w}_l)$, где

$$\check{u}_j \triangleq \max\{H_j(\hat{y}), 0\} \geq 0, \quad \check{w}_j \triangleq \check{u}_j - H_j(\hat{y}) \geq 0. \quad (2.2.31)$$

Нетрудно видеть, что тройка $(\hat{y}, \check{u}, \check{w})$ является допустимой в задаче (\mathcal{P}_2) . Кроме того, с учетом (2.2.29)–(2.2.31), получаем:

$$F_2(\check{u}, \check{w}) \triangleq \sum_{i \neq j} (\hat{u}_i + \hat{w}_i) + \check{u}_j + \check{w}_j < \sum_{i \neq j} (\hat{u}_i + \hat{w}_i) + \hat{u}_j + \hat{w}_j \triangleq F_2(\hat{u}, \hat{w}).$$

Последнее неравенство противоречит тому, что $(\hat{y}, \hat{u}, \hat{w}) \in \text{Sol}(\mathcal{P}_2)$. В силу произвольности выбора $j \in \{1, 2, \dots, l\}$ заключаем, что для любого индекса $i \in \{1, 2, \dots, l\}$ либо $\hat{u}_i = 0$, либо $\hat{w}_i = 0$.

Если для некоторого индекса $i \in \{1, 2, \dots, l\}$ имеет место $\hat{u}_i = 0$, тогда из ограничений задачи (\mathcal{P}_2) получаем $\hat{u}_i + \hat{w}_i = -H_i(\hat{y}) \geq 0$. В противном случае, если $\hat{w}_i = 0$, аналогично получаем $\hat{u}_i + \hat{w}_i = H_i(\hat{y}) \geq 0$. Таким образом, имеет место равенство $\hat{u}_i + \hat{w}_i = |H_i(\hat{y})|$, осуществляя суммирование по i в котором, по определению $F_1(\cdot), F_2(\cdot)$ получаем равенство

$$F_2(\hat{u}, \hat{w}) = F_1(\hat{y}). \quad (2.2.32)$$

Предположим теперь, что $\exists \tilde{y} \in Q$:

$$F_1(\tilde{y}) < F_1(\hat{y}). \quad (2.2.33)$$

Определим \tilde{u}, \tilde{w} следующим образом: $\tilde{u}_i := \max\{H_i(\tilde{y}), 0\} \geq 0$, $\tilde{w}_i := \tilde{u}_i - H_i(\tilde{y}) \geq 0$, $i = 1, 2, \dots, l$, так что $\tilde{u}_i + \tilde{w}_i = |H_i(\tilde{y})|$. Производя суммирование по i в последнем равенстве, получаем

$$F_2(\tilde{u}, \tilde{w}) = F_1(\tilde{y}). \quad (2.2.34)$$

Отметим, что, в силу равенств $\tilde{u}_i - \tilde{w}_i \stackrel{\Delta}{=} H_i(\tilde{y})$, $i = 1, 2, \dots, l$, тройка $(\tilde{y}, \tilde{u}, \tilde{w})$ является допустимой в задаче (\mathcal{P}_2) .

Объединяя (2.2.32)–(2.2.34), получаем

$$F_2(\tilde{u}, \tilde{w}) = F_1(\tilde{y}) < F_1(\hat{y}) = F_2(\hat{u}, \hat{w}),$$

что, с учетом допустимости тройки $(\tilde{y}, \tilde{u}, \tilde{w})$ в задаче (\mathcal{P}_2) , противоречит тому, что $(\hat{y}, \hat{u}, \hat{w}) \in \text{Sol}(\mathcal{P}_2)$. Таким образом, предположение (2.2.33) неверно и $\hat{y} \in \text{Sol}(\mathcal{P}_1)$.

Необходимость. Пусть $\hat{y} \in \text{Sol}(\mathcal{P}_1)$. Требуется показать, что $\exists \hat{u}, \hat{w}$: $(\hat{y}, \hat{u}, \hat{w}) \in \text{Sol}(\mathcal{P}_2)$.

Определим $\hat{u}_i := \max\{H_i(\hat{y}), 0\} \geq 0$, $\hat{w}_i := \hat{u}_i - H_i(\hat{y}) \geq 0$. Как и при доказательстве достаточности для тройки $(\tilde{y}, \tilde{u}, \tilde{w})$ получаем, что тройка $(\hat{y}, \hat{u}, \hat{w})$ допустима в задаче (\mathcal{P}_2) и $F_2(\hat{u}, \hat{w}) = F_1(\hat{y})$.

С другой стороны, из ограничений задачи (\mathcal{P}_2) для произвольной ее допустимой точки (y, u, w) имеем:

$$u_i + w_i = 2u_i - H_i(y) \geq -H_i(y),$$

$$u_i + w_i = 2w_i + H_i(y) \geq H_i(y),$$

где $i = 1, 2, \dots, l$. Значит, $u_i + w_i \geq |H_i(y)|$, $i = 1, 2, \dots, l$. Откуда суммированием по i получаем: $F_2(u, w) \geq F_1(y)$.

Следовательно, $F_2(u, w) \geq F_1(y) \stackrel{\Delta}{\geq} F_1(\hat{y}) = F_2(\hat{u}, \hat{w})$. Отсюда заключаем, что $(\hat{y}, \hat{u}, \hat{w}) \in \text{Sol}(\mathcal{P}_2)$. #

Замечание 2.2.1 Поскольку имеет место равенство (2.2.32), нетрудно видеть, что значения задач (\mathcal{P}_1) и (\mathcal{P}_2) совпадают.

Предложение 2.2.2 (см. также [107]) Пусть выполнены предположения (\mathcal{H}) –(2.1.1). Тогда многозначное отображение $x \rightarrow Y_*(x, 0)$ полунепрерывно снизу на X .

Доказательство. Рассмотрим произвольную последовательность $\{x^k\} : x^k \rightarrow \hat{x}$ и зафиксируем некоторую точку $\hat{y} \in Y_*(\hat{x}, 0)$. Необходимо показать, что найдется последовательность $\{y^k\} : y^k \in Y_*(x^k, 0)$, $y^k \rightarrow \hat{y}$ [16].

Для доказательства существования такой последовательности рассмотрим задачу

$$\left. \begin{aligned} \|\hat{y} - y\|_1 &\stackrel{\Delta}{=} \sum_{i=1}^n |\hat{y}_i - y_i| \downarrow \min_y, \\ A_1 x^k + B_1 y &\leq b, \quad y \geq 0, \quad \langle d, y \rangle = \alpha_k \stackrel{\Delta}{=} \mathcal{V}(\mathcal{FP}(x^k)). \end{aligned} \right\} (\mathcal{MP}(x^k))$$

Допустимое множество задачи $(\mathcal{MP}(x^k))$ есть $Y_*(x^k, 0)$ по построению. Следовательно, в силу леммы 2.1.2 оно является компактным. По теореме Вейерштрасса с учетом непрерывности нормы получаем, что $\exists y^k \in \text{Sol}(\mathcal{MP}(x^k)) \subseteq Y_*(x^k, 0)$. Остается показать, что $y^k \rightarrow \hat{y}$.

В силу леммы 2.2.2 задачу $(\mathcal{MP}(x^k))$ можно представить в следующем эквивалентном виде

$$\left. \begin{aligned} \sum_{i=1}^n (\hat{u}_i + w_i) &\downarrow \min_{y, u, w}, \\ \hat{y}_i - y_i &= u_i - w_i, \quad u_i \geq 0, \quad w_i \geq 0, \quad i = 1, 2, \dots, n, \\ A_1 x^k + B_1 y &\leq b, \quad y \geq 0, \quad \langle d, y \rangle = \alpha_k. \end{aligned} \right\} (\mathcal{LP}(x^k))$$

Поскольку задача $(\mathcal{LP}(x^k))$ есть частный случай задачи $(\mathcal{FP}(z))$, где $z \stackrel{\Delta}{=} (x^k, \alpha_k)$, в силу следствия 2.2.2 при $x^k \rightarrow \hat{x}$, $\alpha_k \stackrel{\Delta}{=} \mathcal{V}(\mathcal{FP}(x^k)) \rightarrow \mathcal{V}(\mathcal{FP}(\hat{x}))$ имеем:

$$\mathcal{V}(\mathcal{LP}(x^k)) \rightarrow \mathcal{V}(\mathcal{LP}(\hat{x})). \quad (2.2.35)$$

Используя замечание 2.2.1, из (2.2.35) получаем:

$$\|\hat{y} - y^k\|_1 \stackrel{\Delta}{=} \mathcal{V}(\mathcal{MP}(x^k)) \rightarrow \mathcal{V}(\mathcal{MP}(\hat{x})). \quad (2.2.36)$$

По построению точка \hat{y} является допустимой в задаче $(\mathcal{MP}(\hat{x}))$. Следовательно, эта точка является решением задачи $(\mathcal{MP}(\hat{x}))$. Таким образом, $\mathcal{V}(\mathcal{MP}(\hat{x})) = 0$. Подставляя последнее равенство в (2.2.36), получаем:

$$\|\hat{y} - y^k\|_1 \rightarrow 0.$$

В силу эквивалентности сходимости по норме $\|\cdot\|_1$ и сходимости по евклидовой норме в пространстве \mathbb{R}^n , заключаем, что $y^k \rightarrow \hat{y}$, что и требовалось. #

В следующем параграфе продолжим исследование двухуровневых задач с использованием полученных результатов.

2.3 Редукция к задачам д.с. оптимизации

Взаимосвязь решений двухуровневой задачи $(\mathcal{BP}(0))$ в гарантированной (пессимистической) постановке и двухуровневой задачи $(\mathcal{BP}_o(\delta, \nu))$ в оптимистической постановке устанавливается следующим результатом, который обобщает теорему 2.0.1, доказанную в [34].

Теорема 2.3.1 Пусть выполнены предположения (\mathcal{H}) –(2.1.1), и последовательности $\{\delta_k\}$, $\{\nu_k\}$, $\{\tau_k\}$ таковы, что $\delta_k \downarrow 0$, $\nu_k \downarrow 0$, $\frac{\delta_k}{\nu_k} \downarrow 0$, $\tau_k \downarrow 0$ при $k \rightarrow +\infty$. Пусть далее пара (x^k, y^k) является приближенным τ_k -решением задачи $(\mathcal{BP}_o(\delta_k, \nu_k))$, $k = 0, 1, 2, \dots$

Тогда любая предельная точка (x_g, y_g) последовательности $\{(x^k, y^k)\} : x^k \in X$, $y^k \in Y_{**}(x^k, \delta_k)$ является решением задачи $(\mathcal{BP}(0))$. Другими словами, $x_g \in X$, $y_g \in Y_*(x_g, 0)$ и $F(x_g, y_g) = W(x_g, 0) = \mathcal{V}(0) \triangleq \mathcal{V}(\mathcal{BP}(0))$.

Доказательство. 1) По условию $x^k \in X \forall k$. Поскольку X является замкнутым множеством, имеем $x_g \in X$.

2) В силу предложения 2.1.1 верно включение $y^k \in Y_*(x^k, \delta_k + \nu_k F_+)$. Переходя к пределу при $x^k \rightarrow x_g$, $\delta_k \downarrow 0$, $\nu_k \downarrow 0$, $y^k \rightarrow y_g$, в силу предложения 2.2.1 получаем, что $y_g \in Y_*(x_g, 0)$.

3) Запишем цепочку (2.1.7) при $\varepsilon = 0$, $\delta = \delta_k$, $\nu = \nu_k$, $x = x^k$, $y = y^k$:

$$\sup_y \{F(x^k, y) \mid y \in Y_*(x^k, 0)\} - \frac{\delta_k}{\nu_k} \leq F(x^k, y^k) \leq \sup_y \{F(x^k, y) \mid y \in Y_*(x^k, \delta_k + \nu_k F_+)\}.$$

По определению точной верхней грани получаем неравенство:

$$F(x^k, y) - \frac{\delta_k}{\nu_k} \leq F(x^k, y^k) \quad \forall y \in Y_*(x^k, 0). \quad (2.3.37)$$

В силу предложения 2.2.2 многозначное отображение $Y_*(\cdot, 0)$ полунепрерывно снизу. Поэтому так же, как было получено неравенство (2.2.25) из (2.2.27) при доказательстве предложения 2.2.1, фиксируя точку $\hat{y} \in Y_*(x_g, 0)$ и переходя к пределу при $y \rightarrow \hat{y}$, $x_k \rightarrow x_g$, $y_k \rightarrow y_g$, из (2.3.37) получаем:

$$F(x_g, y) \leq F(x_g, y_g) \quad \forall y \in Y_*(x_g, 0),$$

что и означает выполнение требуемого равенства $F(x_g, y_g) = W(x_g, 0)$.

4) С использованием первого неравенства цепочки (2.1.10) при $\varepsilon = 0$, $\delta = \delta_k$, $\nu = \nu_k$ и допустимости точки (x^k, y^k) в задаче $(\mathcal{BP}_o(\delta_k, \nu_k))$ получаем следующие неравенства:

$$\mathcal{V}(0) - \frac{\delta_k}{\nu_k} \leq \mathcal{V}_o(\delta_k, \nu_k) \leq F(x^k, y^k). \quad (2.3.38)$$

Продолжим цепочку (2.3.38) с учетом того, что пара (x^k, y^k) является τ_k -решением задачи $(\mathcal{BP}_o(\delta_k, \nu_k))$, с использованием второго неравенства цепочки (2.1.10):

$$F(x^k, y^k) \leq \mathcal{V}_o(\delta_k, \nu_k) + \tau_k \leq \mathcal{V}(\delta_k + \nu_k F_+) + \tau_k. \quad (2.3.39)$$

Объединяя (2.3.38) и (2.3.39), имеем:

$$\mathcal{V}(0) - \frac{\delta_k}{\nu_k} \leq F(x^k, y^k) \leq \mathcal{V}(\delta_k + \nu_k F_+) + \tau_k.$$

Переходя в этой цепочке неравенств к пределу при $k \rightarrow +\infty$, с учетом условия теоремы и предложения 2.1.2 заключаем, что $F(x_g, y_g) = \mathcal{V}(0)$. #

Доказанная теорема позволяет вместо задачи (\mathcal{BP}) при $\varepsilon = 0$ (когда минимизация на нижнем уровне осуществляется точно) решать (δ, ν) -параметрическое семейство задач двухуровневого программирования в оптимистической постановке $(\mathcal{BP}_o(\delta, \nu))$.

Далее рассмотрим двухуровневую задачу $(\mathcal{BP}_o(\nu)) \triangleq (\mathcal{BP}_o(0, \nu))$ при фиксированных значениях параметров $\delta = 0$ и $\nu > 0$:

$$\left. \begin{aligned} F(x, y) &\triangleq \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle - \frac{1}{2} \langle y, C_1 y \rangle + \langle c_1, y \rangle \downarrow \min_{x, y}, \\ x &\in \{x \mid Ax \leq a, \quad x \geq 0\}, \quad y \in Y_{**}(x) \triangleq \\ &\triangleq \operatorname{Argmin}_y \{ \langle d, y \rangle - \nu \langle c_1, y \rangle + \frac{\nu}{2} \langle y, C_1 y \rangle \mid A_1 x + B_1 y \leq b, \quad y \geq 0 \}, \end{aligned} \right\} \quad (\mathcal{BP}_o(\nu))$$

где часть целевой функции нижнего уровня, зависящая только от переменной верхнего уровня, отсутствует, так как эта часть целевой функции является константой для задачи нижнего уровня и не влияет на ее решение.

Несмотря на то, что двухуровневая задача $(\mathcal{BP}_o(\nu))$ является квадратично-квадратичной и, следовательно, отличается видом целевых функций верхнего и нижнего уровня от рассмотренной в главе 1 квадратично-линейной двухуровневой задачи в оптимистической постановке, основные результаты по сведению к одноуровневым задачам остаются для нее справедливыми. Так, в силу того что $C_1 \geq 0$, задача нижнего уровня в $(\mathcal{BP}_o(\nu))$ является выпуклой. Следовательно, мы можем заменить ее необходимыми и достаточными условиями оптимальности типа ККТ [2, 6] и получить тем самым задачу (см. §1.1):

$$\left. \begin{aligned} & F(x, y) \downarrow \min_{x, y, v}, \\ & Ax \leq a, \quad x \geq 0, \quad A_1x + B_1y \leq b, \quad y \geq 0, \\ & d - \nu c_1 + \nu C_1y + vB_1 \geq 0, \quad v \geq 0, \\ & h(x, y, v) \triangleq \langle d - \nu c_1 + \nu C_1y, y \rangle + \langle b - A_1x, v \rangle = 0, \end{aligned} \right\} \quad (\mathcal{P}(\nu))$$

где v — вектор множителей Лагранжа.

Теорема 2.3.2 *Для того, чтобы пара (x_*, y_*) являлась решением задачи $(\mathcal{BP}_o(\nu)) \triangleq (\mathcal{BP}_o(0, \nu))$, необходимо и достаточно существования вектора $v_* \in \mathbb{R}^q$ такого, чтобы тройка (x_*, y_*, v_*) являлась решением задачи $(\mathcal{P}(\nu))$.*

Доказательство этой теоремы проводится точно так же, как и доказательство теоремы 1.1.1.

Отметим, что в полученной задаче $(\mathcal{P}(\nu))$, кроме наличия нелинейного ограничения типа равенства, невыпуклость порождается также целевой функцией, вогнутой по y , что отличает ее от задачи (\mathcal{P}) главы 1.

Далее перейдем от задачи $(\mathcal{P}(\nu))$ к задаче с выпуклым допустимым множеством, штрафую ограничение $h(x, y, v) = 0$, как это было сделано в §1.1:

$$\left. \begin{aligned} & \Phi(x, y, v) \triangleq F(x, y) + \mu h(x, y, v) \downarrow \min_{x, y, v}, \\ & Ax \leq a, \quad x \geq 0, \quad A_1x + B_1y \leq b, \quad y \geq 0, \\ & d - \nu c_1 + \nu C_1y + vB_1 \geq 0, \quad v \geq 0, \end{aligned} \right\} \quad (\mathcal{P}(\mu, \nu))$$

где $\mu \geq 0$ — штрафной множитель.

Пусть $(x(\mu), y(\mu), v(\mu)) \in \text{Sol}(\mathcal{P}(\mu, \nu))$. Аналогично рассуждениям, проведенным ранее в §1.1 для задачи $(\mathcal{P}(\mu))$ из главы 1, можно показать, что при выполнении равенства $h(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu})) = 0$ при некотором значении $\hat{\mu} \geq 0$ тройка $(x(\hat{\mu}), y(\hat{\mu}), v(\hat{\mu}))$ является решением задачи $(\mathcal{P}(\nu))$ и равенство $h(x(\mu), y(\mu), v(\mu)) = 0$ остается верным $\forall \mu \geq \hat{\mu}$. Кроме того, $h(x(\mu), y(\mu), v(\mu)) \rightarrow 0$ при $\mu \rightarrow +\infty$.

В качестве приближенного метода решения невыпуклой задачи $(\mathcal{P}(\mu, \nu))$ при фиксированных значениях параметров μ и ν так же, как и для задачи $(\mathcal{P}(\mu))$ из главы 1, предлагается использовать стратегию глобального поиска для задач д.с. оптимизации, разработанную в [36, 38, 42, 43, 47, 119]. Для построения одного из основных элементов этой стратегии — алгоритма локального поиска в задаче $(\mathcal{P}(\mu, \nu))$ — в следующем параграфе развиваются результаты из §1.2.

2.4 Процедуры локального поиска

Запишем целевую функцию задачи $(\mathcal{P}(\mu, \nu))$ в явном виде:

$$\Phi(x, y, v) \triangleq \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle + \left(\mu\nu - \frac{1}{2} \right) \langle y, C_1 y \rangle + \langle c_1 + \mu d - \mu\nu c_1, y \rangle + \mu \langle b - A_1 x, v \rangle.$$

Заметим, что при $\mu < \frac{1}{2\nu}$ эта функция выпукла по x , вогнута по y и билинейна по x и v . Однако при $\mu \geq \frac{1}{2\nu}$ невыпуклость функции $\Phi(x, y, v)$ (а значит и задачи $(\mathcal{P}(\mu, \nu))$) заключена только в билинейном выражении $-\mu \langle A_1 x, v \rangle$, что позволяет в этом случае использовать идею последовательной минимизации по группам переменных, на которой был основан метод локального поиска из §1.2. Отметим, что это удается сделать, несмотря на то, что в отличие от задачи $(\mathcal{P}(\mu))$ из §1.2 задача $(\mathcal{P}(\mu, \nu))$ является задачей со связанными переменными.

Как было отмечено выше, если решения задач $(\mathcal{P}(\mu, \nu))$ и $(\mathcal{P}(\nu))$ совпали при некотором значении μ (см. §2.3), то дальнейшее увеличение значения этого параметра не может привести к несовпадению решений таких задач. Поэтому, не ограничивая общности, далее будем считать, что выполнено условие согласования параметров:

$$\mu \geq \frac{1}{2\nu}. \tag{2.4.40}$$

В этом случае задача $(\mathcal{P}(\mu, \nu))$ при фиксированном значении переменной v становится выпуклой квадратичной задачей оптимизации, а при фиксированной паре (x, y) мы

получаем задачу линейного программирования. Опираясь, в частности, на результаты вычислительного эксперимента, проведенного в главе 1, можно утверждать, что такие вспомогательные задачи могут быть эффективно решены при помощи известных методов выпуклой оптимизации [2, 4, 6, 14, 39] и стандартных пакетов прикладных программ, реализующих эти методы. Таким образом, возникает следующий специальный метод локального поиска.

Обозначим через $D(v) \triangleq \{(x, y) \mid (x, y, v) \in D\}$, $D(x, y) \triangleq \{v \mid (x, y, v) \in D\}$. Пусть задана некоторая начальная точка $(x_0, y_0, v_0) \in D \triangleq \{(x, y, v) \mid Ax \leq a, x \geq 0, A_1x + B_1y \leq b, y \geq 0, d - \nu c_1 + \nu C_1y + vB_1 \geq 0, v \geq 0\}$.

V-процедура

Шаг 0. Положить $s := 0$, $v^s := v_0$.

Шаг 1. Некоторым методом квадратичного программирования найти $\frac{\rho_s}{2}$ -решение (x^{s+1}, y^{s+1}) задачи:

$$\left. \begin{aligned} & \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle + \left(\mu\nu - \frac{1}{2} \right) \langle y, C_1y \rangle + \langle c_1 + \mu d - \mu\nu c_1, y \rangle + \\ & + \mu \langle b - A_1x, v^s \rangle \downarrow \min_{(x,y)} \end{aligned} \right\} \quad (\mathcal{QP}(v^s))$$

так что справедливым будет неравенство

$$\Phi(x^{s+1}, y^{s+1}, v^s) \leq \inf_{(x,y)} \{ \Phi(x, y, v^s) \mid (x, y) \in D(v^s) \} + \frac{\rho_s}{2}. \quad (2.4.41)$$

Шаг 2. Найти $\frac{\rho_s}{2}$ -решение v^{s+1} задачи ЛП:

$$\langle b - A_1x^{s+1}, v \rangle \downarrow \min_v, \quad v \in D(x^{s+1}, y^{s+1}), \quad (\mathcal{LP}(x^{s+1}, y^{s+1}))$$

так что справедливым будет неравенство

$$\Phi(x^{s+1}, y^{s+1}, v^{s+1}) \leq \inf_v \{ \Phi(x^{s+1}, y^{s+1}, v) \mid v \in D(x^{s+1}, y^{s+1}) \} + \frac{\rho_s}{2}. \quad (2.4.42)$$

Шаг 3. Положить $s := s + 1$ и перейти на шаг 1. #

Отметим, что, как и в главе 1, для начала работы V-процедуры не требуется целиком точка (x_0, y_0, v_0) , достаточно только ее части v_0 . Однако теперь необходимо, чтобы начальная точка была допустимой, поскольку в противном случае допустимое множество $D(v^0)$ вспомогательной задачи, решаемой на шаге 1, может оказаться пустым, и решить эту задачу не удастся.

Существенное отличие приведенной выше V -процедуры от V -процедуры из главы 1 состоит в том, что допустимые множества вспомогательных задач меняются на каждом шаге. Это свойство процедуры возникает в силу того, что все переменные x , y и v в ограничениях задачи ($\mathcal{P}(\mu, \nu)$) связаны. Тем не менее, остается справедливой теорема сходимости V -процедуры.

Теорема 2.4.1 *i) При условии $\rho_s > 0, s = 0, 1, 2, \dots, \sum_{s=0}^{\infty} \rho_s < +\infty$, числовая последовательность значений функции $\Phi_s \triangleq \Phi(x^s, y^s, v^s)$, генерируемая V -процедурой, является сходящейся.*

ii) В случае, если $(x^s, y^s, v^s) \rightarrow (\hat{x}, \hat{y}, \hat{v})$, то предел $(\hat{x}, \hat{y}, \hat{v})$ удовлетворяет следующим неравенствам:

$$\Phi(\hat{x}, \hat{y}, \hat{v}) \leq \Phi(x, y, \hat{v}) \quad \forall (x, y) \in D(\hat{v}), \quad (2.4.43)$$

$$\Phi(\hat{x}, \hat{y}, \hat{v}) \leq \Phi(\hat{x}, \hat{y}, v) \quad \forall v \in D(\hat{x}, \hat{y}). \quad (2.4.44)$$

Доказательство. *i)* Введем обозначение: $\bar{\Phi}_s \triangleq \Phi(x^{s+1}, y^{s+1}, v^s)$. Тогда согласно неравенствам (2.4.41) и (2.4.42) справедливы следующие соотношения ($s \geq 1$):

$$\begin{aligned} \Phi_s &\geq \inf_{(x,y)} \{ \Phi(x, y, v^s) \mid (x, y) \in D(v^s) \} \geq \bar{\Phi}_s - \frac{\rho_s}{2} \geq \\ &\geq \inf_v \{ \Phi(x^{s+1}, y^{s+1}, v) \mid v \in D(x^{s+1}, y^{s+1}) \} - \frac{\rho_s}{2} \geq \Phi_{s+1} - \rho_s. \end{aligned} \quad (2.4.45)$$

Откуда следует $\Phi_{s+1} \leq \Phi_s + \rho_s$, т.е. последовательность $\{\Phi_s\}$, $s = 0, 1, 2, \dots$ является почти монотонно невозрастающей. Кроме того, данная последовательность является ограниченной снизу, поскольку функция $\Phi(\cdot)$ ограничена снизу на D и $(x^s, y^s, v^s) \in D$, $s = 0, 1, 2, \dots$, по построению. Тогда, с учетом условия, наложенного на ρ_s , заключаем, что последовательность $\{\Phi_s\}$ сходится (см. лемму 2.6.2 из [6]).

ii) Согласно шагу 1 V -процедуры имеет место неравенство:

$$\Phi(x^{s+1}, y^{s+1}, v^s) - \frac{\rho_s}{2} \leq \Phi(x, y, v^s) \quad \forall (x, y) \in D(v^s).$$

Переходя здесь к пределу при $s \rightarrow +\infty$ (при этом $\rho^s \downarrow 0$) с учетом непрерывности функции $\Phi(x, y, v)$ и полунепрерывности снизу многогранного многозначного отображения $v \rightarrow D(v)$ (см., например, лемму 2.2.1) так же, как и при доказательстве предложения 2.2.1, получаем неравенство (2.4.43).

Аналогично, согласно шагу 2 V -процедуры справедливо неравенство:

$$\Phi(x^{s+1}, y^{s+1}, v^{s+1}) - \frac{\rho_s}{2} \leq \Phi(x^{s+1}, y^{s+1}, v) \quad \forall v \in D(x^{s+1}, y^{s+1}).$$

Переходя здесь к пределу так же, как и выше, получаем неравенство (2.4.44). #

Определение 2.4.1 *Тройку $(\hat{x}, \hat{y}, \hat{v})$, удовлетворяющую неравенствам (2.4.43) и (2.4.44), будем называть критической точкой задачи $(\mathcal{P}(\mu, \nu))$.*

Определение 2.4.2 *Тройку $(\hat{x}, \hat{y}, \hat{v})$, удовлетворяющую неравенствам (2.4.43) и (2.4.44) с точностью $\rho \geq 0$, так что:*

$$\begin{aligned} \Phi(\hat{x}, \hat{y}, \hat{v}) - \rho &\leq \Phi(x, y, \hat{v}) \quad \forall (x, y) \in D(\hat{v}), \\ \Phi(\hat{x}, \hat{y}, \hat{v}) - \rho &\leq \Phi(\hat{x}, \hat{y}, v) \quad \forall v \in D(\hat{x}, \hat{y}), \end{aligned}$$

будем называть приближенно ρ -критической точкой задачи $(\mathcal{P}(\mu, \nu))$.

Отметим, что определения 2.4.1 и 2.4.2 развивают на случай связанных переменных понятие критической точки, определенное в главе 1. Как и раньше, критическая точка является (частично) глобальным решением задачи $(\mathcal{P}(\mu, \nu))$ по паре (x, y) и по переменной v в отдельности. Кроме того, нетрудно показать, что всякое локальное решение задачи $(\mathcal{P}(\mu, \nu))$ будет для нее критической точкой.

Далее заметим, что с помощью цепочки неравенств (2.4.45) так же, как это было сделано в главе 1, можно получить критерии останова V -процедуры. Так, при останове V -процедуры после шага 2 по критерию

$$\Phi_s - \bar{\Phi}_s \leq \tau_1, \tag{2.4.46}$$

тройка (x^s, y^s, v^s) является частично глобальным $(\tau_1 + \frac{\rho_s}{2})$ -решением задачи $(\mathcal{P}(\mu, \nu))$ по (x, y) и частично глобальным $(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2})$ -решением задачи $(\mathcal{P}(\mu, \nu))$ по v . Следовательно, если выбрать $\tau_1 \leq \frac{\tau}{2}$, $\rho_{s-1} \leq \frac{\tau}{4}$, $\rho_s \leq \frac{\tau}{4}$, тройка (x^s, y^s, v^s) будет приближенно τ -критической точкой.

Аналогично, при останове V -процедуры по критерию

$$\bar{\Phi}_s - \Phi_{s+1} \leq \tau_2, \tag{2.4.47}$$

тройка (x^{s+1}, y^{s+1}, v^s) является частично глобальным $(\tau_2 + \rho_s)$ -решением по (x, y) и частично глобальным $(\tau_2 + \frac{\rho_s}{2})$ -решением по v .

Наконец заметим, что можно предложить другой вариант алгоритма локального поиска, как и в главе 1, в котором вспомогательные выпуклые задачи оптимизации решаются в другом порядке.

Пусть задана некоторая допустимая начальная точка $(x_0, y_0, v_0) \in D$.

XY-процедура

Шаг 0. Положить $s := 0$, $v^s := v_0$.

Шаг 1. Найти $\frac{\rho_s}{2}$ -решение v^{s+1} задачи ЛП:

$$\langle b - A_1 x^s, v \rangle \downarrow \min_v, \quad v \in D(x^s, y^s). \quad (\mathcal{LP}(x^s, y^s))$$

Шаг 2. Найти $\frac{\rho_s}{2}$ -решение (x^{s+1}, y^{s+1}) квадратичной задачи:

$$\left. \begin{aligned} & \frac{1}{2} \langle x, Cx \rangle + \langle c, x \rangle + \left(\mu\nu - \frac{1}{2} \right) \langle y, C_1 y \rangle + \langle c_1 + \mu d - \mu\nu c_1, y \rangle + \\ & + \mu \langle b - A_1 x, v^{s+1} \rangle \downarrow \min_{(x, y)} \end{aligned} \right\} \quad (\mathcal{QP}(v^{s+1}))$$

Шаг 3. Положить $s := s + 1$ и перейти на шаг 1. #

Доказательство теоремы сходимости этого варианта алгоритма локального поиска, аналогичной теореме 2.4.1, и получение критериев останова, родственных критериям (2.4.46) и (2.4.47), проводится точно так же, как и для V -процедуры.

При останове XY-процедуры (после шага 2) на итерации s по критерию

$$\Phi_s - \tilde{\Phi}_s \leq \tau_1, \quad (2.4.48)$$

где $\tilde{\Phi}_s \triangleq \Phi(x^s, y^s, v^{s+1})$, полученная точка (x^s, y^s, v^s) является для задачи $(\mathcal{P}(\mu, \nu))$ частично глобальным $\left(\tau_1 + \frac{\rho_s}{2}\right)$ -решением по v и $\left(\tau_1 + \frac{\rho_{s-1}}{2} + \frac{\rho_s}{2}\right)$ -решением по (x, y) .

При останове XY-процедуры по критерию

$$\tilde{\Phi}_s - \Phi_{s+1} \leq \tau_2, \quad (2.4.49)$$

тройка (x^s, y^s, v^{s+1}) является для задачи $(\mathcal{P}(\mu, \nu))$ частично глобальным $(\tau_2 + \rho_s)$ -решением по v и $\left(\tau_2 + \frac{\rho_s}{2}\right)$ -решением по (x, y) .

Описанные процедуры локального поиска будут использованы внутри алгоритма глобального поиска, которому посвящен следующий параграф.

2.5 Алгоритм глобального поиска

Прежде всего отметим, что целевая функция задачи $(\mathcal{P}(\mu, \nu))$ при фиксированных значениях параметров μ, ν может быть представлена в виде разности двух выпуклых функций, например, следующим образом:

$$\Phi(x, y, v) = g(x, y, v) - f(x, y, v), \quad (2.5.50)$$

где $g(x, y, v) = \frac{1}{2}\langle Cx, x \rangle + \langle c, x \rangle + \left(\mu\nu - \frac{1}{2}\right)\langle C_1y, y \rangle + \mu\left(\langle a_1, v \rangle + \frac{1}{4}\|v - A_1x\|^2\right)$ и $f(x, y, v) = \langle (\mu\nu - 1)c_1 - \mu d, y \rangle + \frac{1}{4}\mu\|v + A_1x\|^2$ — выпуклые функции (с учетом предположения (2.4.40)).

Как уже было отмечено выше, локальный поиск не обеспечивает, вообще говоря, достижения глобального решения (см. §1.2, §1.3, §2.3). Поэтому далее необходимо разработать алгоритм глобального поиска для задачи $(\mathcal{P}(\mu, \nu))$. Разумно его было бы строить на тех же принципах, что и соответствующий алгоритм глобального поиска для задачи $(\mathcal{P}(\mu))$ из главы 1 (см. §1.4), поскольку структура невыпуклости в задачах $(\mathcal{P}(\mu, \nu))$ и $(\mathcal{P}(\mu))$ является сходной. Следуя этим принципам, будем декомпозировать задачу $(\mathcal{P}(\mu, \nu))$ на несколько более простых задач.

Стратегия глобального поиска

1) Методом локального поиска (V - или XU -процедурой) находится некоторая приближенно критическая точка (x^k, y^k, v^k) со значением целевой функции $\zeta_k := \Phi(x^k, y^k, v^k)$. Далее производится следующая цепочка операций.

2) Выбирается некоторое значение $\gamma \in [\gamma_-, \gamma_+]$, где

$$\gamma_- \triangleq \inf_{(x,y,v)} (g, D), \quad \gamma_+ \triangleq \sup_{(x,y,v)} (g, D).$$

Начальное значение параметра $\gamma = \gamma_0$ можно выбирать равным, например, $g(x^k, y^k, v^k)$ [43, 47].

3) Строится конечная аппроксимация

$$\mathcal{A}_k = \{(z^l, u^l, w^l) \mid f(z^l, u^l, w^l) = \gamma - \zeta_k, \quad l = 1, 2, \dots, N\}$$

поверхности уровня выпуклой функции $f(\cdot)$:

$$U(\zeta_k) = \{(x, y, v) \mid f(x, y, v) = \gamma - \zeta_k\}.$$

4) Проверяется неравенство

$$g(z^l, u^l, w^l) \leq \gamma, \quad l = 1, 2, \dots, N, \quad (2.5.51)$$

следующее из условий глобальной оптимальности (см. §1.4). Если неравенство (2.5.51) выполнено, то точка аппроксимации будет использоваться дальше. Если же нет — точка (z^l, u^l, w^l) является бесперспективной с точки зрения улучшения текущей критической точки с ее помощью.

5) Начиная с точек (z^l, u^l, w^l) аппроксимации, строятся решения линеаризованных (по базовой невыпуклости) в этих точках задач:

$$\left. \begin{aligned} g(x, y, v) - \langle \nabla f(z^l, u^l, w^l), (x, y, v) \rangle \downarrow \min_{x, y, v} \\ (x, y, v) \in D, \end{aligned} \right\} \quad (\mathcal{P}\mathcal{L}(z^l, u^l, w^l))$$

возникающих из условий глобальной оптимальности (см. §1.4). Заметим, что задачу выпуклого квадратичного программирования $(\mathcal{P}\mathcal{L}(z^l, u^l, w^l))$, как уже было отмечено ранее, можно успешно решить с использованием существующих алгоритмов и программ [2, 6, 14, 39].

6) Стартуя из точек $(\hat{z}^l, \hat{u}^l, \hat{w}^l) \in D$, осуществляется запуск некоторой процедуры локального поиска (см. §2.3), с целью выйти из текущей приближенно критической точки (x^k, y^k, v^k) .

Если улучшить значение ζ_k с помощью всех точек аппроксимации \mathcal{A}_k не удалось, требуется продолжить одномерный поиск по γ на отрезке $[\gamma_-, \gamma_+]$.

Отдельно отметим, что проведения операции 5 при глобальном поиске в задаче $(\mathcal{P}(\mu))$ в главе 1 не требовалось, поскольку соответствующие процедуры локального поиска не нуждались для своего запуска в допустимой точке (в отличие от V - и XU -процедур из §2.4).

Далее напомним, что аппроксимация $\mathcal{A}_k = \mathcal{A}(\zeta_k)$ может строиться, например, с помощью заранее заданного множества направлений

$$Dir \triangleq \{(a^l, b^l, c^l) \in \mathbb{R}^{m+n+q} \mid (a^l, b^l, c^l) \neq 0, \quad l = 0, 1, \dots, N\}.$$

В этом случае тройки (z^l, u^l, w^l) строятся в виде $(z^l, u^l, w^l) = \lambda_l(a^l, b^l, c^l)$, а множитель λ_l вычисляется из условия $f(z^l, u^l, w^l) = \gamma - \zeta_k$, $l = 1, 2, \dots, N$. Для квадратичных функций (каковой является $f(\cdot)$) вычисление таких λ_l не представляет труда (см. §1.4).

Одна из возможных более алгоритмизованных реализаций вышеописанной схемы глобального поиска может выглядеть следующим образом.

Пусть дана некоторая точка $(x_0, y_0, v_0) \in \mathbb{R}^{m+n+q}$, числовые последовательности $\{\tau_k\}$, $\{\delta_k\}$, $\tau_k, \delta_k > 0$, $k = 0, 1, \dots$, $\tau_k \downarrow 0, \delta_k \downarrow 0$ ($k \rightarrow \infty$), множество направлений Dir , числа $\gamma_- \triangleq \inf_{(x,y,v)} (g, D)$ и $\gamma_+ \triangleq \sup_{(x,y,v)} (g, D)$ и параметры алгоритма M и ξ .

Алгоритм глобального поиска

Шаг 0. Положить $k := 0$, $l := 1$, $\gamma := \gamma_-$, $\Delta\gamma := (\gamma_+ - \gamma_-)/M$.

Шаг 1. Построить точку $(\check{x}^k, \check{y}^k, \check{v}^k) \in D$ как приближенное решение задачи $(\mathcal{P}\mathcal{L}(x_0, y_0, v_0))$ одним из методов выпуклого квадратичного программирования.

Шаг 2. Начиная из точки $(\check{x}^k, \check{y}^k, \check{v}^k)$, с помощью XY - или V -процедуры построить τ_k -критическую точку $(x^k, y^k, v^k) \in D$ в задаче $(\mathcal{P}(\mu, \nu))$. Положить $\zeta_k := \Phi(x^k, y^k, v^k)$.

Шаг 3. По точке $(a^l, b^l, c^l) \in Dir$ построить точку (z^l, u^l, w^l) из аппроксимации \mathcal{A}_k поверхности уровня функции $f(\cdot): \mathcal{A}_k = \{(z^l, u^l, w^l) \mid f(z^l, u^l, w^l) = \gamma - \zeta_k\}$.

Шаг 4. Если $g(z^l, u^l, w^l) > \gamma + \xi\gamma$, то положить $l := l + 1$ и перейти на шаг 3.

Шаг 5. Построить точку $(\check{z}^l, \check{u}^l, \check{w}^l) \in D$ как приближенное решение задачи $(\mathcal{P}\mathcal{L}(z^l, u^l, w^l))$ одним из методов выпуклого квадратичного программирования.

Шаг 6. Начиная с точки $(\check{z}^l, \check{u}^l, \check{w}^l)$, специальным методом локального поиска построить δ_k -критическую точку $(\hat{x}^l, \hat{y}^l, \hat{v}^l) \in D$ в задаче $(\mathcal{P}(\mu, \nu))$.

Шаг 7. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) < \Phi(x^k, y^k, v^k)$, то положить $(\check{x}^{k+1}, \check{y}^{k+1}, \check{v}^{k+1}) := (\hat{x}^l, \hat{y}^l, \hat{v}^l)$, $k := k + 1$, $l := 1$, $\gamma := \gamma_-$ и перейти на шаг 2.

Шаг 8. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l < N$, то положить $l := l + 1$ и вернуться на шаг 3.

Шаг 9. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma < \gamma_+$, то положить $\gamma := \gamma + \Delta\gamma$, $l := 1$ и вернуться на шаг 3.

Шаг 10. Если $\Phi(\hat{x}^l, \hat{y}^l, \hat{v}^l) \geq \Phi(x^k, y^k, v^k)$, $l = N$ и $\gamma = \gamma_+$ (т.е. одномерный поиск по γ на отрезке $[\gamma_-, \gamma_+]$ закончен), то стоп; (x^k, y^k, v^k) — полученное решение задачи. #

Напомним, что при численной реализации необходимо использовать параметр ξ , варьированием которого можно изменять точность выполнения неравенства (2.5.51) на шаге 4. Различные значения параметра M отвечают за разбиение отрезка $[\gamma_-, \gamma_+]$ на

соответствующее количество частей для реализации пассивного одномерного поиска по γ .

Можно заметить, что представленный алгоритм и алгоритм глобального поиска из главы 1 (см. §1.4) в определенной мере родственны. Отличие состоит в том, что здесь появляются дополнительные шаги 1, 5, на которых осуществляется построение допустимой начальной точки, необходимой для запуска процедур локального поиска в задаче $(\mathcal{P}(\mu, \nu))$.

Как и ранее, алгоритм глобального поиска в задаче $(\mathcal{P}(\mu, \nu))$ не является алгоритмом в общепринятом смысле, поскольку не все его этапы конкретизированы. Далее необходимо исследовать следующие моменты: выбор параметров алгоритма M и ξ ; поиск чисел γ_- и γ_+ ; выбор множества направлений Dir при построении точек аппроксимации поверхности уровня на основе этого множества на шаге 3 алгоритма.

Что касается первого вопроса, можно отталкиваться от вариантов, использованных при глобальном поиске в главе 1: а) $M = 10$, $\xi = 0$; б) $M = 5$, $\xi = 0.15$; в) $M = 20$, $\xi = 0.05$. Как и ранее, варианты а) и б) выбора параметров применимы в том случае, когда требуется быстро отыскать некоторое приближение к глобальному решению задачи.

Напомним, что для вычисления интервала одномерного поиска $[\gamma_-, \gamma_+]$ нужно решить две задачи: на минимум выпуклой квадратичной функции $g(\cdot)$ и на максимум этой же функции. При этом первая задача может быть решена любым из методов квадратичного программирования [2, 6, 14, 39], а для решения второй можно использовать известную стратегию глобального поиска для задач выпуклой максимизации [43]. Отдельно отметим, что вычислительный процесс не требует точного знания этих границ — достаточно иметь сравнительно грубые оценки [43].

Наконец заметим, что в качестве множества направлений Dir для построения аппроксимации поверхности уровня функции $f(\cdot)$ можно выбрать множество направлений

$$Dir1 = \{((x, y) + e^i, v + e^j), ((x, y) - e^i, v - e^j), i = 1, \dots, m + n, j = 1, \dots, q\},$$

которое было успешно использовано при глобальном поиске в задаче $(\mathcal{P}(\mu))$ в главе 1 (как и прежде, (x, y, v) — текущая критическая точка, e^i, e^j — векторы евклидоваго базиса). Поскольку число точек в аппроксимации, построенной с помощью этого множества, достаточно быстро растет при увеличении размерности задачи, можно исполь-

зовать прием сокращения аппроксимации, описанный в §1.4. При этом невыпуклость в целевой функции задачи $(\mathcal{P}(\mu, \nu))$, как и ранее, задается билинейным выражением с матрицей A_1 .

Итак, все этапы глобального поиска в задаче $(\mathcal{P}(\mu, \nu))$ при фиксированных значениях μ, ν описаны. Отметим, что в сочетании с направленным перебором при уменьшении значения параметра ν и увеличении значения параметра μ (с учетом выполнения условия согласования (2.4.40)) предложенный алгоритм глобального поиска составляет приближенный алгоритм поиска гарантированного решения квадратично-линейной двухуровневой задачи вида $(\mathcal{BP}(0))$.

2.6 Заключительные замечания

Среди результатов, полученных в данной главе, основными являются следующие:

- осуществлена редукция рассматриваемой двухуровневой задачи в гарантированной (пессимистической) постановке к семейству двухуровневых задач в оптимистической постановке со штрафом на нижнем уровне;
- осуществлена дальнейшая редукция задачи этого семейства к одноуровневым задачам, в которых для нахождения критических точек предложены два варианта алгоритма локального поиска;
- разработан алгоритм глобального поиска для редуцированных задач.

В совокупности перечисленные результаты представляют собой оригинальный метод поиска гарантированных решений в двухуровневых задачах исследуемого класса.

Глава 3

Численный поиск гарантированных решений

В главе описывается вычислительный эксперимент, в ходе которого тестируется алгоритм (глобального) поиска гарантированного решения в квадратично-линейных двухуровневых задачах, разработанный в главе 2. Вначале отдельно проводится тестирование алгоритма локального поиска — одного из основных блоков глобального поиска. При этом для получения поля тестовых примеров предлагается новый метод генерации двухуровневых задач исследуемого класса с гарантированным решением, основанный на известной методике построения тестовых двухуровневых задач с оптимистическим решением из [63, 64].

3.1 Генерация тестовых задач

В предыдущей главе диссертации был предложен и теоретически обоснован приближенный алгоритм поиска гарантированных решений в квадратично-линейных двухуровневых задачах следующего вида:

$$\left. \begin{aligned} W(x) &\triangleq \sup_y \{F(x, y) \mid y \in Y_*(x)\} \downarrow \min_x, \\ x \in X, \quad Y_*(x) &\triangleq \operatorname{Argmin}_y \{G(y) \mid y \in Y(x)\}, \end{aligned} \right\} \quad (\mathcal{BP})$$

где $F(x, y) \triangleq \frac{1}{2}\langle x, Cx \rangle + \langle c, x \rangle - \frac{1}{2}\langle y, C_1y \rangle + \langle c_1, y \rangle$, $G(y) = \langle d, y \rangle$,
 $X \triangleq \{x \in \mathbb{R}^m \mid Ax \leq a, x \geq 0\}$, $Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1x + B_1y \leq b, y \geq 0\}$,

$A \in \mathbb{R}^{p \times m}$, $A_1 \in \mathbb{R}^{q \times m}$, $B_1 \in \mathbb{R}^{q \times n}$, $c \in \mathbb{R}^m$, $c_1, d \in \mathbb{R}^n$, $a \in \mathbb{R}^p$, $b \in \mathbb{R}^q$,
 $C = C^T \geq 0$, $C_1 = C_1^T \geq 0$.

При проведении вычислительного эксперимента по тестированию разработанных алгоритмов для проверки эффективности последних необходимо иметь достаточно широкое поле тестовых примеров. Поскольку коллекций тестовых примеров (таких как DIMACS для задач дискретной оптимизации) для двухуровневых задач класса (\mathcal{BP}) (с гарантированным решением) в доступной литературе обнаружить не удалось, был разработан новый метод генерации двухуровневых задач этого класса, основанный на методике генерации двухуровневых задач с оптимистическим решением из [63, 64].

Необходимо подчеркнуть, что для поиска гарантированного решения интерес представляют только двухуровневые задачи, в которых задачи нижнего уровня имеют неединственное решение при некоторых допустимых значениях $x \in X$. Действительно, в противном случае имеем

$$\forall x \in X \quad Y_*(x) = \{y(x)\}, \quad y(x) : G(y(x)) < G(y) \quad \forall y \in Y(x), y \neq y(x), \quad (3.1.1)$$

и следовательно, максимум и минимум функции $F(x, \cdot)$ на множестве $Y_*(x)$, состоящем из одной точки $y(x)$, будут совпадать:

$$W(x) \triangleq \sup_y \{F(x, y) \mid y \in Y_*(x)\} = F(x, y(x)) = \inf_y \{F(x, y) \mid y \in Y_*(x)\}. \quad (3.1.2)$$

По цепочке равенств (3.1.2) заключаем, что вместо требуемой минимизации оценки эффективности $W(x)$ стратегии игрока верхнего уровня в этом случае можно осуществлять минимизацию его критерия эффективности $F(x, y)$ по совокупности переменных (x, y) , т.е. искать оптимистическое решение рассматриваемой задачи.

Отметим, что тестовые двухуровневые задачи, использованные в главе 1, имеют единственное решение на нижнем уровне при всех допустимых значениях переменных игрока верхнего уровня [63]. Следовательно, эти задачи не представляют интереса с точки зрения поиска в них гарантированного решения.

Таким образом, необходимо предложить новый метод генерации тестовых задач. Развивая методику генерации тестовых задач из [63, 64], **на первом этапе** построим двухуровневые задачи-ядра класса (\mathcal{BP}) небольшой размерности и найдем все их локальные и глобальные решения.

Рассмотрим задачи-ядра следующего вида:

$$\left. \begin{aligned} W(x) &= \sup_y \{x^2 - 8x + p_i y_1 - 2y_2^2 \mid y \in Y_*(x)\} \downarrow \min_x, \\ x \in [0; 6], \quad Y_*(x) &\triangleq \underset{y}{\text{Argmin}} \{-y_1 \mid y_1 + y_2 \leq x, \quad y_1 \leq 3, \quad y_1 \geq 0, \quad y_2 \geq 0\}, \end{aligned} \right\} \quad (\mathcal{KP}_i)$$

где $x \in \mathbb{R}$, $y = (y_1, y_2) \in \mathbb{R}^2$, а p_i — некоторый вещественный параметр.

Заметим, что задачи-ядра вида (\mathcal{KP}_i) отличаются от задач-ядер из [63, 64] тем, в особенности, что имеют гарантированную (пессимистическую), а не оптимистическую постановку, поскольку на нижнем уровне нет единственности решения оптимизационной задачи.

Для нахождения решений задачи (\mathcal{KP}_i) прежде всего представим множество $Y_*(x)$ решений задачи нижнего уровня при различных значениях $x \in [0; 6]$ в явном виде. Рассмотрим задачу нижнего уровня, входящую в задачу (\mathcal{KP}_i) , отдельно:

$$\left. \begin{aligned} y_1 &\uparrow \max_y, \\ y_1 + y_2 &\leq x, \quad y_1 \leq 3, \quad y_1 \geq 0, \quad y_2 \geq 0. \end{aligned} \right\} \quad (\mathcal{FKP}(x))$$

Задача $(\mathcal{FKP}(x))$ является задачей ЛП в пространстве \mathbb{R}^2 . Следовательно, ее решение можно провести графически [4].

На рис. 3.1 и 3.2 допустимое множество задачи $(\mathcal{FKP}(x))$ обозначено через $Y(x)$, а вектор градиента целевой функции задачи $(\mathcal{FKP}(x))$ — через ∇ . Нетрудно видеть, что при $x \in [0; 3]$ множество $Y_*(x)$ решений задачи $(\mathcal{FKP}(x))$ состоит из одной точки $(y_1, y_2) = (x, 0)$, а при $x \in]3; 6]$ оно представляет собой отрезок $\{(3, y_2) \mid 0 \leq y_2 \leq x - 3\}$. Заметим, что при $x = 3$ этот отрезок вырождается в точку $(x, 0)$. Таким образом, множество решений задачи $(\mathcal{FKP}(x))$ имеет следующий вид:

$$Y_*(x) = \begin{cases} \{(x, 0)\}, & \text{если } x \in [0; 3], \\ \{(3, y_2) \mid 0 \leq y_2 \leq x - 3\}, & \text{если } x \in [3; 6]. \end{cases} \quad (3.1.3)$$

Вернемся к решению задачи (\mathcal{KP}_i) . Используя представление (3.1.3) множества $Y_*(x)$, запишем выражение для целевой функции задачи (\mathcal{KP}_i) :

$$W(x) = \begin{cases} x^2 - 8x + p_i x, & \text{если } x \in [0; 3], \\ \sup_y \{x^2 - 8x + 3p_i - 2y_2^2 \mid 0 \leq y_2 \leq x - 3\}, & \text{если } x \in [3; 6]. \end{cases} \quad (3.1.4)$$

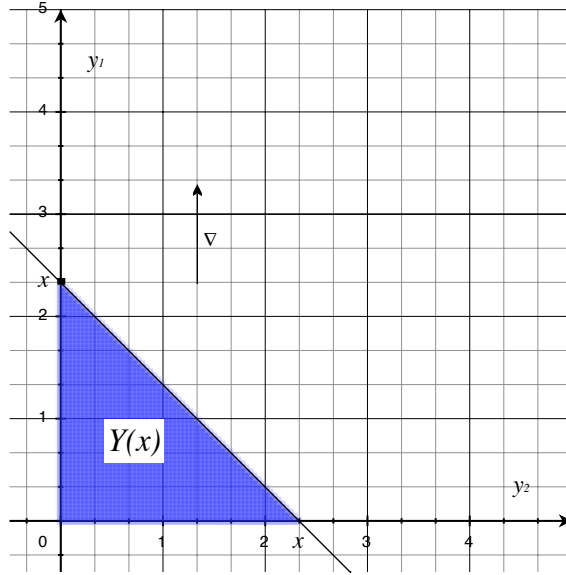


Рис. 3.1: $Y_*(x)$ при $x \leq 3$

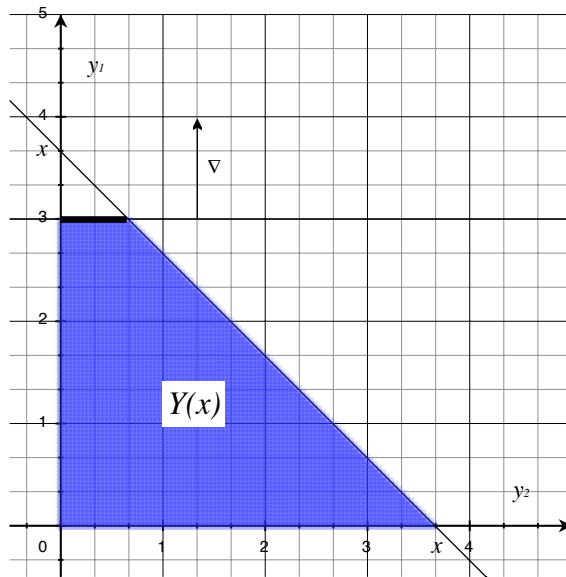


Рис. 3.2: $Y_*(x)$ при $x > 3$

Заметим, что функция, стоящая под знаком \sup в выражении (3.1.4) при $x \in [3; 6]$, убывает по $y_2 \geq 0$. Следовательно, точная верхняя грань достигается в точке $\hat{y}_2 = 0$:

$$\sup_y \{x^2 - 8x + 3p_i - 2y_2^2 \mid 0 \leq y_2 \leq x - 3\} = x^2 - 8x + 3p_i. \quad (3.1.5)$$

Объединяя (3.1.4) и (3.1.5), получаем выражение целевой функции $W(x)$ задачи

(\mathcal{KP}_i) в явном виде:

$$W(x) = \begin{cases} x^2 - 8x + p_i x, & \text{если } x \in [0; 3], \\ x^2 - 8x + 3p_i, & \text{если } x \in [3; 6]. \end{cases} \quad (3.1.6)$$

Таким образом, $W(x)$ является непрерывной кусочно-квадратичной функцией. Нетрудно видеть, что при $p_i \in [2; 8]$ она имеет две стационарные точки: $\hat{x} = 4 - 0.5p_i \in [0; 3]$, $\check{x} = 4 \in [3; 6]$, а при других значениях p_i — одну стационарную точку $\check{x} = 4$. Всюду далее ограничимся рассмотрением трех фиксированных значений параметра $p_i = 3, 4, 6$, $i = 1, 2, 3$, которые лежат в отрезке $[2; 8]$.

На рис. 3.4, 3.3, 3.5 изображены графики функции $W(x)$, соответствующие трем различным значениям параметра p_i : $p_1 = 3$, $p_2 = 4$, $p_3 = 6$. Нетрудно видеть, что задача (\mathcal{KP}_1) (задача (\mathcal{KP}_i) при $p_i = p_1$) имеет локальное решение в точке $\hat{x} = 2.5$ и глобальное решение в точке $\check{x} = 4$. Задача (\mathcal{KP}_2) имеет два глобальных решения в точках $\tilde{x} = 2, \check{x} = 4$ и не имеет локальных решений, не являющихся глобальными. Задача (\mathcal{KP}_3) — глобальное решение в точке $\bar{x} = 1$ и локальное решение в точке $\check{x} = 4$.

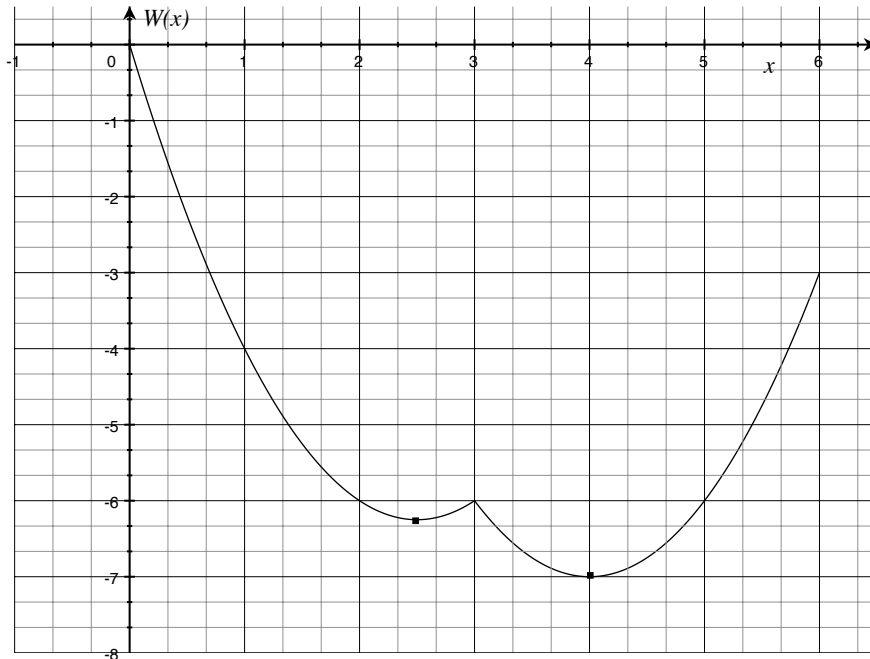


Рис. 3.3: $W(x)$ при $p_i = p_1 = 3$

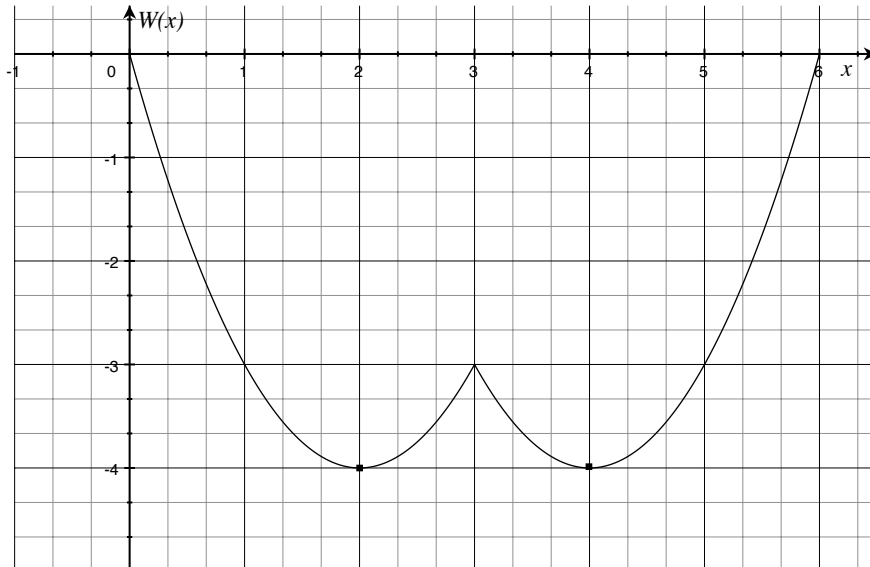


Рис. 3.4: $W(x)$ при $p_i = p_2 = 4$

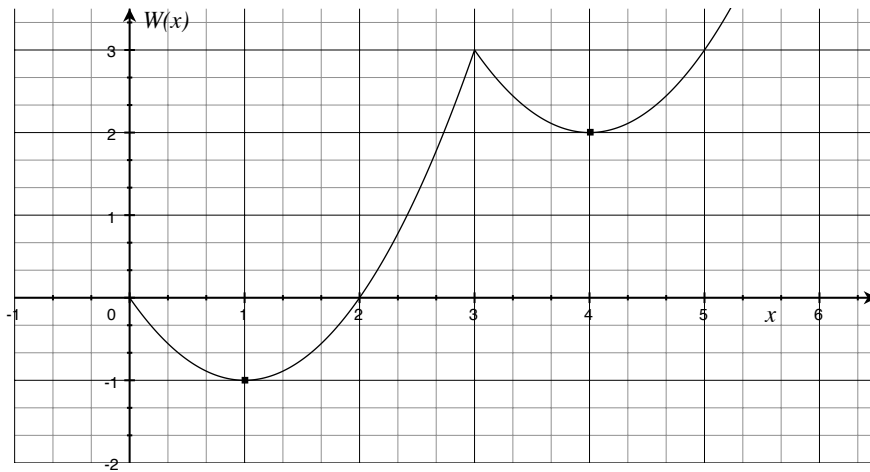


Рис. 3.5: $W(x)$ при $p_i = p_3 = 6$

Отдельно отметим, что, выполняя действия, аналогичные описанным выше, можно отыскать оптимистические решения в задачах (\mathcal{KP}_i) , однако это исследование не представляет интереса с точки зрения цели диссертационной работы и здесь не приводится.

На **втором этапе** используемой методики построения тестовых задач, рассмотрим двухуровневую задачу, полученную объединением произвольного числа r задач-ядер

(\mathcal{KP}_k) , $k \in \{1, 2, 3\}$:

$$\left. \begin{aligned} & \sup_y \left\{ \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(x) \right\} \downarrow \min_x, \quad x \in \Pi[0; 6], \\ & Y_*(x) \triangleq \operatorname{Argmin}_y \left\{ \sum_{i=1}^r G(y_{2i-1}) \mid (y_{2i-1}, y_{2i}) \in Y(x_i), \quad i = 1, \dots, r \right\}, \end{aligned} \right\} \quad (\mathcal{SP})$$

где $F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \triangleq x_i^2 - 8x + \hat{p}_i y_{2i-1} - 2y_{2i}^2$, $\hat{p}_i \in \{p_1, p_2, p_3\}$, $G(y_{2i-1}) \triangleq -y_{2i-1}$, $\Pi[0; 6] \triangleq \{x = (x_1, \dots, x_r) \mid x_i \in [0; 6], \quad i = 1, \dots, r\}$, $Y(x_i) \triangleq \{(y_1, y_2) \mid y_1 + y_2 \leq x_i, y_1 \leq 3, y_1 \geq 0, y_2 \geq 0\}$, $i = 1, \dots, r$, $x \in \mathbb{R}^r$, $y \in \mathbb{R}^{2r}$.

Заметим, что задача нижнего уровня двухуровневой задачи (\mathcal{SP})

$$\left. \begin{aligned} & \sum_{i=1}^r G(y_{2i-1}) \downarrow \min_y, \\ & (y_{2i-1}, y_{2i}) \in Y(x_i), \quad i = 1, \dots, r, \end{aligned} \right\} \quad (\mathcal{FSP}(x))$$

является сепарабельной задачей оптимизации. Как известно [2], ее решение может быть найдено следующим образом.

Лемма 3.1.1 *Вектор $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{2r})$ является решением задачи $(\mathcal{FSP}(x))$ тогда и только тогда, когда его компоненты (y_{2i-1}, y_{2i}) — суть решения задач $(\mathcal{FKP}(x_i))$, $i = 1, \dots, r$.*

Доказательство. Необходимость. Пусть $\hat{y} \in \operatorname{Sol}(\mathcal{FSP}(x))$ или, другими словами, $(\hat{y}_{2i-1}, \hat{y}_{2i}) \in Y(x_i)$, $i = 1, \dots, r$, и имеет место неравенство

$$\sum_{i=1}^r G(\hat{y}_{2i-1}) \leq \sum_{i=1}^r G(y_{2i-1}) \quad \forall y : (y_{2i-1}, y_{2i}) \in Y(x_i), \quad i = 1, \dots, r. \quad (3.1.7)$$

Произвольно выберем индекс $j \in \{1, \dots, r\}$ и покажем справедливость включения $(\hat{y}_{2j-1}, \hat{y}_{2j}) \in \operatorname{Sol}(\mathcal{FKP}(x_j))$. Для этого рассмотрим векторы

$$y(y_{2j-1}, y_{2j}) := (\hat{y}_1, \dots, \hat{y}_{2j-2}, y_{2j-1}, y_{2j}, \hat{y}_{2j+1}, \dots, \hat{y}_{2r}) : (y_{2j-1}, y_{2j}) \in Y(x_j).$$

В силу того, что ограничения в задаче $(\mathcal{FSP}(x))$ накладываются на пары (y_{2i-1}, y_{2i}) , $i = 1, \dots, r$, отдельно, точки вида $y(y_{2j-1}, y_{2j})$ будут допустимыми в этой задаче. Следовательно, неравенство (3.1.7) остается справедливым при $y = y(y_{2j-1}, y_{2j})$:

$$\sum_{i=1}^r G(\hat{y}_{2i-1}) \leq G(y_{2j-1}) + \sum_{i \neq j}^r G(\hat{y}_{2i-1}) \quad \forall (y_{2j-1}, y_{2j}) \in Y(x_j).$$

Упрощая последнее неравенство, получаем:

$$G(\hat{y}_{2j-1}) \leq G(y_{2j-1}) \quad \forall (y_{2j-1}, y_{2j}) \in Y(x_j),$$

что равносильно выполнению включения $(\hat{y}_{2j-1}, \hat{y}_{2j}) \in \text{Sol}(\mathcal{FKP}(x_j))$.

Достаточность. Пусть имеют место включения $(\hat{y}_{2i-1}, \hat{y}_{2i}) \in \text{Sol}(\mathcal{FKP}(x_i))$ для всех $i = 1, \dots, r$. Следовательно, $(\hat{y}_{2i-1}, \hat{y}_{2i}) \in Y(x_i), i = 1, \dots, r$, и точка \hat{y} является допустимой в задаче $(\mathcal{FSP}(x))$. Кроме того, имеют место неравенства

$$G(\hat{y}_{2i-1}) \leq G(y_{2i-1}) \quad \forall (y_{2i-1}, y_{2i}) \in Y(x_i), \quad i = 1, \dots, r. \quad (3.1.8)$$

Суммируя неравенства (3.1.8) по i , получаем неравенство (3.1.7), которое, с учетом допустимости точки \hat{y} в задаче $(\mathcal{FSP}(x))$, равносильно требуемому включению $\hat{y} \in \text{Sol}(\mathcal{FSP}(x))$. #

Для того, чтобы описать структуру локальных решений задачи (\mathcal{SP}) , потребуется следующее вспомогательное утверждение.

Лемма 3.1.2 *Для любых значений $x \in \Pi[0; 6]$ имеют место равенства*

$$\begin{aligned} & \sup_y \left\{ \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(x) \right\} = \\ & = \max_y \left\{ \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(x) \right\} = \\ & = \sum_{i=1}^r \max_{(y_{2i-1}, y_{2i})} \left\{ F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid (y_{2i-1}, y_{2i}) \in Y_*(x_i) \right\}. \end{aligned} \quad (3.1.9)$$

Доказательство. Рассмотрим вспомогательную задачу

$$\sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \uparrow \max_y, \quad y \in Y_*(x), \quad (\mathcal{P}_1(x))$$

где $x \in \Pi[0; 6]$.

Заметим, что в силу леммы 3.1.1 включение $y \in Y_*(x)$ равносильно тому, что $(y_{2i-1}, y_{2i}) \in Y_*(x_i) \quad \forall i = 1, \dots, r$. С учетом сделанного замечания перепишем задачу $(\mathcal{P}_1(x))$ в следующем эквивалентном виде

$$\left. \begin{aligned} & \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \uparrow \max_y, \\ & (y_{2i-1}, y_{2i}) \in Y_*(x_i), \quad i = 1, \dots, r, \end{aligned} \right\} \quad (\mathcal{P}_1(x))$$

и отметим, что она является сепарабельной задачей оптимизации. Значит, для задачи $(\mathcal{P}_1(x))$ справедливо утверждение, аналогичное лемме 3.1.1, а именно: $\hat{y} \in \text{Sol}(\mathcal{P}_1(x))$

тогда и только тогда, когда

$$(\hat{y}_{2i-1}, \hat{y}_{2i}) \in \underset{(y_{2i-1}, y_{2i})}{\text{Argmax}} \{F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid (y_{2i-1}, y_{2i}) \in Y_*(x_i)\} \quad \forall i = 1, \dots, r.$$

Последнее утверждение влечет выполнение второго равенства в (3.1.9).

Отметим, что первое равенство в (3.1.9) справедливо в силу того, что множества $Y_*(x_i)$ решений задач $(\mathcal{FKP}(x_i))$, $i = 1, \dots, r$, в зависимости от выбора значения $x_i \in [0; 6]$ представляют собой точку либо отрезок (непустое компактное множество), следовательно, точная верхняя грань в (3.1.9) достигается. #

Замечание 3.1.1 В силу первого равенства из (3.1.9), задачу (\mathcal{SP}) можно переписать в виде

$$\left. \begin{aligned} & \max_y \left\{ \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(x) \right\} \downarrow \min_x, \\ & x \in \Pi[0; 6], \quad Y_*(x) \triangleq \underset{y}{\text{Argmin}} \left\{ \sum_{i=1}^r G(y_{2i-1}) \mid (y_{2i-1}, y_{2i}) \in Y(x_i), \quad i = 1, \dots, r \right\}. \end{aligned} \right\} \quad (\mathcal{SP})$$

Предложение 3.1.1 Точка $\hat{x} \in \mathbb{R}^r$ является локальным решением задачи (\mathcal{SP}) тогда и только тогда, когда ее компоненты \hat{x}_i являются локальными решениями задач (\mathcal{KP}_k) , соответствующих значениям параметров $p_k = \hat{p}_i$, $i = 1, \dots, r$, использованных при построении задачи (\mathcal{SP}) .

Доказательство. Необходимость. Пусть выполнено включение $\hat{x} \in \text{LocSol}(\mathcal{SP})$ или, другими словами, $\hat{x} \in \Pi[0; 6]$ и существует число $\varepsilon > 0$ такое, что для всех $x \in \Pi[0; 6] \cap U(\hat{x}, \varepsilon)$, $U(\hat{x}, \varepsilon) \triangleq \{x \mid \|x - \hat{x}\| \leq \varepsilon\}$, имеет место неравенство

$$\max_y \left\{ \sum_{i=1}^r F(\hat{x}_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(\hat{x}) \right\} \leq \max_y \left\{ \sum_{i=1}^r F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid y \in Y_*(x) \right\}. \quad (3.1.10)$$

Требуется показать, что $\hat{x}_i \in \text{LocSol}(\mathcal{KP}_k)$ при $p_k = \hat{p}_i$, $i = 1, \dots, r$.

С учетом леммы 3.1.2 из неравенства (3.1.10) получаем, что при $\forall x \in \Pi[0; 6] : \|x - \hat{x}\| \leq \varepsilon$ выполнено следующее неравенство:

$$\begin{aligned} & \sum_{i=1}^r \max_{(y_{2i-1}, y_{2i})} \{F(\hat{x}_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid (y_{2i-1}, y_{2i}) \in Y_*(\hat{x}_i)\} \leq \\ & \leq \sum_{i=1}^r \max_{(y_{2i-1}, y_{2i})} \{F(x_i, y_{2i-1}, y_{2i}; \hat{p}_i) \mid (y_{2i-1}, y_{2i}) \in Y_*(x_i)\}. \end{aligned} \quad (3.1.11)$$

Учитывая, что неравенство (3.1.11) верно при $\forall x \in \Pi[0; 6] : \|x - \hat{x}\| \leq \varepsilon$, полагаем в (3.1.11) $x_i = \hat{x}_i, i \neq j$, где $j \in \{1, \dots, r\}$ — некоторый фиксированный индекс. Упрощая (3.1.11), получаем

$$\begin{aligned} & \forall x_j \in [0; 6] : |x_j - \hat{x}_j| \leq \varepsilon \\ & \max_{(y_{2j-1}, y_{2j})} \{F(\hat{x}_j, y_{2j-1}, y_{2j}; \hat{p}_j) \mid (y_{2j-1}, y_{2j}) \in Y_*(\hat{x}_j)\} \leq \\ & \leq \max_{(y_{2j-1}, y_{2j})} \{F(x_j, y_{2j-1}, y_{2j}; \hat{p}_j) \mid (y_{2j-1}, y_{2j}) \in Y_*(x_j)\}, \end{aligned} \quad (3.1.12)$$

что, с учетом произвольности выбора $j \in \{1, \dots, r\}$, равносильно выполнению включений $\hat{x}_i \in \text{LocSol}(\mathcal{KP}_k)$, $p_k = \hat{p}_i, i = 1, \dots, r$.

Достаточность. Пусть теперь $\hat{x}_i \in \text{LocSol}(\mathcal{KP}_k)$, $k : p_k = \hat{p}_i, i = 1, \dots, r$, или, что то же самое, $\hat{x}_j \in [0; 6]$ и при $\varepsilon = \varepsilon_j$ имеет место неравенство (3.1.12) $\forall j \in \{1, \dots, r\}$. Полагая $\varepsilon = \min\{\varepsilon_1, \dots, \varepsilon_r\}$ и суммируя неравенства (3.1.12) по j , получаем

$$\begin{aligned} & \forall x \in \Pi[0; 6] : |x_j - \hat{x}_j| \leq \varepsilon, \quad j = 1, \dots, r \\ & \sum_{j=1}^r \max_{(y_{2j-1}, y_{2j})} \{F(\hat{x}_j, y_{2j-1}, y_{2j}; \hat{p}_j) \mid (y_{2j-1}, y_{2j}) \in Y_*(\hat{x}_j)\} \leq \\ & \leq \sum_{j=1}^r \max_{(y_{2j-1}, y_{2j})} \{F(x_j, y_{2j-1}, y_{2j}; \hat{p}_j) \mid (y_{2j-1}, y_{2j}) \in Y_*(x_j)\}. \end{aligned}$$

С учетом леммы 3.1.2 и последнего неравенства заключаем, что $\forall x \in \Pi[0; 6] : |x_i - \hat{x}_i| \leq \varepsilon, i = 1, \dots, r$, имеет место неравенство (3.1.10).

Заметим, что из неравенства

$$\|x - \hat{x}\| \triangleq \left(\sum_{i=1}^r (x_i - \hat{x}_i)^2 \right)^{\frac{1}{2}} \leq \varepsilon$$

вытекают неравенства $|x_i - \hat{x}_i| \leq \varepsilon \quad \forall i \in \{1, \dots, r\}$, а значит,

$$U(\hat{x}, \varepsilon) \subset \{x \mid |x_i - \hat{x}_i| \leq \varepsilon, i = 1, \dots, r\}.$$

Следовательно, неравенство (3.1.10) справедливо и при $\forall x \in \Pi[0; 6] \cap U(\hat{x}, \varepsilon)$. Другими словами, имеет место включение $\hat{x} \in \text{LocSol}(\mathcal{SP})$. #

Отметим, что для глобальных решений задачи (\mathcal{SP}) имеет место утверждение, аналогичное предложению 3.1.1.

Предложение 3.1.2 *Точка $\hat{x} \in \mathbb{R}^r$ является (глобальным) решением задачи (\mathcal{SP}) тогда и только тогда, когда ее компоненты \hat{x}_i являются решениями задач (\mathcal{KP}_k), соответствующих значениям параметров $p_k = \hat{p}_i, i = 1, \dots, r$, использованных при построении задачи (\mathcal{SP}).*

Доказательство. Достаточно положить при доказательстве предложения 3.1.1 $\varepsilon = +\infty$ и считать, что $U(\hat{x}, \varepsilon) = \mathbb{R}^r$. #

Следующее предложение описывает зависимость количества локальных и глобальных решений задачи (\mathcal{SP}) от того набора задач (\mathcal{KP}_i) , который был использован при ее построении.

Предложение 3.1.3 (см. также [63, 64]) *Задача (\mathcal{SP}) , в которую входит r_1 задач-ядер вида (\mathcal{KP}_1) , r_2 задач-ядер вида (\mathcal{KP}_2) и r_3 задач-ядер вида (\mathcal{KP}_3) , так что $r_1 + r_2 + r_3 = r$, имеет 2^r локальных решений, из которых 2^{r_2} являются ее глобальными решениями.*

Доказательство. В силу предложения 3.1.1 каждое локальное решение задачи (\mathcal{SP}) представимо в виде $\hat{x} = (\hat{x}_1, \dots, \hat{x}_r)$, где \hat{x}_j — некоторое локальное решение задачи (\mathcal{KP}_i) при $p_i = \hat{p}_j$, $j = 1, \dots, r$. Учитывая, что компоненты вектора \hat{x} не зависят друг от друга и каждая задача-ядро (\mathcal{KP}_i) имеет ровно два локальных решения, по правилу прямого произведения [18] получаем, что в задаче (\mathcal{SP}) существует ровно 2^r локальных решений.

Далее, в силу предложения 3.1.2 каждое глобальное решение задачи (\mathcal{SP}) представимо в виде $x_* = (x_1^*, \dots, x_r^*)$, где x_j^* — некоторое глобальное решение задачи (\mathcal{KP}_i) при $p_i = \hat{p}_j$, $j = 1, \dots, r$. Отметим, что $r_1 + r_3$ компонент вектора x_* фиксированы, поскольку задачи-ядра (\mathcal{KP}_1) и (\mathcal{KP}_3) имеют по одному глобальному решению. Каждая из оставшихся r_2 компонент вектора x_* может принимать два различных значения, равных двум глобальным решениям задачи (\mathcal{KP}_2) . Применяя правило прямого произведения, как и выше, получаем, что задача (\mathcal{SP}) имеет ровно 2^{r_2} глобальных решений. #

На **третьем этапе** построения тестовой задачи в задаче (\mathcal{SP}) , принадлежащей классу двухуровневых задач (\mathcal{BP}) , для того чтобы избавиться от сепарабельности, осуществляется подстановка $x = M_x M_x^{-1} x$, $y = M_y M_y^{-1} y$, где $M_x \in \mathbb{R}^{m \times m} : \det M_x \neq 0$, $M_y \in \mathbb{R}^{n \times n} : \det M_y \neq 0$ (для задачи (\mathcal{SP}) $m = r, n = 2r$), и производится замена переменных $z = M_x^{-1} x$, $u = M_y^{-1} y$.

Осуществляя описанное преобразование с произвольной задачей класса (\mathcal{BP}) , полу-

чаем задачу

$$\left. \begin{aligned} & \sup_u \left\{ \frac{1}{2} \langle z, M_x^T C M_x z \rangle + \langle M_x^T c, z \rangle - \frac{1}{2} \langle u, M_y^T C_1 M_y u \rangle + \langle M_y^T c_1, u \rangle \mid \right. \\ & \quad \left. u \in Y'_*(z) \right\} \downarrow \min_z, \quad z \in X', \\ & Y'_*(z) \triangleq \underset{u}{\text{Argmin}} \{ \langle M_y^T d, u \rangle \mid u \in Y'(z) \}, \end{aligned} \right\} \quad (\mathcal{BP}')$$

где $X' \triangleq \{z \in \mathbb{R}^m \mid A M_x z \leq a, M_x z \geq 0\}$, $Y'(z) \triangleq \{y \in \mathbb{R}^n \mid A_1 M_x z + B_1 M_y u \leq b, M_y u \geq 0\}$.

Заметим, что преобразования $M_x^T C M_x, M_y^T C_1 M_y$ матриц C, C_1 оставляют их положительно полуопределенными. Например, по определению положительной полуопределенности [50] матрицы C имеем

$$\langle x, Cx \rangle \geq 0 \quad \forall x \neq 0. \quad (3.1.13)$$

В силу того, что $\det M_x \neq 0$, неравенство $x \neq 0$ равносильно неравенству $M_x x \neq 0$ [50]. Значит, неравенство (3.1.13) равносильно следующему неравенству:

$$\langle M_x x, C M_x x \rangle \geq 0 \quad \forall (M_x x) \neq 0. \quad (3.1.14)$$

Пользуясь известным свойством скалярного произведения и равносильностью неравенств $x \neq 0$ и $M_x x \neq 0$, из (3.1.14) получаем:

$$\langle x, M_x^T C M_x x \rangle \geq 0 \quad \forall x \neq 0 \quad (3.1.15)$$

или, что то же самое, $M_x^T C M_x \geq 0$.

Таким образом, критерий эффективности игрока верхнего уровня задачи (\mathcal{BP}')

$$F'(z, u) \triangleq \frac{1}{2} \langle z, M_x^T C M_x z \rangle + \langle M_x^T c, z \rangle - \frac{1}{2} \langle u, M_y^T C_1 M_y u \rangle + \langle M_y^T c_1, u \rangle$$

остается выпуклой функцией по переменным игрока верхнего уровня и вогнутой — по переменным игрока нижнего уровня. Следовательно, полученная в результате замены переменных задача (\mathcal{BP}') остается в классе квадратично-линейных двухуровневых задач (\mathcal{BP}) .

Для установления взаимосвязи (локальных и глобальных) решений исходной задачи (\mathcal{BP}) и преобразованной задачи (\mathcal{BP}') потребуются следующие вспомогательные утверждения.

Лемма 3.1.3 *Верны следующие соотношения:*

- i) $x \in X \Leftrightarrow M_x^{-1}x \in X'$.
- ii) $y \in Y(x) \Leftrightarrow M_y^{-1}y \in Y'(M_x^{-1}x)$.
- iii) $y \in Y_*(x) \Leftrightarrow M_y^{-1}y \in Y'_*(M_x^{-1}x)$.

Доказательство. i) По определению множества X включение $x \in X$ равносильно включению $x \in \{x \mid Ax \leq a, x \geq 0\}$. Пользуясь очевидным равенством $E = M_x M_x^{-1}$, перепишем последнее включение в виде:

$$AM_x M_x^{-1}x \leq a, \quad M_x M_x^{-1}x \geq 0,$$

что по определению множества X' равносильно включению $M_x^{-1}x \in X'$.

ii) Второе соотношение доказывается аналогично.

iii) По определению, включение $\hat{y} \in Y_*(x)$ равносильно выполнению следующего неравенства:

$$\langle d, \hat{y} \rangle \leq \langle d, y \rangle \quad \forall y \in Y(x).$$

Как и выше, перепишем последнее неравенство в эквивалентном виде:

$$\langle d, M_y M_y^{-1} \hat{y} \rangle \leq \langle d, M_y M_y^{-1} y \rangle \quad \forall y \in Y(x).$$

Пользуясь свойством скалярного произведения и пунктом 2 этой леммы, из последнего неравенства получаем:

$$\langle M_y^T d, M_y^{-1} \hat{y} \rangle \leq \langle M_y d, M_y^{-1} y \rangle \quad \forall (M_y^{-1} y) \in Y(M_x^{-1} x),$$

что равносильно включению $M_y^{-1} \hat{y} \in Y'_*(M_x^{-1} x)$. #

Лемма 3.1.4 *При согласованности матричной и векторной нормы и произвольном выборе $\varepsilon > 0$*

- i) *из включения $(M_x^{-1}x) \in U\left(M_x^{-1}\hat{x}, \frac{\varepsilon}{\|M_x\|}\right)$ следует включение $x \in U(\hat{x}, \varepsilon)$,*
- ii) *из включения $x \in U\left(\hat{x}, \frac{\varepsilon}{\|M_x^{-1}\|}\right)$ вытекает включение $(M_x^{-1}x) \in U(M_x^{-1}\hat{x}, \varepsilon)$.*

Другими словами, точка x лежит в окрестности точки \hat{x} тогда и только тогда, когда точка $M_x^{-1}x$ принадлежит определенной окрестности точки $M_x^{-1}\hat{x}$.

Доказательство. i) Включение $(M_x^{-1}x) \in U\left(M_x^{-1}\hat{x}, \frac{\varepsilon}{\|M_x\|}\right)$ равносильно неравенству

$$\|M_x^{-1}x - M_x^{-1}\hat{x}\| \leq \frac{\varepsilon}{\|M_x\|}.$$

С учетом согласованности матричной и векторной нормы из последнего неравенства получаем:

$$\|M_x(M_x^{-1}x - M_x^{-1}\hat{x})\| \leq \|M_x\| \|M_x^{-1}x - M_x^{-1}\hat{x}\| \leq \|M_x\| \frac{\varepsilon}{\|M_x\|}. \quad (3.1.16)$$

Глядя на начало и на конец цепочки (3.1.16), получаем

$$\|x - \hat{x}\| \leq \varepsilon,$$

что равносильно включению $x \in U(\hat{x}, \varepsilon)$.

ii) Доказывается аналогично. #

Следующее предложение позволяет перейти от задачи (\mathcal{SP}) (принадлежащей к классу двухуровневых задач (\mathcal{BP})) к задаче (\mathcal{BP}') , что позволяет избавиться от сепарабельности, как это осуществляется в [63, 64] для двухуровневых задач в оптимистической постановке.

Предложение 3.1.4 *Точка \hat{x} является локальным (глобальным) решением задачи (\mathcal{BP}) тогда и только тогда, когда точка $M_x^{-1}\hat{x}$ является локальным (глобальным) решением задачи (\mathcal{BP}') .*

Доказательство. *Необходимость.* Пусть $\hat{x} \in \text{LocSol}(\mathcal{BP})$, тогда $\hat{x} \in X$ и, кроме того, $\exists \varepsilon > 0$:

$$\sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x})\} \leq \sup_y \{F(x, y) \mid y \in Y_*(x)\} \quad \forall x \in X \cap U(\hat{x}, \varepsilon). \quad (3.1.17)$$

В силу леммы 3.1.3 из включения $\hat{x} \in X$ следует включение $M_x^{-1}\hat{x} \in X'$, т.е. $M_x^{-1}\hat{x}$ является допустимой точкой в задаче (\mathcal{BP}') .

Далее, используя очевидное равенство

$$F'(M_x^{-1}x, M_y^{-1}y) \triangleq F(x, y),$$

и лемму 3.1.3, перепишем (3.1.17) в следующем виде:

$$\begin{aligned} & \forall x \in U(\hat{x}, \varepsilon) : M_x^{-1}x \in X' \\ & \sup_y \{F'(M_x^{-1}\hat{x}, M_y^{-1}y) \mid M_y^{-1}y \in Y_*(M_x^{-1}\hat{x})\} \leq \\ & \leq \sup_y \{F'(M_x^{-1}x, M_y^{-1}y) \mid M_y^{-1}y \in Y_*(M_x^{-1}x)\}. \end{aligned} \quad (3.1.18)$$

Пользуясь пунктом i) леммы 3.1.4, из (3.1.18) получаем:

$$\begin{aligned} & \forall (M_x^{-1}x) \in U\left(M_x^{-1}\hat{x}, \frac{\varepsilon}{\|M_x\|}\right) : M_x^{-1}x \in X' \\ & \sup_y \{F'(M_x^{-1}\hat{x}, M_y^{-1}y) \mid M_y^{-1}y \in Y_*(M_x^{-1}\hat{x})\} \leq \\ & \leq \sup_y \{F'(M_x^{-1}x, M_y^{-1}y) \mid M_y^{-1}y \in Y_*(M_x^{-1}x)\}. \end{aligned} \quad (3.1.19)$$

Осуществив замену переменных $z = M_x^{-1}x$, $u = M_y^{-1}y$ в (3.1.19), получим:

$$\begin{aligned} & \forall z \in U\left(M_x^{-1}\hat{x}, \frac{\varepsilon}{\|M_x\|}\right) \cap X' \\ & \sup_u \{F'(M_x^{-1}\hat{x}, u) \mid u \in Y_*(M_x^{-1}\hat{x})\} \leq \sup_u \{F'(z, u) \mid u \in Y_*(z)\}. \end{aligned}$$

Следовательно, $M_x^{-1}\hat{x} \in \text{LocSol}(\mathcal{BP}')$.

Достаточность. Теперь предположим, что $M_x^{-1}\hat{x} \in \text{LocSol}(\mathcal{BP}')$, тогда $M_x^{-1}\hat{x} \in X'$ и $\exists \varepsilon > 0$:

$$\begin{aligned} & \forall z \in U(M_x^{-1}\hat{x}, \varepsilon) \cap X' \\ & \sup_u \{F'(M_x^{-1}\hat{x}, u) \mid u \in Y_*(M_x^{-1}\hat{x})\} \leq \sup_u \{F'(z, u) \mid u \in Y_*(z)\}. \end{aligned} \quad (3.1.20)$$

По лемме 3.1.3 из включения $M_x^{-1}\hat{x} \in X'$ следует включение $\hat{x} \in X$, т.е. точка \hat{x} является допустимой в задаче (\mathcal{BP}) .

Далее, осуществляя замену переменных $z = M_x^{-1}x$, $u = M_y^{-1}y$ и используя лемму 3.1.3 и пункт ii) леммы 3.1.4 так же, как и при доказательстве необходимости, из (3.1.20) получаем:

$$\sup_y \{F(\hat{x}, y) \mid y \in Y_*(\hat{x})\} \leq \sup_y \{F(x, y) \mid y \in Y_*(x)\} \quad \forall x \in U\left(\hat{x}, \frac{\varepsilon}{\|M_x^{-1}\|}\right) \cap X.$$

Таким образом, $\hat{x} \in \text{LocSol}(\mathcal{BP})$.

Наконец, отметим, что для доказательства совпадения глобальных решений задач (\mathcal{BP}) и (\mathcal{BP}') достаточно положить $\varepsilon = +\infty$ и считать, что $U(\hat{x}, \varepsilon) = \mathbb{R}^m$, в приведенном выше доказательстве совпадения локальных решений этих задач. #

Следствие 3.1.1 *Количество локальных (глобальных) решений задачи (\mathcal{BP}') совпадает с количеством локальных (глобальных) решений задачи (\mathcal{BP}) .*

Доказательство. Поскольку $\det M_x^{-1} \neq 0$, отображение $x \rightarrow M_x^{-1}x$ биективно [50]. С учетом этого замечания, доказываемое вытекает из предложения 3.1.4. #

Опишем один из способов построения матриц M_x, M_y , предложенный в [63, 64]. Пусть $M_x := H_x D_x H_x$, $M_y := H_y D_y H_y$, где D_x, D_y — произвольные положительно определенные диагональные матрицы, H_x, H_y — матрицы Хаусхолдера, удовлетворяющие равенствам:

$$H_x \triangleq E - 2v_x v_x^T, \quad H_y \triangleq E - 2v_y v_y^T,$$

а $v_x \in R^m, v_y \in R^n$ — векторы, такие что $\|v_x\| = \|v_y\| = 1$.

Отметим, что в силу равенства $v_x^T v_x = \|v_x\|^2 = 1$ имеет место следующая цепочка:

$$H_x H_x = (E - 2v_x v_x^T)(E - 2v_x v_x^T) = E - 4v_x v_x^T + 4v_x v_x^T v_x v_x^T = E.$$

Аналогичные равенства справедливы и для v_y .

Следовательно, матрицы, обратные к матрицам M_x, M_y , могут быть в этом случае легко вычислены по формулам: $M_x^{-1} = H_x D_x^{-1} H_x$, $M_y^{-1} = H_y D_y^{-1} H_y$.

В заключение параграфа укажем основные преимущества разработанного метода генерации двухуровневых задач класса (\mathcal{BP}) .

1) Данный метод прост в реализации. Действительно, для построения задачи не требуется ничего, кроме элементарных действий с матрицами и векторами.

2) В то же время, метод позволяет построить двухуровневые задачи произвольной размерности $r = r_1 + r_2 + r_3$. При этом выбором чисел r_1, r_2, r_3 можно добиться как относительной простоты получаемой задачи (например, при $r_1 = r_3 = 0$ каждое ее локальное решение будет глобальным решением в силу предложения 3.1.3 и следствия 3.1.1), так и высокой сложности этой задачи (количество ее локальных решений, не являющихся глобальными, по предложению 3.1.3 и следствию 3.1.1 равно $2^r - 2^{r_2}$, т.е. зависит от r экспоненциально).

3.2 Тестирование локального поиска

При проведении вычислительного эксперимента по тестированию процедур локального поиска, разработанных в §2.4, на языке программирования C++ были написаны программы, реализующие XY- и V-процедуры. Для решения вспомогательных задач квадратичного и линейного программирования, возникающих в этих процедурах, как и ранее в §1.3, были использованы метод особых точек и пакет GLPK соответственно.

Также была написана программа, реализующая метод генерации тестовых задач, разработанный в §3.1. При этом, для построения векторов v_x, v_y , участвующих в конструировании тестовой задачи, был использован генератор псевдослучайных чисел из стандартной библиотеки языка C++ (компилятор GCC 4.0.1). С помощью этой программы была сгенерирована серия тестовых задач размерности от 2×4 до 20×40 , имеющих различное количество локальных и глобальных решений.

Счет производился на компьютере с процессором Intel Core 2 Duo 1.6ГГц. В качестве критерия останова V -процедуры использовалось условие (2.4.46), а в качестве критерия останова XU -процедуры — условие (2.4.48). Как и ранее при тестировании локального поиска в двухуровневых задачах с оптимистическим решением в §1.3, требовалось выполнение критерия останова с точностью $\tau = 10^{-4}$.

Целью **первого этапа** тестирования локального поиска был выбор значений параметров штрафа ν , μ , возникающих при редукции (см. §2.3) исходной двухуровневой задачи (\mathcal{BP}) к параметрическому семейству одноуровневых задач

$$\left. \begin{aligned} \Phi(x, y, \nu) &\triangleq F(x, y) + \mu h(x, y, \nu) \downarrow \min_{x, y, \nu}, \\ (x, y, \nu) \in D &\triangleq \{(x, y, \nu) \mid Ax \leq a, \ x \geq 0, \ A_1x + B_1y \leq b, \ y \geq 0, \\ &\quad d - \nu c_1 + \nu C_1y + \nu B_1 \geq 0, \ \nu \geq 0\}, \end{aligned} \right\} \quad (\mathcal{P}(\mu, \nu))$$

где $h(x, y, \nu) \triangleq \nu \langle y, C_1y \rangle + \langle d - \nu c_1, y \rangle + \langle b - A_1x, \nu \rangle$. Выбранные здесь значения параметров μ , ν будут далее использованы в качестве начальных значений этих параметров для направленного их перебора (уменьшение значения ν и увеличение значения μ , если необходимо) при глобальном поиске.

Параметр штрафа μ , возникающий при переходе от вспомогательной задачи с нелинейным ограничением-равенством $h(x, y, \nu) = 0$ к задаче $(\mathcal{P}(\mu, \nu))$ (см. §2.3), ранее встречался в главе 1. Следовательно, при его выборе можно отталкиваться от значения $\mu = 10$, которое позволило уже при тестировании локального поиска в §1.3 получить глобальные решения некоторых задач и, наконец, решить все тестовые задачи из главы 1 в §1.5.

Выбор же значения параметра ν является новым моментом, возникающим при исследовании двухуровневых задач с гарантированным решением. Напомним, что при

построении процедур локального поиска в §2.4 требовалось выполнение условия

$$\mu\nu \geq \frac{1}{2}.$$

С учетом этого условия для тестирования были выбраны следующие значения штрафного параметра ν : $\nu_1 = \frac{1}{10}$, $\nu_2 = \frac{1}{15}$, $\nu_3 = \frac{1}{20}$, соответствующие значению $\mu = 10$.

Начальной точкой для запуска алгоритма локального поиска на данном этапе тестирования была выбрана точка $(x_0, y_0, v_0) = (0, 0, \dots, 0)$. Поскольку процедуры локального поиска требуют для своего запуска допустимую начальную точку (см. §2.4), перед запуском этих процедур решалась квадратичная задача проектирования выбранной точки (x_0, y_0, v_0) на допустимое множество D :

$$\left. \begin{array}{l} \frac{1}{2} \|(x, y, v) - (x_0, y_0, v_0)\|^2 \downarrow \min_{x, y, v}, \\ (x, y, v) \in D. \end{array} \right\} (\mathcal{PR}(x^0, y^0, v^0))$$

В табл. 7 и 8, где представлены результаты тестирования XY- и V-процедур соответственно, использовались следующие обозначения:

№ — номер тестового примера;

Name — название примера в формате “ $m \times n_k$ ”, где, как и ранее, m — размерность вектора x , n — размерность вектора y , k — номер примера данной размерности;

r_1, r_2, r_3 — количество задач-ядер в тестовой задаче, соответствующих значению $p_1 = 3$, $p_2 = 4$ и $p_3 = 6$ соответственно;

Φ_1, Φ_2, Φ_3 — значение целевой функции задачи $(\mathcal{P}(\mu, \nu))$ в найденной приближенно критической точке при соответствующем значении $\nu = \nu_1, \nu_2, \nu_3$;

Φ^* — значение целевой функции задачи $(\mathcal{P}(\mu, \nu))$ в гарантированном решении соответствующей двухуровневой задачи.

Время счета было небольшим (до 2.5с для задач размерности 20×40) и в таблицы не включено. Также в таблицах не представлено количество итераций локального поиска. Как и при тестировании локального поиска в §1.3, критерий останова здесь выполнялся с точностью $\tau = 10^{-4}$ во всех задачах после проведения двух итераций. Кроме того, в таблицах не указано значение функции $h(\cdot)$ в найденных приближенно критических точках, поскольку во всех случаях она обращалась в 0 (с точностью 10^{-4}), т.е. полученные точки были допустимыми во вспомогательных двухуровневых задачах с оптимистическим решением (см. §2.3).

№	Name	r_1	r_2	r_3	Φ_1	Φ_2	Φ_3	Φ^*
1	$2 \times 4_1$	0	2	0	-5.5526	-8.0	-8.0	-8.0
2	$2 \times 4_2$	2	0	0	-12.50	-12.50	-12.50	-14.0
3	$2 \times 4_3$	0	0	2	-1.3881	-1.3881	-1.3881	-2.0
4	$2 \times 4_4$	1	1	0	-8.1025	-10.250	-10.250	-11.0
5	$2 \times 4_5$	0	1	1	-5.0	-5.0	-5.0	-5.0
6	$2 \times 4_6$	1	0	1	-7.250	-7.250	-7.250	-8.0
7	$5 \times 10_1$	1	3	1	<i>-27.250</i>	-19.250	-19.250	-20.0
8	$5 \times 10_2$	3	1	1	-23.750	-23.750	-23.750	-26.0
9	$5 \times 10_3$	1	1	3	-13.250	<i>-19.0</i>	-13.250	-14.0
10	$5 \times 10_4$	2	1	2	<i>-24.250</i>	-18.50	-18.50	-20.0
11	$5 \times 10_5$	2	0	3	<i>-19.0</i>	3.0	3.0	-17.0
12	$10 \times 20_1$	2	7	1	-39.0526	<i>-49.50</i>	-41.50	-43.0
13	$10 \times 20_2$	5	3	2	-45.250	-45.250	-45.250	-49.0
14	$10 \times 20_3$	3	2	5	-29.3026	<i>-39.750</i>	-31.750	-34.0
15	$10 \times 20_4$	0	0	10	-9.3881	-10.0	-9.3881	-10.0
16	$10 \times 20_5$	9	1	0	-66.0	-66.0	-60.250	-67.0
17	$20 \times 40_1$	0	20	0	-80.0	-80.0	-80.0	-80.0
18	$20 \times 40_2$	20	0	0	-125.0	-125.0	-125.0	-140.0
19	$20 \times 40_3$	10	0	10	-72.50	-72.50	-72.50	-80.0
20	$20 \times 40_4$	5	10	5	-73.8026	-76.250	-76.250	-80.0

Таблица 9: **XY-процедура при $\mu = 10$ и различных ν**

Анализируя результаты вычислительного эксперимента, прежде всего отметим, что в некоторых случаях (они выделены в таблицах жирным шрифтом) уже локальным поиском удалось отыскать гарантированное решение двухуровневой задачи. Поскольку функция $h(\cdot)$ обращалась в 0 во всех найденных приближенно критических точках, увеличение выбранного значения параметра $\mu = 10$ на данном этапе не потребовалось.

Кроме того, необходимо отметить, что при $\nu_1 = \frac{1}{10}, \nu_2 = \frac{1}{15}$ в некоторых приближенно критических точках значение целевой функции оказалось лучше, чем в известном гарантированном решении задачи (\mathcal{BP}) (эти случаи выделены в табл. 7 курсивом). Напомним, что функция $h(\cdot)$ в этих точках обращалась в 0. Значит, в этих случаях вспомогательная задача двухуровневого программирования с оптимистическим решением имела допустимые точки, значение целевой функции верхнего уровня в которых лучше, чем в гарантированном решении (см. §2.3). Следовательно, решение этой вспомогательной (оптимистической) задачи не совпадало с гарантированным решением исходной задачи и требовалось использовать меньшее значение параметра ν . Таким образом, значений

$\nu_1 = \frac{1}{10}, \nu_2 = \frac{1}{15}$ недостаточно для отыскания гарантированного решения задачи (BP) и в дальнейших экспериментах будет использоваться значение $\nu = \nu_3 = \frac{1}{20}$.

№	Name	r_1	r_2	r_3	Φ_1	Φ_2	Φ_3	Φ^*
1	$2 \times 4_1$	0	2	0	-8.0	-8.0	-8.0	-8.0
2	$2 \times 4_2$	2	0	0	-13.250	-13.250	-13.250	-14.0
3	$2 \times 4_3$	0	0	2	1.0	1.0	1.0	-2.0
4	$2 \times 4_4$	1	1	0	-10.250	-10.250	-10.250	-11.0
5	$2 \times 4_5$	0	1	1	-5.0	-5.0	-5.0	-5.0
6	$2 \times 4_6$	1	0	1	-8.0	-8.0	-8.0	-8.0
7	$5 \times 10_1$	1	3	1	-17.0	-17.0	-17.0	-20.0
8	$5 \times 10_2$	3	1	1	-22.250	-22.250	-22.250	-26.0
9	$5 \times 10_3$	1	1	3	-4.250	-4.250	-4.250	-14.0
10	$5 \times 10_4$	2	1	2	-13.250	-13.250	-13.250	-20.0
11	$5 \times 10_5$	2	0	3	-7.250	-7.250	-7.250	-17.0
12	$10 \times 20_1$	2	7	1	-40.0	-40.0	-40.0	-43.0
13	$10 \times 20_2$	5	3	2	-46.0	-46.0	-43.0	-49.0
14	$10 \times 20_3$	3	2	5	-22.0	-22.0	-19.0	-34.0
15	$10 \times 20_4$	0	0	10	-7.0	-7.0	17.0	-10.0
16	$10 \times 20_5$	9	1	0	-66.250	-66.250	-66.250	-67.0
17	$20 \times 40_1$	0	20	0	-80.0	-80.0	-80.0	-80.0
18	$20 \times 40_2$	20	0	0	-139.250	-139.250	-139.250	-140.0
19	$20 \times 40_3$	10	0	10	-48.50	-48.50	-49.250	-80.0
20	$20 \times 40_4$	5	10	5	-73.250	-73.250	-65.0	-80.0

Таблица 10: V-процедура при $\mu = 10$ и различных ν

Отметим, что при $\mu = 10, \nu = \frac{1}{20}$ более успешной оказалась V-процедура. С ее помощью удалось решить 4 задачи, а с использованием XY-процедуры — 3 задачи.

На **втором этапе** тестирования локального поиска для той же серии тестовых задач было исследовано поведение XY- и V-процедур в зависимости от стартовой точки. Для этого были выбраны следующие точки:

$$(x_0^{(1)}, y_0^{(1)}, v_0^{(1)}) = (0, 0, \dots, 0), \quad (x_0^{(2)}, y_0^{(2)}, v_0^{(2)}) = (1, 1, \dots, 1),$$

$$(x_0^{(3)}, y_0^{(3)}, v_0^{(3)}) = (1, 0, 1, 0, \dots, 1, 0).$$

Использовались значения параметров $\mu = 10, \nu = \frac{1}{20}$, выбранные на первом этапе тестирования локального поиска.

В табл. 9, 10 использовались следующие новые обозначения:

$\Phi^{(1)}, \Phi^{(2)}, \Phi^{(3)}$ — значения целевой функции задачи $(\mathcal{P}(\mu, \nu))$ в приближенно критических точках, найденных при старте из точек $(x_0^{(1)}, y_0^{(1)}, v_0^{(1)})$, $(x_0^{(2)}, y_0^{(2)}, v_0^{(2)})$ или $(x_0^{(3)}, y_0^{(3)}, v_0^{(3)})$ соответственно;

$h^{(1)}, h^{(2)}, h^{(3)}$ — значения функции $h(\cdot)$ в этих точках.

№	Name	$\Phi^{(1)}$	$h^{(1)}$	$\Phi^{(2)}$	$h^{(2)}$	$\Phi^{(3)}$	$h^{(3)}$	Φ^*
1	$2 \times 4_1$	-8.0	0	-8.0	0	-8.0	0	-8.0
2	$2 \times 4_2$	-12.50	0	-14.0	0	-13.250	0	-14.0
3	$2 \times 4_3$	-1.3881	0	4.0	0	1.0	0	-2.0
4	$2 \times 4_4$	-10.250	0	-11.0	0	-10.250	0	-11.0
5	$2 \times 4_5$	-5.0	0	-2.0	0	-5.0	0	-5.0
6	$2 \times 4_6$	-7.250	0	-5.0	0	-8.0	0	-8.0
7	$5 \times 10_1$	-19.250	0	-17.0	0	-17.0	0	-20.0
8	$5 \times 10_2$	-23.750	0	-23.0	0	-23.0	0	-26.0
9	$5 \times 10_3$	-13.250	0	-5.0	0	-5.0	0	-14.0
10	$5 \times 10_4$	-18.50	0	-14.0	0	-14.0	0	-20.0
11	$5 \times 10_5$	3.0	0	-8.0	0	-8.0	0	-17.0
12	$10 \times 20_1$	-41.50	0	-40.0	0	-40.0	0	-43.0
13	$10 \times 20_2$	-45.250	0	-43.0	0	-43.0	0	-49.0
14	$10 \times 20_3$	-31.750	0	-19.0	0	-19.0	0	-34.0
15	$10 \times 20_4$	-9.3881	0	20.0	0	17.0	0	-10.0
16	$10 \times 20_5$	-60.250	0	-67.0	0	-66.250	0	-67.0
17	$20 \times 40_1$	-80.0	0	-80.0	0	-80.0	0	-80.0
18	$20 \times 40_2$	-125.0	0	-140.0	0	-139.250	0	-140.0
19	$20 \times 40_3$	-72.50	0	-50.0	0	-49.250	0	-80.0
20	$20 \times 40_4$	-76.250	0	-65.0	0	-65.0	0	-80.0

Таблица 11: **XY-процедура при $\mu = 10, \nu = \frac{1}{20}$**

По результатам тестирования, прежде всего, отметим, что в задачах №1, 17 удалось отыскать (глобальное) решение с помощью обеих процедур локального поиска с использованием любого из начальных приближений. Вероятно, это связано с тем, что эти задачи составлены только из задач-ядер вида (\mathcal{KP}_2) (см., например, табл. 7), которые не имеют локальных решений, не являющихся глобальными (см. §3.1).

№	Name	$\Phi^{(1)}$	$h^{(1)}$	$\Phi^{(2)}$	$h^{(2)}$	$\Phi^{(3)}$	$h^{(3)}$	Φ^*
1	$2 \times 4_1$	-8.0	0	-8.0	0	-8.0	0	-8.0
2	$2 \times 4_2$	-13.250	0	-12.50	0	-13.250	0	-14.0
3	$2 \times 4_3$	1.0	0	-2.0	0	1.0	0	-2.0
4	$2 \times 4_4$	-10.250	0	-10.250	0	-10.250	0	-11.0
5	$2 \times 4_5$	-5.0	0	-5.0	0	-5.0	0	-5.0
6	$2 \times 4_6$	-8.0	0	-7.250	0	-8.0	0	-8.0
7	$5 \times 10_1$	-17.0	0	-19.250	0	-17.0	0	-20.0
8	$5 \times 10_2$	-22.750	0	-23.750	0	-22.250	0	-26.0
9	$5 \times 10_3$	-4.250	0	-13.250	0	-4.250	0	-14.0
10	$5 \times 10_4$	-13.250	0	-18.50	0	-13.250	0	-20.0
11	$5 \times 10_5$	-7.250	0	-15.50	0	-7.250	0	-17.0
12	$10 \times 20_1$	-40.0	0	-41.50	0	-40.0	0	-43.0
13	$10 \times 20_2$	-43.0	0	-45.250	0	-43.0	0	-49.0
14	$10 \times 20_3$	-19.0	0	-31.750	0	-19.0	0	-34.0
15	$10 \times 20_4$	17.0	0	-10.0	0	17.0	0	-10.0
16	$10 \times 20_5$	-66.250	0	-63.0	<i>0.20</i>	-66.250	0	-67.0
17	$20 \times 40_1$	-80.0	0	-80.0	0	-80.0	0	-80.0
18	$20 \times 40_2$	-139.250	0	-127.750	<i>0.20</i>	-139.250	0	-140.0
19	$20 \times 40_3$	-49.250	0	-72.50	0	-49.250	0	-80.0
20	$20 \times 40_4$	-65.0	0	-76.250	0	-65.0	0	-80.0

Таблица 12: **V-процедура** при $\mu = 10, \nu = \frac{1}{20}$

Далее, заметим, что с помощью V -процедуры для задач №16, 18 при использовании второй стартовой точки были получены приближенно критические точки, функция $h(\cdot)$ в которых не обратилась в 0. Поэтому было проведено повторное тестирование V -процедуры с увеличением значения параметра μ до 20.

Результаты тестирования V -процедуры со значениями параметров $\mu = 20, \nu = \frac{1}{20}$ представлены в табл. 11. Увеличение значения параметра μ помогло добиться того, чтобы функция $h(\cdot)$ принимала значение 0 во всех найденных приближенно критических точках. Следовательно, значения параметров $\mu = 20, \nu = \frac{1}{20}$ могут быть использованы в качестве начальных значений этих параметров при глобальном поиске с использованием V -процедуры.

Сравнивая между собой результаты работы XU -процедуры из различных начальных приближений, замечаем, что наименьшее количество тестовых задач (3 задачи) было решено с использованием первой стартовой точки. Следовательно, эта точка может быть использована далее для проверки эффективности алгоритма глобального поиска с ис-

пользованием XU -процедуры. Для V -процедуры подобным свойством обладает третье начальное приближение.

№	Name	$\Phi^{(1)}$	$h^{(1)}$	$\Phi^{(2)}$	$h^{(2)}$	$\Phi^{(3)}$	$h^{(3)}$	Φ^*
1	$2 \times 4_1$	-8.0	0	-8.0	0	-8.0	0	-8.0
2	$2 \times 4_2$	-14.0	0	-14.0	0	-13.250	0	-14.0
3	$2 \times 4_3$	1.0	0	-2.0	0	1.0	0	-2.0
4	$2 \times 4_4$	-10.250	0	-10.250	0	-10.250	0	-11.0
5	$2 \times 4_5$	-5.0	0	-5.0	0	-5.0	0	-5.0
6	$2 \times 4_6$	-8.0	0	-7.250	0	-8.0	0	-8.0
7	$5 \times 10_1$	-17.0	0	-17.0	0	-17.0	0	-20.0
8	$5 \times 10_2$	-23.0	0	-23.0	0	-22.250	0	-26.0
9	$5 \times 10_3$	-5.0	0	-10.250	0	-4.250	0	-14.0
10	$5 \times 10_4$	-14.0	0	-16.250	0	-13.250	0	-20.0
11	$5 \times 10_5$	-7.250	0	-13.250	0	-7.250	0	-17.0
12	$10 \times 20_1$	-43.0	0	-38.50	0	-40.0	0	-43.0
13	$10 \times 20_2$	-43.0	0	-46.0	0	-43.0	0	-49.0
14	$10 \times 20_3$	-19.0	0	-31.0	0	-22.0	0	-34.0
15	$10 \times 20_4$	-7.0	0	-7.0	0	17.0	0	-10.0
16	$10 \times 20_5$	-67.0	0	-67.0	0	-66.250	0	-67.0
17	$20 \times 40_1$	-80.0	0	-80.0	0	-80.0	0	-80.0
18	$20 \times 40_2$	-139.250	0	-138.50	0	-139.250	0	-140.0
19	$20 \times 40_3$	-48.50	0	-76.250	0	-49.250	0	-80.0
20	$20 \times 40_4$	-65.0	0	-77.0	0	-65.0	0	-80.0

Таблица 13: V -процедура при $\mu = 20, \nu = \frac{1}{20}$

В целом, по результатам тестирования можно сделать следующие выводы. С помощью различных вариантов локального поиска из различных стартовых точек удалось найти гарантированное решение в половине сгенерированных тестовых задач, что говорит об эффективности разработанных процедур локального поиска. Однако, основную часть таких задач составляют задачи небольшой размерности 2×4 .

Что касается сравнения эффективности XU - и V -процедур, на данном этапе сложно отдать предпочтение одной из них. Значит, обе они могут быть использованы в дальнейшем внутри алгоритма глобального поиска.

3.3 Численный поиск гарантированных решений в сгенерированных задачах

Численный поиск гарантированных решений в тестовых задачах, сгенерированных методом из §3.1, состоял из двух этапов. На начальном этапе осуществлялся численный поиск решений в сгенерированных в предыдущем параграфе тестовых задачах с помощью алгоритма глобального поиска из §2.5. На втором этапе тестирования глобального поиска была предпринята попытка увеличить эффективность глобального поиска при решении задач высокой размерности путем замены подпрограмм решения вспомогательных выпуклых задач оптимизации более производительными процедурами.

Первый этап тестирования алгоритма глобального поиска, предложенного в §2.5, посвящен поиску гарантированных решений в двухуровневых задачах серии, сгенерированной в §3.2 при тестировании процедур локального поиска. Напомним, что с помощью XU -процедуры не удалось найти (глобальное) решение в 17 из 20 задач серии при старте из точки $(0, 0, \dots, 0)$, а с помощью V -процедуры — в 16 задачах при старте из точки $(1, 0, 1, 0, \dots, 1, 0)$. Поэтому эти точки были использованы в качестве начальных точек при тестировании глобального поиска с соответствующими процедурами локального поиска. Стартовые значения штрафных параметров $\nu = \nu_0 = \frac{1}{20}$, $\mu = \mu_0 = 10$ для глобального поиска с XU -процедурой и $\mu = \mu_0 = 20$ для глобального поиска с V -процедурой также были выбраны в §3.2.

Для проведения глобального поиска с использованием XU -процедуры экспериментально были выбраны следующие значения параметров алгоритма, отвечающих за точность и скорость его работы (см. §2.5): $M = 10$, $\xi = 0$, с использованием V -процедуры — $M = 5$, $\xi = 0$. Отметим, что при выборе этих значений учитывались значения соответствующих параметров, использованные при глобальном поиске в §1.5.

Программа, реализующая алгоритм глобального поиска из §2.5, была написана на C++. При этом, для построения допустимой точки по заданной начальной точке на шаге 1 вместо линейризованной задачи ($\mathcal{PL}(x_0, y_0, v_0)$) решалась задача проектирования ($\mathcal{PR}(x_0, y_0, v_0)$) (см. §3.2) с той целью, чтобы начальными критическими точками при глобальном поиске были точки, полученные в §3.2 с помощью процедур локального поиска. Для решения упомянутой задачи проектирования точки на допустимое

множество и частично линеаризованных задач (на шаге 5) применялась подпрограмма решения выпуклых задач квадратичной оптимизации, входящая в программы XU - и V -процедур, описанные и протестированные в §3.2. Для локального поиска на шагах 2, 6 алгоритма глобального поиска были использованы упомянутые программы XU - и V -процедур. Заметим, что для построения аппроксимации поверхности уровня выпуклой функции $f(\cdot)$ использовались множество направлений $Dir1$ и указанный способ сокращения количества точек в аппроксимации из §2.5.

Результаты тестирования программы глобального поиска описаны в табл. 12, где используются следующие новые обозначения:

$LocSol$ — количество локальных решений, не являющихся глобальными, в соответствующей тестовой задаче;

Loc_X, Loc_V — число запусков XU - и V -процедур;

St_X, St_V — количество итераций алгоритма глобального поиска при использовании XU - и V -процедур;

T_X, T_V — время работы программы, реализующей алгоритм глобального поиска с использованием XU - и V -процедуры соответственно;

$\hat{\Phi}$ — значение целевой функции задачи $(\mathcal{P}(\mu, \nu))$ в точке, найденной методом глобального поиска с использованием XU - или V -процедуры (совпало для всех задач серии);

Φ^* — значение целевой функции задачи $(\mathcal{P}(\mu, \nu))$ в гарантированном решении соответствующей двухуровневой задачи.

Анализируя результаты тестирования, следует отметить, что с помощью алгоритма глобального поиска удалось получить (глобальное) решение с точностью $\varepsilon = 10^{-4}$ во всех тестовых примерах. При этом увеличения значения параметра μ и уменьшения параметра ν не потребовалось — начальных значений $\nu = \nu_0 = \frac{1}{20}$, $\mu = \mu_0 = 10$ для глобального поиска с XU -процедурой и $\mu = \mu_0 = 20$ для глобального поиска с V -процедурой оказалось достаточно.

№	Name	LocSol	Loc _X	St _X	T _X	Loc _V	St _V	T _V	$\hat{\Phi}$	Φ^*
1	2 × 4_1	0	25	1	0.30	7	1	0.10	-8.0	-8.0
2	2 × 4_2	3	32	2	0.45	5	2	0.08	-14.0	-14.0
3	2 × 4_3	3	33	2	0.32	9	2	0.10	-2.0	-2.0
4	2 × 4_4	2	34	2	0.44	6	2	0.09	-11.0	-11.0
5	2 × 4_5	2	29	1	0.35	10	1	0.15	-5.0	-5.0
6	2 × 4_6	3	28	2	0.34	7	1	0.09	-8.0	-8.0
7	5 × 10_1	24	138	4	17.84	42	3	6.70	-20.0	-20.0
8	5 × 10_2	30	90	4	14.32	77	5	13.48	-26.0	-26.0
9	5 × 10_3	30	93	3	11.63	47	3	7.57	-14.0	-14.0
10	5 × 10_4	30	105	5	16.38	75	5	13.11	-20.0	-20.0
11	5 × 10_5	31	90	3	12.50	46	2	8.23	-17.0	-17.0
12	10 × 20_1	896	365	5	6:23.00	94	2	1:39.74	-43.0	-43.0
13	10 × 20_2	1016	288	7	5:00.41	159	4	3:14.98	-49.0	-49.0
14	10 × 20_3	1020	325	3	4:29.05	273	6	4:36.09	-34.0	-34.0
15	10 × 20_4	1023	320	2	11:06.42	162	4	1:45.24	-10.0	-10.0
16	10 × 20_5	1022	184	2	5:32.34	59	2	1:57.09	-67.0	-67.0
17	20 × 40_1	0	1152	1	1:16:38.65	536	1	45:38.95	-80.0	-80.0
18	20 × 40_2	1048575	839	2	2:10:38.96	353	2	1:02:27.62	-140.0	-140.0
19	20 × 40_3	1048575	961	3	1:27:36.88	394	4	39:59.01	-80.0	-80.0
20	20 × 40_4	1047552	1127	5	2:28:50.65	595	4	58:20.47	-80.0	-80.0

Таблица 14: **Первый вариант программы глобального поиска**

Сравнивая работу программ, реализующих алгоритм глобального поиска с использованием XU -процедуры и с использованием V -процедуры, по времени и количеству найденных приближенно критических точек, отметим, что почти во всех задачах серии (кроме задачи №14) более эффективным оказался алгоритм, использующий V -процедуру для локального поиска. По-видимому, это связано с тем, что для отыскания глобального решения во всех тестовых задачах алгоритмом глобального поиска с V -процедурой оказалось достаточным использование значения параметра счета $M = 5$, которое является менее “строгим”, чем значение $M = 10$, используемое при глобальном поиске с использованием XU -процедуры.

Второй этап тестирования глобального поиска посвящен численному поиску гарантированных решений в сериях сгенерированных двухуровневых задач большей размерности. С этой целью был разработан второй вариант программы, реализующей алгоритм глобального поиска из §2.5, с использованием более производительного пакета Xpress-MP [85] для решения вспомогательных выпуклых задач оптимизации, возникающих на

этапе локального поиска, как это было сделано в §1.5 при численном поиске оптимистических решений в квадратично-линейных двухуровневых задачах. На данном этапе тестирования глобального поиска счет проводился на компьютере с процессором Intel Core 2 Duo 2.0GHz.

При проведении тестирования методом из §3.1 были сгенерированы серии двухуровневых задач размерностей от 5×10 до 35×70 . Использовался вариант алгоритма локального поиска с V -процедурой локального поиска, показавший свое преимущество по времени счета на предыдущем этапе тестирования. Все остальные условия вычислительного эксперимента остались без изменений.

Результаты тестирования глобального поиска на сериях тестовых задач представлены в таблице 15, в которой использовались следующие обозначения:

$m + n$ — суммарная размерность (по переменным x и y) каждой тестовой двухуровневой задачи в серии;

N — количество задач в серии;

$LocSol_{avg}$ — среднее количество локальных решений, не являющихся глобальными, в задачах серии;

Loc_{avg} — среднее число запусков алгоритма локального поиска при проведении глобального поиска в серии задач;

St_{avg} — среднее количество итераций алгоритма глобального поиска;

T_{avg} — среднее время работы программы, реализующей алгоритм глобального поиска.

$m + n$	N	$LocSol_{avg}$	Loc_{avg}	St_{avg}	T_{avg}
15	10	28.0	30.5	1.5	1.29
30	10	951.5	150.1	1.5	17.17
45	10	32680.0	447.7	2.7	1:43.16
60	10	942783.0	746.2	4.9	5:08.00
75	10	$3.3 \cdot 10^7$	799.1	4.1	8:01.57
90	10	$1.0 \cdot 10^9$	1462.9	5.9	20:46.09
105	10	$3.4 \cdot 10^{10}$	5863.4	13.6	1:51:21.80

Таблица 15: Тестирование глобального поиска на сериях задач

По результатам вычислительного эксперимента отметим, что вторым вариантом программы, реализующей алгоритм глобального поиска, удалось за приемлемое время найти (глобальные) гарантированные решения с точностью $\varepsilon = 10^{-3}$ во всех тестовых задачах до размерности 105. При этом изменения выбранных при тестировании локального

поиска значений параметров штрафа, как и на предыдущем этапе тестирования глобального поиска, не потребовалось.

Отдельно отметим, что количество локальных решений, не являющихся глобальными, в тестовых задачах в среднем достигает очень большого числа $3.4 \cdot 10^{10}$, с которым оказываются не сравнимы число запусков алгоритма локального поиска при глобальном поиске (количество найденных критических точек), равное в среднем 5863.4, и количество итераций алгоритма глобального поиска (число критических точек, позволивших улучшить значение целевой функции), которое в среднем не превысило числа 14.

Таким образом, использование более эффективных подпрограмм для решения вспомогательных задач квадратичного и линейного программирования позволило значительно повысить эффективность глобального поиска.

3.4 Заключительные замечания

В главе получены следующие основные результаты:

- разработан и обоснован метод генерации квадратично-линейных двухуровневых задач с гарантированным решением;
- разработаны программы, реализующие алгоритмы локального и глобального поисков из главы 2, которые были успешно протестированы на сгенерированных упомянутым методом сериях задач размерности до 105.

Проведенный вычислительный эксперимент продемонстрировал эффективность разработанного метода поиска гарантированных решений в исследуемых двухуровневых задачах.

Заключение

В работе исследованы два класса одной из актуальных задач современного исследования операций — непрерывной задачи двухуровневой оптимизации. Разработана и обоснована методика их решения, которая была протестирована при проведении вычислительных экспериментов.

Основные результаты диссертационной работы заключаются в следующем.

1) Предложены и обоснованы методы локального и глобального поисков оптимистических решений в квадратично-линейных двухуровневых задачах. Численное тестирование показало их эффективность на широком спектре случайно сгенерированных тестовых задач размерности до 300.

2) Теоретически обоснованы взаимосвязи задачи поиска гарантированного решения квадратично-линейной двухуровневой задачи и квадратично-квадратичной двухуровневой задачи специального вида в оптимистической постановке. Обоснована редукция поиска гарантированного решения квадратично-линейной двухуровневой задачи к решению семейства таких вспомогательных задач в оптимистической постановке.

3) Для задачи поиска гарантированных решений в квадратично-линейных двухуровневых задачах предложены и обоснованы методы локального и глобального поисков. Разработан новый метод генерации квадратично-линейных двухуровневых задач различной сложности и размерности с известными гарантированными решениями. Численное тестирование разработанных методов локального и глобального поисков на сериях сгенерированных задач размерности до 105 продемонстрировало их эффективность.

Отметим, что в литературе не удалось отыскать результатов численного решения нелинейных двухуровневых задач в оптимистической постановке такой размерности, а численный поиск гарантированного решения непрерывных двухуровневых задач, в которых допустимое множество задачи нижнего уровня зависит от переменных верхнего уровня, насколько известно, был осуществлен впервые.

Литература

- [1] Антипин А.С. Экстрапроксимальный метод решения равновесных и игровых задач // Журн. вычисл. матем. и матем. физики. 2005. Т. 45, №11, 12. С. 1969–1990, 2102–2111.
- [2] Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. М.: Мир, 1982.
- [3] Береснев В.Л., Мельников А.А. Приближённые алгоритмы для задачи конкурентного размещения предприятий // Дискретный анализ и исследование операций. 2010. Т. 17, №6. С. 3–19.
- [4] Васильев Ф.П., Иваницкий А.Ю. Линейное программирование. М.: Факториал, 1998.
- [5] Васильев И.Л., Климентова К.Б., Орлов А.В. Параллельный глобальный поиск равновесных ситуаций в биматричных играх // Вычислительные методы и программирование. Издательство Московского университета. 2007. Т. 8, №2. С. 84–94.
- [6] Васильев Ф.П. Методы оптимизации. М.: Факториал-пресс, 2002.
- [7] Васин А.А., Морозов В.В. Теория игр и модели математической экономики. М.: МАКС Пресс, 2005.
- [8] Гермейер Ю.Б. Игры с противоположными интересами. М.: Наука, 1976.
- [9] Гимади Э.Х., Гончаров Е.Н. Двухуровневая задача выбора системы машин и узлов с нелинейной производственной функцией // Сибирский журнал индустриальной математики. 2006. Т. 9, №2. С. 44–54.

- [10] Горбачевская Л.Е., Дементьев В.Т., Шамардин Ю.В. Двухуровневая задача стандартизации с условием единственности оптимального потребительского выбора // Дискретный анализ и исследование операций, серия 2. 1999. Т. 6, №2. С. 3–11.
- [11] Горелик В.А., Кононенко А.Ф. Теоретико-игровые модели принятия решений в эколого-экономических системах. М.: Радио и связь, 1982.
- [12] Груздева Т.В. Решение задачи о клике сведением к задаче с д.с. ограничением // Дискретн. анализ и исслед. опер. 2008. Т. 15, №6. С. 20–33.
- [13] Груздева Т.В., Петрова Е.Г. Численное решение линейной двухуровневой задачи // Журн. вычисл. матем. и матем. физ. 2010. Т. 50, № 10. С. 1715–1726.
- [14] Даугавет В.А. Численные методы квадратичного программирования: учеб. пособие. СПб.: Изд-во С.-Петербур. ун-та, 2004.
- [15] Дементьев В.Т., Шамардин Ю.В. Двухуровневая задача о назначениях при обобщенном условии Монжа // Дискретный анализ и исследование операций, серия 2. 2003. Т. 10, №2. С. 19–28.
- [16] Демьянов В.Ф., Васильев Л.В. Недифференцируемая оптимизация. М.: Наука, 1981.
- [17] Иваненко Д.С., Плясунов А.В. О сводимости задач двухуровневого программирования к задачам векторной оптимизации // Дискретный анализ и исследование операций, серия 2. 2007. Т. 14, №1. С. 72–99.
- [18] Иванов Б.Н. Дискретная математика. Алгоритмы и программы: учеб. пособие. М.: Лаборатория Базовых Знаний, 2002.
- [19] Измаилов А.Ф., Погосян А.Л. Условия оптимальности и ньютоновские методы для задач оптимизации с исчезающими ограничениями // Журнал вычислительной математики и математической физики. 2009. Т. 49, №7. С. 1184–1196.
- [20] Малышев А.В. Алгоритм поиска гарантированного решения квадратично-линейной двухуровневой задачи // Материалы Российской конференции “Дискретная оптимизация и исследование операций” (Алтай, 27 июня – 3 июля 2010 года). Новосибирск, 2010. С. 116.

- [21] Малышев А.В. Алгоритм поиска гарантированного решения одного класса двухуровневых задач // Материалы XI Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям (Иркутск, Байкал, 15–21 марта 2010). Иркутск, 2010. С. 51.
- [22] Малышев А.В. Алгоритм поиска гарантированного решения одной иерархической задачи оптимизации // Тезисы докладов II Международной школы-семинара “Нелинейный анализ и экстремальные задачи” (28 июня – 4 июля 2010 г., Иркутск). Иркутск, 2010. С. 50.
- [23] Малышев А.В. Локальный поиск гарантированного решения квадратично-линейной задачи двухуровневого программирования // Тезисы докладов конференции “Ляпуновские чтения & презентация информационных технологий” (Иркутск, 21–23 декабря 2009). Иркутск, 2009. С. 34.
- [24] Малышев А.В. О поиске гарантированного решения в квадратично-линейных двухуровневых задачах // IV Всероссийская конференция “Проблемы оптимизации и экономические приложения”: Материалы конференции (Омск, 29 июня – 4 июля 2009) / Омский филиал Института математики им. С.Л. Соболева СО РАН. Омск: Полиграфический центр КАН. С. 184.
- [25] Малышев А.В. О поиске гарантированного решения одной иерархической задачи оптимизации // Материалы Всероссийской конференции “Математическое моделирование и вычислительно-информационные технологии в междисциплинарных научных исследованиях” (Иркутск, 6–7 июня 2009). Иркутск, 2009.
- [26] Малышев А.В. Поиск гарантированного решения одного класса задач двухуровневого программирования // Труды VI Московской международной конференции по исследованию операций (ORM2010: Москва, 19–23 октября 2010). М.: МАКС Пресс, 2010. С. 239–240.
- [27] Малышев А.В. Численный поиск гарантированного решения в одной задаче двухуровневой оптимизации // Информационный бюллетень Ассоциации математического программирования. №12. “Тезисы докладов XIV Всероссийской конференции

Математическое программирование и приложения (Екатеринбург, 28 февраля – 4 марта 2011)”. Екатеринбург: УрО РАН, 2011. С. 47–48.

- [28] Малышев А.В., Орлов А.В. Иерархическая задача оптимального выбора тарифов телекоммуникационным оператором // Материалы школы-семинара молодых ученых “Информационные технологии и моделирование социальных эколого-экономических систем” (Иркутск — Ханх (Монголия), 1-6 октября 2008). Иркутск: ИДСТУ СО РАН, 2008. С. 43–45.
- [29] Малышев А.В., Орлов А.В. Поиск оптимистических решений в квадратично-линейных двухуровневых задачах // Материалы IX школы-семинара молодых ученых “Математическое моделирование и информационные технологии” (Иркутск — Ангасолка, 22–27 октября 2007). Иркутск, 2007. С. 107–110.
- [30] Малышев А.В., Орлов А.В. Численное решение квадратично-линейных двухуровневых задач в оптимистической постановке // Тезисы докладов конференции “Ляпуновские чтения & презентация информационных технологий” (Иркутск, 29–30 ноября 2007). Иркутск, 2007. С. 24.
- [31] Малышев А.В., Стрекаловский А.С. О взаимосвязи некоторых задач двухуровневой и нелинейной оптимизации // Известия вузов. Математика. 2011. №4. С. 99–103.
- [32] Малышев А.В., Стрекаловский А.С. Глобальный поиск гарантированных решений в квадратично-линейных задачах двухуровневой оптимизации // Известия Иркутского государственного университета. Математика. 2011. Т. 4, №1. С. 73–82.
- [33] Моисеев Н.Н. Элементы теории оптимальных систем. М.: Наука, 1975.
- [34] Молодцов Д.А. О решении одного класса неантагонистических игр // Журн. вычисл. матем. и матем. физики. 1976. Т. 16, №6. С. 1451–1456.
- [35] Молодцов Д.А., Федоров В.В. Аппроксимация игр двух лиц с передачей информации // Журн. вычисл. матем. и матем. физики. 1973. Т. 13, №6. С. 1469–1484.
- [36] Орлов А.В. Численное решение задач билинейного программирования // Журн. вычисл. матем. и матем. физики. 2008. Т. 48, №2. С. 45–62.

- [37] Орлов А.В., Малышев А.В. К решению квадратично-линейных двухуровневых задач // Российская конференция “Дискретная оптимизация и исследование операций”: Материалы конференции (Владивосток, 7 – 14 сентября 2007). Новосибирск: Изд-во Института математики, 2007. С. 110.
- [38] Орлов А.В., Стрекаловский А.С. О численном поиске ситуаций равновесия в биматричных играх // Журн. вычисл. матем. и матем. физики. 2005. Т. 45, №6. С. 983–997.
- [39] Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. М.: Наука, 1986.
- [40] Стрекаловский А.С. Минимизирующие последовательности в задачах с d.c. ограничениями // Журн. вычисл. математики и мат. физики. 2005. Т. 45, №3. С. 435–447.
- [41] Стрекаловский А.С. Об экстремальных задачах с d.c. ограничениями // Журн. вычисл. математики и мат. физики. 2001. Т. 41, №12. С. 1833–1843.
- [42] Стрекаловский А.С. О минимизации разности двух выпуклых функций на допустимом множестве // Журн. вычисл. матем. и матем. физики. 2003. Т. 43, №3. С. 399–409.
- [43] Стрекаловский А.С. Элементы невыпуклой оптимизации. Новосибирск: Наука, 2003.
- [44] Стрекаловский А.С., Малышев А.В. О взаимосвязи гарантированного и оптимистического решений задачи двухуровневого программирования // Алгоритмический анализ неустойчивых задач: тезисы докладов Международной конференции, посвященной 100-летию со дня рождения В.К. Иванова (Екатеринбург, 1–6 сентября 2008). Екатеринбург: Изд-во Уральского университета, 2008. С. 306–307.
- [45] Стрекаловский А.С., Малышев А.В. О поиске гарантированного решения в линейной задаче двухуровневого программирования // Тезисы докладов конференции “Ляпуновские чтения & презентация информационных технологий” (Иркутск, 19–23 декабря 2008). Иркутск, 2008. С. 24.

- [46] Стрекаловский А.С., Малышев А.В. О поиске гарантированного решения задачи двухуровневого программирования // Труды XIV Байкальской международной школы-семинара “Методы оптимизации и их приложения” (Иркутск, Байкал, 2–8 июля 2008). Иркутск: Изд-во ИСЭМ СО РАН, 2008. Т. 1. С. 602–608.
- [47] Стрекаловский А.С., Орлов А.В. Биматричные игры и билинейное программирование. М.: Физматлит, 2007.
- [48] Стрекаловский А.С., Орлов А.В., Малышев А.В. Локальный поиск в квадратично-линейной задаче двухуровневого программирования // Сибирский журнал вычислительной математики. 2010. Т. 13, №1. С. 75–88.
- [49] Стрекаловский А.С., Орлов А.В., Малышев А.В. Численное решение одного класса задач двухуровневого программирования // Сибирский журнал вычислительной математики. 2010. Т. 13, №2. С. 201–212.
- [50] Фаддеев Д.К. Лекции по алгебре: учеб. пособие. СПб.: Изд-во «Лань», 2005.
- [51] Шамардин Ю.В. О двухуровневой задаче размещения при ограничениях на объем производства // Дискретный анализ и исследование операций, серия 2. 2000. Т. 7, №2. С. 114–118.
- [52] Aboussoror A, Mansouri A. Weak linear bilevel programming problems: existence of solutions via a penalty method // Journal of Mathematical Analysis and Applications. 2005. Vol. 304, №1. P. 399–408.
- [53] Al-Khayyal F.A., Horst R., Pardalos P.M. Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming // Annals of Operations Research. 1992. Vol. 34, №1-4. P. 125–147.
- [54] Alexandrov N., Dennis J.E. Algorithms for Bilevel Optimization // AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization. 1994. P. 810–816.
- [55] Amouzegar M.A. A global optimization method for nonlinear bilevel programming problems // IEEE Transactions on Systems, Man, and Cybernetics, Part B. 1999. Vol. 29, №6. P. 771–777.

- [56] Anandalingam G., White D. A solution for the linear static Stackelberg problem using penalty function // IEEE Transactions Automatic Control. 1990. №35. P. 1170–1173.
- [57] Audet C., Hansen P., Jaumard B., Savard G. Links Between Linear Bilevel and Mixed 0–1 Programming Problems // Journal of Optimization Theory and Applications. 1997. Vol. 93, №2. P. 273–300.
- [58] Audet C., Savard G., Zghal W. New Branch-and-Cut Algorithm for Bilevel Linear Programming // Journal of Optimization Theory and Applications. 2007. Vol. 134, №2. P. 353–370.
- [59] Bard J.F. Convex two-level optimization // Mathematical programming. 1988. Vol. 40, №1. P. 15–27.
- [60] Bard J.F. Practical Bilevel Optimization. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1998.
- [61] Bard J.F. Some properties of the bilevel programming problem // Journal of Optimization Theory and Applications. 1991. Vol. 68, №2. P. 371–378.
- [62] Ben-Ayed O., Blair C.E., Boyce D.E., LeBlanc L.J. Construction of a real-world bilevel linear programming model of the highway network design problem // Annals of Operations Research. 1992. Vol. 34, №1-4. P. 219–254.
- [63] Calamai P., Vicente L. Generating Linear and Linear-Quadratic Bilevel Programming Problems // SIAM Journal on Scientific Computing. 1993. Vol. 14, №4. P. 770–782.
- [64] Calamai P., Vicente L. Generating Quadratic Bilevel Programming Test Problems // ACM Transactions on Mathematical Software. 1994. Vol. 20. P. 103–119.
- [65] Calvete H. I., Gale C. On the quasiconcave bilevel programming problem // Journal of Optimization Theory and Applications. 1998. Vol. 98, №3. P. 613–622.
- [66] Colson B. BIPA (Bilevel programming with approximation methods): Software guide and test problems. Technical Report CRT-2002-38. Technical Report, Centre de Recherche sur les Transports, Universite de Montreal, Montreal, QC, Canada, 2002.

- [67] Colson B., Marcotte P., Savard G. An overview of bilevel optimization // *Annals of operations research*. 2007. Vol. 153, №1. P. 235–256.
- [68] Colson B., Marcotte P., Savard G. A trust-region method for nonlinear bilevel programming: algorithm and computational experience // *Computational optimization and applications*. 2005. Vol. 30, №3. P. 211–227.
- [69] Candler W., Townsley R. A linear two-level programming problem // *Computers and Operations Research*. 1982. №9. P. 59–76.
- [70] DeMiguel V., Murray W. A local convergence analysis of bilevel decomposition algorithms // *Optimization and Engineering*. 2006. Vol. 7, №2. P.99–133.
- [71] Dempe S. A Bundle Algorithm Applied to Bilevel Programming Problems with Non-Unique Lower Level Solutions // *Comput. Optim. Appl.* 2000. Vol. 15, №2. P.145–166.
- [72] Dempe S. Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints // *Optimization*. 2003. Vol. 52, №3. P. 333–359.
- [73] Dempe S. First-order necessary optimality conditions for general bilevel programming problems // *Journal of Optimization Theory and Applications*. 1997. Vol. 95, №3. P.735–739.
- [74] Dempe S. *Foundations of Bilevel Programming*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002.
- [75] Dempe S., Bard J.F. Bundle Trust-Region Algorithm for Bilinear Bilevel Programming // *Journal of Optimization Theory and Applications*. 2001. Vol. 110, №2. P. 265-288.
- [76] Dempe S., Dutta J. Bilevel programming with convex lower level problems // *Optimization with Multivalued Mappings, Springer Optimization and Its Applications*. 2006. Vol. 2. P. 51–71.
- [77] Dempe S., Dutta J., Mordukhovich B.S. New necessary optimality conditions in optimistic bilevel programming // *Optimization: A Journal of Mathematical Programming and Operations Research*. 2007. Vol. 56, №5. P. 577–604.

- [78] Dempe S., Gadhi N. Necessary optimality conditions for bilevel set optimization problems // Journal of Global Optimization. 2007. Vol. 39, №4. P. 529–542.
- [79] Dempe S., Kalashnikov V.V., Kalashnykova N. Optimality conditions for bilevel programming problems // Optimization with Multivalued Mappings, Springer Optimization and Its Applications. 2006. Vol. 2. P. 3–28.
- [80] Dempe S., Richter K. Bilevel programming with knapsack constraints // Central European Journal of Operations Research. 2000. Vol. 8, №2. P. 93–107.
- [81] Edmunds T.A., Bard J.F. Algorithms for nonlinear bilevel mathematical programs // IEEE Transactions on Systems, Man, and Cybernetics. 1991. Vol. 21, №1. P. 83–89.
- [82] Etoa Etoa J.B. Solving convex quadratic bilevel programming problems using an enumeration sequential quadratic programming algorithm // Journal of Global Optimization. 2010. (In Print)
- [83] Faisca N.P., Dua V., Rustem B., Saraiva P.M., Pistikopoulos E. Parametric global optimisation for bilevel programming // Journal of Global Optimization. 2007. Vol. 38, №4. P. 609–623.
- [84] Falk J.E., Liu J. On bilevel programming, part I: general nonlinear cases // Mathematical Programming. 1995. Vol. 70, №1. P. 47–72.
- [85] FICOTM Xpress Optimization Suite. URL: <http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx> (дата обращения: 17.08.2010).
- [86] Fliege J., Vicente L.N. Multicriteria approach to bilevel optimization // Journal of Optimization Theory and Applications. 2006. Vol. 131, №2. P. 209–225.
- [87] GLPK (GNU Linear Programming Kit). URL: <http://www.gnu.org/software/glpk> (дата обращения: 17.08.2010).
- [88] Gumus Z.H., Floudas C.A. Global Optimization of Nonlinear Bilevel Programming Problems // Journal of Global Optimization. 2001. Vol. 20, №1. P. 1–31.

- [89] Hansen P., Jaumard B., Savard G. New branch-and-bound rules for linear bilevel programming // *SIAM Journal on Science and Statistical Computing*. 1992. Vol. 13, №5. P. 1194–1217.
- [90] Herskovits J, Leontiev A An interior point technique for solving bilevel programming problems. Technical report, Institute of Mathematics, Federal University of Rio de Janeiro, 2002.
- [91] Hoai An L.T. DC Programming for solving a class of global optimization problems via reformulation by exact penalty // *Proceedings of COCOS'2002*. P. 87–101.
- [92] Hoai An L.T., Tao P.D., Canh N.N., Thoai N. DC programming techniques for solving a class of nonlinear bilevel programs // *Journal of Global Optimization*. 2009. Vol. 44, №3. P. 313-337.
- [93] Horst R., Tuy H. *Global Optimization. Deterministic Approaches*. Berlin: Springer-Verlag, 1993.
- [94] Jaumard B., Savard G., Xiong X. An exact algorithm for convex bilevel programming. Technical Report G-95-33. Technical Report, Groupe d'études et de recherche en analyse des décisions, Université de Montréal, Montréal PQ, CANADA, 1995.
- [95] Jia F., Yang F., Wang S.-Y. Sensitivity analysis in bilevel linear programming // *Systems Science and Mathematical Sciences*. 1998. Vol. 11. P. 359–366.
- [96] Judice J.J., Faustino A.M., Ribeiro I.M., Neves A.S. On the use of bilevel programming for solving a structural optimization problem with discrete variables // *Optimization with Multivalued Mappings, Springer Optimization and Its Applications*. 2006. Vol. 2. P. 123–142.
- [97] Kolstad C.D., Lasdon L.S. Derivative evaluation and computational experience with large bilevel mathematical programs // *Journal of Optimization Theory and Applications*. 1990. Vol. 65, №3. P. 485–499.
- [98] Konnov I.V. *Equilibrium Models and Variational Inequalities. Mathematics in Science and Engineering*, Vol. 210. Amsterdam: Elsevier B.V., 2007.

- [99] Lignola M.B., Morgan J. Stability of regularized bilevel programming problems // Journal of Optimization Theory and Applications. 1997. Vol. 93, №3. P. 575–596.
- [100] Li H., Wang Y. A hybrid genetic algorithm for solving a class of nonlinear bilevel programming problems // Proceedings of 6th International Conference Simulated Evolution and Learning. Hetei, China, October 15-18, 2006. Eds: Wang T.D., Li X. et al. Springer. P. 408–415.
- [101] Lin L.-J. Mathematical programming with system of equilibrium constraints // Journal of Global Optimization. 2007. Vol. 37, №2. P.275–286.
- [102] Liu G.S., Han J.Y., Zhang J.Z. Exact penalty functions for convex bilevel programming problems // Journal of optimization theory and applications. 2001. Vol. 110, №3. P. 621–643.
- [103] Loridan P., Morgan J. Weak via strong Stackelberg problem: New results // Journal of Global Optimization. 1996. Vol. 8, №3. P. 263–287.
- [104] Luo Z.-Q., Pang J.-S., Ralph D. Mathematical programs with equilibrium constraints. Cambridge: Cambridge University Press, 1996.
- [105] Malyshev A.V., Strekalovsky A.S. Global search for pessimistic solution in bilevel problems // Proceedings of the Toulouse global optimization workshop 2010. Toulouse, France, 2010. P. 77–80.
- [106] Marcotte P., Savard G. Bilevel programming: a combinatorial perspective // Graph Theory and Combinatorial Optimization. Springer US. 2005. P. 191-217.
- [107] Meyer R.R. Continuity properties of linear programs. Technical Report 373. Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1979.
- [108] Mitsos A., Lemonidis P., Barton P.I. Global solution of bilevel programs with a nonconvex inner program // Journal of Global Optimization. 2008. Vol. 42, №4. P. 475–513.
- [109] Muu L.D., Quy N.V. A global optimization method for solving convex quadratic bilevel programming problems // Journal of global optimization. 2003. Vol. 26, №2. P. 199–219.

- [110] Nguyen T.Q., Bouhtou M., Lutton J-L. D.C. Approach to Bilevel Bilinear Programming Problem: Application in Telecommunications Pricing // Optimization and Optimal Control, International conference on optimization and optimal control, Ulaanbaatar, Mongolia, August 13–17, 2002. River Edge, NJ: World Scientific (ISBN 981-238-597-5/hbk). Series on Computers and Operations Research 1. P.211-231.
- [111] Norton R.D., Schiefer G.W. Agricultural sector programming models: a review of alternative approaches. Collaborative Paper, CP-80-30. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980.
- [112] Outrata J.V., Kocvara M., Zowe J. Nonsmooth approach to optimization problems with equilibrium constraints: Theory, applications and numerical results. Boston: Kluwer Acad. Publ., 1998.
- [113] Pang J.-S. Three modeling paradigms in mathematical programming // Mathematical Programming. 2010. Vol. 124, №2. P. 297–323.
- [114] Rajesh J., Gupta K., Kusumakar H.S., Jayaraman V.K., Kulkarni B.D. A Tabu Search Based Approach for Solving a Class of Bilevel Programming Problems in Chemical Engineering // Journal of Heuristics. 2003. Vol. 9, №4. P. 307–319.
- [115] Saboia C.H., Campelo M., Scheimberg S. A computational study of global algorithms for linear bilevel programming // Numerical Algorithms. 2004. Vol. 35, №2–4. P. 155–173.
- [116] Savard G., Gauvin J. The Steepest Descent Direction for the Nonlinear Bilevel Programming Problem. Technical Report G-90-37. Groupe d'études et de recherche en analyse des decisions, Universite de Montreal, Montreal PQ, CANADA, August 1990, Revised: January 1993.
- [117] Schelling T.C. The Strategy Of Conflict. Harvard University Press, 1980.
- [118] Stackelberg H.F. Marktform und Gleichgewicht. Berlin, Germany: Springer-Verlag, 1934. Перевод на англ.: The Theory of the Market Economy. Oxford University Press, 1952.

- [119] Strekalovsky A.S., Orlov A.V. A new approach to nonconvex optimization // Вычислительные методы и программирование. Издательство Московского университета. 2007. Т. 8, №2. P. 11–27.
- [120] Strekalovsky A.S., Orlov A.V., Malyshev A.V. On computational search for optimistic solutions in bilevel problems // Journal of Global Optimization. 2010. Vol. 48, №1. P. 159–172.
- [121] Tsoukalas A., Wiesemann W., Rustem B. Global Optimization of Pessimistic Bi-Level Problems // Lectures on global optimization / Ed. by P.M. Pardalos, T.F. Coleman. Toronto, Canada, 2009. Vol. 55. P. 215–243.
- [122] Tuy H., Migdalas A., Hoai-Phuong N.T. A novel approach to Bilevel nonlinear programming // Journal of Global Optimization. 2007. Vol. 38, №4. P. 527–554.
- [123] Unlu G. A linear bilevel programming algorithm based on bicriteria programming // Computers and Operations Research. 1987. Vol. 14, №2. P. 173–179.
- [124] Vicente L.N., Calamai P.H. Geometry and local optimality conditions for bilevel programs with quadratic strictly convex lower levels // Minimax and Applications / Du, D. and Pardalos, P. M., editors. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1995.
- [125] Vicente L., Savard G., Judice J. Descent approaches for quadratic bilevel programming // Journal of Optimization Theory and Applications. 1994. Vol. 81, №2. P. 379-399.
- [126] Vicente L., Savard G., Judice J. Discrete Linear Bilevel Programming Problem // Journal of Optimization Theory and Applications. 1996. Vol. 89, №3. P. 597-614.
- [127] Wang Y., Jiao Y.-C., Li H. An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme // IEEE Transactions on Systems, Man, and Cybernetics, Part A. 2005. Vol. 35, №2. P. 221–232.
- [128] Wang G.-M., Wan Z.-P., Wang X.-J. Genetic Algorithm for Solving Convex Quadratic Bilevel Programming Problem // Wuhan University Journal of Natural Sciences(EI). 2007. Vol. 12, №3. P. 421–425.

- [129] Wets R.J.-B. On the continuity of the value of a linear program and of related polyhedral-valued multifunctions // Mathematical Programming Essays in Honor of George B. Dantzig Part I. Mathematical Programming Studies. 1985. Vol. 24. P.14–29.
- [130] White D., Anandalingam G. A penalty function approach for solving bi-level linear programs // Journal of global optimization. 1993. Vol. 3. P. 397–419.
- [131] Yeza A. First-order necessary optimality conditions for general bilevel programming problems // Journal of Optimization Theory and Applications. 1996. Vol. 89, №1. P. 189–219.
- [132] Zhigljavsky A., Zilinskas A. Stochastic global optimization. Berlin: Springer, 2008.
- [133] Zhu X., Yu. Q., Wang X. A Hybrid Differential Evolution Algorithm for Solving Nonlinear Bilevel Programming with Linear Constraints // Proceedings of the Firth IEEE International Conference on Cognitive Informatics, ICCI 2006, July 17-19, Beijing, China, 2006. P. 126–131.
- [134] Ziena Optimization LLC. KNITRO. URL: <http://www.ziena.com/knitro.htm> (дата обращения: 26.02.2011).