

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

На правах рукописи

Петрова Елена Геннадьевна

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ  
ДОПОЛНИТЕЛЬНОСТИ И ДВУХУРОВНЕВОГО  
ПРОГРАММИРОВАНИЯ

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

Диссертация на соискание учёной степени  
кандидата физико-математических наук

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

Иркутск-2011

# Оглавление

Введение	4
<b>1 Оптимизационный подход к решению линейной задачи дополнительности</b>	<b>21</b>
1.1 Постановка задачи. Различные формулировки линейной задачи дополнительности . . . . .	22
1.2 Редукция к задаче d.c. минимизации . . . . .	27
1.3 Специальный метод локального поиска . . . . .	29
1.4 Построение тестовых примеров и апробация алгоритма локального поиска . . . . .	32
1.5 Алгоритм глобального поиска . . . . .	35
1.6 Вычислительный эксперимент . . . . .	38
1.6.1 Выбор аппроксимации поверхности уровня . . . . .	38
1.6.2 Согласование точностей локального и глобального поиска . . . . .	42
1.6.3 Решение серий задач . . . . .	43
1.7 Основные результаты главы . . . . .	46
<b>2 Поиск оптимистических решений в линейной двухуровневой задаче</b>	<b>47</b>
2.1 Постановка линейной двухуровневой задачи . . . . .	48
2.2 Сведение двухуровневой задачи к задаче с d.c. ограничением . . . . .	50
2.3 Условия глобальной оптимальности для задачи минимизации с d.c. ограничением-равенством . . . . .	52
2.4 Минимизирующие последовательности . . . . .	57
2.5 Стратегия глобального поиска . . . . .	60
2.6 Решение возмущенной задачи . . . . .	61

2.7	Локальный поиск в задаче с d.c. ограничением-равенством . . . . .	65
2.7.1	Генерация тестовых задач . . . . .	68
2.7.2	Тестирование метода локального поиска . . . . .	71
2.8	Глобальный поиск в линейной двухуровневой задаче . . . . .	76
2.9	Вычислительный эксперимент . . . . .	78
2.9.1	Первый этап. Выбор аппроксимации поверхности уровня . . . . .	78
2.9.2	Второй этап. Оценка сложности задач . . . . .	83
2.9.3	Третий этап. Решение задач высокой размерности . . . . .	85
2.10	Задача планирования производства в условиях неизвестного спроса . . . . .	86
2.11	Основные результаты главы . . . . .	89
<b>3</b>	<b>Методы решения уравнений с d.c. функциями</b>	<b>90</b>
3.1	Решение одного уравнения с d.c. функцией . . . . .	91
3.2	Доказательство сходимости ПВК . . . . .	97
3.3	Вычислительный эксперимент . . . . .	102
3.4	Решение систем нелинейных уравнений . . . . .	106
3.5	Особенности решения квадратичных систем уравнений . . . . .	110
3.6	Численное решение систем уравнений . . . . .	113
3.6.1	Квадратичные уравнения . . . . .	113
3.6.2	Нелинейные уравнения . . . . .	114
3.7	Основные результаты главы . . . . .	115
	<b>Заключение</b>	<b>116</b>
	<b>Список использованной литературы</b>	<b>117</b>

# Введение

В последние годы круг приложений методов нелинейной оптимизации неуклонно расширяется. Если прежде в задачах планирования и управления экономическими объектами использовалось, как правило, линейное программирование, то теперь все чаще в экономико-математических исследованиях возникают нелинейные экстремальные задачи [60, 63]. При этом, по мнению ряда специалистов, наиболее сложными для решения являются оптимизационные задачи с нелинейными ограничениями-равенствами, в которых зачастую затруднительно нахождение даже допустимой точки [84, 110, 126]. Однако необходимость решения таких задач на практике побуждает исследователей к разработке эффективных численных методов.

Методы решения экстремальных задач с равенствами многочисленны и разнообразны. Их можно классифицировать как по формальным признакам, так и по содержательным. Так, выделяются методы нулевого, первого и второго порядков в зависимости от порядка используемых производных [3, 9, 15, 57]. Методы также делятся на прямые (в которых итерации ведутся в пространстве прямых переменных) [3, 4, 9, 57, 84] и двойственные (которые существенно используют двойственные переменные) [5, 9, 57]. Во многих методах на каждом шаге решается некоторая вспомогательная задача, и, с вычислительной точки зрения, удобно вести классификацию по ее типу. Это может быть задача безусловной минимизации, задача минимизации линейной или квадратичной функции при линейных ограничениях и т.д. Кроме того, одни методы направлены на поиск глобального решения задачи, другие же позволяют найти только локальный экстремум. Наконец, сами идеи, лежащие в основе методов, весьма различаются. Выделим основные направления в развитии данных методов.

В задачах небольшой размерности для нахождения стационарных точек, которые могут быть описаны системой Лагранжа [31], можно применять, например, метод Ньютона и его различные модификации [25, 45], а также методы безусловной минимизации [9, 57, 126] после сведения системы к оптимизационной задаче. Принципиальным недо-

статком такого подхода является то, что при этом пропадает оптимизационная сущность самой задачи, и, в частности, локальные максимумы и минимумы при использовании данного подхода не различаются.

Методы типа приведенных градиентов [15], получающиеся обобщением стандартных подходов к решению задач с линейными ограничениями на нелинейный случай, неплохо работают для задач, ограничения которых "почти линейны". Если же нелинейность ограничений значительна и начальная точка далека от оптимальной, то продвижение к решению будет осуществляться очень малыми шагами.

В методе линеаризации на каждой итерации минимизируемая функция и ограничения линеаризуются [9, 15, 57]. Поскольку задача минимизации линейной функции при линейных ограничениях может не иметь решения, в функцию вводится выпуклая квадратичная добавка, которая позволяет избежать этой ситуации.

На подобной идее основаны популярные в настоящее время методы последовательного квадратичного программирования (SQP-методы) [15, 31, 126], заключающиеся в решении задач выпуклого квадратичного программирования, аппроксимирующих исходную задачу оптимизации. Правильно выбранная задача квадратичного программирования оказывается достаточно адекватной локальной аппроксимацией исходной задачи. В то же время квадратичная задача достаточно проста, и для нее существуют эффективные методы решения. На сегодняшний день SQP-методы входят в число наиболее эффективных методов условной оптимизации [31, 126].

Другим направлением в решении задач с ограничениями-равенствами являются двойственные методы [57] (например, метод Эрроу-Гурвица), в которых одновременно происходит минимизация функции Лагранжа по  $x$  и максимизация по двойственным переменным  $y$ . Другим двойственным подходом является метод модифицированной функции Лагранжа [5], который основан на добавлении гладкого штрафного слагаемого к функции Лагранжа минимизируемой функции, в результате чего получается так называемое семейство модифицированных функций Лагранжа.

Однако все утверждения относительно сходимости перечисленных методов носят локальный характер, т.е. гарантируют нахождение лишь локального экстремума. Исключением является метод штрафов и различные его модификации [9, 36, 57, 60, 84, 126], сходимость которых носит глобальный характер. Метод штрафных функций является одним из наиболее простых и широко известных методов решения задач математического программирования. Основная его идея состоит в приближенном сведении задачи мини-

мизации с ограничениями к задаче безусловной минимизации некоторой функции. При этом вспомогательная функция подбирается так, чтобы она совпадала с заданной минимизируемой функцией внутри допустимой области и быстро возрастала вне ее. Здесь все трудности перенесены на этап решения вспомогательных задач безусловной минимизации, которые обычно оказываются невыпуклыми. Основным же недостатком метода состоит в том, что параметр штрафа должен бесконечно увеличиваться, а с ростом параметра задача становится все хуже обусловленной: целевая функция приобретает все более "овражный" характер.

Другой подход в области штрафных функций заключается в таком построении вспомогательных функций, что для соответствующего выбора параметра однократная безусловная оптимизация дает решение исходной задачи. Это — так называемый метод точной штрафной функции [22, 26, 27, 28]. Однако точные штрафные функции, как правило, оказываются недифференцируемыми, что придает дополнительную сложность при их минимизации.

Наконец, все большую популярность для поиска глобального решения невыпуклых задач, в том числе задач с равенствами, приобретают методы ветвей и границ, отсечений, аппроксимаций и т.д., идеи которых заимствованы из дискретной оптимизации [63, 116, 124, 129, 139]. К настоящему моменту предложено огромное количество алгоритмов, использующих различные варианты отыскания оценок и построения дерева поиска в задачах с равенствами. Одним из недостатков этих подходов является так называемое "проклятие размерности", которое означает, что с увеличением размерности объем вычислений по этим методам возрастает экспоненциально [129, 139], что не позволяет найти именно глобальное решение.

В целом можно констатировать, что проблема поиска глобального решения задач с ограничениями-равенствами очень сложна, и удовлетворительных универсальных способов ее решения в настоящее время не существует. Поэтому наиболее приоритетным направлением развития методов численного решения невыпуклых задач, по оценке многих специалистов [9, 57], оказывается исследование и решение специальных классов задач с учетом их специфики и свойств. Настоящая диссертационная работа лежит в указанном направлении и посвящена разработке алгоритмов решения ряда классов невыпуклых задач.

Достаточно сложно привести полную классификацию задач невыпуклой или, как еще принято говорить, глобальной оптимизации. Тем не менее, в литературе [110, 112] при-

нято рассматривать следующие основные постановки задач.

1. D.C. минимизация

$$F(x) = g(x) - h(x) \downarrow \min, \quad x \in D, \quad (1)$$

где  $g(\cdot)$ ,  $h(\cdot)$  — выпуклые функции на некотором открытом выпуклом множестве  $\Omega \subset \mathbb{R}^n$ ,  $D \subset \Omega$ , т.е.  $F(\cdot)$  — d.c. функция.

2. Экстремальные задачи с d.c. ограничениями, простейшей из которых является задача следующего вида

$$f(x) \downarrow \min, \quad x \in D, \quad F(x) \leq 0, \quad (2)$$

где  $F(x) = g(x) - h(x)$ ,  $x \in \Omega$ , а  $h(\cdot)$ ,  $g(\cdot)$ ,  $f(\cdot)$  являются выпуклыми функциями на выпуклом открытом множестве  $\Omega \subset \mathbb{R}^n$ , содержащем множество  $D$ ,  $F(\cdot)$  — непрерывная функция на  $\mathbb{R}^n$ .

Отметим, что в случае, когда функция  $g(\cdot)$  тождественно равна нулю, то в первом случае получаем задачи выпуклой максимизации [66, 69], а во втором — обратно-выпуклые задачи [69, 71].

Понятие d.c. функции (функции, представимой в виде разности двух выпуклых функций) является одним из базовых в невыпуклой оптимизации.

Пионером в изучении свойств d.c. функций является А.Д. Александров [1]. Позже этой проблемой занимались Е.М. Ландис [40] и П. Хартман [105]. Некоторые более поздние работы по d.c. структурам можно найти в [107, 128].

Хотя исследование задач d.c. оптимизации (исключая частный случай — выпуклую максимизацию) началось сравнительно недавно, не более 40 лет назад, к настоящему моменту достигнуты определенные успехи. Во-первых, предложены условия глобальной оптимальности [65, 108, 109, 134], использующие современный аппарат выпуклого анализа [37, 59]. Кроме того, разрабатывается теория двойственности [107, 111, 137], а также связанные с характеристикой оптимума и двойственностью задач различные условия регулярности [111, 137].

В частности, на основе предложенных А.С. Стрекаловским условий глобальной оптимальности для представленных выше классов задач [67, 69, 70] были разработаны алгоритмы глобального поиска для решения многих актуальных теоретических и прикладных невыпуклых задач оптимизации [19, 44, 66, 68, 72, 78].

Стратегия глобального поиска состоит из двух основных этапов: а) локального поиска и б) процедуры выхода из локального экстремума (критической точки), включающей в себя построение аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в исследуемой задаче, решение линеаризованных задач и ряд других [67, 69, 70].

В данной работе, основываясь на разработанных стратегиях глобального поиска, предпринимается попытка подойти к решению более сложных задач следующего вида:

$$f(x) \downarrow \min, \quad x \in S, \quad F(x) = 0, \quad (3)$$

где  $F(x) = g(x) - h(x)$ ,  $g(\cdot)$ ,  $h(\cdot)$ ,  $f(\cdot)$  — выпуклые функции.

В работе предложены алгоритмы глобального поиска в задачах вида (3), базирующиеся на теории глобального поиска А.С. Стрекаловского. [67, 69, 70, 72].

В соответствии с упомянутым приоритетным направлением в развитии методов невыпуклой оптимизации (а именно, построении специальных методов для отдельных классов экстремальных задач) в работе рассмотрены частные случаи задач оптимизации с равенствами.

Были выбраны следующие **объекты диссертационного исследования**:

- линейная задача дополненности;
- линейная двухуровневая задача;
- одно уравнение с d.c. функцией и системы уравнений с d.c. функциями.

Как известно [88, 99, 100], линейная задача дополненности (ЛЗД) содержит в себе комплементарное ограничение-равенство. Следующий же объект — линейная двухуровневая задача — может быть проинтерпретирован как некоторое усложнение предыдущей ЛЗД, поскольку после сведения двухуровневой задачи к оптимизационной, мы получаем задачу с линейной целевой функцией и схожими нелинейными комплементарными ограничениями. Для получения же допустимой точки в задачах оптимизации с равенствами требуется нахождение решения уравнения или системы уравнений.

**Актуальность темы диссертации** обусловлена большим количеством практических приложений каждого из упомянутых объектов. Наличие же неявных и явных ограничений-равенств создает серьезные трудности на пути отыскания глобального решения в поставленных задачах.



**Основной целью диссертационного исследования** является разработка эффективных алгоритмов численного решения указанных задач.

Далее представим более подробно постановки задач и их особенности.

В **первой главе** диссертации исследуется **линейная задача дополнителности** (ЛЗД) [88, 99, 100, 101, 104, 113, 123, 125, 127, 131, 135, 117], состоящая в нахождении пары векторов  $(x, w)$ , удовлетворяющих следующим условиям:

$$\left. \begin{aligned} Mx + q &= w, \\ \langle x, w \rangle &= 0, \\ x \geq 0, \quad w &\geq 0, \end{aligned} \right\} \quad (4)$$

где  $x, w \in \mathbb{R}^n$ , а вектор  $q \in \mathbb{R}^n$  и вещественная, подчеркнем, знаконеопределенная  $(n \times n)$ -матрица  $M$  заданы. Нетрудно видеть, что задачу (4) можно также записать в следующем виде:

$$\left. \begin{aligned} \langle x, Mx + q \rangle &= 0, \\ x \geq 0, \quad Mx + q &\geq 0. \end{aligned} \right\} \quad (5)$$

Теория линейной дополнителности, впервые представленная в работах Коттла [99], Данцига [101], Лемке [122], интенсивно развивается три последних десятилетия, главной причиной чего служит связь ЛЗД с задачами линейного и квадратичного программирования [58]. Исторически задачу ЛЗД можно рассматривать в качестве объединяющей формулировки для задач линейного, квадратичного программирования и биматричных игр [100].

Отметим также, что задачи дополнителности тесным образом связаны с другим важнейшим объектом современной математики — вариационными неравенствами [104], имеющими широкие приложения в математической физике, в теории равновесий, а также в других областях.

Решение ЛЗД в общем случае является нетривиальной задачей. Как известно [97], получение ответа на вопрос, имеет ли решение ЛЗД с целочисленными коэффициентами, оказывается NP-полной задачей [131]. Поэтому наиболее эффективными являются методы, ориентированные на использование свойств матрицы  $M$ , т.е. на отдельные классы ЛЗД. Таковыми могут быть, в частности, задачи с матрицами, имеющими неотрицательные главные миноры (или  $P_0$ -матрицами) [99, 100]. Другим подобным классом могут быть задачи с матрицами, имеющими неположительные внедиагональные элементы (или  $Z$ -матрицами). Тогда можно построить эффективные методы решения как для ЛЗД, так и для ее нелинейных обобщений (напр., [104, 135]). Именно поэтому наибольшее количество

работ традиционно посвящено методам решения ЛЗД с неотрицательно определенными матрицами и с матрицами, имеющими положительные главные миноры. В качестве приоритетных можно выделить два основных направления развития.

Первое направление — это методы крайних точек ("pivoting methods"), являющиеся, по сути, разнообразными модификациями метода Лемке-Хаусона [100]. Процедура поиска решения основана на переборе крайних точек многогранного множества

$$S = \{(x, w) \in \mathbb{R}^{n+n} : w = q + Mx, x \geq 0, w \geq 0\}. \quad (6)$$

Каждая итерация метода соответствует движению из крайней точки множества  $S$  вдоль некоторого его ребра, почти удовлетворяющего условиям дополненности, т.е. ребра, на котором  $x_i w_i \neq 0$  только для одного значения индекса  $i \in \{1, \dots, n\}$ .

Этот метод является конечным вследствие конечного (но, возможно, очень большого) числа крайних точек многогранного множества. Однако нахождение решения ЛЗД (в случае его существования) гарантировано только при определенных условиях, налагаемых на матрицу  $M$  [58], например, при положительности всех ее главных миноров. Кроме того, данные методы эффективны в основном на задачах малой и средней размерности. С ростом размерности задачи резко увеличивается количество перебираемых вершин (как известно, количество вершин многогранного множества возрастает экспоненциально).

Второй подход к решению ЛЗД (который применяется в диссертационной работе) — это вариационный метод, заключающийся в минимизации некоторой целевой функции, например, скалярного произведения  $\langle x, w \rangle$  на допустимом множестве  $S$ . Для решения такой задачи применяются, например, методы внутренних точек [23, 24, 30, 120] и специализированные методы квадратичного программирования [3, 16, 81, 126]. При решении вариационной задачи, как и в первом случае, успешность применяемых методов зависит от свойств матрицы  $M$ . Так, в случае знаконеопределенности матрицы  $M$  (что порождает невыпуклость в задаче), возникают проблемы с поиском глобального решения, поскольку стандартные методы выпуклой оптимизации в этом случае позволяют найти, в лучшем случае, локальное решение, а чаще — лишь стационарную точку, которая может быть далека от глобального оптимума.

Приведем пример прикладной задачи линейной дополненности.

### Пример 0.0.1 Равновесие рынка

Рассмотрим следующую модель рыночного равновесия [100]:

Поведение производственной стороны описывается задачей линейного программирования:

$$\left. \begin{aligned} \langle c, x \rangle \downarrow \min_x, \\ Ax \geq b, \quad Bx \geq r, \\ x \geq 0. \end{aligned} \right\} \quad (7)$$

Здесь  $x$  — количество производимого товара,  $c$  — вектор затрат на производство, неравенство  $Ax \geq b$  описывает технологические ограничения,  $B$  — матрица ограничения на спрос, вектор  $r$  задает количество потребления.

Поведение потребителей характеризуется следующим образом:

$$r = Dp + d, \quad (8)$$

где  $p$  — вектор цен, функция  $Q(p) = Dp + d$  — вектор-функция потребления, описывающая зависимость спроса от рыночных цен.

При этом должны выполняться условия равновесия:

$$p = y^*, \quad (9)$$

где  $y^*$  — двойственный вектор теневых (скрытых) цен, который может быть найден как решение двойственной [10] к (7) задачи, которая имеет следующий вид:

$$\left. \begin{aligned} \langle b, v \rangle + \langle Dp + d, y \rangle \uparrow \max_y, \\ vA + yB \leq c, \\ v \geq 0, \quad y \geq 0. \end{aligned} \right\} \quad (10)$$

Тогда по теореме двойственности [10] или применив теорему Каруша-Куна-Таккера к задачам (7) и (10), получаем, что нахождение решения задач (7),(10) с учетом (8) эквивалентно решению системы

$$\left. \begin{aligned} \langle x, c - A^T v - B^T y \rangle = 0, \\ x \geq 0, \quad c - A^T v - B^T y \geq 0, \\ \langle v, -b + Ax \rangle = 0, \\ v \geq 0, \quad -b + Ax \geq 0, \\ \langle y, -d - Dp + Bx \rangle = 0, \\ y \geq 0, \quad -d - Dp + Bx \geq 0. \end{aligned} \right\} \quad (11)$$

Отметим, что в системе (11) присутствуют нелинейные ограничения-равенства. При этом

нетрудно видеть, что система (11) имеет вид ЛЗД с переменной  $z = (x, v, y)^T$ , где

$$q = \begin{pmatrix} c \\ -b \\ -d \end{pmatrix}, \quad M = \begin{pmatrix} 0 & -A^T & -B^T \\ A & 0 & 0 \\ B & 0 & -D \end{pmatrix}. \quad (12)$$

В этом случае свойства матрицы  $M$ , в частности, симметричность и положительная определенность, напрямую зависят от вида матрицы  $-D$ , которая характеризует поведение потребителей и может, вообще говоря, иметь произвольный вид. Таким образом, в общем случае получившаяся ЛЗД имеет закононеопределенную матрицу  $M$ . #

Кроме того, к задачам линейной дополнителности часто сводятся различные технические и экономические задачи. В [100] авторы рассматривают следующие постановки: задача торможения, контактная задача, задача об опорном подшипнике, задача упруго-пластического кручения, задача об оптимальном постоянном основном капитале и т.д. В последнее время ЛЗД широко применяется для физического моделирования в компьютерных играх [121]. При этом в некоторых задачах матрица  $M$  является закононеопределенной.

С целью преодоления невыпуклости вариационной задачи в первой главе применяется теория глобального поиска [69], [70] для минимизации целевой функции, представимой в виде разности двух выпуклых функций (d.c. функции).

В §1.1 представлены различные формулировки ЛЗД, показана ее связь с задачами линейного и квадратичного программирования, с биматричными играми. Далее, в §1.2 производится сведение исходной задачи к задаче d.c. минимизации. Параграфы 1.3–1.4 посвящены методу локального поиска в задаче линейной дополнителности. В §1.5 предлагается алгоритм глобального поиска, заключающийся в последовательном решении более простых подзадач и основанный на условиях глобальной оптимальности. Наконец, в §1.6 представлены результаты численного эксперимента на серии случайно сгенерированных задач. Здесь решается вопрос выбора аппроксимации поверхности уровня выпуклой функции, позволяющей более эффективно находить глобальные решения, вопрос согласования точностей локального и глобального поиска. Далее проводится вычислительный эксперимент на задачах повышенной размерности.

**Вторая глава** посвящена численному решению одного класса задач двухуровневого программирования.

В настоящее время задачи с иерархической структурой, возникающие на практике при исследовании сложноорганизованных систем в технике и экономике, являются весьма

популярным объектом для математического исследования [11, 89, 90, 92, 96, 98, 103, 114, 132, 136, 138], популярность которого объясняется прежде всего широким полем приложений [90]. С другой стороны, иерархическая структура задач и связанная с ней скрытая невыпуклость даже в простейшем линейном случае [103] вызывают трудности исследования таких задач.

В работе исследуется непрерывная двухуровневая задача с линейными целевыми функциями на верхнем и нижнем уровнях [90, 103]. При этом предполагается, что из всех возможных решений на нижнем уровне выбирается то, которое благоприятствует достижению цели верхнего уровня. Такая постановка двухуровневой задачи носит название оптимистической (кооперативной) [103]. Будем рассматривать линейно-линейную двухуровневую задачу в следующей постановке:

$$(\mathcal{BP}) \quad \left. \begin{aligned} f(x, y) &\triangleq \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y}, \\ x \in X &\triangleq \{x \in \mathbb{R}^m \mid Ax \leq b\}, \\ y \in Y_*(x) &\triangleq \operatorname{Arg} \min_y \{\langle d^1, y \rangle \mid y \in Y(x)\}, \end{aligned} \right\} \quad (13)$$

где множество  $Y(x)$  определено системой неравенств:

$$Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1 x + B_1 y \leq b^1\}, \quad (14)$$

$c \in \mathbb{R}^m$ ,  $d, d^1 \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^p$ ,  $b^1 \in \mathbb{R}^q$  — заданные векторы,  $A$ ,  $A_1$ ,  $B_1$  — матрицы размера  $(p \times m)$ ,  $(q \times m)$ ,  $(q \times n)$  соответственно.

Несмотря на внешнюю простоту, задача  $(\mathcal{BP})$ –(13) оказывается весьма сложной для решения. В §2.2 представлен пример, иллюстрирующий невыпуклость линейной двухуровневой задачи в самом простейшем случае.

За три десятилетия интенсивного исследования непрерывных двухуровневых задач различных классов было предложено достаточно много методов их решения (см., например, обзоры [98, 103]).

В линейно-линейном случае двухуровневая задача обладает тем свойством, что хотя бы одно ее глобальное решение достигается в крайней точке допустимого множества, поэтому широкий класс методов решения таких двухуровневых задач базируется на переборе вершин допустимого множества. Первые результаты в этом направлении были опубликованы в [96], а затем получили развитие в [102, 136] и др.

Еще одним интенсивно развивающимся направлением является разработка методов спуска, предназначенных для поиска стационарных точек и локальных минимумов в двухуровневых задачах [132].

Наиболее популярным подходом к решению двухуровневых задач является разработка методов, основанных на замене задачи нижнего уровня ее условиями оптимальности (что возможно в случае выпуклой, в частности, линейной, целевой функции нижнего уровня по своей переменной) [103]. В результате вместо двухуровневой получаем эквивалентную ей одноуровневую задачу, но содержащую в своей структуре невыпуклое ограничение-равенство, которое и создает сложности при ее решении. В этом случае мы имеем дело с задачей, принадлежащей классу задач с комплементарными ограничениями, специальная структура которых создает трудности ее эффективного численного решения [32]. Тем не менее, учитывая комплементарность множителей Лагранжа и ограничений задачи нижнего уровня, можно предложить различные варианты метода ветвей и границ [89, 130]. Один из подходов в этом случае также составляют так называемые методы релаксации, в которых ограничение-равенство параметрически возмущается так, чтобы получаемая задача могла обладать свойствами регулярности ограничений [32].

Другой подход к решению двухуровневых задач — решение последовательности вспомогательных задач линейной дополнителности (см., например, [114]), что, по сути, представляет собой симбиоз методов крайних точек и ветвей и границ.

Кроме того, при использовании подхода, основанного на сведении двухуровневой задачи к одноуровневой, для решения последней часто применяется метод штрафов [94], хотя в силу невыпуклости оштрафованной целевой функции, такой подход обоснован только лишь для поиска локальных экстремумов [95]. В то же время уже удалось успешно использовать данный метод для численного поиска глобального решения [79, 80] в двухуровневых задачах с квадратичной целевой функцией на верхнем и линейной целевой функцией на нижнем уровнях.

Насколько можно судить по доступной литературе, размерность, которую приводят авторы публикаций при тестировании предлагаемых алгоритмов, является недостаточной для решения практических задач. Эффективно решаются только линейно-линейные задачи средней размерности. Так, в [89] приведены результаты решения таких задач с суммарным числом переменных до 130, а в [130] — до 200.

В качестве экономического приложения двухуровневой задачи можно привести следующий пример.

### **Пример 0.0.2 Планирование производства в условиях неизвестного спроса [90]**

Предположим, что производитель желает определить, в каких объемах ему следует производить  $n$  товаров в каждый из рассматриваемых  $T$  периодов. Вектор объемов

производства обозначим через  $x \in \mathbb{R}^{n \times T}$ . Задача производителя — максимизировать прибыль в условиях  $q$  ресурсных ограничений. Предполагается, что производитель работает только с одним потребителем (например, производство комплектующих деталей для последующей сборки автомобилей на автозаводе). Вектор спроса в  $t$ -й момент не известен точно, а лежит во множестве  $Y_t$ , границы которого зависят от его затрат на рекламу, которые выражаются вектором  $v \in \mathbb{R}^{n \times T}$ .

В модели будем использовать следующие обозначения.

### Параметры

$a_{ijt}$  — количество ресурса  $i$ , необходимого для производства единицы продукции  $j$  в момент  $t$ ,  $i = 1, \dots, q$ ,  $j = 1, \dots, n$ ,  $t = 1, \dots, T$ ;

$b_{it}$  — количество ресурса  $i$ , доступное в момент  $t$ ;

$p_{jt}$  — цена продажи продукта  $j$  в момент  $t$ ;

$c_{jt}$  — стоимость производства продукта  $j$  в момент  $t$ ;

$h_{jt}$  — цена хранения единицы продукта  $j$  в момент  $t$ ;

$s_{jt}$  — цена производства единицы продукта  $j$  по субконтракту в момент  $t$ ;

$d_j$  — место, требуемое для хранения единицы продукции  $j$ ;

$l_t$  — свободное место под хранение, имеющееся в момент  $t$ .

### Переменные

$x_{jt}$  — количество продукта  $j$ , производимого в момент  $t$ ;

$v_{jt}$  — затраты на рекламу продукта  $j$  в момент  $t$ ;

$z_{jt}$  — нехватка продукта  $j$  в момент  $t$ ;

$w_{jt}$  — избыток продукта  $j$  в момент  $t$ ;

$y_{jt}$  — спрос на продукт  $j$  в момент  $t$ .

### Множества

$V_j$  — ограничения на рекламные расходы на продукт  $j$ ;

$Y_t(v)$  — множество, содержащее вектор спроса  $y_t$  в момент  $t$ .

Верхний уровень управляет значениями переменных  $x_{jt}$  и  $v_{jt}$ , а нижний — переменными  $z_{jt}$ ,  $w_{jt}$  и  $y_{jt}$ .

Таким образом, задача верхнего уровня имеет следующий вид:

$$F(x, v, z, w, y) = \sum_{j=1}^n \sum_{t=1}^T (c_{jt}x_{jt} + h_{jt}w_{jt} + s_{jt}z_{jt} + v_{jt}) - \sum_{j=1}^n \sum_{t=1}^T p_{jt}y_{jt} \downarrow \min, \quad (15)$$

$$\sum_{j=1}^n a_{ijt}x_{jt} \leq b_{it} \quad \forall i \in \{1, \dots, q\}, t \in \{1, \dots, T\}, \quad (16)$$

$$x_{jt} \geq 0, \quad v_j \in V_j \quad \forall j \in \{1, \dots, n\}, t \in \{1, \dots, T\}, \quad (17)$$

$$(y, w, z) \in \text{Sol}((19) - (23)), \quad (18)$$

в то время как задача нижнего уровня формализуется следующим образом:

$$f(y) = \sum_{j=1}^n \sum_{t=1}^T p_{jt} y_{jt} \downarrow \min_{y, w, z}, \quad (19)$$

$$w_{jt} - w_{j,t-1} + y_{jt} - z_{jt} = x_{jt} \quad \forall j \in \{1, \dots, n\}, t \in \{1, \dots, T\}, \quad (20)$$

$$l_t - \sum_{j=1}^n d_j w_{jt} \geq 0 \quad \forall t \in \{1, \dots, T\}, \quad (21)$$

$$z_{jt} \geq 0, \quad w_{jt} \geq 0, \quad w_{j0} = 0 \quad \forall j \in \{1, \dots, n\}, t \in \{1, \dots, T\}, \quad (22)$$

$$y_t \in Y_t(v) \quad \forall t \in \{1, \dots, T\}. \quad (23)$$

Неравенства (16) задают ресурсные ограничения. Ограничения на рекламные затраты заданы в (18) и влияют на спрос посредством (23). Уравнение (20) отражает материальный баланс. Начальные запасы полагаются нулевыми (формула (22)). Если  $V_j$  и  $Y_t$  — многогранные множества, задача (15)–(23) является задачей линейного двухуровневого программирования. #

В диссертационной работе производится разработка эффективного метода линейной двухуровневой оптимизации, позволяющего численно решать задачи высокой размерности. Для этого используется подход, основанный на редукции двухуровневой задачи в оптимистической постановке к задаче оптимизации с билинейным ограничением-равенством. Как известно, билинейная функция может быть представлена в виде разности двух выпуклых функций [44, 78], поэтому для решения получившейся задачи оптимизации в работе применяется теория глобального поиска для задач с d.c. ограничением-равенством.

В §2.1 дается постановка задачи и обсуждается ее так называемая скрытая невыпуклость. В §2.2 производится сведение двухуровневой задачи к оптимизационной задаче с d.c. ограничением-равенством. §2.3 посвящен условиям глобальной оптимальности для задач с d.c. равенством в общем виде. Затем §2.4 обобщает условия глобальной оптимальности на случай минимизирующих последовательностей. В §2.5–§2.6 описаны общие концепции глобального поиска для задачи с d.c. ограничением.

В §2.6 обсуждается возможность применения предложенной выше теории для решения линейной двухуровневой задачи. Для выполнения условий регулярности вводится возмущенная задача и изучается ее связь с исходной двухуровневой задачей.



В §2.7 представлена процедура локального поиска для линейной двухуровневой задачи, учитывающая ее специфику. Здесь же описана методика генерации тестовых задач и приведены результаты тестирования метода локального поиска.

В §2.8 предлагается модификация алгоритма глобального поиска для линейной двухуровневой задачи, а в §2.9 проводится многоэтапное численное тестирование предложенного алгоритма, в процессе которого выбираются необходимые параметры алгоритма, а также отмечаются наиболее сложные для решения задачи. В результате удается решить двухуровневые задачи размерности до 500 переменных на каждом уровне.

В §2.10 приводятся результаты работы алгоритма глобального поиска для решения практической задачи планирования производства в условиях неизвестного спроса.

**Третья глава** посвящена поиску точек, удовлетворяющих ограничениям равенствам. Во второй главе при применении теории глобального поиска возникает необходимость нахождения допустимой по ограничению-равенству точки с использованием двух векторов, в которых функция принимает значения противоположных знаков. Решению этого вопроса посвящена первая часть главы, в которой предлагаются новый метод решения нелинейных уравнений со многими неизвестными:

$$g(x) - h(x) = 0, \quad x \in \mathbb{R}^n, \quad (24)$$

где  $g(x), h(x)$  — выпуклые на  $\mathbb{R}^n$  функции. Необходимость решения одного уравнения со многими неизвестными возникает, например, в случае построения допустимой точки в задаче оптимизации с ограничением-равенством (см. §2.3). При этом разработанный метод позволяет находить специальную допустимую точку, наиболее подходящую, с точки зрения целевой функции. В одномерном случае это означает нахождение минимального корня уравнения, что находит широкое приложение во многих прикладных задачах: фазового детектирования, частотно-временного анализа, теории фильтров (см., например, [62]).

Далее в третьей главе рассматривается задача поиска точки, удовлетворяющей системе уравнений с д.с. функциями:

$$f_i(x) = g_i(x) - h_i(x) = 0, \quad i = 1, \dots, m, \quad (25)$$

где  $g_i(x), h_i(x)$  — выпуклые на  $\mathbb{R}^n$  функции,  $i = 1, \dots, m$ . Отметим, что в работе не ставится задача поиска всех корней системы уравнений. Результаты по решению задачи в такой постановке можно найти, например, в [6, 7].

Несмотря на широкий спектр методов [4, 12, 25, 45], разработанных в этой области, проблема численного поиска решений систем нелинейных уравнений остается весьма актуальной.

Так, при применении методов типа Ньютона возникает трудность, заключающаяся в выборе подходящего начального приближения, обеспечивающего сходимость к решению [4, 9, 45, 64, 83]. Причем с ростом размерности системы сложность поиска стартовой точки многократно возрастает. Большие трудности при применении Ньютоновских методов также создает наличие в системе кратных корней [8].

В случае же использования вариационных методов, мы имеем дело с невыпуклой экстремальной задачей, а для ее решения неприемлемы стандартные методы выпуклой оптимизации, имеющие свойство отыскивать в лучшем случае лишь локальные решения или стационарные точки, которые могут быть весьма далеки от глобального решения [9, 57, 69, 81].

Один из способов решения систем уравнений заключается в следующем. Строится функция, минимум которой достигается на решении системы. Затем, начиная с некоторого стартового приближения к точке минимума, проводятся итерации каким-либо из методов минимизации. Таким путем получается удовлетворительное приближение к решению системы. Далее, исходя из этого приближения, производится уточнение при помощи какого-либо итерационного метода, предназначенного именно для решения систем уравнений, например, метода Ньютона [25, 45]. Применение методов минимизации на первоначальном этапе вызвано тем, что обычно они имеют более широкую область сходимости, чем методы, направленные на решение систем уравнений. В то же время последние обычно обладают лучшей скоростью сходимости при наличии достаточно хорошего начального приближения [4].

В последние десятилетия разработаны и успешно применяются на практике интервальные методы для доказательного глобального решения уравнений и систем уравнений [86], однако непреодолимым ограничением в этом случае является размерность систем.

Третья глава посвящена решению нелинейных уравнений и систем нелинейных уравнений. В §3.1 предлагается процедура выпуклой комбинации (ПВК) для решения одного уравнения со многими неизвестными, задаваемого d.c. функцией. В §3.2 доказываются теоремы сходимости предложенного алгоритма, а в §3.3 проводится численное тестирование.

В §3.4–§3.6 рассматривается решение систем нелинейных уравнений. В §3.4 приведена

постановка задачи. Как и линейная задача дополненности в главе 1, система уравнений сводится к задаче д.с. минимизации, для которой предлагается алгоритм глобального поиска. В §3.5 уточняется выбор параметров алгоритма на случай квадратичных уравнений, а в §3.6 проводится вычислительный эксперимент.

**Методы исследования** При выполнении работы использовались аппарат математического программирования, элементы выпуклого анализа, а также новейшие достижения теории глобального экстремума.

**Основные результаты диссертационной работы** являются новыми и заключаются в следующем:

1. Для решения линейных задач дополненности со знаконеопределенными матрицами предложен и обоснован алгоритм, основанный на теории глобального поиска для задач минимизации д.с. функций (функций, представимых в виде разности двух выпуклых). Проведен вычислительный эксперимент по сравнению разработанного алгоритма с существующими подходами, показавший эффективность нового алгоритма на широком спектре тестовых примеров различной размерности.
2. Для линейных двухуровневых задач разработаны алгоритмы локального и глобального поисков оптимистических решений, позволяющие решать задачи высокой размерности (до 500 переменных на каждом уровне). Предложенные алгоритмы базируются на взаимосвязях линейных двухуровневых задач и невыпуклых задач математического программирования с нелинейными ограничениями-равенствами, а также теории глобального поиска в задачах с д.с. ограничениями.
3. Разработан и обоснован специальный метод решения д.с. уравнений со многими переменными. Доказана теорема сходимости предложенного метода, проведен численный эксперимент, показавший возможность применения данного метода для поиска допустимых по ограничению-равенству точек. Разработан алгоритм решения систем д.с. уравнений, основанный на теории глобального поиска для негладких задач д.с. минимизации, подтвердивший свою эффективность на серии тестовых примеров.

**Структура диссертационной работы.** Диссертация состоит из введения, трех глав и списка литературы из 143 наименований. В каждой главе используется своя нумерация параграфов, предложений, лемм, теорем, замечаний и формул.

**Апробация работы.** Основные результаты диссертации опубликованы в работах [20, 42, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 73, 74, 75, 76], из них три статьи [20, 42, 54] опубликованы в журналах, рекомендованных ВАК РФ. Также результаты работы докладывались на ежегодных школах-семинарах молодых ученых "Математическое моделирование и информационные технологии" (Иркутск–Ангасолка, 2005–2007, 2010); XIII, XIV Байкальских международных школах-семинарах "Методы оптимизации и их приложения" (Иркутск–Северобайкальск, 2005, 2008); Всероссийской конференции "Математика, информатика, управление" (Иркутск, 2005); ежегодных конференциях "Ляпуновские чтения и презентация информационных технологий" в ИДСТУ СО РАН (Иркутск, 2006, 2008, 2009); II Всероссийской конференции с международным участием "Инфокоммуникационные и вычислительные технологии и системы" (Улан-Удэ–Байкал, 2006); III, IV Всероссийских конференциях "Проблемы оптимизации и экономические приложения" (Омск, 2006, 2009); II Международной конференции по оптимизации и оптимальному управлению (Монголия, Улан-Батор, 2007); Российской конференции "Дискретная оптимизация и исследование операций" (Владивосток, 2007).

Результаты работы обсуждались на семинарах отделения Системного анализа и оптимизации ИДСТУ СО РАН, объединенном семинаре ИДСТУ СО РАН, семинаре факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова.

# Глава 1

## Оптимизационный подход к решению линейной задачи дополнительности

В данной главе рассматривается известная линейная задача дополнительности (ЛЗД) со закононеопределенными матрицами. Производится редукция поставленной задачи к невыпуклой задаче математического программирования. Для решения последней используется стратегия глобального поиска, основанная на условиях глобальной оптимальности [69, 70], одним из ключевых моментов которой является специальный метод локального поиска [69]. Для задачи линейной дополнительности конкретизированы и реализованы основные этапы стратегии глобального поиска (построение аппроксимации поверхности уровня, выбор параметров счета, и т.д.). На этой основе разработан алгоритм глобального поиска решения ЛЗД. Проведено тестирование разработанного алгоритма на достаточно широком спектре случайно сгенерированных задач и дан подробный анализ вычислительного эксперимента.

# 1.1 Постановка задачи. Различные формулировки линейной задачи дополнителности

Рассматривается линейная задача дополнителности [88], состоящая в нахождении пары векторов  $(x, w)$ , удовлетворяющих следующим условиям:

$$(\mathcal{LCP}) \quad \left. \begin{aligned} Mx + q &= w, \\ \langle x, w \rangle &= 0, \\ x \geq 0, \quad w &\geq 0, \end{aligned} \right\} \quad (1.1.1)$$

где  $x, w \in \mathbb{R}^n$ . Вектор  $q \in \mathbb{R}^n$  и вещественная знаконеопределенная  $(n \times n)$ -матрица  $M$  считаются заданными.

Несложно видеть, что задачу (LCP)–(1.1.1) можно записать в следующем равносильном виде:

$$(\mathcal{LCP}') \quad \left. \begin{aligned} \langle x, Mx + q \rangle &= 0, \\ x \geq 0, \quad Mx + q &\geq 0. \end{aligned} \right\} \quad (1.1.2)$$

Исторически ЛЗД можно рассматривать в качестве объединяющей формулировки для задач линейного, квадратичного программирования и биматричных игр [100]. Различным способам формулировки ЛЗД соответствуют разные подходы к ее решению, упомянутые во введении. Рассмотрим, как связана задача (LCP)–(1.1.1) с некоторыми задачами математического программирования.

## 1. Задачи линейного программирования.

Известно [10], что ограничения пары двойственных задач линейного программирования (ЛП)

$$\left. \begin{aligned} \langle c, x \rangle \downarrow \min, \quad \langle b, y \rangle \uparrow \max, \\ Ax \geq b, \quad A^T y \leq c, \\ x \geq 0, \quad y \geq 0, \end{aligned} \right\} \quad (1.1.3)$$

где  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ ,  $A$  –  $(m \times n)$ -матрица, можно переписать в виде

$$\begin{aligned} Ax - v &= b, \quad x \geq 0, \quad v \geq 0, \\ A^T y + u &= c, \quad y \geq 0, \quad u \geq 0. \end{aligned} \quad (1.1.4)$$

При этом, согласно теории двойственности в ЛП [10], решение задач (1.1.3) сводится к поиску  $x, y$ , удовлетворяющих (1.1.4), таких, что

$$\langle b, y \rangle = \langle c, x \rangle. \quad (1.1.5)$$

Далее, введем новые обозначения:

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, w = \begin{pmatrix} u \\ v \end{pmatrix}, q = \begin{pmatrix} c \\ -b \end{pmatrix}, M = \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}. \quad (1.1.6)$$

Тогда, переписав (1.1.4)–(1.1.5) в новых обозначениях, получаем в точности задачу  $(\mathcal{LCP})$ –(1.1.1).

## 2. Квадратичные задачи.

Нетрудно видеть, что  $(\mathcal{LCP})$ –(1.1.1) выражает необходимые условия оптимальности для квадратичной задачи

$$f(x) = \frac{1}{2}\langle Mx, x \rangle + \langle q, x \rangle \downarrow \min, \quad x \geq 0. \quad (1.1.7)$$

Действительно, если  $z$  — локальный минимум задачи (1.1.7), то при выполнении условий регулярности найдется такой ненулевой вектор  $\lambda$ , что для пары  $(z, \lambda)$  будут выполнены условия Каруша-Куна-Таккера:

$$\begin{aligned} \nabla f(z) - \lambda &= 0, \\ \lambda &\geq 0, \quad z \geq 0, \\ \langle \lambda, z \rangle &= 0, \end{aligned}$$

что может быть в точности записано как ЛЗД  $(\mathcal{LCP})$ –(1.1.1).

Однако решать задачу (1.1.7) вместо  $(\mathcal{LCP})$ –(1.1.1) не представляется возможным, поскольку в случае знаконеопределенности матрицы  $M$  функция  $f(x)$  из (1.1.7) может оказаться неограниченной снизу.

В этом случае предпочтительнее рассматривать следующую задачу:

$$\left. \begin{aligned} F(x) &= \langle x, Mx + q \rangle \downarrow \min_x, \\ x &\geq 0, \quad Mx + q \geq 0. \end{aligned} \right\} \quad (1.1.8)$$

Нетрудно видеть, что в задаче (1.1.8) целевая функция всегда ограничена снизу нулевым значением. Если некоторый вектор является решением задачи  $(\mathcal{LCP})$ –(1.1.1), то он обращает в нуль функцию  $F(\cdot)$ , а следовательно, является решением задачи (1.1.8). Обратно, если  $z$  — глобальный минимум задачи (1.1.8), то в случае, когда  $F(z) = 0$ , вектор  $z$  является также и решением  $(\mathcal{LCP})$ –(1.1.1). Если же  $F(z) > 0$ , то соответствующая ЛЗД не имеет решения. Таким образом, задача  $(\mathcal{LCP})$ –(1.1.1) может быть сведена к задаче (1.1.8).

Рассмотрим теперь произвольную квадратичную задачу:

$$\left\{ \begin{aligned} f(x) &= \frac{1}{2}\langle Qx, x \rangle + \langle c, x \rangle \downarrow \min, \\ Ax &\geq b, \quad x \geq 0, \end{aligned} \right. \quad (1.1.9)$$

где  $Q$  — симметричная  $(n \times n)$ -матрица,  $A \in \mathbb{R}^{m \times n}$ , и покажем ее связь с ЛЗД.

По теореме Каруша-Куна-Таккера [9] найдутся такие ненулевые векторы  $y \in \mathbb{R}^m$  и  $u \in \mathbb{R}^n$ , что выполнены следующие условия:

$$\begin{aligned} \nabla f(x) - u - A^T y &= 0, \\ x &\geq 0, \quad Ax - b \geq 0, \\ y &\geq 0, \quad u \geq 0, \\ \langle u, x \rangle &= 0, \\ \langle y, Ax - b \rangle &= 0. \end{aligned} \tag{1.1.10}$$

Выразим, учитывая вид функции  $f(x)$ , из первого равенства вектор  $u$ :

$$u = c + Qx - A^T y. \tag{1.1.11}$$

Несложно видеть, что, в силу неотрицательности векторов, входящих в скалярные произведения и равенства (1.1.11), систему (1.1.10) можно переписать в следующем виде:

$$\begin{aligned} \langle c + Qx - A^T y, x \rangle + \langle Ax - b, y \rangle &= 0, \\ c + Qx - A^T y &\geq 0, \quad x \geq 0, \\ Ax - b &\geq 0, \quad y \geq 0, \end{aligned}$$

т.е. получаем линейную задачу дополнителности, в которой

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad w = \begin{pmatrix} c + Qx - A^T y \\ Ax - b \end{pmatrix}, \quad q = \begin{pmatrix} c \\ -b \end{pmatrix}, \quad M = \begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix}.$$

В случае же, когда условия положительной определенности матрицы  $Q$  не выполнены, на решении данной ЛЗД выполняются лишь необходимые условия оптимальности в задаче (1.1.9).

### 3. Биматричные игры.

Рассмотрим биматричную игру  $\Gamma(A, B)$ . Без ограничения общности можем считать, что все элементы матриц  $A$  и  $B$  положительны:  $a_{ij} > 0, b_{ij} > 0, i = 1, \dots, m, j = 1, \dots, n$ .

Пусть  $(x^*, y^*) \in S_m \times S_n$  — ситуация равновесия по Нэшу в биматричной игре  $\Gamma(A, B)$ . Тогда выполнены условия оптимальности [78]:

$$\left. \begin{aligned} (Ay^*)_i &\leq \langle x^*, Ay^* \rangle \triangleq \alpha_* \quad \forall i \in I, \\ (x^*B)_j &\leq \langle x^*, By^* \rangle \triangleq \beta_* \quad \forall j \in J; \end{aligned} \right\} \tag{1.1.12}$$

и соотношения дополняющей нежесткости:

$$x_i^* > 0 \Rightarrow (Ay^*)_i = \alpha_*; \quad y_j^* > 0 \Rightarrow (x^*B)_j = \beta_*. \tag{1.1.13}$$



Для того, чтобы найти ситуацию равновесия по Нэшу в биматричной игре, необходимо и достаточно отыскать пары  $(x^*, \alpha_*)$  и  $(y^*, \beta_*)$ , удовлетворяющие условиям (1.1.12) и (1.1.13).

Преобразуем систему (1.1.12). Введем вспомогательные переменные  $u \in \mathbb{R}^m$ ,  $u \geq 0$  и  $v \in \mathbb{R}^n$ ,  $v \geq 0$ . Тогда можно записать:

$$\left. \begin{aligned} (Ay^*) + u &= e_m \alpha_*, & \langle y^*, e_n \rangle &= 1, & y^* &\geq 0, & u &\geq 0, \\ (x^*B) + v &= e_n \beta_*, & \langle x^*, e_m \rangle &= 1, & x^* &\geq 0, & v &\geq 0. \end{aligned} \right\} \quad (1.1.14)$$

С учетом соотношений дополняющей нежесткости 1.1.13 на вспомогательные переменные  $u$  и  $v$  нужно наложить дополнительные условия

$$\langle x^*, u \rangle = 0, \quad \langle y^*, v \rangle = 0. \quad (1.1.15)$$

Далее, поскольку все элементы матриц  $A$  и  $B$  положительны, а компоненты равновесных стратегий  $x^*$  и  $y^*$  неотрицательны, из (1.1.12) получаем, что  $\alpha_* > 0$  и  $\beta_* > 0$ .

Разделим в (1.1.14) соотношения в первой строчке на  $\alpha_*$ , а во второй — на  $\beta_*$  и введем новые переменные:

$$y' = \frac{y^*}{\alpha_*}, \quad x' = \frac{x^*}{\beta_*}, \quad u' = \frac{u}{\alpha_*}, \quad v' = \frac{v}{\beta_*}.$$

В новых переменных система (1.1.14), (1.1.15) запишется следующим образом:

$$\left. \begin{aligned} (Ay') + u' &= e_m, & x' &\geq 0, & u' &\geq 0, \\ (x'B) + v' &= e_n, & y' &\geq 0, & v' &\geq 0. \end{aligned} \right\} \quad (1.1.16)$$

$$\langle y', e_n \rangle = 1/\alpha_*, \quad \langle x', e_m \rangle = 1/\beta_*, \quad (1.1.17)$$

$$\langle x', u' \rangle = 0, \quad \langle y', v' \rangle = 0. \quad (1.1.18)$$

В результате получаем классическую форму линейной задачи дополнителности ( $\mathcal{LCP}$ )–(1.1.1), где

$$z = \begin{pmatrix} x' \\ y' \end{pmatrix}, \quad w = \begin{pmatrix} -u' \\ -v' \end{pmatrix}, \quad q = \begin{pmatrix} -e_m \\ -e_n \end{pmatrix}, \quad M = \begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix}.$$

с дополнительными условиями (1.1.17).

#### 4. Вариационные неравенства.

Рассмотрим вариационное неравенство, заключающееся в нахождении такого вектора  $x^* \in K$ , что

$$\langle f(x^*), y - x^* \rangle \geq 0 \quad \forall y \in K,$$

где  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , а  $K$  — полиэдральное множество:

$$K = \{y \in \mathbb{R}^n : Ay \geq b, y \geq 0\}.$$

Вектор  $x^*$  является решением вариационного неравенства тогда и только тогда, когда доставляет решение задаче линейного программирования

$$\left. \begin{array}{l} \langle f(x^*), y \rangle \downarrow \min_y \\ Ay \geq b, y \geq 0. \end{array} \right\} \quad (1.1.19)$$

Тогда из теории двойственности в линейном программировании следует, что найдется вектор  $u^*$ , являющийся решением двойственной задачи

$$\left. \begin{array}{l} \langle b, u \rangle \downarrow \max_u \\ A^T u \leq f(x^*), u \geq 0. \end{array} \right\} \quad (1.1.20)$$

При этом пара  $(x^*, y^*)$  удовлетворяют следующим условиям:

$$\begin{aligned} f(x^*) - A^T u^* \geq 0, \quad x^* \geq 0, \quad \langle f(x^*) - A^T u^*, x^* \rangle = 0, \\ -b + Ax^* \geq 0, \quad u^* \geq 0, \quad \langle -b + Ax^*, u^* \rangle = 0. \end{aligned} \quad (1.1.21)$$

В случае, когда  $f(x) = Cx + d$  — аффинное отображение, (1.1.21) представляет собой линейную задачу дополнителности:

$$z = \begin{pmatrix} x \\ u \end{pmatrix}, \quad w = \begin{pmatrix} Cx - A^T u + q \\ Ax - b \end{pmatrix}, \quad q = \begin{pmatrix} d \\ -b \end{pmatrix}, \quad M = \begin{pmatrix} C & -A^T \\ A & 0 \end{pmatrix}$$

## 5. Кусочно-линейные функции

Для решения линейной задачи дополнителности часто используется ее формулировка в виде минимизации кусочно-линейной функции [100]:

$$\begin{aligned} \sum_{i=1}^n \min\{x_i, (Mx + q)_i\} \downarrow \min_x, \\ x \geq 0, \quad Mx + q \geq 0. \end{aligned} \quad (1.1.22)$$

Задача (1.1.22) представляет собой задачу негладкой вогнутой минимизации.

Аналогично можно записать ЛЗД в виде системы уравнений:

$$\min(x, Mx + q) = 0,$$

где операция  $\min(a, b)$  означает взятие минимума двух векторов покомпонентно.

## 1.2 Редукция к задаче d.c. минимизации

Как упоминалось в §1.1, задача (1.1.1) может быть сформулирована как задача математического программирования [131]:

$$\left. \begin{aligned} F(x) = \langle x, Mx + q \rangle \downarrow \min_x, \\ x \geq 0, \quad Mx + q \geq 0. \end{aligned} \right\} \quad (1.2.1)$$

Для случая выпуклой задачи (1.2.1) (неотрицательной определенности матрицы  $M$ ) разработано большое количество методов выпуклой оптимизации [3, 9, 81, 126]. Ситуация меняется принципиальным образом, если  $M$  — знаконеопределенная, поскольку в этом случае прямое применение известных методов выпуклой оптимизации к задаче (1.2.1), как известно [9, 31, 34, 36], вообще говоря, не приводит, глобальному решению, а чаще всего лишь к стационару.

С развитием методов выпуклой оптимизации [57, 59, 60, 126], вычислительных технологий и созданием многочисленных решателей линейных и выпуклых квадратичных задач (например, известные пакеты прикладных программ CPLEX [141], Xpress-MP [140] и т.д.) решение выпуклых задач даже повышенной размерности не представляет, как правило, больших проблем. Вопрос в этом случае состоит не столько в нахождении самого решения, сколько в скорости работы того или иного алгоритма. Иначе дело обстоит с невыпуклыми задачами, поскольку, как будет показано ниже, готовые пакеты прикладных программ, основанные на методах выпуклой оптимизации, как правило, с ними не справляются.

В работе рассматривается задача (1.2.1) именно со знаконеопределенной матрицей  $M$ , т.е. задача с невыпуклой целевой функцией. Как было показано во введении, для невыпуклой задачи (1.2.1) может быть применен алгоритм глобального поиска (АГП) [69, 70], основанный на условиях глобальной оптимальности для задачи с целевой функцией, представимой в виде разности двух выпуклых функций, т.е. d.c. функции.

Для применения данной методики прежде всего необходимо получить явное d.c. разложение целевой функции.

Как известно, знаконеопределенная матрица  $M$  может быть представлена в виде разности двух положительно определенных матриц:

$$M = M_1 - M_2, \quad M_i > 0, \quad i = 1, 2.$$

Существуют различные схемы получения такого разложения. Ниже представлен один довольно простой прием [69] конструирования матриц  $M_1$ ,  $M_2$ .

Известно, что матрица с неотрицательными элементами и строго доминирующей диагональю является положительно определенной [82]. Именно в таком виде и будем искать матрицы  $M_1$  и  $M_2$  на основе заданной матрицы  $M = \{m_{ij}\}$ .

Прежде всего представим матрицу в виде разности матриц с неотрицательными компонентами:  $M = D_1 - D_2$ ,  $D_1 = \{d_{ij}^{(1)}\}$ ,  $D_2 = \{d_{ij}^{(2)}\}$ , где

$$d_{ij}^{(1)} = \begin{cases} m_{ij}, & \text{если } m_{ij} \geq 0, \\ 0, & \text{если } m_{ij} < 0, \end{cases} \quad d_{ij}^{(2)} = \begin{cases} 0, & \text{если } m_{ij} \geq 0, \\ -m_{ij}, & \text{если } m_{ij} < 0. \end{cases}$$

Далее, введем в рассмотрение матрицы  $F_1 \triangleq D_1 + \Lambda_1$ ,  $F_2 \triangleq D_2 + \Lambda_1$ , где элементы диагональной матрицы  $\Lambda_1 = \{\lambda_{ij}^{(1)}\}$  вычисляются по формулам

$$\lambda_{ii}^{(1)} = \begin{cases} 0, & \text{если } d_{ii}^{(1)} > S_i, \\ S_i - d_{ii}^{(1)} + \chi, & \text{если } d_{ii}^{(1)} \leq S_i. \end{cases}$$

Здесь  $S_i \triangleq \sum_{i \neq j} d_{ij}^{(1)}$  — сумма внедиагональных элементов  $i$ -ой строки матрицы  $D_1$ ,  $\chi > 0$  — некоторое фиксированное число. Заметим, что элементы обеих матриц  $F_1$  и  $F_2$  остаются при этом неотрицательными, а матрица  $F_1$  оказывается положительно определенной.

Аналогичным образом построим положительно определенные матрицы  $M_1$  и  $M_2$ :

$M_1 = F_1 + \Lambda_2$ ,  $M_2 = F_2 + \Lambda_2$ , где диагональная матрица  $\Lambda_2 = \{\lambda_{ij}^{(2)}\}$  определена элементами

$$\lambda_{ii}^{(2)} = \begin{cases} 0, & \text{если } f_{ii}^{(2)} > T_i, \\ T_i - f_{ii}^{(2)} + \chi, & \text{если } f_{ii}^{(2)} \leq T_i, \end{cases}$$

где  $T_i = \sum_{i \neq j} f_{ij}^{(2)}$  — сумма внедиагональных элементов  $i$ -ой строки матрицы  $F_2$ . Нетрудно проверить, что  $M = M_1 - M_2$ .

Таким образом, оказывается, что матрица  $M$  представлена в виде разности матриц  $M_1$  и  $M_2$  с неотрицательными компонентами и доминирующими диагоналями.

С использованием вышеприведенного метода декомпозиции получаем следующее д.с. представление целевой функции в задаче (1.2.1):

$$F(x) = G(x) - H(x),$$

где

$$G(x) = \langle M_1 x, x \rangle + \langle q, x \rangle, \quad H(x) = \langle M_2 x, x \rangle \quad (1.2.2)$$

являются сильно выпуклыми функциями. Следовательно, возникает следующая задача д.с. минимизации:

$$F(x) = \langle M_1 x, x \rangle + \langle q, x \rangle - \langle M_2 x, x \rangle \downarrow \min, \quad x \in S, \quad (1.2.3)$$

где допустимое множество  $S$ , определено следующим образом:

$$S = \{x \in \mathbb{R}^n \mid x \geq 0, Mx + q \geq 0\}. \quad (1.2.4)$$

Далее, для решения задачи (1.2.3) будем применять алгоритм глобального поиска [69, 70], основными этапами которого являются локальный поиск и процедура выхода из критической точки, полученной на этапе локального поиска.

**Замечание 1.2.1** Для сведения ЛЗД к задаче минимизации вместо формулировки (1.2.1) можно использовать (1.1.22). В этом случае необходимо искать минимум негладкой вогнутой функции. Отметим, что эффективность алгоритма глобального поиска напрямую зависит от эффективности решения вспомогательных задач. Поэтому в данной работе был выбран первый подход, поскольку он дает возможность применения для решения выпуклых гладких подзадач современных решателей (см. [140], [141]).

### 1.3 Специальный метод локального поиска

Локальный поиск является одним из основных модулей алгоритмов решения невыпуклых задач, основанных на теории глобального поиска [18, 44, 69, 78, 71]. Целью локального поиска обычно является отыскание критических (стационарных) точек в рассматриваемой невыпуклой задаче. При этом понятие критической точки зависит от структуры конкретной задачи и соответствующего локального метода.

Поскольку, как было показано в §1.2, задача (1.2.3) является задачей д.с. минимизации, то для ее решения будем применять специальный метод локального поиска (СМЛП), заключающийся в решении последовательности выпуклых частично линеаризованных задач. Приведем описание этого метода.

Пусть задано некоторое начальное приближение  $x^0 \in S$ . Далее, если известна допустимая точка  $x^s \in S$ , то следующее приближение  $x^{s+1}$  будем искать как приближенное решение линеаризованной в точке  $x^s$  задачи

$$(\mathcal{P}_{\mathcal{L}_s}) \quad \left. \begin{aligned} J_s(x) = \langle M_1 x, x \rangle + \langle q - 2M_2 x^s, x \rangle \downarrow \min, \\ x \geq 0, \quad Mx + q \geq 0. \end{aligned} \right\} \quad (1.3.1)$$

Более точно это означает, что точка  $x^{s+1}$  удовлетворяет неравенству

$$\langle M_1 x^{s+1}, x^{s+1} \rangle + \langle q - 2M_2 x^s, x^{s+1} \rangle \leq \inf_x \{ \langle M_1 x, x \rangle + \langle q - 2M_2 x^s, x \rangle \} + \delta_s, \quad (1.3.2)$$

где последовательность  $\delta_s$  такова, что

$$\delta_s \geq 0, \quad s = 1, 2, \dots, \quad \sum_{s=0}^{\infty} \delta_s < \infty. \quad (1.3.3)$$

Отметим, что задача (1.3.1) оказывается уже выпуклой и поэтому может быть решена посредством современных пакетов и библиотек программ методов выпуклой оптимизации. Для численного решения этой задачи в работе применялся известный решатель Xpress-MP [140], предназначенный, в частности, для решения выпуклых квадратичных задач и задач линейного программирования.

В силу ограниченности снизу целевой функции  $F$  на допустимом множестве  $S$  и непрерывной дифференцируемости функции  $H(x) = \langle M_2 x, x \rangle$ , справедлива следующая теорема сходимости данного алгоритма.

**Теорема 1.3.1** [69] *i) Последовательность  $\{x^s\}$ , генерируемая по правилу (1.3.2) удовлетворяет условию*

$$\lim_{s \rightarrow \infty} \left\{ \inf_{x \in S} [G(x) - G(x^{s+1}) - \langle \nabla H(x^s), x - x^{s+1} \rangle] \right\} = 0. \quad (1.3.4)$$

*ii) При этом последовательность  $\{x^s\}$  сходится, кроме того ее предел  $\hat{x}$  является решением следующей линеаризованной задачи:*

$$G(x) - \langle \nabla H(\hat{x}), x \rangle \downarrow \min, \quad x \in S. \quad (1.3.5)$$

*iii) Кроме того, из (1.3.5) следует классическое условие стационарности*

$$\langle \nabla G(\hat{x}) - \nabla H(\hat{x}), x - \hat{x} \rangle \geq 0, \quad \forall x \in S. \quad (1.3.6)$$

**Доказательство.** Из (1.3.2) в силу сильной выпуклости  $H(\cdot)$  вытекает

$$\begin{aligned} -\delta_s &\leq \inf_{x \in S} \{G(x) - \langle \nabla H(x^s), x \rangle\} - G(x^{s+1}) + \langle \nabla f(x^s), x^{s+1} \rangle \leq \\ &\leq G(x^s) - G(x^{s+1}) + \langle \nabla H(x^s), x^{s+1} - x^s \rangle \leq G(x^s) - G(x^{s+1}) + H(x^{s+1}) - H(x^s) - \\ &\quad - \mu \|x^{s+1} - x^s\|^2 = F(x^s) - F(x^{s+1}) - \mu \|x^{s+1} - x^s\|^2, \end{aligned} \quad (1.3.7)$$

где  $\mu$  — константа сильной выпуклости функции  $H(\cdot)$ . Таким образом, получена числовая последовательность  $\{F(x^s)\}$ , для которой  $F(x^{s+1}) \leq F(x^s) + \delta_s$ , т.е. почти монотонно убывающая последовательность. Поэтому в силу ограниченности снизу функции  $F(\cdot)$  на  $D$ , а также с учетом (1.3.3), существует конечный предел  $\lim_{s \rightarrow \infty} F(x^s) = F_*$ . Поэтому из цепочки (1.3.7) следует справедливость условия (1.3.4).

Далее, из (1.3.7) получаем

$$\|x^{s+1} - x^s\|^2 \leq \frac{1}{\mu}(F(x^s) - F(x^{s+1}) + \delta^s). \quad (1.3.8)$$

Переходя к пределу при  $s \rightarrow \infty$  получаем существование предела  $\lim_{s \rightarrow \infty} x^s = \hat{x}$ . Тогда в силу непрерывности  $G(\cdot)$  и  $\nabla H(\cdot)$  из (1.3.4) получаем

$$\inf_{x \in S} \{G(x) - G(\hat{x}) - \langle \nabla H(\hat{x}), x - \hat{x} \rangle\} = 0,$$

откуда непосредственно следует (1.3.6). #

Введем определение.

**Определение 1.3.1** *i) Точно критической в задаче (1.2.3) назовем точку  $z$ , являющуюся решением линеаризованной задачи:*

$$\langle M_1 x, x \rangle + \langle q - 2M_2 z, x \rangle \downarrow \min, \quad x \in S, \quad (\mathcal{P}\mathcal{L}_z)$$

*т.е. линеаризация происходит в самой этой точке  $z$ .*

*ii) Приблизленно критической называется точка  $z$ , являющаяся  $\tau$ -решением задачи  $(\mathcal{P}\mathcal{L}_z)$ :*

$$\langle M_1 z, z \rangle - \langle q - 2M_2 z, z \rangle - \tau \leq \inf_x \{M_1 x, x - \langle q - 2M_2 z, x \rangle \mid x \in S\}. \quad \#$$

В качестве критерия останова для метода локального поиска можно использовать одно из следующих неравенств [69]:

$$F(x^s) - F(x^{s+1}) \leq \frac{\tau}{2}, \quad (1.3.9)$$

$$G(x^s) - G(x^{s+1}) + \langle \nabla H(x^s), x^{s+1} - x^s \rangle \leq \frac{\tau}{2}, \quad (1.3.10)$$

где  $\tau > 0$  — заданная точность.

Нетрудно показать, что если выполнен один из критериев останова (1.3.9) или (1.3.10), то в этом случае точка, в которой произошел останов, является  $\tau$ -критической в задаче (1.2.3). Действительно, из (1.3.7) следует

$$G(x^s) - \langle \nabla H(x^s), x^s \rangle \leq \frac{\tau}{2} + G(x^{s+1}) - \langle \nabla H(x^s), x^{s+1} \rangle \leq \inf_{x \in D} \{G(x) - \langle \nabla H(x^s), x \rangle\} + \frac{\tau}{2} + \delta_s.$$

Следовательно, если  $\delta_s \leq \frac{\tau}{2}$ , то точка  $x^s$  является  $\tau$ -решением задачи  $(\mathcal{P}\mathcal{L}_s)$ .

Похожая технология нахождения критических точек в задачах д.с. минимизации предложена в публикациях [117, 118]. Однако разработанный в указанных работах метод предназначен для задач безусловной минимизации.

Следующий параграф посвящен тестированию специального метода локального поиска на серии задач линейной дополнительнойности.

## 1.4 Построение тестовых примеров и апробация алгоритма локального поиска

В литературе можно встретить достаточно много практических примеров линейных задач о дополнителности (см. введение), а также тестовых задач [88, 123, 131] для проверки работы алгоритмов. К сожалению, большинство из них оказываются либо выпуклыми, с неотрицательно определенной матрицей  $M$  и поэтому решаемыми посредством существующих пакетов прикладных программ (например, пакетами для решения выпуклых квадратичных задач [140] или специализированных пакетов для ЛЗД [142]), либо имеют небольшую размерность, что позволяет эффективно использовать методы перебора крайних точек допустимого множества [100].

Более интересным представляется тестирование предлагаемой здесь методики на задачах со знаконеопределенными матрицами  $M$ , тестовые примеры которых в доступной литературе обнаружить не удалось. В данной работе для генерации тестовых задач был предложен следующий способ.

Элементы матрицы  $M$  случайно выбирались из отрезка  $[-n, n]$ , где  $n$  — размерность задачи. Далее задавалось решение задачи для каждого  $i = 1, \dots, n$ , удовлетворяющее одному из условий:

$$1) x_i^* = 0, (Mx^* + q)_i = 1 \text{ либо } 2) x_i^* = 1, (Mx^* + q)_i = 0. \quad (1.4.1)$$

Затем из соотношений (1.4.1) вычислялись компоненты вектора  $q$ . Выбор варианта 1) или 2) для  $i$ -й компоненты происходил случайным образом. Для сгенерированных задач размерностей от 2 до 100 выбиралось 3 различные стартовые точки, что, позволило, с одной стороны, рассмотреть поведение метода в различных начальных условиях, а с другой — найти подходящие (не сразу приводящие к глобальному решению) начальные точки для дальнейшего тестирования глобального поиска.

Отметим, что в силу теоремы 1.3.1, линейризованные задачи в методе локального поиска можно решать на его начальных шагах с достаточно низкой точностью, постепенно увеличивая затем точность решения линейризованных задач  $\delta_s$ . Поэтому точность решения линейризованных задач выбиралась следующим образом:  $\delta_0 = 0.1$ ,  $\delta_{s+1} = 0.5\delta_s$ ,  $s = 0, 1, \dots$

Все программы, реализующие алгоритмы, представленные в данной главе, были написаны в системе Visual C++, счет производился на компьютере Pentium 4,



3.00 Ghz с 1 Gb оперативной памяти. Выпуклые вспомогательные линеаризованные задачи решались с помощью решателя Xpress-MP [140] барьерным методом Ньютона.

В табл. 1.1, представленной ниже, использовались следующие обозначения:  $n$  — размерность задачи;  $N_{x_0}$  — номер стартовой точки;  $F_0$  — начальное значение целевой функции;  $F_{loc}$  — полученное локальным поиском значение функции;  $PL$  — количество решенных линеаризованных задач;  $T_{loc}$  — время счета (в секундах).

Напомним, что точка глобального решения задачи (1.2.3) обращает целевую функцию в ноль, поэтому 0 в колонке  $F_{loc}$  означает, что найдено глобальное решение с точностью  $10^{-4}$ .

Для сравнения в каждой задаче вначале была предпринята попытка нахождения глобального минимума задачи (1.2.1) с помощью прямого применения решателя Xpress-MP к задаче (1.2.3). В колонке  $F_X$  приведены полученные в этом случае значения целевой функции или прочерки при невозможности решить задачу указанным пакетом программ. Подчеркнем, что Xpress-MP предназначен для решения только выпуклых задач линейного и квадратичного программирования. Прочерки в таблице свидетельствуют о том, что сгенерированные задачи оказались невыпуклыми.

Поскольку стартовые точки не обязательно являются допустимыми в исходной задаче, в качестве начального значения целевой функции  $F_0$  в таблице приведены значения функции  $F$  после решения первой линеаризованной задачи в локальном поиске.

Как можно видеть из таблицы, для небольших размерностей (2 и 4) специальным методом локального поиска удалось решить задачи, начиная из всех стартовых точек. При увеличении размерности, картина качественно меняется. Так для размерностей 6 и 10 со всех начальных точек алгоритм попадает в точки, близкие к решению, но тем не менее не являющиеся глобальным экстремумом, а на размерностях, равных 8, 15 и 25, глобальное решение удается найти только из одного начального приближения, что еще раз подчеркивает сложность решения невыпуклых задач.

С дальнейшим ростом числа переменных, а значит и числа локальных экстремумов, как нетрудно заметить, сложность задач существенно увеличивается. Во всех остальных случаях глобальное решение не удалось найти ни из одной стартовой точки, хотя значение целевой функции в полученной точке значительно меньше, чем в стартовой.

В отдельных случаях из точки, наиболее удаленной от решения по значению целевой функции, удается за меньшее число итераций алгоритма попасть в точку, близкую к решению (например, на размерности 40 для первого начального приближения). Кроме

$n$	$F_X$	$N_{x_0}$	$F_0$	$F_{loc}$	$PL$	$T_{loc}$	$n$	$F_X$	$N_{x_0}$	$F_0$	$F_{loc}$	$PL$	$T_{loc}$
2	2.25	1	0.211	0	12	0.14	30	-	1	168.707	0.089	91	1.17
		2	0.010	0	2	0.14			2	70.127	0.051	90	1.16
		3	68.741	0	7	0.14			3	27.807	0.048	85	1.11
4	0.04	1	0.975	0	12	0.16	40	-	1	295.140	0.0043	88	2.05
		2	2.250	0	11	0.17			2	79.203	0.241	89	2.09
		3	11.660	0	31	0.19			3	84.455	0.123	90	2.13
6	0	1	1.659	0.0016	12	0.17	50	-	1	357.251	2.805	84	2.08
		2	0.456	0.0015	8	0.16			2	103.259	1.580	85	2.11
		3	0.022	0.0016	12	0.17			3	87.91	0.446	86	2.13
8	1.71	1	15.177	0.069	33	0.22	60	-	1	400.340	0.590	83	2.86
		2	4.571	0	16	0.17			2	148.380	1.007	85	2.94
		3	3.078	0.056	30	0.22			3	90.558	0.23	86	3.05
10	0	1	6.708	0.056	30	0.22	70	-	1	622.85	3.033	85	4.03
		2	6.848	0.056	20	0.22			2	195.55	1.905	86	4.11
		3	4.597	0.056	29	0.23			3	10.245	0.073	87	4.22
15	-	1	57.0044	0.28	91	0.47	80	-	1	649.048	14.53	83	5.00
		2	32.643	0	15	0.20			2	272.761	2.669	86	5.23
		3	15.871	1.063	92	0.48			3	129.916	0.873	84	5.19
20	-	1	74.613	0.074	88	0.40	90	-	1	847.95	11.827	81	6.55
		2	42.107	0.057	36	0.36			2	282.134	8.050	83	6.83
		3	42.136	0.0051	38	0.36			3	140.428	3.360	81	6.73
25	-	1	69.151	0.012	93	0.91	100	-	1	1279.593	20.899	80	8.01
		2	22.779	0	92	0.89			2	447.298	5.337	82	8.19
		3	2.717	0.0311	34	0.48			3	199.42	3.701	67	7.13

Таблица 1.1: Тестирование локального поиска

того, с ростом размерности полученное значение целевой функции все сильнее отличается от нуля. В то же время число решенных линеаризованных задач, начиная с размерности 30, изменяется незначительно, принимая, в среднем, значения от 80 до 90.

Следует также отметить, что с помощью метода локального поиска, предназначенного, по сути, только для нахождения критических точек, во многих случаях (например, на размерностях 2, 4, 8, 15, 25) удалось получить решение задачи. При этом СМЛП уступил решателю Xpress-MP только в задачах с шестью и десятью переменными. Во всех

остальных случаях прямое применение пакета программ Xpress-MP оказалось неэффективным, что говорит о трудности и невыпуклости сгенерированных задач. В то же время специальный метод локального поиска позволил найти критические точки со значительно меньшим, по сравнению с исходным, значением функции, для всех размерностей, что говорит о возможности его применения в алгоритме глобального поиска.

## 1.5 Алгоритм глобального поиска

В результате численного тестирования специального метода локального поиска, проведенного в предыдущем параграфе, было установлено, что локальный поиск, вообще говоря, не обеспечивает достижения глобального решения, даже в задачах небольших размерностей. Поэтому далее предлагается рассмотреть для решения ЛЗД алгоритм глобального поиска (АГП) [69], который фактически состоит из двух основных этапов: а) локального поиска и б) процедуры выхода из полученной локальным поиском критической точки.

Теоретической основой АГП являются следующие условия глобальной оптимальности.

**Теорема 1.5.1** [69, 70] *Если допустимая точка  $z \in S$  является глобальным решением задачи (1.2.1), то*

$$\begin{aligned} \forall (y, \beta) : y \in S, \beta - H(y) = \zeta \triangleq F(z), \\ G(y) \leq \beta \leq \sup(G, S), \\ G(x) - \beta \geq \langle \nabla H(y), x - y \rangle \quad \forall x \in S. \end{aligned} \tag{1.5.1}$$

*Если, кроме того, выполнено условие*

$$\exists v \in S : F(v) > F(z) \triangleq \zeta, \tag{1.5.2}$$

*то условия (1.5.1) становятся и достаточными для того, чтобы допустимая точка  $z$  была глобальным решением задачи (1.2.1). #*

Используя д.с. разложение целевой функции  $F$  из §1.2, в соответствии с теорией глобального поиска [69], можно декомпозировать задачу (1.2.1) на несколько более простых задач.

Пусть известна некоторая приближенно критическая точка  $z_k$ , полученная методом локального поиска, со значением целевой функции  $F(z_k) := \zeta_k$ . Тогда производится следующая цепочка операций [69, 70].

1) Выбирается число  $\beta \in [\beta_-, \beta_+]$ , где  $\beta_- \triangleq \inf(G, S)$ ,  $\beta_+ \triangleq \sup(G, S)$ ,  $G(x) = \langle M_1 x, x \rangle + \langle q, x \rangle$  и строится некоторая аппроксимация

$$\mathcal{R}_k(\beta) = \{v^1, \dots, v^N | H(v^i) = \beta - \zeta_k, i = 1, \dots, N\}.$$

поверхности уровня сильно выпуклой функции  $H(x) \triangleq \langle x, M_2 x \rangle$ ,  $M_2 = M_2^T > 0$ ,

$$U(\zeta_k) = \{x | H(x) = \beta - \zeta_k\}.$$

Вопрос построения аппроксимации является ключевым в реализации АГП и, вообще говоря, зависит от вида целевой функции и допустимого множества задачи. Ниже будут рассмотрены несколько способов построения аппроксимации поверхности уровня в задаче (1.2.1).

2) Для всех точек  $v^i$  аппроксимации  $\mathcal{R}_k$  проверяется неравенство

$$G(v^i) \leq \beta, \quad i = 1, \dots, N, \quad (1.5.3)$$

следующее из условий глобальной оптимальности для задач d.c. программирования [69]. Если неравенство (1.5.3) выполнено, то точка аппроксимации будет использоваться дальше. Если же (1.5.3) нарушено, то согласно теории точка  $v^i$  является бесперспективной с точки зрения улучшения с ее помощью текущей критической точки.

3) Исходя из точек  $v^i$  ( $\langle M_2 v^i, v^i \rangle = \beta - \zeta_k$ ) аппроксимации (которые не обязательно выбирать допустимыми, что удобно при проведении вычислительного эксперимента), осуществляется локальный поиск, доставляющий приближенно критические точки  $u^i$ ,  $i = 1, \dots, N$ .

4) Далее, значение целевой функции в каждой точке  $u^i$  сравнивается со значением целевой функции в текущей критической точке  $z^k$ . Если в какой-либо полученной точке значение целевой функции лучше, чем в текущей, происходит обновление последней, и весь процесс повторяется.

Ниже представлена схема глобального поиска для рассматриваемой задачи в более алгоритмизованном виде.

Пусть заданы начальная точка  $x_0 \in S$ ,  $\beta_-$  и  $\beta_+$ , числовые последовательности  $\{\tau_k\}$  и  $\{\delta_k\}$ :  $\tau_k, \delta_k > 0$ ,  $k = 0, 1, 2, \dots$ ,  $\tau_k \downarrow 0$ ,  $\delta_k \downarrow 0$  ( $k \rightarrow \infty$ ) и параметры алгоритма  $\nu$  и  $m$ .

*Алгоритм глобального поиска*

**Шаг 0.** Положить  $k := 0$ ,  $x^k := x_0$ ,  $s := 1$ ,  $\beta := \beta_-$ ,  $\Delta\beta = (\beta_+ - \beta_-)/m$ .

**Шаг 1.** Начиная с  $x^k \in S$ , посредством специального метода локального поиска построить  $\tau_k$ -критическую точку  $z^k$ , положить  $\zeta_k := F(z^k)$ .

**Шаг 2.** Если  $F(z^k) < \chi$ , где  $\chi$  — заданная точность, то Stop.  $z^k$  — приближенное решение задачи.

**Шаг 3.** Построить точку  $v^i$  аппроксимации поверхности уровня функции  $H$

$$\mathcal{R}_k(\beta) = \{v^1, \dots, v^N | H(v^i) = \beta - \zeta_k, i = 1, \dots, N\}. \quad (1.5.4)$$

**Шаг 4.** Если  $G(v^i) > \beta + \nu\beta$ , то положить  $i := i + 1$  и перейти на шаг 3. Иначе перейти на шаг 5.

**Шаг 5.** Начиная с точки  $v^i$ , построить специальным методом локального поиска  $\tau_k$ -критическую точку  $u^i \in S$ .

**Шаг 6.** Если  $F(u^i) < \chi$ , то Stop.  $u^i$  — приближенное решение задачи.

**Шаг 7.** Если  $F(u^i) \geq \zeta_k$ ,  $i < N$ , то положить  $i := i + 1$  и вернуться на шаг 3.

**Шаг 8.** Если  $F(u^i) \geq \zeta_k$ ,  $i = N$ ,  $\beta < \beta_+$  то положить  $\beta := \beta + \Delta\beta$ ,  $i := 1$  и вернуться на шаг 3.

**Шаг 9.** Если  $F(u^i) < \zeta_k$  то положить  $z^{k+1} := u^i$ ,  $k := k + 1$ ,  $i := 1$ ,  $\beta := \beta_-$  и перейти на шаг 2.

**Шаг 10.** Если  $i = N$ ,  $F(u^s) \geq \zeta_k \forall \beta \in [\beta_-; \beta_+]$  и  $\tau_k \leq \chi$ , то Stop.  $z^k$  — полученное решение задачи.

**Замечание 1.5.1** 1. Поскольку в рассматриваемой задаче в точке глобального решения значение целевой функции равно 0, в АГП вводятся дополнительные критерии останова (на шагах 2, 6), которые отсутствуют в АГП для произвольной д.с. функции. Это позволяет, в случае получения решения, прекратить перебор по  $\beta$  и точкам аппроксимации, а значит, сократить время работы алгоритма.

2. Параметр  $t$  отвечает за разбиение отрезка одномерного поиска по  $\beta$  на различное количество частей. С помощью параметра  $\nu$  можно изменять точность выполнения неравенства (1.5.3), вытекающего из условий глобальной оптимальности для задач д.с. программирования, (с целью уменьшить влияние машинных ошибок округления) [44, 78]. Управляя этими двумя параметрами, можно изменять время работы алгоритма. При увеличении  $t$  и  $\nu$  возрастает точность работы алгоритма, но за это приходится платить пропорциональным увеличением времени решения.

3. Заметим, что в качестве  $\beta_-$  можно выбрать 0, так как  $G(x) = \langle M_2x, x \rangle \geq 0 \forall x \in S$ . Поскольку выпуклая квадратичная функция  $G(x)$  оказывается неограниченной

сверху на множестве  $S$ , то верхняя граница  $\beta_+$  вводится искусственно.

## 1.6 Вычислительный эксперимент

Вычислительный эксперимент по тестированию алгоритма глобального поиска в задаче (1.2.1) проходил в несколько этапов.

Целью **первого этапа** был выбор наиболее подходящей из предложенных аппроксимаций поверхности уровня. На **втором этапе** происходил выбор точности осуществления локального поиска. Учитывая тот факт, что на начальных итерациях АГП критические точки можно находить с меньшей точностью, была предпринята попытка уменьшить время работы алгоритма. **Третий этап** эксперимента посвящен решению серий задач высокой размерности, а также сравнению реализации АГП с некоторыми другими алгоритмами и программными средствами.

### 1.6.1 Выбор аппроксимации поверхности уровня

При численном решении, с учетом опыта ранее решенных задач [18, 19, 44, 69, 80], использовались три аппроксимации.

Первый набор точек аппроксимации наиболее часто используется при решении невыпуклых задач в силу простоты вычисления и имеет следующий вид [18, 19, 69]:

$$\mathcal{R}_1 = \{y^i = \mu_i e^i, y^{i+n} = -\mu_i e^i \mid i = 1, \dots, n\},$$

где  $e^i = (0, \dots, 1, \dots, 0)$  — вектор из стандартного базиса,  $\mu_i$  — число, которое находится из уравнения

$$H(\mu_i e^i) = \beta - \zeta. \quad (1.6.1)$$

Поскольку  $H$  в рассматриваемой задаче является квадратичной и сильно выпуклой, то уравнение (1.6.1) разрешается аналитически:

$$\mu_i = \pm \sqrt{\frac{\beta - \zeta}{\{M_2\}_{ii}}}.$$

Достоинством такого набора является простота построения векторов  $y^i$ .

Одним из способов изменения точек аппроксимации в зависимости от итерации является добавление текущей критической точки  $z^k$  в определение  $y^i$  [18, 44, 80]:

$$\mathcal{R}_2 = \{y^i = z^k + \mu_i e^i, y^{i+n} = z^k - \mu_i e^i, i = 1, \dots, n\},$$

где  $\mu_i$  также находится аналитически из квадратного уравнения  $H(z^k \pm \mu_i e^i) = \beta - \zeta$ .

Далее, учитывая структуру задачи в исходной постановке, а именно, тот факт, что глобальное решение обязательно достигается в вершине допустимого множества  $S$ , точками третьей аппроксимации  $\mathcal{R}_3$  были выбраны векторы

$$y^i = \mu_i d^i, \quad i = 1, \dots, n+1,$$

где в качестве  $d^i$  для  $i = 1, \dots, n$  использовались решения следующих задач линейного программирования:

$$(\mathcal{LP}_i) \quad \left. \begin{array}{l} \langle e^i, x \rangle \downarrow \min, \\ x \geq 0, \quad Mx + q \geq 0, \quad i = 1, \dots, n, \end{array} \right\} \quad (1.6.2)$$

а  $d^{n+1}$  находилось как решение аналогичной задачи

$$(\mathcal{LP}_{n+1}) \quad \left. \begin{array}{l} \langle e_n, x \rangle \downarrow \min, \\ x \geq 0, \quad Mx + q \geq 0, \end{array} \right\} \quad (1.6.3)$$

где  $e_n$  — вектор, состоящий из единиц,  $e_n = (1, 1, \dots, 1, 1)$ .

Поскольку решения этих задач также достигаются в вершинах множества  $S$ , то предполагается, что такая аппроксимация полнее учитывает структуру допустимого множества. Очевидно, что для построения третьей аппроксимации требуется больше времени вследствие необходимости решения серии задач линейного программирования, однако заметим, что в течение работы алгоритма глобального поиска вычисления достаточно провести один раз, в начале его работы.

Для проведения первого этапа вычислительного эксперимента были выбраны следующие параметры алгоритма. Точность работы локального поиска  $\tau_k$  выбиралась равной  $5 \cdot 10^{-5}$  на каждой итерации, точность глобального поиска (по значению целевой функции)  $\chi = 10^{-4}$  (в соответствии с теоремой сходимости АГП [69], [70], условия на согласование точностей имеют вид  $\chi \geq 2\tau_k$ ). Параметр  $\nu$ , участвующий в проверке условия на шаге 4 брался равным  $10^{-4}$ . Для вычисления шага по  $\beta$  отрезок  $[\beta_-, \beta_+]$  делился на 10 частей ( $m = 10$ ), а в случае, если решение не было найдено, происходило разбиение отрезка на 50 частей. В качестве стартовой выбиралась та из трех точек, участвовавших в тестировании специального метода локального поиска, начиная с которой было получено наилучшее значение целевой функции.

Ниже в табл. 1.2 представлены результаты сравнения работы АГП с применением трех описанных аппроксимаций.

Здесь  $St$  — количество итераций алгоритма, или, что то же самое, число пройденных критических точек;  $F_{glob}$  — полученное значение целевой функции. Время  $T$  приводится в минутах, секундах и долях секунд.

$n$	$\mathcal{R}_1$				$\mathcal{R}_2$				$\mathcal{R}_3$			
	$PL$	$St$	$F_{glob}$	$T$	$PL$	$St$	$F_{glob}$	$T$	$PL$	$St$	$F_{glob}$	$T$
6	24	2	0	0:00.16	24	2	0	0:00.17	24	2	0	0:00.17
8	488	2	0	0:01.20	61	2	0	0:00.27	358	2	0	0:00.91
10	42	2	0	0:00.25	42	2	0	0:00.25	42	2	0	0:00.25
15	128	3	0	0:00.61	203	3	0	0:00.91	545	7	0	0:02.16
20	134	2	0	0:00.84	104	2	0	0:00.70	134	2	0	0:00.85
25	97	3	0	0:00.97	142	3	0	0:01.25	95	3	0	0:00.95
30	279	5	0	0:02.81	202	3	0	0:02.17	256	5	0	0:02.59
40	318	4	0	0:05.39	7851	5	0	4:46.24	250	4	0	0:04.42
50	349	5	0	0:07.61	11127	10	0	3:56.08	302	4	0	0:06.66
60	329	5	0	0:10.27	9020	6	0	4:32.28	325	5	0	0:10.63
70	8143	10	0	5:41.42	139015	6	0.22	>1ч	356	6	0	0:15.13
80	273	5	0	0:14.59	263145	6	3.61	>1ч	790	4	0	0:39.95
90	4129	5	0	4:31.13	271814	6	2.15	>1ч	1013	5	0	1:07.88
100	329	4	0	0:27.42	352614	6	2.96	>1ч	320	5	0	0:27.16

Таблица 1.2: Сравнение эффективности аппроксимаций

Основными показателями эффективности аппроксимации являются количество решенных линейаризованных задач, время работы алгоритма и число улучшений целевой функции. Сравнивая аппроксимации, можно отметить, что на небольших размерностях (до 30) все задачи решаются в одинаковой степени эффективно, т.е. нельзя говорить о преимуществах той или иной аппроксимации.

Однако на размерностях 8, 20 и 30 аппроксимация  $\mathcal{R}_2$  выигрывает по числу решенных линейаризованных задач, а следовательно, и по времени счета. Третья аппроксимация в этом случае выглядит предпочтительнее первой, за исключением примера размерности 15, где пришлось решить достаточно много линейаризованных задач.

Начиная с размерности 40, вторая аппроксимация теряет свои преимущества: число линейаризованных задач увеличивается слишком быстро, а в задачах, где  $n > 60$  глобальное решение получить не удастся. В данном случае использование дополнительной



информации (текущей точки  $z^k$ ) не дает преимуществ.

Третья аппроксимация при возрастании размерности оказывается стабильнее первой. Это можно видеть на размерностях 70 и 90. Единственным случаем, когда третий набор точек оказывается хуже, является пример с  $n = 80$ .

Таким образом, после анализа табл. 1.2 можно отказаться от применения аппроксимации  $\mathcal{R}_2$ , однако выбор между первой и третьей аппроксимациями требует дополнительного исследования.

На графике (рис. 1.6.1) приведены сравнительные данные о количестве решенных линейаризованных задач с использованием первой и третьей аппроксимаций на всех размерностях от 6 до 50.

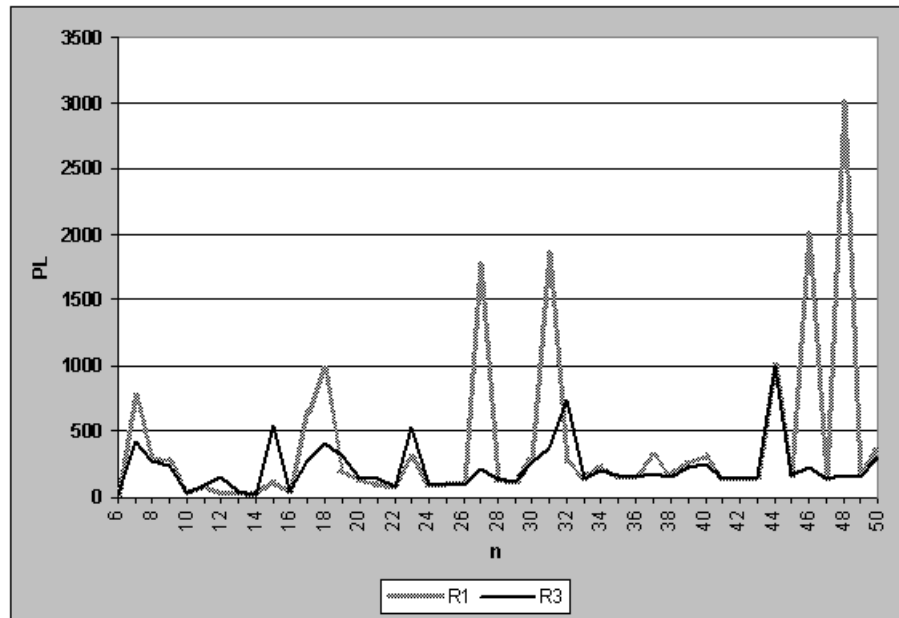


Рис. 1.1: Сравнение аппроксимаций

Несмотря на несколько примеров, в которых аппроксимация  $\mathcal{R}_1$  требует решения меньшего числа линейаризованных задач, третья аппроксимация выглядит более выигрышной. Заметим к тому же, что суммарное число решенных линейаризованных задач на всех размерностях при использовании третьей аппроксимации составило 9947 против 17865 с применением первой аппроксимации.

С учетом полученных результатов в дальнейших экспериментах будет использоваться только третья аппроксимация.

## 1.6.2 Согласование точностей локального и глобального поиска

В описанном выше алгоритме точность глобального поиска существенно зависит от точности  $\tau_k$  нахождения критических точек. Одним из вариантов реализации глобального поиска может быть нахождение на каждой итерации критических точек с постоянной, высокой точностью  $\tau$ . Другой вариант — поиск критических точек на начальных итерациях с низкой точностью, с последующим уменьшением величины  $\tau_k$ .

На следующем этапе тестирования алгоритма глобального поиска предполагается сравнить работу АГП при постоянной точности локального поиска  $\tau = 0.00005$  и изменяющейся точности

$$\tau_0 = 0.01, \tau_{k+1} = \frac{1}{5}\tau_k, k = 0, 1, 2, \dots$$

Поскольку критические точки во втором случае на начальных шагах находятся достаточно грубо, можно ожидать увеличения числа итераций АГП по сравнению с постоянной точностью. Но, благодаря уменьшению числа решенных линеаризованных задач в каждом локальном поиске, предполагается уменьшение общего времени счета. Данные по работе алгоритма приведены в табл. 1.3.

Как и следовало ожидать, при переменной точности (уменьшающемся  $\tau_k$ ) во всех примерах было решено меньшее число линеаризованных задач. Следствием этого стало увеличение числа найденных критических точек в задачах размерности 8, 25, 40, 50 и 80. В остальных случаях с уменьшением числа линеаризованных задач, число итераций метода не увеличилось. Возможно, это объясняется наличием в алгоритме глобального поиска на шагах 2 и 6 дополнительных критериев останова по значению функции, в связи с чем не возникает необходимости каждый раз уменьшать точность локального поиска  $\tau_k$  до необходимого значения  $\tau_k \leq \chi$ .

Таким образом, в результате первых двух этапов численного эксперимента выбрана наиболее подходящая аппроксимация поверхности уровня и установлено правило построения последовательности точностей нахождения критических точек, что в значительной степени увеличивает эффективность алгоритма глобального поиска.

$n$	Постоянная точность				Переменная точность			
	$PL$	$St$	$F_{glob}$	$T$	$PL$	$St$	$F_{glob}$	$T$
6	24	2	0	0:00.17	20	2	0	0:00.17
8	358	2	0	0:00.91	266	3	0	0:00.72
10	42	2	0	0:00.25	33	2	0	0:00.24
15	545	7	0	0:02.16	388	7	0	0:01.67
20	134	2	0	0:00.85	125	2	0	0:00.84
25	95	3	0	0:00.95	74	5	0	0:00.83
30	256	5	0	0:02.59	131	5	0	0:01.53
40	250	4	0	0:04.42	141	5	0	0:02.91
50	302	4	0	0:06.66	234	6	0	0:05.28
60	325	5	0	0:10.63	106	5	0	0:03.72
70	356	6	0	0:15.13	164	6	0	0:07.53
80	790	4	0	0:39.95	142	5	0	0:08.08
90	1013	5	0	1:07.88	863	5	0	0:58.03
100	320	5	0	0:27.16	145	5	0	0:13.31

Таблица 1.3: **Изменение точности локального поиска**

### 1.6.3 Решение серий задач

Для выявления общих свойств метода далее были рассмотрены серии задач, генерация которых происходила ранее описанным образом. Одновременно была увеличена размерность рассматриваемых задач.

Для сравнения эффективности алгоритма глобального поиска, те же серии задач были предложены для решения известной процедуре `quadprog` (Matlab 6.5 [143]), реализующей метод активных множеств для задач квадратичного программирования [126], и решателю `RATN` [142], специально предназначенному для решения линейных задач дополнителности. Отметим, что с помощью `quadprog` решались задачи математического программирования (1.2.1), а решателю `RATN` предлагались ЛЗД в исходной постановке ( $\mathcal{LCP}$ )–(1.1.1).

Ниже в табл. 1.4 производится сравнение численной эффективности известных пакетов прикладных программ и программного комплекса, разработанного в диссертации.

В табл. 1.4 использовались следующие обозначения:  $N$  — число протестированных за-

дач соответствующей размерности,  $LocSol$  — число задач, решенных только специальным методом локального поиска, без использования процедур глобального поиска,  $LP$  и  $St$  — среднее количество решенных линеаризованных задач и пройденных критических точек соответственно, потребовавшихся для нахождения глобального решения в одной задаче из серии,  $T$  — среднее время решения одной задачи алгоритмом глобального поиска. Для сравнения в колонках с названиями  $Quadprog$  и  $PATN$  приведено число задач из серии, в которых получено глобальное решение с помощью решателей  $quadprog$  и  $PATN$  соответственно.

Как можно видеть из таблицы, с увеличением размерности уменьшается число задач, решенных с помощью одного лишь метода локального поиска. Это еще раз подчеркивает сложность задачи ЛЗД со знаконеопределенной матрицей.

Так, начиная с размерности 35, локальным поиском не удалось найти глобального решения ни в одной задаче. При этом стоит отметить, что с ростом размерности задач среднее число пройденных алгоритмом критических точек практически не меняется, оставаясь, в среднем, в пределах 4-6. Поскольку число локальных минимумов с ростом размерности задачи увеличивается, этот факт указывает на высокую способность АГП "отсекать" большое число локальных экстремумов на каждой итерации.

Что касается результатов работы  $quadprog$  и  $PATN$ , то данные решатели оказываются эффективны на небольших размерностях. Однако с ростом числа переменных, увеличивается число нерешенных (глобально) задач. Для решателя  $PATN$  резкий скачок уменьшения количества решенных задач происходит при переходе к размерности 40, для  $quadprog$  — после размерности 50. Отметим еще раз, что алгоритм глобального поиска справился со всеми предложенными задачами, т.е. нашел в них глобальное решение.

На рис. 1.6.3 представлены графики, показывающие характер изменения среднего числа решенных линеаризованных задач ( $PL$ ), найденных критических точек ( $St$ ) и среднего времени ( $T$ ) решения одной задачи с увеличением размерности. Поскольку показатели средних величин могут отражать реальное поведение алгоритма только при большом числе решенных задач, для графиков использовались данные из таблицы до размерности 60. Как можно видеть из рисунка, наибольшей скоростью роста обладает кривая времени. Рост числа линеаризованных задач имеет почти линейную скорость. Число итераций алгоритма возрастает незначительно, о чем уже говорилось выше. Итак, можно сказать, что в ходе численного эксперимента представленный выше алгоритм глобального поиска показал высокую эффективность на (невыпуклых) задачах ЛЗД со знаконеопределенной

$n$	$N$	<i>Quadprog</i>	<i>PATH</i>	<i>LocSol</i>	<i>PL</i>	<i>St</i>	$T$
2	10000	10000	10000	9929	12,5	1,0	0:00.00
4	10000	9999	10000	8758	25,1	1,1	0:00.00
6	10000	9999	9998	6997	47,7	1,4	0:00.02
8	10000	9997	9996	5306	80,9	1,7	0:00.12
10	1000	993	992	390	110,9	2,0	0:00.32
15	1000	976	989	169	184,6	2,7	0:00.19
20	1000	979	983	71	273,2	3,3	0:01.49
25	1000	973	955	49	356,1	3,7	0:02.67
30	100	94	92	3	373,9	4,2	0:03.36
40	100	91	75	0	450,3	4,5	0:05.04
50	100	78	52	0	533,8	4,9	0:07.64
60	100	43	31	0	587,3	5,6	0:12.24
70	10	4	5	0	453,2	6,0	0:14.02
80	10	2	6	0	457,9	4,8	0:14.86
90	10	4	2	0	570,1	5,3	0:18.23
100	10	4	4	0	434	5,1	0:29.32
110	10	4	5	0	675	6,2	0:45.64
120	10	3	6	0	399	5,3	2:29.47
130	10	2	4	0	478	5,4	1:58.23
140	10	1	6	0	712	5,1	6:56.82
150	10	1	5	0	465	5,4	8:51.44
200	10	0	4	0	532	4,9	12:14.04
300	10	1	2	0	633	4,8	16:23.78
400	10	0	4	0	845	5,3	18:46.02

Таблица 1.4: Решение серий задач

матрицей  $M$  вплоть до размерности  $n = 400$ , обойдя по количеству решенных глобально задач решатели *PATH* и *quadprog*.

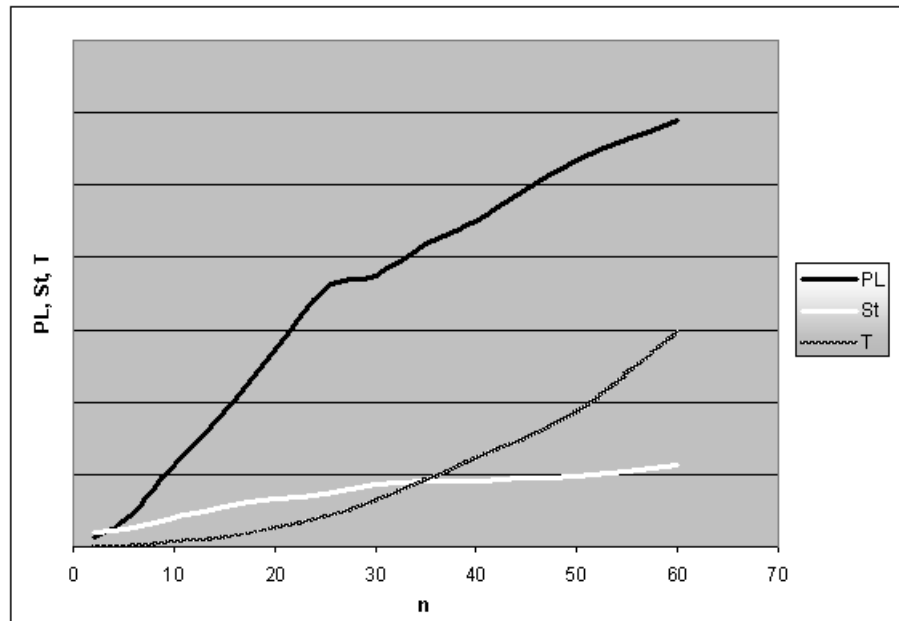


Рис. 1.2: Решение серии задач

## 1.7 Основные результаты главы

Задача линейной дополнителности представляет собой популярный математический объект, имеющий большое количество практических приложений.

Несмотря на кажущуюся внешнюю простоту постановки, решение данной задачи в случае знаконеопределенности матрицы представляет собой очень сложную задачу. Для решения данной задачи можно избрать две стратегии [100, 127]: работать напрямую с нелинейным ограничением-равенством [122, 123] или решать оптимизационную задачу [117, 131]. В диссертации был выбран второй путь. В данной главе была разработана методика решения линейных задач дополнителности со знаконеопределенными матрицами, основанная на теории глобального поиска из [69, 70] для d.c. целевой функции. Разработаны новые алгоритмы локального и глобального поиска для данной задачи. Результаты тестирования демонстрируют эффективность предложенного подхода на достаточно широком поле невыпуклых задач высокой размерности.

## Глава 2

# Поиск оптимистических решений в линейной двухуровневой задаче

Глава посвящена разработке численных методов поиска оптимистических (оптимальных) решений в линейной задаче двухуровневого программирования. Для решения поставленной задачи предлагается использовать вариационный подход, заключающийся в сведении исходной двухуровневой задачи к задаче математического программирования с невыпуклым ограничением-равенством, представимым в виде разности двух выпуклых функций (d.c. функции).

В начале главы представлены полученные теоретические результаты по решению задач с d.c. ограничением-равенством. В частности, доказываются и обобщаются на случай минимизирующих последовательностей условия глобальной оптимальности в задаче с d.c. равенством. Затем предлагается алгоритм глобального поиска для задачи с невыпуклым ограничением-равенством. Далее предложенная теоретическая база применяется для разработки метода решения двухуровневой задачи, где целевые функции верхнего и нижнего уровня являются линейными функциями. На основе условий оптимальности предлагается алгоритм решения линейной двухуровневой задачи, включающий специальный метод локального поиска критических точек. В заключение представлены результаты многоэтапного вычислительного эксперимента по решению серий специальным образом сгенерированных двухуровневых задач.

## 2.1 Постановка линейной двухуровневой задачи

Рассмотрим двухуровневую задачу с линейными целевыми функциями на верхнем и нижнем уровнях [11, 89, 90, 92, 96, 98, 103, 114, 132, 136, 138]. Предположим, что из всех возможных решений на нижнем уровне выбирается то, которое благоприятствует достижению цели верхнего уровня. Такая постановка двухуровневой задачи носит название оптимистической [90, 103] и может быть записана, например, следующим образом:

$$(\mathcal{BP}) \quad \left. \begin{aligned} f(x, y) &\triangleq \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y}, \\ x \in X &\triangleq \{x \in \mathbb{R}^m \mid Ax \leq b\}, \\ y \in Y_*(x) &\triangleq \text{Arg min}_y \left\{ \langle d^1, y \rangle \mid y \in Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1x + B_1y \leq b^1\} \right\}, \end{aligned} \right\} \quad (2.1.1)$$

где  $c \in \mathbb{R}^m$ ,  $d, d^1 \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^p$ ,  $b^1 \in \mathbb{R}^q$  — заданные векторы,  $A$ ,  $A_1$ ,  $B_1$  — матрицы размера  $(p \times m)$ ,  $(q \times m)$ ,  $(q \times n)$  соответственно.

Сделаем следующие общепринятые предположения относительно задачи  $(\mathcal{BP})$ –(2.1.1).

**(Н1)** : Функция  $f(x, y)$  ограничена снизу на непустом множестве

$$Z \triangleq \{x \in \mathbb{R}^m, y \in \mathbb{R}^n \mid Ax \leq b, A_1x + B_1y \leq b^1\}.$$

**(Н2)** : Функция  $\langle d^1, y \rangle$  ограничена снизу на множестве  $Y(x)$  для всех  $x \in X$ , так что

$$\inf_x \inf_y \{ \langle d^1, y \rangle \mid y \in Y(x), x \in X \} > -\infty.$$

Несмотря на внешнюю простоту, разработка эффективных численных методов решения линейных двухуровневых задач большой размерности является весьма сложной проблемой. Этот факт объясняется, в частности, тем, что иерархическая структура задачи порождает так называемую скрытую невыпуклость. Следующий пример иллюстрирует неявную невыпуклость линейных двухуровневых задач.

**Пример 2.1.1** [103] *Рассмотрим линейную двухуровневую задачу со скалярными переменными на верхнем и нижнем уровнях соответственно:*

$$\begin{aligned} (P_U) \quad & \left. \begin{aligned} f(x, y) &= x + 3y \downarrow \min_{x, y}, \\ 1 \leq x \leq 6, \quad y \in Y_*(x) &= \text{Sol}(P_L), \end{aligned} \right\} \\ (P_L) \quad & \left. \begin{aligned} -y &\downarrow \min_y, \\ x + y \leq 8, \quad x + 4y \geq 8, \quad x + 2y &\leq 13. \end{aligned} \right\} \end{aligned}$$



На рис. 2.1 представлена графическая иллюстрация этой задачи. Множество  $M$  определяет совокупность пар  $(x, y)$ , удовлетворяющих ограничениям верхнего и нижнего уровня. Минимизируя функцию нижнего уровня на допустимом множестве при каждом фиксированном  $x$ , получаем ломаную  $AED$ , которая представляет собой допустимое множество задачи верхнего уровня:

$$y(x) = \begin{cases} 6.5 - 0.5x, & 1 \leq x \leq 3, \\ 8 - x, & 3 \leq x \leq 6. \end{cases}$$

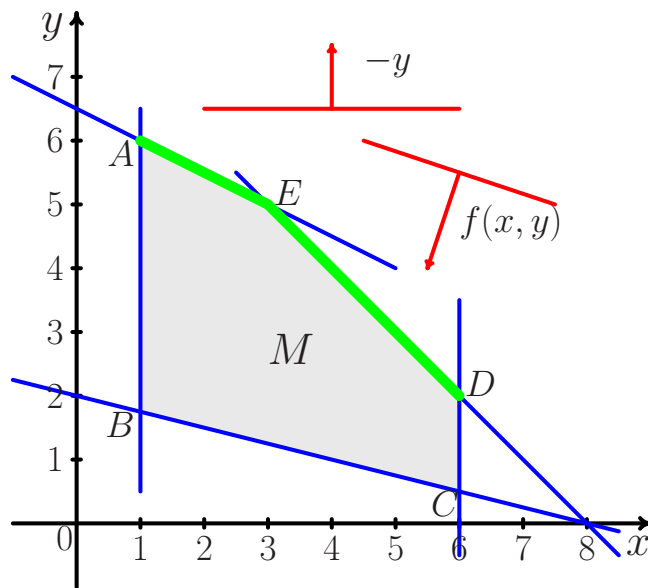


Рис. 2.1: Линейная двухуровневая задача

Нетрудно видеть, что допустимое множество верхнего уровня оказывается невыпуклым, и целевая функция верхнего уровня  $f(x, y)$  на нем принимает следующий вид:

$$f(x, y(x)) = \begin{cases} 19.5 - 0.5x, & 1 \leq x \leq 3, \\ 24 - 2x, & 3 \leq x \leq 6. \end{cases}$$

Задача заключается в минимизации кусочно-линейной функции  $f(x, y(x))$  по  $x$ . Нетрудно видеть, что глобальное решение этой задачи дает точка  $D = (6, 2)$ , значение целевой функции в которой  $f_*(6, 2) = 12$ . #

Из представленного примера нетрудно видеть, что даже в самом простом линейном случае двухуровневая задача обладает невыпуклой структурой, а, как известно, разработка эффективных методов решения невыпуклых задач высокой размерности является одной из актуальных проблем современной теории и методов оптимизации [12, 15, 31, 69, 110, 126]. Для решения сформулированной двухуровневой задачи далее предлагается

подход, который использует ее сведение к задаче математического программирования с билинейным ограничением-равенством.

## 2.2 Сведение двухуровневой задачи к задаче с d.c. ограничением

В данном параграфе будет показано, каким образом двухуровневая задача  $(\mathcal{BP})$ –(2.1.1) может быть сведена к одноуровневой задаче с билинейным ограничением-равенством.

С этой целью для нижнего уровня двухуровневой задачи  $(\mathcal{BP})$ –(2.1.1), который представляет собой следующую задачу параметрического линейного программирования:

$$(\mathcal{F}(x)) \quad \langle d^1, y \rangle \downarrow \min_y, \quad y \in Y(x) \triangleq \{y \in \mathbb{R}^n \mid A_1x + B_1y \leq b^1\}, \quad (2.2.1)$$

запишем двойственную задачу [10], считая  $x \in X$  заданным параметром:

$$(\mathcal{DF}(x)) \quad \langle A_1x - b^1, v \rangle \uparrow \max_v, \quad v \in V \triangleq \{v \in \mathbb{R}^q \mid vB_1 = -d^1, v \geq 0\}. \quad (2.2.2)$$

Нетрудно проверить [10], что на допустимых векторах  $x, y, v$  (т.е. таких, что  $y \in Y(x), v \in V$ ) выполнено следующее неравенство:

$$\langle d^1, y \rangle - \langle A_1x - b^1, v \rangle \geq 0. \quad (2.2.3)$$

Кроме того, с учетом сделанных предположений **(Н1)** – **(Н2)**, взаимодвойственные задачи  $(\mathcal{F}(x))$ –(2.2.1) и  $(\mathcal{DF}(x))$ –(2.2.2) при фиксированном  $x \in X$  разрешимы, а значит по теореме двойственности [10] найдутся векторы  $y \in \mathbb{R}^n$  и  $v \in \mathbb{R}^q$  такие, что тройка  $(x, y, v)$  удовлетворяет системе

$$\left. \begin{aligned} \langle d^1, y \rangle &= \langle A_1x - b^1, v \rangle, \\ A_1x + B_1y &\leq b^1, \quad vB_1 = -d^1, \quad v \geq 0. \end{aligned} \right\} \quad (2.2.4)$$

Заменяя в задаче  $(\mathcal{BP})$ –(2.1.1) экстремальное ограничение эквивалентной ему системой (2.2.4), получим следующую невыпуклую задачу математического программирования:

$$(\mathcal{P}) \quad \left. \begin{aligned} \langle c, x \rangle + \langle d, y \rangle &\downarrow \min_{x,y,v} \\ (x, y, v) &\in S, \\ \langle d^1, y \rangle &= \langle A_1x - b^1, v \rangle, \end{aligned} \right\} \quad (2.2.5)$$

где

$$S \triangleq \{Ax \leq b, A_1x + B_1y \leq b^1, vB_1 = -d^1, v \geq 0\}. \quad (2.2.6)$$

Нетрудно видеть, что невыпуклость в задаче  $(\mathcal{P})$ –(2.2.5) порождается билинейным ограничением-равенством, более точно, наличием произведения  $\langle A_1x, v \rangle$ , при этом множество  $S$  выпукло. Кроме того, при переходе от задачи  $(\mathcal{BP})$ –(2.1.1) к  $(\mathcal{P})$ –(2.2.5) увеличивается ее размерность: появляется вектор переменных  $v \in \mathbb{R}^q$ . Взаимосвязь задач  $(\mathcal{P})$ –(2.2.5) и  $(\mathcal{BP})$ –(2.1.1) устанавливается следующей теоремой.

**Теорема 2.2.1** [90, 103]. *Для того, чтобы пара  $(x^*, y^*)$  являлась решением двухуровневой задачи  $(\mathcal{BP})$ –(2.1.1), необходимо и достаточно существования вектора  $v^*$  такого, чтобы тройка  $(x^*, y^*, v^*)$  была решением задачи  $(\mathcal{P})$ –(2.2.5).*

**Доказательство.** Пусть  $(x^*, y^*) \in \text{Sol}(\mathcal{BP})$ , тогда очевидно найдется вектор  $v^*$ , являющийся решением задачи, двойственной к задаче нижнего уровня  $(\mathcal{DF}(x^*))$  задачи  $(\mathcal{BP})$ . При этом тройка  $(x^*, y^*, v^*)$  будет удовлетворять системе (2.2.4), т.е. является допустимой в задаче  $(\mathcal{P})$ .

Пусть теперь существует тройка  $(\bar{x}, \bar{y}, \bar{v})$  допустимая в задаче  $(\mathcal{P})$  такая, что  $f(\bar{x}, \bar{y}) < f(x^*, y^*)$ . Тогда пара  $(\bar{x}, \bar{y})$  будет допустимой в  $(\mathcal{BP})$  и при этом  $f(\bar{x}, \bar{y}) < f(x^*, y^*)$ , что противоречит тому, что  $(x^*, y^*) \in \text{Sol}(\mathcal{BP})$ . Доказательство достаточности проводится аналогично. #

Таким образом, для отыскания решения в двухуровневой задаче  $(\mathcal{BP})$ –(2.1.1) можно решать невыпуклую задачу оптимизации  $(\mathcal{P})$ –(2.2.5), а затем использовать ее решение  $(x^*, y^*, v^*)$  для построения решения  $(x^*, y^*)$  задачи  $(\mathcal{BP})$ –(2.1.1).

Одним из стандартных подходов к решению задач с ограничениями типа равенства является применение метода штрафов [9, 36, 57, 60, 84, 126]. В этом случае невыпуклость задачи (билинейное ограничение-равенство) переходит в целевую функцию и возникает дополнительный штрафной параметр [90, 94, 95, 138]. В частности, существуют программные реализации стратегии глобального поиска для задач минимизации функции, представимой в виде разности двух выпуклых функций (d.c. функции), для решения линейно-квадратичных двухуровневых задач методом штрафов [79, 80].

В работе предлагается решать задачу  $(\mathcal{P})$ –(2.2.5) напрямую, с использованием теории глобального поиска для задач с d.c. ограничением [18, 72]. Действительно, билинейное ограничение в задаче  $(\mathcal{P})$ –(2.2.5) можно представить в виде разности двух выпуклых

функций, например, следующим образом:

$$F(x, y, v) = g(x, y, v) - h(x, v) = 0, \quad (2.2.7)$$

где  $g(x, y, v) = \frac{1}{4}\|A_1x - v\|^2 + \langle d^1, y \rangle + \langle b^1, v \rangle$ ,  $h(x, v) = \frac{1}{4}\|A_1x + v\|^2$ .

Таким образом, задача  $(\mathcal{P})$ –(2.2.5), в которой билинейное равенство представлено в виде (2.2.7), оказывается задачей с d.c. ограничением-равенством ( $z \triangleq (x, y, v)$ )

$$f(z) \downarrow \min, \quad z \in S, \quad F(z) = 0.$$

Для ее решения будем применять теорию глобального поиска [69] в задачах с d.c. ограничением. Теоретическое обоснование подхода к решению задачи с d.c. равенством представлено в следующих параграфах.

### 2.3 Условия глобальной оптимальности для задачи минимизации с d.c. ограничением-равенством

Рассмотрим следующую задачу с d.c. ограничением-равенством в общем виде:

$$(\mathcal{P}_0) \quad \left. \begin{array}{l} f(x) \downarrow \min, \quad x \in S, \\ F(x) = 0, \end{array} \right\} \quad (2.3.1)$$

где функция  $F(x)$  представима в виде разности двух выпуклых функций  $g(x)$  и  $h(x)$ :  $F(x) \triangleq g(x) - h(x)$ ,  $S$  — выпуклое множество из  $\mathbb{R}^n$ .

В [67, 72] разработана теория решения задач с d.c. ограничением-неравенством вида

$$(\mathcal{DCC}) \quad \left. \begin{array}{l} f(x) \downarrow \min, \quad x \in S, \\ F(x) \triangleq g(x) - h(x) \leq 0. \end{array} \right\} \quad (2.3.2)$$

Предполагается, что ограничение-неравенство в задаче  $(\mathcal{DCC})$ –(2.3.2) активно на ее решении. В противном случае задача  $(\mathcal{DCC})$ –(2.3.2) теряет особенность, связанную с невыпуклым ограничением.

Нетрудно показать, что в случае активности ограничения-неравенства значение задачи  $(\mathcal{DCC})$ –(2.3.2) совпадает со значением задачи  $(\mathcal{P}_0)$ –(2.3.1). Более того, справедлива следующая лемма.

**Лемма 2.3.1** *Если  $z \in \text{Sol}(\mathcal{DCC})$  и  $F(z) = 0$ , то  $z \in \text{Sol}(\mathcal{P}_0)$ .*

**Доказательство.** Поскольку допустимое множество задачи  $(DCC)$  с ограничением-неравенством содержит допустимое множество задачи  $(P_0) : D(P_0) \subset D(DCC)$ , то  $\mathcal{V}(DCC) \leq \mathcal{V}(P_0)$ .

Тогда, если  $z \in \text{Sol}(DCC)$  и, что важно,  $F(z) = 0$ , то  $f(z) \leq \mathcal{V}(P_0)$ . Но очевидно, что  $z$  является допустимой в  $(P_0)$ . Поэтому  $f(z) = \mathcal{V}(P_0)$ , что и требовалось доказать.

#

Далее, нетрудно видеть, что в силу леммы 2.3.1 всякое достаточное условие глобальной оптимальности в задаче  $(DCC)$ –(2.3.2) для точки  $z \in S$ ,  $F(z) = 0$  окажется достаточным для того, чтобы точка  $z$  была и решением задачи  $(P_0)$ –(2.3.1).

В частности для задачи  $(P_0)$ –(2.3.1) будут справедливы достаточные условия глобальной оптимальности, доказанные для задачи  $(DCC)$ –(2.3.2) [67], которые сформулированы ниже.

**Теорема 2.3.1** [67] Пусть выполнены следующие условия регулярности:

$$\exists v \in S : F(v) > 0, \quad (2.3.3)$$

$$(\mathcal{H}) \quad \left. \begin{array}{l} \forall y \in S : F(y) = 0 \ (g(y) = h(y)) \\ \exists p = p(y) \in S : g(p) - g(y) < \langle \nabla h(y), p - y \rangle. \end{array} \right\} \quad (2.3.4)$$

Если  $z \in S$ ,  $F(z) = 0$  и, кроме того, справедливо условие

$$(\mathcal{E}_0) \quad \left. \begin{array}{l} \forall (y, \beta) : h(y) = \beta, \ y \in S, \\ g(y) \leq \beta \leq \sup(g, S), \\ g(x) - \beta \geq \langle \nabla h(y), x - y \rangle \\ \forall x \in S : f(x) \leq f(z), \end{array} \right\} \quad (2.3.5)$$

то тогда  $z \in \text{Sol}(P_0)$ .

#

Доказательство теоремы полностью повторяет доказательство теоремы из [67] и поэтому здесь не приводится.

Далее рассмотрим вопрос о необходимых условиях глобальной оптимальности для задачи  $(P_0)$ –(2.3.1).

**Теорема 2.3.2** Пусть выполнено предположение

$$(\mathcal{H}_1) \quad \exists v \in S : F(v) > 0, \ f(v) < \mathcal{V}(P_0) \triangleq \inf_x \{f(x) \mid x \in S, F(x) = 0\}. \quad (2.3.6)$$

Тогда, если  $z \in \text{Sol}(P_0)$ , то

$$(\mathcal{E}_1) \quad \left. \begin{array}{l} \forall (y, \beta) : h(y) = \beta, \ y \in S, \\ g(x) - \beta \geq \langle \nabla h(y), x - y \rangle \ \forall x \in S : f(x) \leq f(z). \end{array} \right\} \quad (2.3.7)$$

**Доказательство.** Пусть  $(\mathcal{E}_1)$ –(2.3.7) нарушено:

$$\exists(y, \beta) : h(y) = \beta, y \in S, \exists u \in S, f(u) \leq f(z)$$

$$g(u) - \beta < \langle \nabla h(y), u - y \rangle.$$

Тогда из выпуклости  $h(\cdot)$  следует

$$0 < \beta - g(u) + h(u) - h(y) = -F(u).$$

Итак, нашелся элемент  $u$  такой, что  $u \in S, f(u) \leq f(z), F(u) < 0$ . Теперь из  $(\mathcal{H}_1)$ –(2.3.6) вытекает существование  $\lambda \in ]0, 1[ : F(x(\lambda)) = 0$ , где  $x(\lambda) = \lambda u + (1 - \lambda)v$ .

При этом, в силу выпуклости  $S$  и  $f(\cdot)$ , имеем:  $x(\lambda) \in S$ , и, поскольку  $f(v) < f(z)$ ,

$$f(x(\lambda)) \leq \lambda f(u) + (1 - \lambda)f(v) < f(z).$$

Таким образом, нашелся допустимый элемент  $x(\lambda) \in S, F(x(\lambda)) = 0$ , лучший по значению целевой функции, чем  $z : f(x(\lambda)) < f(z)$ . Это противоречит тому, что  $z \in \text{Sol}(\mathcal{P}_0)$ .

#

Следовательно, при предположении  $(\mathcal{H}_1)$ –(2.3.6) условие  $(\mathcal{E}_1)$ –(2.3.7) оказывается и необходимым для того, чтобы точка  $z$  была решением задачи  $(\mathcal{P}_0)$ –(2.3.1).

**Замечание 2.3.1** *Отметим, что в случае ограничения-равенства можно предполагать выполнение условия регулярности  $(\mathcal{H}_1)$ –(2.3.6), не ограничивая общности. Действительно, в случае его невыполнения можно рассмотреть ограничение-равенство в виде  $\bar{F}(x) \triangleq -F(x) = 0$ . Тогда либо условие регулярности будет выполнено для функции  $\bar{F}(x)$  (которая также является д.с. функцией: достаточно лишь поменять местами функции  $g$  и  $h$  в д.с. разложении), либо получаем, что*

$$F(x) = 0 \quad \forall x \in S : f(x) \leq \mathcal{V}(\mathcal{P}_0). \quad (2.3.8)$$

Последнее означает, что решение задачи  $(\mathcal{P}_0)$ –(2.3.1) совпадает с решением релаксированной задачи

$$f(x) \downarrow \min, x \in S, \quad (2.3.9)$$

и ограничение  $F(x) = 0$  можно не принимать во внимание. Другими словами, теряется особенность, связанная с ограничением-равенством.

В теории глобального поиска [69] ключевую роль играет функция, задающая (в д.с. разложении) базовую невыпуклость задачи. Именно по этой функции в дальнейшем производится линеаризация при построении численных методов, и именно эта функция входит в условия глобальной оптимальности (УГО) в линеаризованном виде.

В задаче (DCC)–(2.3.2) такой функцией является функция  $h(\cdot)$ , поскольку при  $h(x) \equiv 0$  получаем выпуклую задачу. Для выполнения условия регулярности  $(\mathcal{H}_1)$ –(2.3.6) в задаче  $(\mathcal{P}_0)$ –(2.3.1) выбирается ограничение типа равенства в виде  $g(x) - h(x) = 0$  или  $h(x) - g(x) = 0$ . И, соответственно, линеаризация в УГО происходит либо по функции  $h(\cdot)$ , либо по функции  $g(\cdot)$ . При этом в задаче с ограничением-равенством говорить о функции, задающей базовую невыпуклость, вообще говоря, нельзя, поскольку обе функции  $g(\cdot)$  и  $h(\cdot)$  порождают невыпуклость, если не являются аффинными.

В частности, какую бы функцию мы не положили равной нулю,  $g(\cdot)$  или  $h(\cdot)$ , получаем, вообще говоря, невыпуклую задачу с ограничением-равенством.

Чтобы обосновать, почему в условиях глобальной оптимальности линеаризация происходит именно по функции  $h(\cdot)$ , докажем предварительно следующую теорему.

**Теорема 2.3.3** Пусть выполнено предположение  $(\mathcal{H}_1)$ –(2.3.6). Тогда задача  $(\mathcal{P}_0)$ –(2.3.1) эквивалентна задаче (DCC)–(2.3.2).

**Доказательство.** 1. Пусть некоторая точка  $z$  является решением задачи (DCC)–(2.3.2). Тогда ясно, что  $z \in S$ ,  $F(z) \leq 0$  и

$$f(z) \leq f(x) \quad \forall x \in S, \quad F(x) \leq 0.$$

Покажем, что  $F(z) = 0$ . Пусть это не так, т.е.  $F(z) < 0$ .

Поскольку по условию  $(\mathcal{H}_1)$ –(2.3.6)  $F(v) > 0$  и функция  $\theta(\alpha) := F(x_\alpha)$ , где  $x_\alpha = \alpha z + (1 - \alpha)v$ , является непрерывной, то  $\exists \alpha \in ]0, 1[ : F(x_\alpha) = 0$ . Кроме того, в силу выпуклости функции  $f(\cdot)$  имеем

$$f(x_\alpha) \leq \alpha f(z) + (1 - \alpha)f(v) < \mathcal{V}(\mathcal{P}_0).$$

Таким образом, получили допустимую в  $(\mathcal{P}_0)$ –(2.3.1) точку  $x_\alpha$ , для которой  $f(x_\alpha) < \mathcal{V}(\mathcal{P}_0)$ , что невозможно. Значит,  $F(x_\alpha) = 0$ , и тогда  $z \in \text{Sol}(\mathcal{P}_0)$ .

2. Пусть теперь  $z \in \text{Sol}(\mathcal{P}_0)$ . Предположим, что  $z \notin \text{Sol}(\text{DCC})$ , так что  $\exists u \in S :$

$$f(u) < f(z), \quad F(u) \leq 0, \quad u \in S.$$

Ясно, что в этом случае  $F(u) < 0$ , иначе бы  $u$  являлась допустимой в  $(\mathcal{P}_0)$  и лучшей, чем  $z$ , что противоречит предположению. Так же, как и в пункте 1 получаем, что  $\exists \alpha \in ]0, 1[$ :

$$F(u) < 0 = F(x_\alpha) < F(v),$$

$$f(x_\alpha) \leq \alpha f(v) + (1 - \alpha)f(u) < \mathcal{V}(\mathcal{P}_0).$$

Итак,  $x_\alpha \in S$ ,  $F(x_\alpha) = 0$  и при этом  $f(x_\alpha) < \mathcal{V}(\mathcal{P}_0)$ , что невозможно. #

С учетом результата теоремы 2.3.3 понятие базовой невыпуклости в задаче  $(\mathcal{P}_0)$ –(2.3.1) можно проинтерпретировать следующим образом. Пусть в задаче  $(\mathcal{P}_0)$ –(2.3.1) выполнены условия регулярности  $(\mathcal{H}_1)$ –(2.3.6). Предположим, что  $h(x) \equiv 0$ , тогда получаем следующую задачу:

$$(\mathcal{P}'_0) \quad \left. \begin{array}{l} f(x) \downarrow \min, x \in S, \\ g(x) = 0. \end{array} \right\} \quad (2.3.10)$$

При этом условия регулярности  $(\mathcal{H}_1)$ –(2.3.6) запишутся следующим образом:

$$(\mathcal{H}'_1) \quad \exists v \in S : f(v) < \mathcal{V}(\mathcal{P}_0), g(v) > 0. \quad (2.3.11)$$

В этом случае из теоремы 2.3.3 вытекает следующий результат.

**Следствие 2.3.1** Пусть выполнено  $(\mathcal{H}'_1)$ –(2.3.11). Тогда всякое решение задачи  $(\mathcal{P}'_0)$ –(2.3.10) является решением задачи

$$(\mathcal{CP}) \quad \left. \begin{array}{l} f(x) \downarrow \min, x \in S. \\ g(x) \leq 0. \end{array} \right\} \quad (2.3.12)$$

Обратно, всякое решение задачи  $(\mathcal{CP})$  является решением задачи  $(\mathcal{P}'_0)$ –(2.3.10). #

Таким образом, получаем, что при  $h(x) \equiv 0$  решение задачи  $(\mathcal{P}'_0)$ –(2.3.10) сводится к решению выпуклой задачи  $(\mathcal{CP})$ . Другими словами, можно считать, что именно функция  $h(x)$  задает базовую невыпуклость в задаче  $(\mathcal{P}_0)$ .

Заметим далее, что, как и для задачи  $(\mathcal{DCC})$ –(2.3.2), необходимые условия глобальной оптимальности обладают так называемым алгоритмическим (конструктивным) свойством, которое заключается в следующем: если  $(\mathcal{E}_1)$ –(2.3.7) нарушено, то существует возможность построить точку лучшую, чем исследуемая.

Действительно, пусть найдется тройка  $(y, \beta, u)$  такая, что

$$h(y) = \beta, \quad u \in S, \quad f(u) \leq f(z).$$



и на которых неравенство из  $(\mathcal{E}_1)$ –(2.3.7) нарушено, т.е.

$$g(u) - \beta < \langle \nabla h(y), u - y \rangle.$$

Отсюда, в силу выпуклости функции  $h(\cdot)$ , немедленно следует что

$$0 > g(u) - \beta - \langle \nabla h(y), u - y \rangle \geq g(u) - \beta - h(u) + h(y).$$

Значит точка  $u$  удовлетворяет условиям

$$u \in S, \quad f(u) \leq f(z), \quad F(u) = g(u) - h(u) < 0.$$

С другой стороны, в силу условия регулярности  $(\mathcal{H}_1)$ –(2.3.6) существует выпуклая комбинация точек  $u$  и  $\hat{v}$

$$x_\lambda = \lambda u + (1 - \lambda)\hat{v} \in S \quad \text{такая, что } F(x_\lambda) = 0.$$

При этом, в силу выпуклости функции  $f(\cdot)$ ,

$$f(x_\lambda) \leq \lambda f(u) + (1 - \lambda)f(\hat{v}) < f(z).$$

Иными словами, имеется возможность улучшить значение целевой функции  $f(\cdot)$ , построив точку, удовлетворяющую всем ограничениям задачи, в том числе  $F(x) = 0$ . В третьей главе (см. §3.1) представлен численный метод построения такой точки.

## 2.4 Минимизирующие последовательности

В этом параграфе будет произведено обобщение условий глобальной оптимальности на минимизирующие последовательности в задаче  $(\mathcal{P}_0)$ –(2.3.1).

**Определение 2.4.1** Последовательность  $\{z^k\}$  назовем минимизирующей в задаче  $(\mathcal{P}_0)$ –(2.3.1), если выполнены три следующих условия:

$$(Df1) : \quad \lim_{k \rightarrow \infty} f(z^k) = f_* \triangleq \inf(f, D), \quad (2.4.1)$$

$$\text{где } D \triangleq \{x \in S \mid F(x) = 0\};$$

$$(Df2) : \quad \{z^k\} \subset S; \quad (2.4.2)$$

$$(Df3) : \quad \lim_{k \rightarrow \infty} F(z^k) = 0. \quad (2.4.3)$$

Множество всех минимизирующих последовательностей в задаче  $(\mathcal{P}_0)$ –(2.3.1) будем обозначать через  $\mathcal{M} = \mathcal{M}(\mathcal{P}_0)$ .

Теперь введем следующую функцию

$$\begin{aligned}\varphi(z) &\triangleq \sup_{x,y,\beta} \{ \langle \nabla h(y), x - y \rangle + \beta - g(x) \mid x \in S, f(x) \leq f(z), \\ g(y) \leq \beta \leq \sup(g, S), \beta - h(y) = F(z) &\triangleq g(z) - h(z) \triangleq \xi \}.\end{aligned}\quad (2.4.4)$$

Поскольку при  $x = y = z \in S$  и  $\beta = \beta_0 \triangleq h(z)$  тройка  $(x, y, \beta_0)$  удовлетворяет ограничениям в определении (2.4.4) и, кроме того,

$$0 = \langle \nabla h(z), z - z \rangle + \beta_0 - g(z),$$

то получаем

$$\varphi(z) \geq 0 \quad \forall z \in S. \quad (2.4.5)$$

**Теорема 2.4.1** Пусть выполнено условие регулярности  $(\mathcal{H}_1)$ –(2.3.6), а функция  $f(\cdot)$  непрерывна на открытом множестве  $\Omega \supset S$ . Кроме того, пусть множество  $X_* = \{x \in S \mid f(x) \leq \mathcal{V}(\mathcal{P}_0)\}$  непусто и ограничено.

Тогда для любой последовательности  $\{z^k\} \in \mathcal{M}(\mathcal{P}_0)$  справедливо равенство:

$$(\mathcal{E}_M) : \quad \lim_{k \rightarrow \infty} \varphi(z^k) = 0. \quad (2.4.6)$$

**Доказательство.** Пусть последовательность  $\{z^k\}$  является минимизирующей, но условие  $(\mathcal{E}_M)$ –(2.4.6) нарушено. Тогда с точностью до подпоследовательности можно считать, что

$$\varphi(z^k) \geq 2\gamma > 0 \quad \forall r = 1, 2, \dots \quad (2.4.7)$$

По определению функции  $\varphi(\cdot)$  найдется тройка  $(x^k, y^k, \beta_k)$  такая, что  $\forall k = 1, 2, \dots$

$$x^k \in S, f(x^k) \leq f(z^k),$$

$$g(y^k) \leq \beta_k \leq \sup(g, S),$$

$$\beta_k - h(y^k) = F(z^k),$$

и при этом

$$\begin{aligned}0 < \gamma < \varphi(z^k) - \gamma &\leq \langle \nabla h(y^k), x^k - y^k \rangle + \beta_k - g(x^k) \leq \\ &\leq h(x^k) - h(y^k) + h(y^k) + F(z^k) - g(x^k) = F(z^k) - F(x^k).\end{aligned}$$

Итак, для последовательности  $\{x^k\}$  справедливы следующие соотношения:

$$x^k \in S, f(x^k) \leq f(z^k), F(z^k) - F(x^k) > \gamma. \quad (2.4.8)$$

Поскольку  $z^k \in \mathcal{M}(\mathcal{P}_0)$ , то  $\lim_{k \rightarrow \infty} f(z^k) = \mathcal{V}(\mathcal{P}_0)$ . В силу непрерывности функции  $f(\cdot)$  множество Лебега функции  $f(\cdot)$  замкнуто, а в силу ограниченности множества  $X_* = \{x \in S \mid f(x) \leq \mathcal{V}(\mathcal{P}_0)\}$ , ограничено [9]. Тогда последовательность  $\{x^k\}$  сходится с точностью до подпоследовательности к некоторой предельной точке  $\bar{x} \in S$ , причем

$$f(\bar{x}) \leq \mathcal{V}(\mathcal{P}_0).$$

Кроме того, из (2.4.8) получаем, что

$$F(\bar{x}) \leq \lim_{k \rightarrow \infty} F(z^k) - \gamma = \gamma < 0.$$

Теперь для точек  $\bar{x}$  и  $v$  из предположения  $(\mathcal{H}_1)$ –(2.3.6) построим выпуклую комбинацию

$$x_\lambda = \lambda \bar{x} + (1 - \lambda)v$$

такую, что  $F(x_\lambda) = 0$ ,  $\lambda \in ]0, 1[$ ,  $x_\lambda \in S$ . При этом в силу выпуклости  $f(\cdot)$

$$f(x_\lambda) \leq \lambda f(\bar{x}) + (1 - \lambda)f(v) < \mathcal{V}(\mathcal{P}_0),$$

так что  $f(x_\lambda) < \mathcal{V}(\mathcal{P}_0)$  в допустимой точке  $x_\lambda$ , что невозможно. Полученное противоречие доказывает, что  $\lim_{k \rightarrow \infty} \varphi(z^k) = 0$ . #

Утверждение теоремы обратимо при выполнении условий регулярности из теоремы 2.3.1 о достаточных условиях глобальной оптимальности.

**Теорема 2.4.2** Пусть в задаче  $(\mathcal{P}_0)$ –(2.3.1) функция  $f(\cdot)$  полунепрерывна сверху на открытом множестве  $\Omega \supset S$ , и выполнены условия регулярности (2.3.3) и (2.3.4). Пусть для некоторой последовательности  $\{z^k\} \subset S$  выполнено условие  $(\mathcal{E}_M)$ –(2.4.6) и имеет место равенство  $\lim_{k \rightarrow \infty} F(z^k) = 0$ . Тогда последовательность  $\{z^k\}$  оказывается минимизирующей в задаче  $(\mathcal{P}_0)$ . #

Как и в случае с теоремой 2.3.1, доказательство данной теоремы аналогично доказательству соответствующей теоремы для задачи  $(DCC)$ –(2.3.2) с ограничением-неравенством и в работе не приводится.

## 2.5 Стратегия глобального поиска

Для решения вопроса, является ли исследуемая точка  $z \in S, F(z) = 0$ , (приближенным) решением задачи  $(\mathcal{P}_0)$ –(2.3.1), теоремы 2.4.1 и 2.4.2 предлагают рассмотреть следующую задачу:

$$\left. \begin{aligned} \Psi(x, y, \beta) &\triangleq \langle \nabla h(y), x - y \rangle + \beta - g(x) \uparrow \max_{x, y, \beta}, \\ x \in S, f(x) &\leq f(z), g(y) \leq \beta \leq \sup(g, S), \\ \beta - h(y) = F(z) &\triangleq g(z) - h(z) = 0. \end{aligned} \right\} \quad (2.5.1)$$

Данная задача совпадает с задачей, возникающей в случае д.с. ограничения-неравенства. Следовательно, при выполнении дополнительных условий регулярности для нахождения глобального минимума задачи  $(\mathcal{P}_0)$ –(2.3.1) необходимо выполнить те же шаги, что и для задачи с д.с. ограничением-неравенством [72]. А именно, задачу (2.5.1) предлагается разбить на несколько более простых этапов.

а) Пусть задано число  $\beta$  такое, что

$$\beta_- \triangleq \inf(g, S) \leq \beta \leq \beta_+ \triangleq \sup(g, S), \quad (2.5.2)$$

и точка  $z$ , полученная некоторым методом локального поиска. Тогда для поверхности уровня

$$Y(\beta) = \{y \in \mathbb{R}^n \mid h(y) = \beta\}$$

выпуклой функции  $h(\cdot)$  построим некоторую конечную аппроксимацию

$$\mathcal{A}(\beta) = \{y^1, \dots, y^N \mid h(y^i) = \beta, g(y^i) \leq \beta, i = 1, \dots, N\}.$$

б) Теперь для каждого  $y^i \in \mathcal{A}(\beta)$  решим линеаризованную задачу:

$$(\mathcal{P}\mathcal{L}_i) : \quad g(x) - \langle \nabla h(y^i), x \rangle \downarrow \min_x, \quad x \in S, f(x) \leq f(z). \quad (2.5.3)$$

Пусть  $u^i$  — (приближенное) глобальное решение задачи (2.5.3).

в) Затем для каждого  $i \in \{1, \dots, N\}$  найдем (приближенное) глобальное решение  $w^i$  задачи уровня:

$$\langle \nabla h(v), u^i - v \rangle \uparrow \max_v, \quad h(v) = \beta. \quad (2.5.4)$$

д) Далее формируем величину  $\eta(\beta) := \beta + \hat{\eta}(\beta)$ , где

$$\begin{aligned} \hat{\eta}(\beta) &:= \langle \nabla h(w^j), u^j - w^j \rangle - g(u^j) \triangleq \\ &\triangleq \max_{1 \leq i \leq N} \{ \langle \nabla h(w^i), u^i - w^i \rangle - g(u^i) \}. \end{aligned} \quad (2.5.5)$$

Если  $\eta(\beta) > 0$ , то в силу выпуклости  $h(\cdot)$  и равенства  $h(w^i) = \beta$ , имеем

$$\begin{aligned} 0 &< \beta + \langle \nabla h(w^j), w^j - w^j \rangle - g(u^j) \leq \\ &\leq h(u^j) - h(w^j) + \beta - g(u^j) = \\ &= -F(u^j) - g(w^j) + \beta = -F(u^j). \end{aligned} \tag{2.5.6}$$

е) Итак, получена точка  $u^j \in S$ ,  $F(u^j) < 0$ , которая по целевой функции не хуже, чем  $z$  ( $f(u^j) \leq f(z)$ ), и в которой ограничение типа равенства не выполнено. В этом случае можно предложить процедуру, которая из точки  $u^j$  строит (например, методом локального поиска) другую допустимую (критическую) точку  $\hat{z} \in S$ , со свойством  $F(\hat{z}) = 0$ . При этом значение целевой функции  $f(\hat{z})$  будет лучше, чем в точке  $u^j$ :  $f(\hat{z}) < f(u^j) \leq f(z)$ .

Иногда в вычислительном плане эффективнее оказывается применение вместо шагов с) и д) прямого сравнения значения целевой функции в текущей точке  $z$  со значениями в точках, полученных локальным поиском, стартовавшим из точек  $u^i$ ,  $i = 1 \dots, N$ . В частности, это обосновано в случае, когда получение решения задачи уровня на шаге с) оказывается весьма затратным по времени.

Итак, предложено разбить задачу (2.5.1) на несколько этапов. При этом выбор методов решения задач (2.5.3) и (2.5.4), а также метода одномерной оптимизации весьма важен для общего успеха глобального поиска. В то же время, как показывает вычислительный опыт [69], задача а), или построение "хорошей" аппроксимации на шаге 3 стратегии глобального поиска (СГП), является решающим элементом для выхода из текущей стационарной точки при условии квалифицированного исполнения остальных этапов глобального поиска. Кроме того, эффективность СГП напрямую зависит от метода локального поиска на шаге е).

Следующий параграф посвящен вопросу применимости стратегии глобального поиска к исследуемой задаче (P)–(2.2.5).

## 2.6 Решение возмущенной задачи

Вернемся от общей постановки задачи с равенством (P<sub>0</sub>)–(2.3.1) к задаче математического программирования, к которой сводится рассматриваемая в главе линейная двух-

уровневая задача:

$$(\mathcal{P}) \quad \left. \begin{aligned} f(x, y) &= \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y, v} \\ (x, y, v) &\in S, \\ F(x, y, v) &= g(x, y, v) - h(x, v) = 0, \end{aligned} \right\} \quad (2.6.1)$$

где

$$S := \{Ax \leq b, \quad A_1x + B_1y \leq b^1, \quad vB_1 = -d^1, \quad v \geq 0\}, \quad (2.6.2)$$

$$g(x, y, v) = \frac{1}{4} \|A_1x - v\|^2 + \langle d^1, y \rangle + \langle b^1, v \rangle, \quad h(x, v) = \frac{1}{4} \|A_1x + v\|^2.$$

Напомним, что в силу неравенства (2.2.3)

$$F(x, y, v) \geq 0 \quad \forall (x, y, v) \in S. \quad (2.6.3)$$

Запишем условие регулярности  $(\mathcal{H})$ –(2.3.4), обеспечивающее достаточность условий глобальной оптимальности, в терминах задачи  $(\mathcal{P})$ –(2.6.1):

$$\left. \begin{aligned} \forall (x, y, v) \in S : F(x, y, v) = 0 \quad (g(x, y, v) = h(x, v)) \\ \exists (x_p, y_p, v_p) \in S : g(x_p, y_p, v_p) - g(x, y, v) < \langle \nabla h(x, v), (x_p, v_p) - (x, v) \rangle \end{aligned} \right\}. \quad (2.6.4)$$

Оказывается, условие (2.6.4) в задаче  $(\mathcal{P})$ –(2.6.1) не выполнено. Действительно, из (2.6.4), в силу выпуклости функции  $h(\cdot)$ , получаем

$$g(x_p, y_p, v_p) - g(x, y, v) < \langle \nabla h(x, v), (x_p, v_p) - (x, v) \rangle \leq h(x_p, v_p) - h(x, v),$$

откуда

$$F(x_p, y_p, v_p) = g(x_p, y_p, v_p) - h(x_p, v_p) < g(x, y, v) - h(x, v) = 0,$$

что невозможно с учетом (2.6.3).

С другой стороны, можно считать без ограничения общности, что условие  $(\mathcal{H}_1)$ –(2.3.6) в задаче выполнено:

$$\exists (\tilde{x}, \tilde{y}, \tilde{v}) \in S : F(\tilde{x}, \tilde{y}, \tilde{v}) > 0, \quad f(\tilde{x}, \tilde{y}) < \mathcal{V}(\mathcal{P}). \quad (2.6.5)$$

Действительно, в качестве точки  $(\tilde{x}, \tilde{y}, \tilde{v})$  может быть взято решение релаксированной задачи без ограничения-равенства. Если в этом случае окажется, что  $F(\tilde{x}, \tilde{y}, \tilde{v}) = 0$  или  $f(\tilde{x}, \tilde{y}) = \mathcal{V}(\mathcal{P})$ , то задача с равенством не представляет интереса для дальнейшего рассмотрения, поскольку тогда ограничение-равенство не влияет на решение задачи.

Для того, чтобы обеспечить выполнение условия (2.6.4), далее будем рассматривать следующую возмущенную задачу:

$$(\mathcal{P}_\rho) \quad \left. \begin{aligned} f(x, y) &= \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y, v}, \\ (x, y, v) &\in S, \\ F(x, y, v) - \rho &= 0, \end{aligned} \right\} \quad (2.6.6)$$

где  $\rho > 0$ . При этом, чтобы в задаче  $(\mathcal{P}_\rho)$  условие  $(\mathcal{H}_1)$ –(2.3.6) также было выполнено, параметр возмущения  $\rho$  должен удовлетворять условию

$$\rho < F(\check{x}, \check{y}, \check{v}). \quad (2.6.7)$$

Напомним, что функция  $F(\cdot)$  представляет собой невязку двойственности в задаче нижнего уровня исходной задачи  $(\mathcal{BP})$ –(2.1.1). Поэтому одновременно рассмотрим возмущенную двухуровневую задачу, в которой задача нижнего уровня решается приближенно с точностью  $\rho$ :

$$(\mathcal{BP}_\rho) \quad \left. \begin{aligned} f(x, y) &\stackrel{\Delta}{=} \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y}, \\ x \in X &\stackrel{\Delta}{=} \{x \in \mathbb{R}^m \mid Ax \leq b\}, \\ y \in Y_*(x, \rho) &\stackrel{\Delta}{=} \{y \in Y(x) \mid \langle d^1, y \rangle \leq \inf_z [\langle d^1, z \rangle \mid z \in Y(x)] + \rho\}, \end{aligned} \right\} \quad (2.6.8)$$

где  $Y(x) \stackrel{\Delta}{=} \{y \in \mathbb{R}^n \mid A_1x + B_1y \leq b^1\}$ . Между задачами  $(\mathcal{P}_\rho)$ –(2.6.6) и  $(\mathcal{BP}_\rho)$ –(2.6.8) установлена следующая взаимосвязь.

**Теорема 2.6.1** Пусть тройка  $(x^*, y^*, v^*)$  – решение задачи  $(\mathcal{P}_\rho)$ –(2.6.6). Тогда пара  $(x^*, y^*)$  является решением двухуровневой задачи  $(\mathcal{BP}_\rho)$ –(2.6.8).

**Доказательство.** Пусть  $(x^*, y^*, v^*) \in \text{Sol}(\mathcal{P}_\rho)$ . Тогда для  $(x^*, y^*, v^*)$  выполнены следующие условия:

$$\left. \begin{aligned} Ax^* &\leq b, \\ A_1x^* + B_1y^* &\leq b^1, \\ v^*B_1 &= -d^1, \quad v^* \geq 0, \\ \rho + \langle A_1x^* - b^1, v^* \rangle &= \langle d^1, y^* \rangle. \end{aligned} \right\} \quad (2.6.9)$$

Рассмотрим прямую и двойственные задачи нижнего уровня для  $(\mathcal{BP}_\rho)$ –(2.6.8) при фиксированном  $x^*$ :

$$(\mathcal{F}(x^*)) \quad \langle d^1, y \rangle \downarrow \min_y, \quad y \in Y(x^*) \stackrel{\Delta}{=} \{y \in \mathbb{R}^n \mid A_1x^* + B_1y \leq b^1\}, \quad (2.6.10)$$

$$(\mathcal{DF}(x^*)) \langle A_1 x^* - b^1, v \rangle \uparrow \max_v, \quad v \in V \triangleq \{v \in \mathbb{R}^q \mid vB_1 = -d^1, v \geq 0\}. \quad (2.6.11)$$

В силу теоремы двойственности [10],

$$\langle A_1 x^* - b^1, v \rangle \leq \langle d^1, y \rangle \quad \forall y \in Y(x^*), \quad \forall v \in V. \quad (2.6.12)$$

Тогда из (2.6.9) и (2.6.12) следует

$$\langle d^1, y^* \rangle = \langle A_1 x^* - b^1, v^* \rangle + \rho \leq \langle d^1, y \rangle + \rho \quad \forall y \in Y(x^*),$$

откуда получаем, что  $y^* \in \rho\text{-Sol}(\mathcal{F}(x^*))$ . Следовательно, точка  $(x^*, y^*)$  — допустимая в задаче  $(\mathcal{BP}_\rho)$ —(2.6.8).

Покажем теперь, что  $(x^*, y^*)$  — решение задачи  $(\mathcal{BP}_\rho)$ —(2.6.8). Пусть это не так и существует допустимая в задаче  $(\mathcal{BP}_\rho)$ —(2.6.8) точка  $(\hat{x}, \hat{y})$ , в которой значение целевой функции верхнего уровня меньше:

$$f(\hat{x}, \hat{y}) < f(x^*, y^*). \quad (2.6.13)$$

Тогда

$$\langle d^1, \hat{y} \rangle \leq \langle d^1, y \rangle + \rho \quad \forall y \in Y(\hat{x}). \quad (2.6.14)$$

Пусть  $\hat{v} \in \text{Sol}(\mathcal{DF}(\hat{x}))$ ,  $\bar{y} \in \text{Sol}(\mathcal{F}(\hat{x}))$ . В этом случае

$$\langle d^1, \hat{y} \rangle \leq \langle d^1, \bar{y} \rangle + \rho = \langle A_1 \hat{x} - b^1, \hat{v} \rangle + \rho,$$

откуда заключаем, что

$$F(\hat{x}, \hat{y}, \hat{v}) - \rho \leq 0, \quad (\hat{x}, \hat{y}, \hat{v}) \in S.$$

Далее возможны два случая.

а) Если последнее неравенство выполняется как равенство, то сразу приходим к противоречию, поскольку получаем допустимую в задаче  $(\mathcal{P}_\rho)$ —(2.6.6) точку, в которой значение целевой функции меньше, чем  $f(x^*, y^*) = \mathcal{V}(\mathcal{P}_\rho)$ .

б) Пусть теперь  $F(\hat{x}, \hat{y}, \hat{v}) - \rho < 0$ . При достаточно малом  $\rho$ , удовлетворяющем (2.6.7), из (2.6.5) следует, что

$$\exists(\check{x}, \check{y}, \check{v}) \in S : F(\check{x}, \check{y}, \check{v}) - \rho > 0, \quad f(\check{x}, \check{y}) < \mathcal{V}(\mathcal{P}_\rho). \quad (2.6.15)$$

Тогда, в силу выпуклости  $S$ , найдется такая выпуклая комбинация  $(x_\lambda, y_\lambda, v_\lambda) \in S$ ,  $\lambda \in ]0, 1[$ , точек  $(\check{x}, \check{y}, \check{v})$  и  $(\hat{x}, \hat{y}, \hat{v})$ , что  $F(x_\lambda, y_\lambda, v_\lambda) - \rho = 0$  и при этом, в силу линейности функции  $f(\cdot)$ , а также (2.6.13) и (2.6.15),

$$f(x_\lambda, y_\lambda) = \lambda f(\check{x}, \check{y}) + (1 - \lambda)f(\hat{x}, \hat{y}) < f(\hat{x}, \hat{y}) < f(x^*, y^*).$$



Приходим к противоречию с тем, что  $(x^*, y^*, v^*) \in \text{Sol}(\mathcal{P}_\rho)$ . #

Таким образом, для отыскания оптимистического решения двухуровневой задачи  $(\mathcal{BP})$ –(2.1.1) предлагается вместо задачи  $(\mathcal{P})$ –(2.6.1) решать возмущенную задачу  $(\mathcal{P}_\rho)$ –(2.6.6), в которой выполнены условия регулярности. При этом параметр  $\rho$  должен выбираться достаточно малым, чтобы выполнялись условия (2.6.15) и точность решения задачи нижнего уровня была приемлемой для рассматриваемой задачи  $(\mathcal{BP})$ –(2.1.1). Тогда, применяя стратегию глобального поиска к возмущенной задаче  $(\mathcal{P}_\rho)$ –(2.6.6), получаем решение линейной двухуровневой задачи, в которой допускается приближенное (с точностью  $\rho$ ) решение задачи нижнего уровня.

## 2.7 Локальный поиск в задаче с д.с. ограничением-равенством

Одним из базовых элементов стратегии глобального поиска, основанной на условиях глобальной оптимальности, является поиск критических точек или локальных решений в невыпуклой задаче. При этом понятие критической точки зависит от структуры конкретной задачи и соответствующего метода локального поиска.

Первый метод, который можно предложить, принимая во внимание представление задачи  $(\mathcal{P}_\rho)$ –(2.6.6) и ее связь с задачей  $(DCC)$ –(2.3.2), — специальный метод локального поиска, предложенный в [77]. Он основан на последовательном приближенном решении линеаризованных задач, в данном случае имеющих следующий вид:

$$(\mathcal{LQ}((x^s, v^s), \gamma^s)) \quad \left. \begin{array}{l} g(x, y, v) - \langle \nabla h(x^s, v^s), (x, v) \rangle \downarrow \min_{x, y, v}, \\ (x, y, v) \in S, \quad f(x, y) \leq \gamma^s \triangleq f(x^s, y^s). \end{array} \right\} \quad (2.7.1)$$

Можно заметить, что это выпуклая квадратичная задача, и ее решение может быть осуществлено любым из известных методов квадратичного программирования.

Однако специфика задачи  $(\mathcal{P}_\rho)$ –(2.6.6), ее билинейность, в этом случае не используется, а как уже отмечалось, основную сложность при решении задачи  $(\mathcal{P}_\rho)$ –(2.6.6) составляет наличие билинейного ограничения-равенства

$$F(x, y, v) = \langle d_1, y \rangle - \langle A_1 x - b, v \rangle - \rho = 0.$$

В [43], а позже в [44, 78] была предложена идея метода локального поиска для задачи

вида

$$F(x, y) \downarrow \min, \quad x \in X, \quad y \in Y, \quad (2.7.2)$$

где  $F(x, y)$  — билинейная функция, заключающаяся в решении задач линейного программирования по разделенным переменным:

$$F(x, \hat{y}) \downarrow \min_x, \quad x \in X, \quad (LP_x(\hat{y}))$$

$$F(\hat{x}, y) \downarrow \min_y, \quad y \in Y. \quad (LP_y(\hat{x}))$$

Метод, основанный на последовательном решении данных задач, позволяет отыскивать точку, удовлетворяющую следующим условиям:

$$F(x^*, y^*) \leq F(x, y^*) \quad \forall x \in X, \quad (2.7.3)$$

$$F(x^*, y^*) \leq F(x^*, y) \quad \forall y \in Y. \quad (2.7.4)$$

**Определение 2.7.1** Точка, удовлетворяющая неравенствам (2.7.3) и (2.7.4) называется *частичным глобальным решением* или *критической точкой* задачи (2.7.2).

Как было показано в [44, 78], критической точки с такими свойствами достаточно для эффективной работы алгоритма глобального поиска в задаче минимизации билинейной функции.

Покажем, каким образом может быть найдена критическая точка в смысле определения 2.7.1 в рассматриваемой задаче  $(\mathcal{P}_\rho)$ –(2.6.6) с учетом ее билинейности и двухуровневой природы. В задаче  $(\mathcal{P}_\rho)$ –(2.6.6) переменные можно разбить на две группы:  $(x, y) \in Z \triangleq \{(x, y) \mid Ax \leq b, A_1x + B_1y \leq b^1\}$  и  $v \in V \triangleq \{v \mid vB_1 = -d^1, v \geq 0\}$ .

### Процедура поиска критической точки

Прежде всего, необходимо найти вспомогательную точку

$$\bar{x} \in Pr_X(Z) \triangleq \{x \in X \mid \exists y \in Y(x) : (x, y) \in Z\}.$$

Такая точка может быть найдена, например, с помощью решения следующей задачи линейного программирования:

$$\langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x, y}, \quad (x, y) \in Z, \quad (2.7.5)$$

значение которой является нижней оценкой для значения двухуровневой задачи  $(\mathcal{BP})$ –(2.1.1). Отметим, что в силу предположения **(Н2)** задача (2.7.5) имеет решение. Пусть  $(\bar{x}, \bar{y})$  — решение задачи (2.7.5).

1. При фиксированном  $x := \bar{x} \in Pr_X(Z)$  решим задачу

$$\left. \begin{aligned} \langle d, y \rangle \downarrow \min_{y,v}, \\ A_1 \bar{x} + B_1 y \leq b^1, \\ v B_1 = -d^1, \quad v \geq 0, \\ \langle d_1, y \rangle = \langle A_1 \bar{x} - b_1, v \rangle + \rho. \end{aligned} \right\} \quad (2.7.6)$$

Пусть  $(\tilde{y}, v^*)$  — решение данной задачи.

2. При фиксированном  $v := v^*$  решим задачу

$$\left. \begin{aligned} \langle c, x \rangle + \langle d, y \rangle \downarrow \min_{x,y}, \\ Ax \leq b, \quad A_1 x + B_1 y \leq b^1, \\ \langle d_1, y \rangle = \langle A_1 x - b_1, v^* \rangle + \rho. \end{aligned} \right\} \quad (2.7.7)$$

Пусть  $(x^*, y^*)$  — решение задачи (2.7.7), тогда, очевидно,  $(x^*, y^*, v^*)$  — допустимая в задаче  $(\mathcal{P}_\rho)$ –(2.6.6) точка. #

Нетрудно видеть, что

$$f(x^*, y^*) \leq f(x, y) \quad \forall (x, y) \in Z : F(x, y, v^*) = 0, \quad (2.7.8)$$

так что точка  $(x^*, y^*, v^*)$  является частичным минимумом функции  $f$  по паре  $(x, y)$  при фиксированном  $v^*$ . Заметим, что от переменной  $v$  целевая функция не зависит, а следовательно, нет необходимости выполнения второго неравенства в определении критической точки.

**Замечание 2.7.1** *Покажем, что задача (2.7.6) имеет решение. Действительно, поскольку  $(\bar{x}, \bar{y})$  — решение релаксированной задачи, то в силу (2.6.7), найдется  $\bar{v}$ :*

$$F(\bar{x}, \bar{y}, \bar{v}) - \rho > 0 \quad (2.7.9)$$

С другой стороны, вектору  $\bar{x}$  соответствуют решение прямой задачи  $\hat{y} \in \text{Sol}(\mathcal{F}(\bar{x}))$  и решение двойственной задачи  $\hat{v} \in \text{Sol}(\mathcal{DF}(\bar{x}))$ , так что функция  $F(\cdot)$ , как невязка двойственности, обращается в нуль:  $F(\bar{x}, \hat{y}, \hat{v}) = 0$ , или

$$F(\bar{x}, \hat{y}, \hat{v}) - \rho < 0. \quad (2.7.10)$$

Из (2.7.9) и (2.7.10) следует, что найдутся такие  $y, v$ , что  $F(\bar{x}, y, v) - \rho = 0$ . Следовательно, допустимое множество задачи (2.7.6) не пусто.

Очевидно, что и допустимое множество задачи (2.7.7) также не пусто.

**Замечание 2.7.2** Полученная точка  $(x^*, y^*, v^*)$  является допустимой в задаче  $(\mathcal{P}_\rho)$ –(2.6.6), при этом пара  $(x^*, y^*)$  оказывается допустимой в двухуровневой задаче  $(\mathcal{BP}_\rho)$ –(2.6.8). Действительно,  $(x^*, y^*) \in Z$ , а из выполнения ограничения

$$\langle d_1, y^* \rangle = \langle A_1 x^* - b, v^* \rangle + \rho$$

следует, что  $y^*$  является  $\rho$ -приближенным решением задачи нижнего уровня  $(\mathcal{F}(x^*))$ .

## 2.7.1 Генерация тестовых задач

Для проверки работоспособности и эффективности новых методов решения задач оптимизации почти всегда возникает проблема отыскания или генерации тестовых задач с известными решениями и свойствами. В некоторых случаях можно воспользоваться существующими библиотеками тестовых примеров (такими как, например, DIMACS для задач дискретной оптимизации [19]), в некоторых — исследуемая задача имеет решение при любых исходных данных или, скажем, известно значение целевой функции в глобальном решении [78]. В задачах вида  $(\mathcal{BP})$ –(2.1.1) нельзя гарантировать существование решения при произвольных исходных данных. Более того, для такого сорта задач не удалось отыскать достаточно репрезентативных тестовых библиотек. Поэтому был применен оригинальный метод генерации тестовых задач вида  $(\mathcal{BP})$ –(2.1.1) с известными локальными и глобальными решениями, предложенный в [92].

Схема генерации тестовых задач из [92] состоит в следующем. Сначала вводятся так называемые задачи-ядра небольшой размерности. Затем из них конструируется задача специального вида. К последней применяется специальное преобразование, и в результате получается случайная задача общего вида с известными локальными и глобальными решениями. Опишем эту схему более подробно.

Определим задачи-ядра следующего вида:

$$\begin{cases} f_k^u(x_k, y_k) = 3 - x_k + y_k \downarrow \min_{x_k, y_k}, \\ 1 \leq x_k \leq 3, \quad y_k \in Y_*(x_k) = \text{Sol}(2.7.12), \end{cases} \quad (2.7.11)$$

$$\begin{cases} f_k^l(y_k) = -y_k \downarrow \min_{y_k}, \\ y_k \geq 0, \quad -2x_k + Y_k \leq 0, \quad x_k + y_k \leq t_k, \end{cases} \quad (2.7.12)$$

где  $3 \leq t_k \leq 9$ .

Можно заметить, что задачи-ядра различаются между собой параметром  $t_k$  в ограничении нижнего уровня. В зависимости от выбора  $t_k$  выделяются 5 классов задач с разным количеством локальных и глобальных решений.

В табл. 2.1 представлены свойства двухуровневой задачи в зависимости от выбора параметра  $t_k$ , где  $class$  — номер класса,  $t_k$  — значение параметра (или интервал возможных значений), определяющего класс,  $Extr$  — точка локального или глобального минимума  $(x_k, y_k)$ ,  $f_k^u$  — значение целевой функции задачи в соответствующей точке (жирным шрифтом выделено глобальное решение).

$class$	$t_k$	$Extr$	$f_k^u$
1	3	(3,0)	<b>0</b>
2	7	(1,2)	<b>4</b>
		(3,4)	<b>4</b>
3	9	(1,2)	<b>4</b>
4	]3;7[	$(3, t_k - 3)$	<b>2</b>
		(1,2)	4
5	]7;9[	(1,2)	<b>4</b>
		$(3, t_k - 3)$	5

Таблица 2.1: Свойства задач-ядер

Следующий этап генерации — это процесс объединения некоторого числа задач-ядер в одну задачу.

Пусть  $m$  и  $n$  — количество переменных на верхнем и нижнем уровнях строящейся двухуровневой задачи.

Двухуровневые задачи с одинаковым (или близким к этому) количеством переменных на верхнем и нижнем уровне считаются специалистами [90, 103] наиболее сложными для численного решения, поэтому представим схему генерации задач, в которых  $m = n$ .

Введем обозначения:  $x \triangleq (x_1, \dots, x_m)$ ,  $y \triangleq (y_1, \dots, y_m)$ ,  $t \triangleq (t_1, \dots, t_m)$ . Рассмотрим следующую двухуровневую задачу:

$$\begin{cases} f^u(x, y) \triangleq \sum_{k=1}^m (3 - x_k + y_k) \downarrow \min_{x, y}, \\ 1 \leq x_k \leq 3, \quad k = 1, \dots, m, \\ y \in Y_*(x) = Sol(2.7.14), \end{cases} \quad (2.7.13)$$

$$\begin{cases} f^l(x, y) = - \sum_{k=1}^m y^k \downarrow \min_y, \\ -2x_k + y_k \leq 0, \quad x_k + y_k \leq t_k, \quad y_k \geq 0, \quad k = 1, \dots, m, \end{cases} \quad (2.7.14)$$

где  $3 \leq t_k \leq 9$ .

Далее определим множества  $M_i$ ,  $i = 1, \dots, 5$ , следующим образом:

$$M_i = \{k \in \{1, \dots, m\} : t_k \in class\ i\}, \quad i = 1, \dots, 5.$$

Кроме того, пусть  $m_i$  — мощность множества  $M_i$ ,  $i = 1, \dots, 5$ . Тогда задача (2.7.13)–(2.7.14) будет иметь следующие свойства.

**Предложение 2.7.1** [92]. *Задача (2.7.13)–(2.7.14) имеет  $2^{m_2}$  глобальных минимумов в точках*

$$(x_k^G, y_k^G) = \begin{cases} (3, 0), k \in M_1, \\ (1, 2) \text{ или } (3, 4), k \in M_2, \\ (1, 2), k \in M_3 \cup M_5, \\ (3, t_k - 3), k \in M_4. \end{cases}$$

При этом  $f^u(x^G, y^G) = 4(m_2 + m_3 + m_5) + \sum_{k \in M_4} (t_k - 3)$ .

**Предложение 2.7.2** [92]. *Задача (2.7.13)–(2.7.14) имеет локальные решения, не являющиеся глобальными только в случае  $m_4 + m_5 > 0$ . Их количество равно  $2^{m_2+m_4+m_5} - 2^{m_2}$ , а хотя бы одна компонента  $(x_k, y_k)$  имеет вид*

$$(x_k, y_k) = \begin{cases} (1, 2), k \in M_4, \\ (3, t_k - 3), k \in M_5. \end{cases}$$

Все остальные компоненты определяются так же, как в предложении (2.7.1).

Сформулированные предложения говорят о том, что в полученной двухуровневой задаче (2.7.13)–(2.7.14) довольно легко вычислить глобальное (локальное) решение. Для этого достаточно объединить известные глобальные (локальные) решения всех задач-ядер в один вектор.

Последний этап генерации — случайное преобразование задачи. Сначала запишем задачу (2.7.13)–(2.7.14) в другой форме (с перенесением ограничений верхнего уровня по  $x$  на нижний.) Это возможно сделать благодаря тому, что в ограничениях верхнего уровня присутствует только переменная  $x$  [29, 90].

$$\begin{cases} f^u(x, y) = \langle c, x \rangle + \langle d, y \rangle + \kappa \downarrow \min_{x, y}, \\ y \in \text{Sol}(2.7.16) \end{cases} \quad (2.7.15)$$

$$\begin{cases} f^l(x, y) = \langle c_1, x \rangle + \langle d_1, y \rangle + \kappa \downarrow \min_y, \\ Ax + By \leq b. \end{cases} \quad (2.7.16)$$

Будем обозначать эту задачу  $\mathcal{B}\mathcal{L}\mathcal{P}(a, a_1, C, b)$ , где

$$a = \begin{bmatrix} c \\ d \end{bmatrix}, a_1 = \begin{bmatrix} c_1 \\ d_1 \end{bmatrix}, C = \begin{bmatrix} A & B \end{bmatrix}.$$

При этом  $c = -1_m$ ,  $d = 1_m$ ,  $c_1 = 0_m$ ,  $(d_1) = -1_m$ ,  $\kappa = 3m$ ,

$$A = \begin{bmatrix} P_x \\ -2P_x \\ \frac{1}{3}P_x \\ -P_x \\ 0_{m \times m} \end{bmatrix}, \quad B = \begin{bmatrix} P_y \\ P_y \\ 0_{m \times m} \\ 0_{m \times m} \\ -P_y \end{bmatrix}, \quad b = \begin{bmatrix} t \\ 0_m \\ 1_m \\ -1_m \\ 0_m \end{bmatrix},$$

где

$$P_x \in \mathbb{R}^{m \times m}, P_y \in \mathbb{R}^{m \times m} \text{ — матрицы с элементами } P_{ij} = \begin{cases} 1, & 1 \leq i = j \leq m \\ 0, & i \neq j \end{cases}.$$

Можно показать [92], что задача  $\mathcal{BLCP}(a, a_1, C, b)$  эквивалентна задаче (2.7.13)–(2.7.14) [90]. Теперь выберем произвольный ненулевой вектор  $h \in \mathbb{R}^N$  и определим для него матрицу  $M = HDH$  размерности  $N \times N$ , где  $N = m + n$ ,  $D$  — положительно определенная диагональная матрица,  $H$  — матрица Хаусхолдера [35, 64]:

$$H = E - \frac{2}{\langle h, h \rangle} hh^T,$$

$E$  — единичная матрица. Заметим, что выражение для обратной к  $M$  матрицы с учетом ортогональности матрицы Хаусхолдера можно записать следующим образом:  $M^{-1} = HD^{-1}H$ . Известно, что обратная матрица к диагональной  $D \triangleq \text{diag}(d_1, \dots, d_k)$  вычисляется по формуле  $D^{-1} \triangleq \text{diag}(\frac{1}{d_1}, \dots, \frac{1}{d_k})$ . При этом справедлива следующая теорема.

**Теорема 2.7.1** [92] *Задача  $\mathcal{BLCP}(a, a_1, C, b)$  с переменными  $x$  и  $y$  эквивалентна задаче  $\mathcal{BLCP}(Ma, Ma_1, CM, b)$  с переменными  $\bar{x}, \bar{y}$  при невырожденном преобразовании*

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = M^{-1} \begin{bmatrix} x \\ y \end{bmatrix}. \quad \#$$

Итак, в результате получается двухуровневая задача с известными локальными и глобальными решениями, которые вычисляются с использованием данных о задачах-ядрах. Отметим простоту и удобство этого метода. Во-первых, имеется возможность легко изменять размерность и сложность задачи, а, во-вторых, в методе нет никаких сложных и громоздких вычислений.

## 2.7.2 Тестирование метода локального поиска

Данный параграф посвящен тестированию метода локального поиска на двухуровневых задачах, построенных по описанной выше методике. Вначале проведено тестирование на задачах небольшой (2–5) размерности. Выделены классы задач, на которых не удалось

найти глобальное решение. Далее произведено тестирование метода локального поиска на задачах средней (10–100) размерности для наиболее сложных классов задач.

Заметим, что предлагаемый алгоритм локального поиска работает не с задачей  $(\mathcal{BP})$ –(2.1.1), а с  $(\mathcal{P})$ –(2.6.1). А как известно, говорить об эквивалентности задач  $(\mathcal{BP})$ –(2.1.1) и  $(\mathcal{P})$ –(2.6.1) можно только с точки зрения глобального решения, поскольку совпадение локальных решений в этих задачах, вообще говоря, не имеет места [90, 103]. Поэтому, если для задачи  $(\mathcal{BP})$ –(2.1.1) наиболее сложным случаем (с точки зрения количества локальных экстремумов, в соответствии с предложением 2.7.2) представляется наличие в ней задач-ядер 4 и 5 классов, то для задачи  $(\mathcal{P})$ –(2.6.1) ситуация может выглядеть иначе.

Все программы, реализующие алгоритмы, были написаны в системе Visual C++ 2005. Счет производился на компьютере с процессором Intel Core 2 Duo 2.0ГГц. Для решения вспомогательных задач линейного и квадратичного программирования использовался решатель Xpress-MP [140]. Параметр  $\rho$  выбирался равным  $10^{-5}$ .

В табл. 2.2 представлены результаты тестирования локального поиска на задачах размерности от 2 до 5.

Здесь и далее *class* — набор задач ядер, используемых при генерации (например, (2,0,1,0,0) обозначает, что в примере использованы 2 задачи-ядра первого класса, 1 задача-ядро — третьего, а задачи других классов не участвовали в генерации примера),  $m = n$  — размерность задачи,  $f_*$  — значение целевой функции в точке глобального минимума,  $f_{loc}$  — значение функции, полученное локальным поиском,  $T$  — время счета в секундах.

Задачи, в которых локальным поиском удалось найти глобальный минимум, выделены жирным шрифтом.

Из табл. 2.2 видно, что задачи, сгенерированные при помощи задач-ядер 5 класса, не были решены локальным поиском ни в одном случае. В то же время задачи с 4 классом в некоторых случаях удалось решить, несмотря на одинаковую с 5 классом сложность, с точки зрения локальных экстремумов. Также есть задачи, сгенерированные без использования 4 и 5 классов, в которых, тем не менее, не удалось найти глобальный минимум.

Поскольку по данной таблице судить о строгой зависимости сложности задачи от ее структуры затруднительно, с целью выявления наиболее сложных классов задач был проведен расширенный эксперимент на сериях задач.

Далее на рис. 2.2 в виде диаграммы представлены результаты тестирования алгоритма локального поиска на задачах размерности от 2 до 5 и обозначена зависимость доли



решенных задач в серии от их вида (например, X00X0 означает, что при построении задачи участвовали только задачи-ядра 1 и 4 классов).

<i>class</i>	<i>m = n</i>	<i>f*</i>	<i>f<sub>loc</sub></i>	<i>T</i>	<i>class</i>	<i>m = n</i>	<i>f*</i>	<i>f<sub>loc</sub></i>	<i>T</i>
(2, 0, 0, 0, 0)	2	<b>0</b>	<b>0</b>	0.03	(1, 1, 0, 1, 1)	4	10	11	0.00
(0, 2, 0, 0, 0)	2	<b>8</b>	<b>8</b>	0.00	(1, 1, 1, 0, 1)	4	12	13	0.00
(0, 0, 2, 0, 0)	2	8	10	0.00	(1, 1, 1, 1, 0)	4	<b>10</b>	<b>10</b>	0.01
(0, 0, 0, 2, 0)	2	<b>4</b>	<b>4</b>	0.00	(1, 2, 1, 0, 0)	4	12	14	0.00
(0, 0, 0, 0, 2)	2	8	10	0.00	(1, 1, 0, 2, 0)	4	<b>8</b>	<b>8</b>	0.00
(1, 1, 0, 0, 0)	2	<b>4</b>	<b>4</b>	0.01	(2, 1, 0, 0, 1)	4	8	9	0.00
(1, 0, 1, 0, 0)	2	<b>4</b>	<b>4</b>	0.00	(1, 0, 2, 1, 0)	4	<b>10</b>	<b>10</b>	0.00
(1, 0, 0, 1, 0)	2	<b>2</b>	<b>2</b>	0.00	(1, 0, 1, 0, 2)	4	12	14	0.00
(1, 0, 0, 0, 1)	2	4	5	0.00	(2, 0, 0, 1, 1)	4	6	7	0.01
(0, 1, 1, 0, 0)	2	<b>8</b>	<b>8</b>	0.00	(0, 1, 2, 0, 1)	4	16	17	0.00
(0, 1, 0, 1, 0)	2	<b>6</b>	<b>6</b>	0.00	(0, 1, 1, 2, 0)	4	<b>12</b>	<b>12</b>	0.00
(0, 1, 0, 0, 1)	2	8	9	0.00	(0, 2, 0, 1, 1)	4	14	15	0.00
(0, 0, 1, 1, 0)	2	6	8	0.00	(0, 0, 1, 2, 1)	4	12	13	0.00
(0, 0, 1, 0, 1)	2	8	11	0.00	(1, 3, 0, 0, 0)	4	<b>12</b>	<b>12</b>	0.01
(0, 0, 0, 1, 1)	2	6	7	0.00	(2, 0, 2, 0, 0)	4	8	10	0.00
(0, 0, 0, 2, 1)	3	8	9	0.00	(3, 0, 0, 1, 0)	4	<b>2</b>	<b>2</b>	0.00
(0, 0, 2, 1, 0)	3	10	14	0.00	(2, 0, 0, 0, 2)	4	8	10	0.00
(3, 0, 0, 0, 0)	3	<b>0</b>	<b>0</b>	0.01	(0, 1, 3, 0, 0)	4	16	18	0.00
(0, 3, 0, 0, 0)	3	<b>12</b>	<b>12</b>	0.00	(0, 3, 0, 1, 0)	4	<b>14</b>	<b>14</b>	0.01
(0, 0, 3, 0, 0)	3	12	18	0.00	(0, 2, 0, 0, 2)	4	16	18	0.00
(0, 0, 0, 3, 0)	3	<b>6</b>	<b>6</b>	0.00	(0, 0, 1, 3, 0)	4	<b>10</b>	<b>10</b>	0.00
(0, 0, 0, 0, 3)	3	12	15	0.00	(0, 0, 3, 0, 1)	4	16	17	0.00
(2, 1, 0, 0, 0)	3	<b>4</b>	<b>4</b>	0.00	(0, 0, 0, 2, 2)	4	12	14	0.00
(2, 0, 1, 0, 0)	3	4	6	0.00	(5, 0, 0, 0, 0)	5	<b>0</b>	<b>0</b>	0.00
(2, 0, 0, 1, 0)	3	<b>2</b>	<b>2</b>	0.00	(0, 5, 0, 0, 0)	5	<b>20</b>	<b>20</b>	0.00
(2, 0, 0, 0, 1)	3	4	5	0.00	(0, 0, 5, 0, 0)	5	20	26	0.01
(0, 2, 1, 0, 0)	3	12	14	0.00	(0, 0, 0, 5, 0)	5	<b>10</b>	<b>10</b>	0.00
(0, 2, 0, 1, 0)	3	<b>10</b>	<b>10</b>	0.00	(0, 0, 0, 0, 5)	5	20	25	0.00
(0, 2, 0, 0, 1)	3	12	13	0.00	(0, 2, 1, 1, 1)	5	18	21	0.00
(0, 0, 2, 0, 1)	3	12	17	0.00	(1, 0, 2, 1, 1)	5	14	19	0.00
(1, 2, 0, 0, 0)	3	<b>8</b>	<b>8</b>	0.00	(1, 1, 0, 2, 1)	5	12	13	0.00
(1, 0, 2, 0, 0)	3	8	12	0.00	(1, 1, 1, 0, 2)	5	16	20	0.00
(1, 0, 0, 2, 0)	3	<b>4</b>	<b>4</b>	0.00	(2, 1, 1, 1, 0)	5	<b>10</b>	<b>10</b>	0.00
(1, 0, 0, 0, 2)	3	8	10	0.00	(2, 2, 1, 0, 0)	5	12	14	0.01
(0, 1, 2, 0, 0)	3	12	16	0.00	(2, 1, 0, 2, 0)	5	<b>8</b>	<b>8</b>	0.00
(0, 1, 0, 2, 0)	3	<b>8</b>	<b>8</b>	0.00	(1, 2, 0, 0, 2)	5	16	18	0.00
(0, 1, 0, 0, 2)	3	12	14	0.00	(1, 0, 1, 3, 0)	5	10	12	0.00
(0, 0, 1, 2, 0)	3	8	10	0.00	(3, 0, 1, 0, 1)	5	8	9	0.00
(0, 0, 1, 0, 2)	3	12	16	0.00	(1, 0, 0, 3, 1)	5	10	11	0.00
(0, 0, 0, 1, 2)	3	10	12	0.00	(0, 2, 2, 0, 1)	5	20	23	0.01
(1, 1, 1, 0, 0)	3	8	10	0.00	(0, 1, 2, 2, 0)	5	16	20	0.00
(1, 1, 0, 1, 0)	3	<b>6</b>	<b>6</b>	0.01	(0, 1, 0, 1, 3)	5	18	21	0.00
(1, 1, 0, 0, 1)	3	8	9	0.00	(0, 0, 3, 1, 1)	5	18	23	0.00
(0, 1, 1, 0, 1)	3	12	15	0.00	(4, 1, 0, 0, 0)	5	<b>4</b>	<b>4</b>	0.00
(0, 1, 1, 1, 0)	3	10	12	0.00	(1, 0, 4, 0, 0)	5	16	22	0.00
(0, 1, 0, 1, 1)	3	10	11	0.00	(2, 0, 0, 3, 0)	5	<b>6</b>	<b>6</b>	0.00
(0, 0, 1, 1, 1)	3	10	13	0.01	(3, 0, 0, 0, 2)	5	8	10	0.00
(0, 0, 0, 0, 4)	4	16	20	0.01	(0, 0, 4, 0, 1)	5	20	25	0.00
(0, 1, 1, 1, 1)	4	14	15	0.00	(0, 0, 0, 2, 3)	5	16	19	0.00

Таблица 2.2: Локальный поиск в задачах размерности от 2 до 5

Тестирование проводилось на сериях случайно сгенерированных задач, в которые входило от 10000 до 50000 задач каждого вида. На диаграмме указано, сколько процентов задач каждого вида удалось решить только локальным поиском. Двухуровневая задача здесь и далее считалась решенной, если значение целевой функции в полученной точке отличалось от известного глобального не более, чем на  $10^{-4}$ .

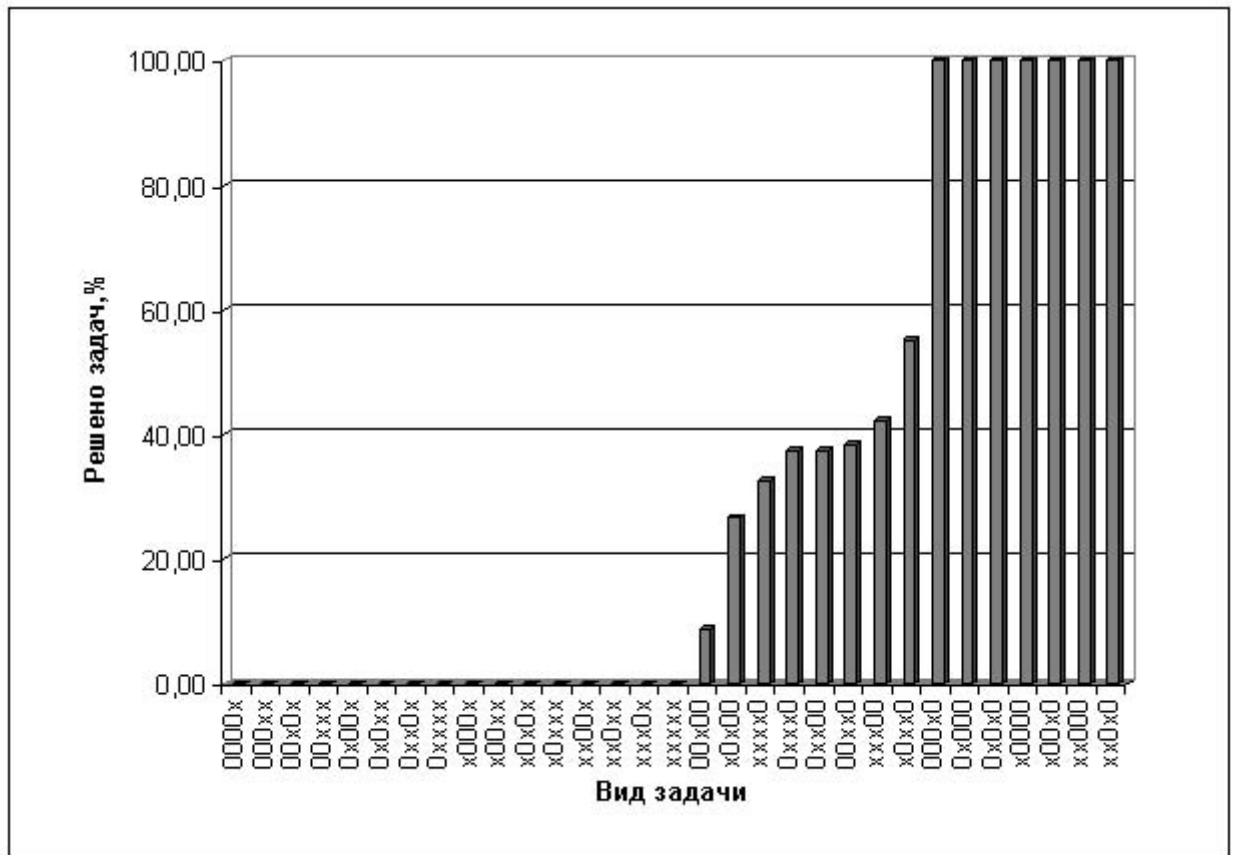


Рис. 2.2: Тестирование локального поиска

Уже на задачах малой размерности можно видеть, что для локального поиска наиболее сложными являются случаи, в которых присутствует 5-й класс задач-ядер: ни одна такая задача не была решена методом локального поиска (это коррелирует с представленными выше теоретическими результатами).

Однако вторым по сложности классом задач является не 4, а 3 класс: локальным поиском решено в лучшем случае около 60% рассмотренных задач, построенных с использованием задач-ядер 3 класса. Все остальные задачи (содержащие только ядра 1, 2 и 4 классов) были успешно решены алгоритмом локального поиска. Можно высказать гипотезу о том, что эта тенденция сохранится и при увеличении размерности задачи. Поэтому при тестировании алгоритма локального поиска на более высоких размерно-

стях, а в дальнейшем и алгоритма глобального поиска, будем использовать задачи, в обязательном порядке содержащие ядра 3 и/или 5 класса.

В табл. 2.3 представлены результаты тестирования алгоритма локального поиска на задачах средней размерности от 10 до 100.

В соответствии с результатами предыдущего этапа эксперимента алгоритму были предложены только задачи, содержащие задачи-ядра 3 и/или 5 класса. Во всех случаях была найдена допустимая (критическая) в задаче  $(\mathcal{P})$ –(2.6.1) точка менее, чем за 1 секунду.

При этом, как и предполагалось, ни одна из таких задач локальным поиском решена не была.

Разработке алгоритма глобального поиска и численному решению таких задач посвящены следующие параграфы.

<i>class</i>	<i>m = n</i>	<i>f*</i>	<i>f<sub>loc</sub></i>	<i>T</i>	<i>class</i>	<i>m = n</i>	<i>f*</i>	<i>f<sub>loc</sub></i>	<i>T</i>
( 0, 0, 10, 0, 0)	10	40	46	0.00	( 0, 0, 25, 25, 0)	50	150	176	0.09
( 0, 0, 0, 0, 10)	10	40	50	0.00	( 0, 0, 0, 25, 25)	50	150	175	0.11
( 2, 2, 2, 2, 2)	10	28	34	0.00	( 0, 0, 25, 0, 25)	50	200	251	0.09
( 4, 0, 2, 2, 2)	10	20	26	0.01	( 10, 10, 10, 10, 10)	50	140	156	0.11
( 2, 2, 0, 4, 2)	10	24	26	0.00	( 15, 15, 0, 20, 0)	50	100	100	0.11
( 2, 4, 2, 0, 2)	10	32	34	0.00	( 6, 4, 10, 14, 16)	50	148	178	0.09
( 2, 2, 2, 4, 0)	10	24	28	0.00	( 0, 0, 0, 0, 50)	50	200	250	0.11
( 2, 1, 0, 0, 7)	10	32	39	0.00	( 0, 0, 50, 0, 0)	50	200	250	0.09
( 0, 2, 7, 0, 1)	10	40	47	0.00	( 0, 0, 1, 0, 49)	50	200	251	0.11
( 0, 2, 0, 7, 1)	10	26	27	0.00	( 14, 6, 30, 0, 0)	50	144	170	0.11
( 0, 0, 10, 0, 10)	20	80	96	0.01	( 5, 0, 0, 30, 40)	75	220	260	0.27
( 0, 0, 10, 10, 0)	20	60	66	0.01	( 0, 5, 20, 30, 20)	75	240	284	0.23
( 4, 4, 4, 4, 4)	20	56	64	0.01	( 0, 0, 70, 0, 5)	75	300	369	0.25
( 0, 5, 5, 5, 5)	20	70	79	0.01	( 0, 0, 0, 70, 5)	75	160	165	0.24
( 0, 10, 10, 0, 0)	20	80	90	0.01	( 0, 0, 0, 0, 75)	75	300	375	0.27
( 0, 10, 0, 5, 5)	20	70	75	0.02	( 15, 15, 15, 15, 15)	75	210	237	0.25
( 0, 0, 15, 5, 0)	20	70	84	0.01	( 0, 0, 30, 45, 0)	75	210	236	0.27
( 0, 0, 15, 0, 5)	20	80	99	0.01	( 25, 25, 0, 25, 0)	75	150	150	0.25
( 0, 2, 4, 4, 10)	20	72	84	0.01	( 5, 10, 15, 20, 25)	75	240	283	0.25
( 2, 4, 4, 0, 10)	20	72	86	0.01	( 25, 20, 15, 10, 5)	75	180	201	0.27
( 0, 0, 20, 0, 20)	40	160	194	0.06	( 10, 0, 0, 80, 10)	100	200	210	0.52
( 0, 0, 0, 20, 20)	40	120	140	0.06	( 0, 10, 60, 0, 30)	100	400	488	0.50
( 0, 0, 20, 0, 20)	40	160	194	0.06	( 0, 0, 10, 70, 20)	100	260	288	0.53
( 10, 0, 10, 10, 10)	40	100	116	0.06	( 0, 0, 0, 90, 10)	100	220	230	0.52
( 8, 8, 8, 8, 8)	40	112	124	0.06	( 0, 0, 60, 30, 10)	100	340	418	0.48
( 0, 0, 30, 10, 0)	40	140	156	0.06	( 5, 25, 25, 25, 20)	100	330	376	0.55
( 15, 0, 15, 0, 10)	40	100	118	0.06	( 25, 25, 15, 10, 25)	100	280	315	0.53
( 0, 15, 10, 15, 0)	40	130	134	0.06	( 0, 0, 0, 50, 50)	100	300	350	0.50
( 10, 0, 20, 2, 8)	40	116	132	0.06	( 0, 0, 50, 0, 50)	100	400	504	0.52
( 5, 5, 5, 20, 5)	40	100	105	0.06	( 30, 20, 30, 0, 20)	100	280	332	0.50

Таблица 2.3: Локальный поиск в задачах размерности от 10 до 100

## 2.8 Глобальный поиск в линейной двухуровневой задаче

По результатам численного тестирования локального поиска можно сделать вывод о том, что предлагаемый метод, вообще говоря, не обеспечивает достижения глобального решения в двухуровневых задачах даже небольших размерностей. Поэтому следующим естественным этапом является разработка процедуры глобального поиска в поставленной задаче  $(\mathcal{P}_\rho)$ –(2.6.6).

Стратегия глобального поиска, основанная на условиях глобальной оптимальности для общей задачи с ограничением-равенством, приведена в разделе 2.5.

С использованием д.с. разложения (2.2.7) целевой функции  $F(\cdot)$ , в соответствии с условиями глобальной оптимальности в задаче  $(\mathcal{P}_\rho)$ –(2.6.6) [69, 72], процедура выхода из критической точки  $(x^k, y^k, v^k)$ , полученной алгоритмом локального поиска, состоит из следующих этапов.

1. Выбирается некоторое значение  $\beta$  из отрезка  $[\beta_-, \beta_+]$ , где  $\beta_- \triangleq \inf(g, S)$ ,  $\beta_+ \triangleq \sup(g, S)$ .

2. Строится аппроксимация поверхности уровня выпуклой функции  $h(u_i, w_i) = \beta - \xi_k$ , где  $\xi_k \triangleq F(x^k, y^k, v^k) - \rho$ .

3. Из каждой точки построенной аппроксимации решается линейаризованная задача и реализуется дополнительный локальный поиск, в результате чего формируются новые критические точки.

4. Осуществляется сравнение значений целевой функции в текущей критической точке со значениями во вновь полученных. Если в некоторой новой точке значение целевой функции лучше, она принимается к рассмотрению, и начинается новая итерация алгоритма. Если ни одна из точек аппроксимации не позволяет выйти из текущей критической точки, то строится новая аппроксимация для другого значения  $\beta$ . Далее в более алгоритмической форме представлена реализация стратегии глобального поиска в виде алгоритма с учетом особенностей задачи  $(\mathcal{P}_\rho)$ –(2.6.6).

Пусть заданы числа  $\beta_- \triangleq \inf(g, S)$ ,  $\beta_+ \triangleq \sup(g, S)$ ,  $M > 0$ ,  $N_A > 0$  — целые.

### Алгоритм глобального поиска (АГП)

**Шаг 0.** Положить  $k := 0$ . Выбрать  $x^0 \in Pr_X(Z)$ .

**Шаг 1.** Начиная с точки  $x^0$ , методом локального поиска найти критическую точку

$(x^k, y^k, v^k)$ . Положить  $\Delta\beta := (\beta_+ - \beta_-)/M$ .

**Шаг 2.** Положить  $\beta_k := g(x^k, y^k, v^k) - \rho$ ,  $\gamma_k := \langle c, x^k \rangle + \langle d, y^k \rangle$ ,  $i := 1$ .

**Шаг 3.** Построить  $i$ -ю точку аппроксимации поверхности уровня функции  $h(\cdot)$ :

$$\mathcal{A}_k = \{(u^i, w^i) \mid h(u^i, w^i) = \beta_k, i = 1, \dots, N_A\}.$$

**Шаг 4.** Найти  $(x^{ik}, y^{ik}, v^{ik})$  — решение линеаризованной задачи  $(\mathcal{LQ}((u^i, w^i), \gamma_k))$ —(2.7.1).

**Шаг 5.** Начиная с точки  $x^{ik}$ , методом локального поиска найти критическую точку  $(\bar{x}^i, \bar{y}^i, \bar{v}^i)$ .

**Шаг 6.** Если  $f(\bar{x}^i, \bar{y}^i) < \gamma_k$ , то положить  $x^{k+1} := \bar{x}^i, y^{k+1} := \bar{y}^i, v^{k+1} := \bar{v}^i, k := k + 1$  и идти на Шаг 2.

**Шаг 7.** Если  $f(\bar{x}^i, \bar{y}^i) \geq \gamma_k$  и  $i < N_A$ , то  $i := i + 1$  идти на Шаг 3.

**Шаг 8.** Если  $f(\bar{x}^i, \bar{y}^i) \geq \gamma_k$ ,  $\beta < \beta_+$  и  $i = N_A$ , то положить  $\beta_k := \beta_k + \Delta\beta$ ,  $i := 1$  и идти на Шаг 3.

**Шаг 9.** Если  $f(\bar{x}^i, \bar{y}^i) \geq \gamma_k$ ,  $i = N_A$  и  $\beta \geq \beta_+$ , то стоп:  $(x^k, y^k, v^k)$  — решение задачи  $(\mathcal{P}_\rho)$ —(2.6.6), найденное алгоритмом. #

Заметим, что в результате работы алгоритма локального поиска [18], благодаря его специфике, строится точка, в которой функция  $F - \rho$ , задающая д.с. ограничение, равна нулю. Поэтому на шаге 3 при построении аппроксимации полагается  $\xi_k = 0$ .

Кроме того, нетрудно видеть, что, поскольку  $(x^{ik}, y^{ik}, v^{ik})$  на шаге 4 является решением линеаризованной задачи  $(\mathcal{LQ}((u^i, w^i), \gamma_k))$ —(2.7.1), то  $(x^{ik}, y^{ik}, v^{ik}) \in S$  и  $x^{ik} \in Pr_X(Z)$ , т.е. точка  $x^{ik}$  может служить стартовой для локального поиска на шаге 5.

Для вычисления интервала одномерного поиска  $[\beta_-, \beta_+]$  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 [3, 9]. Например, в данной работе для этой цели использовалась соответствующая подпрограмма из пакета Xpress-MP [140]. Для решения второй можно использовать известную стратегию глобального поиска для задач выпуклой максимизации [69], при этом достаточно иметь сравнительно грубые оценки на  $\beta_+$ .

Теперь обратимся к линеаризованной задаче  $(\mathcal{LQ}((u, w), \gamma_k))$ —(2.7.1) (шаг 4), которая в терминах задачи  $(\mathcal{P})$ —(2.2.5) записывается следующим образом:

$$(\mathcal{LQ}((u, w), \gamma_k)) \quad \left. \begin{aligned} & \frac{1}{4} \|A_1 x - v\|^2 + \langle d^1, y \rangle + \langle b^1, v \rangle - \langle \nabla h(u, w), (x, v) \rangle \downarrow \min_{x, y, v} \\ & Ax \leq b, \quad A_1 x + B_1 y \leq b^1, \quad v B_1 = -d^1, \quad v \geq 0, \\ & \langle c, x \rangle + \langle d, y \rangle \leq \gamma_k, \end{aligned} \right\} \quad (2.8.1)$$

где  $\nabla h(u, w) = (\nabla_x h, \nabla_v h)$ ,  $\nabla_x h = \frac{1}{2}(A_1 u + w)A_1$ ,  $\nabla_v h = \frac{1}{2}(A_1 u + w)$ .

Задача  $(\mathcal{LQ}((u, w), \gamma_k))$ —(2.8.1) — это задача выпуклого квадратичного программирования, и для ее решения также применимы соответствующие методы выпуклой оптимизации [3, 9] и современные пакеты прикладных программ, например, Xpress-MP [140].

Далее, обратим внимание на то, что функция  $h(\cdot)$ , задающая базовую невыпуклость в задаче  $(\mathcal{P}_\rho)$ —(2.6.6), не зависит от переменной  $y$ . В связи с этим аппроксимация поверхности уровня функции  $h(\cdot)$  на шаге 3 строится только по переменным  $x, v$ , что, вообще говоря, уменьшает количество точек в аппроксимации. Кроме того, целевая функция  $f(\cdot)$  не зависит от переменной  $v$ , что также будет использовано ниже при построении аппроксимации.

Отметим также, что этапы решения задачи уровня и вычисления величины  $\eta_k(\beta)$  заменены на непосредственное сравнения значений целевой функции в точках  $(\bar{x}^i, \bar{y}^i)$  со значением в текущей точке  $(x^k, y^k)$  (шаги 6–9), что обычно эффективнее с вычислительной точки зрения [19, 44, 80].

## 2.9 Вычислительный эксперимент

Тестирование алгоритма глобального поиска проходило в 3 этапа. Вначале с помощью численного эксперимента была выбрана наиболее подходящая аппроксимация поверхности уровня выпуклой функции  $h(\cdot)$ . Затем были решены серии случайно сгенерированных задач средней размерности, а также определены наиболее сложные классы задач для алгоритма глобального поиска. Далее на этих классах были решены задачи высокой размерности.

### 2.9.1 Первый этап. Выбор аппроксимации поверхности уровня

Как показывает вычислительный опыт по решению невыпуклых задач [19, 44, 69, 80], при построении аппроксимации поверхности уровня функции, задающей базовую невыпуклость, в каждой задаче требуется некоторый предварительный этап вычислительного эксперимента, позволяющий определить, какой вид аппроксимации наиболее полно учитывает специфику задачи. С одной стороны, аппроксимация должна позволять строить точки лучшие, чем текущая критическая точка, с другой стороны, слишком большое количество точек аппроксимации может существенно замедлять работу алгоритма. Для исследуемой в работе задачи  $(\mathcal{P}_\rho)$ —(2.6.6) был предложен следующий способ построения

аппроксимаций (см. также [19, 44, 69, 80]).

Для заданного множества направлений

$$Dir = \{(a^i, c^i) \in \mathbb{R}^{m+q} \mid (a^i, c^i) \neq 0, i = 1, \dots, N_A\}$$

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

$$(u^i, w^i) = \lambda_i(a^i, c^i), \quad \lambda_i > 0, \quad i = 1, \dots, N_A, \quad (2.9.1)$$

$$h(u^i, w^i) = \beta_k, \quad i = 1, \dots, N_A. \quad (2.9.2)$$

Подставляя выражение (2.9.1) для  $(u^i, w^i)$  в (2.9.2) и решая соответствующее квадратное уравнение с учетом коллинеарности векторов из  $\mathcal{A}_k$  и  $Dir$ , получаем явное выражение для  $\lambda_i$ :

$$\lambda_i = 2\sqrt{\frac{\beta_k}{\|A_1 a^i + c^i\|^2}}.$$

Вопрос выбора множества направлений  $Dir$  для построения аппроксимации является ключевым, и для его решения необходимо использовать опыт решения невыпуклых задач с помощью теории глобального поиска [19, 44, 69, 80].

В задачах с многогранным допустимым множеством хорошо зарекомендовало себя множество направлений, в котором присутствуют стандартные базисные векторы [19, 44, 80]. Кроме того, неплохие результаты для задач д.с. программирования были получены с использованием дополнительной информации, получаемой в процессе вычислений, в частности, данных о текущей критической точке [44, 69, 80]. На основании вышесказанного, для решения задачи  $(\mathcal{P}_\rho)$ –(2.6.6) были выбраны следующие три множества направлений.

Первое множество направлений имеет самую простую структуру:

$$Dir_1 = \{e^i, i = 1, \dots, m + q\}.$$

При построении данного множества векторы  $x$  и  $v$  рассматриваются как компоненты одного вектора размерности  $m + q$ . При этом направления не зависят от текущей критической точки, и общее количество точек небольшое ( $m + q$ ).

Во втором множестве увеличим количество точек (теперь их  $m \cdot q$ ) за счет того, что изменение индексов будет происходить отдельно по переменной  $x$  и отдельно по  $v$ :

$$Dir_2 = \{(e^i, e^j), i = 1, \dots, m, j = 1, \dots, q\},$$

что, вообще говоря, больше отвечает структуре задачи, поскольку ограничения на переменные  $x$  и  $v$  задаются независимо друг от друга.

В третьем множество направлений (в отличие от первых двух) появляется зависимость от компоненты  $x^k$  текущей критической точки, а  $v^k$  остается без изменения, поскольку целевая функция  $f(\cdot)$  задачи  $(\mathcal{P}_\rho)$ –(2.6.6) от нее не зависит:

$$Dir_3 = \{(x^k + e^i, v^k), i = 1, \dots, m\}.$$

Аппроксимация, построенная с помощью  $Dir_3$ , содержит всего  $m$  точек.

В ходе вычислительного эксперимента по тестированию алгоритма глобального поиска будет, в частности, определено, какая из аппроксимаций  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  или  $\mathcal{A}_3$  (построенных с помощью направлений  $Dir_1$ ,  $Dir_2$  и  $Dir_3$  соответственно) наиболее полно учитывает специфику двухуровневой задачи.

Тестирование трех предложенных аппроксимаций проводилось на задачах размерностей от 2 до 60, в которых локальным поиском найти глобальный минимум не удалось. Для вычисления шага по  $\beta$  параметр  $M$  выбирался равным 4. В табл. 2.4 приведена информация о рассматриваемых тестовых задачах.

$N$	$class$	$m = n$	$f_*$	$ls$	$N$	$class$	$m = n$	$f_*$	$ls$
1	(0, 0, 2, 0, 0)	2	8	1	26	(0, 0, 5, 0, 15)	20	80	32678
2	(0, 0, 1, 1, 0)	2	6	2	27	(8, 2, 5, 5, 0)	20	38	128
3	(0, 0, 0, 0, 2)	2	8	4	28	(0, 5, 2, 10, 3)	20	60	32678
4	(0, 0, 1, 0, 1)	2	8	2	29	(4, 4, 4, 4, 4)	20	56	4096
5	(0, 0, 0, 1, 1)	2	6	4	30	(0, 10, 0, 5, 5)	20	70	1048576
6	(0, 0, 0, 2, 1)	3	8	8	31	(6, 6, 6, 6, 6)	30	84	262144
7	(0, 0, 2, 1, 0)	3	10	2	32	(5, 5, 4, 10, 6)	30	80	2097152
8	(0, 0, 0, 0, 3)	3	12	8	33	(0, 0, 10, 10, 10)	30	100	1048576
9	(1, 0, 0, 0, 2)	3	8	4	34	(0, 0, 0, 20, 10)	30	80	$1.074 \cdot 10^9$
10	(0, 0, 1, 2, 0)	3	8	4	35	(0, 10, 10, 10, 0)	30	100	1048576
11	(0, 0, 0, 2, 2)	4	12	16	36	(8, 8, 8, 8, 8)	40	112	16777216
12	(0, 0, 0, 0, 4)	4	16	16	37	(2, 8, 0, 0, 30)	40	152	$2.749 \cdot 10^{11}$
13	(0, 1, 1, 1, 1)	4	14	8	38	(0, 15, 10, 15, 0)	40	130	$1.074 \cdot 10^9$
14	(0, 0, 1, 2, 1)	4	12	8	39	(0, 0, 20, 0, 20)	40	160	1048576
15	(0, 0, 3, 0, 1)	4	16	2	40	(2, 8, 8, 2, 20)	40	148	$1.074 \cdot 10^9$
16	(0, 0, 0, 0, 5)	5	20	32	41	(10, 10, 10, 10, 10)	50	140	$1.074 \cdot 10^9$
17	(0, 1, 0, 1, 3)	5	18	32	42	(0, 0, 1, 0, 49)	50	200	$5.629 \cdot 10^{14}$
18	(1, 1, 1, 1, 1)	5	14	8	43	(14, 6, 30, 0, 0)	50	144	64
19	(0, 0, 4, 0, 1)	5	20	2	44	(4, 14, 6, 16, 10)	50	152	$1.099 \cdot 10^{12}$
20	(0, 0, 0, 2, 3)	5	16	32	45	(10, 0, 20, 0, 20)	50	160	1048576
21	(7, 0, 0, 1, 2)	10	10	8	46	(12, 12, 12, 12, 12)	60	168	$6.872 \cdot 10^{10}$
22	(0, 2, 7, 0, 1)	10	40	8	47	(0, 20, 0, 0, 40)	60	240	$1.153 \cdot 10^{18}$
23	(0, 0, 4, 6, 0)	10	28	64	48	(15, 0, 15, 0, 30)	60	180	$1.074 \cdot 10^9$
24	(0, 0, 6, 0, 4)	10	40	16	49	(5, 15, 10, 20, 10)	60	180	$3.518 \cdot 10^{13}$
25	(2, 2, 2, 2, 2)	10	28	64	50	(0, 25, 25, 5, 5)	60	230	$3.436 \cdot 10^{10}$

Таблица 2.4: Свойства сгенерированных задач



В дополнение к обозначениям табл. 2.3 здесь и далее  $N$  — номер задачи,  $ls$  — приближенное количество локальных экстремумов в задаче, вычисленное в соответствии с предложением 2.7.1. Как можно видеть из табл. 2.4, общее количество локальных решений для задач всех типов возрастает с увеличением размерности и для задач размерности более  $10 \times 10$  уже весьма велико.

В табл. 2.5 для каждой из задач приведены результаты работы АГП с использованием аппроксимаций  $\mathcal{A}_i$  ( $i = 1, 2, 3$ ) на шаге 3. Здесь  $f_*$  — значение целевой функции в точке глобального минимума,  $f_i$  — найденное значение целевой функции,  $St_i$  — количество пройденных алгоритмом критических точек,  $Loc_i$  — количество запусков процедуры локального поиска,  $T_i$  — время работы алгоритма в секундах.

Как можно видеть из табл. 2.5, точек первой аппроксимации явно недостаточно для поиска глобального решения. Причем с ростом размерности количество решенных задач заметно уменьшается. В итоге, с помощью  $\mathcal{A}_1$  решено всего 54% сгенерированных задач (полученное значение целевой функции для нерешенных задач выделено жирным шрифтом).

Применение второй аппроксимации позволило решить все задачи, но для этого алгоритму потребовалось гораздо больше времени: с ростом размерности количество точек в  $\mathcal{A}_2$  увеличивается быстрее, чем в первом случае. Так, для решения задачи №50 алгоритму потребовалось более трех часов.

С добавлением же информации о текущей критической точке (аппроксимация  $\mathcal{A}_3$ ) удалось решить все задачи с помощью  $m$  точек в аппроксимации, причем гораздо быстрее. Например, для решения последней задачи потребовалось менее двух минут. В дальнейшем вычислительном эксперименте на шаге 3 АГП аппроксимация будет строиться с помощью множества направлений  $Dir_3$ .

Далее отметим, что, несмотря на то, что в тестовых двухуровневых задачах количество локальных решений ( $ls$ ) невелико, для отыскания глобального решения алгоритму зачастую требуется значительно большее число запусков процедур локального поиска ( $Loc_i$ ). Кроме того, в некоторых случаях количество итераций алгоритма ( $St_i$ ), т.е. число критических точек, доставляющих улучшение целевой функции, также превышает  $ls$ . Например, в первой задаче всего один локальный экстремум, тем не менее, количество запусков локального поиска для АГП с использованием, например, аппроксимации  $\mathcal{A}_2$ , равно 50, а количество итераций АГП равно 2.

$N$	$f_*$	$f_1$	$St_1$	$Loc_1$	$T_1$	$f_2$	$St_2$	$Loc_2$	$T_2$	$f_3$	$St_3$	$Loc_3$	$T_3$
1	8	8	2	34	0.09	8	2	50	0.08	8	2	10	0.03
2	6	6	6	35	0.06	6	2	49	0.08	6	2	9	0.03
3	8	8	2	34	0.05	8	2	50	0.08	8	2	10	0.03
4	8	8	2	34	0.05	8	2	50	0.08	8	2	10	0.03
5	6	6	2	34	0.06	6	2	50	0.08	6	2	10	0.03
6	8	<b>9</b>	1	49	0.09	8	2	110	0.23	8	2	22	0.08
7	10	10	3	65	0.13	10	2	110	0.22	10	3	20	0.08
8	12	12	3	51	0.11	12	2	110	0.22	12	3	15	0.05
9	8	8	2	50	0.09	8	2	110	0.23	8	2	14	0.06
10	8	<b>10</b>	1	49	0.09	8	2	110	0.22	8	2	19	0.06
11	12	<b>14</b>	1	65	0.16	12	2	194	0.48	12	3	39	0.19
12	16	16	3	67	0.16	16	2	194	0.48	16	3	19	0.08
13	14	14	2	68	0.17	14	2	194	0.50	14	2	20	0.09
14	12	<b>13</b>	1	65	0.16	12	2	193	0.48	12	2	17	0.08
15	16	16	2	66	0.16	16	2	194	0.47	16	2	18	0.08
16	20	20	3	83	0.25	20	2	302	0.91	20	3	23	0.13
17	18	18	3	84	0.25	18	2	302	0.92	18	3	24	0.13
18	14	14	3	86	0.28	14	2	302	0.99	14	3	26	0.16
19	20	20	2	82	0.25	20	2	302	0.92	20	2	22	0.11
20	16	<b>18</b>	2	82	0.25	16	2	302	0.94	16	3	39	0.22
21	10	10	3	177	1.17	10	2	1202	7.69	10	2	42	0.50
22	40	40	2	162	1.05	40	2	1202	8.00	40	2	42	0.50
23	28	<b>30</b>	1	161	1.08	28	2	1502	10.20	28	2	54	0.66
24	40	40	3	163	1.08	40	2	1202	7.83	40	2	42	0.50
25	28	<b>30</b>	4	218	1.42	28	2	1202	8.11	28	3	59	0.72
26	80	80	3	323	6.41	80	2	4802	93.61	80	3	83	3.03
27	38	<b>46</b>	1	321	6.33	38	2	4802	94.86	38	3	119	4.19
28	60	<b>65</b>	1	321	6.41	60	2	4802	93.53	60	3	123	4.31
29	56	<b>62</b>	2	415	8.16	56	2	4802	95.42	56	2	82	2.92
30	70	<b>75</b>	1	321	6.42	70	2	4802	95.19	70	4	147	5.27
31	84	<b>94</b>	2	623	26.84	84	2	10802	491.47	84	4	191	14.88
32	80	<b>94</b>	1	481	21.06	80	2	10802	474.33	80	7	308	23.91
33	100	<b>112</b>	3	743	33.34	100	2	10802	494.00	100	6	313	24.81
34	80	<b>90</b>	1	481	22.33	80	2	10802	454.52	80	6	294	22.84
35	100	<b>112</b>	1	481	21.17	100	2	10802	477.52	100	4	233	18.17
36	112	<b>124</b>	1	641	48.98	112	2	19202	1481.86	112	5	366	50.69
37	152	152	2	642	48.78	152	2	19202	1458.23	152	2	162	22.45
38	130	<b>134</b>	1	641	49.34	130	2	24002	1787.19	130	3	231	31.84
39	160	160	2	642	48.50	160	2	19202	1466.84	160	2	162	22.25
40	148	<b>150</b>	3	661	50.22	148	3	24003	1812.73	148	2	162	22.55
41	140	<b>156</b>	1	801	102.83	140	3	37503	4207.84	140	4	370	82.42
42	200	200	2	802	100.14	200	2	30002	3058.93	200	2	202	45.27
43	144	144	5	926	115.47	144	2	30002	3158.30	144	2	202	44.80
44	152	<b>168</b>	1	801	101.70	152	3	37503	4184.02	152	6	525	117.73
45	160	160	3	805	102.14	160	2	30002	3065.39	160	2	202	45.89
46	168	<b>188</b>	2	1471	282.45	168	3	54003	8990.30	168	5	459	154.34
47	240	240	2	962	178.42	240	2	43202	6824.62	240	2	242	81.48
48	180	180	4	1088	206.42	180	2	43202	6806.95	180	2	242	80.38
49	180	<b>202</b>	1	961	185.95	180	3	54003	8813.71	180	7	743	244.75
50	230	<b>236</b>	4	1523	285.50	230	4	75604	11893.82	230	3	353	118.66

Таблица 2.5: Сравнение аппроксимаций

Подобные ситуации можно объяснить, во-первых, тем фактом, что решаемая невыпуклая задача эквивалентна двухуровневой только в глобальном смысле (т.е. в задаче  $(\mathcal{P})$ –(2.6.1) могут существовать локальные решения, не являющиеся таковыми в  $(\mathcal{BP})$ –(2.1.1)) [90, 103], а, во-вторых, предложенная процедура локального поиска гарантирует нахождение только критических точек (т.е., вообще говоря, только лишь допустимых), которые

оказываются слабее, чем локальные решения в задаче  $(\mathcal{BP})$ –(2.1.1). Наконец, некоторые из найденных критических точек могут совпадать. Тем не менее, с ростом размерности, количество запусков локального поиска, которое требуется для нахождения глобального решения, становится значительно меньше числа локальных экстремумов в задаче  $(\mathcal{BP})$ –(2.1.1).

## 2.9.2 Второй этап. Оценка сложности задач

Целью **второго этапа** вычислительного эксперимента было определение наиболее сложных (в зависимости от набора классов) задач для АГП. Были решены серии задач средней размерности с различным сочетанием задач-ядер, обязательно содержащих задачи-ядра 3 и 5 классов в соответствии с результатами тестирования локального поиска (см. табл. 2.3).

В табл. 2.6 представлены результаты тестирования АГП на задачах размерности  $m+n$  от 4 до 100.

В табл. 2.6 использовались следующие обозначения:  $m+n$  — размерность задачи (суммарное количество переменных на обоих уровнях задачи  $(\mathcal{BP})$ –(2.1.1)); *constr* — число ограничений в задаче  $(\mathcal{BP})$ –(2.1.1); *Seria* — количество рассматриваемых задач (например,  $9 \times 10000$  означает, что было решено 9 серий по 10000 случайно сгенерированных задач); *ls<sub>max</sub>* — максимальное (среди всех серий задач данной размерности) количество локальных экстремумов в генерируемых задачах, характеризующее их сложность; *UnSol* — процент нерешенных задач; *St<sub>av</sub>* — среднее число пройденных критических точек; *Loc<sub>av</sub>* — среднее число запусков процедуры локального поиска; *T<sub>av</sub>* — среднее время решения одной задачи.

По результатам, представленным в табл. 2.6, можно видеть, что в среднем АГП потребовалось не более пяти итераций для нахождения глобального минимума. Отметим, что с помощью АГП в целом удалось решить более 99% сгенерированных задач. При этом с ростом размерности количество нерешенных задач уменьшилось, несмотря на значительное возрастание количества локальных экстремумов в задаче (см. колонку *ls<sub>max</sub>*). Начиная с размерности 40, были решены все предложенные задачи. Возможно, это объясняется уменьшением (в связи с повышением размерности) количества задач в серии, вследствие чего вероятность "плохих" задач уменьшается, однако тенденция уменьшения количества нерешенных задач прослеживается и для одинакового числа задач в серии (размерности 6, 8, 10).

$m + n$	$constr$	$Seria$	$ls_{max}$	$UnSol(\%)$	$St_{av}$	$Loc_{av}$	$T_{av}$
4	10	$9 \times 10000$	4	0	1,80	9,88	0,05
6	15	$22 \times 10000$	8	0,20	1,96	14,48	0,09
8	20	$22 \times 10000$	16	0,16	1,94	19,14	0,15
10	25	$22 \times 10000$	32	0,12	2,11	24,38	0,23
20	50	$32 \times 1000$	1024	0,05	2,37	48,65	0,94
40	100	$21 \times 100$	$2^{20}$	0	3,44	125,91	2,60
60	150	$21 \times 100$	$2^{30}$	0	4,18	217,34	9,87
80	200	$21 \times 100$	$2^{40}$	0	3,68	258,36	12,44

Таблица 2.6: Решение серий задач средних размерностей

Очевидно, что наибольший интерес для анализа представляют именно те серии задач, в которых с помощью АГП получить решение удалось не во всех случаях. Подробные результаты решения именно таких задач содержатся в табл. 2.7, где в дополнение к предыдущим таблицам  $Solv$  — число решенных задач из серии;  $St$  — общее количество пройденных критических точек;  $Loc$  — общее на серию количество запусков локального поиска.

$N$	$m + n$	$class$	$ls$	$Seria$	$Solv$	$St$	$Loc$	$T_{av}$
1	6	(0, 0, 0, 2, 1)	8	10000	9586	19586	184667	0.04
2	6	(0, 0, 1, 2, 0)	4	10000	9966	12819	144847	0.03
3	8	(0, 1, 1, 2, 0)	8	10000	9997	13433	190314	0.05
4	8	(0, 0, 1, 2, 1)	8	10000	9898	22547	241883	0.06
5	8	(0, 0, 1, 3, 0)	8	10000	9980	13014	192376	0.05
6	8	(0, 0, 0, 2, 2)	16	10000	9755	22669	258735	0.07
7	10	(1, 1, 0, 2, 1)	16	10000	9997	19997	274013	0.09
8	10	(1, 0, 0, 3, 1)	16	10000	9951	19951	291451	0.09
9	10	(0, 1, 2, 2, 0)	8	10000	9980	19899	274085	0.09
10	10	(0, 0, 3, 2, 0)	4	10000	9961	23754	301979	0.10
11	10	(0, 0, 0, 2, 3)	32	10000	9829	28522	309005	0.10
12	20	(0, 0, 1, 2, 7)	512	1000	998	3545	58451	0.40
13	20	(0, 0, 4, 6, 0)	64	1000	997	2800	64684	0.45
14	20	(0, 0, 0, 5, 5)	1024	1000	996	3526	71299	0.50
15	20	(1, 0, 4, 5, 0)	32	1000	996	2781	63629	0.44
16	20	(1, 1, 2, 3, 3)	128	1000	999	3497	66655	0.46
17	20	(0, 0, 0, 2, 8)	1024	1000	998	2615	50316	0.35
18	20	(0, 0, 4, 6, 0)	64	1000	997	2800	64682	0.45

Таблица 2.7: Характеристики сложных задач

Стоит отметить, что для глобального поиска наиболее сложными оказались сочетания классов 3 и 4, а также классов 4 и 5. Именно при таких комбинациях некоторые задачи из

серии не были решены (см. табл. 2.7). При этом всего в 5 из 18 сложных для АГП серий количество локальных экстремумов  $ls$  достигает максимально возможного значения  $ls_{max}$  (серии №№ 1, 6, 11, 14 и 17), т.е. можно говорить о том, что сложность задач придает не только количество локальных экстремумов, но и сама структура сгенерированных задач.

### 2.9.3 Третий этап. Решение задач высокой размерности

Полученная информация о сложности задач для АГП использована на **третьем этапе** вычислительного эксперимента при генерации задач высокой размерности, результаты решения которых представлены в табл. 2.8.

$m + n$	$constr$	$Seria$	$ls_{max}$	$Solv$	$St$	$Loc$	$T_{av}$
150	375	10	$2^{75}$	10	57	6177	0:03:26.68
200	500	10	$2^{100}$	10	51	6861	0:07:59.72
300	750	10	$2^{150}$	10	103	16418	0:46:24.81
400	1000	10	$2^{200}$	10	37	8607	0:47:42.74
500	1250	10	$2^{250}$	10	20	5010	0:53:24.26
600	1500	10	$2^{300}$	9	21	13018	1:56:33.12
800	2000	10	$2^{400}$	10	24	17628	11:00:04.94
1000	2500	10	$2^{500}$	10	24	20029	23:04:45.37

Таблица 2.8: Решение задач средней и высокой размерности

Для ускорения работы АГП значение параметра  $M$ , определяющего количество частей, на которые делится отрезок  $[\beta_-, \beta_+]$ , здесь было выбрано равным 2.

Интересно отметить, что наибольшее количество критических точек алгоритм прошел на размерности 300 (в среднем более 10 итераций). С дальнейшим же увеличением размерности, количество итераций уменьшилось до двух. Тем не менее, поскольку количество точек в аппроксимации зависит от размерности задачи, с увеличением числа переменных растет и количество запусков локального поиска, что отражается на времени работы АГП.

По результатам, представленным в табл. 2.8, можно видеть, что АГП решил с заданной точностью все, кроме одной, из предложенных задач вплоть до размерности 1000 за приемлемое время.

Насколько можно судить по доступной литературе, размерность, которую приводят

авторы публикаций при тестировании предлагаемых алгоритмов, является недостаточной для решения практических задач. Эффективно решаются только линейно-линейные задачи средней размерности. Так, в [89] приведены результаты решения таких задач с суммарным числом переменных до 130, а в [130] — до 200.

Таким образом, проведенный вычислительный эксперимент подтверждает эффективность применения к решению двухуровневых задач высокой размерности подхода, базирующегося на теории глобального поиска для задач с d.c. ограничением.

## 2.10 Задача планирования производства в условиях неизвестного спроса

После проверки работы АГП на тестовых задачах была рассмотрена практическая задача планирования производства в условиях неизвестного спроса [90], постановка которой приведена во введении.

Напомним, что в данной модели производитель рассматривается как игрок верхнего уровня, который выбирает объемы производства в каждый период ( $x_{jt}$ ) и рекламную стратегию ( $v_{jt}$ ) для каждого из своих  $n$  товаров на каждый момент времени ( $j = 1, \dots, n$ ,  $t = 1, \dots, T$ ). Предполагается, что потребности покупателя ( $y_{jt}$ ) (игрока нижнего уровня), формируются под влиянием рекламных усилий производителя, и он стремится удовлетворить их за наименьшую цену. Спрос обязательно должен быть удовлетворен. Для этого можно использовать излишки ( $w_{jt}$ ), произведенные ранее, а для покрытия нехватки ( $z_{jt}$ ) можно прибегать к субконтрактам, то есть заказывать другому производителю то, что дороже будет производить самому. Цель верхнего уровня — максимизировать прибыль, которая представляет собой разницу между доходом и затратами. Нижний уровень минимизирует свои затраты.

В [90] приведены числовые данные для расчета оптимального плана производства для предприятия, выпускающего электрооборудование. В поставленной задаче количество производимых продуктов  $n = 2$ , количество временных интервалов  $T = 4$ . Таким образом, в данной линейной двухуровневой задаче 16 переменных на верхнем уровне и 24 переменные на нижнем уровне. Полная математическая постановка задачи приведена ниже.

На верхнем уровне минимизируются издержки производителя (максимизируется при-

быль):

$$\begin{aligned}
 F(x, v, y, w, z) = & 50x_{11} + 75x_{12} + 100x_{13} + 60x_{14} + 75x_{21} + 95x_{22} + 200x_{23} + 100x_{24} + \\
 & + v_{11} + v_{12} + v_{13} + v_{14} + v_{21} + v_{22} + v_{23} + v_{24} - 75y_{11} - 115y_{12} - 160y_{13} - \\
 & - 90y_{14} - 115y_{21} - 200y_{22} - 300y_{23} - 160y_{24} + 25w_{11} + 25w_{12} + 25w_{13} + 25w_{14} + \\
 & + 25w_{21} + 25w_{22} + 25w_{23} + 25w_{24} + 80z_{11} + 110z_{12} + 140z_{13} + 95z_{14} + 110z_{21} + \\
 & + 170z_{22} + 140z_{23} + 95z_{24} \downarrow \min,
 \end{aligned}$$

Ресурсные ограничения:

$$\begin{aligned}
 3x_{11} + 3x_{12} + 3x_{13} + 3x_{14} &\leq 2750, \\
 2x_{21} + 2x_{22} + 2x_{23} + 2x_{24} &\leq 2000, \\
 2x_{11} + 3x_{21} &\leq 2700, \quad 2x_{12} + 3x_{22} \leq 2700, \\
 2x_{13} + 3x_{23} &\leq 2700, \quad 2x_{14} + 3x_{24} \leq 2700.
 \end{aligned}$$

Расходы на рекламу:

$$\begin{aligned}
 v_{11} + v_{12} + v_{13} + v_{14} + v_{21} + v_{22} + v_{23} + v_{24} &\leq 10000, \\
 -v_{11} - v_{12} &\leq -1500, \quad -v_{12} - v_{13} \leq -1500, \\
 -v_{13} - v_{14} &\leq -1500, \quad -v_{21} - v_{22} \leq -1500, \\
 -v_{22} - v_{23} &\leq -1500, \quad -v_{23} - v_{24} \leq -1500.
 \end{aligned}$$

Материальный баланс:

$$w_{jt} - w_{j,t-1} + y_{jt} - z_{jt} = x_{jt} \quad \forall j \in \{1, 2\}, t \in \{1, \dots, 4\}.$$

Целевая функция нижнего уровня:

$$75y_{11} + 115y_{12} + 160y_{13} + 90y_{14} + 115y_{21} + 200y_{22} + 300y_{23} + 160y_{24} \downarrow \min_{y,w,z}.$$

Ограничения на спрос:

$$\begin{aligned}
 -y_{11} - y_{12} - y_{13} - y_{14} &\leq -100, \quad -y_{21} - y_{22} - y_{23} - y_{24} \leq -100, \\
 2y_{11} - 3y_{12} &\leq 0, \quad y_{12} - 2y_{13} \leq 0, \quad 3y_{13} - 4y_{14} \leq 0, \\
 y_{21} - 2y_{22} &\leq 0, \quad y_{22} - 4y_{23} \leq 0, \quad 3y_{23} - 5y_{24} \leq 0.
 \end{aligned}$$

Влияние рекламы на спрос:

$$\begin{aligned}
 5v_{11} - 100y_{11} &\leq 0, \quad 3v_{11} + 3.5v_{12} - 100y_{12} \leq 0, \\
 8v_{21} - 100y_{21} &\leq 0, \quad 2v_{21} + 4v_{22} - 100y_{22} \leq 0, \\
 1.5v_{11} + 2v_{12} + 3.25v_{13} - 100y_{13} &\leq 0, \\
 0.5v_{11} + v_{12} + 1.75v_{13} + 3.7v_{14} - 100y_{14} &\leq 0, \\
 1.5v_{21} + 2v_{22} + 3.1v_{23} - 100y_{23} &\leq 0, \\
 0.5v_{21} + v_{22} + 1.9v_{23} + 3.9v_{24} - 100y_{24} &\leq 0.
 \end{aligned}$$

Ограничения на хранение товара:

$$\begin{aligned} w_{11} + 1.5w_{21} &\leq 225, & w_{12} + 1.5w_{22} &\leq 225, \\ w_{13} + 1.5w_{23} &\leq 225, & w_{14} + 1.5w_{24} &\leq 225. \end{aligned}$$

Остановимся на некоторых особенностях реализации АГП для данной задачи. Во-первых, целевая функция верхнего уровня не ограничена снизу на множестве  $Z$ . В связи с этим для решения вспомогательной задачи (2.7.5) в постановку были добавлены дополнительные ограничения: предполагалось что все переменные спроса не превосходят значения 100000, что согласуется с практикой.

Далее, для построения аппроксимации поверхности уровня функции  $h(\cdot)$  вместо множества направлений

$$Dir_3 = \{(x^k + e^i, v^k), i = 1, \dots, m\}$$

было выбрано

$$Dir_4 = \{(q^i, v^k), i = 1, \dots, m\},$$

где  $q^i$  — решение задачи линейного программирования

$$\begin{aligned} \langle c^i, x \rangle \downarrow \min_{x,y}, \\ (x, y) \in Z, \end{aligned} \tag{2.10.1}$$

где  $c^i$  — вектор, у которого  $i$ -я компонента совпадает с  $i$ -й компонентой вектора  $c$  из целевой функции верхнего уровня, а остальные равны нулю. Иными словами, для нахождения точек аппроксимации исходная задача заменяется на релаксированную (без ограничения-равенства), а затем осуществляется минимизация отдельных слагаемых функции верхнего уровня.

В табл. 2.9 представлены результаты работы АГП, а также решение из [90].

Для получения данной точки алгоритму потребовалось 2 итерации. Проанализируем результаты, полученные с помощью АГП. Как можно видеть из таблицы, в первый и четвертый периоды для первого продукта оказывается выгоднее покупать продукцию по субконтракту, чем производить, оставляя ресурсы на производство товара в третий период (когда цена на товар максимальна). При этом в первом периоде продукция закупается и хранится до второго периода, поскольку в этом случае затраты (включая затраты на хранение) ниже, чем стоимость продукта по субконтракту во второй период.

Производство второго продукта осуществляется в первый и во второй периоды, а оставшаяся нехватка товара восполняется с помощью субконтрактов в третий и четвертый временные отрезки. В результате такой стратегии производитель получает прибыль



Переменная	АГП				[90]			
	Период				Период			
	1	2	3	4	1	2	3	4
	Продукт 1				Продукт 1			
Производство	0	0	916.67	0	75.0	121.3	0	53.4
Реклама	0	1500	0	1500	1500	0	1500.3	0
Спрос	0	52.5	691.67	518.75	75.0	50.0	71.3	53.4
Излишек	52.5	0	225	0	0	71.3	0	0
Субконтракт	52.5	0	0	293.75	0	0	0	0
	Продукт 2				Продукт 2			
Производство	100	900	0	0	440.0	349.0	0	77.4
Реклама	0	1500	0	5500	1500.0	0	1500.0	0
Спрос	100	750	187.5	229.5	440.0	220.0	129.0	77.4
Излишек	0	150	0	0	0	129	0	0
Субконтракт	0	0	37.5	229.5	0	0	0	0

Таблица 2.9: Решение задачи планирования производства в условиях неизвестного спроса

\$153348.75. Издержки потребителя составляют \$417681.67. Прибыль верхнего уровня в случае, представленном в [90], равна \$68.317, а издержки нижнего уровня \$173.273, т.е. полученная алгоритмом глобального поиска точка лучше по значению задачи, чем в [90].

## 2.11 Основные результаты главы

В данной главе получены следующие результаты.

1. Предложен и обоснован специальный метод локального поиска (отыскания критических точек) в линейной двухуровневой задаче (ЛДЗ), учитывающий ее структуру.
2. На основе взаимосвязи ЛДЗ и невыпуклой задачи математического программирования с билинейным ограничением-равенством, а также теории глобального поиска в задачах с d.c. ограничением разработан алгоритм глобального поиска (АГП) оптимистических решений в ЛДЗ.
3. Разработаны программы, реализующие предложенные алгоритмы локального и глобального поисков. Проведено их тестирование на серии специальным образом сгенерированных двухуровневых задач с большим количеством локальных экстремумов.
4. Проведен обширный вычислительный эксперимент по решению ЛДЗ различной сложности и размерности (до 500 переменных на каждом уровне), показавший эффективность предложенной методики решения двухуровневых задач. В частности, в доступной литературе не удалось обнаружить результаты решения ЛДЗ подобной размерности. Кроме того, алгоритм был протестирован на одной практической задаче из экономики.

## Глава 3

# Методы решения уравнений с d.c. функциями

Как уже говорилось во второй главе (см., например, §2.3), условия глобальной оптимальности в задачах с ограничением-равенством обладают алгоритмическим свойством, позволяющим построить точку лучшую, чем исследуемая. Однако для этого необходима процедура, позволяющая эффективно находить допустимую по ограничению-равенству точку, используя два вектора, в которых данная функция принимает значения противоположного знака. При этом поиск решения уравнения должен осуществляться с учетом целевой функции исходной задачи. В первой части данной главы предлагается алгоритм поиска решения одного уравнения со многими неизвестными с учетом d.c. структуры функции-ограничения. При этом в результате работы алгоритма находится наиболее подходящей, с точки зрения целевой функции, корень уравнения. Доказывается теорема сходимости предложенного метода.

Вторая часть главы посвящена нахождению точек, удовлетворяющих нескольким нелинейным ограничениям-равенствам. Для решения системы уравнений предлагается использовать вариационный подход, заключающийся в сведении системы уравнений к задаче безусловной минимизации. Показано, что целевая функция в полученной задаче является d.c. функцией. Далее для решения негладкой оптимизационной задачи применяется аппарат d.c. минимизации [69], основанный на условиях глобальной оптимальности.

### 3.1 Решение одного уравнения с d.c. функцией

Рассмотрим уравнение вида

$$(\mathcal{EP}) \quad F(x) = 0, \quad x \in \mathbb{R}^n, \quad (3.1.1)$$

где  $F(\cdot)$  является d.c. функцией, т.е. функцией, представимой в виде разности двух выпуклых функций  $g(\cdot)$  и  $h(\cdot)$ :

$$F(x) = g(x) - h(x), \quad (3.1.2)$$

где  $g$  — непрерывно дифференцируемая функция. Предположим, что существуют точки  $u$  и  $v$  из  $\mathbb{R}^n$ , удовлетворяющие следующим соотношениям:

$$F(u) \triangleq g(u) - h(u) < 0 < F(v) \triangleq g(v) - h(v). \quad (3.1.3)$$

Напомним, что решение поставленной задачи необходимо в общей задаче с d.c. ограничением-равенством для поиска допустимой точки лучшей, чем текущая критическая (см. §2.3). В задаче

$$(\mathcal{P}_0) \quad \left. \begin{array}{l} f(x) \downarrow \min, \quad x \in S, \\ F(x) \triangleq g(x) - h(x) = 0, \end{array} \right\} \quad (3.1.4)$$

в качестве точки  $u$  выступает точка, нарушившая условия оптимальности, в которой  $F(u) < 0$  и при этом  $f(u) \leq f(z)$ , где  $z$  — текущая критическая точка. Для нахождения же точки  $v$ , в силу условий регулярности  $(\mathcal{H}_1)$ –(2.3.6), достаточно решить выпуклую релаксированную задачу (2.3.9) без учета ограничения-равенства. В этом случае

$$F(v) > 0, \quad f(v) < f(u) \leq f(z). \quad (3.1.5)$$

В силу (3.1.3) существует число  $\lambda \in ]0; 1[$ , удовлетворяющее условию:

$$x_\lambda = \lambda u + (1 - \lambda)v = v + \lambda(u - v) : F(x_\lambda) = 0, \quad (3.1.6)$$

или, что то же самое,

$$g(x(\lambda)) = h(x(\lambda)). \quad (3.1.7)$$

Таким образом, для отыскания корня уравнения (3.1.7) достаточно провести одномерный поиск по  $\lambda \in ]0; 1[$ . Однако решение уравнения (3.1.7) может быть не единственным и желательно выбрать тот коэффициент выпуклой комбинации  $\lambda$ , которому будет соответствовать наименьшее значение целевой функции  $f(x(\lambda))$ . Это объясняется тем, что, поскольку

$$f(x(\lambda)) \leq \lambda f(u) + (1 - \lambda)f(v) < f(z), \quad (3.1.8)$$

то с учетом (3.1.5) получаем, что чем ближе  $\lambda$  к нулю, тем меньше верхняя оценка для значения  $f(x(\lambda))$ , а в случае линейной функции  $f(\cdot)$  и меньше само значение  $f(x(\lambda))$ .

Поэтому далее будем искать среди решений уравнения (3.1.7) то, которое соответствует минимальному  $\lambda$ .

Далее предлагается процедура выпуклой комбинации, целью которой является приближенное вычисление числа  $\lambda$ , удовлетворяющего (3.1.6). Будем искать число  $\mu_0$  не из условия

$$g(x(\mu_0)) = h(x(\mu_0)), \quad (3.1.9)$$

а из условия

$$g(v) + \mu_0 \langle \nabla g(v), u - v \rangle = \mu_0 h(u) + (1 - \mu_0) h(v), \quad (3.1.10)$$

т.е. произведем линеаризацию функции  $g(\cdot)$  в точке  $v$ , а вместо функции  $h$  запишем выпуклую комбинацию ее значений в точках  $u$  и  $v$ , так что

$$\mu_0 = \frac{F(v)}{h(u) - h(v) - \langle \nabla g(v), u - v \rangle}. \quad (3.1.11)$$

Таким образом, при описанном выборе  $\mu_0$  используется явное представление функции  $F(\cdot)$  как разности двух выпуклых функций. При этом в силу (3.1.3), а также выпуклости  $g(\cdot)$  знаменатель в (3.1.11) удовлетворяет условию

$$\begin{aligned} \varphi(u, v) &\triangleq h(u) - h(v) - \langle \nabla g(v), u - v \rangle \geq \\ &\geq h(u) - h(v) - g(u) + g(v) = F(v) - F(u) > 0. \end{aligned} \quad (3.1.12)$$

Поэтому, используя (3.1.3), (3.1.11) и (3.1.12), получаем

$$0 < \mu_0 = \frac{F(v)}{\varphi(u, v)} \leq \frac{F(v)}{F(v) - F(u)} < 1. \quad (3.1.13)$$

Следовательно, число  $\mu_0 \in ]0, 1[$  может быть коэффициентом выпуклой комбинации векторов  $u$  и  $v$ , удовлетворяющих условию (3.1.3).

Далее, в силу выпуклости  $g(\cdot)$  и  $h(\cdot)$ , а также, используя равенство (3.1.10), имеем

$$\begin{aligned} F(x(\mu_0)) &= g(x(\mu_0)) - h(x(\mu_0)) = g(\mu_0 u + (1 - \mu_0)v) - h(\mu_0 u + (1 - \mu_0)v) \geq \\ &\geq g(v + \mu_0(u - v)) - \mu_0 h(u) - (1 - \mu_0) h(v) \geq \\ &\geq g(v) + \mu_0 \langle \nabla g(v), u - v \rangle - \mu_0 h(u) - (1 - \mu_0) h(v) = 0, \end{aligned}$$

т.е. для точки  $x(\mu_0)$  имеет место неравенство

$$F(x(\mu_0)) \geq 0. \quad (3.1.14)$$

Принимая во внимание (3.1.11)–(3.1.14), с помощью последовательности коэффициентов выпуклой комбинации  $\{\mu_s\}$  можно предложить итеративную процедуру выпуклой комбинации (ПВК) построения последовательности  $\{x^s\}$ , начинающейся с  $x^0 = v$ , все точки которой лежат на отрезке  $[u; v]$ .

### Процедура выпуклой комбинации

**Шаг 0.** Положить  $s := 0$ ,  $x^s := v$ .

**Шаг 1.** Вычислить

$$\mu_s := \frac{F(x^s)}{h(u) - h(x^s) - \langle \nabla g(x^s), u - x^s \rangle}. \quad (3.1.15)$$

**Шаг 2.** Построить выпуклую комбинацию

$$x := x^s + \mu_s(u - x^s). \quad (3.1.16)$$

**Шаг 3.** Если  $F(x) \leq \varepsilon$ , то **Stop**,  $x$  —  $\varepsilon$ -решение уравнения  $F(x) = 0$ .

**Шаг 4.** Положить  $s := s + 1$ ,  $x^s := x$ , перейти на **Шаг 1**. #

Приведем геометрическую интерпретацию предложенного метода. На рис. 3.1 представлена первая итерация ПВК в одномерном случае. На отрезке  $[u, v]$  вместо функции  $h(\cdot)$  в д.с. разложении (3.1.2) в уравнении берется ее линейная интерполяция, т.е. строится отрезок, проходящий через точки  $(u, h(u))$ ,  $(v, h(v))$ . Затем происходит линейаризация функции  $g(\cdot)$ , т.е. строится касательная к графику функции  $y = g(x)$  в точке  $v$ . Точка  $x^1$  находится на пересечении двух прямых. При этом все последующие точки  $x^s$  оказываются "справа" от  $x_*$ .

Рассмотрим далее случаи, когда уравнение (3.1.1) задается выпуклой функцией и проследим, как при этом видоизменяется ПВК.

**1 случай.** Пусть  $g(x) \equiv 0$ . Тогда на шаге 1 коэффициент выпуклой комбинации находится следующим образом:

$$\mu_s = -\frac{h(x^s)}{h(u) - h(x^s)}. \quad (3.1.17)$$

Подставляя  $\mu_s$  в (3.1.16), получаем

$$x^{s+1} = x^s - \frac{h(x^s)(u - x^s)}{h(u) - h(x^s)}, \quad s = 1, 2, \dots \quad (3.1.18)$$

т.е. формулу для решения уравнения  $h(x) = 0$  методом хорд (метод ложного положения) [2, 4].

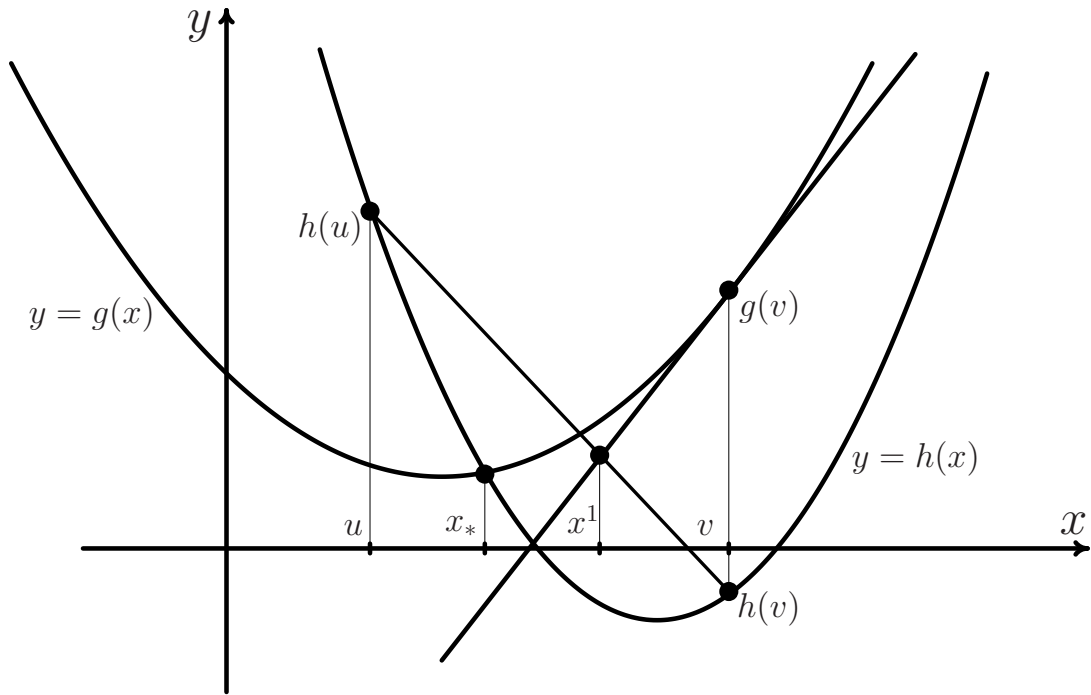


Рис. 3.1: Процедура выпуклой комбинации

**2 случай.** Положим теперь в уравнении (3.1.1) тождественно равной нулю функцию  $h(\cdot)$ . В этом случае на шаге 1 имеем

$$\mu_s = -\frac{g(x^s)}{\langle \nabla g(x^s), u - x^s \rangle}, \quad (3.1.19)$$

откуда получаем итерационный процесс

$$x^{s+1} = x^s - \frac{g(x^s)(u - x^s)}{\langle \nabla g(x^s), u - x^s \rangle}, \quad s = 1, 2, \dots \quad (3.1.20)$$

В случае, когда  $x \in \mathbb{R}^1$ , это не что иное, как формула метода Ньютона для уравнения  $g(x) = 0$ :

$$x^{s+1} = x^s - \frac{g(x^s)}{g'(x^s)}.$$

Отметим при этом, что прямое применение метода Ньютона или метода хорд к невыпуклой функции  $F(x)$  не всегда позволяет найти корень уравнения. Итерационный процесс может расходиться или сходиться к другому корню уравнения. Проиллюстрируем это на следующем примере [126].

### Пример 3.1.1 [126]

Рассмотрим уравнение  $-x^5 + x^3 + 4x = 0$  (см. рис. 3.2).

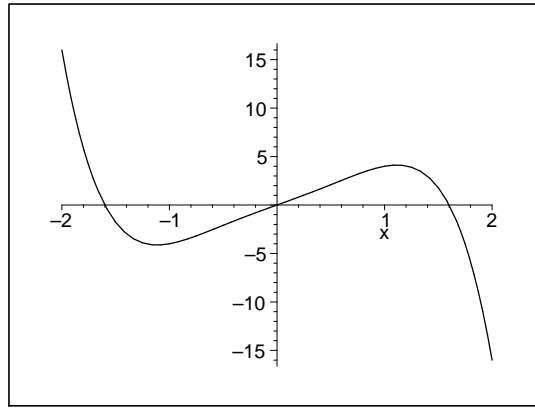


Рис. 3.2:  $F(x) = -x^5 + x^3 + 4x$

Отрезок  $[-1; 1]$  содержит корень уравнения  $x_* = 0$ . Если выбрать для метода Ньютона стартовую точку  $x_0 = 1$  или  $x_0 = -1$ , то метод заикнется, выдавая поочередно в качестве следующих приближений 1 или -1.

Если же применить формулу (3.1.18) для отрезка  $[u; v] = [-1.5; 1]$ , то итерационный процесс сойдется к корню  $x \approx -1.6005$ , не принадлежащему заданному отрезку.

В то же время ПВК позволяет найти корень  $x_* = 0$  с точностью 0.001 в первом случае за 6 итераций, во втором — за 9. На рис. 3.3 изображены графики функций  $g(x)$  и  $h(x)$  из д.с. разложения функции  $F$ , задающей уравнение.

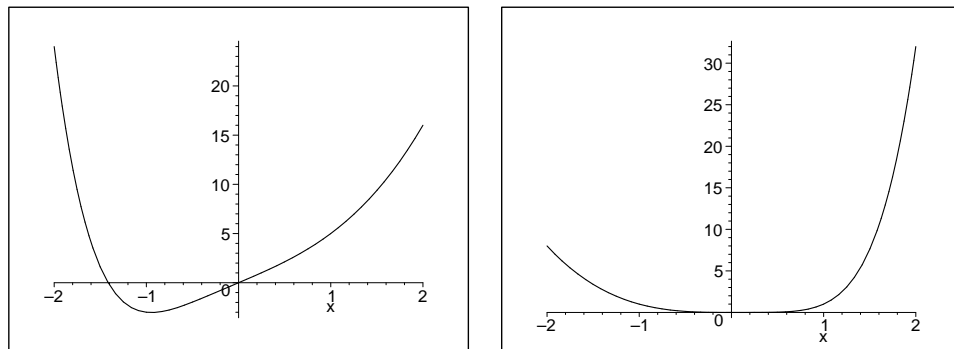


Рис. 3.3:  $y = g(x)$ ,  $y = h(x)$

$$\text{Здесь } g(x) = \begin{cases} -x^5 + 4x, & x < 0 \\ x^3 + 4x, & x \geq 0 \end{cases}, h(x) = \begin{cases} -x^3, & x < 0 \\ x^5, & x \geq 0 \end{cases}.$$

Остановимся поподробнее на д.с. представлении функции, задающей уравнение. Как известно [107, 110], для любой д.с. функции  $F(x)$  разложение на разность двух выпуклых функций не единственно, более того существует бесконечное число д.с. представлений для любой д.с. функции. Возникает закономерный вопрос, какое из д.с. представлений

является наиболее подходящим, с точки зрения применяемого метода, использующего явное разложение на разность двух выпуклых функций.

В связи с этим рассмотрим некоторые два д.с. представления одной и той же функции  $F(x)$ . Пусть

$$F(x) = g_1(x) - h_1(x), \quad (3.1.21)$$

и одновременно

$$F(x) = g_2(x) - h_2(x), \quad (3.1.22)$$

где  $g_2(x) = g_1(x) + r(x)$ ,  $h_2(x) = h_1(x) + r(x)$ , а  $r(\cdot)$ ,  $g_1(\cdot)$  и  $h_1(\cdot)$  — выпуклые функции. Иными словами, к функциям  $g_1(x)$  и  $h_1(x)$ , участвующим в первом д.с. разложении, добавим одну и ту же выпуклую функцию  $r(x)$ .

В первом случае коэффициент выпуклой комбинации на первом шаге имеет следующий вид:

$$\mu_0^1 := \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle}. \quad (3.1.23)$$

Тогда во втором случае, учитывая выпуклость функции  $r(\cdot)$ , получаем

$$\begin{aligned} \mu_0^2 &:= \frac{F(x^0)}{h_2(u) - h_2(x^0) - \langle \nabla g_2(x^0), u - x^0 \rangle} = \\ &= \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle + (r(u) - r(x^0) - \langle \nabla r(x^0), u - x^0 \rangle)}. \end{aligned}$$

В силу неравенства

$$r(u) - r(x^0) - \langle \nabla r(x^0), u - x^0 \rangle \geq 0$$

получаем

$$\mu_0^2 \leq \frac{F(x^0)}{h_1(u) - h_1(x^0) - \langle \nabla g_1(x^0), u - x^0 \rangle} = \mu_0^1.$$

В частности, при  $r(x) = \|x\|^2$  коэффициент  $\mu_0^2$  находится как

$$\mu_0^2 = \frac{F(x^0)}{\varphi(u, x^0) + (\|u\|^2 - \|x^0\|^2 - 2\langle x^0, u - x^0 \rangle)} = \frac{F(x^0)}{\varphi(u, x^0) + \|u - x^0\|^2}.$$

Таким образом, наличие сильно-выпуклой добавки  $\|x\|^2$  уменьшает коэффициент выпуклой комбинации, а значит, и шаг метода, поскольку  $\|x^1 - x^0\| = \mu_0 \|u - x^0\|$ . Этот факт необходимо учитывать при выборе д.с. представления функции, задающей уравнение, например, выбирая, по возможности, в качестве д.с. декомпозиции минимальную [107].



## 3.2 Доказательство сходимости ПВК

Исследуем свойства последовательности  $\{x^s\}$ , генерируемой этой процедурой при  $\varepsilon = 0$ , т.е. когда итеративный процесс бесконечен.

Поскольку для  $x^0 := v$ ,  $F(x^0) > 0$ , то из (3.1.14) получаем

$$F(x^s) \geq 0, \quad s = 0, 1, 2, \dots \quad (3.2.1)$$

С другой стороны, по построению  $x^0 := v$ ,

$$\begin{aligned} x^1 - x^0 &= \mu_0(u - x^0), \quad 0 < \mu_0 < 1; \\ x^2 - x^1 &= \mu_1(u - x^1) = \mu_1[u - x^0 - \mu_0(u - x^0)] = \mu_1(1 - \mu_0)(u - x^0); \\ x^3 - x^2 &= \mu_2(u - x^2) = \mu_2[u - x^1 - \mu_1(u - x^1)] = \mu_2(1 - \mu_1)(u - x^1) = \\ &= \mu_2(1 - \mu_1)[u - x^0 - \mu_0(u - x^0)] = \mu_2(1 - \mu_1)(1 - \mu_0)(u - x^0) = \\ &= \mu_2(1 - \mu_1)(1 - \mu_0)(u - v); \end{aligned} \quad (3.2.2)$$

Отсюда сразу же следует

$$x^{s+1} - x^s = \mu_s(1 - \mu_{s-1})(1 - \mu_{s-2}) \cdots (1 - \mu_0)(u - v). \quad (3.2.3)$$

Аналогично получаем

$$x^{s+1} - u = x^s + \mu_s(u - x^s) - u = (1 - \mu_s)(1 - \mu_{s-1}) \cdots (1 - \mu_0)(v - u). \quad (3.2.4)$$

Таким образом, поведение последовательности  $\{x^s\}$  зависит от поведения числовых последовательностей  $\{\mu_s\}$  и  $\{1 - \mu_s\}$ . Поэтому вначале исследуем свойства последовательности  $\{\mu_s\}$ .

**Теорема 3.2.1** Числовая последовательность  $\{\mu_s\}$ , строящаяся по правилу (3.1.15), сходится к нулю:  $\lim_{s \rightarrow \infty} \mu_s = 0$ .

**Доказательство.** Предположим противное. Пусть

$$\mu_s \geq \gamma > 0 \quad \forall s = 0, 1, 2, \dots$$

Тогда, поскольку

$$0 < \gamma \leq \mu_s < 1,$$

то

$$1 - \mu_s \leq 1 - \gamma = q < 1 \quad \forall s = 0, 1, 2, \dots$$

Поэтому из (3.2.4) вытекает

$$\|x^{s+1} - u\| = (1 - \mu_s)(1 - \mu_{s-1}) \cdots (1 - \mu_0) \|v - u\| \leq q^{s+1} \|v - u\|.$$

Значит,

$$\lim_{s \rightarrow \infty} \|x^{s+1} - u\| \leq \lim_{s \rightarrow \infty} q^{s+1} \|v - u\| = 0.$$

Это означает, что  $x^s \rightarrow u$ , что невозможно, в силу непрерывности функции  $F(\cdot)$ , поскольку

$$F(x^s) \geq 0 > F(u). \quad (3.2.5)$$

#

**Следствие 3.2.1** *Последовательность  $\{x^s\}$ , генерируемая ПВК, сходится:  $\lim_{s \rightarrow \infty} x^s = x_*$ .*

**Доказательство.** Из равенства (3.2.3) следует

$$\|x^{s+1} - x^s\| = \mu_s(1 - \mu_{s-1}) \cdots (1 - \mu_0) \|u - v\| \leq \mu_s \|u - v\|,$$

поскольку

$$(1 - \mu_i) \leq 1, \quad i = 0, 1, 2, \dots, s - 1.$$

С учетом условия  $\mu_s \downarrow 0$ , получаем

$$\lim_{s \rightarrow \infty} \|x^{s+1} - x^s\| = 0.$$

Следовательно, существует точка  $x_* \in \mathbb{R}^n : x_* = \lim_{s \rightarrow \infty} x^s$ .

#

Итак, последовательность  $\{x^s\}$  сходится к некоторому пределу  $x_*$ . Отсюда по непрерывности  $F(\cdot)$  получаем:

$$\lim_{s \rightarrow \infty} F(x^s) = F(x_*).$$

Введем множество чисел  $\lambda$ , являющихся корнем уравнения (3.1.6):

$$\Lambda_* = \{\lambda \in [0, 1] \mid F(\lambda u + (1 - \lambda)v) = 0\}$$

и множество соответствующих им векторов  $x(\lambda)$ :

$$X_* = \{x \in \mathbb{R}^n \mid \exists \lambda \in \Lambda_* : x = \lambda u + (1 - \lambda)v, F(x) = 0\}.$$

Покажем далее, что последовательность  $\{x^s\}$  сходится к решению уравнения (3.1.1), и при этом алгоритм находит ближайший к  $v$  корень уравнения из множества  $X_*$ .

**Теорема 3.2.2** Предел  $x_*$  последовательности  $\{x^s\}$ , генерируемой ПВК,

- i) является корнем уравнения  $F(x) = 0$ ;
- ii) удовлетворяет соотношению

$$\|x_* - v\| = \min_x \{\|x - v\| \mid x \in X_*\}. \quad (3.2.6)$$

**Доказательство.** i) Поскольку  $F(x^s) \geq 0 \quad \forall s = 0, 1, \dots$ , то, в силу непрерывности функции  $F(\cdot)$ ,  $F(x_*) \geq 0$ . Пусть предел последовательности  $\{x^s\}$  не является корнем уравнения, т.е.  $\exists \eta > 0$ , такое, что

$$\lim_{s \rightarrow \infty} F(x^s) = F(x_*) = \eta > 0.$$

Тогда по построению коэффициента выпуклой комбинации  $\mu_s$  на шаге 1 алгоритма из (3.1.15) получаем

$$h(u) - h(x^s) - \langle \nabla g(x^s), u - x^s \rangle \triangleq \varphi(u, x^s) = \frac{F(x^s)}{\mu_s} \geq \frac{\eta}{\mu_s},$$

откуда в силу теоремы 3.2.1 следует, что

$$\lim_{s \rightarrow \infty} [h(u) - h(x^s) - \langle \nabla g(x^s), u - x^s \rangle] = \lim_{s \rightarrow \infty} \varphi(u, x^s) = +\infty. \quad (3.2.7)$$

С другой стороны, так как  $x^s \rightarrow x_*$ , то по непрерывности функции  $h(\cdot)$ , отображения  $\nabla g(\cdot)$  и скалярного произведения имеем

$$\begin{aligned} \lim_{s \rightarrow \infty} \varphi(u, x^s) &= \lim_{s \rightarrow \infty} [h(u) - h(x^s) - \langle \nabla g(x^s), u - x^s \rangle] = \\ &= h(u) - h(x_*) - \langle \nabla g(x_*), u - x_* \rangle \in \mathbb{R}, \end{aligned}$$

что не совпадает с (3.2.7). Итак, предположение о том, что  $F(x_*) > 0$ , неверно, и тем самым справедливо первое утверждение теоремы.

ii) Покажем теперь справедливость (3.2.6). Пусть  $\bar{x} \in X_*$  — решение уравнения, для которого

$$\|\bar{x} - v\| = \min_x \{\|x - v\| \mid x \in X_*\}.$$

Покажем, что  $x_*$  совпадает с  $\bar{x}$ . С этой целью наряду с последовательностью  $\{\mu_s\}$  будем рассматривать последовательность  $\{\gamma_s\}$  такую, что

$$\bar{x} = \gamma_s u + (1 - \gamma_s) x^s, \quad s = 0, 1, 2, \dots$$

Докажем, что на каждом шаге алгоритма  $\mu_s \leq \gamma_s$ , и точка  $x^s$  оказывается ближе к  $v$ , чем  $\bar{x}$ , т.е. выполнено неравенство

$$\|x^s - v\| \leq \|\bar{x} - v\| \quad \forall s = 0, 1, 2, \dots \quad (3.2.8)$$

Покажем для  $s = 0$ , что  $\mu_0 \leq \gamma_0$ .

В силу выпуклости функций  $g(\cdot)$  и  $h(\cdot)$  справедлива следующая цепочка неравенств:

$$\begin{aligned} g(v) + \gamma_0 \langle \nabla g(v), u - v \rangle &\leq g(\gamma_0 u + (1 - \gamma_0)v) = \\ &= h(\gamma_0 u + (1 - \gamma_0)v) \leq \gamma_0 h(u) + (1 - \gamma_0)h(v). \end{aligned} \quad (3.2.9)$$

Рассмотрим аффинную функцию одной переменной

$$\Psi_0(\gamma) = g(v) + \gamma \langle \nabla g(v), u - v \rangle - \gamma h(u) - (1 - \gamma)h(v),$$

которая представляет собой разность левой и правой частей в (3.1.10). Тогда единственным корнем уравнения  $\Psi_0(\lambda) = 0$  является  $\mu_0$ . В силу (3.2.9)  $\Psi_0(\gamma_0) \leq 0$ .

С другой стороны,

$$\Psi_0(0) = g(v) - h(v) > 0,$$

откуда вытекает, что  $\mu_0 \leq \gamma_0$ . Тогда

$$\|x^1 - v\| = \mu_0 \|u - v\| \leq \gamma_0 \|u - v\| = \|\bar{x} - v\|.$$

Следовательно, корень  $\bar{x}$  может быть также представлен в виде выпуклой комбинации точек  $u$  и  $x_1$ , так что  $\gamma_1 \in [0; 1[$ .

Далее, пусть для  $s = k$  выполнено

$$\mu_{k-1} \leq \gamma_{k-1}, \|x^k - v\| \leq \|\bar{x} - v\|, \gamma_k \in [0; 1[. \quad (3.2.10)$$

Перейдем к  $(k + 1)$ -й итерации. Аналогичным образом для функции

$$\Psi_k(\gamma) = g(x^k) + \gamma \langle \nabla g(x^k), u - x^k \rangle - \gamma h(u) - (1 - \gamma)h(x^k)$$

получаем  $\Psi_k(\mu_k) = 0$ ,  $\Psi_k(0) = g(x^k) - h(x^k) \geq 0$ ,  $\Psi_k(\gamma_k) \leq 0$ , откуда вытекает, что

$$\mu_k \leq \gamma_k. \quad (3.2.11)$$

Далее покажем, что  $\|x^{k+1} - v\| \leq \|\bar{x} - v\|$ . Действительно, из (3.2.4) получаем

$$\begin{aligned} \|x^{k+1} - v\| &= \|(x^{k+1} - u) + (u - v)\| = \\ &= \|(1 - \mu_k)(1 - \mu_{k-1}) \dots (1 - \mu_0)(v - u) + (u - v)\| = \\ &= (1 - (1 - \mu_k)(1 - \mu_{k-1}) \dots (1 - \mu_0)) \|u - v\|. \end{aligned} \quad (3.2.12)$$

С другой стороны,

$$\begin{aligned} \|\bar{x} - v\| &= \|(\gamma_k u + (1 - \gamma_k)v - u) + (u - v)\| = \|(1 - \gamma_k)(x_k - u) + (u - v)\| = \\ &= (1 - (1 - \gamma_k)(1 - \mu_{k-1})(1 - \mu_{k-2}) \dots (1 - \mu_0)) \|u - v\|. \end{aligned} \quad (3.2.13)$$

С учетом (3.2.11) из (3.2.12) и (3.2.13) следует, что  $\|x^{k+1} - v\| \leq \|\bar{x} - v\|$ .

Таким образом, справедливо (3.2.8).

Переходя в (3.2.8) к пределу при  $k \rightarrow \infty$ , получаем

$$\|x_* - v\| \leq \|\bar{x} - v\| = \min_x \{\|x - v\| \mid x \in X_*\},$$

откуда вытекает утверждение теоремы (3.2.6). #

Как уже говорилось во введении, в одномерном случае задача поиска специального решения, удовлетворяющего (3.2.6), может быть проинтерпретирована как задача нахождения минимального корня уравнения. Приведем пример подобной задачи, возникающий на практике [62].

Пусть требуется определить время корректной работы некоторого устройства на временном интервале  $[t_0, t_1]$ . При этом характеристика его корректного функционирования есть функция  $f(t)$ ,  $f(t_0) > 0$ . Устройство работает правильно в момент времени  $t$ , если  $f(t) > 0$ . Необходимо определить момент времени  $t^*$  такой, что

$$f(t^*) = 0, \quad f(t) > 0, \quad t \in [t_0, t^*[, \quad t^* \in ]t_0, t_1].$$

В [62] предложен подход для решения такой задачи, использующий технику построения вспомогательных функций для аппроксимации функции  $f$  с различными вариантами оценки константы Липшица.

В [85] также разработан метод решения д.с. уравнения с помощью построения вогнутых опорных функций, позволяющий находить корень, ближайший к  $v$ . На каждой итерации этого метода производится линеаризация функции  $g(\cdot)$  и ищется корень  $x^{k+1}$  уравнения

$$\varphi(x, x^k) \triangleq g(x^k) + p^k(x - x^k) - h(x) = 0, \quad (3.2.14)$$

где  $p^k \in \partial g(x^k)$ ,  $x^k$  — точка, полученная на предыдущей итерации.

Очевидно, что при  $h(\cdot) \equiv 0$  данный метод, как и ПВК, совпадает с методом Ньютона. Рассмотрим взаимосвязь этих методов в общем случае.

Заметим, что, если в методе из [85] вместо нахождения точного решения уравнения (3.2.14) произвести всего одну итерацию метода хорд, то получим:

$$\mu\varphi(u, x^k) + (1 - \mu)\varphi(x^k, x^k) = 0$$

или

$$g(x^k) + \mu p^k(u - x^k) = \mu h(u) + (1 - \mu)h(x^k),$$

откуда при  $p^k = \nabla g(x^k)$  получаем коэффициент  $\mu$  для вычисления следующего приближения в ПВК.

Таким образом, в ПВК вместо решения уравнения (3.2.14) каким-либо численным методом осуществляется всего один шаг метода хорд, что, тем не менее, оказывается достаточным для сходимости к корню уравнения. В этом случае отпадает необходимость решать уравнение (3.2.14) с высокой точностью, т.е. уменьшается трудоемкость каждой итерации.

### 3.3 Вычислительный эксперимент

Цель вычислительного эксперимента состояла в проверке работоспособности предложенного метода решения уравнений с d.c. функцией и сравнении данного метода с некоторыми другими, гарантирующими нахождение корня на заданном интервале. Для проведения сравнения были выбраны модифицированный метод хорд (ММХ) и метод половинного деления.

Модифицированный метод хорд позволяет все время оставаться внутри рассматриваемого отрезка, переопределяя границы интервала, содержащего корень. Алгоритм ММХ представлен ниже.

#### Модифицированный метод хорд

**Шаг 0.** Положить  $s := 0$ ,  $v^s := v$ ,  $u^s := u$ .

**Шаг 1.** Вычислить

$$\mu_s := \frac{F(v^s)}{F(v^s) - F(u^s)}. \quad (3.3.1)$$

**Шаг 2.** Составить выпуклую комбинацию точек  $u^s$ ,  $v^s$ :

$$x^s := \mu_s u^s + (1 - \mu_s) v^s = v^s + \mu_s (u^s - v^s). \quad (3.3.2)$$

**Шаг 3.** Если  $|F(x^s)| \leq \varepsilon$ , то  $x^s$  —  $\varepsilon$ -решение уравнения  $F(x) = 0$ .

**Шаг 4.** Если  $F(x^s) < 0$ , то положить  $s := s + 1$ ,

$$u^s := x^{s-1}, \quad v^s := v^{s-1}, \quad (3.3.3)$$

перейти на шаг 1.

**Шаг 5.** Если  $F(x^s) > 0$ , то положить  $s := s + 1$ ,

$$u^s := u^{s-1}, \quad v^s := x^{s-1}, \quad (3.3.4)$$

перейти на шаг 1.

#

Здесь, как и в ПВК на каждой итерации строится выпуклая комбинация точек, в которых функция  $F$  имеет противоположные знаки. Однако знак  $F$  в получаемой точке определить заранее (как это удалось для ПВК) нельзя. Поэтому, в отличие от ПВК, в котором последовательность  $\{x^s\}$  приближается к решению с одной стороны, здесь строятся две последовательности  $\{u^s\}$  и  $\{v^s\}$ .

Вначале тестирование проводилось на д.с. уравнениях с одной переменной. Данные о тестовых уравнениях содержатся в табл. 3.1.

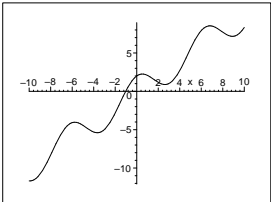
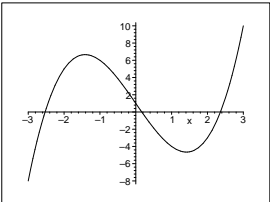
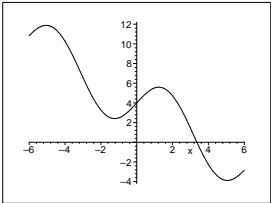
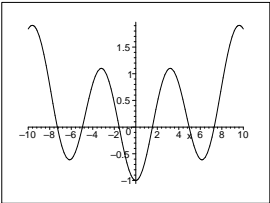
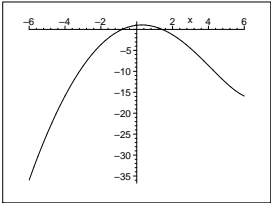
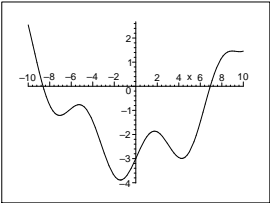
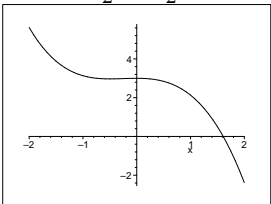
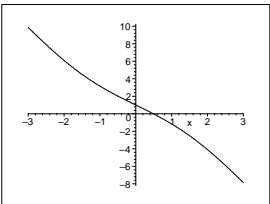
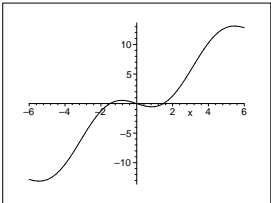
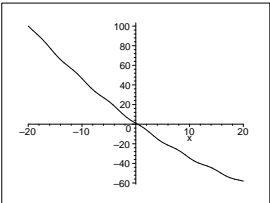
№	Уравнение	№	Уравнение
1	$2 \cos x + x = 0$ 	6	$x^3 - 6x + 1 = 0$ 
2	$3 \sin x - x + 4 = 0$ 	7	$0.01x^2 - \cos x = 0$ 
3	$e^{\frac{x}{2}} - x^2 = 0$ 	8	$0.05x^2 + \sin x = 0$ 
4	$3 \cos \frac{x}{2} - \frac{x^3}{2} = 0$ 	9	$\sin x - 3x + 1 = 0$ 
5	$2x - 3 \sin x = 0$ 	10	$\sin x - 4x + 0.05x^2 + 1 = 0$ 

Таблица 3.1: Тестовые уравнения

В табл. 3.2 включены результаты тестирования данных уравнений. Здесь  $N$  — номер уравнения, далее приведены отрезок, содержащий корень уравнения, и количества итераций для ПВК, модифицированного метода Хорд и метода половинного деления. Сравнение осуществлялось в среде MatLab r2009a. Жирным шрифтом выделены минимальные для соответствующей задачи количества итераций, требуемые для нахождения корня. Точность решения уравнения  $\varepsilon$  выбиралась равной  $10^{-7}$ .

Как можно видеть из таблицы, ПВК выигрывает по количеству итераций во всех случаях. Однако необходимо учитывать, что трудоемкость одной итерации у предлагаемого метода выше, поскольку необходимо вычислять производную функции  $g$ . Тем не менее на данных примерах все методы показали сравнимую по времени производительность.

$N$	Интервал	ПВК	М.М.Хорд	М.Бисекции
1	[-5;0]	<b>4</b>	8	26
2	[0;5]	<b>5</b>	<b>5</b>	27
3	[1;2]	<b>7</b>	10	16
	[-5;0]	<b>8</b>	43	19
4	[1;2]	<b>4</b>	11	23
5	[1;2]	<b>4</b>	14	23
	[0.3;1.8]	<b>3</b>	14	24
6	[2;3]	<b>5</b>	16	25
	[-1;1]	<b>3</b>	4	25
7	[0;2]	<b>3</b>	4	23
	[3;6]	<b>4</b>	5	24
	[6;10]	<b>4</b>	6	24
8	[6;8]	<b>6</b>	7	27
9	[0;1]	<b>3</b>	6	23
10	[0;2]	<b>5</b>	9	21

Таблица 3.2: Решение одномерных уравнений

Если же увеличивать интервал, содержащий корень (см. табл. 3.3), то ситуация меняется. Как и можно было предполагать, в этом случае более стабильно работает метод бисекции, методы же хорд и ПВК, хотя и выигрывают в отдельных случаях, требуют очень большого количества итераций в некоторых других (см., например, задачу 4).



$N$	Интервал	CoComba	М.М.Хорд	М.Бисекции
1	[-10;10]	33	<b>9</b>	28
2	[10;-10]	30	<b>6</b>	29
3	[0;6]	<b>12</b>	25	19
	[-10;0]	<b>9</b>	86	20
4	[-10;10]	170	222	<b>25</b>
5	[1;10]	<b>7</b>	15	26
6	[1;10]	<b>8</b>	221	29
7	[-6;10]	39	<b>6</b>	25
8	[-5;15]	32	<b>11</b>	26
9	[-10;10]	27	<b>15</b>	26
10	[-10;10]	23	<b>17</b>	27

Таблица 3.3: Увеличение интервала, содержащего корень

На задачах большой размерности наблюдалась похожая ситуация, когда метод бисекции оказывается более быстрым, но, напомним, он не выполняет дополнительной функции, т.е. не находит ближайший к  $v$  корень, что важно в случае поиска допустимой точки в задаче с д.с. ограничением. Тем не менее, можно построить примеры уравнений, когда даже на задачах высокой размерности ПВК оказывается наиболее эффективной среди сравниваемых методов.

В табл. 3.4 представлены результаты решения уравнения  $\sin(x_1 + x_2 + \dots + x_n) = 0$ . В таблице использовались следующие обозначения:  $n$  — количество переменных,  $u_i, v_i$  — компоненты векторов  $u$  и  $v$  соответственно,  $It$  — количество итераций,  $T_{1000}$  — время решения 1000 одинаковых задач (поскольку время решения одной задачи слишком мало).

Здесь независимо от роста размерности ПВК стабильно проходит около четырех итераций, лидируя среди остальных методов как по количеству итераций, так и по времени счета.

В заключение приведем пример из [85], на котором осуществлялось сравнение ПВК с методом вогнутых опорных функций.

Требуется найти корень уравнения  $f(x) \triangleq -\sin(x) - \sin(3x + 1) + 1.5 = 0$ . В качестве интервала, содержащего корень, был выбран отрезок  $[5, 8.5]$ .

Методу вогнутых опорных функций потребовалось 13 итераций для нахождения ре-

$n$	$u_i$	$v_i$	$\varepsilon$	ПВК		М.Хорд		М.Бисекции	
				$It$	$T_{1000}$	$It$	$T_{1000}$	$It$	$T_{1000}$
10	$\frac{\pi}{2n}$	1	0.001	3	0.09	4	<b>0.06</b>	12	0.14
			0.00001	5	0.16	<b>5</b>	<b>0.07</b>	19	0.19
100	$\frac{\pi}{2n}$	1.1	0.001	<b>2</b>	<b>0.09</b>	8	0.11	14	0.18
			0.00001	4	0.16	9	0.13	23	0.21
500	$\frac{\pi}{2n}$	1	0.001	<b>2</b>	<b>0.13</b>	11	0.24	18	0.39
			0.00001	<b>3</b>	<b>0.18</b>	12	0.27	24	0.51
1000	1	$-\frac{\pi}{2n}$	0.001	<b>3</b>	<b>0.23</b>	13	0.43	17	0.52
			0.00001	<b>4</b>	<b>0.31</b>	14	0.45	25	0.78

Таблица 3.4: Решение уравнения  $f(x) \triangleq \sin(x_1 + x_2 + \dots + x_n) = 0$

шения с заданной точностью  $10^{-7}$ . ПВК затратила 48 итераций. Однако трудоемкость итераций в этих методах различается: помимо вычисления значений функций  $h$  и  $g'$  на каждой итерации в первом случае осуществляется запуск некоторого алгоритма решения уравнения (3.2.14) (например, метода половинного деления).

Таким образом, ПВК показала сравнительную эффективность на рассмотренных тестовых примерах. Поскольку время, затрачиваемое алгоритмом, оказывается приемлемым, и при этом учитывается специфика оптимизационной задачи, то в задачах минимизации с d.c. равенствами предлагается использовать предложенный метод для поиска допустимых точек.

### 3.4 Решение систем нелинейных уравнений

Далее будем рассматривать системы уравнений с d.c. функциями, необходимость в решении которых может возникнуть как в задаче оптимизации с несколькими ограничениями-равенствами, так и в виде отдельной постановки [25, 45].

Рассматривается следующая задача:

$$f_i(x) = 0, \quad i = 1, \dots, m, \quad (3.4.1)$$

где  $x \in \mathbb{R}^n$ , а  $f_i(x)$  — d.c. функции, т.е.

$$f_i(x) = g_i(x) - h_i(x), \quad (3.4.2)$$

где  $g_i(x), h_i(x)$  — выпуклые на  $\mathbb{R}^n$  функции,  $i = 1, \dots, m$ . Как известно [4, 12], система уравнений (3.4.1) может быть сведена к задаче минимизации

$$F(x) = \Phi(f_1(x), \dots, f_m(x)) \downarrow \min, \quad x \in \mathbb{R}^n, \quad (3.4.3)$$

где в качестве целевой могут выступать, например, следующие функции:

$$F(x) = \sum_{i=1}^m |f_i(x)|, \quad (3.4.4)$$

$$F(x) = \sum_{i=1}^m (f_i(x))^2. \quad (3.4.5)$$

Нетрудно видеть, что задача (3.4.3), ввиду невыпуклости  $f_i(\cdot)$  (см. представление (3.4.2)), оказывается в общем случае невыпуклой.

Используя известные свойства d.c. функций [69, 137], нетрудно видеть, что, например, для функции (3.4.4) справедливо следующее представление:

$$F(x) = G(x) - H(x), \quad (3.4.6)$$

где

$$G(x) = 2 \sum_{i=1}^m \max\{g_i(x), h_i(x)\},$$

$$H(x) = \sum_{i=1}^m (g_i(x) + h_i(x))$$

— выпуклые функции. Значит, функция  $F(\cdot)$ , определяемая формулой (3.4.4), является d.c. функцией. Таким образом, для решения задачи

$$F(x) = G(x) - H(x) \downarrow \min, \quad x \in \mathbb{R}^n, \quad (3.4.7)$$

являющейся эквивалентной системе уравнений (3.4.1), может быть использован аппарат d.c. минимизации [69].

Напомним (см. §§1.3, 1.5), что общая процедура глобального поиска минимума d.c. функции состоит из двух основных этапов: локального поиска и процедуры выхода из критической точки, основанной на условиях глобальной оптимальности, с последующим повторным применением локального поиска.

Метод локального поиска для задачи d.c. минимизации (см. §1.3) состоит из последовательного решения задач

$$G(x) - \langle \nabla H(x^s), x \rangle \downarrow \min_x, \quad x \in \mathbb{R}^n. \quad (3.4.8)$$

Зная  $x^s$ , можно найти  $x^{s+1}$  как приближенное решение (3.4.8). Можно показать (см. §1.3), что процесс в этом случае сходится к решению линеаризованной задачи

$$G(x) - \langle \nabla H(x_*), x \rangle \downarrow \min_x, \quad x \in \mathbb{R}^n. \quad (3.4.9)$$

Точка  $x_*$  в этом случае называется критической точкой в задаче (3.4.7).

В случае негладкости функции  $G$  в д.с. представлении (3.4.6) для нахождения минимума в задаче (3.4.8) необходимо применять методы негладкой оптимизации. В частности, для данной задачи использовался известный г-алгоритм Н.З. Шора [41, 87].

Далее, с использованием условий глобальной оптимальности [69] задача (3.4.7) декомпозируется на ряд более простых задач: одномерный поиск, построение аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в задаче, проверка точек аппроксимации на пригодность к дальнейшим исследованиям и локальный поиск.

В результате можно предложить следующий алгоритм глобального поиска (АГП) (см. также §1.5).

Пусть заданы начальная точка  $x_0 \in \mathbb{R}^n$  и числовые последовательности  $\{\tau_k\}$  и  $\{\delta_k\}$ :  $\tau_k, \delta_k > 0$ ,  $k = 0, 1, 2, \dots$ ,  $\tau_k \downarrow 0$ ,  $\delta_k \downarrow 0$  ( $k \rightarrow \infty$ ).

### Алгоритм глобального поиска (АГП)

**Шаг 0.** Положить  $k := 0$ ,  $x^k := x_0$ .

**Шаг 1.** Начиная с  $x^k \in \mathbb{R}^n$  посредством специального метода локального поиска построить  $\tau_k$ -критическую точку  $z^k$ ,  $\xi_k := F(z^k) \leq F(x^k)$ .

**Шаг 2.** Если  $F(z^k) < \chi$ , где  $\chi$  — заданная точность, то Stop,  $z^k$  — приближенное решение системы уравнений.

**Шаг 3.** Выбрать некоторое  $\beta \in [\beta_-, \beta_+]$ .

**Шаг 4.** Построить аппроксимацию

$$\mathcal{A}_k(\beta) = \{v^1, \dots, v^N \mid H(v^i) = \beta - \xi_k, \quad i = 1, \dots, N, \quad N = N(\beta)\}. \quad (3.4.10)$$

**Шаг 5.** Для каждого  $i = 1, \dots, N$  отыскать  $u^i \in \mathbb{R}^n$  такое, что

$$G(u^i) - \langle \nabla H(v^i), u^i \rangle - \delta_k \leq \inf_x \{G(x) - \langle \nabla H(v^i), x \rangle \mid x \in \mathbb{R}^n\}. \quad (3.4.11)$$

**Шаг 6.** Для каждого  $i = 1, \dots, N$  найти  $w^i$ , для которого

$$\langle \nabla H(w^i), u^i - w^i \rangle + \delta_k \geq \sup_v \{ \langle \nabla H(v), u^i - v \rangle \mid H(v) = \beta - \xi_k \}. \quad (3.4.12)$$

**Шаг 7.** Положить  $\eta_k(\beta) := \eta_k^0(\beta) + \beta$ , где

$$\eta_k^0(\beta) := \langle \nabla H(w^j), w^j - w^j \rangle - G(w^j) \triangleq \max_{1 \leq i \leq N} \{ \langle \nabla H(w^i), w^i - w^i \rangle - G(w^i) \}. \quad (3.4.13)$$

**Шаг 8.** Если  $\eta_k(\beta) > 0$ , то положить  $x^{k+1} := w^j$  и вернуться на шаг 1.

**Шаг 9.** Если  $\eta_k(\beta) \leq 0$ , то положить  $\beta := \beta + \Delta\beta \in [\beta_-, \beta_+]$  и идти на шаг 3.

**Шаг 10.** Если  $\eta_k(\beta) \leq 0 \quad \forall \beta \in [\beta_-, \beta_+]$  (т.е. одномерный поиск по  $\beta$  завершен), то положить  $k := k + 1$ ,  $x^{k+1} := z^k$  и вернуться на шаг 1. #

Более кратко схема алгоритма представлена на рис. 3.4.



Рис. 3.4: Схема алгоритма глобального поиска

Отметим, что, в отличие от § 1.5, здесь представлен иной вариант алгоритма глобального поиска [69], в котором, в соответствии с УГО, решается задача уровня на шаге 6 и формируется величина  $\eta_k(\beta)$ , с помощью которой непосредственно проверяется выполнение неравенства из УГО (см. (1.5.1)). В первой главе вычисление  $\eta_k(\beta)$  было заменено на сравнение целевой функции в полученных точках с текущим значением  $\xi_k$ .

Наибольшую трудность в предложенном методе вызывают шаги 4, 5 и 6, на которых возникает необходимость построить аппроксимацию поверхности уровня функции  $H$ , решить линейаризованную задачу:

$$G(x) - \langle \nabla H(v^i), x \rangle \downarrow \min_x, \quad (3.4.14)$$

и задачу уровня:

$$\langle \nabla H(v), u^i - v \rangle \uparrow \max_v, \quad H(v) = \beta - \zeta_k. \quad (3.4.15)$$

Можно заметить, что в силу представления (3.4.4) линеаризованная функция оказывается негладкой, поэтому для нахождения ее минимума необходимо воспользоваться методами негладкой минимизации [41, 87]. Задача уровня (3.4.12), как будет показано далее, в некоторых случаях имеет аналитическое решение. В случае же, когда решение задачи (3.4.12) представляет большую вычислительную сложность (в общем случае она может быть невыпуклой), ее решение и вычисление величины  $\eta_k(\beta)$  заменяется прямым сравнением значения целевой функции в полученной точке со значением  $\zeta_k$  (как это было в первой главе).

Для вычисления нижней границы интервала одномерного поиска  $[\beta_-, \beta_+]$  решается задача выпуклой негладкой минимизации. Поскольку функция в данном случае не ограничена сверху на  $\mathbb{R}^n$ , то верхняя граница  $\beta_+$  вводится искусственным образом. Для нахождения же  $\beta_-$  можно использовать методы выпуклой минимизации.

### 3.5 Особенности решения квадратичных систем уравнений

Для непосредственной реализации метода глобального поиска вначале было рассмотрено решение систем квадратичных уравнений следующего вида:

$$f_i(x) = \frac{1}{2} \langle C_i x, x \rangle + \langle b_i, x \rangle + d_i = 0, \quad i = 1, \dots, n, \quad (3.5.16)$$

где  $x \in \mathbb{R}^n$ ,  $C_i$  — симметричные, не обязательно положительно определенные  $(n \times n)$ -матрицы. Рассмотрим по порядку основные моменты реализации стратегии глобального поиска для этого случая.

**а) Д.С. представление.** Как было показано выше (см. §1.2), знаконеопределенную, но симметричную матрицу  $C_i$  можно представить в виде разности двух симметричных положительно определенных матриц  $C_i = A_i - B_i$ . Это влечет следующее д.с. представление квадратичных функций:

$$f_i(x) = \frac{1}{2} \langle A_i x, x \rangle + \langle b_i, x \rangle + d_i - \left[ \frac{1}{2} \langle B_i x, x \rangle \right] \triangleq g_i(x) - h_i(x), \quad i = 1, \dots, n, \quad (3.5.17)$$

где  $g_i(x) = \frac{1}{2} \langle A_i x, x \rangle + \langle b_i, x \rangle + d_i$ ,  $h_i(x) = \frac{1}{2} \langle B_i x, x \rangle$ .

С использованием (3.5.17) и д.с. представления модуля (см.(3.4.6)) для целевой функции  $F(x)$  получаем

$$G(x) = \sum_{i=1}^n \max\{\langle A_i x, x \rangle + \langle b_i, x \rangle + d_i ; \langle B_i x, x \rangle - \langle b_i, x \rangle - d_i\}, \quad (3.5.18)$$

$$H(x) = \frac{1}{2} \langle Sx, x \rangle, \quad (3.5.19)$$

где  $S = \sum_{i=1}^n (A_i + B_i)$  — симметричная, положительно определенная  $(n \times n)$ -матрица.

Заметим, что при использовании общего д.с. разложения (3.4.6) в состав  $H(x)$  должна еще входить аффинная добавка  $\sum_{i=1}^n \langle b_i, x \rangle + d_i$ . Однако структура функции  $H$  должна быть максимально простой, поскольку она задает базовую невыпуклость в задаче, по ней происходит линейризация и строится аппроксимация поверхности уровня. Поэтому данные слагаемые были "переброшены" в  $G(x)$ , что привело к разложению (3.5.18)–(3.5.19).

**б) Решение задачи уровня.** Остановимся теперь на одном важном этапе решения невыпуклых задач с помощью АГП — решении задачи уровня.

Учитывая (3.5.19) получаем:

$$H(v) = \frac{1}{2} \langle Sv, v \rangle,$$

$$\nabla H(v) = Sv,$$

где  $S$  — симметричная положительно определенная матрица.

Поэтому задача уровня записывается следующим образом:

$$\left. \begin{aligned} J(v) &\triangleq \langle Sv, u - v \rangle \uparrow \max_v, \\ \frac{1}{2} \langle Sv, v \rangle &= \beta - \xi. \end{aligned} \right\} \quad (3.5.20)$$

Нетрудно показать, что задача (3.5.20) имеет аналитическое решение [69]:

$$w = tu, \quad t = \left( \frac{2(\beta - \xi)}{\langle Su, u \rangle} \right)^{\frac{1}{2}}.$$

**с) Построение аппроксимации поверхности уровня функции  $H$ .**

В алгоритме глобального поиска остается неопределенным выбор некоторой аппроксимации  $\mathcal{A}_k(\beta)$ .

Для построения аппроксимации, исходя из опыта ранее решенных задач [18, 19, 44, 69, 80] (см. также §1.5, §2.9.1), был предложен следующий способ. В качестве точек аппроксимации были выбраны

$$y^i = z^k - \mu_i r^i,$$

где  $r^i$  — некоторый вектор, который, вообще говоря, строится на основе данных о задаче,  $\mu_i$  — число, которое находится из условия

$$H(y^i) = \frac{1}{2} \langle Sy^i, y^i \rangle = \beta - \xi.$$

В этом случае набор точек аппроксимации зависит от текущей критической точки  $z^k$  и изменяется от итерации к итерации. Были рассмотрены следующие четыре аппроксимации.

$$\mathcal{R}_1 = \{y^i = z^k - \mu_i e^i \mid i = 1, \dots, n\},$$

где  $e^i = (0, \dots, 1, \dots, 0)$  — единичные векторы,

$$\mathcal{R}_2 = \{y^i = z^k - \mu_i s^i \mid i = 1, \dots, n\},$$

где вектор  $s^i$  — столбцы матрицы  $S$ . Такое задание вектора  $p^i$  отражает структуру функции  $H(x)$ .

Следующая аппроксимация построена в процессе численного эксперимента

$$\mathcal{R}_3 = \{y^i = z^k - \mu_i c^i \mid i = 1, \dots, n\},$$

где  $c_i = (\overbrace{-1, \dots, -1}^i, 1, 1, \dots, 1)$ . Здесь в  $i$ -й точке аппроксимации первые  $i$  компонент составляют  $-1$ , а оставшиеся равны  $1$ .

Предварительный вычислительный эксперимент на задачах малой размерности показал, что ни одна из аппроксимаций  $\mathcal{R}_1$ ,  $\mathcal{R}_2$ ,  $\mathcal{R}_3$  по отдельности не нашла решения для всех предложенных систем, однако совместное использование всех трех аппроксимаций позволило решить все задачи.

Поэтому в рассмотрение была введена следующая аппроксимация:

$$\mathcal{R}_{123} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3.$$

**d) Выбор шага  $\Delta\beta$ .** Для одномерной минимизации по  $\beta$  было использовано разбиение отрезка  $[\beta_-, \beta_+]$  с постоянным шагом, зависящим от размерности задачи: шаг  $\Delta\beta$  брался равным  $\frac{n^2}{10}$ . Когда значение  $F(z^k)$  становилось меньше  $0.1$ , шаг уменьшался в  $5$  раз, поскольку предварительный вычислительный эксперимент показал, что в случае, когда значение  $F$  близко к нулю, при слишком грубом разбиении по  $\beta$  не удастся улучшить значение целевой функции.

Следующие параграфы посвящены тестированию АГП на некоторых известных из литературы примерах.



## 3.6 Численное решение систем уравнений

### 3.6.1 Квадратичные уравнения

В качестве тестов были взяты примеры из набора тестовых систем нелинейных уравнений [61], решения которых заранее известны (индекс  $i$ , если не указано противное, изменяется от 1 до  $n$ ):

1)  $f_i = (3 - 2x_i)x_i + 1 - x_{i-1} - 2x_{i+1}$ ,  $x_0 = x_{n+1} = 0$ ;

2)  $f_i = 3x_i - x_{i-1} - 2x_{i+1} - kx_i^2 + 1$ ,  $x_0 = x_{n+1} = 0$ ;

3)  $f_1 = 1 - x_1$ ,  $f_i = 10(i - 1)(x_i - x_{i-1})^2$ ,  $i = 2, 3, \dots, n$ ;

4)  $f_i = 1 - x_i$ ,  $i = 1, 3, 5, \dots, n - 1$ ,  $f_i = 10(x_i - x_{i-1}^2)$ ,  $i = 2, 4, \dots, n$  ( $n$  — четное);

5)  $f_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + \frac{(x_{i+1} - x_{i-1})^2}{4}$ ,  $x_0 = 0$ ;  $x_{n+1} = 20$ .

Результаты численного решения этих систем, представленные в табл. 3.5, получены с помощью PC Intel Pentium 4, 3.0 ГГц, программа написана на языке C++. Точность по значению целевого функционала бралась равной 0.001. Если алгоритм заканчивал работу, достигая этой точности, то в таблице в столбце  $F_*$  стоит 0.

В табл. 3.5 и далее используются следующие обозначения:  $N$  — номер системы,  $n$  — размерность системы,  $x_0$  — начальное приближение задачи,  $F_0$  — начальное значения целевой функции  $F(x)$ ,  $F_*$  — полученное значение функции,  $St$  — количество итераций,  $PL$  — количество решенных линеаризованных задач,  $T$  — время работы алгоритма (мин:сек). Для сравнения в таблице приведены результаты работы Matlab-солвера STRSCNE [91], предназначенного для решения систем нелинейных уравнений. В колонке  $F_{STR}$  приведено значение целевой функции  $F$ , полученное в результате работы данного солвера, запущенного из стартовой точки  $x^0$ .

Отметим, что в первых двух системах солверу STRSCNE не удалось получить решение из предложенной стартовой точки, в то время как АГП по всех случаях получил глобальное решение.

№	$x_0$	$n$	$F_0$	$F_*$	$St$	$PL$	$T$	$F_{STR}$
1	(1, -1, ..., 1, -1)	10	57	0	2	29	00:00.00	2.5630
		20	117	0	2	46	0:05.12	2.6924
		30	177	0	2	251	01:32.30	2.6962
		40	237	0	2	302	03:04.30	2.6962
2	(1, ..., 1)	10	12	0	6	72	00:00.12	0
		20	21	0	10	452	00:00.64	0
		30	30	0	16	532	00:31.48	7.2293
		40	39	0	18	724	01:37.15	7.1053
3	(2, -2, ..., 2, -2)	10	7201	0	10	352	00:01.14	0
		20	30400	0	18	756	00:26.18	0
		30	69600	0	20	1443	07:28.42	0
		40	124800	0	22	1612	10:02.34	0
4	(0, ..., 0)	10	5	0	38	4874	00:08.04	0
		20	10	0	54	6234	00:56.46	0
		30	20	0	74	10562	02:15.22	0
5	(10, ..., 10)	10	600	0	10	412	00:01.04	0
		20	600	0	21	1205	01:58.52	0

Таблица 3.5: Решение систем квадратичных уравнений

### 3.6.2 Нелинейные уравнения

После тестирования предложенного метода на системах квадратичных уравнений интересной показалась проверка эффективности алгоритма глобального поиска на системах другой структуры ( $i = 1, \dots, n$ ):

6)  $f_i = 2x_i - x_{i-1} - x_{i+1} + \frac{\nu^2}{2}(x_i + t_i + 1)^3$ , где  $\nu = \frac{1}{n+1}$ ,  $t_i = i\nu$ ,  $x_0 = x_{n+1} = 0$ ;

7)  $f_i = x_{i-1} - 2x_i + x_{i+1} - \nu^2 \cdot e^{x_i}$ , где  $\nu = \frac{1}{n+1}$ ,  $x_0 = x_{n+1} = 0$ ;

8)  $f_i = x_i - \frac{1}{2n}(\sum_{j=1}^n x_j^3 + i)$ .

Для этих функций также несложно получить д.с. разложение, учитывая, что

$$xy = \frac{1}{2}(x+y)^2 - \frac{1}{2}(x^2 + y^2),$$

$$x^3 = \frac{1}{4}[(x^2 + (x+1)^2)^2 + 1] - \frac{1}{4}[(x+1)^4 + 2x^4 + (x^2 + 1)^2].$$

Однако в этом случае затрудняется выбор точек аппроксимации поверхности уровня, поскольку не удастся найти (аналитически) точки, удовлетворяющие условиям

$$y^i = z^k - \mu_i r^i, \quad H(y^i) = \beta - \xi.$$

Поэтому коэффициент  $\mu_i$  находился приближенно. В качестве аппроксимации был выбран набор  $\mathcal{R}_1$ . В табл. 3.6 приведены результаты решения систем 6–8. Время решения

ограничивалось 10 минутами. В то время как с помощью решателя STRSCNE системы 6-8 решены с необходимой степенью точности (за исключением системы размерности 70), алгоритму глобального поиска в трех случаях (см. табл. 3.6) не удалось найти решение с заданной точностью.

№	$x_0$	$n$	$F_0$	$F^*$	$St$	$PL$	$T$	$F_{STR}$
6	$(1, -1, \dots, 1, -1)$	10	38.3	0.01229	2	167	10:00.00	0
		20	78.2	0.07898	2	214	10:00.00	0
7	$(1, \dots, 1)$	10	38.01	0	1	9	00:01.31	0
		20	78.1	0.04537	1	173	10:00.00	0
8	$(2, -2, \dots, 2, -2)$	10	4.6	0	2	12	00:00.43	0
		20	9	0	3	21	00:02.64	0
		30	13.37	0	3	24	00:18.62	0
		40	17.75	0	2	28	00:20.84	0
		50	22.15	0	3	42	00:16.18	0
		60	26.5	0	3	47	00:53.78	0
		70	30.87	0	3	58	01:56.35	0.017

Таблица 3.6: Решение систем уравнений

Значение целевой функции при этом уменьшилось достаточно, что позволяет использовать полученную в результате работы АГП точку в качестве стартовой для стандартных численных методов решения систем. Таким образом, за приемлемое время решены системы повышенной размерности, что подтверждает возможность применения новой методики решения систем нелинейных уравнений.

### 3.7 Основные результаты главы

В данной главе получены следующие результаты.

1. Предложен и обоснован специальный метод выпуклой комбинации для решения уравнения со многими неизвестными. Доказана теорема сходимости предложенного метода. Произведено численное сравнение с некоторыми другими методами решения.
2. Для решения систем нелинейных уравнений, задаваемых д.с. функциями, разработан алгоритм глобального поиска, показавший эффективность на ряде тестовых примеров.

# Заключение

Основные результаты диссертационной работы, выносимые на защиту, заключаются в следующем.

1. Для решения линейных задач дополненности со знаконеопределенными матрицами предложен и обоснован алгоритм, основанный на теории глобального поиска для задач минимизации d.c. функций (функций, представимых в виде разности двух выпуклых). Проведен вычислительный эксперимент по сравнению разработанного алгоритма с существующими подходами, показавший эффективность нового алгоритма на широком спектре тестовых примеров различной размерности.
2. Для линейных двухуровневых задач разработаны алгоритмы локального и глобального поисков оптимистических решений, позволяющие решать задачи высокой размерности (до 500 переменных на каждом уровне). Предложенные алгоритмы базируются на взаимосвязях линейных двухуровневых задач и невыпуклых задач математического программирования с нелинейными ограничениями-равенствами, а также теории глобального поиска в задачах с d.c. ограничениями.
3. Разработан и обоснован специальный метод решения d.c. уравнений со многими переменными. Доказана теорема сходимости предложенного метода, проведен численный эксперимент, показавший возможность применения данного метода для поиска допустимых по ограничению-равенству точек. Разработан алгоритм решения систем d.c. уравнений, основанный на теории глобального поиска для негладких задач d.c. минимизации, подтвердивший свою эффективность на серии тестовых примеров.

# Литература

- [1] Александров А.Д. О поверхностях, представимых разностью выпуклых функций / А.Д. Александров // Изв. АН КазССР. Сер. Матем. и механика — 1949. — № 3. — С. 3–20.
- [2] Амосов А.А. Вычислительные методы для инженеров / А.А. Амосов, Ю.А. Дубинский, Н.В. Копченова — М.: Мир, 1998.
- [3] Базара М. Нелинейное программирование. Теория и алгоритмы / М. Базара, К. Шетти — М.: Мир, 1982.
- [4] Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков — М.: Наука, 1987.
- [5] Бертсекас Д. Условная оптимизация и методы множителей Лагранжа / Д. Бертсекас — М.: "Радио и связь", 1978.
- [6] Булатов В.П. Численные методы поиска всех вещественных корней систем нелинейных уравнений / В.П. Булатов // ЖВМ и МФ. — 2000. — Т.40, №3. — С. 348–355.
- [7] Булатов В.П. Глобальная оптимизация и методы нахождения всех корней систем нелинейных алгебраических уравнений / В.П. Булатов, Т.И. Белых // Дискретн. анализ и исслед. опер. — 2006. — Т.13, №1. — С. 3–9.
- [8] Булатов М.В. О свойствах конечномерных систем нелинейных уравнений с кратными решениями / В.П. Булатов, В.Ф. Чистяков // Труды XIV Байкальской международной школы-семинара. Том 3. — Иркутск: Изд-во ИСЭМ СО РАН, 2008. — С. 43–57.
- [9] Васильев Ф.П. Методы оптимизации / Ф.П. Васильев — М.: Факториал Пресс, 2002.

- [10] Васильев Ф.П. Линейное программирование / Ф.П. Васильев, А.Ю. Иваницкий — М.: Факториал Пресс, 2008.
- [11] Васин А.А. Теория игр и модели математической экономики / А.А. Васин, В.В. Морозов — М.: МАКС Пресс, 2005.
- [12] Вержбицкий В.М. Основы численных методов / В.М. Вержбицкий — М.: Высшая школа, 2002.
- [13] Втюрина М.В. Барьерно-проективный метод с наискорейшим спуском для линейных задач дополнителности / М.В. Втюрина, В.Г. Жадан // ЖВМ и МФ. — 2005. — Т.45, № 5. — С. 792–812.
- [14] Гермейер Ю.Б. Игры с противоположными интересами / Ю.Б. Гермейер — М.: Наука, 1976.
- [15] Гилл Ф. Практическая оптимизация / Ф. Гилл, У. Мюррей, М.Райт — М.: Мир, 1985.
- [16] Даугавет В.А. Численные методы квадратичного программирования / В.А. Даугавет — СПб.: Изд-во С.-Петербур. ун-та, 2004.
- [17] Горелик В.А. Теоретико-игровые модели принятия решений в эколого-экономических системах / В.А. Горелик, А.Ф. Кононенко — М.: Радио и связь, 1982.
- [18] Груздева Т.В. Локальный поиск в задачах с невыпуклыми ограничениями / Т.В. Груздева, А.С. Стрекаловский // ЖВМ и МФ. — 2007. — Т. 47, № 3. — С. 397–413.
- [19] Груздева Т.В. Решение задачи о клике сведением к задаче с d.c. ограничением / Т.В. Груздева // Дискретный анализ и исследование операций. — 2008. — Т. 15, № 6. — С. 20–34.
- [20] Груздева Т.В. Численное решение линейной двухуровневой задачи / Т.В. Груздева, Е.Г. Петрова // ЖВМ и МФ. — 2010. — Т.50, № 10. — С. 1715–1726.
- [21] Демьянов В.Ф. Недифференцируемая оптимизация / В.Ф. Демьянов, Л.В. Васильев — М.: Наука, 1981.

- [22] Демьянов В.Ф. Условия экстремума и вариационное исчисление / В.Ф. Демьянов — М.: Высшая школа, 2005.
- [23] Дикин И.И. Непрерывный процесс для задачи линейной дополнителности / И.И. Дикин — Дискретн. анализ и исслед. опер., серия 2. — 2001. — Т. 8, № 2. — С. 27–30.
- [24] Дикин И.И. Метод внутренних точек в линейном и нелинейном программировании / И.И. Дикин — М.: КРАСАНД, 2010.
- [25] Дэннис Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений / Дж. Дэннис, Р. Шнабель — М.: Мир, 1988.
- [26] Еремин И.И. Метод "штрафов" в выпуклом программировании / И.И. Еремин // ДАН. — 1967. — Т.173. — №4. — С. 748–751.
- [27] Еремин И.И. Введение в теорию линейного и выпуклого программирования / И.И. Еремин, Н.Н. Астафьев. - М.:Наука, 1976.
- [28] Еремин И.И. К методу штрафов в математическом программировании / И.И. Еремин // Доклады РАН. — 1996. — Т.346, № 4. —С. 459–461.
- [29] Ершова М.С. Введение в двухуровневое программирование / М.С. Ершова — Иркутск: Иркутский государственный университет, 2006.
- [30] Жадап В.Г. Метод Ньютона с наискорейшим спуском для линейной задачи дополнителности / В.Г. Жадап, А.В. Люлько — М.: ВЦ им. А.А. Дородницына РАН, 2002.
- [31] Измаилов А.Ф. Численные методы оптимизации / А.Ф. Измаилов, М.В. Солодов — М.: Физматлит, 2005.
- [32] Измаилов А.Ф. Чувствительность в оптимизации / А.Ф. Измаилов — М.: Физматлит, 2006.
- [33] Ильин В.А. Линейная алгебра / В.А. Ильин, Э.Г. Поздняк — М.: Наука, 1978.
- [34] Иоффе А.Д. Теория экстремальных задач / А. Д. Иоффе, В. М. Тихомиров — М.: Наука, 1974.

- [35] Калиткин Н.Н. Численные методы / Н.Н. Калиткин — М.: Гл. ред. физ.-мат. лит., 1978.
- [36] Карманов В.Г. Математическое программирование / В.Г. Карманов — М.: Наука, 1986.
- [37] Кларк Ф. Оптимизация и негладкий анализ / Ф. Кларк — М.: Наука, 1988.
- [38] Колмогоров А. Н. Элементы теории функций и функционального анализа / А.Н. Колмогоров, С.В. Фомин — М.: Наука, Гл. ред. физ.-мат. лит., 1972.
- [39] Красносельский М.А. Приближенное решение операторных уравнений / М.А. Красносельский — М.: Наука, 1969.
- [40] Ландис Е.М. О функциях, представимых в виде разности двух выпуклых / Е.М. Ландис // Доклады Академии Наук СССР. — 1951. — Т.80, № 1. — С. 9–11.
- [41] Михалевич В.С. Оптимизационные задачи производственно-транспортного планирования: Модели, методы, алгоритмы / В.С. Михалевич, В.А. Трубин, Н.З. Шор — М.: Наука, Гл. ред. физ.-мат. лит., 1986.
- [42] Мазуркевич Е.О. О численном решении линейной задачи дополнителности / Е.О. Мазуркевич, Е.Г. Петрова, А.С. Стрекаловский // ЖВМ и МФ. — 2009. — Т.49, № 8. — С. 1385–1398.
- [43] Мухамедиев Б.М. О решении задачи билинейного программирования и отыскании всех ситуаций равновесия в биматричных играх / Б.М. Мухамедиев // ЖВМ и МФ. — 1978. — Т.18, № 2. — С. 351–359.
- [44] Орлов А.В. Численное решение задач билинейного программирования / А.В. Орлов // ЖВМ и МФ. — 2008. — Т.48, № 2. — С. 237–254.
- [45] Ортега Дж. Итерационные методы решения нелинейных систем уравнений со многими неизвестными / Дж. Ортега, В. Рейнболдт — М.: Мир, 1975.
- [46] Петрова Е.Г. О локальном поиске в вариационном методе решения систем нелинейных уравнений / Е.Г. Петрова // Тезисы докладов VI Байкальской школы-семинара молодых ученых "Математическое моделирование и информационные технологии". — Иркутск-Ангарска.— Иркутск, 2005. — С. 32–33.



- [47] Петрова Е.Г. О вариационных методах решения систем нелинейных уравнений / Е.Г. Петрова, А.С. Стрекаловский // Труды XIII Байкальской международной школы-семинара "Методы оптимизации и их приложения". Том 1. Математическое программирование. — Иркутск, 2005. — С. 319-324.
- [48] Петрова Е.Г. Решение систем D.C. уравнений / Е.Г. Петрова // Тезисы докладов VII Байкальской школы-семинара молодых ученых "Математическое моделирование и информационные технологии". — Иркутск, 2005. — С. 29-30.
- [49] Петрова Е.Г. Вариационный подход для решения систем нелинейных уравнений / Е.Г. Петрова // Материалы III Всероссийской конференции "Проблемы оптимизации и экономические приложения". — Омск, 2006. — С. 151.
- [50] Петрова Е.Г. К теории двойственности в невыпуклых задачах / Е.Г. Петрова // Труды III межвузовской зональной конференции "Математика и проблемы ее преподавания в вузе". — Иркутск: Изд-во Иркут. гос. пед. ун-та, 2007. — С. 114.
- [51] Петрова Е.Г. Решение линейной задачи о дополнителности / Е.Г. Петрова // Тезисы докладов IX Байкальской школы-семинара молодых ученых "Математическое моделирование и информационные технологии", 22–27 октября 2007 г. — Иркутск: Изд-во ИДСТУ СО РАН, 2007. — С. 139–143.
- [52] Петрова Е.Г. Линейные двухуровневые задачи как задачи оптимизации с невыпуклым ограничением / Е.Г. Петрова, Т.В. Груздева // Труды XIV Байкальской международной школы-семинара. Том 1. — Иркутск: Изд-во ИСЭМ СО РАН, 2008. — С. 596–601.
- [53] Петрова Е.Г. Задача с невыпуклым ограничением и линейная двухуровневая задача / Е.Г. Петрова, Т.В. Груздева // Алгоритмический анализ неустойчивых задач: тез. докл. Международ. конф., посвящен. 100-летию со дня рождения В.К.Иванова, Екатеринбург, 1–6 сент., 2008 г. — Екатеринбург: изд-во Урал. ун-та, 2008. — С. 296–297.
- [54] Петрова Е.Г. О решении систем нелинейных алгебраических уравнений / Е.Г. Петрова, А.С. Стрекаловский // Современные технологии. Системный анализ. Моделирование. — 2009. — №24(4). — С. 30–36.

- [55] Петрова Е.Г. Численное решение линейных двухуровневых задач высокой размерности / Е.Г. Петрова // Материалы IV Всероссийской конференции "Проблемы оптимизации и экономические приложения". — Омск, 2008. — С. 189.
- [56] Петрова Е.Г. Процедура выпуклой комбинации для решения д.с. уравнений / Е.Г. Петрова, А.С. Стрекаловский // Тезисы докладов XI Байкальской школы-семинара молодых ученых "Математическое моделирование и информационные технологии", 15–21 марта 2010 г. — Иркутск: Изд-во ИДСТУ СО РАН, 2010. — С. 63.
- [57] Поляк Б.Т. Введение в оптимизацию / Б.Т. Поляк — М: Наука, 1983.
- [58] Попов Л.Д. Введение в теорию, методы и экономические приложения задач о дополнителности / Л.Д. Попов // Учеб. пособие. Екатеринбург: Изд-во Урал. ун-та, 2001.
- [59] Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи / Б.Н. Пшеничный — М.: Наука, Гл. ред. физ.-мат. лит., 1980.
- [60] Пшеничный Б.Н. Численные методы в экстремальных задачах // Б.Н. Пшеничный, Ю.М. Данилин — М.: Наука, 1975.
- [61] Роозе А. Набор тестовых систем нелинейных уравнений / А. Роозе, В. Кулла, М. Ломп, Т. Мерессоо — Таллин: Валгус, 1989.
- [62] Сергеев Я.Д. Диагональные методы глобальной оптимизации / Я.Д. Сергеев, Д.Е. Квасов — М.: Физматлит, 2008.
- [63] Современное состояние теории исследования операций / Под редакцией Н. Н. Моисеева. — М.: Наука, Гл. ред. физ.-мат. лит., — 1979.
- [64] Срочко В.А. Численные методы: курс лекций / В.А. Срочко — Иркутск: Иркут. ун-т, 2004.
- [65] Стрекаловский А.С. Условия глобальной оптимальности в задачах д.с. программирования / А.С. Стрекаловский // Иркутский государственный университет, серия: Оптимизация и управление. — Иркутск: Изд-во ИГУ, 1997.

- [66] Стрекаловский А. С. О сходимости алгоритма глобального поиска в задаче выпуклой максимизации на допустимом множестве / А.С. Стрекаловский, А.А. Кузнецова // Известия высших учебных заведений. Математика. — 1999. — №12. — С. 74–81.
- [67] Стрекаловский А.С. Об экстремальных задачах с д.с. ограничениями / А.С. Стрекаловский // ЖВМ и МФ. — 2001. — Т. 41, № 12. — С. 1833–1843.
- [68] Стрекаловский А.С. О численном решении задач невыпуклой оптимизации / А.С. Стрекаловский, А.А.Кузнецова, Т.В. Яковлева // Сибирский журнал вычислительной математики. — 2001. — Т. 4, № 2. — С. 185–199.
- [69] Стрекаловский А.С. Элементы невыпуклой оптимизации / А.С. Стрекаловский — Новосибирск: Наука, 2003.
- [70] Стрекаловский А.С. О минимизации разности двух выпуклых функций на допустимом множестве / А.С. Стрекаловский // ЖВМ и МФ. — 2003. — Т. 43, № 1. — С. 49–59.
- [71] Стрекаловский А.С. Модификация метода Розена в обратно-выпуклой задаче / А.С. Стрекаловский, Т.В. Яковлева // Известия вузов. Математика 2005. — № 12(523). — С. 70–75.
- [72] Стрекаловский А.С. Минимизирующие последовательности в задачах с д.с. ограничениями / А.С. Стрекаловский // ЖВМ и МФ. — 2005. — Т. 45, № 3. — С. 435–447.
- [73] Стрекаловский А.С. О вариационном подходе при решении систем нелинейных уравнений / А.С. Стрекаловский, Е.Г. Петрова // Материалы II Всероссийской конференции с международным участием "Инфокоммуникационные и вычислительные технологии и системы", 1–4 июля, 2006 г., Улан-Удэ, Байкал. — Иркутск, 2006. — С. 139–145.
- [74] Стрекаловский А.С. Элементы теории двойственности для задач д.с. минимизации / А.С. Стрекаловский, Е.Г. Петрова // Материалы конференции. Ляпуновские чтения и презентация информационных технологий — Иркутск, 2006. — С. 42.
- [75] Стрекаловский А.С. О задачах с ограничениями типа равенств / А.С. Стрекаловский, Е.Г. Петрова // Информационный бюллетень № 11, Тезисы докладов кон-

- ференции "Математическое программирование и приложения". — Екатеринбург, 2007. — С. 84–85.
- [76] Стрекаловский А.С. Вариационный подход к линейной задаче о дополнителности / А.С. Стрекаловский, Е.Г. Петрова // Материалы Российской конференции "Дискретная оптимизация и исследование операций", Владивосток, 7–14 сентября 2007. — Новосибирск: Изд-во Института математики, 2007. — С. 112.
- [77] Груздева Т.В. Локальный поиск в задачах с невыпуклыми ограничениями / Т.В. Груздева, А.С. Стрекаловский // ЖВМ и МФ. — 2007. — Т. 47. № 3. — С. 397–413.
- [78] Стрекаловский А.С. Биматричные игры и билинейное программирование / А.С. Стрекаловский, А.В. Орлов — М.: Физматлит, 2007.
- [79] Стрекаловский А.С. Локальный поиск в квадратично-линейной задаче двухуровневого программирования / А.С. Стрекаловский, А.В. Орлов, А.В. Малышев // СибЖВМ. — 2010. — Т.13. №1 — С. 75–88.
- [80] Стрекаловский А.С. Численное решение одного класса задач двухуровневого программирования / А.С. Стрекаловский, А.В. Орлов, А.В. Малышев // СибЖВМ. — 2010. — Т.13. №2 — С. 201–212.
- [81] Сухарев А.Г. Курс методов оптимизации / А.Г. Сухарев, А.В. Тимохов, В.В. Федоров — М.: Наука, Гл. ред. физ.-мат. лит., 1986.
- [82] Тыртышников Е.Е. Матричный анализ и линейная алгебра / Е.Е. Тыртышников — М.: Физматлит, 2007.
- [83] Фаддеев Д.К. Вычислительные методы линейной алгебры / Фаддеев Д.К., Фаддеева В.Н. — С-Пб.: Лань, 2002.
- [84] Фиакко А. Нелинейное программирование. Методы последовательной безусловной минимизации / А. Фиакко, Г.П. Мак-Кормик — М.: Мир, 1972.
- [85] Хамисов О.В. Нахождение корней нелинейного уравнения методом вогнутых опорных функций / О.В. Хамисов // Труды XII Байкальской международной конференции. Том 4. — Иркутск: Изд-во ИСЭМ СО РАН, 2001. — С. 194–198.

- [86] Шарый С.П. Интервальные методы для систем уравнений и необходимость переформулировки задачи / С.П. Шарый — Вычислительные Технологии. — в печати.
- [87] Шор Н.З. Методы минимизации недифференцируемых функций и их приложения / Н.З. Шор — Киев: Наукова думка, 1976.
- [88] Alefeld G. Solutions of Linear Complementarity Problems for H-matrices / G. Alefeld, Z. Wang, Z. Shen // *Reliable Computing*. — 2004. — V.10. — P. 423–435.
- [89] Audet C. New Branch-and-Cut Algorithm for Bilevel Linear Programming / C. Audet, G. Savard, W. Zghal // *Journal of Optimization Theory and Applications*. — 2007. — V.134, № 2. — P.353–370.
- [90] Bard J.F. *Practical Bilevel Optimization* / J.F. Bard — Dordrecht: Kluwer Academic Publishers, 1998.
- [91] Bellavia S. STRSCNE: A Scaled Trust Region Solver for Constrained Nonlinear Equations / S. Bellavia, M. Macconi, B. Morini // *COAP*, 2004. — V. 28, №. 1. — P. 31–50.
- [92] Calamai P. Generating Linear and Linear-Quadratic Bilevel Programming Problems / P. Calamai, L. Vicente // *SIAM Journal on Scientific Computing archive*. — 1993. — V.14, № 4. — P. 770–782.
- [93] Calamai P. Generating Quadratic Bilevel Programming Test Problems / P. Calamai, L. Vicente // *ACM Transactions on Mathematical Software*. — 1994. — V.20. — P. 103–119.
- [94] Campelo M. A Note on a Penalty Function Approach for Solving Bilevel Linear Programs / M. Campelo, S. Dantas, S. Scheimberg // *Journal of global optimization*. — 2000. — V. 16. — P. 245–255.
- [95] Campelo M., Scheimberg S. A Study of Local Solutions in Linear Bilevel Programming / M. Campelo, S. Scheimberg // *Journal of optimization theory and applications*. — 2005. — V.125, №1. — P. 63–84.
- [96] Candler W. A linear two-level programming problem / W. Candler, R. Townsley // *Computers and Operations Research*. — 1982. — V.9. — P. 59–76.

- [97] Chung S. J. NP-Completeness of the Linear Complementarity Problem / S.J. Chung // Journal of optimization theory and applications. — 1989. — V. 60 , №3. — P. 393–399.
- [98] Colson B. An overview of bilevel optimization / B. Colson, P. Marcotte, G. Savard // Annals of operations research. — 2007. — V.153, №1. — P.235–256.
- [99] Cottle R.W. Linear Complementarity since 1978 / R.W. Cottle // Variational Analysis and Appls. — 2007. — V.79. — P. 239–257.
- [100] Cottle R.W. The Linear complementarity problem / R.W. Cottle, J.S. Pang, R.E. Stone — Academic Press, 1999.
- [101] Dantzig G.B. Positive (semi-)definite programming / G.B. Dantzig, R.W. Cottle // Nonlinear Programming. — Amsterdam: North-Holland, 1967. — P. 55–73.
- [102] Dempe S. A simple algorithm for the linear bilevel programming problem / S. Dempe // Optimization. — 1987. — V.18. — P. 373–385.
- [103] Dempe S. Foundations of Bilevel Programming / S. Dempe — Dordrecht: Kluwer Academic Publishers, 2002.
- [104] Facchinei F. Finite-Dimensional Variational Inequalities and Complementarity Problems / F. Facchinei, J.-S. Pang — Berlin: Springer - Verlag, 2003 (two volumes).
- [105] Hartman P. On Functions Representable as a Difference of Convex functions / P. Hartman // Pacific Journal of Mathematics. — 1959. — V.9 — P. 707–713.
- [106] Hiriart-Urruty J.-B. Conditions for Global Optimization / J.-B. Hiriart-Urruty // Handbook of Global Optimization, Kluwer Academic Publishers. — 1995. — P. 1–26.
- [107] Hiriart-Urruty J.-B. Generalized Differentiability, Duality and Optimization for Problem Dealing with differences of convex functions / J.-B. Hiriart-Urruty // Lecture Notes in Economics and Mathematical Systems. — Berlin: Springer Verlag, 1985. — V.256. — P. 37–69.
- [108] Hiriart-Urruty J.-B. Conditions for Global Optimality / J.-B. Hiriart-Urruty // Handbook of Global Optimization. — Dordecht: Kluwer Academic Publishers, 1995. — P.1–26.

- [109] Hiriart-Urruty J.-B. From convex optimization to non convex optimization. Part I: Necessary and sufficient conditions for global optimality // *Nonsmooth Optimization and Related Topics* / Ed. by F. H. Clarke, V.F. Demyanov, F. Gianessi, — New York: Plenum Press, 1989. — P. 219–239.
- [110] Horst R. *Global Optimization. Deterministic Approaches* / R. Horst, H. Tuy — Berlin: Springer-Verlag, 1993.
- [111] Horst R. D.C. Programming: Overview / R. Horst, N.V. Thoai // *Journal of Optimization Theory and Applications*. — 1999. — V.103, №1. — P. 1–43.
- [112] Horst R. *Introduction to global optimization*/ R. Horst, P. Pardalos, N. V. Thoai — Dordrecht–Boston–London: Kluwer Academic Publishers, 1995. — V.3. — 246 p.
- [113] Jia F. Sensitivity analysis in bilevel linear programming / F. Jia, F. Yang, S.-Y. Wang. // *Systems Science and Mathematical Sciences*. — 1998. — V. 11. — P. 359–366.
- [114] Judice J.J. The linear-quadratic bilevel programming problem / J.J. Judice, A. Faustino // *Information Systems and Operational Research*. — 1994. — V. 32. — P. 87–98.
- [115] L.V. Kolev A new method for global solution of systems of non-linear equations / L.V. Kolev // *Reliable Computing*. — 1998. — V.4. — P. 125–146.
- [116] Land A.H. An automatic method of solving discrete programming problems / A.H. Land, A.G. Doig // *Econometrica*. — 1960 — V. 28, №3. — P. 497–520.
- [117] Le Th.H. *DC Programming Approaches and DCA for globally solving Linear Complementarity Problems* / Th.H. Le, D.T. Pham // *Research Report, National Institute for Applied Sciences, Rouen, 2004*.
- [118] Le Th.H. The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems / Th.H. Le, D.T. Pham // *Annals of Operations Research*. — 2005. — V. 133. — P. 23–46.
- [119] Lemke C.E. Equilibrium points of bimatrix games / C.E. Lemke, J.T. Howson // *SIAM Journal on Applied Mathematics*. — 1964. — V.12. — P. 413–423.

- [120] Kojima M. A unified approach to interior point algorithms for linear complementarity problems / M. Kojima, N. Megiddo, T. Noma, A. Yoshise — Berlin: Springer-Verlag, 1991.
- [121] Morales J.L. An algorithm for the fast solution of linear complementarity problems / J.L. Morales, J. Nocedal, M. Smelyanskiy // *Numerische Mathematik*. — 2008. —V.3, №2. —P. 251–266.
- [122] More J.J. Classes of functions and feasibility conditions in nonlinear complementarity problems / J.J. More // *Math. Programming*. — 1974. — V.6, №3. — P. 327–338.
- [123] Li L. A Block Recursive Algorithm For the Linear Complementarity Problem With an M-matrix / L. Li, Y. Kobayashi // *International Journal of Innovative Computing, Information and Control ICIC International*. — 2006. — V.2, №6. — P. 1327–1335.
- [124] Nemhauser G.N. *Integer and Combinatorial Optimization* / G.N. Nemhauser , L.A. Wolsey — Wiley-Interscience Publication, 1999.
- [125] Nguyen V.Th. Global Optimization Method for Solving Mathematical Programs with Linear Complementarity Constraints / V.Th. Nguyen, Y. Yamamoto, A. Yoshise // *Journal of Optimization Theory and Application*. — 2005. — №124. — P. 467–490.
- [126] Nocedal J. *Numerical optimization* / J. Nocedal, S.J. Wright — New York, Berlin, Heidelberg: Springer-Verlag, 2000.
- [127] Pang J.S. Complementarity problems / J.S. Pang // *Handbook of Global Optimization*, Kluwer Academic Publishers. — 1995. — P. 271–338.
- [128] Penot J.-P. Approximation and decomposition properties of some classes of locally d.c. functions / J.-P. Penot, M.L. Bougeard // *Mathematical Programming*. — 1988. — V.41. — P. 195–227.
- [129] Pochet Y. *Production Planning by Mixed Integer Programming* / Y. Pochet , L.A. Wolsey — Springer, 2006.
- [130] Saboia C.H. A computational study of global algorithms for linear bilevel programming / C.H. Saboia, M. Campelo, S. Scheimberg // *Numerical Algorithms*. — 2004. — V.35, №2–4. — P.155–173.



- [131] Simantiraki E.M. An Infeasible-Interior-Point method for linear complementarity problems / E.M. Simantiraki, D.F. Shanno // Rutcor Research Report. — 1996.
- [132] Still G. Linear bilevel problems: Genericity results and an efficient method for computing local minima / G. Still // Math. Meth. Oper. Res. — 2002. — V.55. — P. 383–400.
- [133] Strekalovsky A.S. On an approach to linear complementarity problem / A.S. Strekalovsky, E.G. Petrova // The Second International Conference on Optimization and Optimal Control. — Mongolia, Ulaanbaatar, 2007. — P. 40–41.
- [134] Strekalovsky A. S. One way to construct a global search algorithm for d.c. minimization problems / A.S. Strekalovsky // Nonlinear optimization and related topics / Ed. by G. Di Pillo, F.Giannessi, Applied Optimization Series. — Dordrecht: Kluwer Academic Publishers, 2000. — V.36. — P. 429–443.
- [135] Tamir A. Minimality and complementarity problems associated with Z-functions and M-functions // Mathematical Programming. — 1974. — V.7, №1. — P. 17–31.
- [136] Tuy H. A global optimization approach for the linear two-level program / H. Tuy, A. Migdalas, P. Vbrand // Journal of Global Optimization. — 1993. — V.3. — P. 1–23.
- [137] Tuy H. D.C. Optimization: Theory, Methods and Algorithms / H. Tuy // Handbook of Global Optimization, Kluwer Academic Publishers, 1995. — P. 149–216.
- [138] White D. A penalty function approach for solving bi-level linear programs / D. White, G. Anandalingam // Journal of Global Optimization. — 1993. — V.3. — P. 397–419.
- [139] Wolsey L.A. Integer Programming / L.A. Wolsey — Wiley & Sons, Inc., 1998.
- [140] Fair Isaac Corporation: Xpress Optimization Suite. URL: [www.fico.com](http://www.fico.com).
- [141] CPLEX user's manual. ILOG, Inc. URL: [www.ilog.com](http://www.ilog.com)
- [142] URL: <http://pages.cs.wisc.edu/~ferris/path.html>.
- [143] URL: <http://www.mathworks.com>.