Серия
Суперкомпьютерное
Образование

Координационный совет
Системы научно-образовательных центров суперкомпьютерных технологий

Председатель Координационного совета

В. А. Садовничий,
ректор МГУ имени М. В. Ломоносова,
академик

Заместители председателя совета

Е. И. Моисеев,
декан факультета вычислительной математики и кибернетики
МГУ имени М. В. Ломоносова,
академик

А. В. Тихонравов,
директор Научно-исследовательского вычислительного центра
МГУ имени М. В. Ломоносова,
профессор

Члены совета

В. Н. Васильев, ректор Санкт-Петербургского национального исследовательского государственного университета информационных технологий, механики и оптики, чл.-корр. РАН, профессор; **В. Г. Захаревич**, ректор Южного федерального университета, профессор; **Н. Н. Кудрявцев**, ректор Московского физико-технического института, чл.-корр. РАН, профессор; **Г. В. Майер**, ректор национального исследовательского Томского государственного университета, профессор; **А. А. Фаткулин**, проректор по науке и инновациям Дальневосточного федерального университета, профессор; **Е. В. Чупрунов**, ректор национального исследовательского Нижегородского государственного университета, профессор; **А. Л. Шестаков**, ректор национального исследовательского Южно-Уральского государственного университета, профессор; **В. Н. Чубариков**, декан механико-математического факультета МГУ имени М. В. Ломоносова, профессор; **М. И. Панасюк**, директор Научно-исследовательского института ядерной физики МГУ имени М. В. Ломоносова, профессор; **Вл. В. Воеводин**, заместитель директора Научно-исследовательского вычислительного центра МГУ имени М. В. Ломоносова, исполнительный директор НОЦ «СКТ-Центр», член-корреспондент РАН.

Нижегородский государственный университет
имени Н. И. Лобачевского

Р. Г. Стронгин, В. П. Гергель, В. А. Гришагин,
К. А. Баркалов

Параллельные вычисления в задачах глобальной оптимизации

Монография

Рекомендовано Ученым советом
факультета вычислительной математики и кибернетики (ВМК)
Нижегородского государственного университета имени Н. И. Лобачевского
для использования в учебном процессе

Издательство Московского университета
2013

УДК 007 (075)
ББК 32.973.2
С86

Стронгин Р. Г., Гергель В. П., Гришагин В. А., Баркалов К. А.

С86 Параллельные вычисления в задачах глобальной оптимизации: Монография / Предисл.: В. А. Садовничий. – М.: Издательство Московского университета, 2013. – 280 с., илл. – (Серия «Суперкомпьютерное образование»)

ISBN 978-5-211-06479-9

Данная работа посвящена проблематике анализа сложных многоэкстремальных моделей принятия решений, которая является традиционной для научной школы проф. Р. Г. Стронгина, имеющей мировое признание в этой области. Работа содержит необходимый теоретический аппарат для конструирования и анализа параллельных методов, основанный на информационно-статистическом подходе к построению методов оптимизации, характеристической теории сходимости и эффективности методов поиска экстремума и методологии редукции сложности исследуемых моделей. Приводится описание вычислительных схем методов оптимизации, их теоретическое обоснование и примеры практического применения. Дается краткая характеристика программных средств, реализующих в виде законченных программных систем предложенные методы поиска глобально-оптимальных решений и нашедших свое применение при решении сложных прикладных задач и в учебном процессе как среда выполнения лабораторных работ и исследований.

Книга предназначена для широкого круга студентов, аспирантов и специалистов, желающих изучить и практически использовать параллельные методы глобальной оптимизации для решения вычислительно трудоемких прикладных задач.

Данная монография рекомендована Ученым советом факультета вычислительной математики и кибернетики (ВМК) Нижегородского государственного университета имени Н. И. Лобачевского для использования в учебном процессе.

Подготовка монографии была выполнена в рамках реализации проекта комиссии Президента РФ по модернизации и технологическому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения».

УДК 007 (075)
ББК 32.973.2

© Р. Г. Стронгин, В. П. Гергель, В. А. Гришагин, К. А. Баркалов., 2012

ISBN 978-5-211-06479-9 © Издательство Московского университета, 2013

Уважаемый читатель!

Вы держите в руках одну из книг серии «Суперкомпьютерное образование», выпущенную в рамках реализации проекта комиссии Президента РФ по модернизации и технологическому развитию экономики России «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения». Инициатором издания выступил Суперкомпьютерный консорциум университетов России.

Серия включает более 20 учебников и учебных пособий, подготовленных ведущими отечественными специалистами в области суперкомпьютерных технологий. В книгах представлен ценный опыт преподавания суперкомпьютерных технологий в таких авторитетных вузах России, как МГУ, ННГУ, ТГУ, ЮУрГУ, СПбГУ ИТМО и многих других. При подготовке изданий были учтены рекомендации, сформулированные в Своде знаний и умений в области суперкомпьютерных технологий, подготовленном группой экспертов Суперкомпьютерного консорциума, а также международный опыт.

Современный уровень развития вычислительной техники и методов математического моделирования дает уникальную возможность для перевода промышленного производства и научных исследований на качественно новый этап. Эффективность такого перехода напрямую зависит от наличия достаточного числа высококвалифицированных специалистов. Данная серия книг предназначена для широкого круга студентов, аспирантов и специалистов, желающих изучить и практически использовать параллельные компьютерные системы для решения трудоемких вычислительных задач.

Издание серии «Суперкомпьютерное образование» наглядно демонстрирует тот вклад, который внесли участники Суперкомпьютерного консорциума университетов России в создание национальной системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий, а также их четкое понимание ответственности за подготовку высококвалифицированных специалистов и формирование прочного научного фундамента, столь необходимого для эффективного использования суперкомпьютерных технологий на практике.

Ректор Московского университета,
Президент Суперкомпьютерного консорциума
университетов России,
академик РАН *В. А. Садовничий*

Оглавление

Оглавление 7

Введение 11

Глава 1. Глобальный поиск: от одномерных к многомерным задачам через редукцию размерности 16

1.1. Задача глобальной оптимизации 16

1.2. Численные методы оптимизации 21

1.2.1. Понятие алгоритма. Последовательная схема 21

1.2.2. Параллельные алгоритмы оптимизации 24

1.2.3. Сходимость и оценки экстремума 29

1.3. Многоэкстремальные задачи и редукция сложности 32

Глава 2. Алгоритмы глобального поиска для одномерных многоэкстремальных задач оптимизации 36

2.1. Информационно-статистический подход к глобальному поиску 36

2.2. Базовый алгоритм глобального поиска 41

2.2.1. Вычислительная схема алгоритма 42

2.2.2. Условия сходимости и оценка эффективности 43

2.3. Характеристическая схема представимости алгоритмов глобального поиска 49

2.4. Ускорение глобального поиска на основе использования дополнительной информации 54

2.4.1. Адаптивный прогноз оптимума 55

2.4.2. Проблема оценки константы Липшица и локально-адаптивные схемы глобального поиска 60

2.4.3. Алгоритмы глобального поиска с использованием значений производных 69

2.5. От эффективных последовательных алгоритмов глобального поиска — к параллельным вычислениям 77
2.5.1. Параллельные синхронные характеристические алгоритмы глобального поиска 77
2.5.2. Условия безызбыточного распараллеливания 95
2.5.3. Численные эксперименты 101
2.5.4. Параллельные асинхронные характеристические алгоритмы глобального поиска 107

Глава 3. Алгоритмы глобального поиска для одномерных многоэкстремальных задач оптимизации с нелинейными ограничениями 111
3.1. Постановка задачи 111
3.2. Частичная вычислимость и индексная схема учета ограничений. 111
3.3. Индексный алгоритм глобального поиска 116
3.4. Индексный алгоритм, учитывающий существование ε-резервированных решений 121
3.5. Индексный метод с адаптивным порядком проверки ограничений. 129
3.6. Параллельные вычисления для многоэкстремальных задач оптимизации с нелинейными ограничениями 141
3.6.1. Асинхронный параллельный индексный алгоритм 141
3.6.2. Условия сходимости алгоритма. 143

Глава 4. Параллельные вычисления для многомерной многоэкстремальной оптимизации на основе многошаговой схемы редукции размерности 151
4.1. Общая схема многошаговой схемы редукции размерности. . . 151
4.2. Свойства одномерных подзадач многошаговой схемы 157
4.2.1. Структура допустимых областей одномерного поиска. 157
4.2.2. Свойства целевых функций в одномерных подзадачах 164
4.3. Параллельные вычисления для многошаговой схемы редукции размерности 166
4.3.1. Вычислительные эксперименты 169

4.4. Обобщенная адаптивная многошаговая схема редукции размерности 176
4.4.1. Общее описание подхода. 176
4.4.2. Алгоритмическое описание 178
4.4.3. Многомерные характеристические алгоритмы глобального поиска на основе адаптивной многошаговой схемы редукции размерности 183
4.4.4. Информационно-статистические алгоритмы глобального поиска в рамках адаптивной многошаговой схемы редукции размерности 186
4.4.5. Вычислительные эксперименты для оценки эффективности адаптивной многошаговой схемы редукции размерности 192
4.4.6. Операционные характеристики алгоритмов глобального поиска при адаптивной многошаговой схеме редукции размерности 193
4.5. Адаптивные параллельные вычисления для многомерной многоэкстремальной оптимизации 195
4.5.1. Централизованная схема параллельного глобального поиска 195
4.5.2. Централизованная схема параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности . . 197
4.5.3. Централизованная схема параллельного глобального поиска с несколькими управляющими процессорами для адаптивной многошаговой схемы редукции размерности 199
4.5.4. Распределенная схема параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности . . 200

Глава 5. Параллельные вычисления для многомерной многоэкстремальной оптимизации при редукции размерности с использованием кривых Пеано 209
5.1. Общая схема подхода 209
5.2. Многомерный обобщенный алгоритм глобального поиска . . . 216
5.3. Параллельные многомерные многоэкстремальные методы на основе множественных кривых Пеано 220
5.3.1. Использование множественных отображений 220
5.3.2. Организация параллельных вычислений 223
5.3.3. Условия сходимости алгоритма. 228

5.4. Параллельные вычисления на основе новых схем построения множественных кривых Пеано 237
5.4.1. Схема построения вращаемых разверток 237
5.4.2. Сравнение эффективности алгоритмов. 241

Глава 6. Суперкомпьютерные параллельные вычисления при решении прикладных оптимизационных задач 245
6.1. Оптимизация профиля колеса для рельсовых видов транспорта 245
6.1.1. Постановка задачи 245
6.1.2. Результаты численных экспериментов 247
6.2. Решение обратной задачи химической кинетики 249
6.2.1. Постановка задачи 249
6.2.2. Результаты численных экспериментов 251
6.3. Идентификация динамической балансовой нормативной модели региональной экономики 253
6.3.1. Постановка задачи 253
6.3.2. Описание модели региональной экономики 254
6.3.3. Результаты численных экспериментов 257

Глава 7. Программные системы глобальной оптимизации 259
7.1. Программная система АБСОЛЮТ для исследований и изучения методов глобальной оптимизации 259
7.2. Программная система ПараЛаб для изучения и исследования методов параллельных вычислений 263
7.3. Программная система GlobalExpert для параллельного решения задач глобальной оптимизации 267

Список литературы 271

Введение

Несмотря на поразительно быстрый рост производительности компьютеров, в любое время существуют актуальные задачи фундаментальной и прикладной науки, для анализа и исследования которых производительности существующих средств вычислительной техники оказывается недостаточно. К числу таких задач относятся, например, так называемые проблемы «большого вызова» — моделирование климата, генная инженерия, проектирование интегральных схем, анализ загрязнения окружающей среды, создание лекарственных препаратов и др. К данной категории, без сомнения, можно отнести многие задачи моделирования и анализа интеллектуальных процессов принятия решений, имеющие широкий спектр прикладных применений и обладающие высокой степенью трудоемкости вследствие зависимости моделей от большого числа параметров и сложного характера функционалов, описывающих данные зависимости.

Возможный путь для решения вычислительно трудоемких задач состоит в построении параллельных вычислительных систем (ПВС), на которых независимые части алгоритмов решения вычислительных задач могут выполняться параллельно. Компьютерная индустрия сформировала несколько основных направлений для построения подобных систем — в их числе суперкомпьютеры, симметричные мультипроцессорные системы (symmetric multiprocessors — SMP), массивно-многопроцессорные системы (massively parallel processors — MPP) и др. Перспективное направление построения параллельных ВС, активно развиваемое в последнее время, состоит в построении *кластерных вычислительных систем* (*clusters*), под которыми обычно понимается множество отдельных компью-

теров, объединенных в сеть, для которых при помощи специальных аппаратно-программных средств обеспечивается возможность унифицированного управления (*single system image*), надежного функционирования (*reliability*) и эффективного использования (*performance*). Кластерный подход может в значительной степени снизить проблемы использования параллельных ВС.

Вместе с тем, при любой архитектуре сдерживающим фактором применения ПВС является сложность разработки эффективных параллельных алгоритмов и программ. Темпы развития моделей, методов и программного обеспечения для решения сложных вычислительно трудоемких задач не отличаются столь же высокими показателями, как в случае компьютерного оборудования.

Настоящая монография, предлагаемая вниманию заинтересованного читателя, основана на результатах исследований Нижегородской научной школы параллельных вычислений проф. Р. Г. Стронгина, выполненных за последние десятилетия в Нижегородском государственном университете им. Н. И. Лобачевского — национальном исследовательском университете. В центре внимания авторов книги — проблематика анализа сложных многоэкстремальных моделей принятия решений, которая является традиционной для научной школы проф. Р. Г. Стронгина, имеющей мировое признание в этой области. Монография является первой попыткой систематизации результатов по разработке моделей и методов параллельных вычислений при решении задач многоэкстремальной оптимизации, которые ранее были опубликованы во многих отдельных источниках.

В качестве модели принятия решений рассматривается модель с единственным критерием оптимальности при наличии ограничений на выбор параметров модели. Критерий оптимальности и функции-ограничения предполагаются *многоэкстремальными*, что приводит к формированию в качестве изучаемой модели *многомерной задачи многоэкстремального нелинейного программирования*, искомым решением которой является *глобальный экстремум* (минимум или максимум). Задачи данного класса широко распространены в приложениях (задачи оптимального проектирования объектов и технологических процессов в различных отраслях

народного хозяйства, задачи идентификации моделей по данным наблюдений в научных исследованиях, оптимизация процессов управления и т. п.), что определяет их высокую актуальность и практическую значимость.

Многоэкстремальные оптимизационные модели обладают узкой сферой возможностей аналитического исследования и высокой трудоемкостью численного анализа, поскольку в последнем случае для них характерен экспоненциальный рост вычислительных затрат с ростом размерности (количества параметров модели). Именно поэтому разработка эффективных параллельных методов для численного решения задач многоэкстремальной оптимизации и создание программных средств их реализации на современных многопроцессорных системах является стратегическим направлением по существенному развитию проблематики исследования сложных задач оптимального выбора. Предлагаемые авторами способы параллельных вычислений позволяют эффективно использовать тысячи процессоров современных суперкомпьютерных систем.

Монография содержит необходимый теоретический аппарат для конструирования и анализа параллельных методов, основанный на информационно-статистическом подходе к построению методов оптимизации, характеристической теории сходимости и эффективности методов поиска экстремума и методологии редукции сложности исследуемых моделей. Приводится описание вычислительных схем методов оптимизации, их теоретическое обоснование и примеры практического применения. Дается краткая характеристика программных средств, реализующих в виде законченных программных систем предложенные методы поиска глобально-оптимальных решений и нашедших свое применение при решении сложных прикладных задач и в учебном процессе как среда выполнения лабораторных работ и исследований.

Книга имеет следующую структуру.

Первая глава содержит постановку исследуемой задачи многоэкстремального нелинейного программирования, общие вычислительные схемы последовательного, параллельного синхронного и параллельного асинхронного методов оптимизации. В ней также

обсуждаются методы редукции сложности многоэкстремальных задач.

Вторая глава посвящена рассмотрению одномерных алгоритмов поиска глобального экстремума как основы для конструирования численных методов решения многомерных задач оптимизации посредством применения схем редукции размерности. Глава содержит краткое изложение информационно-статистического подхода к созданию методов поиска экстремума, а также вычислительную схему и теоретические характеристики базового информационно-статистического алгоритма глобального поиска.

Далее вводится широкий класс *характеристических алгоритмов оптимизации*, объединяющий многие известные методы поиска экстремума, и формулируются основные результаты по изучению сходимости методов данного класса. Рассматриваются возможности повышения эффективности методов оптимизации за счет использования дополнительной информации о локальных оценках констант Липшица и значений производных.

Формулируется обобщение схемы характеристической представимости на параллельный случай как для синхронного, так и для асинхронного случаев. Дается (впервые в русскоязычной публикации) полное теоретическое обоснование условий сходимости и эффективности распараллеливания алгоритмов данного класса и приводятся результаты вычислительных экспериментов.

В третьей главе рассматривается и обосновывается параллельный метод редукции задачи с ограничениями к задаче без ограничений, получивший название *индексного метода*. Данный метод в отличие от классического метода штрафных функций позволяет избежать проблемы подбора параметров (типа константы штрафа или выравнивающих коэффициентов при ограничениях), а также позволяет решать задачи с так называемыми частично вычислимыми ограничениями, которые никаким другим методом решены быть не могут.

Четвертая глава содержит описание многошаговой схемы редукции размерности, посредством которой многомерная задача оптимизации сводится к решению рекурсивно вложенных одномерных подзадач. Сначала рассматривается классическая многошаговая

схема и ее параллельная версия, а затем адаптивная многошаговая схема, позволяющая повысить эффективность решения задачи за счет более полного учета информации, получаемой в процессе поиска.

Пятая глава посвящена рассмотрению другой схемы редукции размерности, когда исходная многомерная задача заменяется эквивалентной одномерной задачей на основе применения отображений многомерной области на отрезок вещественной оси (такие отображения называются кривыми, или развертками Пеано). Дается общее описание подхода и его обоснование, рассматривается развитие классической схемы на основе применения семейства отображений (множественная развертка, вращаемые развертки) и реализация новых типов разверток в параллельном варианте.

В шестой главе приводятся примеры решения прикладных задач в области оптимального проектирования (определение оптимального профиля колеса транспортного устройства), химических исследований (определение неизвестных кинетических параметров в математических моделях сложных химических реакций), экономики (идентификация параметров модели региональной экономики).

В седьмой главе дается краткое описание программных систем анализа многоэкстремальных оптимизационных моделей.

Подготовка настоящей книги была выполнена при финансовой поддержке РФФИ (грант № 11-01-00682), Совета по грантам Президента РФ (грант № НШ-1960.2012.9) и Министерства образования и науки (соглашения № 14.B37.21.0878 и № 14.B37.21.0393).

Глава 1

Глобальный поиск: от одномерных к многомерным задачам через редукцию размерности

1.1. Задача глобальной оптимизации

Целью этого раздела является введение необходимых понятий и формулировка общей задачи глобальной оптимизации, исследованию которой посвящен дальнейший материал книги.

Пусть $\varphi(y)$ — действительная функция, определенная в области Q N-мерного евклидова пространства R^N и принимающая во всех точках области конечные значения.

Обозначим через

$$\inf_{y \in Q} \varphi(y) = \inf\{\varphi(y)\colon y \in Q\}$$

точную нижнюю грань функции $\varphi(y)$ на множестве Q.

Определение 1.1. Если существует точка $y^* \in Q$ такая, что

$$\varphi(y^*) = \inf\{\varphi(y)\colon y \in Q\}, \tag{1.1}$$

то говорят, что функция $\varphi(y)$ достигает своей точной нижней грани в области Q, а y^* называют *точкой глобального* (*абсолютного*) *минимума*, или глобальным минимайзером. Величина $\varphi^* = \varphi(y^*)$ называется наименьшим или *глобально-оптимальным* (*минимальным*) *значением* φ на Q и обозначается

$$\min_{y \in Q} \varphi(y) = \min\{\varphi(y)\colon y \in Q\}.$$

Множество всех точек $y^* \in Q$, удовлетворяющих (1.1), будем обозначать через

$$Q^* = \operatorname{Arg}\min_{y \in Q} \varphi(y) \equiv \operatorname{Arg}\min\{\varphi(y)\colon y \in Q\}. \tag{1.2}$$

Дадим еще одно важное определение, характеризующее экстремальные свойства функций.

Определение 1.2. Точка $y' \in Q$ называется *точкой локального минимума* функции на множестве Q, если существует такое число $\varepsilon > 0$, что для всех $y \in Q$ таких, что $||y - y'|| < \varepsilon$, выполняется $\varphi(y') \leqslant \varphi(y)$.

Функции, имеющие в области Q несколько локальных минимумов, называются многоэкстремальными в этой области, а имеющие единственный локальный минимум – унимодальными (одноэкстремальными). Очевидно, что любой глобальный минимум является локальным, а для унимодальных функций локальный минимум в силу его единственности является глобальным.

В оптимизационных моделях, связанных с анализом функции $\varphi(y)$, возможна различная формализация понятия искомого решения и, как следствие, возникают различные постановки задач оптимизации. В связи с этим дадим обобщение подобных постановок в следующей формулировке:

Определение 1.3. *Задачей оптимизации* будем называть задачу следующего вида: найти заданные экстремальные характеристики функции $\varphi(y)$ на множестве Q.

В зависимости от искомых экстремальных характеристик возможны различные конкретные постановки задачи оптимизации.

Постановка A. Найти точную нижнюю грань функции $\varphi(y)$

$$\varphi^* = \inf\{\varphi(y)\colon y \in Q\}. \tag{1.3}$$

Постановка B. Определить точную нижнюю грань из (1.3) и, если множество точек глобального минимума из (1.2) не пусто, найти хотя бы одну точку $y^* \in Q^*$.

Постановка C. Найти точную нижнюю грань φ^* из (1.3) и определить все точки глобального минимума (либо убедиться, что множество Q^* пусто).

Постановка D. Найти все координаты и значения локальных минимумов функции $\varphi(y)$ на множестве Q.

Постановки A–D связаны с нахождением глобального минимума, но иногда даже для многоэкстремальных функций используется

Постановка E. Найти локальный минимум функции $\varphi(y)$ на множестве Q.

Постановки A–E задачи оптимизации являются наиболее распространенными, хотя возможны и другие варианты, например, указанные в [1]. Искомые экстремальные характеристики, определяемые постановкой $\mathrm{Z} \in \{\mathrm{A, B, C, D, E}\}$, назовем *Z-решением задачи оптимизации*.

Символически общую задачу оптимизации будем записывать в виде

$$\varphi(y) \to \inf, \quad y \in Q. \tag{1.4}$$

Данную задачу также довольно часто называют *задачей математического программирования*.

Если необходимо конкретизировать рассматриваемую постановку, будем указывать соответствующее искомое Z-решение задачи.

Так как точная верхняя грань функции $\varphi(y)$ на множестве Q связана с точной нижней гранью функции $-\varphi(y)$ соотношением

$$\sup\{\varphi(y)\colon y \in Q\} = -\inf\{-\varphi(y)\colon y \in Q\},$$

то задача определения экстремальных характеристик, связанных с наибольшим значением функции $\varphi(y)$ (задача максимизации), сводится к задаче минимизации функции $-\varphi(y)$. Поэтому везде далее задача оптимизации будет рассматриваться в форме (1.4). Функцию $\varphi(y)$ из (1.4) будем называть *целевой, минимизируемой*, или *оптимизируемой функцией*, множество Q — *допустимой областью*, а элементы множества Q — *допустимыми точками*.

Задачу (1.4), для которой заведомо известно, что множество точек глобального минимума Q^* не пусто (достаточные условия непустоты Q^* даются, например, теоремой Вейерштрасса [51]), будем записывать как

$$\varphi(y) \to \min, \quad x \in Q. \tag{1.5}$$

Заметим, что часто задача математического программирования формулируется именно в таком виде, хотя, разумеется, задача (1.4) является более общей.

В настоящей книге будем всегда предполагать существование глобального минимайзера, а задачу (1.5) запишем в форме *задачи нелинейного программирования*:

$$\varphi(y) \to \min, \quad y \in Q \subseteq R^N, \tag{1.6}$$

$$Q = \{y \in D\colon g_j(y) \leqslant 0,\ 1 \leqslant j \leqslant m\}, \tag{1.7}$$

$$D = \{y \in R^N\colon y_i \in [a_i, b_i],\ 1 \leqslant i \leqslant N\}, \tag{1.8}$$

т. е. задачи отыскания экстремальных (в смысле постановок A-D) значений целевой (минимизируемой) функции $f(y)$ в области Q, задаваемой *координатными* (1.8) и *функциональными* (1.7) *ограничениями* на выбор допустимых точек (векторов) $y = (y_1, y_2, \ldots, y_N)$. В данной модели величины a_i, b_i, $1 \leqslant i \leqslant N$, задающие границы изменения варьируемых параметров задачи (координат вектора y), либо константы, либо, когда соответствующая нижняя и (или) верхняя границы отсутствуют, принимаются равными $a_i = -\infty$ и (или) $b_i = +\infty$.

Довольно часто в формулировку задачи нелинейного программирования включают также ограничения в виде равенств. Однако любое равенство $h(y) = 0$, во-первых, формально можно представить в виде системы двух неравенств $h(y) \leqslant 0$ и $-h(y) \leqslant 0$. Во-вторых, при численном решении задачи оптимизации на ЭВМ точная реализуемость равенства невозможна, поэтому предполагают, что допустима его выполнимость с некоторой погрешностью $\delta > 0$, т. е. вместо равенства $h(y) = 0$ рассматривается неравенство $|h(y)| \leqslant \delta$. Таким образом, учитывая указанные обстоятельства, можно утверждать, что (1.6)–(1.8) является формулировкой *общей многомерной задачи нелинейного программирования*.

Если $m = 0$, т. е. функциональные ограничения отсутствуют, будем полагать $Q = D$. Задача (1.6)–(1.8) в этом случае будет называться задачей *безусловной оптимизации*.

В качестве примера конкретной постановки рассмотрим задачу минимизации функции

$$\varphi(y_1, y_2) = -1{,}5y_1^2 \exp\{1 - y_1^2 - 20{,}25(y_1 - y_2)^2\} - \\ - [0{,}5(y_1 - 1)(y_2 - 1)]^4 \exp\{2 - [0{,}5(y_1 - 1)]^4 - (y_2 - 1)^4\}$$

в области поиска $0 \leqslant y_1 \leqslant 4$, $-1 \leqslant y_2 \leqslant 3$, при ограничениях

$$g_1(y_1, y_2) = 0{,}01[(y_1 - 2{,}2)^2 + (y_2 - 1{,}2)^2 - 2{,}25] \leqslant 0,$$
$$g_2(y_1, y_2) = 100[1 - (y_1 - 2)^2/1{,}44 - (0{,}5y_2)^2] \leqslant 0,$$
$$g_3(y_1, y_2) = 10[y_2 - 1{,}5 - 1{,}5\sin(6{,}283(y_1 - 1{,}75))] \leqslant 0.$$

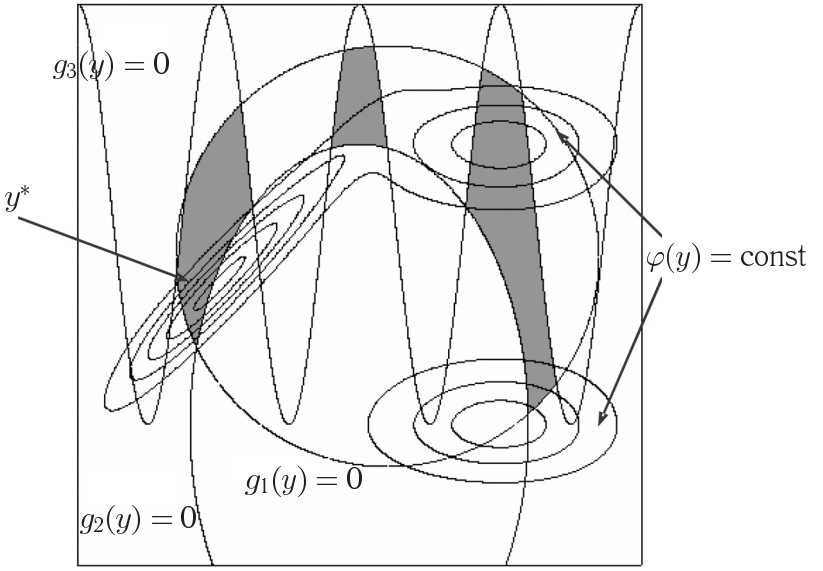

Рис. 1.1. Задача глобальной оптимизации

Допустимые по первому ограничению точки образуют круг с границей $g_1(y_1, y_2) = 0$. Допустимые по второму ограничению точки находятся во внешности эллипса $g_2(y_1, y_2) = 0$. Точки, допустимые по третьему ограничению, находятся ниже синусоиды $g_3(y_1, y_2) = 0$. Следовательно, допустимая область является неодносвязной и состоит из трех невыпуклых подобластей (на рис. 1.1 допустимая область выделена цветом). Глобальный минимум $\varphi(y_1^*, y_2^*) = -1{,}489$ достигается в точке $(y_1^*, y_2^*) = (0{,}942, 0{,}944)$.

1.2. Численные методы оптимизации

1.2.1. Понятие алгоритма. Последовательная схема

Сформулировав задачу минимизации, мы теперь должны дать ответ на основной вопрос: каким образом ее решать?

Классический подход математического анализа предлагает следующую процедуру аналитического решения задачи (на примере одномерного случая). Пусть $\varphi(y)$ — кусочно-гладкая на отрезке $[a, b]$ функция одной переменной. Тогда минимум $\varphi(y)$ на $[a, b]$ может достигаться лишь в тех точках, где $\varphi'(y) = 0$, либо производная разрывна, либо в граничных точках. Остается найти все такие точки и выбрать из них точку с наименьшим значением. Иными словами, чтобы решить задачу этим способом, требуется:

а) указание аналитического вида функции;

б) кусочная гладкость функции;

в) возможность вычисления производной;

г) умение решать уравнение $\varphi'(y) = 0$, т. е. задачу поиска корня;

д) информация о точках разрыва производной или способ определения этих точек.

К сожалению, эти требования на практике выполняются в редчайших случаях. Типичный же случай описывается ситуацией, когда функция $\varphi(y)$ задается алгоритмически, т. е. в виде некоей расчетной схемы, когда по заданному аргументу y рассчитывается значение $\varphi(y)$. В этом случае ни о каких аналитических способах исследования говорить не приходится. Заметим, что даже при аналитическом задании функции и способности посчитать производную исходная задача (1.4) сводится к решению задачи поиска корня, которая по сложности сравнима с задачей минимизации.

Узкая сфера применения аналитических методов обусловила развитие и широкое распространение *численных методов (алгоритмов)* решения задач оптимизации. Различные формулировки определений численного метода оптимизации даны многими авторами. Общим во всех формулировках является представление метода как некоторой итерационной процедуры, которая осуществляет вычисление в точках области поиска определенных характеристик минимизируемой функции (такими характеристиками могут быть зна-

чение функции, ее градиент, матрица вторых производных и т. п.). Назовем операцию вычисления характеристик функции в точке *поисковым испытанием*, а совокупность значений характеристик в этой точке — *результатом испытания*. Далее в настоящей главе в качестве результата испытания будем рассматривать только значение функции в испытуемой точке.

Если реализация алгоритма осуществляется на однопроцессорной установке, то вычислительная схема алгоритма состоит из последовательно выполняемых операций, а все испытания осуществляются последовательно, т. е. мы имеем чисто последовательный численный метод.

Основываясь на методологии теории исследования операций, дадим формальное определение последовательного алгоритма оптимизации или, более широко, *модели вычислений* при решении задачи (1.4) [15]. Обобщение данной модели на случай параллельных вычислений будет дано в п. 1.2.2.

Построение модели вычислений предполагает наличие некоторой априорной (доопытной, имеющейся до начала вычислений) информации о решаемой задаче. Данная информация может быть получена, исходя из физической сущности задачи, описывающей моделируемый реальный объект. Такими свойствами могут быть непрерывность, гладкость, монотонность, выпуклость и т. п. Имеющаяся информация служит для исследователя основанием для отнесения задачи (в нашем случае функции $\varphi(y)$) к тому или иному множеству (классу) Φ. После того, как класс Φ зафиксирован, априорная информация о задаче, используемая исследователем, состоит в том, что ему известна принадлежность задачи к классу Φ.

Следующим важным этапом построения модели вычислений является выбор *алгоритма* (*метода*) *решения задачи*. В самом общем виде численный метод s решения задачи из класса Φ представляет собой набор (кортеж) [42]

$$s = <\{G_k\}, \{E_k\}, \{H_k\}>, \tag{1.9}$$

в котором $\{G_k\}$ — совокупность правил выбора точек испытаний, $k = 1, 2, \ldots$; $\{E_k\}$ — совокупность правил построения приближен-

ного решения (оценки экстремума), $k = 1, 2, \ldots$; $\{H_k\}$ — совокупность правил остановки вычислительного процесса, $k = 1, 2, \ldots$.

Порядок проведения испытаний, или *вычислительная схема* алгоритма состоит в следующем.

1. Выбирается точка первого испытания

$$y^1 = G_1(\Phi) \in Q, \tag{1.10}$$

и номер испытания k устанавливается в единицу.

2. Пусть выбрана точка k-го испытания $y^k \in Q$ $(k \geqslant 1)$. Производится вычисление значения функции $z^k = \varphi(y^k)$. После этого имеем следующую поисковую (апостериорную) информацию о функции φ:

$$\omega_k = \{(y^1, z^1), (y^2, z^2), \ldots, (y^k, z^k)\}. \tag{1.11}$$

Полученная информация позволяет сузить класс, которому принадлежит функция $\varphi(y)$ до множества

$$\Phi(\omega_k) = \{\psi \in \Phi\colon\ \psi(y^i) = z^i,\ 1 \leqslant i \leqslant k\}. \tag{1.12}$$

3. Определяется текущая оценка экстремума (приближенное решение)

$$e^k = E_k(\Phi, \omega_k). \tag{1.13}$$

4. Вычисляется точка очередного испытания

$$y^{k+1} = G_{k+1}(\Phi, \omega_k). \tag{1.14}$$

5. Определяется величина

$$h^k = H_k(\Phi, \omega_k, y^{k+1}) \in \{0, 1\}, \tag{1.15}$$

принимающая одно из двух возможных значений: ноль или единица. Если $h^k = 1$, номер шага поиска k увеличиваем на единицу и переходим к выполнению пункта 2 схемы. Если $h^k = 0$, вычисления прекращаем и в качестве решения задачи берем оценку e^k.

Общая модель вычислений описана.

Пример. Рассмотрим простейший метод решения одномерной задачи (1.4) на отрезке $[a, b]$ — метод перебора значений по узлам равномерной сетки. Метод состоит в том, что отрезок разбивается на m равных частей, в точках (узлах) разбиения, в число которых

входят и концы отрезка, вычисляются значения функции и в качестве решения задачи рассматривается наименьшее вычисленное значение (и его координата, если это требуется соответствующей постановкой). Для применимости метода достаточно вычислимости функции в любой точке области поиска, так что в качестве априорного класса Φ можно рассмотреть класс функций, определенных на отрезке $[a, b]$ и вычислимых в каждой его точке.

Для данного метода

$$G_1(\Phi) = a, \quad G_{k+1}(\Phi, \omega_k) = a + k\frac{b-a}{m}, k \geqslant 1,$$

$$H_k(\Phi, \omega_k) = \begin{cases} 0, & k > m, \\ 1, & k \leqslant m, \end{cases}$$

$$e^k = \varphi_k^*, \tag{1.16}$$

где

$$\varphi_k^* = \min_{1 \leqslant i \leqslant k} \varphi(y^i), \tag{1.17}$$

либо

$$e^k = (\varphi_k^*, y_k^*),$$

где

$$y_k^* = \arg \min_{1 \leqslant i \leqslant k} \varphi(y^i).$$

1.2.2. Параллельные алгоритмы оптимизации

Вычислительная схема (1.10)–(1.15) предполагает последовательную реализацию всех стадий алгоритма, что может быть обеспечено вычислительной установкой с единственным процессором. Если же в нашем распоряжении имеется многопроцессорная система, то возможно распараллеливание вычислительной схемы и достижение за счет этого ускорения решения задачи. Распараллеливание можно проводить в нескольких направлениях. Во-первых, можно распараллеливать вычисление целевой функции, во-вторых, реализацию правил G_k, E_k, H_k и, в-третьих, собственно схему (1.10)–(1.15), обеспечивая *одновременное выполнение нескольких испытаний*. Однако распараллеливание целевой функции, описывающей оптимизируемый объект, специфично для разных моделей

целевых функций и к общей схеме оптимизации, вообще говоря, отношения не имеет. Распараллеливание функционалов G_k, E_k, H_k существенно зависит от конкретного класса алгоритмов и, кроме того, часто эти функционалы достаточно просты, и распараллеливать их просто нецелесообразно. Поэтому предметом рассмотрения в настоящем разделе является конструирование некоторых *основных принципов* параллельного проведения испытаний.

Будем использовать термин *итерация* для одновременного (параллельного) выполнения нескольких испытаний (каждое на своем процессоре), обозначая числом $p(n)$ количество испытаний n-й итерации, а величиной $k(n)$ — суммарное количество испытаний, выполненных в процессе всех n итераций. Иными словами, будем считать, что на n-й итерации в нашем распоряжении имеется $p(n)$ процессоров (вообще говоря, для разных итераций количество процессоров может быть различным).

Рассмотрим *синхронный* вариант распараллеливания, согласно которому переход к следующей итерации происходит, когда текущая итерация полностью заканчивается, т. е. по завершении последнего испытания итерации. В этом случае очевидно, что $k(n) = p(1) + \ldots + p(n)$, $n \geqslant 1$. Обозначим через $u(n)$ вектор координат испытаний n-й итерации, т. е.

$$u(n) = \left(y^{k(n-1)+1}, y^{k(n-1)+2}, \ldots, y^{k(n)}\right),$$

а через $u_j(n)$ — первые j координат вектора $u(n)$, т. е.

$$u_j(n) = \left(y^{k(n-1)+1}, y^{k(n-1)+2}, \ldots, y^{k(n-1)+j}\right), \quad 1 \leqslant j \leqslant p(n).$$

Тогда общая вычислительная схема синхронного параллельного алгоритма оптимизации может быть представлена следующим образом.

Схема синхронного параллельного алгоритма

1. Полагаем номер итерации $n = 1$ и величину $k(0) = 0$. Выбираем $p(1)$ точек первой итерации

$$y^1 = G_1(\Phi) \in Q, \quad y^i = G_i(\Phi, u_{i-1}(1)) \in Q, \quad 2 \leqslant i \leqslant p(1) \quad (1.18)$$

и устанавливаем количество испытаний $k(1) = p(1)$.

2. Пусть выбраны точки n-й итерации $u(n)$, $n \geqslant 1$. Производится вычисление значений функции $z^j = \varphi(y^j)$, $k(n-1)+1 \leqslant j \leqslant \leqslant k(n)$ — каждое значение отдельным процессором. После вычисления всех значений имеем следующую поисковую (апостериорную) информацию о функции φ:

$$\omega_k = \omega_{k(n)} = \{(y^1, z^1), (y^2, z^2), \ldots, (y^k, z^{k(n)})\}. \tag{1.19}$$

Как и ранее, полученная информация позволяет сузить класс, которому принадлежит функция $\varphi(y)$, до множества

$$\Phi(\omega_k) = \{\psi \in \Phi\colon\ \psi(y^i) = z^i,\ 1 \leqslant i \leqslant k(n)\}. \tag{1.20}$$

3. По завершении всех испытаний итерации определяется текущая оценка экстремума (приближенное решение)

$$e^k = E_{k(n)}(\Phi, \omega_k). \tag{1.21}$$

4. Вычисляются $p(n+1)$ координат испытаний $(n+1)$-й итерации:

$$\begin{gathered} y^{k(n)+1} = G_{k(n)+1}(\Phi, \omega_k), \\ y^i = G_i(\Phi, \omega_k, u_{i-1}(n+1)), \quad k(n)+2 \leqslant i \leqslant k(n)+p(n+1) \end{gathered} \tag{1.22}$$

и устанавливается количество шагов поиска $k(n+1) = k(n) + + p(n+1)$.

5. Определяется величина

$$h^k = H_{k(n)}(\Phi, \omega_k, u(n+1)) \in \{0, 1\}, \tag{1.23}$$

принимающая одно из двух возможных значений: ноль или единица. Если $h^k = 1$, номер итерации увеличиваем на единицу $(n = n+1)$ и переходим к выполнению пункта 2 схемы. Если $h^k = 0$, вычисления прекращаем и в качестве решения задачи берем оценку e^k.

Замечание. В предложенной схеме неявно предполагается, что заранее известно число $p(n)$ процессоров для каждой итерации алгоритма. Вместе с тем легко дополнить схему правилом выбора количества процессоров в зависимости от текущей ситуации поиска, если ввести функционал $P_n(\Phi, \omega_{k(n-1)})$, задающий число

процессоров на n-й итерации, т. е.

$$p(n) = P_n(\Phi, \omega_{k(n-1)}).$$

Тогда правило 4 вычислительной схемы запишется следующим образом.

4. Определяется количество процессоров для выполнения $(n+1)$-й итерации

$$p(n+1) = P_{n+1}(\Phi, \omega_{k(n)})$$

и вычисляются $p(n+1)$ координат испытаний данной итерации согласно (1.22).

В качестве примера параллельного алгоритма вновь рассмотрим метод полного перебора в узлах равномерной сетки и предположим для простоты, что количество p процессоров одинаково на каждой итерации, а число узлов $m > p$ и кратно p. Тогда очевидным образом решающие правила $G_s(\bullet)$ в выражениях (1.18), (1.22) при $s = 1, k(n)+1, i$ имеют вид

$$G_s(\bullet) = a + (s-1)\frac{b-a}{m},$$

а оценки экстремума и условие остановки такие же, как в последовательном прототипе.

Рассмотренная синхронная схема обладает тем недостатком, что процессоры, закончившие свои испытания в рамках итерации раньше других, будут простаивать, ожидая полного завершения итерации. Опишем вычислительную схему *асинхронного* параллельного алгоритма оптимизации, свободного от этого недостатка. В асинхронном алгоритме итерация становится распределенной во времени, поэтому удобнее оперировать с количеством испытаний k. В связи с этим введем множество $w(k) = \{w_1, w_2, \dots, w_k\}$ координат завершенных испытаний и множество $v(k) = \{v_1, v_2, \dots, v_{p(k)}\}$ координат, в которых проводятся новые испытания после завершения предшествующих k испытаний. Здесь $p(k)$ — количество процессоров, осуществляющих очередные испытания, а $w_j = y^{i_j}$, $1 \leqslant j \leqslant k$, $v_s = y^{i_s}$, $1 \leqslant s \leqslant p(k)$, $1 \leqslant i_j$, $i_s \leqslant k + p(k)$. Очевидно, что $w(k) \cup v(k) = \{y^1, \dots, y^k, \dots, y^{k+p(k)}\}$.

Предположим для простоты, что в нашем распоряжении для реализации алгоритма имеется $p \geqslant 2$ процессоров, и это число не меняется в процессе оптимизации. Вариант вычислительной схемы с различным числом процессоров на разных стадиях поиска легко может быть получен посредством обобщения предлагаемой асинхронной схемы, состоящей в следующем.

Схема асинхронного параллельного алгоритма

1. Выбираем p точек начальных испытаний

$$\begin{aligned} &y^1 = G_1(\Phi) \in Q, \\ &y^i = G_i(\Phi, y^1, \ldots, y^{i-1}) \in Q, \quad 2 \leqslant i \leqslant p, \end{aligned} \tag{1.24}$$

формируем вектор $v(0)$ с компонентами $v_i = y^i$, $1 \leqslant i \leqslant p$ и устанавливаем количество испытаний $k = 0$.

2. Ожидаем завершения работы одного или нескольких процессоров. Пусть закончили проведение испытаний $1 \leqslant \pi = \pi(k) \leqslant p$ процессоров, которые вычислили значения функции $z^{i_j} = \varphi(y^{i_j})$ в точках y^{i_1}, y^{i_2}, $\ldots$, y^{i_π}, $y^{i_j} \in v(k)$, $1 \leqslant j \leqslant \pi$. Полагаем $k = k + \pi$, формируем множества $v(k) = v(k-\pi) \backslash \{y^{i_1}, y^{i_2}, \ldots, y^{i_\pi}\}$, $w(k) = w(k-\pi) \cup \{y^{i_1}, y^{i_2}, \ldots, y^{i_\pi}\}$. После этого в нашем распоряжении находится поисковая (апостериорная) информация о функции φ

$$\omega_k = \{(y^{i_1}, z^{i_1}), (y^{i_2}, z^{i_2}), \ldots, (y^{i_k}, z^{i_k})\}, \tag{1.25}$$

где $y^{i_s} \in w(k)$, $z^{i_s} = \varphi(y^{i_s})$, $1 \leqslant s \leqslant k$.

3. Определяем текущую оценку экстремума (приближенное решение)

$$e^k = E_k(\Phi, \omega_k). \tag{1.26}$$

4. Вычисляем π точек очередных испытаний

$$\begin{aligned} &y^{k+1} = G_{k+1}(\Phi, \omega_k), \\ &y^i = G_i(\Phi, \omega_k, y^{k+1}, \ldots, y^{k+i-1}), \quad k+2 \leqslant i \leqslant k+\pi \end{aligned} \tag{1.27}$$

и добавляем их в множество $v(k)$.

5. Определяем величину

$$h^k = H_k(\Phi, \omega_k, y^{k+1}, \ldots, y^{k+\pi}) \in \{0, 1\}, \tag{1.28}$$

принимающую одно из двух возможных значений: ноль или единица. Если $h^k = 1$, переходим к выполнению пункта 2 схемы. Если $h^k = 0$, вычисления прекращаем и в качестве решения задачи берем оценку e^k.

Конкретные варианты параллельных алгоритмов и результаты экспериментов с их использованием будут описаны в разделе 2.5.

1.2.3. Сходимость и оценки экстремума

Итак, решая задачу минимизации функции $\varphi(y)$, метод поиска порождает (сопоставляет функции) последовательность y^1, y^2, $\ldots$, y^k, $\ldots$ координат испытаний, или просто последовательность испытаний (y^k — координата k-го испытания), а также последовательность z^1, z^2, $\ldots$, z^k, $\ldots$ результатов испытаний (напомним, что мы ограничились случаем значения функции в качестве результата, т. е. $z^i = \varphi(y^i)$). При этом свойства метода определяются свойствами последовательностей $\{y^k\}$ и $\{z^k\}$, поэтому исследование метода поиска может быть проведено посредством изучения последовательностей испытаний, им порождаемых.

В связи с этим зададимся вопросом: какие требования должны быть предъявлены к последовательности испытаний численного метода оптимизации? Разумеется, основное требование заключается в том, что проведение испытаний в точках $\{y^k\}$ должно обеспечить на основе результатов $\{z^k\}$ решение задачи, т. е. отыскание решения, соответствующего выбранной постановке. При этом, поскольку вычислитель может осуществить лишь конечное число испытаний, желательно получить точное решение, построив конечную последовательность $\{y^k\}$. К сожалению, такая приятная ситуация имеет место лишь в редких и достаточно простых случаях, например, в задачах линейного программирования. Поэтому часто интересуются асимптотически точной оценкой, рассматривая бесконечную последовательность испытаний (в модели (1.9) $H_k = 1$ для любого k, т. е. условие остановки отсутствует) и требуя, чтобы эта последовательность сходилась к точному решению задачи. Поскольку в постановках B-D искомое решение может содержать *несколько* точек минимума, сходимость метода будем понимать в смысле следующего определения.

Определение 1.4. Последовательность испытаний $\{y^k\}$ сходится к решению задачи (1.4), определенному соответствующей постановкой, если:

1) она содержит подпоследовательность $\{\overline{y}^k\}$, для которой $\lim\limits_{k\to\infty} \varphi(\overline{y}^k) = \varphi^*$;

2) в случае, когда решение включает одну или несколько точек минимума, для каждой такой точки существует сходящаяся к ней подпоследовательность последовательности $\{y^k\}$.

Последовательность испытаний, сходящуюся к точному решению постановки Z $\in$ {A, B, C, D, E}, будем называть *минимизирующей* последовательностью для постановки Z, либо Z-минимизирующей последовательностью. Термин «минимизирующая последовательность» введен в [3] и соответствует понятию A-минимизирующей последовательности.

Остановим внимание читателя на следующем важном аспекте. Вопросам сходимости в теории методов поиска экстремума уделяется значительное внимание, поскольку асимптотика обеспечивает потенциальную возможность получения точного решения с любой наперед заданной точностью за конечное число испытаний. Но самой по себе такой возможности для практической реализации методов недостаточно. Необходимо еще уметь определять меру близости получаемого приближенного решения к точному решению, т. е. уметь оценивать погрешность решения задачи при конечном числе испытаний.

Рассмотрим следующий простой пример. Пусть класс Φ — класс непрерывных одномерных функций, заданных на отрезке $Q = [a, b]$, т. е. априорно известно, что минимизируемая функция $\varphi(y)$ непрерывна в области Q. Предположим, что вычислены значения функции $\varphi(y)$ в конечном числе точек $y^1, y^2, \ldots, y^k$. Что после этого можно сказать о координате глобального минимума? Каковы бы ни были точки $y^1, y^2, \ldots, y^k$ и значения $z^1, z^2, \ldots, z^k$, для любой точки $y^* \in Q$ ($y^* \notin \{y^1, \ldots, y^k\}$) всегда можно построить непрерывную функцию, проходящую через точки (y^i, z^i), $1 \leqslant i \leqslant k$, т. е. принадлежащую классу $\Phi(\omega_k)$ из (1.12), которая имеет глобальный

минимум в точке y^* с любым наперед заданным значением

$$\varphi^* < \min_{1 \leqslant i \leqslant k} z^i.$$

Например, в качестве такой функции можно взять интерполяционный полином k-й степени, проходящий через точки (y^i, z^i), $1 \leqslant i \leqslant k$, и точку (y^*, φ^*).

Все сказанное означает, что по результатам конечного числа испытаний никаких выводов о расположении координаты глобального минимума сделать нельзя. Точно так же о величине φ^* глобального минимума можно лишь сказать, что

$$\varphi^* \leqslant \varphi_k^*,$$

где φ_k^* из (1.17), однако оценить величину

$$\varepsilon_k = |\varphi^* - \varphi_k^*|,$$

т. е. погрешность решения задачи, невозможно.

Возможность получения оценок экстремума по конечному числу испытаний зависит от свойств класса функций, которому принадлежит минимизируемая функция, или, другими словами, от априорной информации о функции $\varphi(y)$. Фактически известны лишь два широких класса функций, допускающих построение таких оценок: класс одномерных унимодальных функций и класс функций, удовлетворяющих условию Липшица (в общем случае многомерных и многоэкстремальных).

Наличие оценок экстремума, построенных по конечной (усеченной) последовательности испытаний, позволяет в некоторых случаях сформулировать и решить проблему теоретического конструирования оптимальных в том или ином смысле методов оптимизации (см., например, [32, 33, 42, 48]). Эта интересная проблематика выходит за рамки настоящей книги, однако в следующих разделах наличие оценок экстремума является важным для исследования эффективности распараллеливания рассматриваемых методов оптимизации.

1.3. Многоэкстремальные задачи и редукция сложности

Напомним, что предметом рассмотрения настоящей книги являются многоэкстремальные задачи оптимизации, т. е. задачи, в которых целевая функция $\varphi(y)$ может иметь в допустимой области Q несколько локальных экстремумов. На сложность решения таких задач существенное влияние оказывает размерность. Например, для класса многоэкстремальных функций, удовлетворяющих условию Липшица, имеет место так называемое «проклятие размерности», состоящее в экспоненциальном росте вычислительных затрат при увеличении размерности. А именно: если в одномерной задаче для достижения точности решения ε требуется p вычислений функции, то в задаче с размерностью N для решения с той же точностью необходимо осуществить αp^N испытаний, где α зависит от целевой функции, допустимой области и используемого метода.

Для специальных узких классов многоэкстремальных задач порядок роста затрат может быть и лучше экспоненциального. Например, для сепарабельных функций, минимизируемых в гиперинтервале D, затраты растут линейно. Этот факт иллюстрирует то обстоятельство, что улучшить эффективность решения можно только на основе глубокого учета априорной информации о задаче.

Формой такого учета может быть построение эффективных методов оптимизации как оптимальных решающих правил на основе минимаксного или байесовского подходов к понятию оптимальности метода поиска экстремума в рамках соответствующих математических моделей. К сожалению, для многомерных многоэкстремальных задач проблема построения оптимальных алгоритмов является очень сложной и связана с решением задач теории покрытий [28, 30, 48].

Другой подход к конструированию численных методов анализа многомерных многоэкстремальных задач использует идею *редукции сложности*, когда решение исходной задачи заменяется решением одной или нескольких более простых задач.

Одна из таких схем редукции базируется на том элементарном факте, что глобальный экстремум является локальным. Отсюда следует прозрачный вывод, что для нахождения глобально-опти-

мального решения достаточно найти все локальные минимумы и из них выбрать наименьший. В контексте теоретического подхода эта конструкция безупречна, однако на практике не все так безоблачно.

Прежде всего, возникает вопрос: как найти все локальные минимумы? Эта задача может быть решена, если для каждого локального минимума известна зона его притяжения, т. е. такая окрестность точки минимума, в которой функция одноэкстремальна. Тогда, поместив начальную точку поиска в эту окрестность, можно тем или иным локальным методом найти искомый локальный минимум. Другими словами, данная схема предполагает известным разбиение области поиска на зоны притяжения локальных минимумов. Однако на практике подобная априорная информация, как правило, отсутствует (даже количество локальных экстремумов обычно не известно). Поэтому при таком подходе возникает дополнительная задача выбора начальных точек.

Простейший способ состоит в том, чтобы выбирать начальные точки по схеме метода Монте-Карло [24, 27], т. е. случайно в соответствии с некоторым распределением в области поиска (обычно равномерным), либо использовать в качестве таких точек узлы некоторой регулярной сетки [41].

При таком способе сходимость к глобальному экстремуму обеспечивается тем обстоятельством, что при увеличении числа начальных точек хотя бы одна из них попадет в зону притяжения глобального экстремума, так как эти сетки (регулярная, либо случайная) строятся так, чтобы обеспечить уплотняющееся покрытие всей области поиска.

У этой схемы есть, однако, существенный недостаток. Дело в том, что в зону притяжения одного и того же локального минимума могут попасть несколько начальных точек, т. е. придется несколько раз искать один и тот же локальный минимум. Для устранения этого недостатка предложено несколько различных схем.

Идея одной из них состоит в следующем. Вначале в области поиска выбираются L базовых точек (обычно более или менее равномерно расположенных в области поиска) и затем вычисляются значения функции в этих точках, т. е., по существу, производится

грубая оценка поведения функции. Из множества базовых точек производится отбор $l < L$ «лучших» точек, т. е. таких, в которых значения функции минимальны и точки не слишком близки друг к другу. Эти l точек и служат начальными точками локального поиска. Существует множество вариантов этой схемы. Например, множество базовых точек может модифицироваться в процессе поиска — туда могут добавляться новые точки. Также в процессе реализации этой схемы может изменяться способ отбора точек с учетом вновь поступившей информации и т. п.

Существуют и другие схемы, построенные на основе редукции многоэкстремальной задачи к решению локальных подзадач. Алгоритмы, сконструированные в рамках данного подхода, обладают асимптотической сходимостью к глобальному экстремуму при слабых предположениях о непрерывности или той или иной степени дифференцируемости целевой функции. Эта сходимость обеспечивается тем свойством алгоритмов, что любая точка области поиска является предельной точкой последовательности испытаний $\{y^k\}$, порождаемой алгоритмом, и, следовательно, для непрерывной функции

$$\lim_{\tau\to\infty} \min_{1\leqslant k\leqslant \tau} \varphi(y^k) = \min_{y\in Q} \varphi(y). \tag{1.29}$$

Другими словами, последовательность испытаний этих методов является всюду плотной в области поиска, т. е. сходится в смысле определения 1.4 ко всем точкам этой области, а следовательно, и к точке глобального минимума.

Вообще говоря, указанный характер сходимости является слишком «расточительным», что не способствует эффективности методов данного типа. Избежать указанного недостатка можно только при наличии достаточной и довольно богатой априорной информации о задаче (типа сведений об областях притяжения). Такая информация довольно редко имеет место на практике и, как правило, может быть получена только для простых многоэкстремальных задач с небольшим числом экстремумов, а для существенно многоэкстремальных задач применение идеи сведения к задачам на локальный минимум не слишком успешно. Вместе с тем, методы данного класса довольно легко распараллеливаются, однако всюду

плотная сходимость (1.29) все равно приводит к существенной избыточности испытаний.

По сравнению с редукцией к локальным задачам, более плодотворным является подход, основанный на идеях *редукции размерности*, т. е. построение таких схем, когда решение многомерной задачи сводится к решению одной или нескольких одномерных подзадач. Известны две эффективные схемы такого рода: *многошаговая схема* оптимизации [18, 29, 67] и редукция размерности на основе кривых, заполняющих пространство (кривых Пеано) [42, 67].

Многошаговая схема редукции размерности сводит решение задачи (1.6)–(1.8) к решению семейства рекурсивно связанных одномерных подзадач, в которых каждое вычисление целевой функции одномерной подзадачи есть решение новой одномерной подзадачи следующего уровня рекурсии, за исключением последнего уровня, где вычисляется значение исходной многомерной целевой функции.

Вторая из указанных схем — *редукция размерности на основе кривых Пеано* — основана на известном фундаментальном факте, согласно которому N-мерный гиперпараллелепипед (1.8) и отрезок $[0, 1]$ вещественной оси являются равномощными множествами, и отрезок $[0, 1]$ может быть однозначно и непрерывно отображен на гиперпараллелепипед (1.8) [42]. Отображения такого рода обычно называют *кривыми*, или *развертками Пеано*.

Пусть $y(x)$, $x \in [0, 1]$, есть кривая Пеано, и функция $\varphi(y)$ из (1.6) непрерывна. Тогда из непрерывности $\varphi(y)$ и $y(x)$ и равенства

$$D = \{y(x)\colon 0 \leqslant x \leqslant 1\}$$

следует, что

$$\min_{y \in D} \varphi(y) = \min_{x \in [0,1]} \varphi(y(x)),$$

т. е. решение многомерной задачи минимизации $\varphi(y)$ сводится к минимизации одномерной функции $\varphi(y(x))$.

Рассмотрение и обоснование параллельных алгоритмов на основе многошаговой схемы оптимизации и схем редукции размерности на основе кривых Пеано станет основным содержанием последующих глав.

Глава 2

Алгоритмы глобального поиска для одномерных многоэкстремальных задач оптимизации

2.1. Информационно-статистический подход к глобальному поиску

Теоретическим базисом для создания алгоритмов поиска глобального экстремума, рассматриваемых в этой книге, является информационно-статистический подход, предложенный Р. Г. Стронгиным [42]. Сначала мы дадим краткое описание общей методики вывода методов в рамках данного подхода, а затем представим простые алгоритмические схемы как результат применения методики.

Согласно формируемой в рамках подхода модели вычислений, для задачи (1.4) предполагается, что на классе измеримых подмножеств множества Φ всех функций φ, определенных в области Q, задано распределение вероятностей F. Принятие данной модели означает, что минимизируемая функция φ рассматривается как реализация некоторого случайного процесса, т. е. как функция $\varphi = \varphi(y, \sigma)$ двух аргументов, где $y \in Q$, а σ является случайной величиной, свойства которой определяются некоторым распределением F.

Любая дополнительная информация о функции φ — как априорная, так и апостериорная — может быть оценена в вероятностном смысле как вероятность $F(\Phi')$ принадлежности функции φ к множеству Φ' функций, обладающих свойствами в соответствии с данной информацией. В частности, априорное распределение F может быть пересчитано в апостериорное по отношению к поисковой информации (1.11) как $F_\omega = F(\Phi(\omega_k))$. Данное распределение позволяет оценивать расположение экстремума, прогнозировать результат очередного испытания и конструировать алгоритмы опти-

мизации на основе статистических решающих правил. Детальное описание методологии дано в монографиях [42] и [67], мы же кратко изложим применение подхода к одномерной задаче (1.4) в дискретной постановке.

Итак, рассмотрим на вещественной оси некоторый отрезок $[a, b]$. Чтобы подчеркнуть, что исследуемая задача является одномерной, вместо аргумента y целевой функции введем переменную x. Зададим на отрезке $[a, b]$ конечную равномерную ε-сеть

$$x_0 = a < x_1 = a + \varepsilon < \ldots < x_i = a + i\varepsilon < \ldots < x_n = b \qquad (2.1)$$

и введем множество номеров узлов сети (2.1)

$$U = \{0, \ldots, n\}. \qquad (2.2)$$

Допустим, что в задаче (1.4) множество $Q = \{x_i,\ i \in U\}$. Тогда требуется определить узел x_ν, в котором функция $\varphi(y)$, заданная в области Q, достигает глобального минимума, т. е.

$$\varphi(x_\nu) \leqslant \varphi(x_i), \quad i \in U. \qquad (2.3)$$

В данной модели любая функция $\varphi(x)$ полностью задается величинами $\varphi_i = \varphi(x_i)$, $0 \leqslant i \leqslant n$, т. е. ее можно представить в виде вектора

$$\varphi = (\varphi_0, \ldots, \varphi_n) \in R^{n+1} \qquad (2.4)$$

$(n+1)$-мерного евклидова пространства $R^{\tau+1}$. Тогда априорные предположения о минимизируемой функции $\varphi \in R^{n+1}$ могут быть описаны посредством задания плотности $p(\varphi)$ распределения вероятностей на пространстве R^{n+1}.

Если плотность $p(\varphi)$ положительна и непрерывна, то с вероятностью единица минимум оптимизируемой функции достигается в единственной точке, поэтому для его оценки достаточно подсчитать вероятности $\eta(\alpha)$ расположения минимума в точках α, совпадающих с узлами сетки. Более того, после проведения испытаний в некоторых узлах сетки и получения поисковой информации ω_k возможно получение условных (апостериорных) вероятностей $\eta(\alpha/\omega_k)$, обеспечивающих текущие оценки экстремума.

К сожалению, вычисление данных вероятностей требует интегрирования по сложным многомерным областям, поэтому предлага-

ется аппарат приближенного описания апостериорных вероятностей $\eta(\alpha/\omega_k)$ с помощью введения дополнительной дискретной случайной величины α (называемой «состоянием природы»), распределение вероятностей $\xi(\alpha)$ которой на узлах сетки связано с априорным вероятностным описанием $p(\varphi)$ с помощью разложения

$$p(\varphi) = \sum_{\alpha \in U} p(\varphi/\alpha)\xi(\alpha). \tag{2.5}$$

Введение такого разложения позволяет установить основанные на методе малого параметра условия близости апостериорных вероятностей $\eta(\alpha/\omega_k)$ и $\xi(\alpha/\omega_k)$ и тем самым перейти к вычислению условных вероятностей состояния природы, которые существенно проще.

В качестве конкретного примера применения информационно-статистического подхода рассмотрим вероятностное описание, в котором априорная плотность $p(\varphi)$ представлена разложением $p(\varphi/\alpha)$, $0 \leqslant \alpha \leqslant n$, где

$$p(\varphi/\alpha) = \prod_{i=0}^{n} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left\{-\sum_{i=0}^{n} \frac{(\Delta\varphi_i - m_i(\alpha))^2}{2\sigma_i^2}\right\}. \tag{2.6}$$

Здесь

$$\Delta\varphi_i = \varphi_i - \varphi_{i-1}, \quad 1 \leqslant i \leqslant n, \quad \Delta\varphi_0 = \varphi_0,$$
$$\sigma_i = cm, \quad 1 \leqslant i \leqslant n, \quad c > 0, \quad m > 0,$$
$$m_i(\alpha) = \begin{cases} -m, & i \leqslant \alpha, \\ m, & i > \alpha, \end{cases} \quad 1 \leqslant i \leqslant n.$$

Согласно данному описанию, первые разности $\Delta\varphi_i$ функции и значение φ_0 этой функции в точке x_0 при любом фиксированном α являются реализациями независимых случайных величин, подчиненных нормальному закону со стандартами $\sigma_i = cm$, $1 \leqslant i \leqslant n$, σ_0 и математическими ожиданиями $m_i(\alpha)$, $1 \leqslant i \leqslant n$, m_0.

Принятые предположения являются некоторым вероятностным аналогом условия Липшица, которое ограничивает первые разности минимизируемой функции и обеспечивает ее равномерную непрерывность. В самом деле, согласно описанию дисперсии раз-

ностей $\Delta\varphi_i$, функции φ ограничены, и, кроме того, эти разности являются реализациями независимых нормальных величин, что при $mn = \text{const}$ обеспечивает вероятностное свойство, аналогичное равномерной непрерывности [16].

Кроме того, априорное математическое ожидание

$$\mu_\alpha(i) = m_0 - m \times \begin{cases} i, & i \leqslant \alpha, \\ 2\alpha - i, & i > \alpha, \end{cases}$$

значений φ_i функции φ при фиксированном α имеет единственный локальный минимум, т. е. функции, близкие к константе, являются маловероятными.

В рассматриваемой модели возможно исключение параметров m_0 и σ_0, задающих среднее значение и дисперсию величины φ_0, при условии проведения нулевого испытания в узле с номером $i_0 = 0$. При этом возникает определенная несимметричность из-за того, что значение функции изначально определено на левой границе. Для преодоления данной несимметричности следующее испытание проводится на правом конце сети, т. е. в точке $i_1 = n$.

Пусть теперь проведены испытания в узлах сетки с номерами

$$a = i_0 < i_1 < i_2 < \ldots < i_{k-1} < b = i_k, \tag{2.7}$$

и получена информация

$$\omega = \omega_k = \{(i_s, z_s),\ 0 \leqslant s \leqslant k\}, \tag{2.8}$$

где $z_s = \varphi(i_s)$, $0 \leqslant s \leqslant k$.

Тогда апостериорные вероятности состояния природы могут быть выражены как

$$\xi(\alpha/\omega_k) = B \cdot h(\omega_k, \alpha)\xi(\alpha),$$

где B — константа, а для любого α из отрезка $i_{s-1} \leqslant \alpha \leqslant i_s$, $1 \leqslant \leqslant s \leqslant k$,

$$h(\omega_k, \alpha) = \exp\left\{-\frac{(\alpha - \alpha_s^*)^2}{2\rho_s^2}\right\} \exp\left\{\frac{R(s)}{2mc^2}\right\},$$

$$\alpha_s^* = \frac{i_s - i_{s-1}}{2} - \frac{z_s - z_{s-1}}{2m}, \tag{2.9}$$

$$R(s) = m\Delta_s + \frac{(z_s - z_{s-1})^2}{m\Delta_s} - 2(z_s + z_{s-1}), \tag{2.10}$$

$$\rho_s^2 = \left(\frac{c}{2}\right)^2 \Delta_s, \quad \Delta_s = i_s - i_{s-1}.$$

Пусть распределение $\xi(\alpha)$ является равномерным ($\xi(\alpha) = 1/(n+1)$), т. е. все исходные состояния природы равновероятны, и параметр m удовлетворяет условию

$$m > \max_{1 \leqslant s \leqslant k} \left| \frac{z_s - z_{s-1}}{\Delta_s} \right|, \tag{2.11}$$

тогда при достаточно малом c наивероятнейшее состояние природы достигается в точке интервала (i_{t-1}, i_t), для которого величина $R(t)$, которую назовем *характеристикой интервала*, максимальна и совпадает с наивероятнейшей точкой расположения глобального минимума. Эта точка вычисляется как

$$\alpha^* = E(\alpha_t^* + 0.5), \tag{2.12}$$

где E является функцией вычисления целой части аргумента. Другими словами, α^* является целым числом, ближайшим к величине α_t^* из (2.9).

Отметим важный аспект, относящийся к параметру m и связанный с условием (2.11). Чтобы добиться выполнения данного условия, будем использовать следующую адаптивную схему оценки параметра m по результатам испытаний (2.8):

$$m = \begin{cases} rM, & M > 0, \\ 1, & M = 0, \end{cases} \tag{2.13}$$

где

$$M = \max_{1 \leqslant s \leqslant k} \frac{|z_s - z_{s-1}|}{i_s - i_{s-1}}, \tag{2.14}$$

а коэффициент r считается параметром метода и удовлетворяет условию $r > 1$.

Таким образом, чтобы продолжить поиск минимума после проведения испытаний в узлах (2.7), естественно выбрать новый узел с наибольшей вероятностью расположения искомого экстремума, т. е. точку (2.12). Но для ее нахождения достаточно вычислить

характеристики (2.10) и выбрать из них наибольшую. Тем самым сложная задача прямого вычисления наивероятнейшего расположения минимума через многомерное интегрирование сведена к простой задаче вычисления характеристик (2.10).

2.2. Базовый алгоритм глобального поиска

Рассмотрим одномерную задачу минимизации функции на отрезке:

$$\varphi(x) \to \min, \quad x \in Q = [a, b]. \tag{2.15}$$

Распространим описанный в предыдущем разделе дискретный алгоритм на непрерывный случай задачи (2.15), неограниченно увеличивая число узлов ($n \to \infty$) так, чтобы в пределе был получен конечный интервал $Q = [a, b]$. При таком переходе трансформируем дискретный алгоритм в его предельный непрерывный вариант следующим образом.

В соотношениях (2.10), (2.14) номера i_s, $0 \leqslant s \leqslant k$ из (2.8) заменим соответствующими им узлами сети (2.1)

$$x_{i_s} = a + i_s((b-a)/n), \quad 0 \leqslant s \leqslant k, \tag{2.16}$$

и с целью упрощения в дальнейшем узел x_{i_s} будем обозначать через x_s. С учетом преобразования (2.16), коэффициент $\widetilde{m}$ из (2.13) для непрерывного алгоритма связан с исходным параметром m дискретного прототипа соотношением $\widetilde{m} = m(n/(b-a))$, что означает неизменность значений характеристик $R(s)$ в (2.10).

Замена (2.16) в (2.9) приводит к выражению для точки нового $(k+1)$-го испытания:

$$x^{k+1} = a + \alpha_t^*((b-a)/n) = \frac{1}{2}(x_t + x_{t-1}) - \frac{z_t - z_{t-1}}{2\widetilde{m}}. \tag{2.17}$$

Заметим, что точка $x_{\alpha^*} = a + \alpha^*((b-a)/n)$, где α^* из (2.12) является узлом сетки (2.1), ближайшим к точке (2.17). Это означает, что при $n \to \infty$ точка x_{α^*} переходит в точку x^{k+1}.

Сформулированный «непрерывный» алгоритм будем называть *базовым алгоритмом глобального поиска*, или для краткости АГП.

2.2.1. Вычислительная схема алгоритма

Дадим детальное описание вычислительной схемы АГП, применяемого к решению задачи (2.15), рассматривая при этом в качестве поисковой информации множество

$$\omega = \omega_k = \{(x_i, z_i),\ 1 \leqslant i \leqslant k\}. \tag{2.18}$$

Согласно алгоритму, два первых испытания проводятся на концах отрезка $[a, b]$, т. е. $x^1 = a$, $x^2 = b$, вычисляются значения функции $z^1 = \varphi(a)$, $z^2 = \varphi(b)$, и количество k проведенных испытаний полагается равным 2.

Пусть проведено $k \geqslant 2$ испытаний и получена информация (2.18). Для выбора точки x^{k+1} нового испытания необходимо выполнить следующие действия.

1. Перенумеровать нижним индексом (начиная с нулевого значения) точки x^i, $1 \leqslant i \leqslant k$, из (2.18) в порядке возрастания, т. е.

$$a = x_0 < x_1 < \ldots < x_{k-1} = b.$$

2. Полагая $z_i = \varphi(x_i)$, $1 \leqslant i \leqslant k$, вычислить величину

$$M = \max_{1 \leqslant i \leqslant k-1} \left| \frac{z_i - z_{i-1}}{x_i - x_{i-1}} \right|$$

и положить

$$m = \begin{cases} rM, & M > 0, \\ 1, & M = 0, \end{cases} \tag{2.19}$$

где $r > 1$ является заданным параметром метода.

3. Для каждого интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k-1$ вычислить характеристику

$$R(i) = m(x_i - x_{i-1}) + \frac{(z_i - z_{i-1})^2}{m(x_i - x_{i-1})} - 2(z_i + z_{i-1}).$$

4. Найти интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i) : 1 \leqslant i \leqslant k-1\} \tag{2.20}$$

(в случае нескольких интервалов, характеристики которых удовлетворяют (2.20), выбирается интервал с наименьшим номером t).

5. Провести новое испытание в точке

$$x^{k+1} = \frac{1}{2}(x_t + x_{t-1}) - \frac{z_t - z_{t-1}}{2m},$$

вычислить значение $z^{k+1} = \varphi(x^{k+1})$ и увеличить номер шага поиска на единицу: $k = k + 1$.

Операции пунктов 1–5 описывают решающее правило АГП (1.14) из общей схемы (1.9).

Правило остановки (1.15) задается в форме

$$H_k(\Phi, \omega_k) = \begin{cases} 0, & x_t - x_{t-1} \leqslant \varepsilon, \\ 1, & x_t - x_{t-1} > \varepsilon, \end{cases} \tag{2.21}$$

где $\varepsilon > 0$ — заданная точность поиска (по координате).

Наконец, в качестве оценки экстремума выбирается пара

$$e^k = (\varphi_k^*, x_k^*),$$

где φ_k^* — минимальное вычисленное значение функции, т. е.

$$\varphi_k^* = \min_{1 \leqslant i \leqslant k} \varphi(x^i),$$

а x_k^* — координата этого значения:

$$x_k^* = \arg \min_{1 \leqslant i \leqslant k} \varphi(x^i).$$

Далее будет проведено теоретическое исследование сходимости АГП; результаты численных экспериментов приведены в п. 2.5.

2.2.2. Условия сходимости и оценка эффективности

Приведем основные теоретические результаты относительно свойств базового информационно-статистического алгоритма. Обоснование этих свойств будет дано в следующем пункте как следствие основных положений характеристической теории алгоритмов оптимизации.

Рассмотрим бесконечную последовательность испытаний $\{x^k\} = x^1, x^2, \ldots, x^k, \ldots$, порождаемую АГП в отсутствие условия остановки. Поскольку $\{x^k\} \subset [a, b]$, данная последовательность обязательно содержит хотя бы одну сходящуюся подпоследовательность.

Пусть x^+ является точкой накопления (пределом некоторой сходящейся подпоследовательности) последовательности $\{x^k\}$. Тогда:

1) если $a < x^+ < b$, последовательность испытаний содержит две подпоследовательности, одна из которых сходится к x^+ слева, другая справа;

2) если функция $\varphi(x)$ имеет на отрезке $[a, b]$ конечное число локальных экстремумов, то точка x^+ является точкой локального минимума;

3) если наряду с x^+ существует другая предельная точка $x^{++} \neq \neq x^+$ последовательности $\{x^k\}$, то $\varphi(x^+) = \varphi(x^{++})$;

4) для любого $k \geqslant 1$ выполняется $\varphi(x^k) \geqslant \varphi(x^+)$;

5) если функция $\varphi(x)$ удовлетворяет на отрезке $[a, b]$ условию Липшица с константой $L > 0$, т. е.

$$|\varphi(x') - \varphi(x'')| \leqslant L|x' - x''|, \quad x',\, x'' \in [a, b], \tag{2.22}$$

и, начиная с некоторого шага поиска, справедливо неравенство

$$m > 2L, \tag{2.23}$$

тогда любая точка глобального минимума функции $\varphi(x)$ на отрезке $[a, b]$ является предельной (в смысле определения 1.4) точкой последовательности испытаний АГП.

Следствием пунктов 3 и 5 является утверждение, что при минимизации липшицевой целевой функции (2.22) при выполнении условия (2.23) АГП будет иметь в качестве предельных все точки глобального минимума и только такие точки. Это означает, что метод строит существенно неравномерную сетку испытаний в области поиска. Чтобы оценить данную неравномерность, характеризующую эффективность размещения испытаний алгоритма, введем понятие плотности испытаний как меры подобной эффективности.

Определение 2.1. *Плотностью последовательности* $\{x^k\}$ в подынтервале $[\alpha, \beta] \subset [a, b]$ назовем величину

$$P_{\alpha\beta} = \frac{K_{\alpha\beta}}{\beta - \alpha},$$

где $K_{\alpha\beta}$ — количество точек последовательности $\{x^k\}$, принадлежащих отрезку $[\alpha, \beta]$.

Допустим, что в предположении липшицевости (2.22) функция $\varphi(x)$ на отрезке $[\alpha, \beta]$ удовлетворяет неравенству

$$\varphi(x) \geqslant \varphi(x^*) + \Delta, \quad x \in [\alpha, \beta] \subset [a, b], \tag{2.24}$$

где $\Delta > 0$ и x^* является точкой глобального минимума $\varphi(x)$ на $[a, b]$. Если при этом $\beta - \alpha > 2\Delta/L$ и выполнено условие $m \geqslant 2L$, то [42]

$$P_{\alpha\beta} \leqslant \frac{3m}{2\Delta}. \tag{2.25}$$

Заметим, что на подынтервале $[\alpha, \beta]$, удовлетворяющем (2.24), глобальный минимум располагаться не может, а это означает вследствие утверждения 3, что данный подынтервал может содержать лишь конечное число испытаний. Соотношение (2.25) дает количественную оценку данной величины.

Более того, соотношение (2.25) позволяет сравнить плотности испытаний АГП и метода простого перебора. Если минимальное значение $\varphi(x^*)$ задачи (2.15) с липшицевой функцией (2.22) отыскивается с точностью $\gamma > 0$ (по значению функции) методом перебора в узлах равномерной сетки, то шаг этой сетки для достижения точности γ не должен превосходить величины $2\gamma/L$. В этом случае плотность испытаний $\widetilde{P}_{\alpha\beta}$ переборной схемы на отрезке $[\alpha, \beta]$ будет удовлетворять неравенству

$$\widetilde{P}_{\alpha\beta} \geqslant \frac{L}{2\gamma}. \tag{2.26}$$

Считая, что константа Липшица L задана, положим для АГП $m = 2L$. Тогда из (2.25) и (2.26) получаем неравенство

$$P_{\alpha\beta} \leqslant \left(\frac{6\gamma}{\Delta}\right) \widetilde{P}_{\alpha\beta}. \tag{2.27}$$

Из (2.27) следует, что чем выше требуемая точность решения задачи γ, тем выше степень «экономичности» АГП по сравнению с перебором.

Приведем также теоретическое сравнение АГП с методом ломаных (МЛ), предложенным С. А. Пиявским [36] для минимизации липшицевых функций (2.22). Оба метода в качестве своего параметра используют некоторую оценку m константы Липшица L.

Предположим, что данная оценка одинакова для обоих методов, и на отрезке $[\alpha, \beta]$ целевая функция линейна, тогда [20] имеет место соотношение

$$P_{\alpha\beta} \leqslant 2\widehat{P}_{\alpha\beta},$$

где $\widehat{P}_{\alpha\beta}$ — плотность МЛ на отрезке $[\alpha, \beta]$.

Сложность исследуемой задачи и применяемых методов позволяет получить теоретические оценки эффективности и провести теоретическое сравнение алгоритмов в достаточно редких случаях. В этой ситуации важным элементом исследования алгоритмов становится вычислительный эксперимент. Авторами была предложена [19, 42] методика сравнения алгоритмов оптимизации, основанная на построении операционных характеристик методов.

Приведем краткое описание данного подхода к сопоставлению эффективности алгоритмов и продемонстрируем экспериментальные результаты сравнения нескольких известных методов глобальной оптимизации.

Эффективность любого алгоритма оптимизации неразрывно связана с двумя показателями: затратами на поиск к моменту остановки и точностью решения задачи, определяемой оценкой экстремума e^k из (1.13). Предположим, что затраты на поиск измеряются количеством испытаний K, выполненным методом до момента остановки, а погрешность — величиной $P = \varphi_K^* - \varphi^*$, где φ_k^* из (1.17) при $k = K$ (минимальное вычисленное значение функции за K шагов), а φ^* из (1.3) (минимальное значение целевой функции в задаче (1.4)). Заметим, что показатели K и P являются противоречивыми: для уменьшения погрешности приходится увеличивать затраты, а при уменьшении числа испытаний ухудшается точность решения задачи. Иными словами, вводя данные показатели, мы формируем двухкритериальную задачу оптимизации с противоречивыми критериями.

Выполнив численное решение конкретной задачи оптимизации при заданном наборе параметров метода, мы получим конкретные значения показателей, которые удобно изобразить на плоскости точкой с координатами (K, P). Множество таких точек, соответ-

ствующих различным значениям параметров алгоритма, называется *операционной характеристикой* метода.

Полученная таким образом операционная характеристика отражает итоги оптимизации только одной функции. Существенно более информативными являются результаты, относящиеся к некоторому классу функций. Расширим понятие операционной характеристики на данный случай.

Рассмотрим класс Ψ функций ψ и будем предполагать, что на этом классе задано распределение вероятностей $F(\psi)$. В соответствии с данным распределением выберем n функций $\psi_i \in \Psi$, $1 \leqslant i \leqslant n$ и проведем их оптимизацию при заданном наборе параметров метода, затратив при этом для i-й функции K_i испытаний и получив для нее погрешность P_i. В качестве критериев эффективности эксперимента возьмем среднее число испытаний

$$K = \sum_{i=1}^{n} K_i/n$$

и вероятность P того, что к моменту остановки экстремум найден с заданной точностью $\delta > 0$, т. е.

$$P = \sum_{i=1}^{n} p_i/n,$$

где

$$p_i = \begin{cases} 1, & P_i \leqslant \delta, \\ 0, & P_i > \delta. \end{cases}$$

Вероятность P будем называть *надежностью* метода.

Выполнив оптимизацию функций выборки при различных параметрах метода, мы получим (приближенную) операционную характеристику алгоритма на классе Ψ.

Если для рассматриваемой выборки построены операционные характеристики двух методов, и оказалось, что при выбранном значении K операционная характеристика одного метода располагается выше операционной характеристики другого, то первый метод окажется более эффективным, поскольку при одинаковых затратах он обеспечивает большую надежность. Аналогично, если

при выбранном значении P операционная характеристика одного метода лежит левее операционной характеристики другого, то первый метод требует меньших затрат для обеспечения одинаковой надежности.

Отметим, что операционная характеристика метода в общем случае представляет собой некоторое множество точек на плоскости (K, P), причем одному значению K в этом множестве может соответствовать несколько значений P (полученных при различных значениях параметров метода). В связи с этим операционные характеристики разных методов могут иметь общее непустое пересечение, и в этой общей части при одних параметрах один метод лучше, а при других — второй. Как в этом случае сопоставлять методы в целом? Поскольку параметры методов находятся в нашем распоряжении, предлагается сравнивать алгоритмы по их лучшим достигнутым показателям операционной характеристики, т. е. сопоставлять верхние огибающие операционных характеристик методов. Формализуем данный подход.

Пусть множество $C = \{K(\pi), P(\pi)\}$ является операционной характеристикой алгоритма оптимизации, полученной для набора параметров $\pi \in \Pi$. Построим сечения $C(K)$ множества C для различных фиксированных значений K:

$$C(K) = \{P\colon (K, P) \in C\}.$$

Для всех непустых $C(K)$ выберем максимальную надежность

$$P^*(K) = \max\{P\colon P \in C(K)\}. \tag{2.28}$$

Кривую (2.28) будем называть *наилучшей операционной характеристикой* метода оптимизации. Теперь сравнение эффективности методов можно свести к сопоставлению их наилучших операционных характеристик.

Из условия $P(\pi) = P^*(K)$ можно определить наилучший выбор параметров метода при заданных затратах K и построить соответствующий кривой (2.28) набор $\pi^*(K)$ оптимальных параметров алгоритма оптимизации на исследуемом классе тестовых функций. Пример построения операционных характеристик конкретных алгоритмов приведен в п. 2.4.1.

2.3. Характеристическая схема представимости алгоритмов глобального поиска

Ранее мы уже отмечали, что важнейшим свойством численного метода является его сходимость к искомому решению, в нашем случае — к глобальному минимуму целевой функции задачи (2.15). При этом характер сходимости во многом определяет эффективность метода оптимизации. Настоящий раздел посвящен рассмотрению с единых теоретических позиций вопросов сходимости для широкого класса численных методов поиска глобального экстремума функций одной переменной, называемого *классом характеристических алгоритмов* и включающего многие известные алгоритмы, созданные в рамках различных подходов к конструированию методов оптимизации. Поскольку многие алгоритмы поиска экстремума многомерных функций основаны на редукции многомерной задачи к одной или семейству одномерных подзадач (см. главу 5), то данная теория может быть применима и к анализу ряда методов многомерной оптимизации.

Рассмотрим одномерную задачу (2.15) для области поиска $Q = [a, b]$ и класс методов оптимизации (1.9), в котором $H_k(\Phi, \omega_k) = 1$ для любого $k \geqslant 1$, т. е. условие остановки отсутствует. В этом случае метод порождает бесконечную последовательность испытаний $\{x^k\} = x^1, x^2, \ldots, x^k, \ldots$, изучение свойств которой и будет предметом нашего интереса.

Определение 2.2. Алгоритм решения задачи (2.15) называется *характеристическим*, если, начиная с некоторого шага поиска $k_0 \geqslant 1$, выбор координаты x^{k+1} очередного испытания ($k \geqslant k_0$) заключается в выполнении следующих действий.

1. Задать набор

$$\Lambda_k = \{x_0, x_1, \ldots, x_\tau\} \tag{2.29}$$

конечного числа $\tau + 1 = \tau(k) + 1$ точек области $Q = [a, b]$, полагая, что $a \in \Lambda_k$, $b \in \Lambda_k$, все координаты предшествующих испытаний $x^i \in \Lambda_k$, $1 \leqslant i \leqslant k$, и множество Λ_k упорядочено (нижним индексом) по возрастанию координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_{\tau-1} < x_\tau = b. \tag{2.30}$$

2. Каждому интервалу (x_{i-1}, x_i), $1 \leqslant i \leqslant \tau$, поставить в соответствие число $R(i)$, называемое характеристикой этого интервала.

3. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика $R(t)$, т. е.

$$R(t) = \max\{R(i)\colon 1 \leqslant i \leqslant \tau\}. \tag{2.31}$$

4. Провести очередное испытание в точке

$$x^{k+1} = d(t) \in (x_{t-1}, x_t). \tag{2.32}$$

В соответствии с определением «характеристичность» алгоритма определяет структуру его решающего правила G_{k+1} через последовательность операций, представленных пунктами 1–4. Этим операциям можно дать следующую содержательную интерпретацию.

Для проведения нового испытания отрезок $[a, b]$ точками множества Λ_k разбивается на τ интервалов (x_{i-1}, x_i), $1 \leqslant i \leqslant \tau$. Далее численно оценивается «перспективность» каждого интервала с помощью его характеристики и выбирается интервал, у которого характеристика наилучшая. Точка очередного испытания размещается внутри этого интервала в соответствии с правилом $d(\bullet)$.

Заметим, что множество (2.29) наряду с координатами испытаний может содержать точки, в которых испытания не проводились (например, в ряде информационно-статистических алгоритмов из [42, 67] такими точками являются концы отрезка). При этом *верхний индекс* координаты испытания соответствует *порядку* проведения испытаний в процессе поиска, а *нижний индекс* определяет *расположение* точки в упорядоченном наборе (2.30). Так, координата i-го испытания x^i в множестве Λ_k получит нижний индекс j, т. е. $x^i = x_j$, причем от шага к шагу номер j может меняться ($j = j(k)$).

Понятие характеристичности метода оптимизации впервые было введено В. А. Гришагиным [21] и позднее обобщено и распространено на другие классы задач и типы алгоритмов [17, 53, 61, 63].

В качестве иллюстрации приведем примеры известных алгоритмов глобальной оптимизации. В трех первых алгоритмах два начальных испытания проводятся в точках $x^1 = a$ и $x^2 = b$, характеристическое правило вступает в действие, начиная с k=2, при

этом множество Λ_k $(k \geqslant 2)$ состоит только из точек испытаний, т. е. $\Lambda_k = \{x^1, x^2, \ldots, x^k)$ и, следовательно, $\tau = k - 1$. Будем также использовать обозначение $z_j = \varphi(x_j)$ для значений целевой функции в точках $x_j \in \Lambda_k$.

Метод последовательного сканирования (перебор)

Для этого метода характеристикой интервала является его длина, т. е.

$$R(i) = x_i - x_{i-1}, \tag{2.33}$$

а точка очередного испытания выбирается в середине самого длинного интервала:

$$x^{k+1} = 0.5(x_{t-1} + x_t).$$

Метод Кушнера [57]

Характеристика метода имеет вид

$$R(i) = -\frac{4(\varphi_k^* - \delta_k - z_i)(\varphi_k^* - \delta_k - z_{i-1})}{x_i - x_{i-1}},$$

а очередное испытание проводится в точке

$$x^{k+1} = x_{t-1} + \frac{(x_t - x_{t-1})(\varphi_k^* - \delta_k - z_t)}{2(\varphi_k^* - \delta_k) - z_t - z_{t-1}},$$

где $\delta_k > 0$ — параметр метода, в общем случае зависящий от номера шага поиска k, а φ_k^* — наименьшее вычисленное значение функции (1.17).

Метод ломаных

В данном методе, который был построен С. А. Пиявским [36] для оптимизации липшицевых функций $\varphi \in \mathrm{Lip}[a, b]$, характеристика

$$R(i) = 0{,}5m(x_i - x_{i-1}) - (z_i + z_{i-1})/2, \tag{2.34}$$

а точка очередного испытания выбирается согласно выражению

$$x^{k+1} = 0.5(x_t + x_{t-1}) - (z_t - z_{t-1})/(2m), \tag{2.35}$$

где $m > 0$ — параметр метода.

Базовый информационно-статистический алгоритм глобального поиска (АГП)

Обсуждаемый метод предложен Р. Г. Стронгиным [42] как байесовский одношагово-оптимальный алгоритм и использует характеристику

$$R(i) = m(x_i - x_{i-1}) + \frac{(z_i - z_{i-1})^2}{m(x_i - x_{i-1})} - 2(z_i + z_{i-1}), \tag{2.36}$$

а точку нового испытания формирует согласно (2.35).

Величина $m > 0$ вычисляется в соответствии с выражением

$$m = \begin{cases} rM, & M > 0, \\ 1, & M = 0, \end{cases} \tag{2.37}$$

где

$$M = \max_{1 \leqslant i \leqslant \tau} \frac{|z_i - z_{i-1}|}{x_i - x_{i-1}}, \tag{2.38}$$

а $r > 1$ — параметр метода.

Информационный алгоритм глобального поиска с испытаниями только во внутренних точках (АГПВТ)

Данный алгоритм применяется для оптимизации функций на отрезке без вычисления значений функции на концах отрезка. Это важно для случая, когда границы отрезка не могут быть заданы явно (например, определяются как результат некоторой алгоритмической процедуры — см. п. 4.2.1, где границы области оптимизации являются границами областей неположительности непрерывной функции).

Согласно алгоритму, одно или несколько первых испытаний проводятся во внутренних точках интервала (a, b), а затем применяется характеристическая схема. Множество Λ_k включает только точки испытаний и концы интервала, откуда $\tau = k + 1$. Как и ранее, $z_j = \varphi(x_j)$, но лишь для $1 \leqslant j \leqslant k$.

Характеристика метода вычисляется согласно выражению

$$R(i) = \begin{cases} m(x_i - x_{i-1}) + \dfrac{r(z_i - z_{i-1})^2}{M(x_i - x_{i-1})} - 2(z_i + z_{i-1}), & 1 \leqslant i \leqslant k, \\ m(x_i - x_{i-1}) - 4z_1, & i = 1, \\ m(x_i - x_{i-1}) - 4z_k, & i = \tau, \end{cases}$$

а точка нового испытания как

$$x^{k+1} = 0{,}5(x_t + x_{t-1}) - \begin{cases} 0, & t = 1 \text{ или } t = k+1, \\ (z_t - z_{t-1})/(2m), & 2 \leqslant t \leqslant k, \end{cases}$$

где величина m — из (2.37).

После конкретных примеров представим интересный теоретический результат о связи принципа одношаговой оптимальности алгоритмов оптимизации со свойством характеристичности.

Теорема 2.1. *Одношагово-оптимальный (минимаксный или байесовский) алгоритм является характеристическим.*

Доказательство. В соответствии с принципом одношаговой оптимальности [42] вводится вещественная функция $W_k(x)$, характеризующая эффективность проведения испытания в точке x на шаге k, причем предполагается, что эффективность тем больше, чем меньше значение критерия $W_k(x)$. Если алгоритм выбирает точку очередного испытания x^{k+1}, $k \geqslant k_0 \geqslant 1$, из условия

$$W_k(x^{k+1}) = \min_{x \in [a,b]} W_k(x), \tag{2.39}$$

то такой алгоритм называется одношагово-оптимальным.

Перепишем условие одношаговой оптимальности (2.39) в виде

$$W_k(x^{k+1}) = \min_{x \in [a,b]} W_k(x) = \min_{1 \leqslant i \leqslant \tau} \min_{x \in [x_{i-1}, x_i]} W_k(x).$$

Отсюда следует, что в качестве характеристики интервала (x_{i-1}, x_i) можно взять величину

$$R(i) = -\min_{x \in [x_{i-1}, x_i]} W_k(x),$$

а $d(t) = \arg \min\limits_{x \in [x_{i-1}, x_i]} W_k(x)$. *Теорема доказана.*

Общие теоретические результаты о сходимости характеристических алгоритмов представлены в работах [18, 21]. Среди этих результатов следует отметить условия двусторонней сходимости к внутренним предельным точкам области поиска, позволяющие обосновать введение условия остановки вида (2.21). Кроме того, были исследованы различные типы сходимости последовательностей испытаний характеристических методов, в том числе получены достаточные условия сходимости к глобальному минимуму решаемой задачи.

Характеристическая структура (2.29)–(2.32) обладает существенным потенциалом параллелизма, что демонстрирует следующий раздел, в котором вводится и исследуется класс параллельных характеристических алгоритмов, представляющий собой обобщение последовательной характеристической схемы на параллельный случай.

2.4. Ускорение глобального поиска на основе использования дополнительной информации

Информационно-статистические алгоритмы глобального поиска, рассмотренные в настоящей главе, предназначены для решения многоэкстремальных задач оптимизации с липшицевыми функциями. Построение оптимизирующих последовательностей испытаний осуществляется последовательно и с использованием всей поисковой информации, получаемой в процессе вычислений

$$y^{k+1} = G_{k+1}(\Phi, \omega_k).$$

При этом для решения оптимизационных задач не требуется задания априори значений констант Липшица для функционалов модели — оценка этих значений осуществляется в процессе глобального поиска опять же на основе имеющейся поисковой информации (см., например, (2.19)).

Тем не менее остается еще один важный момент, на который должно быть обращено особое внимание при построении эффективных решающих правил алгоритмов глобального поиска. Этот

момент состоит в том, что в большинстве случаев *поведение функций в задаче многоэкстремальной оптимизации является неоднородным* в разных подобластях области поиска. В некоторых подобластях значения функций модели могут изменяться достаточно быстро (что будет соответствовать большим значениям константы Липшица в таких подобластях), в других подобластях — например, в окрестности точек экстремумов — значения могут изменяться более плавно. Как результат, построение интегральных (единых) оценок характеристик оптимизируемых функций для всей области поиска может не соответствовать полностью параметрам функций в разных подобластях области поиска.

Среди возможных подходов, учитывающих неоднородность поведения оптимизируемых функций в разных подобластях области поиска, можно выделить три направления, в рамках которых получены наиболее значимые результаты, а именно:

- прогнозирование экстремума на основе накопленной поисковой информации, основанное на предположении слабой многоэкстремальности задачи;
- построение отдельных оценок констант Липшица оптимизируемых функций для разных подобластей области поиска;
- использование дополнительной информации о поведении оптимизируемых функций – в большинстве случаев, в качестве такой дополнительной информации рассматриваются значения производных в точках проводимых испытаний глобального поиска.

Изложение материала в этом разделе будем проводить для одномерных задач оптимизации без ограничений; для рассмотрения общего многомерного случая предназначены главы 4 и 5 настоящей книги.

2.4.1. Адаптивный прогноз оптимума

Априорная информация, имеющаяся при решении прикладных многоэкстремальных задач, помимо факта ограниченности разделенных разностей функций задачи, может содержать дополнительные характеристики экстремума, использование которых ускоряет процесс поиска. К числу таких характеристик можно отнести

некоторый априорный прогноз расположения оптимальной точки (основанный, например, на наличии прототипа оптимизируемого объекта). Если же такой прогноз отсутствует, можно построить адаптивные правила прогнозирования, основанные на предположении о большей вероятности нахождения минимума в окрестности итераций, которым соответствуют малые значения оптимизируемой функции.

Вначале рассмотрим случай, когда возможно дать некоторый априорный прогноз расположения точки оптимума x^* в области поиска $[a, b]$. Формально этот прогноз можно описать с помощью функции распределения $F(x)$, $x \in [a, b]$.

Алгоритм, учитывающий априорное распределение глобального оптимума, может быть получен путем обобщения базового информационно-статистического алгоритма (АГП) из предыдущего раздела, состоящего в том, что характеристики $R(i)$ интервалов (x_{i-1}, x_i), $1 \leqslant i \leqslant k$, используемые при выборе точки очередной итерации, принимают вид

$$R_F(i) = \frac{P_i}{x_i - x_{i-1}} R(i), \quad P_i = F(x_i) - F(x_{i-1}). \tag{2.40}$$

При этом чем больше априорная мера нахождения оптимума в интервале (x_{i-1}, x_i), тем больше характеристика (2.40) этого интервала. Напомним, что очередная итерация осуществляется в интервале с максимальной характеристикой. Например, если

$$p_i = \frac{x_i - x_{i-1}}{b - a}, \quad 1 \leqslant i \leqslant k,$$

характеристика (2.40) примет вид

$$R_F(i) = \frac{R(i)}{b - a}, \quad 1 \leqslant i \leqslant k.$$

Отметим, что выбор точки $x^1 \in (a, b)$ первой итерации можно связать с функцией распределения $F(x)$, например осуществлять первую итерацию в точке максимального правдоподобия.

Утверждения о сходимости алгоритма остаются справедливыми и для предложенной модификации [31], учитывающей функцию распределения F, если требования сходимости АГП дополнить

условиями строгой положительности и ограниченности плотности $f(x)$, порождающей функцию $F(x)$, т. е.

$$0 < c_1 \leqslant f(x) \leqslant c_1 < \infty, \quad x \in [a, b].$$

Рассмотрим теперь случай, когда априорные предположения о глобальном оптимуме отсутствуют. В этом случае можно сделать *адаптивный прогноз*, основанный на предположении, что местонахождение глобального оптимума в области $[a, b]$ более вероятно в окрестностях тех точек из ряда x_i, $1 \leqslant i \leqslant k$, которым соответствуют малые значения нормированных разностей

$$\mu_i = \frac{z_i - z_k^*}{\mu}, \quad 0 \leqslant i \leqslant k,$$

где

$$\mu = \max\left\{\frac{|z_i - z_{i-1}|}{x_i - x_{i-1}}, \quad 1 \leqslant i \leqslant k\right\}.$$

Принятая нами гипотеза правдоподобна либо в случае, когда число осуществленных итераций достаточно велико (так как алгоритм при выполнении условий сходимости порождает последовательность испытаний, плотную лишь в точках глобального минимума), либо в случае, когда задача является унимодальной, т. е. когда поиск в окрестности лучшего текущего приближения позволяет достичь улучшения, и такие последовательные улучшения ведут к решению). Первый случай соответствует заключительной фазе поиска, и тогда использование адаптивного прогноза можно интерпретировать как способ ускорения уточнения решения. Второй случай фактически включает в себя предположения о слабой многоэкстремальности задачи.

При использовании адаптивного прогноза нахождения оптимума нет необходимости строго выводить оценки вероятностей $P_i = F(x_i) - F(x_{i-1})$, можно это сделать приближенно (подробнее об этом см. в [44]). При построении оценок будем использовать параметр α, $0 \leqslant \alpha < 1$, характеризующий степень достоверности предположения о большей вероятности расположения глобального оптимума в окрестностях итераций с меньшими значениями функции. Также потребуем, чтобы оценки удовлетворяли следующим

свойствам:

$$\begin{aligned} &P_i \to 0 \quad \text{при } \alpha \to 1 \quad \text{если} \quad \mu_{i-1},\, \mu_i > 0, \\ &P_i \to \frac{x_i - x_{i-1}}{b-a} \quad \text{при } \alpha \to 0. \end{aligned} \tag{2.41}$$

В первом случае мера, соответствующая текущему шагу, сосредотачивается на интервалах (x_{i-1}, x_i), для которых справедливо условие $\min\{\mu_{i-1}, \mu_i\} = 0$. Второй случай соответствует равномерному распределению.

Следовательно, при $\alpha \approx 1$ новые итерации будут осуществляться лишь в интервалах, содержащих лучшее текущее приближение, и поиск приобретет локальный характер. Случай $\alpha \approx 0$ соответствует глобальному характеру поиска, выбор итераций определяется исходным алгоритмом.

Конкретным примером оценки вероятности P_i, приближенно обладающей свойствами (2.41), является выражение

$$P_i = \frac{x_i - x_{i-1}}{\sqrt{\mu_{i-1}\mu_i} + 10^{-q}(b-a)}, \tag{2.42}$$

где целочисленный параметр q, $1 \leqslant q \leqslant 9$ характеризует «степень локальности» алгоритма. При $q = 1$ поиск имеет «глобальный», а при $q = 9$ «локальный» характер.

Алгоритм допускает управление параметром q, входящим в оценку (2.42), что позволяет учитывать различные предположения о характере экстремума. Например, одной из стратегий является чередование с заданной частотой значений $q = 1$ и $q = 9$, что соответствует смеси глобального и локального поиска, которую можно интерпретировать как сочетание процедуры локального уточнения текущего приближения и процедуры глобального обновления этого приближения. В течение первых K итераций рекомендуется использовать значение $q = 1$, при этом глобальный поиск обеспечит адекватность оценок (2.42) при последующем локальном уточнении.

Результаты экспериментов. В качестве примера рассмотрим решение серии задач, сгенерированных из заданного класса. Типичными примерами такого рода генераторов являются случайные механизмы, основанные на предложениях Шекеля и Хилла (см.

соответственно работы [68] и [55]). Первый из них (обозначим его F_{SH}) порождает многоэкстремальные функции в соответствии со схемой

$$\varphi(x) = -\sum_{j=1}^{10} \frac{1}{(K_j(x - A_j)^2 + C_j)}, \quad x \in [0, 10], \tag{2.43}$$

$1 \leqslant K_j \leqslant 3$, $0 \leqslant A_j, C_j \leqslant 10$, где параметры K_j, A_j, C_j, $1 \leqslant j \leqslant$ $\leqslant 10$, — независимые случайные величины, равномерно распределенные в указанных выше интервалах.

Второй механизм (обозначим его F_{HL}) определяется выражением

$$\varphi(x) = \sum_{j=1}^{14} (A_j \sin(2j\pi x) + B_j \cos(2j\pi x)), \quad x \in [0, 1], \tag{2.44}$$

где значения параметров A_j, B_j, $1 \leqslant j \leqslant 14$, независимо и равномерно распределены в интервале $[-1, 1]$.

В качестве примера на рис. 2.1 приведен график функции из класса F_{SH}, а на рис. 2.2 — график функции из класса F_{HL}.

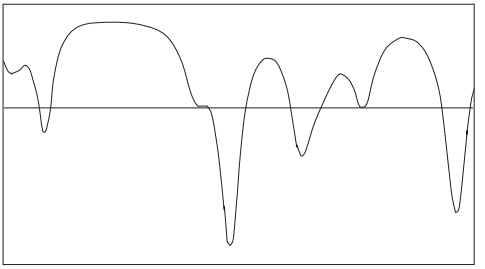

Рис. 2.1. Функция из класса F_{SH}

Примененная процедура оценки эффективности алгоритмов использует аппарат операционных характеристик, изложенный в п. 2.2.2. Задачи из выборок решались с помощью алгоритма глобального поиска (АГП) и смешанного алгоритма (СА). В ходе экспериментов задавались параметры алгоритма $r = 2{,}5$ и $\varepsilon = 10^{-4}(b - a)$. При этом в смешанном алгоритме локальное уточнение включалось после 10-й итерации, после чего чередование «локального» и «глобального» поиска (что соответ-

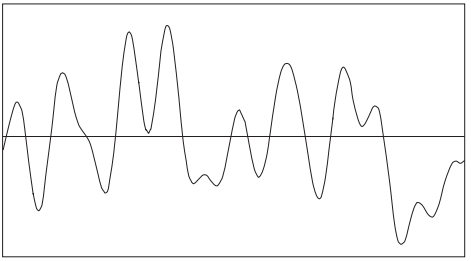

Рис. 2.2. Функция из класса F_{HL}

ствует чередованию значений $q = 9$ и $q = 1$ в формуле (2.42)) осуществлялось на каждой итерации.

Операционные характеристики для обоих методов, полученные на выборках F_{SH} и F_{HL}, представлены соответственно на рис. 2.3 и 2.4. При этом нижние (непрерывные) кривые характеризуют АГП, а верхние (разрывные) линии — СА. Указанное расположение кривых показывает, что смешанный алгоритм с адаптивным прогнозом оптимума обеспечивает значительно более быстрое получение оценок, лежащих в заданной окрестности решения, чем базовый алгоритм глобального поиска. При этом среднее число итераций для АГП и СА составило соответственно 28 и 21 для задач F_{SH}; 72 и 45 для задач F_{HL}.

2.4.2. Проблема оценки константы Липшица и локально-адаптивные схемы глобального поиска

Как отмечалось ранее, другой основной способ учета разнородного поведения оптимизируемых функций в разных областях поиска состоит в построении отдельных оценок констант Липшица оптимизируемых функций для разных подобластей области поиска. Целесообразность такого направления работ достаточно очевидна — так, в окрестности точек экстремумов оптимизируемой функции значения производных должны быть близки к нулю, определяя тем самым невысокое значение константы Липшица в таких подобластях.

Исследования в данном направлении были инициированы в работе [38]; предложенный в работе подход рассматривается

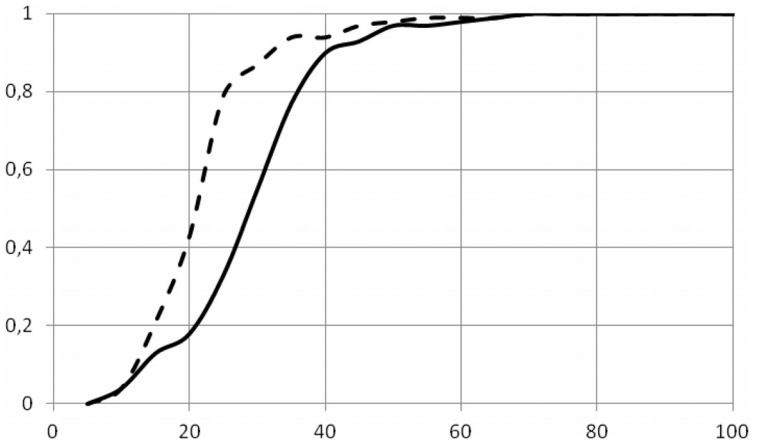

Рис. 2.3. Операционные характеристики алгоритмов на классе F_{SH}

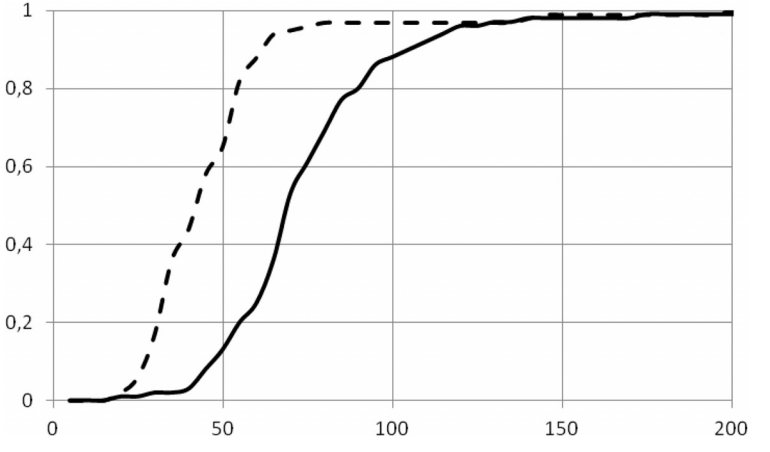

Рис. 2.4. Операционные характеристики алгоритмов на классе F_{HL}

в п. 2.4.2.1. В п. 2.4.2.2 предлагаются новые способы построении отдельных оценок констант Липшица оптимизируемых функций для разных подобластей области поиска.

2.4.2.1. Локальная настройка оценок константы Липшица

Суть предложенного в [38] подхода состоит в следующем: для построения решающих правил алгоритмов глобального поиска предлагается использовать не единую для всей области поиска оценку константы Липшица, а строить такие оценки для каждого поискового интервала в отдельности. Поскольку получение таких оценок может происходить с достаточно большими погрешностями, то при нахождении локальных (интервальных) оценок должна учитываться и общая (глобальная) оценка константы Липшица.

Приведем в соответствии с [39] точное алгоритмическое описание предлагаемого подхода.

Вычислительная схема оценки локальных констант Липшица. Предположим, что алгоритм глобального поиска, в котором используются оценки константы Липшица, выполнил $k \geqslant 1$ итераций поиска, и получены точки испытаний

$$a = x_0 < x_1 < \ldots < x_i < \ldots < x_k = b$$

(напомним, что, как и ранее, нижний индекс соответствует упорядоченному представлению точек испытаний).

Оценка константы Липшица M_i интервала (x_{i-1}, x_i), $1 < i < k$, вычисляется в соответствии со следующими выражениями:

$$m_i = rM_i, \quad M_i = \max\{\mu_i', \mu_i'', \mu_0\},$$

где

$$\mu_i' = \max\{|z_j - z_{j-1}|/(x_j - x_{j-1})\colon j = i-1, i, i+1\}, \tag{2.45}$$

$$\mu_i'' = M(x_i - x_{i-1})/d_{\max}, \tag{2.46}$$

$$M = \max_{1 \leqslant i \leqslant k} \frac{|z_i - z_{i-1}|}{x_i - x_{i-1}}, \tag{2.47}$$

$$d_{\max} = \max_{1 \leqslant i \leqslant k} (x_i - x_{i-1}). \tag{2.48}$$

Когда $i = 1$ или $i = k$, для вычисления μ_i' из (2.45) используются только $j = i, i+1$ или $j = i-1, i$ соответственно.

Поясним приведенные соотношения. Величина M является, как и раньше, оценкой глобальной константы Липшица L на всем интервале поиска $[a, b]$. Величины M_i, $1 \leqslant i \leqslant k$ являются оцен-

ками локальных констант Липшица L_i на интервалах (x_{i-1}, x_i), $1 \leqslant i \leqslant k$. Каждая такая оценка состоит из трех частей: μ_i', которая отвечает за локальные свойства функции $\varphi(x)$ на интервале $[x_{i-1}, x_i]$; μ_i'', которая следит за глобальными свойствами функции $\varphi(x)$ на всем интервале поиска $[a, b]$; параметра μ_0. Когда интервал $[x_{i-1}, x_i]$ велик, роль глобальной информации повышается, так как в этом случае локальная информация может быть ненадежной. Когда интервал $[x_{i-1}, x_i]$ мал, роль глобальной информации понижается, так как в этом случае большее значение имеет локальная информация, а влияние глобальной информации ослабляется. Наличие параметра μ_0 отражает предположение о том, что функция $\varphi(x)$ не является константой на интервале $[a, b]$. Параметр μ_0 также контролирует чувствительность алгоритма. В случае, когда $L \ll \mu_0$, алгоритм практически теряет чувствительность к изменениям локальной информации.

Важно подчеркнуть, что представленная схема локальной настройки сохраняет свойства сходимости алгоритма глобального поиска, к которому она применяется. Так, для алгоритма глобального поиска Стронгина [42] в сочетании со схемой локальной настройки справедлива следующая теорема [39].

Теорема 2.2. *Если существует такая бесконечная последовательность $\{s\}$ номеров итераций $\{s\} \subset \{k\}$, что для интервала*

$$[x_{i-1}, x_i], \quad j = j(l), \quad l \in \{s\},$$

содержащего точку x^* на итерации l алгоритма глобального поиска с локальной настройкой, выполняется правило

$$m_j \geqslant \eta_j + \left(\eta_j^2 - \sigma_j^2\right)^{1/2},$$

то x^* есть предельная точка последовательности $\{x^k\}$.

Использованные в формулировке теоремы обозначения соответствуют следующим выражениям:

$$\eta_j = \max\left\{\frac{z_{j-1} - \varphi(x^*)}{x^* - x_{j-1}}, \frac{z_j - \varphi(x^*)}{x_j - x^*}\right\}, \quad \sigma_j = \frac{z_j - z_{j-1}}{x_j - x_{j-1}}.$$

Для демонстрации эффективности рассмотренного подхода приведем результаты вычислительного эксперимента [39], в котором

в качестве тестового множества оптимизационных задач использован набор функций, сформированный в [54] из большого числа источников. Для получения представления о виде функций из данного набора, на рис. 2.5 показаны графики некоторых из этих функций (номера функций соответствуют порядку расположения функций в таблице 2.1).

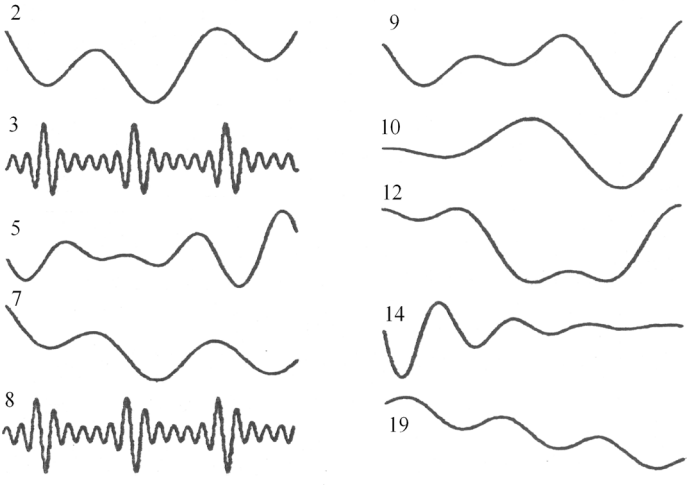

Рис. 2.5. Графики отдельных функций из тестового набора оптимизационных задач

При проведении эксперимента применялись несколько методов глобальной оптимизации — это алгоритм Галперина (АГ) [52], алгоритм Пиявского (АП) [36] и алгоритм глобального поиска Стронгина (АГП) [42]. Сравнение выполнялось с двумя методами, использующими локальную настройку — это метод ломаных (МЛ-ЛН), построенный на основе метода Пиявского, и алгоритм глобального поиска Стронгина с локальной настройкой (АГП-ЛН) [10].

Результаты экспериментов сведены в таблице 2.1. По каждому методу в таблице приведено количество испытаний, выполняемое

Таблица 2.1. Результаты вычислительных экспериментов для одномерных методов глобальной оптимизации, использующих схему локальной настройки (МЛ-ЛН и АГП-ЛН)

Задача	Методы глобального поиска				
	АГ	**АП**	**АГП**	**МЛ-ЛН**	**АГП-ЛН**
1	377	149	127	37	35
2	308	155	135	36	36
3	581	195	224	145	136
4	923	413	379	45	41
5	326	151	126	46	45
6	263	129	112	84	54
7	383	153	115	41	39
8	530	185	188	126	132
9	314	119	125	44	42
10	416	203	157	43	40
11	779	373	405	74	72
12	746	327	271	71	66
13	1829	993	472	73	45
14	290	145	108	43	50
15	1613	629	471	62	63
16	992	497	557	79	53
17	1412	549	470	100	101
18	620	303	243	44	41
19	302	131	117	39	34
20	1412	493	81	70	42
В среднем	720,8	314,6	244,15	65,1	58,35

алгоритмом до выполнения условия остановки. В последней строке таблицы содержится среднее число испытаний по всему набору тестовых задач для каждого метода в отдельности.

Точность поиска в расчетах принималась равной $\varepsilon = 10^{-4}\,(b-a)$. Для алгоритмов Галперина и Пиявского использовалась точная оценка константы Липшица. Параметры остальных алгоритмов были равны 2 (для АГП и АГП-ЛН) и 1,1 для МЛ-ЛН. Для методов МЛ-ЛН и АГП-ЛН параметр μ_0 устанавливался равным $\mu_0 = 10^{-6}$.

Следует подчеркнуть, что схема локальной настройки может быть применена для построения самых разных одномерных и многомерных алгоритмов решения задач липшицевой глобальной оптимизации. Она может быть также обобщена, например, на случай задачи с целевой функцией, удовлетворяющей условию Гёльдера или на случай решения задачи глобальной оптимизации с многоэкстремальными липшицевыми ограничениями [39].

2.4.2.2. Локальная настройка с аддитивной сверткой оценок константы Липшица

Внимательно проанализировав схему локальной настройки, можно сделать два важных замечания.

Замечание 1. Если проанализировать характер зависимости (2.46) влияния оценки глобальной константы Липшица для интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k$, то можно сделать вывод, что при построении данного соотношения явно или неявно делалось предположение о квадратичном поведении оптимизируемой функции в окрестности глобального минимума. На самом деле производная для квадратичной функции имеет вид линейной зависимости, и для определения значения производной на одном интервале при известном ее значении на другом интервале необходимо разделить известное значение производной на длину интервала, на котором задано значение, и умножить на длину искомого интервала. Или аналитически — пусть минимизируемая функция имеет вид

$$\varphi(x) = ax^2,$$

тогда производная этой функции

$$\varphi'(x) = 2ax.$$

Оценка константы Липшица для интервалов $(x_1{=}0, x_2)$ и $(x'{=}0, x'')$, $x'' < x_2$ имеет вид

$$M_1 = \frac{|z_2 - z_1|}{x_2 - x_1} = ax_2, \quad M' = \frac{|z'' - z'|}{x'' - x'} = ax''$$

соответственно. Но тогда оценку M' можно получить через значение оценки M_1, т. е.

$$M' = [M_1/(x_2 - x_1)](x'' - x') = (M_1/x_2)x'',$$

что и соответствует правилу (2.46) схемы локальной настройки. Отсюда следует два возможных вывода:

— схема локальной настройки наиболее эффективно будет действовать для многоэкстремальных оптимизационных задач, в которых поведение минимизируемых функций в окрестности глобального минимума является квадратичным;

— схема локальной настройки может быть построена и для многоэкстремальных оптимизационных задач, в которых поведение минимизируемых функций в окрестности глобального минимума не является квадратичным. Для этого следует только заменить соотношение (2.46) — так, для кубического поведения функций в окрестности глобального минимума соотношение должно иметь вид

$$\mu_i'' = M(x_i - x_{i-1})^2 / (d_{\max})^2.$$

Замечание 2. Локальная (2.45) и глобальная (2.46) составляющие интервальной оценки константы Липшица могут рассматриваться как два разных противоречивых критерия при оценке некоторого компромиссного значения константы Липшица. Предпочтения одного критерия (например, локальной составляющей) приводит к более быстрому завершению работы алгоритма глобального поиска, однако здесь возможна потеря решения из-за оценки константы Липшица с недостатком. Предпочтение другого критерия (глобальной составляющей) может привести к более продолжительной работе алгоритма глобального поиска из-за оценки константы Липшица с избытком. Как результат, для получения некоторого компромиссного (промежуточного) значения константы Липшица необходима та или иная свертка данных критериев. Так, в схеме локальной настройки в п. 2.4.2.1 данная свертка близка к широко известной минимаксной свертке критериев (см., например, [15]). Возможны и иные способы сверток.

Другой возможной схемой локальной настройки интервальных оценок константы Липшица может являться использование аддитивной свертки [15]:

$$m_i = rM_i, \quad M_i = \max\left\{\frac{1}{r}\mu_i' + \frac{r-1}{r}\mu_i'',\ \mu_0\right\},$$

где величины μ_i', μ_i'', μ_0 определяются соотношениями (2.45)–(2.48).

Для оценки эффективности предлагаемого подхода были проведены вычислительные эксперименты, в которых в качестве контрольного множества оптимизационных задач использовался набор функций, рассматриваемый в предыдущем пункте.

Результаты экспериментов сведены в таблицу 2.2. Для сравнения в таблице приведены данные из таблицы 2.1 для алгоритма Галперина (АГ), алгоритма Пиявского (АП), алгоритма глобального поиска Стронгина (АГП) и алгоритма глобального поиска Стронгина с локальной настройкой (АГМ-ЛН). Алгоритм глобального поиска Стронгина с локальной настройкой в соответствии с новой предложенной аддитивной схемой свертки обозначен в таблице как АГП-ЛНА.

Точность поиска в расчетах принималась равной $\varepsilon = 10^{-4}(b-a)$. Для алгоритмов Галперина и Пиявского использовалась точная оценка константы Липшица. Параметры остальных алгоритмов были равны 2 (для АГП, АГП-ЛН и АГП-ЛНА). Для методов АГП-ЛН и АГП-ЛНА параметр μ_0 устанавливался равным $\mu_0 = 10^{-6}$. В процессе решения задач оптимум был найден всеми методами с заданной точностью.

По каждому методу в таблице приведено количество испытаний, выполняемое алгоритмом до выполнения условия остановки. В последней строке таблицы содержится среднее число испытаний по всему набору тестовых задач для каждого метода в отдельности.

Анализируя представленные в таблице 2.2 результаты, можно отметить, что эффективность предложенной аддитивной свертки значительного превышает эффективность исходной схемы минимаксной свертки (по среднему количеству итераций не менее чем на 50%).

Таблица 2.2. Результаты вычислительных экспериментов для одномерных методов глобальной оптимизации, использующих схему локальной настройки

Задача	Методы глобального поиска				
	АГ	**АП**	**АГП**	**АГП-ЛН**	**АГП-ЛНА**
1	377	149	127	35	29
2	308	155	135	36	27
3	581	195	224	136	42
4	923	413	379	41	37
5	326	151	126	45	31
6	263	129	95	54	36
7	383	153	115	39	27
8	530	185	188	132	42
9	314	119	125	42	33
10	416	203	157	40	29
11	779	373	405	72	56
12	746	327	271	69	51
13	1829	993	472	45	40
14	290	145	100	43	27
15	1613	629	471	63	40
16	992	497	557	53	45
17	1412	549	470	101	81
18	620	303	243	41	34
19	302	131	117	34	26
20	1412	493	86	42	29
В среднем	720,8	314,6	243,15	58,15	38,05

2.4.3. Алгоритмы глобального поиска с использованием значений производных

Возможный способ учета разнородного поведения оптимизируемых функций в разных областях поиска состоит в вычислении и использовании значения производных имеющихся функций задачи в точках испытаний глобального поиска. В этом случае к классу

рассматриваемых задач многоэкстремальной оптимизации выдвигаются дополнительные требования — функции задач должны быть дифференцируемыми, и их производные также должны удовлетворять условию Липшица. Данное направление исследований активно развивается в теории многоэкстремальной оптимизации (см., например, работы [8, 9, 39, 59, 62]).

Итак, рассматривается одномерная безусловная задача многоэкстремальной оптимизации

$$\varphi(x^*) = \min\{\varphi(x)\colon x \in [a, b]\},$$

для которой минимизируемая функция $\varphi(x)$ и ее первая производная удовлетворяют условию Липшица с константами L и L_1 соответственно.

Приведем кратко идею получения решающих правил методов многоэкстремальной оптимизации с использованием производных [9].

Липшицевость первой производной с учетом разложения функции $\varphi(x)$ в ряд Тейлора до второго порядка малости приводит к справедливости соотношения

$$\varphi(x) \geqslant \max\big[\varphi(x_1) + \varphi' x_1)(x - x_1) - 0{,}5L_1(x - x_1)^2, \\ \varphi(x_2) - \varphi'(x_2)(x_1 - x) - 0{,}5L_1(x_1 - x)^2\big]. \quad (2.49)$$

На основе полученного неравенства может быть получена уточненная с использованием значений производных миноранта минимизируемой функции $\varphi(x)$. Для сравнения на рис. 2.6 показаны миноранты функции $\varphi(x)$ с учетом (пунктирная линия) и без учета (непрерывная линия) значения производных. Можно заметить, что использование производных позволяет получить более точные оценки снизу для минимизируемой функции.

Соотношение (2.49) послужило основой для разработки базового адаптивного глобального метода с использованием производных (БАГМП [8]).

Вместе с этим можно отметить, что производная новой миноранты функции $\varphi(x)$ является разрывной и не удовлетворяет условию Липшица. Тем самым можно усовершенствовать и уточненную

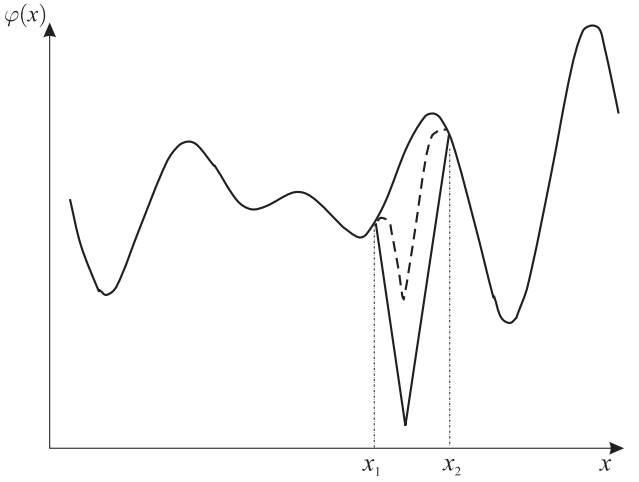

Рис. 2.6. Вид миноранты, построенной с учетом значений производных

миноранту. В работе [9] предлагается формировать такую усовершенствованную миноранту в виде составной функции:

$$\varphi^-(x)=\begin{cases} \varphi_1^-(x) = \varphi(x_1) + \varphi'(x_1)(x-x_1) - 0{,}5m(x-x_1)^2, \\ \qquad x \in [x_1, x'], \\ \varphi_2^-(x) = A(x-x') + 0{,}5m(x-x')^2 + B, \\ \qquad x \in [x', x''], \\ \varphi_3^-(x) = \varphi(x_2) + \varphi'(x_2)(x_2-x) - 0{,}5m(x_2-x)^2, \\ \qquad x \in [x'', x_2], \end{cases} \tag{2.50}$$

где в качестве величины m может быть использовано любое значение, большее L_1, т. е. $m > L_1$. Определение интервала $[x', x'']$ и величин A и B осуществляется таким образом, чтобы значения миноранты и ее производной в точках x' и x'' совпадали. Данные требования выполняются при

$$B = \varphi_1^-(x'), \quad A = \varphi'(x_1) - m(x' - x_1), \tag{2.51}$$

$$x' = \frac{[(z_1 - z_1'x_1) - (z_2 - z_2'x_2)] + m(x_2^2 - x_1^2)/2 - mdx^2}{m(x_2 - x_1) + (z_2' - z_1')}, \qquad (2.52)$$

$$x'' = \frac{[(z_1 - z_1'x_1) - (z_2 - z_2'x_2)] + m(x_2^2 - x_1^2)/2 + mdx^2}{m(x_2 - x_1) + (z_2' - z_1')} \qquad (2.53)$$

где

$$dx = (x_2 - x_1)/2 - (z_2' - z_1')/2m.$$

Вид новой полученной миноранты $\varphi^-(x)$ показан на рис. 2.7; для сравнения там же штриховой линией показан график миноранты, выводимой из соотношений (2.49).

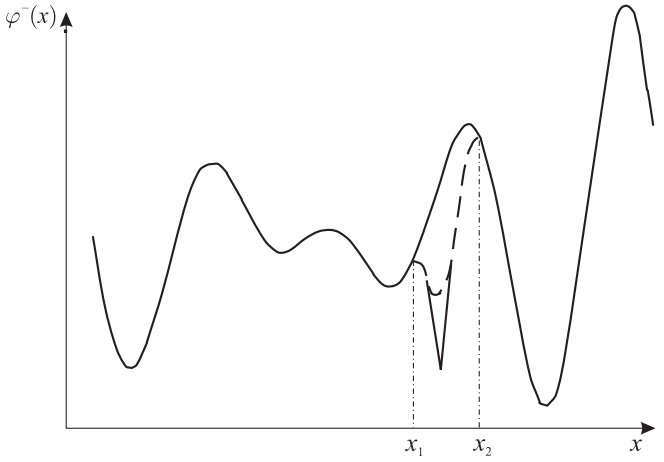

Рис. 2.7. Вид усовершенствованной миноранты, построенной с учетом значений производных

Усовершенствованная миноранта вида (2.50) была положена в основу построения *адаптивного глобального метода с использованием производных* (АГМП) [9]. Дадим краткое его описание (следует отметить, что метод принадлежит классу характеристических алгоритмов).

Первые два вычисления значения функции $\varphi(x)$ и ее производной $\varphi'(x)$ осуществляются в граничных точках интервала $[a, b]$, т. е. $x^0 = a$, $x^1 = b$ (верхним индексом, как и ранее, обозначен

номер испытания). Выбор точки x^{k+1}, $k \geqslant 1$, любого последующего испытания определяется следующими правилами (при описании алгоритма используются соотношения (2.51)–(2.53), в которых вместо индексов 1 и 2 используются индексы $i-1$ и i соответственно).

1. Перенумеровать точки $x^0, x^1, \ldots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_i < \ldots < x_k = b.$$

2. Оценить по результатам выполненных итераций поиска значение константы Липшица первой производной минимизируемой функции

$$M = \max_{1 \leqslant i \leqslant k} M_i, \tag{2.54}$$

$$M_i = \max \begin{cases} |z'_i - z'_{i-1}|/(x_i - x_{i-1}), \\ 2[-(z_i - z_{i-1}) + z'_{i-1}(x_i - x_{i-1})](x_i - x_{i-1})^2, \\ 2[(z_i - z_{i-1}) - z'_i(x_i - x_{i-1})](x_i - x_{i-1})^2 \end{cases}$$

и положить

$$m = \begin{cases} 1, & M = 0, \\ rM, & M > 0, \end{cases}$$

где $r > 1$ — заданный коэффициент (*параметр алгоритма*).

3. Для каждого интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k$, вычислить *характеристику* $R(i)$

$$R(i) = -\begin{cases} \varphi^-(\widehat{x}_i), & \widehat{x}_i \in [x'_i, x''_i], \\ \min[\varphi^-(x'_i), \varphi^-(x''_i)], & \widehat{x}_i \notin [x'_i, x''_i], \end{cases}$$

где

$$\widehat{x}_i = [-z'_{i-1} + m(x_i - x_{i-1}) + mx_i]/m,$$

а точки x' и x'' взяты из (2.52)–(2.53).

4. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i)\colon 1 \leqslant i \leqslant k\}. \tag{2.55}$$

5. Определить точку очередного испытания x^{k+1}согласно выражению

$$x^{k+1} = \begin{cases} \widehat{x}_t, & \widehat{x}_t \in [x'_t, x''_t], \quad \widehat{x}_t \text{ из } (2.11), \\ x'_t, & \varphi^-(x'_t) \leqslant \varphi^-(x''_t), \\ x''_t, & \varphi^-(x'_t) > \varphi^-(x''_t), \end{cases}$$

где индекс t соответствует интервалу с наименьшей характеристикой $R(t)$ из (2.55).

Условие остановки, прекращающее испытания, состоит в правиле

$$x_t - x_{t-1} \leqslant \varepsilon,$$

где t из (2.55) и $\varepsilon > 0$ есть заданная точность, и согласно которому выполнение итераций алгоритма прекращается, если длина интервала t с максимальной характеристикой $R(t)$ меньше, чем требуемая точность решения оптимизационной задачи.

В качестве оценки глобального минимума, соответствующему шагу k, принимается

$$z_k^* = \min_{0 \leqslant i \leqslant k} z^i, \quad x_k^* = \arg \min_{0 \leqslant i \leqslant k} z^i.$$

Для пояснения рассмотренного алгоритма можно отметить, что характеристика интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k$, есть оценка минимального возможного значения функции $\varphi(x)$ на этом интервале, взятая с обратным знаком (при данном приближении m константы Липшица L_1); точка нового испытания x^{k+1} соответствует положению в области $[a, b]$, в котором возможно наименьшее значение функции $\varphi(x)$. Оценка возможных значений $\varphi(x)$ на шаге k, $k \geqslant 1$, осуществляется по миноранте $\varphi^-(x)$ из (2.50) при использовании всей накопленной информации.

Величина M из (2.54) представляет собой приближенную оценку константы Липшица L_1 для первой производной минимизируемой функции по результатам проведенных испытаний.

Условия сходимости алгоритма сформулированы в [9] в виде следующей теоремы.

Теорема 2.3. *Пусть точка $\overline{x}$ есть предельная точка последовательности $\{x^k\}$, порождаемой алгоритмом при минимизации*

функции $\varphi(x)$, $x \in [a, b]$; $\overline{x} \neq a$, $\overline{x} \neq b$, *производная которой удовлетворяет условию Липшица с константой* L_1. *Тогда верно следующее*:

1) *если наряду с точкой* $\overline{x}$ *существует другая предельная точка* $\widehat{x}$ *последовательности* $\{x^k\}$, *то* $\varphi(\overline{x}) = \varphi(\widehat{x})$;

2) $z^k = \varphi(x^k) \geqslant \varphi(\overline{x})\ \forall\ k \geqslant 0$;

3) *если на некотором шаге* $k \geqslant 0$ *для величины* m *из* (2.54) *справедливо неравенство*

$$m > L_1,$$

то любая точка x^* *абсолютного минимума является предельной точкой последовательности* $\{x^k\}$, *сходимость к предельной точке* $\overline{x}$ *является двусторонней и, кроме того, любая предельная точка* $\overline{x}$ *этой последовательности является точкой абсолютного минимума функции* $\varphi(x)$.

Для оценки эффективности рассмотренного метода приведем результаты вычислительного эксперимента [9]. В качестве тестового множества оптимизационных задач использован тот же самый набор функций, что и в пп. 2.4.2.1, 2.4.2.2.

При проведении эксперимента осуществлялось сравнение алгоритма Галперина (АГ), алгоритма Пиявского (АП), алгоритма глобального поиска Стронгина (АГП) и рассмотренные в настоящем разделе методы — базовый адаптивный глобальный метод с производными (БАГМП), адаптивный глобальный метод с производными (АГМП).

Результаты экспериментов сведены в таблицу 2.3. По каждому методу в таблице приведено количество испытаний, выполняемое алгоритмом до выполнения условия остановки. В последней строке таблицы содержится среднее число испытаний по всему набору тестовых задач для каждого метода в отдельности.

Точность поиска в расчетах принималась равной $\varepsilon = 10^{-4}(b - a)$. Для алгоритмов Галперина и Пиявского использовалась точная оценка константы Липшица. Параметры остальных алгоритмов были равны 2 (для АГП) и 1,1 (для БАГМП и АГМП). При минимизации 5-й функции методом АГМП использовалось значение параметра $r = 2$.

Таблица 2.3. Результаты вычислительных экспериментов для одномерных методов глобальной оптимизации

Задача	Методы глобального поиска				
	АГ	**АП**	**АГП**	**БАГМП**	**АГМП**
1	377	149	127	27	16
2	308	155	135	27	13
3	581	195	224	98	50
4	923	413	379	27	15
5	326	151	126	23	14
6	263	129	112	39	22
7	383	153	115	25	13
8	530	185	188	88	47
9	314	119	125	26	12
10	416	203	157	25	12
11	779	373	405	41	29
12	746	327	271	37	23
13	1829	993	472	89	59
14	290	145	108	30	15
15	1613	629	471	47	41
16	992	497	557	75	49
17	1412	549	470	65	44
18	620	303	243	21	10
19	302	131	117	21	12
20	1412	493	81	32	24
В среднем	720,8	314,6	244,15	43,15	26,0

При оценке результатов экспериментов следует учитывать, что испытания для алгоритмов АГ, АП, АГП состоят в вычислении только значений минимизируемой функции; для методов БАГМП и АГМП при каждом испытании дополнительно вычисляется значение первой производной функции $\varphi(x)$.

Как показывают результаты приведенных экспериментов, использование значений производных минимизируемой функции позволяет значительно (почти в 10 раз) повысить эффективность глобального поиска.

2.5. От эффективных последовательных алгоритмов глобального поиска — к параллельным вычислениям

Общие вычислительные процедуры параллельных синхронных и асинхронных методов оптимизации описаны в разделе 1.2 главы 1. В этом разделе мы конкретизируем эти процедуры, обобщив последовательную схему характеристической представимости (2.29)–(2.32) на параллельный случай.

Снова рассмотрим одномерную задачу (2.15) для области поиска $Q = [a, b]$ и классы параллельных синхронных методов оптимизации (1.18)–(1.23) и параллельных асинхронных методов (1.24)–(1.28). Как и в последовательном случае, будем считать, что условие остановки в этих схемах отсутствует, т. е. в (1.23) $H_{k(n)}(\bullet) = 1$ и в (1.28) $H_k(\bullet) = 1$ для любого $k \geqslant 1$, что, как и ранее, обеспечивает формирование бесконечной последовательности испытаний $\{x^s\} = x^1, x^2, \ldots, x^s, \ldots$. В синхронном случае сохраним за символом n обозначение номера текущей итерации, $p = p(n)$ будет определять количество процессоров, используемых на n-й итерации (и, следовательно, количество испытаний данной итерации), а $k = k(n)$ — суммарное количество испытаний, выполненных в процессе всех n итераций.

2.5.1. Параллельные синхронные характеристические алгоритмы глобального поиска

Определение 2.3. Алгоритм решения задачи (2.15) называется *параллельным синхронным характеристическим методом*, если, начиная с некоторого номера итерации $n_0 \geqslant 1$, выбор координат испытаний $(n+1)$-й итерации $(n \geqslant n_0)$ заключается в выполнении следующих действий.

1. Задать набор

$$\Lambda_n = \{x_0, x_1, \ldots, x_\tau\} \tag{2.56}$$

конечного числа $\tau + 1 = \tau(n) + 1$ точек области $Q = [a, b]$, полагая, что $a \in \Lambda_n$, $b \in \Lambda_n$, все координаты предшествующих испытаний $x^i \in \Lambda_n$, $1 \leqslant i \leqslant k$, и множество Λ_n упорядочено (нижним индек-

сом) по возрастанию координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_{\tau-1} < x_\tau = b. \tag{2.57}$$

2. Каждому интервалу (x_{i-1}, x_i), $1 \leqslant i \leqslant \tau$, поставить в соответствие число $R(i)$, называемое характеристикой этого интервала.

3. Характеристики $R(i)$, $1 \leqslant i \leqslant \tau$, упорядочить в порядке убывания:

$$R(t_1) \geqslant R(t_2) \geqslant \ldots \geqslant R(t_{\tau-1}) \geqslant R(t_\tau). \tag{2.58}$$

4. В последовательности (2.58) выбрать $p = p(n+1) \leqslant \tau$ наибольших характеристик с номерами t_j, $1 \leqslant j \leqslant p$, и в интервалах, соответствующих этим характеристикам, провести испытания новой $(n+1)$-й итерации в точках множества

$$T(n+1) = \{x^{k+1}, \ldots, x^{k+p}\} \tag{2.59}$$

согласно правилам

$$x^{k+j} = d(t_j) \in (x_{t_j-1}, x_{t_j}) \tag{2.60}$$

для $1 \leqslant j \leqslant p$.

Примерами параллельных характеристических алгоритмов могут служить параллельные версии методов сканирования, ломаных, Кушнера и информационно-статистического алгоритма, последовательные варианты которых описаны в предшествующем разделе.

Класс параллельных характеристических методов введен и исследован в работе [53]. В данной работе установлена преемственность свойств сходимости параллельных характеристических методов по отношению к последовательным прототипам, а также получены оценки ускорения параллельных характеристических алгоритмов.

Далее мы приводим полное теоретическое обоснование свойств синхронных параллельных характеристических методов в отношения характера их сходимости и эффективности распараллеливания.

Напомним обозначения $z_j = \varphi(x_j)$ для значений целевой функции в точках $x_j \in \Lambda_k$ и введем величины

$$\xi_i = \min\{z_{i-1}, z_i\},$$

$$\widetilde{\xi}_i = \begin{cases} \xi_i, & \text{если} \quad x_{i-1} \in \{x^s\} \text{ и } x_i \in \{x^s\}; \\ z_{i-1}, & \text{если} \quad x_{i-1} \in \{x^s\} \text{ и } x_i \notin \{x^s\}; \\ z_i, & \text{если} \quad x_{i-1} \notin \{x^s\} \text{ и } x_i \in \{x^s\}. \end{cases}$$

Теорема 2.4. *Пусть точка x^* является предельной точкой (точкой накопления) последовательности поисковых испытаний $\{x^k\}$, порождаемой синхронным характеристическим алгоритмом при решении задачи* (2.15) *на отрезке $[a, b]$, причем $x^* \neq a$ и $x^* \neq b$. Предположим, что характеристики $R(i)$ и правила выбора точки очередного испытания $d(t)$ удовлетворяют следующим требованиям*:

а) *если при $n \to \infty$ точка $\overline{x} \in [x_{i(n)-1}, x_{i(n)}]$ и $x_{i(n)-1} \to \overline{x}$, $x_{i(n)} \to \overline{x}$, тогда*

$$R(i(n)) \to -\mu\varphi(\overline{x}) + c; \tag{2.61}$$

б) *в случае, когда, начиная с некоторого шага поиска, интервал (x_{i-1}, x_i), $i = i(n)$, не содержит точек поисковых испытаний, т. е. существует номер $\widetilde{n} \geqslant 1$ такой, что для всех $n \geqslant \widetilde{n}$*

$$(x_{i-1}, x_i) \cap \{x^k\} = \varnothing, \tag{2.62}$$

для характеристики интервала справедливо

$$\lim_{n\to\infty} R(i) > -\mu\widetilde{\xi}_i + c; \tag{2.63}$$

в) *при выборе очередного испытания из множества* (2.59) *имеет место соотношение*

$$\max\{x^{k+j} - x_{t_j-1},\ x_{t_j} - x^{k+j}\} \leqslant \nu(x_{t_j} - x_{t_j-1}), \tag{2.64}$$

где μ, c, ν — некоторые константы, причем $\mu \geqslant 0$, $0 < \nu < 1$.

Тогда последовательность $\{x^k\}$ содержит две подпоследовательности, одна из которых сходится к x^* слева, а другая справа.

Доказательство. Рассмотрим вначале случай, когда $x^* \notin \{x^k\}$. Обозначим через $s = s(n)$, $n \geqslant 1$, номер интервала (x_{s-1}, x_s), содержащего предельную точку x^* после выполнения n-й итерации. Очевидно, что после первой итерации $[x_{s-1}, x_s] \subseteq [a, b]$. После попадания в интервал (x_{s-1}, x_s) точки очередного испытания

$x^{k+l} \in T(n+1)$ для нового интервала $(x_{s(n+1)-1}, x_{s(n+1)})$, содержащего x^*, согласно (2.64), справедлива оценка

$$x_{s(n+1)} - x_{s(n+1)-1} \leqslant \nu(x_{s(n)} - x_{s(n)-1}).$$

Но тогда после попадания с начала поиска λ испытаний в интервал с точкой x^* его длина будет удовлетворять неравенству

$$x_s - x_{s-1} \leqslant \nu^\lambda (b-a). \tag{2.65}$$

Поскольку точка x^* предельная, после образования на некотором шаге интервала (x_{s-1}, x_s) в него попадет бесконечное число испытаний, поэтому из (2.65) следует, что

$$\lim_{n\to\infty} (x_{s(n)} - x_{s(n)-1}) = 0. \tag{2.66}$$

На основании (2.66) в качестве искомых подпоследовательностей мы можем взять последовательность $\{x_{s(n)-1}\}$ левых и последовательность $\{x_{s(n)}\}$ правых концов интервалов, содержащих x^*.

Если $x^* \in \{x^k\}$, найдутся номера $w \geqslant 1$ и $q > k(w-1)$ такие, что $x^q = x^* \in T(w)$. Тогда при любом $n > w$ существует номер $j = j(n)$, $0 \leqslant j \leqslant \tau(n)$, для которого $x_j = x^*$. Допустим, что имеет место односторонняя сходимость к x^*, например, слева (случай односторонней сходимости справа рассматривается аналогично). Тогда найдутся номер итерации $\widetilde{w} \geqslant w$ и номер испытания $v > k(\widetilde{w})$ такие, что $x^v \in T(\widetilde{w})$, и при всех $n \geqslant \widetilde{w}$ испытания в интервал $(x_j, x_{j+1}) = (x^q, x^v)$ попадать не будут.

Из (2.62), (2.63) для интервала (x_j, x_{j+1}) мы получаем, что

$$\lim_{n\to\infty} R(j+1) > -\mu\widetilde{\xi}_{j+1} + c \geqslant -\mu\varphi(x^*) + c.$$

Одновременно для интервала (x_{j-1}, x_j) вследствие соотношения (2.61) справедливо соотношение

$$\lim_{n\to\infty} R(j) = -\mu\varphi(x^*) + c, \tag{2.67}$$

поэтому начиная с некоторой итерации будет выполняться неравенство

$$R(j+1) > R(j).$$

Однако вследствие решающего правила (2.56)–(2.60) данное соотношение противоречит невозможности проведения испытаний в интервале (x_j, x_{j+1}). Данное противоречие завершает доказательство.

Следствие. *В вычислительную схему характеристического алгоритма, удовлетворяющего условиям теоремы 2.4, можно ввести условие остановки вида*

$$\min_{1 \leqslant j \leqslant p} (x_{t_j} - x_{t_j - 1}) \leqslant \varepsilon, \tag{2.68}$$

где номера t_j, $1 \leqslant j \leqslant p$, из (2.58), а $\varepsilon > 0$ — заданная точность поиска (по координате), т. е. прекращать вычисления, когда длина минимального интервала среди интервалов итерации с максимальными характеристиками станет меньше заданной точности ε. Тогда процесс поиска будет остановлен через конечное число шагов.

Доказательство. Вначале укажем, что на конечном отрезке $[a, b]$ последовательность испытаний всегда будет иметь хотя бы одну предельную точку x^*. Обозначим через $s = s(k)$, $k \geqslant 1$, номер интервала, содержащего точку x^* на k-м шаге поиска. Так как данная точка предельная, то в интервал (x_{s-1}, x_s) попадет бесконечное число испытаний, и для него будет иметь место соотношение (2.66), из которого следует, что условие (2.68) неизбежно выполнится на некотором шаге поиска. Заметим, что если точка x^* не является внутренней, для справедливости (2.68) достаточно односторонней сходимости.

Замечание. Условие (2.64) эквивалентно неравенству

$$\min\{x^{k+j} - x_{t_j - 1},\ x_{t_j} - x^{k+j}\} \geqslant (1 - \nu)(x_{t_j} - x_{t_j - 1}).$$

Действительно, легко увидеть, что, к примеру,

$$x^{k+j} - x_{t_j - 1} = (x_{t_j} - x_{t_j - 1}) - (x_{t_j} - x^{k+j}) \geqslant$$
$$\geqslant (x_{t_j} - x_{t_j - 1}) - \nu(x_{t_j} - x_{t_j - 1}) = (1 - \nu)(x_{t_j} - x_{t_j - 1}).$$

Для интервала $x_{t_j} - x^{k+j}$ имеет место аналогичная оценка, откуда следует справедливость рассматриваемого неравенства.

Проверим выполнимость условий теоремы 2.4 для рассмотренных примеров характеристических алгоритмов.

Метод последовательного сканирования

Если интервал (x_{i-1}, x_i) стягивать в точку, характеристика метода (2.33) стремится к нулю. Поэтому в (2.61) в качестве констант μ и c можно взять $\mu = c = 0$. Если же в интервал (x_{i-1}, x_i), начиная с некоторого шага поиска испытания, попадать не будут, то его длина (совпадающая с характеристикой), будет оставаться положительной, что обеспечит выполнимость условия (2.63) для выбранных μ и c. Что касается неравенства (2.64), то оно, очевидно, выполняется при $\nu = 0{,}5$.

Метод Кушнера [57]

Предположим, что функция $\varphi(x)$ ограничена на отрезке $[a, b]$ конечными величинами $\varphi_{\min}$ и $\varphi_{\max}$, т. е. $\varphi_{\min} \leqslant \varphi(x) \leqslant \varphi_{\max}$, $x \in [a, b]$.

Если для любого шага k параметр метода $\delta_k > \delta > 0$, то $\varphi_k^* - z_i - \delta_k < -\delta < 0$ и $\varphi_k^* - z_{i-1} - \delta_k < -\delta < 0$, то

$$R(i(k)) = -\frac{4(\varphi_k^* - \delta_k - z_i)(\varphi_k^* - \delta_k - z_{i-1})}{x_i - x_{i-1}} \to -\infty,$$

когда длина интервала (x_{i-1}, x_i) стремится к нулю. Поэтому (2.61) выполняется при $\mu = 0$ и формальном значении $c = -\infty$. (Доказательство теоремы 2.4 нисколько не изменится, если в правой части (2.61) предел будет равен ∞.)

Поскольку для любого интервала с ненулевой длиной его характеристика будет иметь конечное значение, (2.63) также справедливо.

Для испытания x^{k+j} очередной итерации (2.59) введем обозначение $t = t_j$ и рассмотрим разности

$$x^{k+j} - x_{t-1} = \beta_1(x_t - x_{t-1}), \quad x_t - x^{k+j} = \beta_2(x_t - x_{t-1}),$$

где

$$\beta_1 = \frac{z_{t-1} - \varphi_k^* + \delta_k}{z_{t-1} + z_t - 2(\varphi_k^* - \delta_k)}, \quad \beta_2 = \frac{z_t - \varphi_k^* + \delta_k}{z_{t-1} + z_t - 2(\varphi_k^* - \delta_k)}.$$

Введем вспомогательную функцию $\sigma(w,\alpha) = \frac{w+\alpha}{w+2\alpha}$. Так как $\min\{z_{t-1}, z_t\} \geqslant \varphi_k^*$, то $\beta_1 \leqslant \sigma(z_{t-1}-\varphi_k^*, \delta_k)$, $\beta_2 \leqslant \sigma(z_t-\varphi_k^*, \delta_k)$.

При положительных w функция $\sigma(w,\alpha)$ убывает по α, так как $\sigma'_\alpha(w,\alpha) = -w/(w+2\alpha)^2 < 0$, поэтому для $\delta_k > \delta > 0$ выполняется

$$\beta_1 \leqslant \sigma(z_{t-1}-\varphi_k^*, \delta), \quad \beta_2 \leqslant \sigma(z_t-\varphi_k^*, \delta).$$

В свою очередь функция $\sigma(w,\alpha)$ при положительных α возрастает по w, так как $\sigma'_w(w,\alpha) = \alpha/(w+2\alpha)^2 > 0$, поэтому

$$\max\{\sigma(z_{t-1}-\varphi_k^*, \delta),\ \sigma(z_t-\varphi_k^*, \delta)\} \leqslant \sigma(\varphi_{\max}-\varphi_{\min}, \alpha).$$

Последнее неравенство означает, что

$$\max\{\beta_1, \beta_2\} \leqslant \sigma(\varphi_{\max}-\varphi_{\min}, \delta) = \frac{\varphi_{\max} - \varphi_{\min} + \delta}{\varphi_{\max} - \varphi_{\min} + 2\delta}.$$

Именно последнюю величину можно выбрать в качестве константы ν для соотношения (2.64), так как очевидно, что $0 < \nu < 1$.

Метод ломаных [36]

Для рассмотрения условий теоремы сделаем предположение, что минимизируемая функция $\varphi(x)$ удовлетворяет условию Липшица с константой $L > 0$, т. е.

$$|\varphi(x') - \varphi(x'')| \leqslant L|x' - x''|, \quad x', x'' \in [a, b], \tag{2.69}$$

и, кроме того, параметр метода $m > L$.

В силу липшицевости функция $\varphi(x)$ непрерывна, и, следовательно, при стягивании интервала (x_{i-1}, x_i) к точке $\overline{x}$ характеристика интервала будет стремиться к величине $-\varphi(\overline{x})$, т. е. для выполнимости (2.61) можно положить $\mu = 1$ и $c = 0$. Проверим теперь для данных μ и c справедливость неравенства (2.63) для интервала (x_{i-1}, x_i), удовлетворяющего (2.62). Воспользуемся простым соотношением

$$\min\{z_{i-1}, z_i\} = \frac{1}{2}(z_{i-1} + z_i - |z_{i-1} - z_i|) \tag{2.70}$$

и оценим характеристику (2.34) метода, воспользовавшись условием Липшица и учитывая, что длина интервала остается, начиная

с некоторого шага поиска, положительной и неизменной:

$$m\frac{x_i - x_{i-1}}{2} - \frac{z_i + z_{i-1}}{2} > L\frac{x_i - x_{i-1}}{2} - \frac{z_i + z_{i-1}}{2} \geqslant$$
$$\geqslant \frac{|z_{i-1} - z_i| - (z_{i-1} + z_i)}{2} = -\min\{z_{i-1}, z_i\}.$$

Полученные соотношения устанавливают справедливость (2.63).

Для определения величины ν в (2.64) оценим величину

$$x_{t_j} - x^{k+j} = \frac{x_{t_j} - x_{t_j-1}}{2} + \frac{z_{t_j} - z_{t_j-1}}{2m} \leqslant$$
$$\leqslant \frac{x_{t_j} - x_{t_j-1}}{2} + L\frac{x_{t_j} - x_{t_j-1}}{2m} = \left(1 + \frac{L}{m}\right)\frac{x_{t_j} - x_{t_j-1}}{2}.$$

Аналогичная оценка имеет место и для интервала (x_{t_j-1}, x^{k+j}), поэтому можно взять $\nu = 0{,}5(1 + L/m)$. Очевидно, что $\nu > 0$ и, поскольку $m > L$, то $\nu < 1$.

Базовый информационно-статистический алгоритм глобального поиска (АГП)

Предположив липшицевость (2.69) целевой функции, нетрудно показать, что данный метод также удовлетворяет условиям теоремы 2.4 с $\mu = 4$, $c = 0$ и $\nu = 0{,}5(1 + 1/r)$.

Вследствие липшицевости при стягивании интервала (x_{i-1}, x_i) к точке $\overline{x}$ его характеристика будет стремиться к величине $-4\varphi(\overline{x})$, т. е. для выполнимости (2.61) можно положить $\mu = 4$ и $c = 0$.

Для проверки (2.63) при условии (2.62) рассмотрим два случая. Пусть сначала для интервала (x_{i-1}, x_i) справедливо $z_{i-1} = z_i$. Тогда характеристика

$$R(i) = m(x_i - x_{i-1}) - 2(z_i + z_{i-1}) > -2(z_i + z_{i-1}) = -4\min\{z_{i-1}, z_i\},$$

поскольку длина интервала (x_{i-1}, x_i), начиная с некоторой итерации n_Δ, перестает изменяться, т. е. существует константа $\Delta > 0$ такая, что при $n > n_\Delta$ имеет место $x_i - x_{i-1} > \Delta > 0$.

Предположим теперь, что $z_{i-1} \neq z_i$. Представим характеристику (2.36) в виде

$$R(i) = |z_i - z_{i-1}|(\beta + \frac{1}{\beta}) - 2(z_i + z_{i-1}), \qquad (2.71)$$

где вследствие (2.37), (2.38) $0 < \beta = \frac{m(x_i - x_{i-1})}{|z_i - z_{i-1}|} < 1$. В этом случае величина $\beta + 1/\beta > 2$, поэтому

$$R(i) > 2|z_i - z_{i-1}| - 2(z_i + z_{i-1}) = -4\min\{z_i, z_{i-1}\}.$$

Что касается условия (2.64), то в соответствии с (2.38) справедлива оценка

$$x_{t_j} - x^{k+j} = \frac{x_{t_j} - x_{t_j-1}}{2} + \frac{z_{t_j} - z_{t_j-1}}{2m} \leqslant$$
$$\leqslant \frac{x_{t_j} - x_{t_j-1}}{2} + M\frac{x_{t_j} - x_{t_j-1}}{2m} = \left(1 + \frac{1}{r}\right)\frac{x_{t_j} - x_{t_j-1}}{2}.$$

Длину интервала (x_{t_j-1}, x^{k+j}) можно оценить аналогичным образом, поэтому в качестве ν можно выбрать величину $0{,}5(1 + 1/r)$, которая очевидным образом удовлетворяет условию $0 < \nu < 1$, так как $r > 1$.

Информационный алгоритм глобального поиска с испытаниями во внутренних точках (АГПВТ)

Для данного метода, как и для АГП, предположив липшицевость целевой функции, получим, что при стягивании интервала (x_{i-1}, x_i) к точке $\overline{x}$ его характеристика будет стремиться к величине $-4\varphi(\overline{x})$, т. е. для выполнимости (2.61) можно положить $\mu = 4$ и $c = 0$.

Для характеристик «крайних» интервалов (2.63) принимает вид $\lim\limits_{n\to\infty} R(1) > -4z_1$ для первого интервала и $\lim\limits_{n\to\infty} R(\tau) > -4z_k$ для последнего, а поскольку длина крайних интервалов при условии (2.62) всегда положительна, то и выполнимость данных неравенств и, следовательно, (2.63) также очевидна.

Рассмотрим характеристику «внутреннего» интервала (x_{i-1}, x_i), $1 < i \leqslant k = \tau - 1$. Запишем ее в виде

$$R(i) = m\Delta_i(1 - r^{-2}) + \frac{r(|z_i - z_{i-1}| - M\Delta_i r^{-1})^2}{M\Delta_i} - 4\widetilde{\xi}_i,$$

где $\Delta_i = x_i - x_{i-1}$, $\widetilde{\xi}_i = \min\{z_{i-1}, z_i\}$. Поскольку второе слагаемое неотрицательно, параметр $r > 1$, то при $\Delta_i > 0$ выполняется $R(i) > -4\widetilde{\xi}_i$, т. е. (2.63).

В отношении (2.64) очевидно, что $\nu = 0{,}5(1 + 1/r)$.

Теорема 2.5. *Если в условиях теоремы 2.4 $\mu = 0$, тогда любая точка области поиска является предельной точкой последовательности поисковых испытаний.*

Доказательство. Предположим, что некоторая точка $x' \in [a, b]$ не является предельной для последовательности поисковых испытаний, порождаемой алгоритмом. Это означает, что начиная с некоторой итерации $u \geqslant 1$, испытания в интервал (x_{q-1}, x_q), $q = q(n)$, такой, что $x_{q-1} \leqslant x' \leqslant x_q$, попадать не будут. (Если $x' \in (a, b)$ и для некоторого m справедливо $x' = x^m$, существуют два таких интервала, и можно взять любой из них.)

Тогда для любой итерации $n \geqslant u$ в соответствии с (2.63) имеет место

$$\lim_{n\to\infty} R(q) > c. \tag{2.72}$$

Но последовательность испытаний, будучи ограниченной пределами отрезка $[a, b]$, содержит хотя бы одну сходящуюся подпоследовательность. Для интервала (x_{s-1}, x_s), $s = s(n)$, содержащего предельную точку данной подпоследовательности, вследствие двусторонней сходимости справедливо соотношение (2.61), т. е.

$$\lim_{n\to\infty} R(s) = c. \tag{2.73}$$

Это означает, что вследствие (2.72), (2.73) существует такой номер $w \geqslant u$, что для всех $n \geqslant w$ будет выполнено неравенство

$$R(q) > R(s),$$

которое в соответствии с правилом (2.58) основной схемы параллельного характеристического алгоритма противоречит невозмож-

ности попадания испытаний в интервал (x_{q-1}, x_q) при $n \geqslant w \geqslant u$, а следовательно, и исходному предположению. *Теорема доказана.*

Данная теорема устанавливает условия так называемой «всюду плотной» сходимости, когда метод сходится ко всем точкам области поиска, в том числе, разумеется, и к точкам глобального минимума. Указанный тип сходимости адекватен таким классам задач, для которых невозможно построить оценки экстремума по конечному числу испытаний (например, для класса непрерывных функций), и в этом случае обеспечить сходимость к глобально-оптимальному решению можно лишь за счет свойства всюду плотной сходимости.

Среди примеров, которые мы рассмотрели, подобной сходимостью обладают методы перебора и метод Кушнера, а среди других известных алгоритмов — методы из работ [17, 25].

Для иллюстрации поведения методов данного типа продемонстрируем динамику поиска методом перебора (рис. 2.8). Под графиком минимизируемой функции штрихами отображены координаты проведенных испытаний. Эксперименты выполнены с помощью программной системы АБСОЛЮТ [12], представляющей собой развитый комплекс исследования и анализа эффективности одномерных характеристических алгоритмов оптимизации.

Условия всюду плотной сходимости обеспечивают достаточные условия сходимости к глобально-оптимальному решению задачи оптимизации. Однако характер такого рода сходимости требует дополнительных способов исследования эффективности распределения точек поисковой последовательности. Один из таких подходов основан на получении оценок относительной плотности размещения испытаний в различных подобластях области поиска в сравнении с плотностью точек в окрестности глобального минимума [17].

На рис. 2.9 приведен пример размещения испытаний всюду плотно сходящегося метода Х. Кушнера с автоматической настройкой параметра δ_k [57]. Иллюстрация ясно показывает существенно неравномерный характер размещения испытаний в этом методе (с их концентрацией в окрестности решения), несмотря на всюду плотный тип сходимости.

Другой тип поведения параллельных характеристических алгоритмов устанавливает следующая теорема.

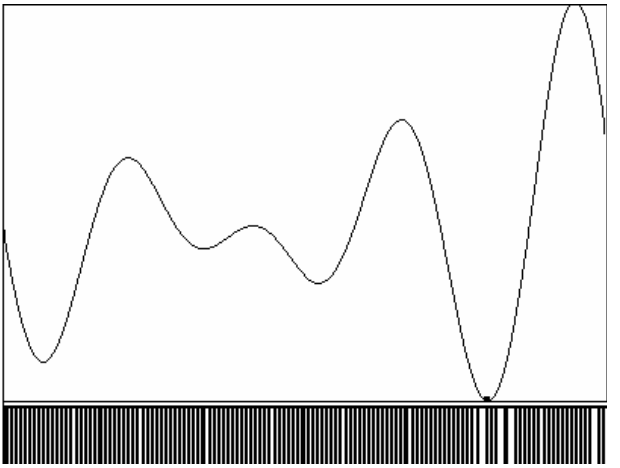

Рис. 2.8. Размещение испытаний в методе равномерного перебора. Всюду плотная сходимость

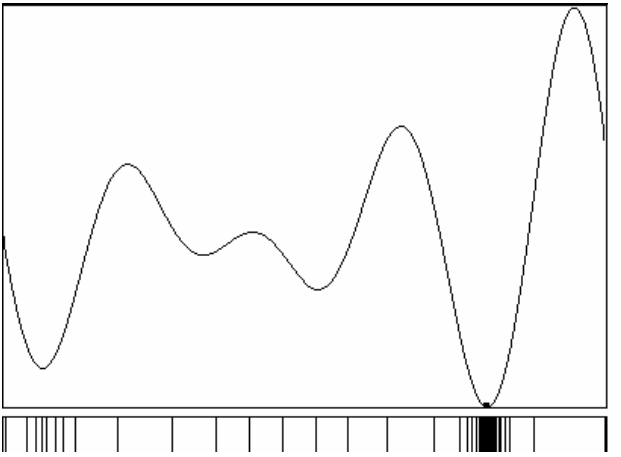

Рис. 2.9. Характер концентрации испытаний в методе Х. Кушнера. Сходимость в пределе всюду плотная, но неравномерная

Теорема 2.6. *Предположим, что*:

а) *целевая функция* $\varphi(x)$ *задачи* (2.15) *непрерывна на отрезке* $[a, b]$;

б) *для параллельного характеристического алгоритма выполнены условия теоремы 2.4 при* $\mu > 0$;

в) *количество параллельных испытаний в итерации равномерно ограничено, т. е.*

$$p(n) \leqslant P,$$

где $P \geqslant 1$ — *заданная целая константа*;

г) x^* — *предельная точка последовательности поисковых испытаний, генерируемой алгоритмом.*

Тогда:

1) $\varphi(x^k) \geqslant \varphi(x^*)$, $k \geqslant 1$;

2) *если существует еще одна предельная точка* $x^{**} \neq x^*$, *то* $\varphi(x^{**}) = \varphi(x^*)$;

3) *если целевая функция* $\varphi(x)$ *является непрерывной и имеет конечное число локальных минимумов на отрезке* $[a, b]$, *то* x^* *является точкой локального минимума целевой функции в области* Q.

Доказательство. Обозначим через $j = j(n)$ номер интервала (x_{j-1}, x_j) такого, что $x_{j-1} \leqslant x^* \leqslant x_j$. Укажем, что для характеристики данного интервала вследствие теоремы 2.4 выполняется соотношение (2.67).

Докажем первое утверждение. Допустим, что на итерации с номером $u \geqslant 1$ проведено испытание в точке $x^m \in T(u)$ такое, что

$$\varphi(x^m) < \varphi(x^*). \tag{2.74}$$

Обозначим через $s = s(n)$, $n \geqslant u$ номер точки x^m в ряду (2.57), т. е. $x^m = x_s$.

Покажем, что точка x^m также является предельной точкой последовательности испытаний. Если это не так, то для характеристики $R(s)$ интервала (x_{s-1}, x_s) вследствие (2.63) имеет место соотношение

$$\lim_{n \to \infty} R(s) > -\mu\varphi(x^m) + c. \tag{2.75}$$

Заметим, что если $s = 0$, аналогичное (2.75) неравенство имеет место для характеристики $R(s+1)$ интервала (x_s, x_{s+1}).

Теперь, принимая во внимание (2.67) и (2.74), мы можем утверждать, что начиная с некоторой итерации, будет выполнено неравенство

$$R(s) > R(j),$$

что противоречит нашему допущению, что точка x^m не является предельной точкой последовательности поисковых испытаний.

Вследствие того, что функция $\varphi(x)$ непрерывна, неравенство $\varphi(x) < \varphi(x^*)$ выполняется в некоторой окрестности $\Theta \subseteq [a, b]$ точки x^*. Так как точка x^m предельная, эта окрестность будет содержать более $P + 1$ точки последовательности испытаний $\{x^k\}$. Но тогда более чем P интервалов, образованных этими точками, будут иметь характеристики, большие, чем характеристика $R(j)$ из (2.67). Это значит, что в соответствии с правилом (2.58) испытания с некоторого момента не смогут попадать в интервал (x_{j-1}, x_j), однако это невозможно, поскольку точка x^* — предельная.

Второе утверждение теоремы является простым следствием первого, поскольку, предположив, к примеру, что $\varphi(x^{**}) < \varphi(x^*)$, мы получим на некоторой итерации испытание x^m такое, что $\varphi(x^m) < \varphi(x^*)$, что невозможно.

Завершим теорему доказательством локальной оптимальности точки x^*. Предположим обратное, а именно что x^* не является точкой локального минимума. Тогда существует непустая окрестность $\Theta \subseteq [a, b]$ точки x^*, в которой функция $\varphi(x)$ строго монотонна, то есть либо слева, либо справа от x^* имеет место неравенство $\varphi(x) < \varphi(x^*)$, $x \in \Theta$. Так как сходимость к точке x^* двусторонняя, в последовательности испытаний обязательно появится точка $x^j \in \Theta$ такая, что $\varphi(x^j) < \varphi(x^*)$ (в случаях $x^* = a$ или $x^* = b$ для последнего неравенства достаточно односторонней сходимости). Этот факт противоречит первому утверждению и завершает доказательство теоремы.

Итак, если в условиях (2.61), (2.63) константа μ положительна (а это означает, что метод *учитывает* в асимптотике информацию о функции), то поведение алгоритма становится более целенаправ-

ленным: он ищет только такие точки, которые обладают свойством локальной минимальности. Более того, утверждение 2 теоремы говорит, что метод не может сходиться к разновысоким локальным минимумам.

Из числа рассмотренных нами методов таким свойством обладают метод ломаных и АГП, а из других известных алгоритмов — весь спектр информационно-статистических алгоритмов [42, 44, 67], а также методы [61].

Для примера проиллюстрируем работу АГП (рис. 2.10). Как и ранее, штрихами под графиком функции отмечаются координаты испытаний метода.

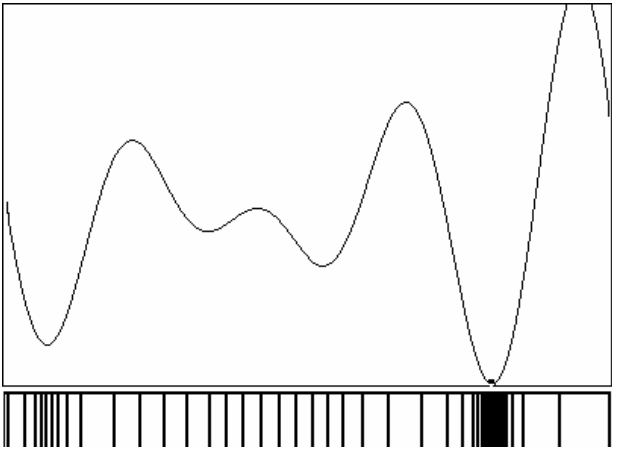

Рис. 2.10. Размещение испытаний в методе АГП

Приведенный рисунок показывает, что в той части области поиска, где нет глобального минимума, метод строит *редкую сетку* испытаний, а сгущение, обусловленное сходимостью, наблюдается лишь в окрестности глобального минимума.

Теорема 2.6 не гарантирует сходимость к глобальному минимуму исследуемой задачи оптимизации. Для получения таких гарантий (достаточных условий сходимости) необходимо предъявить опре-

деленные требования к целевой функции. С этой целью рассмотрим класс функций, удовлетворяющих на отрезке $[a, b]$ условию Гёльдера

$$|\varphi(x') - \varphi(x'')| \leqslant L \sqrt[N]{|x' - x''|}\,, x', x'' \in [a, b], \tag{2.76}$$

где $N \geqslant 1$ — целое число, а константа $L \geqslant 0$ называется константой Гёльдера. Функции вида (2.76) возникают, в частности, при редукции многомерных функций на отрезок с помощью отображений Пеано, что позволяет свести решение многомерной задачи оптимизации к решению одномерной задачи (2.15) (см. раздел 1.3). Отметим, что при $N = 1$ условие (2.76) превращается в условие Липшица (2.69).

Теорема 2.7. *Пусть функция $\varphi(x)$ удовлетворяет на отрезке $[a, b]$ условию Гёльдера* (2.76). *Тогда любая точка x^* глобального минимума функции $\varphi(x)$ на данном отрезке является предельной точкой последовательности поисковых испытаний, порождаемой характеристическим алгоритмом при решении задачи (2.15), если выполняются условия* (2.61) *и* (2.64) *теоремы 2.4, а для интервала (x_{i-1}, x_i) со свойством* (2.62) *справедливо неравенство*

$$\lim_{n\to\infty} R(i) > \alpha \sqrt[N]{(x_i - x_{i-1})} - \mu\widetilde{\xi}_i + c, \tag{2.77}$$

где константы μ, c и ν удовлетворяют условиям теоремы 2.4, *а параметр*

$$\alpha \geqslant \frac{\mu L}{\sqrt[N]{2}}, \tag{2.78}$$

если $x_{i-1} \in \{x^s\}$ и $x_i \in \{x^s\}$ и

$$\alpha \geqslant \frac{2\mu L}{\sqrt[N]{2}}, \tag{2.79}$$

когда либо x_{i-1}, либо x_i не принадлежит последовательности испытаний $\{x^s\}$.

Доказательство. Предположим, что глобальный минимум x^* не является предельной точкой поисковых испытаний, и пусть $i = i(n)$ — номер интервала (x_{i-1}, x_i), содержащего эту точку на n-й итерации поиска, то есть $x_{i-1} \leqslant x^* \leqslant x_i$. Тогда, начиная с неко-

торой итерации, испытания в данный интервал попадать не будут. Находясь в интервале (x_{i-1}, x_i), точка x^* может быть записана в виде $x^* = \lambda x_{i-1} + (1-\lambda)x_i$, $0 \leqslant \lambda \leqslant 1$. Тогда вследствие (2.76) имеем

$$\varphi(x_{i-1}) - \varphi(x^*) \leqslant L\sqrt[N]{(x^* - x_{i-1})} = L\sqrt[N]{(1-\lambda)(x_i - x_{i-1})},$$
$$\varphi(x_i) - \varphi(x^*) \leqslant L\sqrt[N]{(x_i - x^*)} = L\sqrt[N]{\lambda(x_i - x_{i-1})},$$

откуда

$$\varphi(x_i) + \varphi(x_{i-1}) \leqslant 2\varphi(x^*) + L(\sqrt[N]{\lambda} + \sqrt[N]{1-\lambda})\sqrt[N]{(x_i - x_{i-1})} \leqslant$$
$$\leqslant 2\varphi(x^*) + L\sqrt[N]{(x_i - x_{i-1})} \max_{0\leqslant\lambda\leqslant 1}\left\{\sqrt[N]{\lambda} + \sqrt[N]{1-\lambda}\right\} =$$
$$= 2\varphi(x^*) + \frac{2L}{\sqrt[N]{2}}\sqrt[N]{(x_i - x_{i-1})}, \quad (2.80)$$

поскольку функция $\sqrt[N]{\lambda} + \sqrt[N]{1-\lambda}$ достигает своего максимума на отрезке $[0, 1]$ в точке $\lambda = 0{,}5$.

В случае (2.78) $\widetilde{\xi}_i = \xi_i$, причем вследствие (2.70) $\xi_i \geqslant \frac{1}{2}(z_{i-1} + z_i)$, поэтому

$$\lim_{n\to\infty} R(i) > \left(\alpha\sqrt[N]{(x_i - x_{i-1})} - \frac{\mu}{2}(z_{i-1} + z_i)\right) + c.$$

Последнее неравенство в сочетании с (2.78) и (2.80) позволяет получить оценку

$$\lim_{n\to\infty} R(i) > -\mu\varphi(x^*) + c. \quad (2.81)$$

Пусть теперь $\widetilde{\xi}_i = z_i$ в ситуации (2.79) (случай $\widetilde{\xi}_i = z_{i-1}$ рассматривается аналогично). Тогда с учетом (2.80) и глобальной минимальности x^*

$$\lim_{n\to\infty} R(i) > \alpha\sqrt[N]{(x_i - x_{i-1})} - \frac{\mu}{2}z_i + c \geqslant$$
$$\geqslant \alpha\sqrt[N]{(x_i - x_{i-1})} - 2\mu\varphi(x^*) + \mu\varphi(x_{i-1}) - \frac{2\mu L}{\sqrt[N]{2}}\sqrt[N]{(x_i - x_{i-1})} + c \geqslant$$
$$\geqslant (\alpha - \frac{2\mu L}{\sqrt[N]{2}})\sqrt[N]{(x_i - x_{i-1})} - \mu\varphi(x^*) + c \geqslant -\mu\varphi(x^*) + c,$$

т. е. (2.81) снова выполнено.

В то же время последовательность испытаний имеет на отрезке $[a, b]$ хотя бы одну предельную точку $\overline{x}$, для которой необходимо выполнено $\varphi(\overline{x}) \geqslant \varphi(x^*)$ и, кроме того, согласно теореме 2.4, для интервала (x_{j-1}, x_j), $j = j(n)$, содержащего $\overline{x}$, имеет место соотношение

$$\lim_{n\to\infty} R(j) = -\mu\varphi(\overline{x}) + c.$$

Но тогда, начиная с некоторой итерации поиска, получим неравенство $R(i) > R(j)$, которое из-за правила (2.58) противоречит невозможности проведения испытаний в интервале (x_{i-1}, x_i). *Доказательство завершено.*

Следствие 2.7.1. *При* $\mu = 0$ (2.77) *совпадает с* (2.63) *и теорема* 2.7 *становится идентичной теореме* 2.5; *при этом гельдеровость целевой функции необязательна.*

Следствие 2.7.2. *Если в условиях теоремы* $\mu > 0$ *и количество процессоров на итерации равномерно ограничено, то характеристический алгоритм сходится ко всем точкам глобального минимума и только к ним.*

Доказательство. Действительно, сходимость ко всем точкам глобального минимума — это результат самой теоремы, а невозможность сходимости к точкам, отличным от глобально оптимальных, — следствие утверждения 2 теоремы 2.6.

Установим достаточные условия сходимости для метода ломаных и АГП, в которых фактор $\mu > 0$ в предположении липшицевости (2.69) целевой функции. В этом случае условие (2.77) переходит в

$$\lim_{n\to\infty} R(i) > \frac{\mu L}{2}(x_i - x_{i-1}) - \mu\xi_i + c.$$

Для метода ломаных $N = 1$, и неравенство (2.77) очевидно выполняется при $m > L$.

Для характеристики АГП справедливо очевидное неравенство

$$R(i) \geqslant m(x_i - x_{i-1}) - 2(z_i + z_{i-1}) = m(x_i - x_{i-1}) - \frac{\mu}{2}(z_i + z_{i-1}),$$

откуда при $N = 1$ следует выполнимость (2.77) при $m > 2L$.

2.5.2. Условия безызбыточного распараллеливания

Распараллеливание последовательных характеристических алгоритмов по схеме (2.56)–(2.59) приводит к отказу от использования части поисковой информации при формировании координат x^{k+j}, $1 \leqslant j \leqslant p$, очередной итерации (см. (2.59)). Дело в том, что для последовательного алгоритма при выборе (единственной) точки очередного испытания известны результаты (значения функции) *всех* предшествующих испытаний. Для параллельного же метода выбор точки x^{k+j}, $1 < j \leqslant p$, выполняется без учета значений функции в точках x^{k+s}, $1 \leqslant s < j$, поскольку эти значения еще не вычислены. Чем выше уровень распараллеливания $p(n)$ в схеме (2.56)–(2.59), тем больше потери информации при планировании испытаний, и эти потери наиболее значительны при полном распараллеливании, когда $p(n) = \tau(n)$ из (2.56).

Неполный учет информации может привести к ситуации, когда параллельный алгоритм размещает испытания более плотно по сравнению с последовательным прототипом, т. е. генерирует *избыточные* испытания.

Охарактеризуем избыточность параллелизма количественно, следуя подходу работы [66]. Пусть $\{x^k\}$ и $\{y^s\}$ являются бесконечными ($\varepsilon = 0$ в условиях остановки (2.21), (2.68)) последовательностями испытаний, генерируемыми соответственно чисто последовательным характеристическим методом и его параллельным обобщением при решении одной и той же задачи (2.15). Совпадение этих последовательностей, т. е.

$$\{x^k\} = \{y^s\}, \tag{2.82}$$

свидетельствует о том, что параллельный алгоритм размещает испытания в тех же точках, что и чисто последовательный. Заметим, что (2.82) не означает выполнимости равенств $x^q = y^q$, $q \geqslant 1$.

Когда соотношение (2.82) имеет место, распараллеливание называется безызбыточным. Если же (2.82) не выполняется и $\{x^k\} \subseteq \subseteq \{y^s\}$, то параллельная схема обладает определенной избыточностью, для количественной характеристики которой введем *коэффи-*

циент избыточности

$$\lambda(s,q) = V(s,q)/(s-q), \quad s > q, \tag{2.83}$$

где

$$V(s,q) = \operatorname{card}\bigl(\{y^{q+1},\dots,y^s\}\setminus\{x^k\}\bigr)$$

является количеством избыточных точек в $\{y^s\}$ с $(q+1)$-го по s-е испытание. Очевидно, что $V(s,0)=0$ для любого $s \geqslant 1$ соответствует безызбыточному случаю (2.82).

Рассмотрим теперь параллельные характеристические алгоритмы, которые после некоторых начальных n_0 итераций всегда используют ровно два процессора, т. е.

$$p(n) = 2, \quad n > n_0. \tag{2.84}$$

Сделаем ряд не слишком ограничительных допущений. Предположим, что если точка испытания x^{k+q} очередной итерации выбирается на интервале (x_{j-1}, x_j) таком, что уже вычислены значения целевой функции z_{j-1}, z_j на концах этого интервала, то эта точка может быть представлена в виде

$$x^{k+q} = g_j + \operatorname{sign}(z_{j-1} - z_j)\theta_j, \tag{2.85}$$

где

$$g_j = (x_{j-1} + x_j)/2, \tag{2.86}$$

$$0 \leqslant \theta_j \leqslant \sigma(x_j - x_{j-1}), \tag{2.87}$$

причем $0 < \sigma \leqslant 0{,}5$. Заметим, что из (2.85)–(2.87) следует (2.64) при $\nu = 0{,}5 + \sigma$.

Если же на одном из концов интервала (x_{j-1}, x_j) значение целевой функции не вычислялось, будем считать, что

$$x^{k+q} = g_j. \tag{2.88}$$

Наряду с (2.85)–(2.88) допустим, что характеристики $R(i)$, $1 \leqslant i \leqslant \tau$, и величины θ_j, $1 \leqslant j \leqslant p$, зависят только от граничных точек соответствующих интервалов и вычисленных значений функ-

ции в этих точках, т. е.

$$R(i) = \Psi(x_{i-1}, x_i, z_{i-1}, z_i), \tag{2.89}$$

$$\theta_j = \Xi(x_{j-1}, x_j, z_{j-1}, z_j). \tag{2.90}$$

Наконец, предположим, что

$$y^s = x^s, \quad 1 \leqslant s \leqslant k(n_0), \tag{2.91}$$

т. е. начальные испытания последовательного и параллельного методов, не связанные с характеристическим правилом (2.56)–(2.59), являются одинаковыми.

Теорема 2.8. *Пусть функция $\varphi(x)$ удовлетворяет на отрезке $[a, b]$ условию Липшица* (2.22) *с константой L, для параллельного характеристического алгоритма и его последовательного прототипа выполнены соотношения* (2.84), (2.85), (2.87)–(2.91), (2.61), $\mu > 0$, *и, кроме того, характеристика*

$$R(i) > \mu L \Delta_i - \mu \widetilde{\xi}_i + c \tag{2.92}$$

для интервала длины $\Delta_i > 0$.

Тогда при $\sigma \leqslant 1/6$ справедлива оценка

$$\lambda(s, n_0) \leqslant \frac{E[(s - n_0)/6]}{s - n_0} < 0{,}17. \tag{2.93}$$

Здесь $E[\alpha]$ обозначает целую часть α.

Доказательство. Если после s-й итерации параллельного метода и k-й итерации последовательного метода справедливо равенство

$$(y_{t-1}, y_t) = (x_{q-1}, x_q), \quad t = t(s), \quad q = q(k), \tag{2.94}$$

и точки следующих испытаний y^{s+1} и x^{k+1} попадают соответственно в указанные интервалы, то вследствие (2.89)

$$R(t(s)) = R(q(k)) \tag{2.95}$$

и, согласно (2.32), (2.85), (2.90),

$$y^{s+1} = x^{k+1}. \tag{2.96}$$

Соотношения (2.94)–(2.96) при $s = k = n_0$ при наличии предположения (2.91) обеспечивают включение

$$\{x^k\} \subseteq \{y^s\},$$

которое позволяет оценить избыточность с помощью коэффициента (2.83).

Условия (2.61), (2.92) и (2.64) с $\nu = 0{,}5 + \sigma$ гарантируют выполнимость условий теоремы 2.6, согласно которой, при $\mu > 0$ предельными точками последовательностей $\{x^k\}$ и $\{y^s\}$ являются только точки глобального минимума функции $\varphi(x)$. Это означает, что множество предельных точек последовательности $\{y^s\}$ совпадает с множеством предельных точек последовательности $\{x^k\}$.

Пусть первые k точек последовательности $\{x^k\}$ упорядочены согласно (2.57) и $j = j(k)$ является номером интервала (x_{j-1}, x_j) такого, что $x_{j-1} \leqslant x^* \leqslant x_j$, где x^* — точка глобального минимума функции $\varphi(x)$. Вследствие (2.92) и условия Липшица для характеристики этого интервала имеем неравенство

$$R(j) > -\mu\varphi(x^*) + c. \tag{2.97}$$

Поскольку точка x^* — предельная, тогда, согласно (2.61), при $k \to \infty$ справедливо

$$R(j(k)) \to -\mu\varphi(x^*) + c + 0.$$

Принимая во внимание правило (2.58), можно утверждать, что любой интервал (x_{i-1}, x_i), $i = i(k)$, характеристика которого удовлетворяет (2.97), содержит по крайней мере одну точку последовательности $\{x^k\}$. В то же время любой интервал (x_{i-1}, x_i), $i = i(k)$, для которого

$$R(i) < -\mu\varphi(x^*) + c, \tag{2.98}$$

не может содержать точки последовательности $\{x^k\}$. Таким образом, избыточные испытания параллельного метода могут проводиться только в интервалах, для которых выполняется (2.98).

Снова рассмотрим отрезок $[x_{j-1}, x_j]$, $j = j(k)$, содержащий точку глобального минимума x^*. Очередное испытание в точке $x = x^{k+1}$,

принадлежащей этому отрезку, образует два новых подынтервала

$$[x_{j-1}, x], \quad [x, x_j], \tag{2.99}$$

один из которых содержит x^*. Допустим, что это первый из них, т. е.

$$x^* \in [x_{j-1}, x]. \tag{2.100}$$

Тогда для этого интервала справедливо неравенство (2.97). Покажем, что это же неравенство выполняется и для интервала $[x, x_j]$.

Введем следующие обозначения:

$$\Delta_x = x_j - x, \quad \xi_x = \min\{z_j, z\}, \quad z = z^{k+1}. \tag{2.101}$$

Рассмотрим случай, когда точки x_{j-1} и x_j являются координатами поисковых испытаний, т. е. в них вычислены значения функции z_{j-1} и z_j, поэтому $\widetilde{\xi}_j = \min\{z_{j-1}, z_j\}$. Оценим величину ξ_x. Пусть сначала $z_{j-1} < z_j$. Вследствие (2.100), (2.85), (2.87), (2.101)

$$x^* \leqslant x = g_j - \theta_j \leqslant g_j,$$
$$\xi_x \leqslant z \leqslant \varphi(x^*) + L(x - x^*) < \varphi(x^*) + L\Delta_x.$$

Если же $z_{j-1} \geqslant z_j$, рассмотрим два случая. В первом из них, когда $x^* > g_j$, имеем

$$x = g_j + \theta_j,$$
$$\xi_x \leqslant z \leqslant \varphi(x^*) + L(x - x^*) < \varphi(x^*) + L(x - g_j) =$$
$$= \varphi(x^*) + L\theta_j \leqslant \varphi(x^*) + L\sigma\Delta_j \leqslant \varphi(x^*) + L\Delta_x,$$

поскольку $\Delta_x \geqslant (1 - \nu)\Delta_j = (0{,}5 - \sigma)\Delta_j$ (см. замечание к теореме 2.4), и из-за $\sigma \leqslant 1/6$ выполняется неравенство

$$\Delta_x \geqslant \frac{1}{3}\Delta_j > \sigma\Delta_j.$$

Во втором случае, когда $x^* \leqslant g_j$, получаем

$$\xi_x = 0{,}5(z_j + z - |z_j - z|) \leqslant 0{,}5(z_j + z) \leqslant 0{,}5(z_{j-1} + z) \leqslant$$
$$\leqslant \varphi(x^*) + 0{,}5L(x - x_{j-1}) \leqslant \varphi(x^*) + \frac{L(1 + 2\sigma)}{2(1 - 2\sigma)}\Delta_x \leqslant \varphi(x^*) + L\Delta_x.$$

Это неравенство с учетом (2.92) позволяет оценить характеристику R_x интервала (x, x_j):

$$R_x > \mu L \Delta_x - \mu \xi_x + c \geqslant -\mu \varphi(x^*) + c,$$

откуда

$$R_x > -\mu \varphi(x^*) + c. \tag{2.102}$$

Неравенство (2.102) также выполняется, когда $x_j \notin \{x^s\}$, так как в этом случае, согласно (2.86), $x = g_j$. Поскольку $\varphi(x)$ удовлетворяет условию Липшица, то $z \leqslant \varphi(x^*) + L\Delta_x$, поэтому

$$R_x > \mu L \Delta_x - \mu z + c \geqslant -\mu \varphi(x^*) + c.$$

Таким образом, если после k испытаний существует пара интервалов типа (2.99), тогда выбор одновременно двух испытаний в соответствии с решающим правилом (2.56)–(2.59) и (2.84) не может генерировать избыточных испытаний. После $k(n_0)$ начальных испытаний (2.91) наихудшей ситуацией (в смысле генерирования избыточных испытаний на следующем шаге) может быть ситуация, когда существует только один интервал, удовлетворяющий условию (2.97), а именно некий интервал $[x_{j-1}, x_j]$, которому принадлежит глобальный минимайзер x^*, в то время как для остальных интервалов неравенство (2.97) нарушено. Это означает, что интервал $[x_{j-1}, x_j]$ имеет максимальную характеристику среди всех интервалов, поэтому в соответствии с правилом (2.58) первое испытание новой (n_0+1)-й итерации будет проведено именно в нем, а второе испытание будет избыточным.

Проведение испытания в интервале $[x_{j-1}, x_j]$приведет к появлению пары интервалов (2.99), для каждого из которых справедливо (2.97). Но тогда на (n_0+2)-й итерации два очередных испытания $y^q \in \{x^k\}$ будут проведены именно в них. Если одно из испытаний (n_0+3)-й итерации попадает в интервал (2.100), это вновь приводит к образованию пары (2.99) и отсутствию избыточных испытаний на этой итерации. Таким образом, источником избыточных испытаний на следующей (n_0+4)-й итерации может быть ситуация, когда пара испытаний (n_0+3)-й итерации содержит точки из (x, x_j), но не содержит точек из (2.100). Если это происходит на (n_0+3)-й итера-

ции, то (n_0+4)-я итерация может разместить одну точку в интервале (2.100), а другую в некотором интервале (x_{i-1}, x_i) таком, что его характеристика удовлетворяет неравенству (2.98). Последняя точка будет избыточной. После этого мы имеем ситуацию, идентичную той, которая сложилась после выполнения (n_0+1)-й итерации.

Таким образом, за три итерации с (n_0+1)-й по (n_0+3)-ю включительно, на которых выполнено 6 испытаний, мы можем получить не более одной избыточной точки, что свидетельствует о справедливости соотношения (2.93).

Следствие 2.8.1. *Если целевая функция $\varphi(x)$ имеет η точек глобального минимума и между двумя соседними такими точками имеется по меньшей мере одна точка последовательности $\{y^q\}$, построенной при проведении начальных итераций, тогда при выполнении условий теоремы 2.8 параллельный характеристический алгоритм с 2η параллельными процессорами на каждой итерации обеспечит избыточность в соответствии с (2.93).*

2.5.3. Численные эксперименты

Для иллюстрации теоретических выводов приведем результаты численных экспериментов, выполненных на многопроцессорном кластере Нижегородского госуниверситета и минисуперкомпьютере ALLIANT FX/80 на тестовых классах функций различной размерности.

Будем предполагать, что время, затрачиваемое на проведение одного испытания существенно превосходит временные затраты на реализацию вычислительной схемы алгоритма, которым в этом случае можно пренебречь. В данной ситуации мы можем в качестве оценки ускорения использовать ускорение в итерациях

$$s(p) = \frac{K(1)}{K(p)/p}, \quad p > 1$$

для оценки эффективности распараллеливания. Здесь $K(1)$ — количество испытаний, проведенных последовательным характеристическим алгоритмом (после выполнения начальных испытаний) при решении задачи и $K(p)$ — количество испытаний, реализован-

ных параллельным методом с p параллельными процессорами на каждой итерации при решении той же задачи.

При проведении экспериментов с одномерными параллельными характеристическими алгоритмами будем использовать набор тестовых функций, предложенных в работе [54]. Рассмотрим параллельные алгоритмы, построенные на базе последовательных методов Кушнера (МК), Пиявского (метод ломаных — МЛ) и информационного алгоритма АГПВТ. На основе трех функций из [54] и различных стартовых точек сформируем три задачи.

Задача 1. Функция № 2, стартовые точки 3, 4, 5;
Задача 2. Функция № 9, стартовые точки 5, 8, 11;
Задача 3. Функция № 19, стартовые точки 1, 2, 3.

В методах Кушнера и ломаных стартовые испытания проводились также в граничных точках a и b соответствующих интервалов поиска $[a, b]$; для этих методов были использованы точные значения констант Липшица, в отличие от АГПВТ, в котором применялась

Таблица 2.4. Результаты вычислительных экспериментов

q	p	МК		МЛ		АГПВТ	
		Число испытаний	Ускорение	Число испытаний	Ускорение	Число испытаний	Ускорение
1	1	4252	—	151	—	120	—
	2	4353	1,95	151	2,0	119	2,02
	3	4352	2,93	155	2,92	162	2,21
	4	4353	3,91	157	3,83	163	2,93
2	1	4622	—	124	—	125	—
	2	4621	2,0	123	2,02	125	2,0
	3	4622	3,0	122	3,05	126	2,98
	4	4373	4,24	125	3,7	127	3,94
3	1	2780	—	126	—	148	—
	2	2799	2,0	127	1,98	153	1,93
	3	2762	3,02	128	2,95	159	2,79
	4	2769	4,02	133	3,78	163	3,63

адаптивная оценка (2.19) при $r = 2$. В критерии остановки методов (2.68) была использована величина $\varepsilon = 0{,}0001(b - a)$. Результаты экспериментов приведены в таблице 2.4, где q обозначает номер задачи, а p — количество параллельных процессоров. Глобально-оптимальное решение было найдено во всех случаях.

Как следует из результатов экспериментов, в некоторых случаях методы показывают ускорение большее, чем число использованных процессоров (сверхлинейное ускорение). Такая ситуация является возможной в методах Кушнера и АГПВТ из-за того, что в данных алгоритмах поведение целевой функции оценивается адаптивно. Например, в АГПВТ гиперускорение произойдет, если в параллельной версии константа Липшица будет оценена точнее, чем в последовательной.

Следующие эксперименты проводились для многомерных задач в соответствии с подходом к редукции размерности на основе кривых Пеано, изложенным в главе 5. Во второй серии проводилась максимизация 100 двумерных функций вида

$$\varphi(y) = \left\{ \left(\sum_{i=1}^{7} \sum_{j=1}^{7} A_{ij} g_{ij}(y) + B_{ij} h_{ij}(y) \right)^2 + \right.$$
$$\left. + \left(\sum_{i=1}^{7} \sum_{j=1}^{7} C_{ij} g_{ij}(y) - D_{ij} h_{ij}(y) \right)^2 \right\}^{1/2}, \quad (2.103)$$

где

$$y = (y_1, y_2) \in R^2, \quad 0 \leqslant y_s \leqslant 1, \quad s = 1, 2,$$
$$g_{ij}(y) = \sin(i\pi y_1) \sin(j\pi y_2),$$
$$h_{ij}(y) = \cos(i\pi y_1) \cos(j\pi y_2),$$

а коэффициенты A_{ij}, B_{ij}, C_{ij}, D_{ij} равномерно распределены в интервале $[-1, 1]$. В качестве подтверждения существенной многоэкстремальности функций данного класса на рис. 2.11 приведены линии уровня одной из них.

Данные задачи редуцировались к одномерным с помощью разверток Пеано. Для решения редуцированных одномерных задач применялись два параллельных характеристических алгоритма до

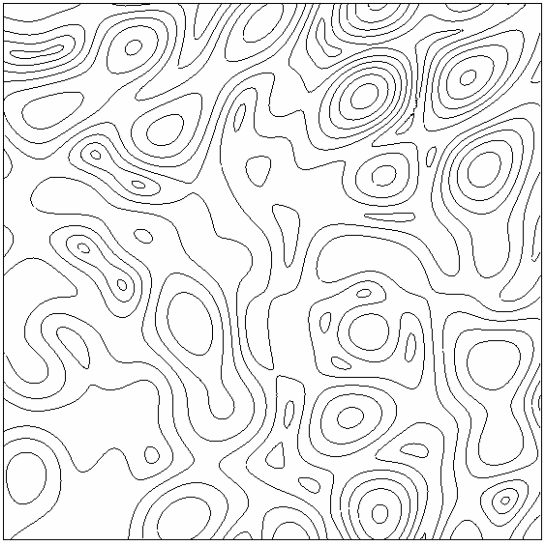

Рис. 2.11. Линии уровня оптимизируемой функции

достижения точности в условии остановки $\varepsilon = 0{,}001$. Первый алгоритм описан в работе [64] и использует фиксированную оценку константы Липшица (назовем его ПМФЛ — параллельный метод с фиксированной константой Липшица). Отметим, что получение указанной оценки потребовало 2000 вычислений значений функции на равномерной сетке. С помощью ПМФЛ глобально-оптимальное решение было найдено в 99 задачах из 100. В единственном случае решение не было найдено (как последовательным, так и параллельным методом) вследствие недостаточно точной оценки константы Липшица: полученная оценка была меньше, чем реальная константа. Второй алгоритм, используемый для сравнения, — это параллельный многомерный информационный алгоритм, использующий кривые Пеано (ПМИП). С помощью ПМИП оптимум был найден во всех 100 задачах. Результаты экспериментов (среднее число испытаний и ускорение) приведены в таблице 2.5, а на рис. 2.12 изображены точки первых 200 испытаний, проведенных алгоритмом ПМИП для одной из задач.

Таблица 2.5. Результаты сравнения алгоритмов

P	ПМФЛ		ПМИП	
	Число испытаний	Ускорение	Число испытаний	Ускорение
1	1013,25	—	1575,12	—
2	1012,82	2,001	1569,08	1,974
3	1012,33	3,003	1562,61	3,024
4	1012,00	4,005	1599,92	3,938

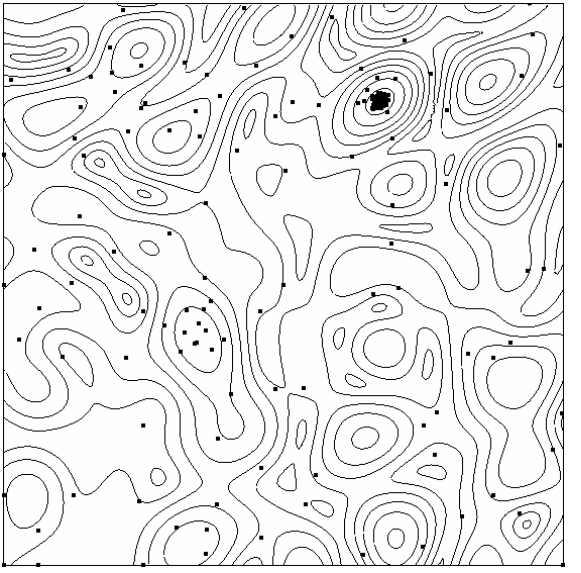

Рис. 2.12. Точки испытаний

Третья серия экспериментов состояла в минимизации многомерной функции

$$\varphi(y) = \frac{\pi}{N}\left\{\sin^2(\pi z_1) + 5(z_N - 1)^2 + \sum_{i=1}^{N-1}\left[(z_i - 1)^2(1 + \sin^2(\pi z_{i+1}))\right]\right\},$$

где $z_i = 1 + 0{,}25(y_i - 1)$, $-10 \leqslant y_i \leqslant 10$, $1 \leqslant i \leqslant N$, для размерностей $N = 2, 3, 4$. Данная функция является модификацией тестовой функции из [58].

Для минимизации указанной функции использовались два метода: алгоритм ПМИП и метод ломаных, обобщенный для многомерного случая в соответствии со схемой редукции размерности на основе кривых Пеано (МЛП). В критерии остановки метода (2.68) использовалась величина $\varepsilon = 10^{-5}$, а константа Липшица оценивалась адаптивно в соответствии с (2.19). При этом исходя из достаточных условий сходимости алгоритмов параметр $r = 3{,}5$ для ПМИП и $r = 1{,}7$ для МЛП; точка глобального оптимума найдена во всех задачах. Результаты экспериментов приведены в таблице 2.6.

Таблица 2.6. Результаты сравнения алгоритмов

N	**P**	**ПМИП**		**МЛП**	
		Число испытаний	**Ускорение**	**Число испытаний**	**Ускорение**
2	1	1082	—	1027	—
	2	1084	1,996	1028	1,998
	3	1083	2,997	1050	2,934
	4	1136	3,810	1224	3,356
3	1	4784	—	4649	—
	2	4776	2,003	5018	1,853
	3	4776	3,005	4647	3,001
	4	4768	4,016	4480	4,151
4	1	7958	—	6274	—
	2	7876	2,021	5238	2,396
	3	6750	3,537	6486	2,902
	4	7820	4,072	6468	3,880

Как уже отмечалось ранее, адаптивная оценка поведения функции в процессе поиска может приводить к сверхлинейному ускорению. В целом же численные эксперименты подтверждают основные теоретические положения о сходимости параллельных алгоритмов и эффективности их распараллеливания, приведенные в настоящей главе.

2.5.4. Параллельные асинхронные характеристические алгоритмы глобального поиска

Как уже было отмечено, синхронная схема обладает тем недостатком, что процессоры, закончившие свои испытания в рамках итерации раньше других, будут простаивать, ожидая полного завершения итерации. В структуре характеристических методов возможно введение асинхронизма, который позволяет обеспечить исключение простоя процессоров, возникающего из-за различий во времени выполнения вычислений разными процессорами, и тем самым повысить эффективность распараллеливания по сравнению с синхронной схемой.

Прежде чем описать структуру асинхронного характеристического алгоритма, вернемся к общей вычислительной схеме асинхронного метода (глава 1) и напомним ряд важных аспектов и обозначений. Поскольку в асинхронном алгоритме итерация распределена во времени, при описании метода удобно опираться на количество испытаний k. В связи с этим введем множество

$$w(k) = \{w_1, w_2, \ldots, w_k\} \tag{2.104}$$

координат завершенных испытаний, $k \geqslant 1$, и множество

$$v(k) = \{v_1, v_2, \ldots, v_{p(k)}\} \tag{2.105}$$

координат, в которых проводятся новые испытания после завершения предшествующих k испытаний. Здесь $p(k)$ — количество процессоров, осуществляющих очередные испытания,

$$\begin{aligned} w_j &= x^{i_j}, \quad 1 \leqslant j \leqslant k, \\ v_s &= x^{i_s}, \quad 1 \leqslant s \leqslant p(k), \quad 1 \leqslant i_j, \quad i_s \leqslant k + p(k). \end{aligned}$$

Очевидно, что

$$w(k) \cup v(k) = \{x^1, \ldots, x^k, \ldots, x^{k+p(k)}\}.$$

Определение 2.4. Алгоритм решения задачи (2.15) называется *параллельным асинхронным характеристическим методом*, если решающее правило алгоритма по выбору координат испытаний состоит в следующем.

Первые $k_1 \geqslant 1$ испытаний проводятся на отрезке $[a, b]$ произвольным образом, но начиная с номера шага поиска $k = k_0 \leqslant k_1$, $k_1 - k_0 \leqslant p(k_0)$, на котором завершено $w(k_0)$ испытаний и продолжаются $v(k_0) = k_1 - k_0$ испытаний, вступает в действие следующее правило выбора координат новых испытаний.

1. При отсутствии свободных процессоров алгоритм находится в состоянии ожидания. Если закончили проведение испытаний $1 \leqslant \pi = \pi(k) \leqslant p(k)$ процессоров, которые вычислили значения функции $z^{i_j} = \varphi(x^{i_j})$ в точках $x^{i_1}, x^{i_2}, \ldots, x^{i_\pi}$, $x^{i_j} \in v(k)$, $1 \leqslant j \leqslant \pi$, то устанавливается номер шага $k = k + \pi$, формируются множества $v(k) = v(k - \pi) \backslash \{x^{i_1}, x^{i_2}, \ldots, x^{i_\pi}\}$, $w(k) = w(k - \pi) \cup \{x^{i_1}, x^{i_2}, \ldots, x^{i_\pi}\}$, и осуществляется переход к следующему пункту.

2. Задается набор

$$\Lambda_k = \{x_0, x_1, \ldots, x_\tau\}$$

конечного числа $\tau + 1 = \tau(n) + 1$ точек области $Q = [a, b]$ при условии, что $a \in \Lambda_k$, $b \in \Lambda_k$, $w(k) \subseteq \Lambda_k$, $v(k) \subseteq \Lambda_k$, и множество Λ_k упорядочено (нижним индексом) по возрастанию координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_{\tau-1} < x_\tau = b.$$

3. Формируется список номеров Σ_k допустимых интервалов (x_{i-1}, x_i), в качестве которых рассматриваются такие интервалы, граничные точки которых являются координатами завершенных испытаний, либо граничными точками отрезка $[a, b]$. Иными словами, номер i интервала (x_{i-1}, x_i) принадлежит множеству Σ_k, если ни один из концов интервала не является координатой незавершенного испытания.

4. Каждому допустимому интервалу (x_{i-1}, x_i), $i \in \Sigma_k$, ставится в соответствие число $R(i)$, называемое его характеристикой.

5. Характеристики $R(i)$, $1 \leqslant i \leqslant \tau$, упорядочиваются в порядке убывания:

$$R(t_1) \geqslant R(t_2) \geqslant \ldots \geqslant R(t_{\tau-1}) \geqslant R(t_\tau). \qquad (2.106)$$

6. В последовательности (2.106) выбираются π наибольших характеристик с номерами t_j, $1 \leqslant j \leqslant \pi$, и в интервалах, соответствующих этим характеристикам, проводятся испытания в точках

множества

$$T(k) = \{x^{\eta+1}, \ldots, x^{\eta+\pi}\}, \tag{2.107}$$

согласно правилам

$$x^{\eta+j} = d(t_j) \in (x_{t_j-1}, x_{t_j})$$

для $1 \leqslant j \leqslant \pi$, где $\eta = \mathrm{card}(w(k)) + \mathrm{card}(v(k))$.

7. Точки множества $T(k)$ добавляются в множество $v(k)$, и осуществляется переход к п. 1.

Теоретические результаты в отношении асинхронных характеристических алгоритмов в основном повторяют результаты для синхронного случая. Для их точной формулировки рассмотрим последовательность испытаний $\{x^k\}$, генерируемую асинхронным характеристическим алгоритмом, и предположим для упрощения, что число процессоров фиксировано и равно p. Свяжем с последовательностью испытаний p подпоследовательностей $\{\vartheta_s^j\}$, $1 \leqslant j \leqslant s$, где ϑ_s^j — момент завершения s-го испытания, выполненного j-м процессором. Рассмотрим величины

$$\vartheta_s^j - \vartheta_{s-1}^j, \quad 1 \leqslant j \leqslant p, \quad s > 1. \tag{2.108}$$

Предположим, что количество свободных процессоров $\pi \leqslant \mathrm{card}(\Sigma_k)$, т. е. ни одному из освободившихся процессоров не надо ждать появления свободного для проведения нового испытания интервала. Если при этом временем формирования множества (2.107) пренебречь, то величина $\vartheta_s^j - \vartheta_{s-1}^j$ есть время выполнения s-го испытания j-м процессором. В общем случае, однако, $\vartheta_s^j - \vartheta_{s-1}^j$ представляет собой время, прошедшее от завершения $(s-1)$-го испытания до завершения s-го испытания j-го процессора и может включать время на выбор точки нового испытания для j-го процессора в соответствии с пунктами 2–6 асинхронного алгоритма.

Допустим, что времена (2.108) ограничены сверху, т. е. существует константа $\widetilde{\vartheta}$, такая что

$$\vartheta_s^j - \vartheta_{s-1}^j \leqslant \widetilde{\vartheta}, \quad 1 \leqslant j \leqslant p, \quad s > 1. \tag{2.109}$$

Соотношение (2.109) фактически гарантирует, что любое испытание в асинхронной схеме будет обязательно завершено за конечное время.

В этом случае (при наличии дополнительного условия (2.109)) для асинхронных методов справедливы теоремы 2.4–2.7, доказательство которых отличается незначительными деталями.

Количественное повышение эффективности асинхронных алгоритмов по сравнению с синхронными продемонстрировано в главе 4 в результате проведения вычислительных экспериментов на существенно многоэкстремальных тестовых функциях.

Глава 3

Алгоритмы глобального поиска для одномерных многоэкстремальных задач оптимизации с нелинейными ограничениями

3.1. Постановка задачи

В этой главе мы будем рассматривать одномерные задачи условной оптимизации. Пусть $\varphi(x)$ и $g_j(x) \leqslant 0$, $1 \leqslant j \leqslant m$ — действительные функции, определенные на отрезке $[a, b]$ вещественной оси, и требуется найти значения x^*, удовлетворяющие условию

$$\varphi(x^*) = \min\{\varphi(x)\colon x \in [a, b],\ g_j(x) \leqslant 0,\ 1 \leqslant j \leqslant m\}. \tag{3.1}$$

Как и ранее, мы предполагаем, что целевая функция φ (в дальнейшем обозначаемая также g_{m+1}) и левые части ограничений g_j, $1 \leqslant j \leqslant m$ удовлетворяют *условию Липшица* с соответствующими константами L_j, $1 \leqslant j \leqslant m+1$, а именно:

$$\begin{gathered} |g_j(x_1) - g_j(x_2)| \leqslant L_j |x_1 - x_2|, \\ 1 \leqslant j \leqslant m+1, \quad x_1,\ x_2 \in [a, b]. \end{gathered} \tag{3.2}$$

Пример 3.1. Рассмотрим задачу вида (3.1) при $x \in [0{,}6; 2{,}2]$, $m = 2$,

$$\begin{aligned} \varphi(x) &= \cos(18x - 3)\sin(10x - 7) + 1{,}5, \\ g_1(x) &= \exp(-x/2)\sin(6x - 1{,}5), \\ g_2(x) &= |x|\sin(2\pi x - 0{,}5). \end{aligned}$$

Точное решение задачи $x^* = 2{,}0795$, $\varphi(x^*) = 0{,}565$. Графики функций $g_1(x)$, $g_2(x)$, $\varphi(x)$ представлены на рис. 3.1.

3.2. Частичная вычислимость и индексная схема учета ограничений

Задача (3.1) может быть рассмотрена в постановке, когда каждая функция g_j, $1 \leqslant j \leqslant m+1$, определена и вычислима лишь в соот-

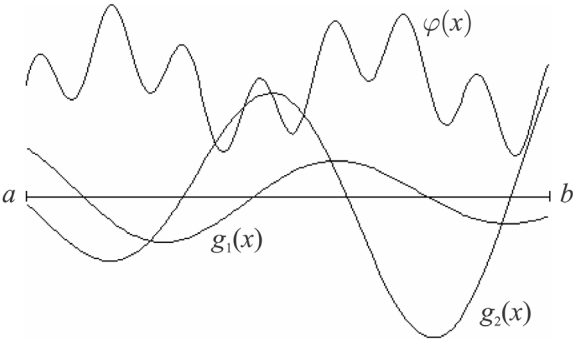

Рис. 3.1. Одномерная задача глобальной оптимизации

ветствующей подобласти $Q_j \subset [a, b]$, где

$$Q_1 = [a, b], \quad Q_{j+1} = \{x \in Q_j\colon g_j(x) \leqslant 0\}, \quad 1 \leqslant j \leqslant m. \tag{3.3}$$

Так, например, в задачах оптимального проектирования некоторые характеристики технической системы оказываются неопределенными, если не выполнены представленные частью ограничений задачи (3.1) условия функционирования системы.

С учетом условий (3.3), исходная задача (3.1) может быть представлена в виде

$$\varphi(x^*) = \min\{g_{m+1}(x)\colon x \in Q_{m+1}\}. \tag{3.4}$$

Для целей дальнейшего изложения введем классификацию точек x из области поиска $[a, b]$ с помощью *индекса* $\nu = \nu(x)$, где $\nu - 1$ — число ограничений, которые выполняются в этой точке. Указанный индекс ν определяется условиями

$$g_j(x) \leqslant 0, \quad 1 \leqslant j \leqslant \nu - 1, \quad g_\nu(x) > 0, \tag{3.5}$$

где последнее неравенство несущественно, если $\nu = m + 1$, и удовлетворяет неравенствам

$$1 \leqslant \nu = \nu(x) \leqslant m + 1.$$

На рис. 3.2 приведена задача из примера 3.1 в предположении частичной вычислимости функций. Изображены дуги функций

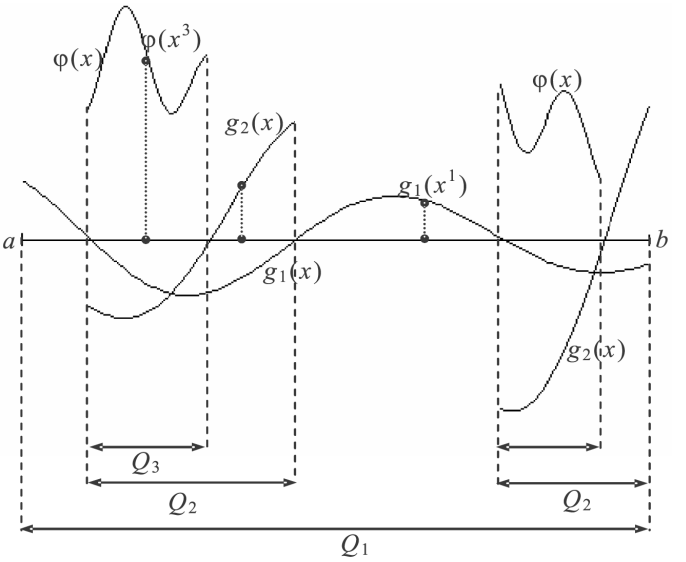

Рис. 3.2. Задача с частично вычислимыми функциями

ограничений $g_1(x)$, $g_2(x)$ и целевой функции $\varphi(x)$, определенные на соответствующих множествах Q_j из (3.3). Изображена также точка x^1 с индексом $\nu(x^1) = 1$, точка x^2 с индексом $\nu(x^2) = 2$ и точка x^3 с индексом $\nu(x^3) = 3$.

Данная классификация порождает функцию

$$f(x) = g_\nu(x), \quad \nu = \nu(x), \tag{3.6}$$

определенную и вычислимую всюду на $[a, b]$. Ее значение в точке x есть либо значение левой части ограничения, нарушенного в этой точке (случай, когда $\nu \leqslant m$), либо значение минимизируемой функции (случай, когда $\nu = m + 1$). Поэтому определение значения $f(x)$, $x \in [a, b]$, сводится к последовательному вычислению величин $g_j(x)$, $1 \leqslant j \leqslant \nu = \nu(x)$, т. е. последующее значение $g_{j+1}(x)$ вычисляется лишь в том случае, когда $g_j(x) \leqslant 0$. Процесс вычислений завершается либо в результате установления неравенства $g_j(x) > 0$, либо в результате достижения значения $\nu(x) = m + 1$.

Описанная процедура, называемая *испытанием* в точке x, автоматически приводит к определению индекса ν этой точки. Пара значений

$$z = f(x) = g_\nu(x), \quad \nu = \nu(x), \tag{3.7}$$

порожденная испытанием в точке $x \in [a, b]$, называется *результатом испытания*.

На рис. 3.3 приведен график функции $f(x)$ из (3.6), который образован дугами функций ограничений $g_1(x)$, $g_2(x)$ и целевой функции $\varphi(x)$ для задачи из примера 3.1.

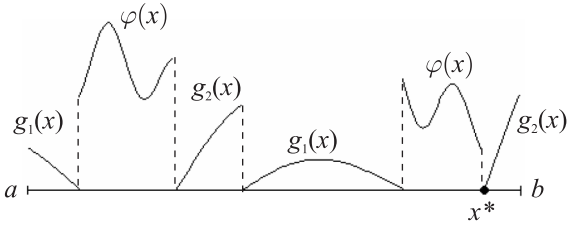

Рис. 3.3. «Индексная» функция

Поскольку в общем случае задача (3.4) может не иметь решения (т. е. допустимая область Q_{m+1} может оказаться пустой в силу несовместимости ограничений), с ней связывается некоторая вспомогательная задача, всегда имеющая решение. Так как условия (3.5) эквивалентны условиям

$$x \in Q_\nu, \quad x \notin Q_{\nu+1},$$

то вспомогательная задача, всегда имеющая решение, может быть записана в виде

$$g_M^* = g_M(x_M^*) = \min\{g_M(x)\colon x \in Q_M\}, \tag{3.8}$$

где M — максимально возможное значение индекса, т. е.

$$1 \leqslant M = \max\{\nu(x)\colon x \in [a, b]\} \leqslant m + 1. \tag{3.9}$$

Поскольку множество Q_M всегда непусто, задача (3.8) всегда имеет решение. При $M = m+1$ решение $x^* = x_{m+1}^*$ является так-

же решением исходной задачи (3.1). При $M < m+1$ необходимо выполняющееся неравенство $g_M^* < 0$ может использоваться как индикатор несовместимости ограничений.

Основная идея индексного подхода состоит в редукции условной задачи (3.8) к безусловной задаче

$$\psi(x^*) = \min\{\psi(x)\colon x \in [a,b]\}, \tag{3.10}$$

где

$$\psi(x) = \begin{cases} g_\nu(x)/L_\nu, & \nu < M, \\ (g_M - g_M^*)/L_M, & \nu = M. \end{cases}$$

Как результат дуги функции $\psi(x)$ будут липшицевыми с константой $L = 1$ на каждом множестве Q_ν, $1 \leqslant \nu \leqslant M$ (на рис. 3.4 приведен график функции $\psi(x)$ для задачи из примера 3.1). Эта новая функция будет иметь разрывы первого рода на граничных точках множеств Q_ν из (3.3). Но тем не менее можно оценить точку глобального оптимума x_M^*, используя результаты k испытаний для (3.7) в точках $x^1, \ldots, x^k$ из $[a,b]$ [44].

В самом деле, из условия Липшица следует, что

$$x_M^* \in \{x \in [a,b]\colon |x - x^i| \geqslant \psi(x^i),\ 1 \leqslant i \leqslant k\}. \tag{3.11}$$

Так, например, на рис. 3.4 изображен случай $k = 4$. Объединение отрезков, выделенных на рисунке жирной линией, является дополнением до множества из правой части (3.11) и не содержит оптимальную точку.

Заметим, что максимальный индекс M, значения констант Липшица L_ν, $1 \leqslant \nu \leqslant M$, и величина g_M^* являются неизвестными. Однако эти проблемы можно преодолеть, используя вместо этих величин их адаптивные оценки, получаемые в процессе решения задачи на основании результатов испытаний.

После того, как введены все требуемые понятия, перейдем к изложению индексного алгоритма.

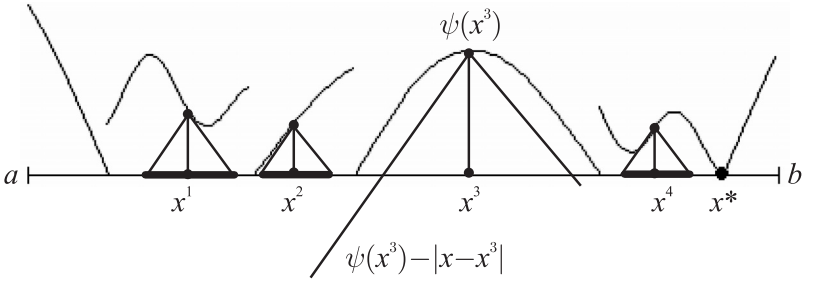

Рис. 3.4. Оценка оптимальной точки

3.3. Индексный алгоритм глобального поиска

Первое испытание осуществляется в произвольной внутренней точке $x^1 \in (a, b)$. Выбор точки x^{k+1}, $k \geqslant 1$, любого последующего испытания определяется следующими правилами.

Правило 1. Перенумеровать точки $x^1, \ldots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_i < \ldots < x_k < x_{k+1} = b, \tag{3.12}$$

и сопоставить им значения $z_i = g_\nu(x_i)$, $\nu = \nu(x_i)$, $1 \leqslant i \leqslant k$, из (3.7), вычисленные в этих точках; точки $x_0 = a$ и $x_{k+1} = b$ введены дополнительно (значения z_0 и z_{k+1} не определены) для удобства последующих обозначений.

Правило 2. Провести классификацию номеров i, $1 \leqslant i \leqslant k$ точек из ряда (3.12) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$I_\nu = \{i\colon\ 1 \leqslant i \leqslant k,\ \nu = \nu(x_i)\}, \quad 1 \leqslant \nu \leqslant m+1, \tag{3.13}$$

содержащих номера всех точек x_i, $1 \leqslant i \leqslant k$, имеющих индексы, равные одному и тому же значению ν. Граничные точки $x_0 = a$ и $x_{k+1} = b$ интерпретируются как имеющие нулевые индексы, и им сопоставляется дополнительное множество $I_0 = \{0, k+1\}$.

Определить максимальное значение индекса

$$M = \max\{\nu = \nu(x_i),\ 1 \leqslant i \leqslant k\}. \tag{3.14}$$

Правило 3. Вычислить текущие нижние границы

$$\mu_\nu = \max\left\{\frac{|z_i - z_j|}{x_i - x_j}, \quad i, j \in I_\nu,\ i > j\right\} \tag{3.15}$$

для неизвестных констант Липшица L_ν функций g_ν, $1 \leqslant \nu \leqslant m+1$. Если множество I_ν содержит менее двух элементов или если μ_ν из (3.15) оказывается равным нулю, то принять $\mu_\nu = 1$. Из (3.15) следует, что оценки μ_ν являются неубывающими, начиная с момента, когда (3.15) порождает первое положительное значение μ_ν.

Правило 4. Для всех непустых множеств I_ν, $1 \leqslant \nu \leqslant m+1$, вычислить оценки

$$z_\nu^* = \begin{cases} 0, & \nu < M, \\ \min\{g_\nu(x_i)\colon i \in I_\nu\}, & \nu = M. \end{cases} \tag{3.16}$$

Правило 5. Для каждого интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k+1$, вычислить *характеристику* $R(i)$, где

$$\begin{aligned} R(i) &= \Delta_i + \frac{(z_i - z_{i-1})^2}{r_\nu^2 \mu_\nu^2 \Delta_i} - 2\frac{(z_i + z_{i-1} - 2z_\nu^*)}{r_\nu \mu_\nu}, \\ &\nu = \nu(x_{i-1}) = \nu(x_i), \\ R(i) &= 2\Delta_i - 4\frac{(z_i - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_i) > \nu(x_{i-1}), \\ R(i) &= 2\Delta_i - 4\frac{(z_{i-1} - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_{i-1}) > \nu(x_i), \\ &\Delta_i = x_i - x_{i-1}. \end{aligned} \tag{3.17}$$

Величины $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$, являются параметрами алгоритма. Подходящий выбор значений r_ν позволяет использовать произведение $r_\nu \mu_\nu$ как оценку константы Липшица L_ν, $1 \leqslant \nu \leqslant m+1$.

Правило 6. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i)\colon 1 \leqslant i \leqslant k+1\}. \tag{3.18}$$

Правило 7. Провести очередное испытание в серединной точке интервала (x_{t-1}, x_t), если индексы его концевых точек не совпадают, т. е.

$$x^{k+1} = \frac{x_t + x_{t-1}}{2}, \quad \nu(x_{t-1}) \neq \nu(x_t).$$

В противном случае провести испытание в точке

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \frac{z_t - z_{t-1}}{2r_\nu \mu_\nu}, \quad \nu = \nu(x_{t-1}) = \nu(x_t). \tag{3.19}$$

Описанные правила можно дополнить условием остановки, прекращающим испытания, если

$$x_t - x_{t-1} \leqslant \varepsilon, \tag{3.20}$$

где t из (3.18) и $\varepsilon > 0$ есть заданная точность.

Сформулируем условия сходимости алгоритма в виде следующей теоремы.

Теорема 3.1. *Если рассмотренный индексный алгоритм применяется для решения задачи* (3.1), *и при этом выполняются условия*:

1) *каждая функция* g_j, $1 \leqslant j \leqslant m+1$, *удовлетворяет на отрезке* $[a, b]$ *условию Липшица с константой* L_j, *т. е.*

$$|g_j(x_1) - g_j(x_2)| \leqslant L_j |x_1 - x_2|, \quad 1 \leqslant j \leqslant m+1, \quad x_1, x_2 \in [a, b];$$

2) *для величин* μ_ν *из* (3.15), *начиная с некоторого шага, справедливы неравенства*

$$r_\nu \mu_\nu > 2L_\nu, \quad 1 \leqslant \nu \leqslant m+1,$$

то множество предельных точек последовательности $\{x^k\}$, *порождаемой индексным алгоритмом, совпадет с множеством решений задачи (3.1) при* $\varepsilon = 0$ *в условии остановки (3.20), причем индекс каждой предельной точки равен* M, *а сходимость к любой предельной точке* $\overline{x}$ *является двусторонней, если* $\overline{x} \neq a$ *и* $\overline{x} \neq b$.

Данная теорема приведена без доказательства, которое может быть найдено, например, в работе [43].

Результаты эксперимента. В качестве иллюстрации рассмотрим задачу из примера 3.1. В предположении частичной вычислимости дуги функций задачи, соответствующие областям Q_j, $1 \leqslant j \leqslant 2$, из (3.3), представлены на рис. 3.6.

Для решения этого примера было использовано три метода.

1. Перебор по равномерной сетке (точность поиска решения $\varepsilon = 10^{-5}$).

2. Метод штрафных функций, т. е. минимизация функции

$$p(x) = \varphi(x) + (\max\{g_1(x), 0\})^2 + (\max\{g_2(x), 0\})^2$$

с помощью алгоритма глобального поиска. Координаты точек испытаний, осуществленных АГП в процессе решения задачи, отмечены на рис. 3.5 вертикальными штрихами. Координаты испытаний, выполненных в близких точках, отмечены темным прямоугольником.

3. Индексный алгоритм с параметрами $r_\nu = 2$, $1 \leqslant \nu \leqslant 3$, и $\varepsilon = 10^{-5}$. Координаты точек испытаний, осуществленных алгоритмом в процессе решения задачи, отмечены на рис. 3.6 тремя рядами

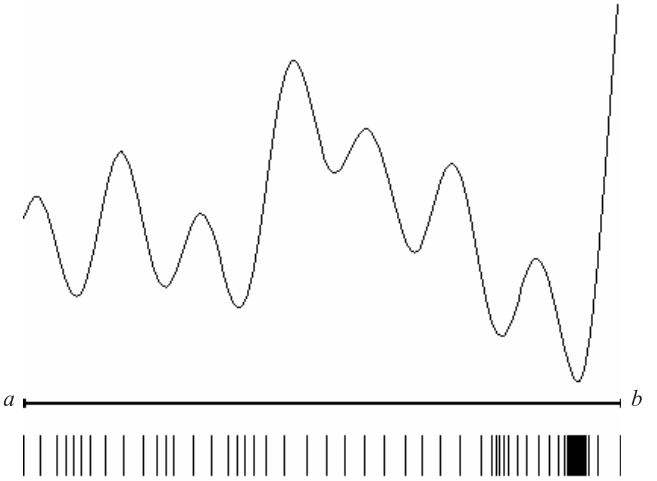

Рис. 3.5. Решение задачи с использованием метода штрафных функций

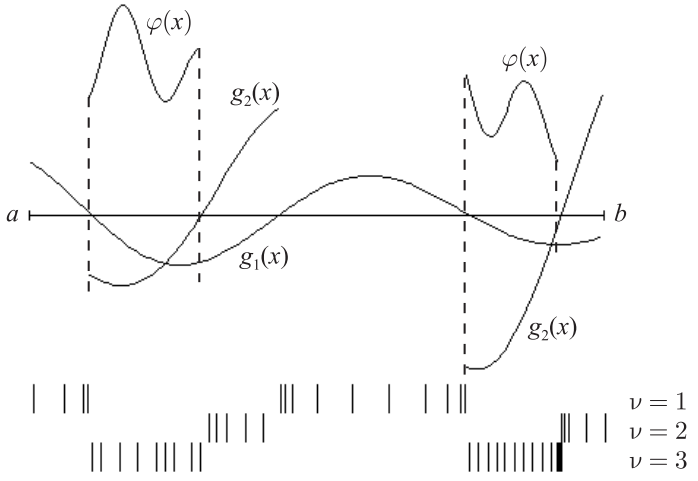

Рис. 3.6. Минимизация «индексной» функции

вертикальных штрихов. Штрихи верхнего ряда соответствуют точкам с единичным индексом, второго — точкам, индексы которых равны 2; точки, отмеченные штрихами нижнего ряда, являются допустимыми. Координаты испытаний, выполненных в близких точках, отмечены темным прямоугольником.

Следует отметить, что весьма часто в прикладных задачах оценка значений функций ограничений и целевой функции требует заметных вычислительных ресурсов. Результаты, приведенные в таблице 3.1, подтверждают экономичность используемой индексной схемы учета ограничений (здесь k_i — число вычислений i-й функции задачи).

Таблица 3.1. Результаты решения тестовой задачи

	k_1	k_2	k_3
Перебор по равномерной сетке	160000	90280	56476
Метод штрафных функций	375	375	375
Индексный метод	63	49	35

3.4. Индексный алгоритм, учитывающий существование ε-резервированных решений

Вернемся к рассмотрению одномерной задачи вида (3.1)

$$\varphi(x^*) = \min\{\varphi(x)\colon\ x \in [a,b],\ \ g_j(x) \leqslant 0,\ \ 1 \leqslant j \leqslant m\}.$$

В случае если целевая функция и левые части ограничений являются частично вычислимыми, следует использовать запись в форме (3.8).

Пример 3.2. В качестве иллюстрации рассмотрим задачу (3.1) при $x \in [-0{,}6, 2{,}2]$, $m = 3$:

$$\begin{aligned}
\varphi(x) &= \cos(18x - 3)\sin(10x - 7) + 1,\\
g_1(x) &= \exp(-x/2)\sin(6x - 1{,}5),\\
g_2(x) &= \sin(4x - 2{,}2) + \cos(6x - 2{,}9),\\
g_3(x) &= |x|\sin(2\pi x - 0{,}5).
\end{aligned}$$

Функции g_j, $1 \leqslant j \leqslant 4$, изображены кривыми на рис. 3.7. В предположении частичной вычислимости соответствующие областям Q_j, $1 \leqslant j \leqslant 3$, из (3.3) дуги этих функций представлены на рис. 3.8.

Описанный в разделе 3.3 индексный алгоритм был использован для решения этого примера при $x_1 = (a+b)/2$, $r = 2$ и $\varepsilon = 10^{-5}$. Координаты 102 точек испытаний, осуществленных алгоритмом в процессе решения задачи, отмечены на рис. 3.7 четырьмя рядами вертикальных штрихов. Штрихи верхнего ряда соответствуют точкам с единичным индексом, второго и третьего — точкам, индексы которых соответственно равны 2 и 3; точки, отмеченные штрихами нижнего ряда, являются допустимыми.

Координаты испытаний, выполненных в близких точках, отмечены темными прямоугольниками. При этом значения функций g_1, g_2, g_3, $\varphi = g_4$ вычислялись соответственно $k_1 = 102$, $k_2 = 80$, $k_3 = 64$ и $k_4 = 26$ раз. Как видно из рис. 3.7, концентрация точек испытаний имеет место не только в окрестности точки $x^* = 2{,}07957$ условного глобального оптимума, но и в окрестностях недопустимых граничных точек областей Q_j из (3.3). Этот эффект можно ослабить

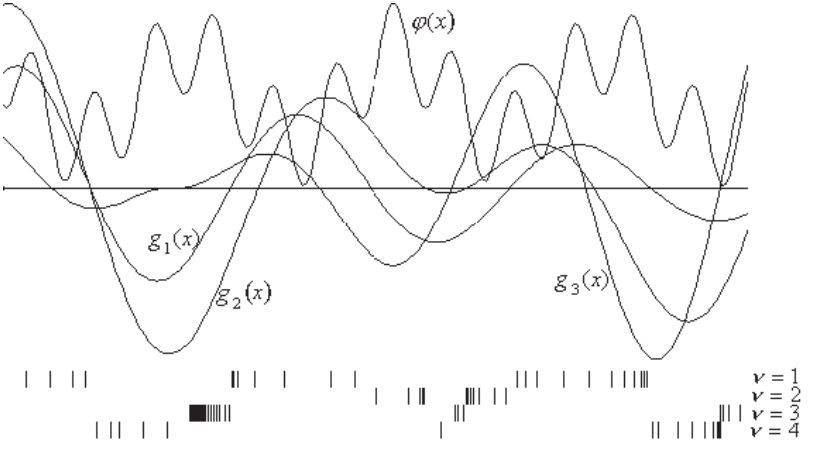

Рис. 3.7. Решение задачи индексным методом

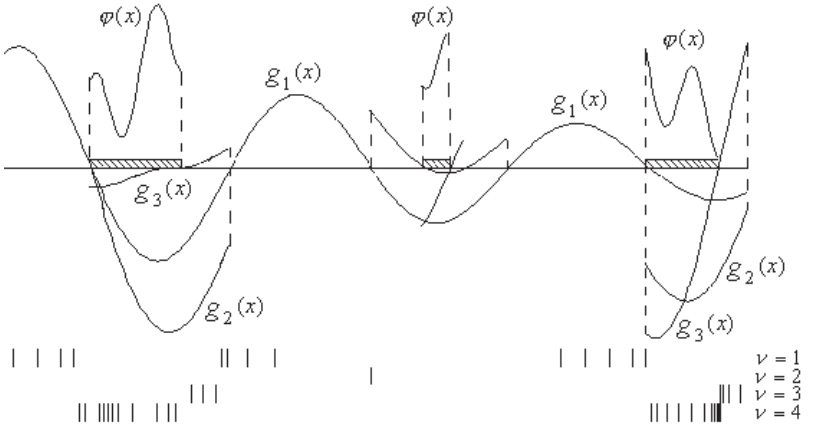

Рис. 3.8. Решение задачи индексным методом с ε-резервированием

заменой оценок (3.16) на оценки

$$z_\nu^* = \begin{cases} -\varepsilon_\nu, & \nu < M, \\ \min\{g_\nu(x_i)\colon i \in I_\nu\}, & \nu = M, \end{cases} \tag{3.21}$$

где вектор $\varepsilon_R = (\varepsilon_1, \dots, \varepsilon_m)$ является заданным вектором с неотрицательными компонентами. Таким образом, $z_\nu^* = 0$, если существуют точки x_i, $1 \leqslant i \leqslant k$, из ряда (3.12), имеющие индекс, больший ν. Вследствие этой замены значения характеристик $R(i)$ из (3.17) станут меньше (из-за вычитания ε_ν), если

$$\nu = \max\{\nu(x_{i-1}), \nu(x_i)\} < \max\{\nu(x_j)\colon 1 \leqslant j \leqslant k\} \leqslant V, \tag{3.22}$$

что сделает менее вероятным проведение испытаний в соответствующих интервалах (x_{i-1}, x_i). В остальном алгоритм остается без изменений.

Этот, на первый взгляд, механический прием имеет серьезные основания, к рассмотрению которых мы и переходим.

Определение 3.1. Точка x_ε называется *ε-резервированным решением задачи* (3.1), если выполнено условие

$$\varphi(x_\varepsilon) = \min\{\varphi(x)\colon x \in [a, b],\ g_j(x) \leqslant -\varepsilon_j,\ 1 \leqslant j \leqslant m\}, \tag{3.23}$$

где $\varepsilon_1, \dots, \varepsilon_m$ — положительные числа («резервы» по каждому ограничению). Введем в рассмотрение также множество

$$X_\varepsilon = \{x \in [a, b]\colon g_j(x) \leqslant 0,\ 1 \leqslant j \leqslant m,\ \varphi(x) \leqslant \varphi(x_\varepsilon)\} \tag{3.24}$$

всех допустимых решений задачи, которые по значению целевой функции «не хуже», чем ε-резервированное решение.

Существование ε-резервированного решения задачи (3.1) можно интерпретировать как некоторый аналог условий регулярности в классических задачах нелинейного программирования. Следует также отметить прикладную роль этого условия. Даже если точное решение x^* известно, его практическая реализация возможна лишь как некоторое приближение к x^*. Поэтому важно, чтобы существовали близкие к x^* (по координатам и значению) допустимые точки из Q. Существование ε-резервированного решения гарантирует

наличие таких точек. При этом область X_ε может играть роль множества подходящих приближений.

Индексный алгоритм с ε-резервированием был применен к рассматриваемому примеру при $\varepsilon_i = 0{,}2$, $1 \leqslant i \leqslant 3$. На рис. 3.8 вертикальными штрихами отмечены координаты поисковых испытаний. Решение потребовало лишь 52 испытания и обеспечило ту же самую точность решения задачи, что и при нулевом векторе ε_R. При этом значения функций вычислялись $k_1 = 52$, $k_2 = 39$, $k_3 = 38$ и $k_4 = 25$ раз.

Условия сходимости алгоритма представлены в виде следующей теоремы (приведена без доказательства — см. в работе [46]).

Теорема 3.2. *Пусть задача* (3.1) *имеет решение* x^* *и выполняются следующие условия*:

1) *каждая область* Q_j, $1 \leqslant j \leqslant m+1$, *из* (3.3) *представляет собой объединение конечного числа отрезков положительной длины*;

2) *каждая функция* g_j, $1 \leqslant j \leqslant m+1$, *допускает липшицево с константой* L_j *продолжение* $G_j(x)$ *на весь отрезок* $[a, b]$, *т. е.*

$$g_j(x) = G_j(x), \quad x \in Q_j, \quad 1 \leqslant j \leqslant m+1; \tag{3.25}$$

3) *компоненты* ε_ν *вектора резервов* ε_R, *соответствующие активным в точке абсолютного минимума* x^* *ограничениям (т. е. ограничениям, для которых выполняется условие* $g_\nu(x^*) = 0$*), удовлетворяют неравенствам*

$$0 < 2\varepsilon_\nu < L_\nu(\beta - \alpha), \tag{3.26}$$

где $\beta - \alpha$ — *длина интервала* $[\alpha, \beta] \subset Q_{m+1}$, *содержащего точку* x^*;

4) *компоненты* ε_ν *вектора резервов* ε_R, *соответствующие неактивным в точке абсолютного минимума* x^* *ограничениям (т. е. ограничениям, для которых выполняется условие* $g_\nu(x^*) < 0$*), удовлетворяют либо неравенствам (3.26), либо неравенствам*

$$0 < \varepsilon_\nu < |g_\nu(x^*)|; \tag{3.27}$$

5) *если некоторым точкам из ряда (3.12) соответствуют значения индекса, равные* ν, *и при достаточно больших значениях* k *для соответствующей величины* μ_ν *из (3.15) справедливо неравенство*

$$r_\nu \mu_\nu > 2L_\nu, \quad 1 \leqslant \nu \leqslant m+1,$$

то:

1) *точка* x^* *является предельной точкой последовательности точек испытаний* $\{x^k\}$, *порождаемой индексным алгоритмом для задачи* (3.1), *при* $\varepsilon = 0$ *в условии остановки* (3.20);

2) *любая предельная точка* $\overline{x}$ *последовательности* $\{x^k\}$ *является решением задачи* (3.1);

3) *сходимость к предельной точке* $\overline{x}$ *является двусторонней, если* $\overline{x} \neq a$ *и* $\overline{x} \neq b$.

Если условия (3.26) *и* (3.27) *не выполняются для заданного вектора резервов* ε_R, *но задача* (3.1) *имеет* ε*-резервированное решение, определяемое отношением* (3.23), *то*:

4) *для любой предельной точки* $\overline{x}$ *последовательности* $\{x^k\}$ *справедливы отношения*

$$\varphi(\overline{x}) = \inf\{\varphi(x^k)\colon\ g_j(x^k) \leqslant 0,\ 1 \leqslant j \leqslant m,\ k = 1, 2, \ldots\} \leqslant \varphi(x_\varepsilon);$$

5) *любая предельная точка* $\overline{x}$ *есть граничная точка множества* X_ε *из* (3.24), *если функция* $\varphi(x)$ *имеет не более одной точки локального максимума в каждом из изолированных интервалов, составляющих множество* X_ε;

6) *приведенное выше утверждение о двустороннем характере сходимости остается в силе.*

Как показал эксперимент с решением описанного выше примера, увеличение значений резервов может существенно снизить концентрацию испытаний в недопустимых областях граничных точек. Теоретические свойства последовательности испытаний, порождаемой индексным алгоритмом, определяются следующим утверждением (доказательство также см. в работе [46]).

Теорема 3.3. *Пусть выполнены условия сходимости индексного алгоритма и пусть в подынтервале* $[\alpha, \beta] \subset Q_\nu$, $1 \leqslant \nu \leqslant m$, *длина*

которого удовлетворяет условию

$$\beta - \alpha > \frac{2(r_\nu - 1)(\delta_\nu + \varepsilon_\nu)}{r_\nu^2 L_\nu},$$

выполняются неравенства

$$g_\nu(x) \geqslant \delta_\nu, \quad x \in [\alpha, \beta], \quad \delta_\nu \geqslant 0.$$

Тогда для плотности $P_{\alpha\beta}$ точек последовательности $\{x^k\}$, порождаемой индексным алгоритмом при решении задачи (3.1), *справедлива оценка*

$$P_{\alpha\beta} < \frac{r_\nu^2 L_\nu}{(r_\nu - 1)(\delta_\nu + \varepsilon_\nu)}.$$

Выбор компонент вектора резервов. При использовании рассматриваемого алгоритма возникает вопрос о конкретных значениях компонент вектора резервов ε_R. В этом нам помогает теорема о сходимости: условие (3.26) можно рассматривать как рекомендацию для выбора значений резервов, при которых сохраняется свойство сходимости к решению задачи (3.1). Однако, в общем случае значения констант Липшица не являются априорно заданными, и длина интервала $[\alpha, \beta]$ из Q_{m+1}, содержащего искомое решение x^*, также неизвестна. Эти трудности можно обойти, заменив неизвестные коэффициенты L_j, $1 \leqslant j \leqslant m$, их адаптивными нижними оценками μ_ν из (3.15) и приняв, что $\beta - \alpha > \delta = \varepsilon q$, где $\varepsilon > 0$ есть параметр (точность), задаваемый в условии остановки, а фактор $q > 1$ можно интерпретировать как априорную оценку того, во сколько раз длина интервала $[\alpha, \beta]$ превышает заданную точность ε. В результате получаем оценки резервов вида

$$\varepsilon_\nu = \mu_\nu \delta, \quad 1 \leqslant \nu \leqslant m.$$

Результаты экспериментов. Для демонстрации эффективности адаптивного резервирования проведем решение серии задач с многоэкстремальными ограничениями.

Рассмотрим генератор G, порождающий задачи с m ограничениями на основе некоторого случайного механизма F, дающего функции $f(x)$, $x \in [a, b]$. Правила, определяющие работу такого механизма G, состоят в следующем.

1. С помощью заданного механизма F независимо генерируются $m+1$ функция $f_j(x)$, $x \in [a,b]$, $1 \leqslant j \leqslant m+1$.

2. Вычисляется минимакс

$$\Delta = \min_{a \leqslant x \leqslant b} \max_{1 \leqslant j \leqslant m} f_j(x),$$

и вводятся левые части ограничений

$$g_j(x) = f_j(x) - \Delta - \delta, \quad 1 \leqslant i \leqslant m, \tag{3.28}$$

которым при $\delta > 0$ соответствует (по построению) непустая допустимая область Q. При этом увеличение значения параметра δ позволяет расширять допустимую область.

3. Функции (3.28) и целевая функция $\varphi(x) = f_{m+1}(x)$ задают задачу вида (3.1), заведомо имеющую решение.

В случаях, когда необходимо указать используемые при запуске генератора параметры m и δ, будем использовать обозначение $G(m,\delta)$. В численных экспериментах, результаты которых описываются ниже, использовалось два генератора. Один из них (обозначим его символом $G_{\mathrm{SH}}(5{,}1)$) основан на описанном выше случайном механизме (2.43). Второй генератор, обозначаемый в дальнейшем как $G_{\mathrm{HL}}(5,1)$, использует случайный механизм (2.44). Каждый из этих двух генераторов был применен для порождения своей выборки из 100 задач вида (3.1).

Задачи решались с помощью исходного индексного алгоритма и алгоритма с ε-резервированием. При этом использовались параметры r_ν, $1 \leqslant \nu \leqslant 6$, из (3.17), определяемые выражениями

$$r_\nu = \begin{cases} \gamma, & k_\nu < 20, \\ 2, & k_\nu > 20, \end{cases}$$

где k_ν — число вычисленных значений функции g_ν (т. е. мощность множества I_ν). В ходе экспериментов на выборке, рожденной генератором G_{SH}, задавались значения $\gamma = 20$, $\varepsilon = 0{,}001$ и резервы $\delta_\nu = 0{,}02$, $1 \leqslant \nu \leqslant 5$. Для задач из второй выборки (генератор G_{HL}) использовались значения $\gamma = 10$, $\varepsilon = 0{,}0001$ и параметры $\delta_\nu = 0{,}001$, $1 \leqslant \nu \leqslant 5$.

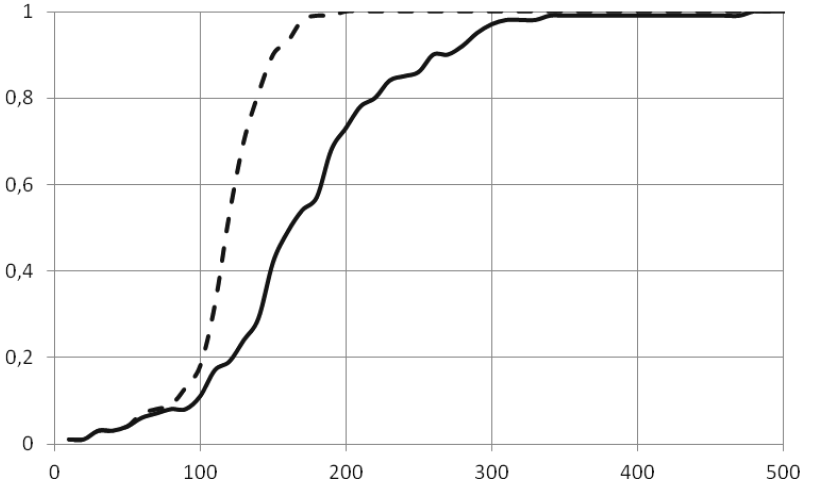

Рис. 3.9. Операционные характеристики на классе G_{SH}

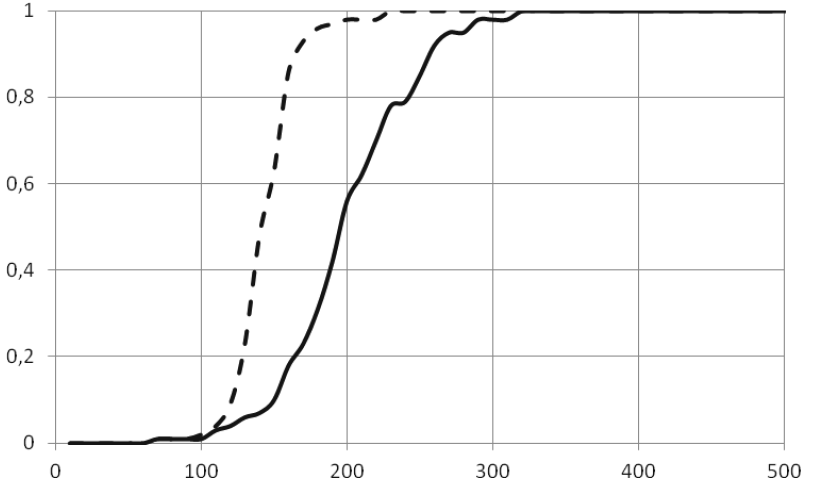

Рис. 3.10. Операционные характеристики на классе G_{HL}

Операционные характеристики для обоих методов, полученные на выборках, рожденных генераторами G_{SH} и G_{HL}, представлены соответственно на рисунках 3.9 и 3.10. При этом нижние (непрерывные) кривые характеризуют исходный индексный алгоритм, а верхние (разрывные) линии — индексный алгоритм с адаптивным резервированием. Указанное расположение кривых показывает, что алгоритм с адаптивным резервированием обеспечивает значительно более быстрое получение оценок, лежащих в заданной окрестности решения, чем исходный индексный алгоритм. При этом среднее число итераций для метода без резервирования и метода с адаптивным резервированием составило соответственно 171 и 117 для задач G_{SH} и 199 и 142 для задач G_{HL}.

3.5. Индексный метод с адаптивным порядком проверки ограничений

В соответствии с правилами индексного метода, рассмотренного в предыдущих разделах, каждое испытание включает последовательную проверку выполнимости ограничений задачи в соответствующей точке области поиска; обнаружение первого нарушенного ограничения прерывает испытание и инициирует переход к точке следующей итерации. Если при этом ввести нумерацию ограничений в порядке возрастания вычислительных затрат на их проверку (т. е. определить, что сначала проверяются более простые ограничения), то это может существенно снизить трудоемкость решения задачи в целом. Сказанное актуально для задач, в которых оценка значений функций требует заметных вычислительных ресурсов, связанных с необходимостью численного моделирования поведения систем, характеризуемых этими функциями (как это имеет место, например, для многих задач оптимального проектирования).

Очевидно, что вычислительные затраты на проведение испытания будут существенным образом зависеть от порядка, в котором вычисляются функции задачи. Если предположить, что время вычисления любой функции задачи не зависит от точки проведения

итерации, то затраты на проведение испытания в точке x будут тем меньше, чем меньше значений будет вычислено в этой точке.

Для многих прикладных задач естественно допустить, что ограничение, нарушенное в некоторой точке области поиска, не будет выполняться и в некоторой окрестности этой точки. В таком случае может оказаться полезным начинать испытания в точках этой окрестности с анализа выполнимости указанного ограничения. В свете сказанного представляется целесообразным построить алгоритм глобальной оптимизации, допускающий изменение порядка проверки ограничений с учетом информации, получаемой в ходе поиска решения. Формирование порядка проверки будет ориентировано на то, чтобы нарушенное в точке итерации ограничение было выявлено по возможности раньше.

Перейдем теперь к обобщению описанной в разделах 3.3 и 3.4 схемы на случай, когда порядок проверки ограничений не является фиксированным. При этом будем ссылаться на рассмотренный исходный алгоритм как на *метод с фиксированным порядком проверок* (МФПП).

Алгоритм, предлагаемый ниже, допускает изменение порядка проверки ограничений в процессе решения задачи (3.1). Указанные изменения направлены на то, чтобы при очередном испытании сначала анализировалось ограничение, нарушенное на конце интервала, содержащего точку очередной итерации. Как уже отмечалось выше, такой подход может уменьшить число проверок в новой точке, поскольку невыполнение указанного ограничения в некоторой окрестности уже обнаруженного нарушения может быть весьма вероятным. При этом возникает потребность как в новых правилах сопоставления индексов точкам из (3.12) (напомним, что эти индексы определяют использование выражений (3.17) для подсчета характеристик), так и в специальной процедуре, формирующей порядок проверки ограничений в точке очередного испытания. Заметим, что формирование такого порядка несущественно для интервалов, концы которых принадлежат допустимому множеству Q из (3.30), ибо в таких интервалах вероятна необходимость проверки всех ограничений.

Пусть ограничение, имеющее номер j, $1 \leqslant j \leqslant m$, выполняется во всех точках области

$$Q_j = \{x \in [a, b]\colon g_j(x) \leqslant 0\}, \quad 1 \leqslant j \leqslant m, \tag{3.29}$$

которую будем называть допустимой областью для этого ограничения. При этом допустимая область Q исходной задачи (3.1) может определяться условием

$$Q = Q_1 \cap \ldots \cap Q_m. \tag{3.30}$$

Порядок проверки ограничений. Нумерацию ограничений и целевой функции, введенную в формулировке задачи (3.1) и определенную последовательностью целых чисел

$$H = \{1, 2, \ldots, m, m{+}1\}, \tag{3.31}$$

будем рассматривать как базовую.

Примем, что каждой конкретной точке x из области поиска $[a, b]$ может быть сопоставлен свой порядок проверки ограничений задачи. При этом порядок такой проверки, соответствующий точке x_i из (3.12), будем задавать как некоторую перестановку $H(x_i)$ базовых номеров из (3.31), т. е.

$$H(x_i) = \{j_{i1}, \ldots, j_{im}, j_{i,m+1}\}, \quad 0 \leqslant i \leqslant k, \tag{3.32}$$

где

$$j_{i,m+1} = m + 1, \quad 0 \leqslant i \leqslant k. \tag{3.33}$$

Поскольку порядок проверки ограничений при проведении испытаний в точке x_i определен соответствующим ей кортежем $H(x_i)$ из (3.32), то прекращение испытания в этой точке после вычисления в ней значений l функций задачи дает величину $z_i = g_\nu(x_i)$, где $\nu = j_{il}$. В связи с этим в качестве значения $\nu(x_i)$ для точки x_i условимся использовать не порядковый номер l последней функции, вычисленной в этой точке, как это было принято в (3.7), а индекс последней функции в базовой нумерации (3.31), т. е.

$$\nu(x_i) = j_{il}. \tag{3.34}$$

Алгоритм. Основу схемы нового алгоритма, который будем называть *методом с адаптивным порядком проверок* (МАПП), со-

ставляют уже рассмотренные правила выбора испытаний в методе с фиксированным порядком проверок (МФПП).

Поскольку переменный порядок проверки ограничений исключает возможность использования выражения (3.13) для определения множеств I_ν, то для их построения необходимо располагать информационным массивом, связывающим с каждым узлом x_i из (3.12) номера всех функций, вычислявшихся в этом узле. В дальнейшем предполагается существование такой базы данных. Прочие модификации состоят в следующем.

Модификация 1 (*фиксация порядков проверки*). Каждой точке x_i из (3.12) сопоставляется соответствующий ей порядок проверки $H(x_i)$ из (3.32), и этот порядок входит в информационную базу алгоритма.

Модификация 2 (*определение индексов точек испытаний*). Каждой точке x_i из (3.12) сопоставляется значение $z_i = g_\nu(x_i)$, где, согласно (3.34), индекс $\nu = \nu(x_i)$ — базовый номер из (3.31), соответствующий значению последней функции, оцененному в этой точке. Указанные номера используются во всех вычислениях, основанных на выражениях (3.13)–(3.17), (3.19).

Изменения способа использования индексов относятся лишь к правилам (3.17) подсчета характеристик $R(i)$ интервалов (x_{i-1}, x_i), $1 \leqslant i \leqslant k$, имеющих несовпадающие индексы концевых точек. Пусть

$$\nu(x_{i-1}) = j_{i-1,p} = j_{i,q} \neq \nu(x_i) = j_{i-1,u} = j_{i,s}. \qquad (3.35)$$

То есть функция с базовым номером $\nu(x_{i-1})$ имеет номер p в последовательности $H(x_{i-1})$, соответствующей точке x_{i-1}, и номер q в последовательности $H(x_i)$, соответствующей точке x_i. Аналогично функции с базовым номером $\nu(x_i)$ соответствует номер u из последовательности $H(x_{i-1})$ и номер s из последовательности $H(x_i)$. При этом вводимая модификация, изменяющая пятое правило из схемы МФПП, состоит в следующем.

Если для номеров p,q и u,s из (3.35) выполняются условия

$$p < u, \quad q < s, \qquad (3.36)$$

то для определения характеристики $R(i)$ используется второе выражение из формул (3.17) при $\nu = \nu(x_i)$ (заметим, что в этом случае значение $g_\nu(x_{i-1})$ не определено).

При выполнении обратных (3.36) условий

$$p > u, \quad q > s \tag{3.37}$$

характеристика $R(i)$ определяется третьим выражением из (3.17) при $\nu = \nu(x_{i-1})$ (в этом случае неопределенным является значение $g_\nu(x_i)$). В случаях когда условия (3.36) и (3.37) не выполняются, сохраняется исходный порядок вычислений, предписываемый пятым правилом исходного алгоритма.

Модификация 3 (*выбор порядка проверки ограничений в точке очередного испытания*). Примем, что порядок проверки ограничений, соответствующий точкам двух первых испытаний, определяется условиями

$$H(a) = H(b) = H, \tag{3.38}$$

где H из (3.31). Пусть

$$p = \nu(x_{t-1}), \quad q = \nu(x_t)$$

есть определенные согласно (3.34) индексы концевых точек интервала (x_{t-1}, x_t), содержащего в соответствии с седьмым правилом точку x^{k+1} очередного испытания. Введем обозначение $\{s, H\backslash s\}$ для кортежа, получаемого из H путем исключения номера s, сдвига всех индексов $1, 2, \ldots, s-1$ на одну позицию вправо и перенесения исключенного из H номера s в первую позицию, т. е.

$$\{s, H\backslash s\} = \{s, 1, 2, \ldots, s-1, s+1, \ldots, m, m+1\}.$$

При $1 \leqslant p < q \leqslant m$ суперпозиция операций описанного типа дает кортеж вида

$$\{p, \{q, H\backslash q\}\backslash p\} = \{p, q, 1, \ldots, p-1, p+1, \ldots, q-1, q+1, \ldots, m+1\}.$$

Заметим, что согласно (3.33), номер $m+1$ всегда занимает последнюю позицию.

Теперь условимся определять порядок $H(x^{k+1})$, $k \geqslant 1$, в соответствии со следующими правилами:

$$
\begin{aligned}
&H(x^{k+1}) = H, \quad p = q = m+1,\\
&H(x^{k+1}) = \{p, H\backslash p\}, \quad (p = q < m+1) \cup (p < q = m+1),\\
&H(x^{k+1}) = \{q, H\backslash q\}, \quad p = m+1 > q,\\
&H(x^{k+1}) = \{p, \{q, H\backslash q\}\backslash p\}, \quad p < q \leqslant m,\\
&H(x^{k+1}) = \{q, \{p, H\backslash p\}\backslash q\}, \quad m \geqslant p > q.
\end{aligned}
\tag{3.39}
$$

Замечание. При решении задачи (3.1) правила (3.38), (3.39), входящие в схему МАПП, однозначно сопоставляют всем точкам испытаний значения индекса из (3.34). При этом любой допустимой точке x всегда соответствует индекс $\nu(x) = m+1$. Это значение может быть предсказано и до проведения испытания в такой точке. Однако если точка x не является допустимой и не принадлежит к числу точек, в которых уже осуществлены испытания, то значение индекса в этой точке может быть не определено.

Условия сходимости алгоритма могут быть сформулированы в виде следующей теоремы.

Теорема 3.4. *Пусть задача* (3.1) *имеет решение* x^* *и выполняются следующие условия*:

1) *каждая область* Q_j, $1 \leqslant j \leqslant m$ *из* (3.29) *представляет собой объединение конечного числа отрезков, имеющих положительную длину*;

2) *каждая функция* $g_j(x)$, $1 \leqslant j \leqslant m+1$, *удовлетворяет условию Липшица с соответствующей константой* L_j;

3) *компоненты* ε_ν *вектора резервов* ε_R *удовлетворяют неравенствам*

$$0 < 2\varepsilon_\nu < L_\nu(\beta - \alpha), \tag{3.40}$$

где $\beta - \alpha$ *есть длина интервала* $[\alpha, \beta]$, *принадлежащего допустимой области* Q *из* (3.30) *и содержащего точку* x^* — *решение задачи*;

4) *при достаточно больших значениях* k *величины* μ_ν *из* (3.15), *соответствующие непустым множествам* I_ν,

удовлетворяют неравенствам

$$r_\nu \mu_\nu > 2L_\nu, \quad 1 \leqslant \nu \leqslant m+1,$$

где $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$ — *параметры алгоритма.*

Тогда:

1) *точка* x^* *является предельной точкой последовательности* $\{x^k\}$, *порождаемой описанным методом с адаптивным порядком проверки ограничений для задачи* (3.1) *при* ε=0 *в условии остановки* (3.20);

2) *любая предельная точка* $\overline{x}$ *последовательности* $\{x^k\}$ *является решением задачи* (3.1);

3) *сходимость к предельной точке* $\overline{x}$ *является двусторонней, если* $\overline{x} \neq a$ *и* $\overline{x} \neq b$.

Доказательство теоремы приведено в работе [2].

Результаты экспериментов. В качестве иллюстрации рассмотрим решение задачи из примера 3.1 с помощью МАПП с параметрами $r_\nu = 2$, $\varepsilon_\nu = 0$, $1 \leqslant \nu \leqslant 3$, и $\varepsilon = 10^{-5}$ в условии остановки. Координаты точек испытаний, осуществленных алгоритмом в процессе решения задачи, практически совпадают с теми, что отмечены на рис. 3.6 тремя рядами вертикальных штрихов. Однако число вычислений значений ограничений при использовании МФПП и МАПП оказывается различным. При фиксированном порядке проверки значения функций вычислялись $k_1 = 63$, $k_2 = 49$, $k_3 = 35$ раз, а при адаптивном порядке — $k_1 = 52$, $k_2 = 48$, $k_3 = 35$ раз. Данные результаты свидетельствует о снижении числа таких проверок при использовании МАПП даже для примера с двумя ограничениями. Более наглядно эффективность предложенной схемы проявляет себя при решении задач со многими ограничениями.

С этой целью мы вернемся к решению серии задач, порожденных генераторами $G_{\mathrm{SH}}(5,1)$ (основан на случайном механизме (2.43)) и $G_{\mathrm{HL}}(5,1)$ (использует случайный механизм (2.44)). Каждый из этих двух генераторов был применен для порождения своей выборки из 1000 задач вида (3.1).

На рис. 3.11 отражено распределение задач из первой выборки по длинам соответствующих им отрезков $[\alpha, \beta]$ из (3.40), содержа-

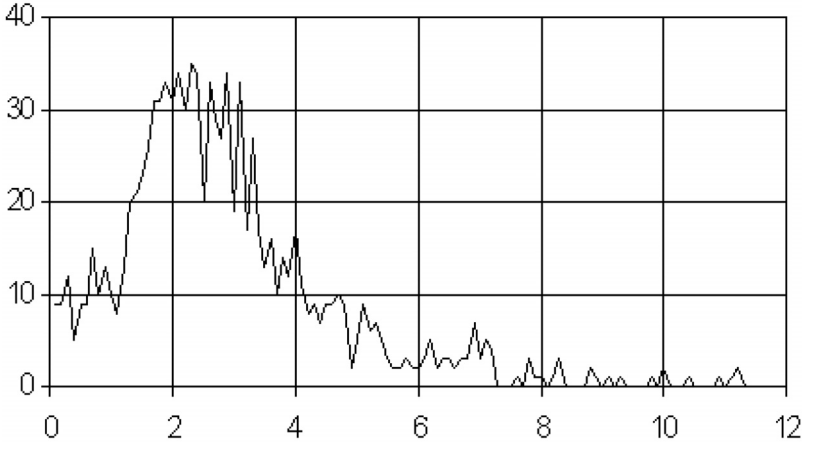

Рис. 3.11. Распределение задач класса G_{SH}

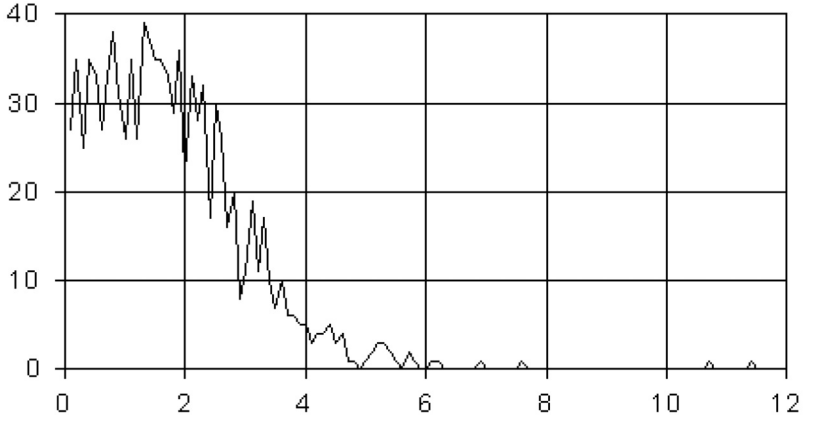

Рис. 3.12. Распределение задач класса G_{HL}

щих точку решения x^*. Рисунок 3.12 демонстрирует аналогичное распределение для задач из второй выборки. При этом по осям абсцисс отложены (в %) значения отношения $(\beta - \alpha)/(b - a)$, а по

осям ординат — числа задач из соответствующих выборок. Для получения обсуждаемых оценок длина отрезка $[\alpha, \beta]$, соответствующая каждой функции, определялась численными методами.

Более детальная информация о числе задач (в каждой из двух выборок), характеризуемых малой (менее 0,1%) мерой отрезка $[\alpha, \beta]$, дана ниже.

Таблица 3.2. Число задач с малой мерой отрезка $[\alpha, \beta]$

%	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
G_{SH}	0	1	2	1	0	0	1	1	1
G_{HL}	3	6	2	3	5	1	2	1	2

Как следует из рисунков и таблицы, задачи из второй выборки, рожденные генератором G_{HL}, характеризуются в целом более узкой допустимой окрестностью точки решения x^*.

В качестве примера на рис. 3.13 приведены графики функций ограничений $g_\nu(x)$, $1 \leqslant \nu \leqslant 5$ и целевой функции $\varphi(x)$ для задачи из класса G_{HL}. На оси абсцисс выделена допустимая область, которую формирует каждое ограничение; для целевой функции показана допустимая область задачи в целом (пересечение всех допустимых областей ограничений).

Эксперименты проводились для описанных выше алгоритмов с фиксированным (МФПП) и адаптивным (МАПП) порядками проверки ограничений. При этом использовались параметры r_ν, $1 \leqslant \nu \leqslant 5$, из (3.17), определяемые выражениями

$$r_\nu = \begin{cases} \gamma, & k_\nu \leqslant 20, \\ 2, & k_\nu > 20, \end{cases}$$

где k_ν — число вычисленных значений функции g_ν (т. е. мощность множества I_ν). В ходе экспериментов на выборке, рожденной генератором G_{SH}, задавались значения $\gamma = 20$, $\varepsilon = 0{,}001$ и резервы $\varepsilon_\nu = 0{,}1$, $1 \leqslant \nu \leqslant 5$. Для задач из второй выборки (генератор G_{HL}) использовались значения $\gamma = 10$, $\varepsilon = 0{,}0001$ и резервы $\varepsilon_\nu = 0{,}001$.

Операционные характеристики для обоих методов, полученные на выборках, рожденных генераторами G_{SH} и G_{HL}, представлены

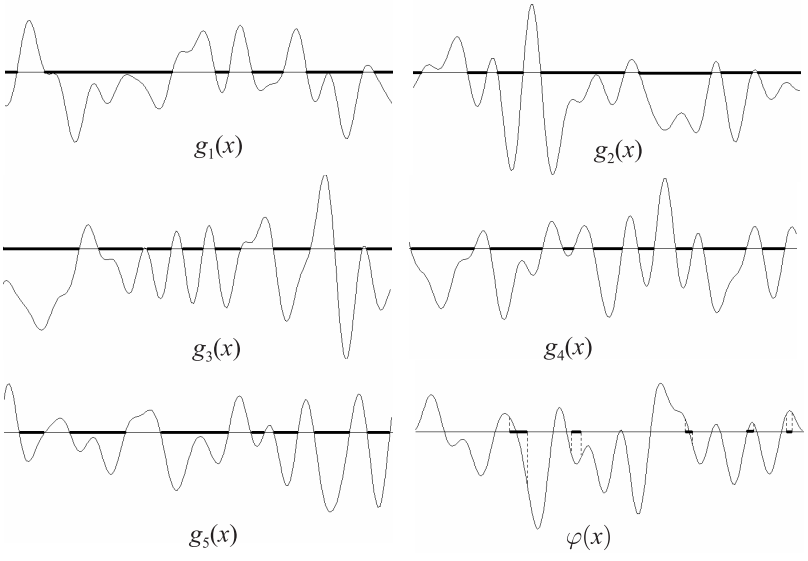

Рис. 3.13. Графики функций, составляющих тестовую задачу из класса G_{HL}

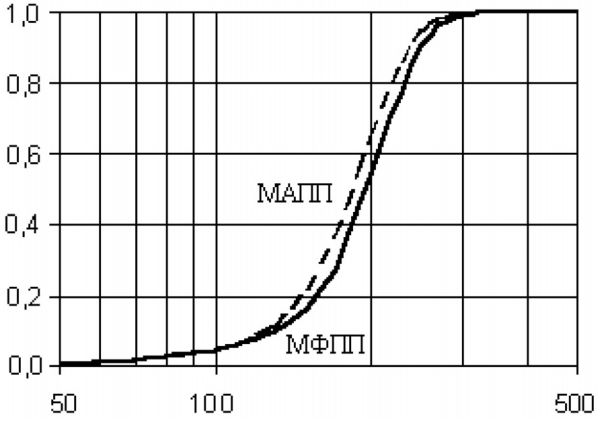

Рис. 3.14.

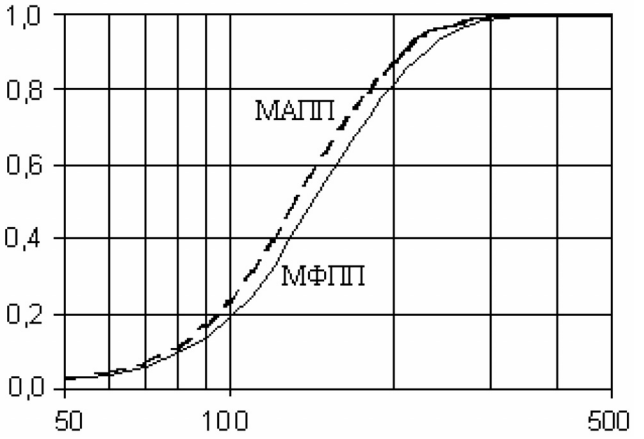

Рис. 3.15.

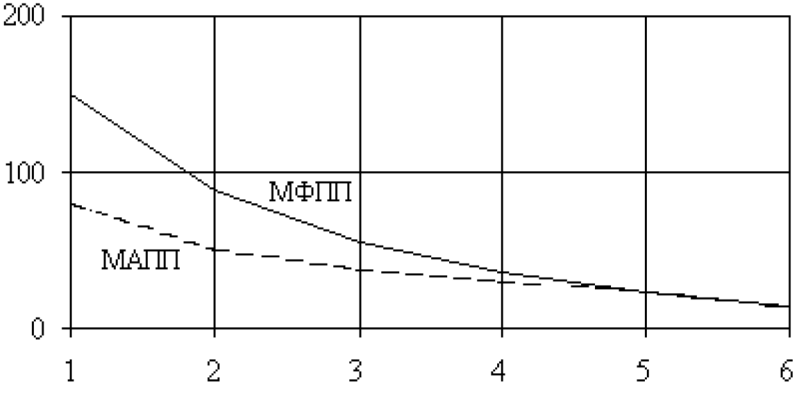

Рис. 3.16.

соответственно на рис. 3.14 и 3.15. При этом нижние (непрерывные) кривые характеризуют МФПП, а верхние (разрывные) линии – МАПП. Указанное расположение кривых показывает, что алгоритм с адаптивным порядком проверок обеспечивает в среднем

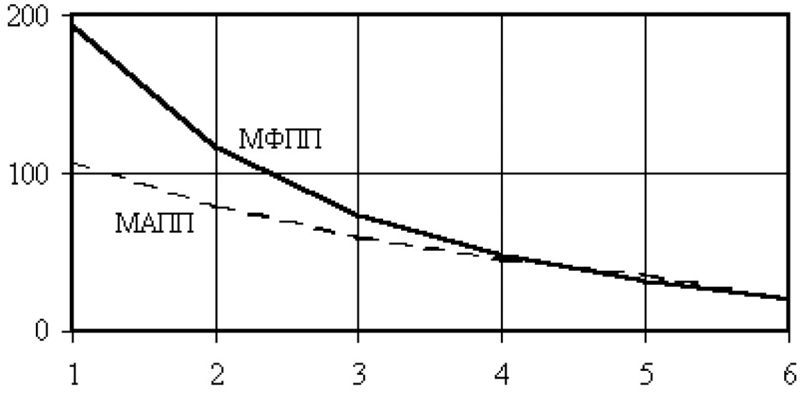

Рис. 3.17.

более быстрое получение оценок, лежащих в заданной окрестности решения, чем алгоритм с фиксированным порядком проверок.

Более того, испытания, осуществляемые в соответствии с правилами МАПП, оказываются (в среднем) существенно менее трудоемкими, чем испытания, реализуемые МФПП. Это обстоятельство иллюстрируется на рис. 3.16 и 3.17. Здесь представлено среднее число вычислений значений каждой функции, осуществленных МФПП и МАПП на задачах выборки. Рисунок 3.16 соответствует генератору G_{SH}, а рисунок 3.17 — генератору G_{HL}. При этом номера функций отложены на осях абсцисс. Оценки среднего числа вычислений значений этих функций, осуществленных в ходе испытаний, нанесены на осях ординат. Сплошные линии на фигурах соединяют точки оценок, соответствующих МФПП, а разрывные линии — точки оценок, соответствующих МАПП. Представленные данные подтверждают, что введение адаптивного порядка проверки ограничений может существенно сократить общее число таких вычислений.

3.6. Параллельные вычисления для многоэкстремальных задач оптимизации с нелинейными ограничениями

3.6.1. Асинхронный параллельный индексный алгоритм

В этом разделе будет изложен асинхронный параллельный индексный алгоритм для решения задачи (3.1) с использованием $p > 1$ параллельных процессоров.

Первые p точек испытаний выбираются в произвольных внутренних точках отрезка $[a, b]$, и каждый процессор начинает проводить испытания в одной из этих точек. После завершения k-го испытания, когда точки $x^{k+1}, \ldots, x^{k+p-1}$ уже выбраны, но результаты испытаний в них еще не получены, точка x^{k+p} очередного испытания выбирается в соответствии со следующими правилами.

Правило 1. Перенумеровать точки $x^1, \ldots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_i < \ldots < x_{k+p-1} < x_{k+p} = b, \qquad (3.41)$$

точки $x_0 = a$ и $x_{k+p} = b$ введены в ряд (3.41) дополнительно для удобства последующих обозначений.

Правило 2. Определить множество I точек x_i, $1 \leqslant i < k + p$, таких что в них еще не завершен процесс проведения испытания.

Правило 3. Всем точкам x_i из ряда (3.41) таким, что $x_i \notin I$, сопоставить значения $z_i = g_\nu(x_i)$, $\nu = \nu(x_i)$, $1 \leqslant i \leqslant k$, вычисленные в этих точках (значения z_0 и z_{k+p} не определяются).

Правило 4. Провести классификацию номеров i, $1 \leqslant i < k + p$, точек из ряда (3.41) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$I_\nu = \{i\colon 1 \leqslant i \leqslant k + p,\ \nu = \nu(x_i),\ x_i \notin I\}, \quad 1 \leqslant \nu \leqslant m + 1, \qquad (3.42)$$

содержащих номера всех точек x_i, $1 \leqslant i \leqslant k$, в которых на данный момент завершены испытания, и имеющих индексы, равные одному и тому же значению ν. Граничные точки $x_0 = a$ и $x_{k+1} = b$ интерпретируются как имеющие нулевые индексы, и им сопоставляется дополнительное множество $I_0 = \{0, k+1\}$.

Определить максимальное значение индекса

$$M = \max\{\nu = \nu(x_i),\ x_i \notin I,\ 1 \leqslant i \leqslant k\}. \tag{3.43}$$

Правило 5. Вычислить текущие нижние границы

$$\mu_\nu = \max\left\{\frac{|z_i - z_j|}{x_i - x_j},\ i, j \in I_\nu,\ i > j\right\} \tag{3.44}$$

для неизвестных констант Липшица L_ν функций g_ν, $1 \leqslant \nu \leqslant m+1$. Если множество I_ν содержит менее двух элементов или если μ_ν из (3.44) оказывается равным нулю, то принять $\mu_\nu = 1$.

Правило 6. Для всех непустых множеств I_ν, $1 \leqslant \nu \leqslant m+1$, вычислить оценки

$$z_\nu^* = \begin{cases} -\varepsilon_\nu, & \nu < M, \\ \min\{g_\nu(x_i)\colon i \in I_\nu\}, & \nu = M, \end{cases} \tag{3.45}$$

где вектор $\varepsilon_R = (\varepsilon_1, \ldots, \varepsilon_m)$ является заданным вектором с неотрицательными компонентами (*вектор резервов*).

Правило 7. Для каждого интервала (x_{i-1}, x_i), $x_{i-1} \notin I$, $x_i \notin I$, $1 \leqslant i \leqslant k+p$, вычислить *характеристику* $R(i)$, где

$$\begin{aligned} R(i) &= \Delta_i + \frac{(z_i - z_{i-1})^2}{r_\nu^2 \mu_\nu^2 \Delta_i} - 2\frac{(z_i + z_{i-1} - 2z_\nu^*)}{r_\nu \mu_\nu}, \\ &\nu = \nu(x_{i-1}) = \nu(x_i), \\ R(i) &= 2\Delta_i - 4\frac{(z_i - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_i) > \nu(x_{i-1}), \\ R(i) &= 2\Delta_i - 4\frac{(z_{i-1} - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_{i-1}) > \nu(x_i), \end{aligned} \tag{3.46}$$

где величины $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$, являются параметрами алгоритма, а $\Delta_i = x_i - x_{i-1}$.

Правило 8. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i)\colon x_{i-1} \notin I,\ x_i \notin I,\ 1 \leqslant i \leqslant k+p\}. \tag{3.47}$$

Правило 9. Провести очередное испытание в серединной точке интервала (x_{t-1}, x_t), если индексы его концевых точек не совпа-

дают, т. е.

$$x^{k+p} = \frac{x_t + x_{t-1}}{2}, \quad \nu(x_{t-1}) \neq \nu(x_t). \tag{3.48}$$

В противном случае провести испытание в точке

$$x^{k+p} = \frac{x_t + x_{t-1}}{2} - \frac{z_t - z_{t-1}}{2r_\nu \mu_\nu}, \quad \nu = \nu(x_{t-1}) = \nu(x_t). \tag{3.49}$$

Описанные правила можно дополнить условием остановки, прекращающим испытания, если

$$x_t - x_{t-1} \leqslant \varepsilon, \tag{3.50}$$

где t из (3.47) и $\varepsilon > 0$ есть заданная точность.

3.6.2. Условия сходимости алгоритма

Теорема 3.5. *Пусть задача* (3.1) *имеет решение* x^* *и выполняются следующие условия:*

1) *каждая область* Q_j, $1 \leqslant j \leqslant m$, *из (3.29) представляет собой объединение конечного числа отрезков положительной длины*;

2) *каждая функция* $g_j(x)$, $1 \leqslant j \leqslant m+1$, *удовлетворяет условию Липшица с соответствующей константой* L_j;

3) *компоненты* ε_ν *вектора резервов* ε_R *удовлетворяют неравенствам*

$$0 < 2\varepsilon_\nu < L_\nu(\beta - \alpha), \tag{3.51}$$

где $\beta - \alpha$ — *длина интервала* $[\alpha, \beta]$, *принадлежащего допустимой области* Q *из (3.30) и содержащего точку* x^* — *решение задачи*;

4) *при достаточно больших значениях* k *величины* μ_ν *из (3.44), соответствующие непустым множествам* I_ν, *удовлетворяют неравенствам*

$$r_\nu \mu_\nu > 2L_\nu, \quad 1 \leqslant \nu \leqslant m+1, \tag{3.52}$$

где величины $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$ *являются параметрами алгоритма*;

5) *при распараллеливании выполняется условие*

$$T_{l+1}^j - T_l^j < T < \infty, \quad 1 \leqslant j \leqslant p, \quad l > 1,$$

где T_l^j *— момент завершения* j*-м процессором выполнения* l*-го испытания, проводимого на этом процессоре, а для числа* p *используемых процессоров имеет место неравенство* $1 < p < P < \infty$.

Тогда:

1) *точка* x^* *является предельной точкой последовательности* $\{x^k\}$, *порождаемой описанным асинхронным алгоритмом для задачи* (3.1) *при* $\varepsilon = 0$ *в условии остановки* (3.20);

2) *любая предельная точка* $\overline{x}$ *последовательности* $\{x^k\}$ *является решением задачи* (3.1);

3) *сходимость к предельной точке* $\overline{x}$ *является двусторонней, если* $\overline{x} \neq a$ *и* $\overline{x} \neq b$.

Прежде чем приступить к доказательству теоремы, сформулируем и докажем несколько вспомогательных утверждений.

Лемма 3.1 (о последовательности стягивающихся отрезков). *Пусть выполнены условия теоремы* 3.5. *Тогда для каждой предельной точки* $\overline{x}$ *последовательности* $\{x^k\}$ *существует бесконечная последовательность отрезков*

$$\{(x_{t-1}, x_t)\colon t = t(q^s)\}, \quad s = 1, 2, \ldots, \tag{3.53}$$

удовлетворяющих условиям

$$\overline{x} = \bigcap_{s=1}^{\infty} [x_{t-1}, x_t], \tag{3.54}$$

$$\lim_{p\to\infty} \Delta_t = 0, \tag{3.55}$$

$$R(t(q^s)) > 0, \quad s = 1, 2, \ldots, \tag{3.56}$$

$$\lim_{p\to\infty} R(t(q^s)) = 0, \tag{3.57}$$

где $q^1 < q^2 < \ldots$; $R(t)$, Δ_t *и* t *— соответственно из правил* (3.46) *и* (3.47).

Доказательство леммы 3.1. Поскольку $\overline{x}$ является предельной точкой, должна существовать последовательность номеров испытаний $q^1, q^2, \ldots$, удовлетворяющая условиям

$$x^{q+1}, \quad \overline{x} \in [x_{t-1}, x_t], \quad t = t(q), \quad q \in \{q^p\}, \quad p = 1, 2, \ldots\ . \tag{3.58}$$

Эти условия отражают то обстоятельство, что точка $(q+1)$-го испытания попала в интервал $[x_{t-1}, x_t]$, содержащий точку $\overline{x}$ на шаге q, т. е. $t = t(q)$.

Из этого условия, правил (3.48) и (3.49) выбора точки очередного испытания, и оценки значения μ_ν из (3.44) следует, что

$$\max\{x_t - x^{q+1},\ x^{q+1} - x_{t-1}\} \leqslant \gamma\Delta_t, \tag{3.59}$$

где $\gamma = (r+1)/2r < 1$, поскольку

$$r = \min\{r_\nu\colon 1 \leqslant \nu \leqslant m+1\} > 1. \tag{3.60}$$

При этом свойство (3.55) является следствием неравенства (3.59).

На каждом шаге $k \geqslant 1$ существует интервал (x_{i-1}, x_i), $i = i(k)$, на одном из концов которого достигается значение z_ω^* из (3.45). Это значение соответствует индексу ω, определяемому (при $\omega \leqslant m$) условиями $I_\omega \neq \varnothing$, $I_{\omega+1} = \varnothing$. При $I_{m+1} \neq \varnothing$ имеем $\omega = m+1$. Если при этом $\nu(x_{i-1}) \neq \nu(x_i)$, то, согласно (3.46) и (3.49), $R(i) = 2\Delta_i > 0$. В случае когда $\nu(x_{i-1}) = \nu(x_i) = \omega$, из (3.44) следует оценка

$$\max(z_{i-1}, z_i) - z_\omega^* \leqslant \mu_\omega \Delta_i,$$

которая, в сочетании с (3.46), дает оценку

$$R(i) = \Delta_i \left(1 - \frac{\max(z_{i-1}, z_i) - z_\omega^*}{r_\omega \mu_\omega \Delta_i}\right) \geqslant \left(1 - r^{-1}\right)\Delta_i,$$

где r из (3.60). Таким образом, на любом шаге $k \geqslant 1$ поиска существует интервал, имеющий строго положительную характеристику. Отсюда, с учетом правила выбора интервала с максимальной характеристикой (3.47), выводим, что интервалы из системы (3.53) также должны иметь положительные характеристики, так как они подвергаются дроблению попадающими в них точками испытаний. Следовательно, условия (3.56) справедливы.

Из правил проведения испытания и условия (3.34) вытекает, что при $\nu(x_i) \leqslant m$ значение z_i, приписываемое точке x_i из (3.41), является положительным. Поэтому в соответствии с (3.45)

$$z_i = g_\nu(x_i) \geqslant z_\nu^*, \quad 1 \leqslant \nu = \nu(x_i) \leqslant m+1, \quad 0 \leqslant i \leqslant k.$$

Теперь из правил вычисления характеристики (3.46) и предела (3.55), учитывая условие (3.2) липшицевости функций и неравенства (3.56), выводим справедливость утверждения (3.57).

Лемма 3.2 (о существовании допустимой точки). *Пусть выполнены условия теоремы* 3.5. *Тогда существует номер испытания* $h \geqslant 1$, *при котором*

$$\nu(x^h) = m + 1. \tag{3.61}$$

Доказательство леммы 3.2. Допустим противное и рассмотрим интервал $[x_{t-1}, x_t]$, содержащий решение задачи x^* из (3.1) на шаге k, т. е. $t = t(k)$. Поскольку мы допустили, что равенство (3.61) не имеет места и, следовательно, последовательность $\{x^k\}$ не содержит точек из допустимой области Q, интервал $[\alpha, \beta]$ из (3.51) должен удовлетворять условиям

$$x^* \in [\alpha, \beta] \subset [x_{t-1}, x_t]. \tag{3.62}$$

Тогда из этого условия, а также из липшицевости функций задачи следуют отношения

$$z_{t-1} \leqslant L_\nu(\alpha - x_{t-1}) \leqslant L_\nu \Delta_t - L_\nu(\beta - \alpha), \quad \nu = \nu(x_{t-1}), \tag{3.63}$$

$$z_t \leqslant L_\nu(x_t - \beta) \leqslant L_\nu \Delta_t - L_\nu(\beta - \alpha), \quad \nu = \nu(x_t), \tag{3.64}$$

при выводе которых учтено, что

$$g_\nu(x) \leqslant 0, \quad x \in [\alpha, \beta] \subset Q.$$

Аналогично случаю $\nu = \nu(x_{t-1}) = \nu(x_t)$ соответствует неравенство

$$z_t + z_{t-1} \leqslant L_\nu \Delta_t - L_\nu(\beta - \alpha). \tag{3.65}$$

Теперь из правил вычисления характеристики (3.46), неравенств (3.63)–(3.65) и вытекающего из (3.45) неравенства

$$z_\nu^* \geqslant -\varepsilon_\nu$$

следует либо оценка

$$R(t) \geqslant \Delta_t(1 - 2L_\nu/r_\nu\mu_\nu) + 2(L_\nu(\beta - \alpha) - 2\varepsilon_\nu)/r_\nu\mu_\nu,$$

либо оценка

$$R(t) \geqslant 2\Delta_t(1 - 2L_\nu/r_\nu\mu_\nu) + 4(L_\nu(\beta - \alpha) - 2\varepsilon_\nu)/r_\nu\mu_\nu.$$

Учитывая третье и четвертое условия теоремы, в обоих случаях приходим к неравенству

$$R(t(k)) > 0, \tag{3.66}$$

справедливому при достаточно больших значениях k.

Из (3.57), (3.66) и правила (3.47), определяющего выбор интервала, содержащего точку следующего испытания, вытекает неизбежность попадания такой точки в рассматриваемый отрезок $[x_{t-1}, x_t]$ из (3.62). Следствием будет разделение $[x_{t-1}, x_t]$ на два подынтервала. При этом хотя бы один из них будет содержать точку x^*. Повторение этого деления порождает (при $k \to \infty$) бесконечную последовательность подынтервалов, стягивающихся к точке x^*. Т.е. при выполнении третьего и четвертого условий теоремы x^* есть предельная точка последовательности $\{x^k\}$.

Поскольку совпадение x^* с одной из точек испытаний означает справедливость (3.61), то остается рассмотреть случай, когда $x^* \in (x_{t-1}, x_t)$, $t = t(k)$, при любом $k \geqslant 1$. При этом в силу первого условия теоремы и определения (3.30) либо $[x_{t-1}, x^*] \subset Q$, либо $[x^*, x_t] \subset Q$, если число k достаточно велико. Но тогда либо $\nu(x_{t-1}) = m + 1$, либо $\nu(x_t) = m + 1$.

Лемма 3.3 (о допустимости любой предельной точки). *Пусть выполнены условия теоремы* 3.5. *Тогда любая предельная точка* $\overline{x}$ *последовательности* $\{x^k\}$ *является допустимой, т. е.*

$$\nu(\overline{x}) = m + 1. \tag{3.67}$$

Доказательство леммы 3.3. Допустим противное. Согласно (3.54) и (3.55),

$$\overline{x} = \lim_{s \to \infty} x_{t-1} = \lim_{s \to \infty} x_t, \quad t = t(q^s).$$

Поэтому из допущения $\overline{x} \notin Q$ следует существование столь большого числа $S > 1$ и столь малого подынтервала

$$(u, w) \subset [a, b] \backslash Q, \tag{3.68}$$

что

$$[x_{t-1}, x_t] \subset (u, w), \quad t = t(q^s), \quad s \geqslant S. \tag{3.69}$$

Сопоставим множествам Q_j, $1 \leqslant j \leqslant m$, из (3.29) их открытые дополнения $D_j = [a,b] \backslash Q_j$, состоящие (как следствие первого условия теоремы) из конечного числа открытых интервалов положительной длины. При этом

$$g_j(x) > 0, \quad x \in D_j, \quad 1 \leqslant j \leqslant m. \tag{3.70}$$

Из (3.68) вытекает существование таких подмножеств индексов M, $N \subset \{1, 2, \ldots, m\}$, что

$$(u, \overline{x}] \subset \bigcap_{j \in M} D_j, \quad [\overline{x}, w) \subset \bigcap_{j \in N} D_j, \tag{3.71}$$

где (u, w) есть открытый интервал из (3.68).

Согласно правилам проведения испытания, условиям (3.34) и (3.71), для любого интервала из (3.69) справедливы отношения

$$\omega_1 = \nu(x_{t-1}) \in M, \quad \omega_2 = \nu(x_t) \in N, \quad t = t(q^s),$$

где $1 \leqslant \omega_1$, $\omega_2 \leqslant m$. При этом в соответствии с правилами (3.38), (3.39), определяющими порядок проверки ограничений в точках испытаний,

$$\nu(x_{t-1})\, \nu(x_t) \in \{\omega_1, \omega_2\}, \quad t = t(q^{p+1}).$$

То есть при достаточно большом значении $s > 1$ индексы, сопоставленные концевым точкам рассматриваемой системы интервалов $[x_{t-1}, x_t]$, $t = t(q^s)$, $s \geqslant S$, могут принимать лишь значения из множества $\{\omega_1, \omega_2\}$.

Пусть $s > 1$ настолько велико, что при $s \geqslant S$ выполняется неравенство $q^s \geqslant h$, где h — из (3.64). При этом, согласно (3.45),

$$z_\nu^* = -\varepsilon_\nu, \quad 1 \leqslant \nu \leqslant m. \tag{3.72}$$

Тогда из (3.46), (3.55) и (3.70), (3.72) получаем оценку

$$R(t(q^p)) < 2\Delta_t - 4\varepsilon_\nu / r_\nu L_\nu < 0, \quad p \geqslant s, \tag{3.73}$$

где ν — некоторый индекс из множества $\{\omega_1, \omega_2\}$.

Теперь из (3.46), (3.73) и (3.56) выводим, что точка $\overline{x}$, не являющаяся допустимой, не может быть предельной точкой после-

довательности $\{x^k\}$, т. е. для любой предельной точки $\overline{x}$ должно выполняться условие (3.67).

Доказательство теоремы 3.5

1. Опираясь на леммы 3.2, 3.3, докажем первое утверждение теоремы, т. е. что точка x^* из (3.1) является предельной. Допустим противное и рассмотрим два возможных случая. Пусть $[x_{t-1}, x_t]$, $t = t(k)$, — интервал, содержащий точку x^* на шаге k. Рассмотрим условие

$$\max\{\nu(x_{t-1}), \nu(x_t)\} = m + 1 \tag{3.74}$$

для индексов концевых точек обсуждаемого интервала и допустим, что оно не выполняется при достаточно больших значениях k. При этом должны иметь место включения (3.62). Такой случай уже рассматривался в лемме 3.2, где была выявлена его несовместимость с допущением непредельного характера точки x^*.

Остается рассмотреть случай, когда условие (3.74) выполняется для некоторой подпоследовательности значений k. При этом из (3.2) следуют отношения

$$z_{t-1} \leqslant g_\nu(x^*) + L_\nu(x^* - x_{t-1}), \quad \nu = \nu(x_{t-1}),$$
$$z_t \leqslant g_\nu(x^*) + L_\nu(x_t - x^*), \quad \nu = \nu(x_t),$$

причем случаю

$$\nu(x_{t-1}) = \nu(x_t) = m + 1 \tag{3.75}$$

соответствуют равенство $g_{m+1}(x^*) = \varphi(x^*)$ и неравенство

$$z_t + z_{t-1} \leqslant 2\varphi(x^*) + L_{m+1}\Delta_t.$$

При этом из (3.46), учитывая (3.74) и вытекающее из (3.45) неравенство $z^*_{m+1} \geqslant \varphi(x^*)$, выводим оценку

$$R(t) \geqslant \gamma\Delta_t(1 - 2L_{m+1}/r_{m+1}\mu_{m+1}), \tag{3.76}$$

где в случае справедливости равенства (3.75) значение коэффициента $\gamma = 1$. В противном случае $\gamma = 2$.

Теперь из (3.76) и четвертого условия теоремы вытекает справедливость неравенства (3.66) для некоторой подпоследовательности

значений k, что, как уже обсуждалось во втором пункте, ведет к противоречию с допущением непредельного характера точки x^*.

2. Покажем, что любая предельная точка $\overline{x}$ последовательности $\{x^k\}$, порождаемой МАПП, является решением задачи (3.1) при выполнении условий теоремы.

Поскольку точка x^* из (3.1) является предельной, то, согласно (3.45),

$$\lim_{p\to\infty} z^*_{m+1} = \varphi(x^*). \tag{3.77}$$

Пусть $\overline{x} \neq x^*$ — некоторая другая предельная точка последовательности $\{x^k\}$, являющаяся допустимой согласно (3.67). Тогда для характеристики интервала $[x_{j-1}, x_j]$, содержащего точку $\overline{x}$ на шаге k, в силу условия Липшица (3.2), (3.44), (3.46), (3.55) и (3.77), справедливо отношение

$$\lim_{k\to\infty} R(j(k)) \leqslant -4(\varphi(\overline{x}) - \varphi(x^*))/r_{m+1}L_{m+1}.$$

При $\varphi(\overline{x}) > \varphi(x^*)$ характеристика интервала $[x_{j-1}, x_j]$, содержащего точку $\overline{x}$, начиная с некоторого шага, окажется отрицательной, что противоречит свойству (3.56), которое должно выполняться для каждой предельной точки. Следовательно, должно быть справедливо равенство $\varphi(\overline{x}) = \varphi(x^*)$. *Теорема доказана.*

Глава 4

Параллельные вычисления для многомерной многоэкстремальной оптимизации на основе многошаговой схемы редукции размерности

В главе 1 было отмечено, что среди схем редукции сложности многоэкстремальных задач оптимизации важное место занимают схемы редукции размерности. Настоящая глава посвящена рассмотрению одной из таких схем — так называемой многошаговой схемы редукции размерности, позволяющей заменить решение многомерной задачи решением семейства одномерных подзадач, рекурсивно связанных между собой.

Наряду с общей схемой редукции рассматривается ее реализация в сочетании с различными алгоритмами глобального поиска. Многошаговая схема обладает значительным потенциалом параллелизма, поскольку для решения одномерных подзадач возможно применение одномерных параллельных алгоритмов оптимизации, в частности, характеристических. Неотъемлемой частью многошаговой схемы является асинхронизм параллельных вычислений, вытекающий из рекурсивной взаимосвязи подзадач.

4.1. Общая схема многошаговой схемы редукции размерности

Рассмотрим задачу нелинейного программирования в форме (1.6)–(1.8) и предположим, что все функции-ограничения $g_j(y)$, $1 \leqslant j \leqslant m$ являются непрерывными, а область D — ограниченной, что обеспечивает компактность области Q.

Введем непрерывную функцию $G(y)$, определенную в области D, такую что

$$\begin{aligned} G(y) &\leqslant 0, \quad y \in Q, \\ G(y) &> 0, \quad y \notin Q. \end{aligned} \tag{4.1}$$

В качестве $G(y)$ можно взять, например,

$$G(y) = \max\{g_1(y), \dots, g_m(y)\} \tag{4.2}$$

или

$$G(y) = \max\{0; g_1(y), \dots, g_m(y)\}. \tag{4.3}$$

Последняя функция тождественно равна нулю в области Q.

Введем обозначения

$$u_i = (y_1, \dots, y_i), \quad v_i = (y_{i+1}, \dots, y_N), \tag{4.4}$$

позволяющие при $1 \leqslant i \leqslant N-1$ записать вектор y в виде пары $y = (u_i, v_i)$, и примем, что $y = v_0$ при $i = 0$ и $y = u_N$ при $i = N$.

Введем сечения множества Q:

$$S_1 = Q, \quad S_{i+1}(u_i) = \{v_i \in R^{N-i} \colon (u_i, v_i) \in Q\}, \quad 1 \leqslant i \leqslant N-1, \tag{4.5}$$

и проекции сечений $S_{i+1}(u_i)$ на ось y_{i+1}:

$$\Pi_{i+1}(u_i) = \{y_{i+1} \in R^1 \colon \exists\ (y_{i+1}, v_{i+1}) \in S_{i+1}(u_i)\}. \tag{4.6}$$

Положим $G^N(y) \equiv G(y)$ и построим семейство функций

$$G^i(u_i) = \min\{G^{i+1}(u_i, y_{i+1}) \colon y_{i+1} \in [a_{i+1}, b_{i+1}]\}, \quad 1 \leqslant i \leqslant N-1, \tag{4.7}$$

определенных в соответствующих проекциях

$$D_i = \{u_i \in R^i \colon y_j \in [a_j, b_j],\ 1 \leqslant j \leqslant i\} \tag{4.8}$$

множества D из (1.8) на координатные оси $y_1, \dots, y_i$, причем по определению $D_N = D$.

В силу непрерывности функции $G^N(y) \equiv G(y)$ и компактности области D функция $G^{N-1}(u_{N-1})$ существует и непрерывна в D_{N-1}, что обуславливает существование и непрерывность функции $G^{N-2}(u_{N-2})$ в D_{N-2} и далее определяет существование и непрерывность всех функций семейства (4.7) в соответствующих областях (4.8).

Справедлива следующая лемма.

Лемма 4.1.

$$G^i(u_i) = \min\{G(u_i, v_i)\colon y_j \in [a_j, b_j],\ i{+}1 \leqslant j \leqslant N\}, \quad 1 \leqslant i \leqslant N-1. \tag{4.9}$$

Доказательство. Вследствие (4.7) доказательство (4.9) состоит в установлении справедливости равенства

$$\min_{y_{i+1}\in[a_{i+1},b_{i+1}]} \min_{y_N\in[a_N,b_N]} G(u_i, v_i) = \\ = \min\{G(u_i, v_i)\colon y_j \in [a_j, b_j],\ i{+}1 \leqslant j \leqslant N\}. \tag{4.10}$$

В силу непрерывности $G(y)$, для любого $u_i \in D_i$ существует $\overline{v}_i = (\overline{y}_{i+1}, \ldots, \overline{y}_N)$ такой, что $\overline{y}_j \in [a_j, b_j]$, $i+1 \leqslant j \leqslant N$, и

$$\min_{y_{i+1}\in[a_{i+1},b_{i+1}]} \ldots \min_{y_N\in[a_N,b_N]} G(u_i, v_i) = G(u_i, \overline{v}_i),$$

откуда следует, что левая часть равенства (4.10) больше или равна правой.

Покажем обратное неравенство. В силу непрерывности $G(y)$ и компактности D существует вектор $v_i^* = (y_{i+1}^*, \ldots, y_N^*)$ такой, что

$$G(u_i, v_i^*) = \min\{G(u_i, v_i)\colon y_j \in [a_j, b_j],\ i{+}1 \leqslant j \leqslant N\}.$$

Согласно (4.7), имеем:

$$G^{N-1}(u_i, y_{i+1}^*, \ldots, y_{N-1}^*) = \\ = \min\{G(u_i, y_{i+1}^*, \ldots, y_{N-1}^*, y_N)\colon y_N \in [a_N, b_N]\} \leqslant G(u_i, v_i^*),$$

$$G^{N-2}(u_i, y_{i+1}^*, \ldots, y_{N-2}^*) = \\ = \min\{G^{N-1}(u_i, y_{i+1}^*, \ldots, y_{N-2}^*, y_{N-1})\colon y_{N-1} \in [a_{N-1}, b_{N-1}]\} \leqslant \\ \leqslant G^{N-1}(u_i, y_{i+1}^*, \ldots, y_{N-1}^*),$$

$$G^i(u_i) = \min\{G^{i+1}(u_i, y_{i+1})\colon y_{i+1} \in [a_{i+1}, b_{i+1}]\} \leqslant G^{i+1}(u_i, y_{i+1}^*) \leqslant \\ \leqslant \ldots \leqslant G(u_i, v_i^*).$$

Лемма доказана.

Введем проекции

$$Q_i = \{u_i \in R^i\colon \exists\ (u_i, v_i) \in Q\}, \quad 1 \leqslant i \leqslant N, \tag{4.11}$$

множества Q на координатные оси $y_1, \ldots, y_i$.

Лемма 4.2. *Представление* (4.11) *эквивалентно соотношению*

$$Q_i = \{u_i \in R^i\colon G^i(u_i) \leqslant 0\}. \tag{4.12}$$

Доказательство. Пусть выполнено (4.11), т. е. для некоторого u_i существует v_i^* такой, что $(u_i, v_i^*) \in Q$. Но тогда $G(u_i, v_i^*) \leqslant 0$, т. е. вследствие леммы 4.1

$$G^i(u_i) = \min\{G(u_i, v_i)\colon y_j \in [a_j, b_j], i+1 \leqslant j \leqslant N\} \leqslant G(u_i, v_i^*) \leqslant 0,$$

и (4.12) справедливо.

Пусть теперь, наоборот, для некоторого $u_i \in Q_i$ выполняется $G^i(u_i) \leqslant 0$. Но согласно лемме 4.1 существует вектор v_i^* такой, что $G^i(u_i) = G(u_i, v_i^*)$, т. е. $(u_i, v_i^*) \in Q$. *Лемма доказана.*

Лемма 3.3. *Определение проекции* (4.6) *эквивалентно представлению*

$$\Pi_{i+1}(u_i) = \{y_{i+1} \in [a_{i+1}, b_{i+1}]\colon G^{i+1}(u_i, y_{i+1}) \leqslant 0\}. \tag{4.13}$$

Доказательство. Прежде всего, заметим, что поскольку $Q \subseteq D$, то необходимо выполнено $a_j \leqslant y_j \leqslant b_j$, $1 \leqslant j \leqslant N$. Пусть теперь y_{i+1} таков, что $G^{i+1}(u_i, y_{i+1}) \leqslant 0$. Тогда существует v_{i+1}^* такой, что $G(u_i, y_{i+1}, v_{i+1}^*) \leqslant 0$, т. е. $(y_{i+1}, v_{i+1}^*) \in S_{i+1}(u_i)$, следовательно, y_{i+1} принадлежит проекции $\Pi_{i+1}(u_i)$ в смысле определения (4.6).

Предположим далее, что для некоторого y_{i+1} существует v_{i+1}^* такой, что $(y_{i+1}, v_{i+1}^*) \in S_{i+1}(u_i)$. Тогда вектор $(u_i, y_{i+1}, v_{i+1}^*) \in Q$, т. е. $G^{i+1}(u_i, y_{i+1}) \leqslant G(u_i, y_{i+1}, v_{i+1}^*) \leqslant 0$. *Лемма доказана.*

Предположим теперь непрерывность функции $\varphi(y)$ и, положив по определению $\varphi^N(y) \equiv \varphi(y)$, построим семейство функций

$$\varphi^i(u_i) = \min\{\varphi^{i+1}(u_i, y_{i+1})\colon y_{i+1} \in \Pi_{i+1}(u_i)\}, \quad 1 \leqslant i \leqslant N-1, \tag{4.14}$$

определенных на соответствующих проекциях Q_i. Тогда имеет место [29, 67] основное соотношение

$$\min_{y \in Q} \varphi(y) = \min_{y_1 \in \Pi_1} \min_{y_2 \in \Pi_2(u_1)} \dots \min_{y y_N \in \Pi_N(u_{N-1})} \varphi(y). \tag{4.15}$$

Как следует из (4.15), для решения задачи (1.6)–(1.8) достаточно решить одномерную задачу

$$\varphi^1(y_1) \to \min, \quad y_1 \in \Pi_1 \subseteq R^1,$$
$$\Pi_1 = \{y_1 \in [a_1, b_1]\colon G^1(y_1) \leqslant 0\}.$$

При этом каждое вычисление функции $\varphi^1(y_1)$ в некоторой фиксированной точке $y_1 \in \Pi_1$ представляет собой, согласно (4.14), решение одномерной задачи

$$\varphi^2(y_1, y_2) \to \min, \quad y_2 \in \Pi_2(y_1) \subseteq R^1,$$
$$\Pi_2(y_1) = \{y_2 \in [a_2, b_2] : G^2(y_1, y_2) \leqslant 0\}.$$

Эта задача является одномерной задачей минимизации по y_2, так как y_1 фиксировано.

В свою очередь, каждое вычисление значения функции $\varphi^2(y_1, y_2)$ при фиксированных y_1, y_2 требует решения одномерной задачи

$$\varphi^3(u_2, y_3) \to \min, \quad y_3 \in \Pi_3(u_2),$$

и т. д. вплоть до решения задачи

$$\varphi^N(u_{N-1}, y_N) = \varphi(u_{N-1}, y_N) \to \min, \quad y_N \in \Pi_N(u_{N-1})$$

при фиксированном u_{N-1}.

Окончательно решение задачи (1.6)–(1.8) сводится к решению семейства «вложенных» одномерных подзадач

$$\varphi^i(u_{i-1}, y_i) \to \min, \quad y_i \in \Pi_i(u_{i-1}), \tag{4.16}$$

где фиксированный вектор $u_{i-1} \in Q_{i-1}$.

Решение исходной многомерной задачи (1.6)–(1.8) через решение системы взаимосвязанных одномерных подзадач (4.16) называется *многошаговой схемой редукции размерности*.

Заметим, что если в задачах (4.16) по некоторым координатам y_i искать не минимум, а максимум, то становится возможным вычисление сложных минимаксных (максиминных) выражений вида

$$\operatorname*{extr}_{y*\in Q} \varphi(y) = \operatorname*{extr}_{y_1\in\Pi_1} \operatorname*{extr}_{y_2\in\Pi_2(u_1)} \ldots \operatorname*{extr}_{y_N\in\Pi_N(u_{N-1})} \varphi(y),$$

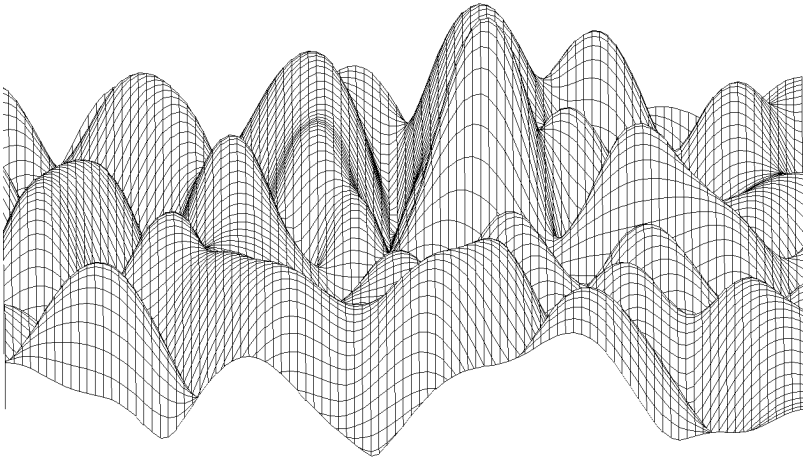

Рис. 4.1. Целевая функция

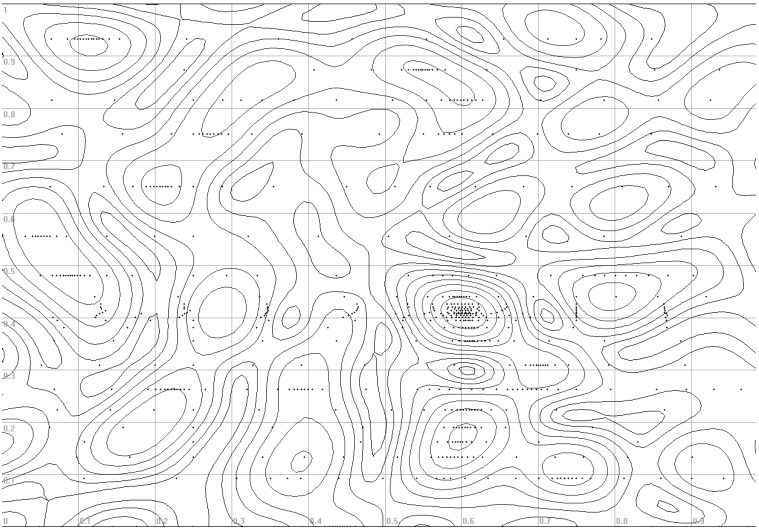

Рис. 4.2. Линии уровня и точки испытаний

где операция extr означает вычисление глобального минимума или максимума.

Для иллюстрации приведем пример максимизации по многошаговой схеме двумерной многоэкстремальной функции из класса (2.103). На рис. 4.1 показана поверхность целевой функции, а на рис. 4.2 — линии уровня данной функции с отображением координат поисковых испытаний по многошаговой схеме с использованием АГП (2.35)–(2.38) при решении одномерных подзадач.

4.2. Свойства одномерных подзадач многошаговой схемы

При анализе одномерных подзадач (4.16) многошаговой схемы возникают две основные проблемы:

а) необходимо сконструировать допустимые области одномерного поиска $\Pi_i(u_{i-1})$;

б) требуется обеспечить минимизацию одномерных функций $\varphi^i(u_{i-1}, y_i)$ в областях $\Pi_i(u_{i-1})$.

Структура и сложность проекций $\Pi_i(u_{i-1})$ полностью определяются сложностью многомерной допустимой области Q. Сложность второй проблемы зависит от характеристик функций $\varphi^i(u_i)$, на которые влияют как свойства целевой функции $\varphi(y)$, так и особенности области поиска Q, определяемые ограничениями (1.7), (1.8).

4.2.1. Структура допустимых областей одномерного поиска

Для анализа структуры областей $\Pi_i(u_{i-1})$ используем результаты леммы 4.3, которая установила эквивалентность определения (4.6) и представления (4.13). Дело в том, что (4.13) дает конструктивный аппарат построения области $\Pi_i(u_{i-1})$, связывая ее с областью неположительности функции $G^i(u_{i-1}, y_i)$.

Так как функция $G(y)$ предполагается непрерывной в области D, то функции $G^i(u_i)$ также являются непрерывными по $u_i \in D_i$ из (4.8), а тем самым и по $y_i \in [a_i, b_i]$. Тогда при фиксированном

$u_{i-1} \in D_{i-1}$ каждая из одномерных задач (4.16) является задачей вида

$$\begin{gathered} \overline{\varphi}(x) \to \min, \quad x \in \overline{Q} \subset R^1, \\ \overline{Q} = \{x \in [a, b]\colon g(x) \leqslant 0\}, \end{gathered} \tag{4.17}$$

причем функция $g(x)$ непрерывна.

Непрерывность ограничения $g(x)$ позволяет утверждать, что допустимая область $\overline{Q}$ может быть записана в виде системы отрезков

$$\overline{Q} = \bigcup_{j=1}^{q} [a^j, b^j] y_i \in \Pi_i(u_{i-1}), \tag{4.18}$$

на каждом из которых функция неположительна. Для примера рассмотрим рисунок, который отражает возможные случаи поведения непрерывной функции, порождающие области неположительности в виде отрезков (помечены утолщениями на оси x), включая касание оси x в точке. Самый правый отрезок, отмеченный жирным пунктиром, соответствует ситуации, когда функция на данном отрезке равна нулю во всех его точках.

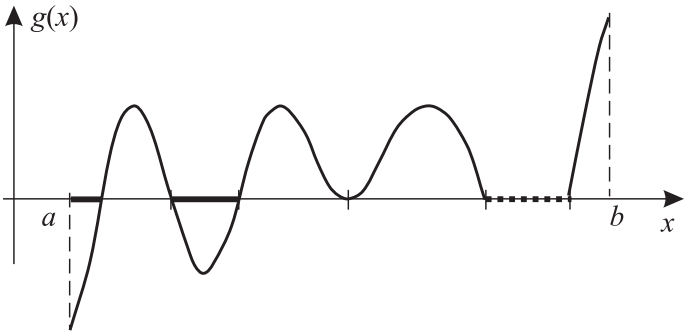

Рис. 4.3. Области неположительности непрерывной функций

В системе (4.18) число q отрезков может быть бесконечным. В качестве примера подобной ситуации приведем функцию

$$g(x) = \begin{cases} x \sin \frac{1}{x}, & x > 0, \\ 0, & x = 0, \end{cases}$$

рассматриваемую на отрезке $[0, 1]$.

Таким образом, в случае непрерывной функции $G(y)$ проекция $\Pi_i(u_{i-1})$ из (4.16) есть множество вида (4.18), т. е.

$$\Pi_i(u_{i-1}) = \bigcup_{j=1}^{q_i} [a_i^j, b_i^j], \tag{4.19}$$

где в общем случае количество отрезков q_iи их границы a_i^j, b_i^j, $1 \leqslant j \leqslant q_i$, зависят от вектора u_{i-1}, т. е.

$$q_i = q_i(u_{i-1}), \quad a_i^j = a_i^j(u_{i-1}), \quad b_i^j = b_i^j(u_{i-1}). \tag{4.20}$$

Если область Q такова, что для всех $1 \leqslant i \leqslant N$ удается указать явные (аналитические) выражения для величин q_i, a_i^j, b_i^j как функций $u_{i-1} \in Q_{i-1}$, тогда область Q называется *областью с вычислимой границей*. Для построения данных областей необходимо уметь аналитически находить все корни функций $G^i(u_{i-1}, y_i)$ по соответствующим переменным y_i.

Проиллюстрируем общие результаты несложным конкретным примером.

Пример 4.1. Рассмотрим двумерную задачу (1.6)–(1.8), в которой

$$\begin{gathered} \varphi(y) = y_1^2 + y_2^2, \\ Q = \{y \in D\colon (y_1 - 1)^2 + (y_2 - 1)^2 - 1 \leqslant 0\}, \\ D = \{y \in R^2\colon 0 \leqslant y_1,\ y_2 \leqslant 2\}. \end{gathered} \tag{4.21}$$

На рисунке серым цветом отмечена допустимая область, а также показаны концентрические линии уровня целевой функции. В этой задаче функция $G^2(y) = (y_1 - 1)^2 + (y_2 - 1)^2 - 1$, а функция $G^1(y) = \min\{G^2(y_1, y_2)\colon\ y_2 \in [0, 2]\} = (y_1 - 1)^2 - 1$, поскольку функция $G^2(y)$ достигает своего минимума по y_2 на отрезке $[0, 2]$ в точке $y_2 = 1$.

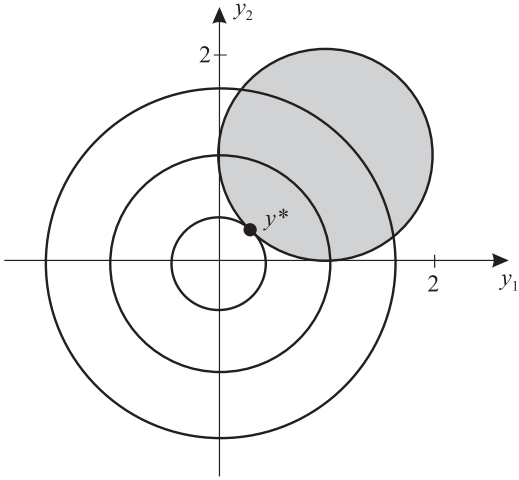

Рис. 4.4. Линии уровня целевой функции и допустимая область

Согласно (4.13), области неположительности функций $G^1(y_1)$ по y_1 и $G^2(y_1, y_2)$ по y_2 определяют проекции Π_1 и $\Pi_2(y_1)$ соответственно. Границы областей задаются корнями указанных функций, принадлежащими отрезку $[0, 2]$. Для функции $G^1(y_1)$ такими корнями являются значения 0 и 2, поэтому $\Pi_1 = [0, 2]$.

Функция $G^2(y) = (y_2 - 1)^2 - \alpha^2$, где $\alpha = \sqrt{1 - (y_1 - 1)^2} \leqslant 1$, имеет корни $1 \pm \alpha$, очевидно принадлежащие отрезку $[0, 2]$, и неположительна между этими корнями, поэтому

$$\Pi_2(y_1) = \left[1 - \sqrt{1 - (y_1 - 1)^2}\,,\ 1 + \sqrt{1 - (y_1 - 1)^2}\right]. \qquad (4.22)$$

Таким образом, нам удалось записать границы проекций (4.19) в явном виде. Тем самым область Q из (4.21) является областью с вычислимой границей (4.20).

Функция $\varphi(y) = y_1^2 + y_2^2$, будучи возрастающей по y_2 на отрезке $[0, 2]$ и, следовательно, в области (4.22), достигает своего минимума

в точке $1-\alpha$, поэтому

$$\varphi^1(y_1) = y_1^2 + \left(1 - \sqrt{1-(y_1-1)^2}\right)^2 = 1 + 2y_1 - 2\sqrt{1-(y_1-1)^2}. \tag{4.23}$$

Первая производная $(\varphi(y_1))' = 2 + 2(y_1-1)\big/\sqrt{1-(y_1-1)^2}$ имеет единственный корень $y_1^* = 1 - 1\big/\sqrt{2}$. К тому же вторая производная $(\varphi(y_1))'' = 2\big/\left(1-(y_1-1)^2\right)^{3/2} > 0$, поэтому точка $y_1^* = 1 - 1\big/\sqrt{2}$ доставляет минимальное значение функции $\varphi^1(y_1)$, равное $3 - 2\sqrt{2}$. Это значение и является искомым минимальным значением функции $\varphi(y)$ в области Q. Для определения координаты y_2, которая совместно с $y_1^* = 1 - 1\big/\sqrt{2}$ задает точку минимума, рассмотрим функцию $\varphi^2(y_1^*, y_2) = \left(1 - 1\big/\sqrt{2}\right)^2 + y_2^2$ и найдем ее минимум в области $\Pi_2(y_1^*) = \left[1-1\big/\sqrt{2},\ 1+1\big/\sqrt{2}\right]$, который очевидно достигается в точке $y_2^* = 1 - 1\big/\sqrt{2}$.

Как итог решением исходной многомерной задачи (4.21) является вектор $y^* = \left(1-1\big/\sqrt{2},\ 1-1\big/\sqrt{2}\right)$, обеспечивающий минимальное значение целевой функции $\varphi(y^*) = 3 - 2\sqrt{2}$. На рис. 4.2 точка оптимума помечена темным кружком.

В рассмотренном примере мы построили границы областей одномерного поиска аналитически, установив области неположительности соответствующих функций $G^i(u_i)$. Вместе с тем можно указать более наглядный «геометрический» способ построения проекций $\Pi_{i+1}(u_i)$. Собственно, этот способ вытекает из определений (4.5) и (4.6) и состоит в том, что необходимо построить сечения области Q плоскостями $u_i = \text{const}$ и затем установить границы этих сечений по координате y_{i+1}.

В связи с этим обратим внимание на следующее. Вычисление глобального минимума в соответствии с соотношением (4.15) аналогично процедуре нахождения многомерного интеграла от функции $\varphi(y)$ в области Q посредством сведения к вычислению повторных одномерных интегралов. При этом области одномерного интегрирования как раз и являются соответствующими проекциями $\Pi_{i+1}(u_i)$.

Для иллюстрации рассмотрим следующий пример.

Пример 4.2. Пусть область оптимизации (или интегрирования) задается как

$$Q = \{y \in R^2 \colon -4 \leqslant y_1,\ y_2 \leqslant 4,\ y_1^2 + y_2^2 \leqslant 4,\ y_1^2 + y_2^2 \geqslant 1\}.$$

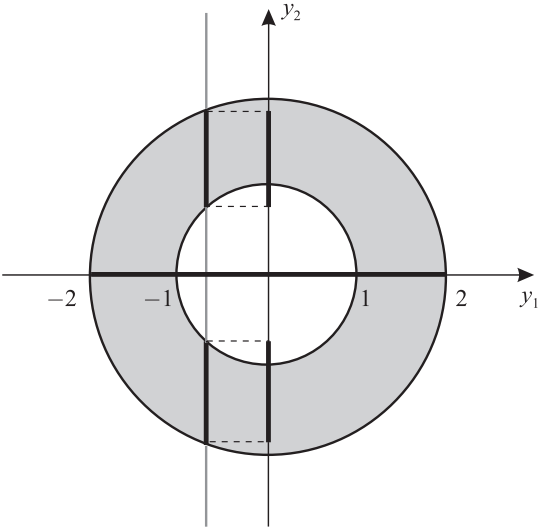

Рис. 4.5. Допустимая область, сечение и проекция

Серый цвет, как и ранее, помечает допустимую область. Левая прямая $y_1 = \text{const}$ на пересечении с допустимой областью формирует сечение $S_1(y_1)$, а его проектирование на ось y_2 определяет проекцию $\Pi_2(y_1)$. Данная проекция, а также проекция Π_1 на рис. 4.3 отмечены утолщенными линиями.

Подобные геометрические соображения позволяют построить требуемые проекции задачи как

$$\Pi_1 = [-2, 2];$$

$$\Pi_2(y_1) = \begin{cases} \left[-\sqrt{4-y_1^2},\ \sqrt{4-y_1^2}\right], \\ \qquad\qquad y_1 \in [-2,-1] \cup [1,2]; \\ \left[-\sqrt{4-y_1^2},\ -\sqrt{1-y_1^2}\right] \bigcup \left[\sqrt{1-y_1^2},\ \sqrt{4-y_1^2}\right], \\ \qquad\qquad y_1 \in [-1,1]. \end{cases}$$

В общем случае, однако, нахождение всех корней непрерывной функции является сложной задачей, не разрешаемой аналитически, и в этом случае можно попытаться найти искомые корни численно. Рассмотрим, к примеру, типовую задачу (4.17). Для отыскания всех корней непрерывной функции $g(x)$ можно предложить численное решение эквивалентной задачи оптимизации

$$|g(x)| \to \min, \quad x \in [a,b],$$

в которой корни $g(x)$ являются точками глобального минимума. Для решения этой задачи могли бы быть использованы характеристические алгоритмы глобального поиска, обеспечивающие сходимость ко всем глобально-минимальным точкам.

Другой подход к учету ограничений в задачах оптимизации (индексный метод), не требующий решения вспомогательных задач (4.23), рассмотрен в разделе 3.3 предыдущей главы.

Практически важным частным случаем задачи (1.6)–(1.8) является случай $Q = D$, когда функциональные ограничения (1.7) отсутствуют. В данной ситуации $G(y) \equiv 0$ в области D, а из (4.9) следует, что функции $G^i(u_i) \equiv 0$, $u_i \in D_i$. Тогда, согласно (4.13),

$$\Pi_i(u_{i-1}) = [a_i, b_i],$$

где a_i, b_i — константы.

Области одномерной оптимизации могут представлять собой единственный отрезок и при наличии функциональных ограничений, обладающих определенными свойствами. Такими свойствами являются выпуклость функций $g_j(y)$, $1 \leqslant j \leqslant m$, обеспечивающая выпуклость допустимой области Q, и свойство монотонной унимодальности функциональных ограничений [22], которое является

более общим, чем понятие выпуклости, а ограничения, обладающие данным свойством, могут порождать невыпуклые области Q.

В случае монотонной унимодальности (в частном случае — выпуклости) ограничений $g_j(y)$, $1 \leqslant j \leqslant m$, области одномерного поиска являются отрезками

$$\Pi_i(u_{i-1}) = [a_i^1(u_{i-1}),\ b_i^1(u_{i-1})], \tag{4.24}$$

границы, которых, однако, не являются константами, но зависят от вектора u_{i-1}.

4.2.2. Свойства целевых функций в одномерных подзадачах

Целевой функцией в подзадаче (4.16) является функция $\varphi^i(u_{i-1}, y_i)$ при фиксированном u_{i-1} и для решения подзадач (4.16) определяющим является характер зависимости функции φ^i от переменной y_i.

Рассмотрим класс задач, в которых функция $\varphi(y)$ является сепарабельной, т. е.

$$\varphi(y) = \sum_{i=1}^{N} \varphi_i(y_i), \tag{4.25}$$

а функциональные ограничения $g_j(y)$ отсутствуют, т. е. $Q = D$.

Тогда, как следует из (4.15), (4.25),

$$\min_{y \in Q} \varphi(y) = \sum_{i=1}^{N} \min_{y_i \in [a_i, b_i]} \varphi_i(y_i),$$

т. е. для решения многомерной задачи требуется решение N независимых одномерных подзадач. Для этого класса задач порядок роста сложности с ростом размерности линейный.

Предположим теперь, что функция $\varphi(y)$ удовлетворяет в области Q условию Липшица с константой $L > 0$, т. е. для любых y', $y'' \in Q$

$$|\varphi(y') - \varphi(y'')| \leqslant L \left\| y' - y'' \right\|. \tag{4.26}$$

Естественным в этом случае является вопрос: а будут ли удовлетворять условию Липшица функции $\varphi^i(u_i)$? Оказывается, ответ не всегда положительный.

Рассмотрим следующий пример. Пусть в задаче (1.6)–(1.8) целевая функция $\varphi(y)$ удовлетворяет условию Липшица (4.26), а область Q имеет вид

$$Q = \{y \in R^2\colon y_1^2 + y_2^2 - 1 \leqslant 0\}.$$

Тогда функция $\varphi^2(y) \equiv \varphi(y)$, естественно, является липшицевой с константой L по координате y_2. Однако функция $\varphi^1(y_1)$ условию (4.26) уже не подчиняется. В [42] показано, что эта функция удовлетворяет обобщенному условию Липшица (условию Гёльдера) в метрике $\rho(y_1', y_1'') = \sqrt{\left|y_1' - y_1''\right|}$ с константой $L_1 = L\left(1 + \sqrt{2}\right)$, т. е. условию

$$\left|\varphi(y_1') - \varphi(y_1'')\right| \leqslant L_1 \sqrt{\left|y_1' - y_1''\right|}.$$

Следующая теорема устанавливает достаточные условия липшицевости функций $\varphi^i(u_i)$.

Теорема 4.1 [42]. *Пусть функция $\varphi(y)$ является липшицевой с константой L в выпуклой области Q из* (1.7) *и граничные пары* (4.24) *являются кусочно-линейными функциями вида*

$$a_{i+1}(u_i) = \max_{1 \leqslant \nu \leqslant p_i} \left\{\alpha_i^\nu u_i + A_i^\nu\right\}, \tag{4.27}$$

$$b_{i+1}(u_i) = \max_{1 \leqslant \nu \leqslant r_i} \left\{\beta_i^\nu u_i + B_i^\nu\right\}, \tag{4.28}$$

где $\alpha_i^\nu u_i$, $\beta_i^\nu u_i$ — скалярные произведения векторов из R^i и A_i^ν, B_i^ν — константы. Тогда функции $\varphi^i(u_i)$, $u_i \in Q_i$ являются липшицевыми с константами L_i, $1 \leqslant i \leqslant N$, где

$$L_N = L, \quad L_i = L \prod_{j=i}^{N-1} (1 + \lambda_j), \quad 1 \leqslant i \leqslant N - 1,$$

$$\lambda_j = \max\left\{\max_{1 \leqslant \nu \leqslant p_i} \left\|\alpha_j^\nu\right\|, \ \max_{1 \leqslant \nu \leqslant r_i} \left\|\beta_j^\nu\right\|\right\}.$$

Доказательство теоремы приведено в [42].

Отметим, что представление граничных пар в виде (4.27), (4.28) имеет место, если допустимая область является выпуклым многогранником.

Если область Q является гиперпараллелепипедом, то все векторы α_i^ν, β_i^ν нулевые, и поэтому $L_i = L$, $1 \leqslant i \leqslant N$.

Основной вывод, который следует из обсуждения липшицевости, состоит в том, что на свойства целевых функций одномерных подзадач существенное влияние оказывают не только свойства исходной целевой функции $f(y)$, но и вид допустимой области Q.

В самом простом случае, когда множество Q является гиперпараллелепипедом, все функции $f^i(u_i)$ будут удовлетворять условию Липшица с той же константой, что и функция $f(y)$.

Если ограничения $g_j(y)$ являются кусочно-линейными выпуклыми функциями, то липшицевость функций $f^i(u_i)$ сохраняется, но константа Липшица для этих функций, вообще говоря, увеличивается, что ухудшает оптимизационные свойства одномерных подзадач.

Наконец, случай нелинейных ограничений может вообще привести к потере липшицевости.

4.3. Параллельные вычисления для многошаговой схемы редукции размерности

Применяя в структуре многошаговой схемы (4.15) для решения одномерных подзадач (4.16) параллельные методы одномерной оптимизации, мы получим параллельный вариант многошаговой схемы с широкой степенью вариативности. Например, можно варьировать количество процессоров на различных уровнях одномерной оптимизации (т. е. при решении одномерных подзадач по различным переменным y_i), применять различные параллельные методы одномерного поиска на разных уровнях и т. п.

Для описания процесса параллелизма многошаговой схемы введем вектор степеней распараллеливания

$$\pi = (\pi_1, \pi_2, \ldots, \pi_N), \tag{4.29}$$

где π_i, $1 \leqslant i \leqslant N$ обозначает число параллельно решаемых одномерных подзадач $(i+1)$-го уровня рекурсии, возникающих в результате выполнения параллельных итераций на i-м уровне одномерной

оптимизации. Для координаты y_N число π_N означает количество параллельных испытаний в процессе минимизации функции $\varphi^N(y_1, \dots, y_N) = \varphi(y_1, \dots, y_N)$ по y_N при фиксированных значениях $y_1, \dots, y_{N-1}$, т. е. количество параллельно вычисляемых значений целевой функции $\varphi(y)$.

В общем случае величины π_i, $1 \leqslant i \leqslant N$ могут зависеть от различных параметров и меняться в процессе оптимизации, но мы ограничимся случаем, когда все компоненты вектора (4.29) постоянны.

Применение одномерных параллельных алгоритмов в сочетании со схемой (4.15)–(4.16) и вектором (4.29) позволяет использовать для решения задачи (1.6)–(1.8) вплоть до

$$\Pi = \prod_{i=1}^{N} \pi_i$$

параллельно работающих процессоров. При этом многошаговая схема генерирует до Π/π_N одномерных подзадач оптимизации, решаемых одновременно.

Пусть, например, размерность $N = 2$. Тогда использование метода с π_1 параллельными испытаниями в итерации для решения внешней одномерной задачи

$$\varphi^1(y_1) \to \min, \quad y_1 \in \Pi_1 \subseteq R^1$$

генерирует π_1 внутренних одномерных подзадач

$$\varphi^2(y_1, y_2) \to \min, \quad y_2 \in \Pi_2(y_1) \subseteq R^1,$$

и каждая из них может быть решена параллельным методом с π_2 испытаниями в итерации. Это означает, что на самом внутреннем уровне, соответствующем $i = N$ в (4.16), вычисляются $\Pi = \pi_1\pi_2$ значений функции $\varphi(y)$, т. е. для вычисления этих значений мы можем применить Π процессоров, работающих параллельно. На рис. 4.6 представлено одно из возможных распределений $\Pi = 6$ параллельных процессоров.

Многошаговая схема обладает существенным резервом эффективности распараллеливания за счет введения асинхронизма. Объяснение этого факта состоит в следующем.

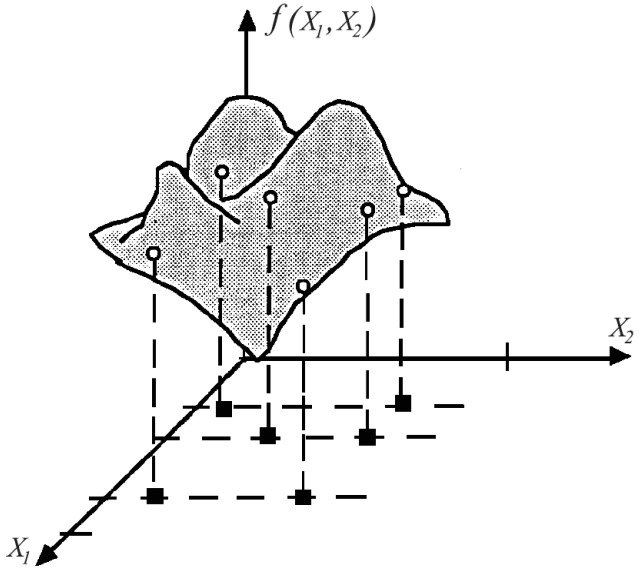

Рис. 4.6. Распределение процессоров

Когда мы комбинируем многошаговую схему с одномерным параллельным алгоритмом, в котором условие остановки отлично от выполнения фиксированного одинакового числа итераций, тогда времена выполнения испытаний (вычислений минимизируемой функции) в параллельной итерации на i-м уровне рекурсии ($1 \leqslant i \leqslant N-1$) будут неизбежно различны, даже если время расчета целевой функция $\varphi(y)$ не зависит от аргументов. Напомним, что вычисление целевой функции на всех уровнях, кроме последнего, состоит в решении новой оптимизационной подзадачи.

Пусть на i-м уровне рекурсии применяется синхронная схема организации параллельной итерации одномерного алгоритма, когда итерация считается завершенной только после проведения всех π_i испытаний. В этом случае по меньшей мере $\prod\limits_{j=i+1}^{N} \pi_i$ процессоров, которые уже завершили решение своих подзадач, будут простаи-

вать до тех пор, пока не завершится решение остальных подзадач i-го уровня. Это означает, что для повышения производительности процесса решения в многошаговой схеме необходимо использовать асинхронные параллельные алгоритмы глобальной оптимизации, в которых вычислительные ресурсы используются сразу же по их освобождении.

4.3.1. Вычислительные эксперименты

Чтобы проиллюстрировать теоретические положения, приведем результаты численного решения нескольких многомерных многоэкстремальных задач. Будем сравнивать три метода.

1. АИПМ — асинхронный многомерный информационный параллельный алгоритм с использованием многошаговой схемы редукции размерности, в котором для решения одномерных подзадач применяется асинхронный одномерный алгоритм, предложенный в [65].

2. АГПМ — алгоритм глобального поиска с использованием многошаговой схемы редукции размерности, где решение одномерных подзадач осуществляется базовым информационно-статистическим алгоритмом АГП.

3. ПСИМ — параллельный синхронный информационный алгоритм с использованием многошаговой схемы, в котором в качестве метода для решения одномерных подзадач применяется АГПВТ.

Для двух первых серий экспериментов введем следующие допущения.

1. Время вычисления значений $\varphi(y)$ одинаково для всех допустимых точек. Это время может быть использовано как некая «единица», или «квант» времени, во время поиска. Заметим, что это допущение является наихудшим случаем для асинхронного метода, поскольку оно фиксирует наиболее предпочтительную ситуацию для синхронного распараллеливания.

2. Время вычисления величины $\varphi(y)$ много больше времени, которое тратится на реализацию вычислительных операций алгоритмической схемы.

Такой подход позволяет абстрагироваться от архитектуры параллельной вычислительной системы, на которой проводится эксперимент, и оценить эффективность решающих правил алгоритмов «в чистом виде». В связи с этим вычислительные эксперименты были реализованы на однопроцессорном персональном компьютере с помощью программной системы, имитирующей параллелизм АИПМ и ПСИМ.

Обозначим через $T_A(\pi) = T_A(\pi_1, \pi_2, \ldots, \pi_N)$ время (измеряемое в вычислительных квантах) решения задачи по многошаговой схеме редукции с использованием для решения одномерных задач асинхронного алгоритма при использовании вектора распараллеливания (4.29), $T_S(\pi) = T_S(\pi_1, \pi_2, \ldots, \pi_N)$ — время решения задачи в аналогичных условиях с использованием синхронного алгоритма, $T_1 = T_S(1, 1, \ldots, 1) = T_A(1, 1, \ldots, 1)$ — время решения задачи с использованием последовательного метода, которое просто равно количеству вычисленных значений функции $\varphi(y)$.

Обозначим также через K_A и K_S количество испытаний, выполненных асинхронным и синхронным методами соответственно.

Используя введенные величины, мы можем определить соответствующие ускорения во времени, достигаемые за счет параллелизма:

- $u_S(\pi) = T_1/T_S(\pi)$ — ускорение синхронного алгоритма,
- $u_A(\pi) = T_1/T_A(\pi)$ — ускорение асинхронного алгоритма.

В качестве меры относительного сравнения эффективности синхронизма и асинхронизма можно ввести величину $u(\pi) = u_A(\pi)/u_S(\pi)$.

Заметим, что вследствие допущения 1 в случае $\pi = (1, 1, \ldots, \pi_N)$ имеем $u_A(\pi) = u_S(\pi)$ и $u(\pi) = 1$.

В первом случае проводилась максимизация 100 двумерных функций из класса (2.103). Все эксперименты были выполнены с точностью условия остановки $\varepsilon = 0{,}01$ и параметром $r = 2$ для решения всех одномерных подзадач (4.16). Все одномерные поиски стартовали с начальных точек $\{0{,}2,\ 0{,}3,\ 0{,}7,\ 0{,}8\}$.

Для исключительно последовательного метода АГПМ среднее число испытаний, полученных в результате максимизации 100 функций класса (2.103) (которое в этом случае может рас-

сматриваться как мера среднего времени поиска), было равно $T_1 = 415{,}08$. Результаты применения параллельных методов показаны в таблице 4.1.

Таблица 4.1. Сравнение параллельных алгоритмов на тестовом классе двумерных функций

q	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
K_A	409,46	397,96	419,41	456,00	510,60
K_S	413,89	409,24	424,08	458,60	514,20
T_A	204,73	99,49	69,90	57,00	51,06
T_S	234,52	114,63	77,79	62,46	56,89
u_A	2,03	4,17	5,94	7,28	8,13
u_S	1,77	3,62	5,34	6,64	7,30
u	1,15	1,15	1,11	1,10	1,23
q	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
K_A	427,23	429,18	432,90	472,20	529,95
K_S	432,40	426,16	433,83	469,44	527,90
T_A	123,41	71,53	48,10	39,35	35,33
T_S	174,38	84,90	55,95	44,63	40,62
u_A	2,91	5,80	8,63	10,55	11,75
u_S	2,38	4,89	7,42	9,30	10,22
u	1,22	1,19	1,16	1,13	1,15
q	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
K_A	459,88	447,12	469,32	494,40	565,20
K_S	458,09	459,38	466,44	502,08	570,70
T_A	114,97	55,89	39,11	30,90	28,26
T_S	144,45	71,38	46,58	36,92	33,69
u_A	3,61	7,43	10,61	13,43	14,69
u_S	2,87	5,82	8,91	11,24	12,32
u	1,26	1,28	1,19	1,19	1,19

Следующая серия экспериментов состояла в минимизации многомерной функции из [58]:

$$\varphi(y) = \frac{\pi}{N}\left\{10\sin^2(\pi y_1) + (y_N - 1)^2 + \right.$$
$$\left. + \sum_{i=1}^{N-1}\left[(y_i - 1)^2\left(1 + 10\sin^2(\pi y_{i+1})\right)\right]\right\}, \quad (4.30)$$

где $-2 \leqslant y_i \leqslant 4$, $1 \leqslant i \leqslant N$ для размерностей $N = 3, 4, 5$. Во всех экспериментах использовалась точность $\varepsilon = 0{,}12$ и параметр надежности $r = 2$. На начальной стадии поиска во всех одномерных подзадачах (4.16) первые четыре испытания проводятся в точках $\{-1{,}2,\ -0{,}5,\ 0{,}0,\ 2{,}2\}$. Результаты работы параллельных алгоритмов приведены в таблице 4.2.

При проведении следующих экспериментов предположим, что время вычисления значения функции зависит от точки испытания. Будем рассматривать задачу (4.30) при $N = 3$; установим точность метода $\varepsilon = 0{,}01$, параметр надежности $r = 2$, и во всех одномерных подзадачах (4.16) первые четыре испытания будем проводить в точках $\{-1{,}2,\ -0{,}5,\ 0{,}0,\ 2{,}2\}$. Используем два вида зависимости времени вычисления значения функции от точки испытания $t(y)$. В эксперименте А

$$t_1(y) = \left\lfloor 999\frac{\varphi(y) - \varphi_{\min}}{\varphi^{\max} - \varphi_{\min}}\right\rfloor,$$

где $\lfloor x \rfloor$ обозначает целую часть x, $\varphi_{\min} = 0$, $\varphi^{\max} = 185$; в эксперименте В

$$t_2(y) = 1000 - t_1(y).$$

Из таблицы 4.3 видно, что оба метода делают лишь небольшое число избыточных испытаний, причем асинхронный алгоритм работает быстрее синхронного.

Рассмотрим теперь некоторый последовательный характеристический алгоритм и построим его обобщения на синхронный и асинхронный случаи, а затем сравним эффективность данных вариантов по показателям ускорения. Данный подход позволяет оценить потенциал различных схем распараллеливания решающего правила алгоритма. В рамках данного подхода для базового информационно-статистического алгоритма глобального поиска (2.36)–(2.38)

Таблица 4.2. Сравнение параллельных алгоритмов на функциях разной размерности

$N = 3$	q	$(1, 1, 1)$	$(2, 2, 2)$	$(3, 3, 3)$	$(4, 4, 4)$
	K_A	7 791	7 056	7 884	10 368
	K_S	7 791	6 772	7 548	10 836
	T_A	7 791	882	292	162
	T_S	7 791	1 013	390	216
	u_A	1,00	8,83	26,68	48,09
	u_S	1,00	7,69	19,98	36,07
	u	1,00	1,15	1,34	1,33
$N = 4$	q	$(1, 1, 1, 1)$	$(2, 2, 2, 2)$	$(3, 3, 3, 3)$	$(4, 4, 4, 4)$
	K_A	105 694	136 576	165 807	201 216
	K_S	155 694	130 168	152 895	232 682
	T_A	155 694	8 536	2 047	786
	T_S	155 694	10 658	3 062	1296
	u_A	1,00	18,24	76,06	198,08
	u_S	1,00	14,61	50,85	120,13
	u	1,00	1,25	1,50	1,65
$N = 5$	q	$(1, 1, 1, 1, 1)$	$(2, 2, 2, 2, 2)$	$(3, 3, 3, 3, 3)$	$(4, 4, 4, 4, 4)$
	K_A	3 111 771	2 635 872	3 258 144	4 786 176
	K_S	3 111 771	2 502 128	3 097 143	4 996 416
	T_A	3 111 771	82 371	13 408	4674
	T_S	3 111 771	112 485	24 119	7776
	u_A	1,00	37,78	232,08	665,76
	u_S	1,00	27,66	129,02	400,18
	u	1,00	1,36	1,80	1,67

были проведены вычислительные эксперименты на классах функций разной размерности. По результатам оптимизации 100 многоэкстремальных функций двух переменных, случайно выбранных из тестового класса (2.103), получены средние значения показателей $u_S(\pi)$ и $u_A(\pi)$, представленные на рис. 4.7.

Графики показывают зависимость ускорения (ось ординат) от вектора распараллеливания (4.29), варианты которого указаны на

Таблица 4.3. Сравнение параллельных алгоритмов при переменном времени вычисления значений функции

	Эксперимент A		Эксперимент B	
q	(2,2,2)	(3,3,3)	(2,2,2)	(3,3,3)
K_A	106 802	117 270	105 599	111 774
K_S	101 510	103 671	101 510	103 671
T_A	738 361	221 739	11 900 017	3 629 895
T_S	1 527 229	791 559	16 943 876	5 769 930
u	2,07	3,57	1,42	1,59

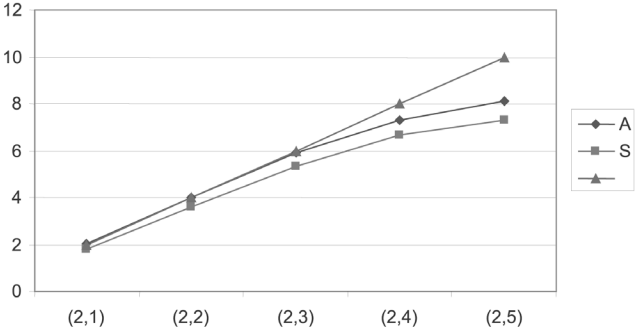

Рис. 4.7. Сравнительное ускорение синхронной и асинхронной схем на тестовом классе двумерных функций

оси абсцисс. Верхний график соответствует «идеальному» ускорению, средний график отражает ускорение асинхронной, нижний — синхронной схемы.

Аналогичный график представлен на рис. 4.8 для случая оптимизации многоэкстремальной пятимерной функции

$$\varphi(y) = \frac{\pi}{5}\left(\sin^2(\pi y_1) + 5(y_5 - 1)^2 + \sum_{i=1}^{4}\left((y_i - 1)^2\left(1 + \sin^2(\pi y_{i+1})\right)\right)\right) \tag{4.31}$$

с использованием для одномерной оптимизации метода ломаных (2.34), (2.35) с адаптивной оценкой константы Липшица (2.37), (2.38).

Как и в предыдущем случае, верхний график соответствует «идеальному» ускорению, средний и нижний графики отражают ускорение асинхронной и синхронной схем соответственно.

Для оценки эффективности асинхронизма по сравнению с синхронным вариантом на рис. 4.9 приводится зависимость отношения $u_S(\pi)/u_A(\pi)$ от вектора (4.29) на задаче (4.31).

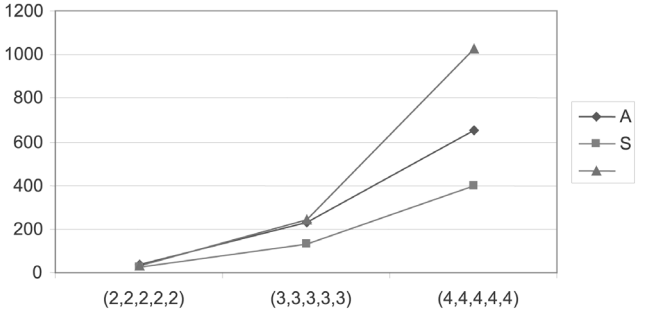

Рис. 4.8. Сравнительное ускорение синхронной и асинхронной схем на пятимерной задаче

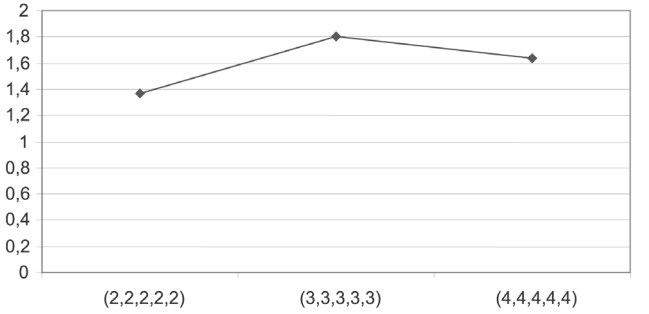

Рис. 4.9. Ускорение асинхронной схемы по сравнению с синхронной

4.4. Обобщенная адаптивная многошаговая схема редукции размерности

Следует отметить, что использование многошаговой схемы редукции размерности к построению многомерных методов обладает рядом недостатков. При использовании этой схемы точность в условии остановки, обеспечивающем прерывание решения любой вложенной одномерной задачи вида (4.16), должна быть задана заранее, и если точность окажется недостаточной, то решение задачи в целом придется повторить заново (с большей точностью). С другой стороны, использование завышенной точности может приводить к существенному увеличению длительности глобального поиска, и завершение вычислений до выполнения условия остановки (в силу исчерпания допустимого времени расчетов) в этом случае может привести к тому, что оценка искомого решения будет выполнена только в некоторой подобласти области D.

В этом разделе предлагается обобщение многошаговой схемы редукции размерности, позволяющей в значительной степени устранить отмеченные выше недостатки многошаговой схемы. Суть обобщения состоит в устранении принципа строгого соподчинения при решении порождаемых в рамках многошаговой схемы одномерных функций $\varphi^i(u_{i-1}, y_i)$, $1 \leqslant i \leqslant N$, из (4.16) — все эти задачи предлагается решать совместно.

Изложение предлагаемой обобщенной многошаговой схемы редукции размерности состоит в следующем (см., например, [4–7]).

4.4.1. Общее описание подхода

Перед изложением обобщенного подхода отметим повторно, что в рамках общепринятой многошаговой схемы порождаемые одномерные задачи решаются строго последовательно (см. раздел 4.1) — получаемая в результате иерархическая схема порождения и решения одномерных задач представлена на рис. 4.10.

Порождаемые в процессе оптимизации функции имеют строгую иерархическую структуру подчинения в виде дерева. Построение этого дерева происходит динамически в ходе вычислений. Вычисление значения функции $\varphi^i(u_{i-1}, y_i)$ уровня i, $1 \leqslant i \leqslant N$, в некоторой

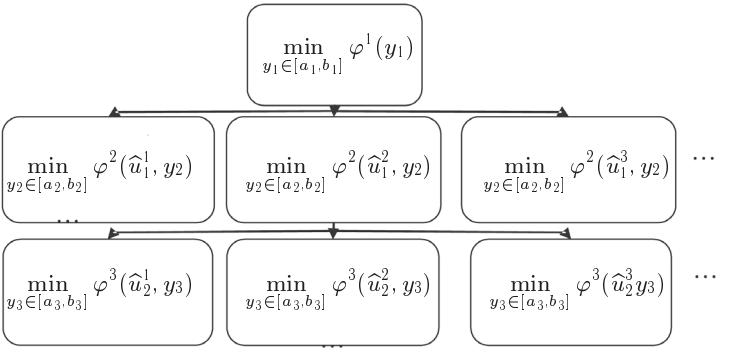

Рис. 4.10. Иерархическая схема порождения и решения одномерных задач в многомерной схеме редукции размерности

точке $y_i = \widehat{y}_i$ требует решения всех задач одного из поддеревьев уровня $i+1$. Листьями дерева задач являются значения целевой функции $\varphi(y)$, $y \in D$, вычисляемые в процессе глобального поиска.

Предлагаемое *обобщение многошаговой схемы* редукции размерности состоит в устранении строгой последовательности задач в соответствии с их иерархией в дереве задач. В рамках нового подхода (см., например, [7]):

1) для вычисления значения функции уровня i, $1 \leqslant i \leqslant N$ порождается новая задача уровня $i+1$, для которой *выполняется только одна итерация метода оптимизации*, после чего новая порожденная задача включается в множество уже имеющихся задач, подлежащих решению;

2) итерация глобального поиска состоит в выборе одной задачи из множества имеющихся задач, для которой и выполняется очередная итерация метода оптимизации; выбор задачи для выполнения итерации осуществляется в соответствии с тем или иным *правилом выбора задач*;

3) необходимые оценки минимально возможных значений оптимизируемых функций заменяются *текущими оценками этих значений* на основе поисковой информации, полученной в ходе вычислений.

Предлагаемый далее вариант будет именоваться *адаптивная многошаговая схема редукции размерности*. Строгое изложение данной схемы приводится в следующем пункте раздела.

4.4.2. Алгоритмическое описание

Введем еще одну систему обозначений одномерных функций, порождаемых в ходе оптимизации. Прежде всего, перенумеруем эти функции. Порядок нумерации может быть произвольным, возможный вариант — последовательная нумерация функций в порядке их порождения. Далее, порождаемые одномерные функции $\varphi^i(u_{i-1}, y_i)$ обозначим

$$f_i(v_i, x), \quad 1 \leqslant i \leqslant l, \tag{4.32}$$

где кортеж v_i содержит набор параметров, полностью определяющих вид одномерной функции и ее расположения в дереве задач:

$$v_i = <l_i,\ n_i,\ (\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_{n_i-1})>,$$

в котором l_i — номер функции, для вычисления значения которой была порождена f_i (номер функции-родителя); n_i — номер уровня функции в дереве задач, что соответствует также номеру варьируемой переменной, т. е. $x = y_{n_i}$; $(\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_{n_i-1})$ — список значений вектора конструктивных параметров y, зафиксированных при переходе к функции f_i (отметим, что количество этих значений также может быть определено при помощи величины n_i).

Обозначим текущее множество задач, сформированных для решения в процесс глобального поиска:

$$F_l = \{f_i(v_i, x),\ 1 \leqslant i \leqslant l\}. \tag{4.33}$$

Алгоритмическое описание процесса глобального поиска с использованием адаптивной обобщенной многошаговой схемы редукции размерности может быть представлено при помощи следующего набора правил.

Инициализация. Процедура инициализации процесса глобального поиска:

— установить начальные значения для параметров процесса глобального поиска (пустое множество задач, нулевая итерация поиска):

$$F_0 = \{\emptyset\}, \quad l = 0, \quad k = 0;$$

— создать первую задачу оптимизации:

$$f_1(<0, 1, (u_0)>, x);$$

Создание новой задачи с параметрами $\widehat{v} = <\widehat{l}, \widehat{n}, (\widehat{y}_1, \widehat{y}_2, \ldots \ldots, \widehat{y}_{\widehat{n}-1})>$. Для создания новой задачи необходимо:

— создать задачу $f_{l+1}(\widehat{v}, x)$;

— включить задачу в множество задач процесса глобального поиска:

$$F_{l+1} = F_l \cup f_{l+1}(\widehat{v}, x), \quad l = l + 1;$$

— выполнить инициализацию глобального поиска для функции $f_l(\widehat{v}, x)$. Значение функции $f_l(\widehat{v}, x)$, вычисленное в ходе инициализации, вернуть как результат выполнения правила.

Инициализация глобального поиска для функции $f_i(v_i, x)$. Процедура инициализации предполагает выполнение следующих действий:

— выбрать начальную точку $x = \widehat{x}^0$ алгоритма оптимизации для функции $f_i(v_i, x)$;

— вычислить значение функции $f_i(v_i, x)$ в точке $x = \widehat{x}^0$ и вернуть как результат выполнения правила.

Вычисление значения функции $f_i(v_i, x)$, $1 \leqslant i \leqslant l$, в точке $\widehat{x}$. Для вычисления значения функции необходимо:

— если функция, для которой необходимо вычислить значение функции, является исходной функцией $\varphi(y)$ (когда $n_i = N$), то вычислить значение $\varphi(\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_{N-1}, \widehat{x})$ и вернуть как результат выполнения правила;

— в противном случае (когда $n_i < N$), необходимо создать новую задачу с номером $l+1$ и параметрами $v_{l+1} = <i, n_i+1, (\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_{n_i-1}, \widehat{x})>$. Значение $f_{l+1}(v_{l+1}, \widehat{x})$, вычисленное в ходе создания функции, вернуть как результат выполнения правила.

Выполнение итерации глобального поиска. Для выполнения итерации $k+1$, $k \geqslant 1$, глобального поиска необходимо осуществить следующие действия:

— выбрать в соответствии с некоторым правилом задачу $f_i(v_i, x)$, $1 \leqslant i \leqslant l$, из множества F_l для выполнения для этой задачи очередной итерации глобального поиска;

— применить правила алгоритма оптимизации к задаче $f_i(v_i, x)$ и выбрать точку очередной итерации x^{k+1};

— вычислить значение функции $f_i(v_i, x)$ в точке x^{k+1};

— вычислить текущую оценку минимального значения функции $f_i(v_i, x)$ и при ее изменении по сравнению с предыдущими итерациями передать новую оценку задаче-родителю;

— проверить условие остановки и при его выполнении остановить процесс глобального поиска.

В завершение алгоритмического описания обобщенной многошаговой схемы отметим важные особенности нового подхода.

— В адаптивной многошаговой схеме все порождаемые задачи решаются совместно — выбор задачи для очередной итерации глобального поиска осуществляется в соответствии с тем или иным правилом выбора задач. Такой подход, с одной стороны, требует запоминания всей поисковой информации, получаемой в ходе вычислений, а с другой стороны, позволяет при необходимости увеличивать необходимую точность решения задач. Данная возможность способствует применению методики оптимизации с последовательно увеличивающейся точностью вычислений — так, начальная стадия оптимизации может быть выполнена с достаточно грубой точностью, что позволит получить оценки решения за сравнительно небольшое время вычислений, далее после анализа результатов точность может быть повышена и процесс глобального поиска может быть продолжен.

— Наличие правила выбора задачи для выполнения очередной итерации глобального поиска позволяет строить разнообразные и гибкие процедуры оптимизации (в частности, исходная многошаговая схема редукции размерности может быть получена как частный случай нового подхода) — этот аспект обобщенной многошаговой

схемы более подробно рассматривается в следующем разделе настоящей главы.

— Существование множества одновременно решаемых задач позволяет ставить вопрос об их параллельном решении и тем самым использовать для решения сложных оптимизационных задач весь потенциал современных высокопроизводительных многопроцессорных компьютеров.

Вместе с этим, как уже отмечалось, новый подход требует запоминания всей поисковой информации, получаемой в ходе вычислений (в отличие от исходной многошаговой схемы) — данный аспект приводит к необходимости разработки специальных методов для организации, хранения и эффективной обработки накапливаемых поисковых данных.

Еще один важный момент, отличающий предложенный подход от исходной многошаговой схемы, состоит в динамическом характере способа вычислений значений одномерных функций. Поясним более подробно данный аспект. Основой многошаговой схемы является принцип, в соответствии с которым вычисление значения любой одномерной функции $\varphi^i(u_{i-1}, y_i)$ (кроме функций конечного уровня N) требует решения задачи минимизации соответствующей одномерной функции следующего уровня (см. (4.16)):

$$\varphi_i(y_i) = \varphi^i(y_1, \ldots, y_i) = \min_{y_{i+1} \in [a_{i+1}, b_{i+1}]} \varphi^{i+1}(y_1, \ldots, y_i, y_{i+1}),$$
$$1 \leqslant i < N.$$

В новой обобщенной многошаговой схеме в качестве искомого значения функции $\varphi_i(y_i)$ используется *текущая оценка минимального значения* функции $\varphi_{i+1}(y_{i+1})$, определяемая на основе выполненных итераций глобального поиска. Используемые оценки могут уточняться в процессе вычислений, но тем не менее эти оценки могут существенно отличаться от значений, получаемых при решении задач (4.16). Более точно можно говорить, что при выполнении глобального поиска вычисления проводятся не с «точными» одномерными функциями $\varphi_i(y_i)$, $1 \leqslant i < N$, а с их приближенными аппроксимациями $\psi_i(y_i)$, $1 \leqslant i < N$. Следует отметить, что определенная приближенность вычисления значений одномер-

ных функций присуща и исходной многошаговой схеме редукции размерности (см. [42]).

Построение оценок минимальных значений одномерных функций $\varphi_{i+1}(y_{i+1})$ зависит от способов конкретизации представленных выше правил адаптивной многошаговой схемы. Для сохранения свойства липшицевости исходной оптимизируемой функции $\varphi(y)$ потребуем соблюдения *устойчивости* для правил адаптивной многошаговой схемы, в соответствии с которой применяемые правила при близких значениях варьируемых параметров должны порождать близкие оценки минимальных значений, т. е.

$$\forall\ i,\quad 1 \leqslant i < N,\quad \forall\ u_i', u_i'' \Rightarrow \exists\ k > k_0 : |y_{i+1}^*(k, u_i') - y_{i+1}^*(k, u_i'')| \leqslant \\ \leqslant \lambda_{i+1} \| u_i' - u_i'' \|, \quad (4.34)$$

где $y_{i+1}^*(k, u_i)$ — значение переменной y_{i+1}, соответствующей текущей оценке значения функции $\varphi_i(y_i)$ на итерации k глобального поиска, а $\|\cdot\|$ — норма в области поиска D из (1.8), определяющая расстояние между векторами u_i' и u_i''.

Соблюдение условий (4.34) обеспечивает выполнимость следующего утверждения (см. также теорему 4.1).

Теорема 4.2. *Пусть функция $\varphi(y)$, $y \in D$, является липшицевой с константой K. Тогда «приближенные» функции $\psi_i(y_i)$, $1 \leqslant i < N$, построенные в соответствии с адаптивной многошаговой схемой при выполнении условия устойчивости* (4.34), *также являются липшицевыми с константами K_i, где*

$$K_i = K \prod_{j=i+1}^{N} (1 + \lambda_j), \quad 1 \leqslant i < N,$$

где λ_i, $1 \leqslant i \leqslant N$, — из (4.34).

Доказательство. В соответствии с правилами построения, $\psi_N(v_N) = \varphi(y)$ является липшицевой с константой K.

Далее пусть для некоторого i, $1 \leqslant i < N$, функция $\psi_{i+1}(y_{i+1})$ является липшицевой с некоторой константой K_{i+1}. Покажем, что функция $\psi_i(y_i)$ также является липшицевой с константой K_i.

Оценим разность

$$\begin{aligned}
\left|\psi_i(y_i') - \psi_i(y_i'')\right| &= \left|\psi^{i+1}(u_i', y_{i+1}^*(u_i')) - \psi^{i+1}(u_i'', y_{i+1}^*(u_i''))\right| = \\
&= \left|\psi^{i+1}(u_i', y_{i+1}^*(u_i')) - \psi^{i+1}(u_i'', y_{i+1}^*(u_i')) + \right. \\
&\qquad \left. + \psi^{i+1}(u_i'', y_{i+1}^*(u'_i)) - \psi^{i+1}(u_i'', y_{i+1}^*(u_i''))\right| \leqslant \\
&\leqslant \left|\psi^{i+1}(u_i', y_{i+1}^*(u_i')) - \psi^{i+1}(u_i'', y_{i+1}^*(u_i'))\right| + \\
&\qquad + \left|\psi^{i+1}(u_i'', y_{i+1}^*(u'_i)) - \psi^{i+1}(u_i'', y_{i+1}^*(u_i''))\right| \leqslant \\
&\qquad \leqslant K_{i+1}\left\|u_i' - u_i''\right\| + K_{i+1}\left|y_{i+1}^*(u_i') - y_{i+1}^*(u_i'')\right|.
\end{aligned}$$

С учетом (4.34) можно получить

$$\begin{aligned}
\left|\psi_i(y_i') - \psi_i(y_i'')\right| &\leqslant K_{i+1}\left\|u_i' - u_i''\right\| + K_{i+1}\lambda_{i+1}\left\|u_i' - u_i''\right\| = \\
&= K_{i+1}(1 + \lambda_{i+1})\left\|u_i' - u_i''\right\|.
\end{aligned}$$

Тем самым функция $\psi_i(y_i)$ является липшицевой с константой

$$K_i = K_{i+1}(1 + \lambda_{i+1}).$$

Применяя индукцию по i, выводим справедливость сформулированной теоремы.

4.4.3. Многомерные характеристические алгоритмы глобального поиска на основе адаптивной многошаговой схемы редукции размерности

Представленное в предшествующем пункте описание адаптивной многошаговой схемы редукции размерности может служить алгоритмической основой для построения разнообразных алгоритмов многомерной многоэкстремальной оптимизации. Для получения определенного алгоритма оптимизации в рамках данного подхода необходимо конкретизировать следующие общие правила адаптивной многошаговой схемы:

— правило выбора задачи для выполнения очередной итерации глобального поиска;

— правило выбора точки начальной итерации поиска;

— правило выбора точки очередной итерации поиска;

— правило получения текущей оценки решения оптимизационной задачи;

— правило проверки условия остановки.

Детализация перечисленных правил может быть получена в рамках класса характеристических алгоритмов глобального поиска — см. раздел 2.3.

При последующем описании правил алгоритмической схемы используем, как и ранее, понятие поисковой информации ω_k (см. также (1.25)):

$$\omega_k = \{(x^i, z^i, l^i) : \ 1 \leqslant i \leqslant k\}, \tag{4.35}$$

представляющей точки x^i, $1 \leqslant i \leqslant k$, всех ранее выполненных итераций поиска, значения z^i, $1 \leqslant i \leqslant k$, минимизируемой функции в этих точках и номера задач l^i, $1 \leqslant i \leqslant k$, порожденных для вычисления значений z^i, $1 \leqslant i \leqslant k$. В рамках многошаговой схемы формирование поисковой информации целесообразно проводить отдельно для каждой решаемой задачи из семейства F_l из (4.33) — для указания принадлежности поисковой информации определенной задаче будем использовать номер этой задачи, т. е.

$$\omega_{k_j}(j), \quad 1 \leqslant j \leqslant l, \tag{4.36}$$

где k_j, $1 \leqslant j \leqslant l$, — количество элементов поисковой информации для задачи с номером j. Важно отметить, что имеющиеся в поисковой информации значения $z^i(j)$: $1 \leqslant i \leqslant k_j$ для функций уровня меньшего, чем N, могут изменяться, так как эти величины представляют собой оценки минимальных значений функций f_{l_i}, а эти оценки могут уточняться в ходе проведения вычислений.

С учетом введенных обозначений описание правил многомерных характеристических алгоритмов глобального поиска может быть выполнено следующим образом (см. описание схемы характеристических алгоритмов в разделе 2.3).

Правило выбора точки очередной итерации глобального поиска (решающее правило алгоритма) включает выполнение следующего набора действий:

— упорядочить поисковую информацию в порядке возрастания значений точек ранее выполненных итераций (новый порядок рас-

положения данных в поисковой информации отражается при помощи нижнего индекса):

$$x_1 \leqslant x_2 \leqslant \ldots \leqslant x_k;$$

— вычислить для каждого интервала (x_{i-1}, x_i), $1 < i \leqslant k$ величину $R(i)$, называемую далее *характеристикой интервала*;

— определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика $R(t)$, т. е.

$$R(t) = \max\{R(i)\colon 1 < i \leqslant k\},$$

(данную величину будем называть также *характеристикой задачи*);

— вычислить точку x^{k+1} очередной итерации глобального поиска в интервале с максимальной характеристикой в соответствии с некоторым правилом

$$x^{k+1} = s(t) \in (x_{t-1}, x_t).$$

Правило получения текущей оценки решения оптимизационной задачи. В качестве текущей оценки искомого решения оптимизационной задачи принимается наименьшее вычисленное значение минимизируемой функции, т. е.

$$z_k^* = \min\{z_i\colon 1 \leqslant i \leqslant k\}.$$

Напомним, что текущая оценка минимального значения передается задаче-родителю как значение функции; еще раз отметим, что оценка решения может уточняться в ходе вычислений в связи с расширением состава поисковой информации.

Правило проверки условия остановки. Условие остановки определяется правилом

$$x_t - x_{t-1} \leqslant \varepsilon,$$

где t — интервал с максимальной характеристикой, т. е. итерации глобального поиска прекращаются, если длина интервала (x_{t-1}, x_t), содержащего точку очередной итерации, не превышает заданной точности $\varepsilon > 0$.

Приведенное правило остановки допускает два варианта применения: «слабое» условие прекращает вычисления при выполнении правила остановки для любой задачи семейства F_l; «строгий» вариант условия осуществляет проверку правила остановки только для функции самого верхнего уровня $\varphi_1(y_1)$.

Напомним (см. описание схемы характеристических алгоритмов в разделе 2.3), что характеристики интервалов дают некоторую меру «важности» интервалов на предмет содержания в них минимальных значений оптимизируемой функции. Используя такое понимание характеристик, можно предложить правило выбора задачи для проведения очередной итерации глобального поиска.

Правило выбора задачи для проведения очередной итерации глобального поиска. Для проведения очередной итерации глобального поиска следует выбирать задачу с максимальной характеристикой (при выборе такой задачи должны рассматриваться все интервалы всех задач множества F_l).

4.4.4. Информационно-статистические алгоритмы глобального поиска в рамках адаптивной многошаговой схемы редукции размерности

Доведение представленной в пп. 4.4.2–4.4.3 алгоритмической схемы характеристически-представимых алгоритмов на основе адаптивной многошаговой редукции размерности до уровня конкретных методов глобального поиска требует доопределения правил вычисления характеристик интервалов и точки очередного испытания в интервале с максимальной характеристикой.

Как и ранее, для дальнейшего исследования применяются алгоритмы глобального поиска, разработанные в рамках информационно-статистического подхода. На основе предложенной алгоритмической схемы могут быть построены:

— базовый многомерный многошаговый алгоритм глобального поиска;

— многомерный многошаговый индексный алгоритм глобального поиска;

— многомерный многошаговый алгоритм глобального поиска с адаптивной локальной настройкой;

— многомерный многошаговый алгоритм глобального поиска с учетом значений первой производной минимизируемой функции и др.

Переформулирование исходных информационно-статистических алгоритмов в рамках адаптивной многошаговой схемы редукции размерности осуществляется достаточно естественным образом и далее не проводится из соображений краткости изложения материала. В качестве иллюстрации, например, для использования алгоритма глобального поиска (АГП) в рамках адаптивной многошаговой схемы достаточно положить:

— правило вычисления характеристики интервала (в соответствии с (2.36))

$$R(i) = m(x_i - x_{i-1}) + \frac{(z_i - z_{i-1})^2}{m(x_i - x_{i-1})} - 2(z_i + z_{i-1}), \qquad (4.37)$$

где m — текущая оценка константы Липшица;

— правило вычисления точки очередного испытания в интервале с максимальной характеристикой (в соответствии с (2.35))

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \frac{z_t - z_{t-1}}{2m}. \qquad (4.38)$$

Как показывается в [42], информационно-статистические алгоритмы глобального поиска обладают устойчивостью. Приведем в целях последующего исследования теоретическое утверждение из [42], в котором обосновывается данное свойство алгоритмов оптимизации.

Пусть информационно-статистический алгоритм глобального поиска применяется для минимизации одномерной функции $\psi(x)$, отличающейся от исходной функции $\varphi(x)$ на ограниченную вариацию δ, т. е.

$$|\varphi(x) - \psi(x)| \leqslant \delta. \qquad (4.39)$$

Соотношение между оценками глобального минимума функций $\varphi(x)$ и $\psi(x)$ определяется посредством следующей теоремы.

Теорема 4.3 [42]. *Пусть $x^k = \{x^1, \ldots, x^k\}$ — усечение (по точности ε) последовательности $\{x^k\}$, порождаемой АГП при ми-*

нимизации функции $\psi(x)$, удовлетворяющей неравенству (4.39), *где $\varphi(x)$ — липшицева функция (с константой K).*

Тогда

$$|\psi_k^* - \varphi(x^*)| \leqslant \Delta = \frac{\alpha K \varepsilon}{2} + \delta, \quad \alpha > 1, \tag{4.40}$$

если в алгоритме для оценки константы Липшица m справедливо неравенство

$$m > K\left[1 + \sqrt{\frac{\alpha + 4\beta(1+\beta)}{\alpha - 1}}\right],$$

где

$$\beta = \frac{2r\delta}{K\varepsilon(r-1)}.$$

Теорема 4.3 позволяет сделать вывод об устойчивости информационно-статистических алгоритмов в рамках адаптивной многошаговой схемы.

Теорема 4.4. *Для АГП в рамках адаптивной многошаговой схемы выполняется условие устойчивости* (4.34), *т. е.*

$$\forall\ i, \quad 1 \leqslant i < N, \quad \forall\ u_i', u_i'' \Rightarrow \exists\ k > k_0:$$
$$|y_{i+1}^*(k, u_i') - y_{i+1}^*(k, u_i'')| \leqslant \lambda_{i+1} \|u_i' - u_i''\|,$$

при

$$\lambda_{i+1} = \Delta / \|u_i' - u_i''\| \quad \text{и} \quad \delta = K\|u_i' - u_i''\|, \quad 1 \leqslant i < N,$$

где λ_{i+1}, $1 \leqslant i < N$, из (4.34), *δ из* (4.39), *Δ из* (4.40).

Доказательство. Справедливость данной теоремы следует из липшицевости функций $\varphi_i(y_i)$, $1 \leqslant i < N$, (теоремы 4.1 и 4.2) и из устойчивости информационно-статистических алгоритмов глобального поиска (теорема 4.3).

В теореме 4.3 об устойчивости информационно-статистических алгоритмов глобального поиска рассматривается вопрос о минимизации одной отдельной многоэкстремальной функции, в то время как в рамках многошаговой схемы происходит решение согласованного множества оптимизационных задач. Анализ подобной ситуации обеспечивает следующая теорема.

Теорема 4.5 [42]. *Пусть при любом i, $1 \leqslant i \leqslant N$, функции $\varphi_i(v_i)$ являются липшицевыми с константами K_i. Тогда для приближения ψ^*, получаемого при минимизации функции $\varphi(y)$, $y \in D$, с помощью АГП при использовании многошаговой схемы редукции, справедливо, что*

$$\lim_{\varepsilon_1 \to 0} \psi^* = \varphi^*,$$

если

$$\varepsilon_{i+1} = O(\varepsilon_i), \quad 1 \leqslant i < N,$$

и, кроме того, при прерывании минимизации по y_i выполняется неравенство

$$m_i \geqslant 2K_i + \theta, \quad 1 \leqslant i < N, \quad \theta > 0.$$

Утверждение теоремы 4.5 является теоретическим обоснованием применимости многошаговой схемы редукции. В случае же использования адаптивной схемы возможность применения данной теоремы является ограниченной. Одномерные задачи $f_i(u_i, x)$, $1 \leqslant i \leqslant l$, семейства F_l из (4.33) рассматриваются одновременно, и часть из них (с достаточно большими значениями минимизируемой функции $\varphi(y)$) могут быть никогда не решены с требуемой точностью ε (именно в этом состоит исходная идея повышения эффективности многошаговой схемы). Как результат, погрешность вычисления функций $\varphi_i(y_i)$ может оказаться значительной, и эта погрешность может оставаться неизменной даже при стремлении точности вычислений ε к нулю. Для разрешения данной проблемы можно отметить, что на самом деле нет необходимости точного вычисления функций $\varphi_i(y_i)$ на всех интервалах их определения, — необходимым требованием может служить лишь условие точного вычисления этих функций только в окрестности глобального минимума исходной функции $\varphi(y)$. Данное требование будет использовано далее в качестве теоретического обоснования адаптивной многошаговой схемы редукции.

Представим поисковую информацию $\omega_{k_j}(j)$, $1 \leqslant j \leqslant l$, из (4.35)–(4.36), получаемую в процессе глобального поиска для

функции $f_j(v_j, x)$, $1 \leqslant j \leqslant l$, в виде

$$\omega_{k_j}(j) = \{(y_{j,i}, z_{j,i}(\delta_{j,i}), l_{j,i}) \colon 1 \leqslant i \leqslant k_j\},$$

где $y_{j,i}$, $1 \leqslant i \leqslant k_j$, — точки выполненных итераций поиска, индекс j соответствует номеру задачи в семействе F_l, индекс i соответствует упорядоченному расположению точек итераций в поисковой информации, $\delta_{j,i}$, $1 \leqslant i \leqslant k_j$, — погрешности вычисления значений $z_{j,i}$, $1 \leqslant i \leqslant k_j$, т. е.

$$(z_{j,i} - f_{l_{j,i}}(v_{l_{j,i}}, y^*_{j,i})) \leqslant \delta_{j,i}. \tag{4.41}$$

Напомним (см. (4.34)), что $l_{j,i}$, $1 \leqslant i \leqslant k_j$, есть номера задач семействе F_l, которые были порождены для вычисления значений $z_{j,i}$, $1 \leqslant i \leqslant k_j$. Важно подчеркнуть также, что погрешность $\delta_{j,i}$, $1 \leqslant i \leqslant k_j$, для разных точек $z_{j,i}$, $1 \leqslant i \leqslant k_j$, может оказаться различной.

Возможность получения оценок глобального минимума исходной функции $\varphi(y)$, $y \in D$, определяется условиями следующей теоремы.

Теорема 4.6. *Пусть точка $\overline{y}$ — предельная точка последовательности $\{y^k\}$, порождаемой информационно-статистическим алгоритмом глобального поиска при использовании адаптивной многошаговой схемы редукции размерности для минимизации липшицевой с константой K функции $\varphi(y)$, $y \in D$. Тогда данная точка $\overline{y}$ является точкой глобального минимума функции $\varphi(y)$, если начиная с некоторого шага k_0 для оценки константы Липшица m из* (4.37) *справедливо неравенство*

$$m > 2K, \tag{4.42}$$

а для любой функции $f_j(v_j, x)$, $1 \leqslant j \leqslant l$, семейства F_l из (4.33) *такой, что значения ее зафиксированных переменных совпадают с координатами глобального минимума, т. е.*

$$\widehat{y}_i = y^*_i, \quad 1 \leqslant i < n_j, \quad (\widehat{y}_1, \ldots, \widehat{y}_{n_j-1}), \quad n_j \ \textit{из} \ (4.35),$$

выполняются следующие условия на погрешность вычисления значений функции

$$(\delta_{j,\tau}+\delta_{j,\tau-1}) \leqslant \big(f_{l_{j,\tau}}(v_{l_{j,\tau}},x_{j,\tau})+f_{l_{j,\tau-1}}(v_{l_{j,\tau-1}},x_{j,\tau-1})\big) - \\ - \big(2\varphi(y^*)+K(x_{j,\tau}-x_{j,\tau-1})\big), \quad (4.43)$$

где τ, $1<\tau\leqslant k_j$, *— номер интервала поисковой информации* $\omega_{k_j}(j)$, *в котором содержится значение координаты* n_j *точки глобального минимума, т. е.*

$$x_{j,\tau-1}\leqslant y^*_{n_j}\leqslant x_{j,\tau}. \quad (4.44)$$

Кроме того, любая другая предельная точка y' *последовательности* $\{y^k\}$ *является точкой глобального минимума функции* $\varphi(y)$.

Доказательство. Покажем сначала справедливость первой части утверждения теоремы. Предположим противное, что предельная точка $\overline{y}$ не является точкой глобального минимума функции $\varphi(y)$.

Из формульного выражения характеристики $R(i)$ из (4.37) и двухсторонней сходимости информационно-статистических алгоритмов следует, что для интервалов, в которых содержатся значения координат точки $\overline{y}$ выполняется выражение

$$\lim_{k\to 0} R(t(k)) = -4\varphi(\overline{y}), \quad (4.45)$$

где t — номер интервала, содержащего значение координат точки $\overline{y}$, а k — номер итерации глобального поиска. С другой стороны, для любого интервала τ, содержащего значение координаты точки глобального минимума y^*, с учетом (4.42) и (4.43), выполняется условие

$$R(\tau)=m(x_{j,\tau}-x_{j,\tau-1})+\frac{(z_{j,\tau}-z_{j,\tau-1})^2}{m(x_{j,\tau}-x_{j,\tau-1})}-2(z_{j,\tau}+z_{j,\tau-1})\geqslant \\ \geqslant m(x_{j,\tau}-x_{j,\tau-1})+\frac{\big(f_{l_{j,\tau}}(v_{l_{j,\tau}},x_{j,\tau})-f_{l_{j,\tau-1}}(v_{l_{j,\tau-1}},x_{j,\tau-1})\big)^2}{m(x_{j,\tau}-x_{j,\tau-1})}- \\ -2\big(f_{l_{j,\tau}}(v_{l_{j,\tau}},x_{j,\tau})+f_{l_{j,\tau-1}}(v_{l_{j,\tau-1}},x_{j,\tau-1})\big).$$

Поскольку функция $\varphi(y)$ удовлетворяет условию Липшица с константой K, то для указанного интервала справедливо неравенство

$$f_{l_{j,\tau}}(\upsilon_{l_{j,\tau}}, x_{j,\tau}) + f_{l_{j,\tau-1}}(\upsilon_{l_{j,\tau-1}}, x_{j,\tau-1}) - K(x_{j,\tau} - x_{j,\tau-1}) \leqslant 2\varphi(y^*),$$

которое, учитывая (4.42), позволяет получить оценку для характеристики интервала

$$R(\tau(k)) > -4\varphi(y^*). \tag{4.46}$$

Тем самым, учитывая, что $\varphi(y^*) \leqslant \varphi(\overline{y})$, полученные выражения (4.45) и (4.46) при достаточно больших значениях k противоречат использованному допущению — точка $\overline{y}$ не может быть предельной, так как характеристики интервалов, содержащих эту точку, являются меньшими, чем характеристики интервалов с точкой глобального минимума.

Выполненные рассуждения позволяют показать справедливость и второго утверждения теоремы. В самом деле, допустим противное, и пусть точка y^* не является точкой глобального минимума функции $\varphi(y)$. Но тогда, как показано ранее, выполняются соотношения (4.45) и (4.46). Поскольку точка $\overline{y}$ является предельной точкой последовательности $\{y^k\}$, то для сколько угодно большого номера $k > 1$, должна существовать итерация $q > k$ такая, что характеристика интервала с точкой $\overline{y}$ должна оказаться наибольшей. Учитывая (4.45) и (4.46), это может оказаться возможным только, если характеристика интервала с точкой y^* также будет стремиться к значению (4.45), т. е.

$$\lim_{k \to 0} R(\tau(k)) = -4\varphi(y^*) = -4\varphi(\overline{y}).$$

Это возможно, в свою очередь, только если точка y^* также будет предельной точкой последовательности $\{y^k\}$, что противоречит сделанному предположению. Тем самым второе утверждение теоремы также является справедливым.

4.4.5. Вычислительные эксперименты для оценки эффективности адаптивной многошаговой схемы редукции размерности

Пример 4.3. Проведем эксперимент для тестовой задачи из семейства (2.103). Для решения задачи использовался алгоритм гло-

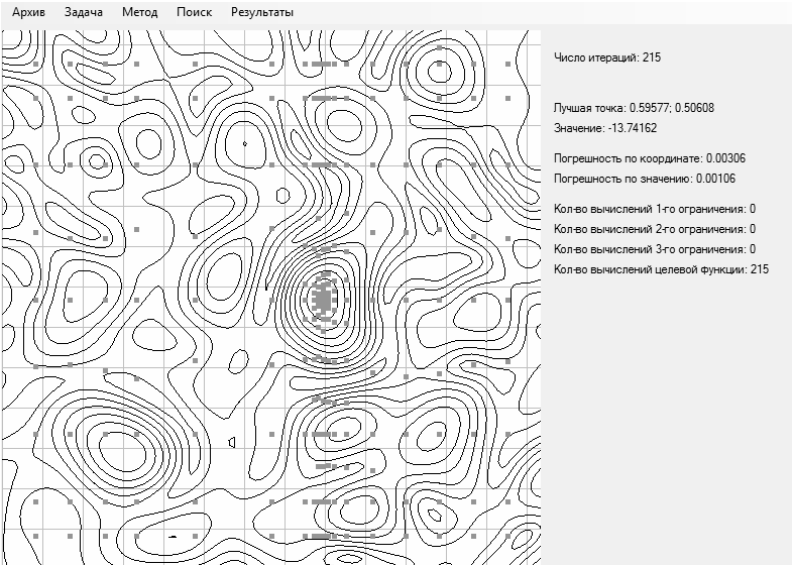

Рис. 4.11. Пример решения двумерной задачи оптимизации с использованием адаптивной многошаговой схемы редукции

бального поиска (АГП) при адаптивной многошаговой схеме редукции размерности. Параметр надежности равен $r = 2$ при точности поиска $\varepsilon = 0{,}01$ по каждой переменной. Всего было выполнено 215 итераций поиска, в результате которых была получена оценка $\varphi_k^* = -13{,}741$ с координатами $y_k^* = (0{,}595; 0{,}506)$. Сокращение объема вычислений по количеству итераций по сравнению с обычной многошаговой схемой составило 51%.

4.4.6. Операционные характеристики алгоритмов глобального поиска при адаптивной многошаговой схеме редукции размерности

Сравнение эффективности алгоритмов оптимизации на примерах решения отдельных тестовых задач, безусловно, не является достаточно убедительным способом доказательства превосходства того или иного метода. Более объективным подходом является проведе-

ние некоторого массового эксперимента и получение статистически достоверных результатов сравнения. Здесь, как и ранее, мы будем строить операционные характеристики изучаемых алгоритмов оптимизации (см. п. 2.2.2).

На рис. 4.12 представлены операционные характеристики следующих методов:

— алгоритм глобального поиска с редукцией размерности с использованием разверток;

— алгоритм глобального поиска с исходной многошаговой схемой редукции размерности;

— алгоритм глобального поиска с адаптивной многошаговой схемой редукции размерности.

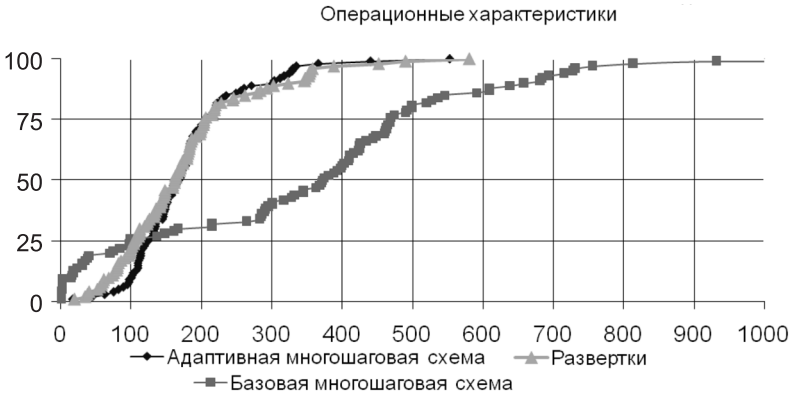

Рис. 4.12. Операционные характеристики алгоритмов глобального поиска

При подсчете операционных характеристик алгоритмов использовались следующие параметры: точность поиска $\varepsilon = 0{,}01$ по каждой координате, параметр надежности $r = 2{,}0$ для адаптивной многошаговой схемы и $r = 3{,}0$ для исходной многошаговой схемы и схемы редукции размерности с использованием разверток (для параметра надежности выбиралось минимально возможное значение, при котором успешно решались все задачи тестового класса).

Для построения операционных характеристик осуществлялось решение 100 тестовых задач из семейства (2.103).

На рис. 4.12 по оси ординат указывается количество успешно решенных задач, по оси абсцисс — количество итераций. Результаты выполненных экспериментов показывают, что при одинаковом числе выполненных итераций доля успешно решенных задач адаптивной многошаговой схемы редукции размерности существенно больше по сравнению с исходной многошаговой схемой. Среднее количество итераций по всему классу тестовых задач для адаптивной схемы в два раза меньше по сравнению с исходной многошаговой схемой. Кроме того, как можно заметить из рис. 4.12, операционная характеристика адаптивной схемы практически совпадает с операционной характеристикой алгоритма глобального поиска с редукцией размерности с использованием разверток.

4.5. Адаптивные параллельные вычисления для многомерной многоэкстремальной оптимизации

4.5.1. Централизованная схема параллельного глобального поиска

Рассмотренные в п. 1.2.2 подходы послужили основой для целого направления работ по разработке параллельных информационно-статистических алгоритмов глобального поиска (см., например, [40, 53, 65, 67]). Один из важных результатов, полученных в данном направлении, состоит в том, что для параллельных алгоритмов оптимизации оказалось возможным формулирование общей схемы характеристической представимости в полной аналогии с разработанной схемой для последовательных методов.

Получаемая схема организации параллельных вычислений при использовании характеристических параллельных алгоритмов многоэкстремальной оптимизации представлена на рис. 4.13. В рамках этой схемы один из имеющихся процессоров выделяется в качестве управляющего. Назначение управляющего процессора — выполнение представленной выше алгоритмической схемы характеристической представимости, вычисление и передачу свободным процессорам точек очередных испытаний, получение и обработку

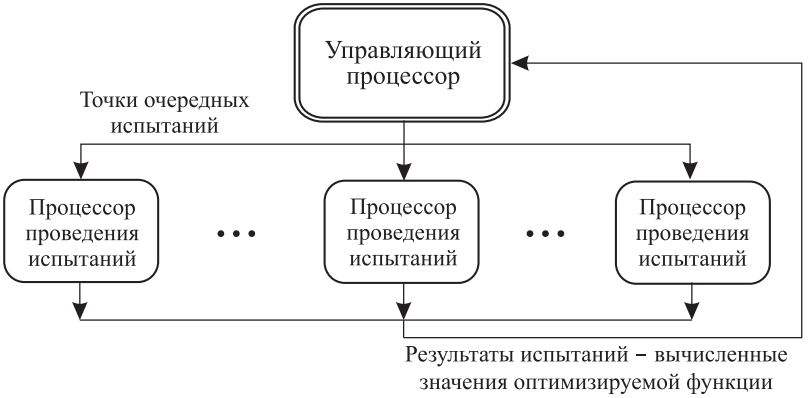

Рис. 4.13. Общая схема организации параллельных вычислений характеристических параллельных алгоритмов многоэкстремальной оптимизации

результатов испытаний от всех других процессоров системы. Все другие имеющиеся процессоры (кроме управляющего) являются исполнителями. Эти процессоры получают от управляющего процессора точки очередных испытаний, проводят вычисление значений оптимизируемой функции и передают результаты выполненных испытаний управляющему процессору. Подобная схема параллельных вычислений соответствует одному из широко используемых способов организации параллельных вычислений «менеджер-исполнители» (см., например, [10]), а происходящие в ходе вычислений процессы приема–передачи данных определяют одну из типовых топологий многопроцессорных систем в виде «звезды».

Отметим, что схема параллельных вычислений является асинхронной, т. е. каждый процессор-исполнитель взаимодействует с управляющим процессором совершенно независимо от других процессоров и получает точки, если есть свободный интервал, для проведения новых испытаний незамедлительно сразу после завершения своих вычислений для ранее полученных точек. Данный момент является важным, если время проведения испытаний может различаться (что и имеет место при частичной вычислимости

функционалов оптимизационной задачи с ограничениями, при использовании многошаговой схемы редукции размерности и др.).

Результаты численных экспериментов для полученного в рамках этой схемы параллельного многомерного информационно-статистического алгоритма глобального поиска при многошаговом способе редукции размерности уже были приведены ранее в п. 4.3.1.

4.5.2. Централизованная схема параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности

Представленная в п. 2.5.4 схема характеристической представимости асинхронных параллельных алгоритмов многоэкстремальной оптимизации может быть в полной мере применена и для информационно-статистических многомерных методов многоэкстремальной оптимизации с адаптивной многошаговой схемой редукции размерности. Небольшие необходимые изменения состоят в следующем.

Правила 2–5 схемы характеристической представимости должны быть выполнены для всего имеющегося набора решаемых задач F_l из (4.33).

Перед правилом 6 при определении точки нового испытания должно быть применено правило адаптивной многошаговой схемы редукции размерности по выбору задачи $f_i(u_i, x)$, $1 \leqslant i \leqslant l$, из множества F_l, для которой и должна выбираться точка очередной итерации глобального поиска.

Для оценки эффективности параллельных вычислений для получаемых в результате параллельных информационно-статистических многомерных методов многоэкстремальной оптимизации с адаптивной многошаговой схемой редукции размерности приведем результаты выполненных вычислительных экспериментов.

Пример 4.4. Для проведения вычислительных экспериментов использована тестовая задача из [37], широко используемая при оценке эффективности методов глобальной оптимизации. Минимизируемая функция имеет вид

$$\varphi(y) = \sum_{i=1}^{N} (y_i^2 - \cos(18y_i)),$$

а область поиска есть N-мерный гиперпараллелепипед

$$D = \{y \in R^N \colon -0.5 \leqslant y_i \leqslant 1.5,\ 1 \leqslant i \leqslant N\},$$

и N означает размерность решаемой задачи. Глобальный минимум функции располагается в точке $y^* = (0, 0, \ldots, 0)$ с наименьшим значением функции, равным $\phi(y^*) = -N$.

При выполнении расчетов параметр надежности алгоритма поиска принимался равным $r = 1{,}3$ при точности $\varepsilon = 0{,}01$ по каждой переменной. Количество процессоров изменялось от 6 до 82. Для моделирования решения вычислительно-трудоемких задач глобальной оптимизации в процедуру вычисления значений функции при проведении расчетов вводилась задержка длительностью 50 мкс.

Результаты вычислительных экспериментов приведены на рис. 4.14.

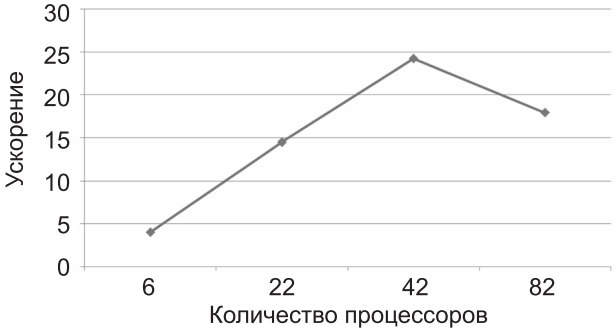

Рис. 4.14. Оценка эффективности централизованной схемы параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности

На рис 4.14 по горизонтали указывается количество процессоров, по вертикали – ускорение. Стоит отметить, что лучшее ускорение достигается при 42 процессорах, при дальнейшем увеличении числа процессоров ускорение снижается. Данная ситуация может быть объяснена тем, что управляющий процессор не успевает вычислить новые точки итераций для процессоров исполнителей.

4.5.3. Централизованная схема параллельного глобального поиска с несколькими управляющими процессорами для адаптивной многошаговой схемы редукции размерности

Существенным недостатком представленной выше централизованной схемы является управляющий процессор, который может оказаться узким местом. При увеличении количества процессоров исполнителей может наступить ситуация, когда управляющий процессор будет не успевать вычислять очередную точку итерации. Возможный выход из сложившегося положения — увеличение количества управляющих процессоров до некоторого числа q. Обозначим множество таких управляющих процессоров

$$P_q^* = \{p_1, p_2, \ldots, p_q\}.$$

При условии сохранения единства поисковой информации между этими q процессорами, выбор очередной итерации осуществляется следующим образом.

1. Выбрать q интервалов с максимальными характеристиками

$$I_q = \{i_1, i_2, \ldots, i_q\}.$$

2. Определить точки очередных испытаний — для процессора p_j, $1 \leqslant j \leqslant q$ точка испытания выбирается в интервале i_j.

Для обеспечения единства поисковой информации для всех центральных процессоров:

— перед каждой новой итерацией производится синхронизация очередей характеристик каждого из процессоров;

— процессор-исполнитель отправляет результаты своих вычислений каждому из управляющих процессоров.

Пример 4.5. Вычислительный эксперимент для разработанного варианта централизованной схемы параллельного глобального поиска, выполненный в полном соответствии с условиями расчетов из примера 4.4 (рис. 4.15).

Результаты выполненных экспериментов показывают, что централизованная схема параллельного глобального поиска с несколькими управляющими процессорами (в приведенных экспериментах использовалось два управляющих процессора) для адаптивной

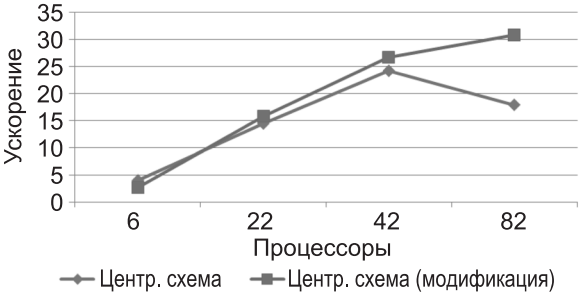

Рис. 4.15. Оценка эффективности централизованной схемы параллельного глобального поиска с несколькими управляющими процессорами для адаптивной многошаговой схемы редукции размерности

многошаговой схемы редукции размерности имеет более высокую эффективность по сравнению с централизованной схемой с единственным управляющим процессором.

4.5.4. Распределенная схема параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности

Другой возможный способ решения проблемы «перегрузки» управляющего процессора состоит в распределении нагрузки управляющего процессора между процессорами-исполнителями и тем самым превращения централизованной схемы параллельных вычислений в ту или иную более равноправную распределенных схему.

Все дальнейшее рассмотрение будет осуществляться только для информационно-статистических многомерных методов многоэкстремальной оптимизации с адаптивной многошаговой схемой редукции размерности.

1. Общая схема распределенных вычислений. Введем множество номеров задач семейства F_l из (4.33)

$$L = \{1, 2, \ldots, l\}, \tag{4.47}$$

и пусть для проведения вычислений имеется $p > 1$ процессоров. Распределим имеющиеся задачи между процессорами — данное

распределение можно зафиксировать при помощи соответствующего разделения множества L на подмножества:

$$\begin{gathered}\Pi = \{\pi_1, \pi_2, \ldots, \pi_p\},\\ \pi_i = \{j_s\colon j_s \in L,\ 1 \leqslant s \leqslant l_i\},\quad 1 \leqslant i \leqslant p,\\ \forall\ i \in L\ \ \exists\ j\colon i \in \pi_j,\quad \forall\ i, j \Rightarrow \pi_i \cap \pi_j = \{\varnothing\},\end{gathered} \tag{4.48}$$

где π_i, $1 \leqslant i \leqslant p$ — множество задач, распределенных для решения на процессоре i, $1 \leqslant i \leqslant p$.

Далее, для проведения вычислений на каждом процессоре может быть задействована адаптивная многошаговая схема глобального поиска для имеющегося на процессоре набора задач.

Построенная схема является децентрализованной — все процессоры работают параллельно и самостоятельно генерируют точки проведения испытаний. С другой стороны, между процессорами остается и взаимодействие — при получении в какой-то задаче новой улучшенной оценки минимального значения оптимизируемой функции эта оценка должна по-прежнему передаваться задаче-родителю. Поскольку решаемые задачи теперь распределены между процессорами, то передача получаемых оценок между задачами может приводить к сложным информационным зависимостям между процессорами. Понятно, что для минимизации таких зависимостей распределение решаемых оптимизационных задач между процессорами должно быть согласовано со структурой иерархической зависимости задач (см. рис. 4.10).

2. Структурная схема распределенных вычислений. Учет иерархической зависимости задач в полном виде приводит к сложной проблеме распределения задач между процессорами; при этом следует учитывать также, что структура иерархической зависимости задач формируется динамически в процессе вычислений и тем самым проблему соответствующего распределения задач между процессорами необходимо разрешать многократно.

Принимая во внимание отмеченные моменты, для решения проблемы распределения задач между процессорами может быть предложена простая и эффективная децентрализованная схема распределенных вычислений, в которой в достаточной степени учи-

тывается структура иерархической зависимости решаемых задач оптимизации (см., например, [7]).

Выделим в семействе решаемых задач оптимизации F_l из (4.33) задачи последнего уровня иерархии (задачи уровня N) — будем называть задачи подобного вида *терминальными* (нетерминальные задачи семейства будут именоваться далее *структурными*). В соответствии с выдвинутыми в начале настоящей главы предположениями, именно при решении таких задач выполняется наибольший объем вычислений, так как только в этих задачах осуществляется вычисление значений функций исходной решаемой задачи оптимизации. Понятно, что при распределении задач между процессорами эффективно должны быть решены вопросы распределения этих вычислительно-трудоемких терминальных задач.

Пусть номера терминальных задач образуют множество

$$L^N = \{i_j\colon i_j \in L,\ 1 \leqslant j \leqslant l^N\}. \tag{4.49}$$

Распределим терминальные задачи между процессорами и зафиксируем применяемое распределение по аналогии с (4.48) при помощи множества

$$\begin{gathered}\Pi^N = \{\pi_0, \pi_1, \pi_2, \ldots, \pi_p\},\\ \pi_0 = L \backslash L^N, \quad \pi_i = \{j_s\colon j_s \in L^N,\ 1 \leqslant s \leqslant l_i\}, \quad 1 \leqslant i \leqslant p,\\ \forall\ i \in L\ \ \exists\ j\colon i \in \pi_j, \ \ \forall\ i, j \Rightarrow \pi_i \cap \pi_j = \{\varnothing\}.\end{gathered} \tag{4.50}$$

В (4.50) каждое подмножество π_i, $1 \leqslant i \leqslant p$, определяет список терминальных задач (задач уровня N), распределенных для решения на процессоре i, $1 \leqslant i \leqslant p$. Подмножество π_0 содержит номера всех структурных задач (задач с уровнем меньшим, чем N). Вопрос распределения задач подмножества π_0 по процессорам необходимо разрешить дополнительно; в наиболее простом варианте для задач данного подмножества можно выделить дополнительный процессор. Получаемая в результате такого решения топология многопроцессорной системы становится близкой к варианту централизованной схемы распределенных вычислений (см. рис. 4.13). Тем не менее новый разработанный вариант организации вычислений по-прежнему остается децентрализованным, так как про-

цессоры с терминальными задачами самостоятельно, независимо от управляющего процессора, определяют точки проведения своих испытаний. Новая схема отличается также и систематическим характером информационных взаимодействий между процессорами. Процессоры с терминальными задачами пересылают получаемые оценки минимальных значений исходной оптимизируемой задачи только управляющему процессору. Управляющий процессор занимается обработкой только структурных задач без выполнения трудоемких вычислений значений функций исходной решаемой задачи оптимизации. При этом управляющий процессор может порождать терминальные задачи, и в этом случае данные задачи должны переправляться для выполнения на тех или иных процессорах с терминальными задачами.

Для завершения определения структурной схемы распределенных вычислений необходимо разрешить следующие вопросы.

- В какие моменты процесса вычислений управляющий процессор порождает новые структурные и терминальные задачи?
- Какие процессоры следует выбирать для размещения новых порождаемых терминальных задач?
- Требуется ли балансировка вычислительной нагрузки процессоров с терминальными задачами?

Для разработки способов решения поставленных вопросов введем дополнительные обозначения. Пусть:

$R(f_j),\ j \in L,$ — максимальная характеристика интервала для задачи $f_j \in F_l$ при текущей имеющейся поисковой информации $\omega_{k_j}(j)$ из (4.36); далее эта величина будет именоваться *характеристикой задачи*;

$R(\pi_j),\ \ \pi_j \in \Pi^N$,– максимальная характеристика задач, решаемых на процессоре i, $0 \leqslant i \leqslant p$; по аналогии эта величина будет именоваться *характеристикой процессора*.

Предлагаемые решения поставленных вопросов разрабатываемой новой децентрализованной схемы распределенных вычислений состоят в следующем (как и ранее, предлагаемые решения основываются на понимании характеристик интервалов как некоторой

меры важности интервалов на предмет содержания в интервалах искомого решения оптимизационной задачи).

1. **Правило порождения новых задач для управляющего процессора.** Порождение новых задач управляющим процессором осуществляется на итерациях поиска, когда характеристика этого процессора превышает характеристики всех остальных процессоров, т. е.

$$R(\pi_0) > R(\pi_i), \quad \pi_i \in \Pi^N, \quad i \neq 0. \tag{4.51}$$

Как результат, из приведенного правила следует, что процессоры с терминальными задачами должны передавать управляющему процессору свои характеристики $R(\pi_i)$, $\pi_i \in \Pi^N$.

2. **Правило распределения порождаемых терминальных задач**. Новые порождаемые терминальные задачи размещаются для решения на процессорах с наименьшими значениями характеристик, т. е.

$$j = \arg\min R(\pi_i), \quad \pi_i \in \Pi^N, \quad i \neq 0, \tag{4.52}$$

т. е. для выбора процессоров для решения новых порождаемых терминальных задач используется предположение, что процессоры с меньшими значениями характеристик в данный момент осуществляют решение наименее «важных» задач (задач, в которых наименьшая возможность содержания искомого решения оптимизационной задачи).

3. **Балансировка вычислительной нагрузки в структурной схеме распределенных вычислений.** Процесс глобального поиска на основе адаптивной многошаговой схемы редукции размерности отличается высокой динамикой: постоянно осуществляется порождение новых одномерных задач оптимизации, в результате чего количество решаемых на процессорах задач может существенно различаться. Кроме того, характеристики задач $R(f_j)$, $j \in L$, могут меняться достаточно быстро, что приводит к частой смене «важных» задач (задач с максимальными характеристиками), подлежащих решению в первую очередь. Отмеченные моменты означают, что в процессе глобального поиска обязательной является балансировка вычислительной нагрузки для процессоров. Данный аспект —

балансировка вычислений — является одним из наиболее важных и сложных проблем организации распределенных вычислений (см., например, [10]).

Далее последовательно будут рассмотрены два способа динамической балансировки. Первый из них — *схема глобальной балансировки* — является в некоторой степени идеальным, однако, данный подход затруднен для практического применения в силу высоких накладных расходов. Второй способ — *локальная попарная балансировка* — обладает меньшей вычислительной сложностью и может достигать эффективности глобального варианта балансировки.

Глобальная балансировка. Преобразуем множество терминальных задач L^N в список задач, упорядоченный по убыванию характеристик задач, т. е.

$$\vec{L}^N = \{i_1, i_2, \ldots, i_{l^N} \colon i_j \in L^N,\ 1 \leqslant j \leqslant l^N\}, \tag{4.53}$$

где i_1 — терминальная задача с максимальной характеристикой

$$i_1 = \arg\max R(f_j), \quad j \in L^N,$$

i_2 — терминальная задача со следующей по порядку максимальной характеристикой и т. д. Тогда для максимального соответствия схеме характеристической представимости асинхронных параллельных алгоритмов многоэкстремальной оптимизации (см. п. 2.5.4.) терминальные задачи из списка $\vec{L}^N$ из (4.53) должны распределяться между процессорами по циклической схеме, т. е. задача i_1 должна быть распределена на первый процессор, задача i_2 — на второй процессор и т. д. При исчерпании процессоров распределение задач далее можно продолжить циклически, т. е. задача $i_1 + p$ должна быть распределена на первый процессор, задача $i_2 + p$ — на второй процессор и т. д. В общем виде сформированное правило можно зафиксировать следующим образом: процессор j, $0 \leqslant j \leqslant p$, используется для решения терминальных задач с номерами i_j, $i_j + p$, $i_j + 2p$ и т. д.

Предложенное правило балансировки является практически оптимальным, поскольку в максимальной степени соответствует схеме характеристической представимости асинхронных параллельных

алгоритмов многоэкстремальной оптимизации, однако реальное применение данного правила затруднительно в силу существенных накладных расходов, необходимых для его реализации (высокая интенсивность передачи данных между процессорами). Далее будет предложен еще один способ балансировки, для выполнения которого требуется меньшие вычислительные затраты.

Локальная попарная балансировка. Для проведения балансировки данного вида представим все имеющиеся процессоры (кроме управляющего процессора) в виде набора непересекающихся пар процессоров:

$$\Sigma = \{\sigma_s = (i_s, j_s)\colon 1 \leqslant s \leqslant p/2\}. \tag{4.54}$$

В предлагаемом способе локальной попарной балансировки перераспределение терминальных задач осуществляется только между процессорами образованных пар. Балансировка состоит в следующем: для каждой пары процессоров (i, j) разбиения Σ выполняются действия:

— формируется объединенный список терминальных задач процессорной пары:

$$\pi = \pi_i \cap \pi_j = \{j_1, j_2, \ldots, j_\tau\colon \tau = l_i + l_j\};$$

список задач упорядочивается по убыванию характеристик задач:

$$\vec{\pi} = \{i_1, i_2, \ldots, i_\tau\colon i_j \in \pi,\ 1 \leqslant j \leqslant \tau\},$$

где i_1 — задача с максимальной характеристикой среди задач набора π; i_2 — задача со следующей по порядку максимальной характеристикой и т. д.;

— набор задач $\vec{\pi}$ разделяется между процессорами пары, для чего на первом процессоре пары оставляются задачи, находящиеся в списке $\vec{\pi}$ на позициях с нечетными номерами (т. е. это задачи i_1, i_3, и т. д.), а на втором процессоре пары — задачи, находящиеся в списке $\vec{\pi}$ на позициях с четными номерами.

Для конкретизации локальной попарной балансировки необходимо определить способ формирования процессорных пар. Два возможных подхода состоят в следующем.

Способ чет-нечетного группирования разработан по аналогии с методом чет-нечетной перестановки (*the odd-even transposition method*; см., например, [56]). Суть подхода состоит в том, что для группирования процессоров вводятся два разных правила выполнения — в зависимости от четности или нечетности номера итерации глобального поиска для группировки выбираются процессоры с четными или нечетными индексами соответственно, образование пар для выделяемых процессоров всегда осуществляется с их правыми соседними процессорами. Таким образом, на всех нечетных итерациях глобального поиска сравниваются пары процессоров

$$(1,2),\ (3,4),\ \ldots,\ (p-1,p)\ \ (\text{при четном } n),$$

а на четных итерациях обрабатываются процессоры

$$(2,3),\ (4,5),\ \ldots,\ (p-2,p-1).$$

Способ группирования процессоров на основе применения топологии гиперкуба. При тщательном анализе способа чет-нечетной группировки можно заметить, что данный подход недостаточно эффективен в ситуациях с существенной локальной несбалансированностью вычислительной нагрузки, т. е. в случаях, когда на ряде процессоров с последовательными номерами располагается или существенно большее количество задач, чем на всех остальных процессорах, или же сосредоточено большое количество задач с максимальными характеристиками. Поскольку в способе чет-нечетной группировки перемещение нагрузки на каждой итерации балансировки выполняется только на соседние процессоры, то устранение рассмотренных выше ситуаций пикового дисбаланса вычислительной нагрузки может потребовать большого количества итераций балансировки.

Устранение данного недостатка способа чет-нечетной группировки может состоять в объединении в пары при проведении балансировки процессоров, которые могут сильно различаться по номерам (и соответственно располагаться достаточно далеко при линейном расположении процессоров). Возможный способ для воплощения предложенного подхода состоит в представлении линейного расположения процессоров в виде топологии гиперкуба и в группиро-

вании процессоров, являющихся соседними уже в этой топологии. Итеративное описание нового способа группирования процессоров состоит в следующем:

— выполнение балансировки осуществляется в форме последовательно реализуемых этапов;

— на первой итерации каждого этапа имеющиеся процессоры делятся на две равные группы; первую группу образуют процессоры с номерами $1, \ldots, p/2$, вторую группу — процессоры с номерами $p/2+1, \ldots, p$; для балансировки пары формируются из процессоров, номера которых различаются на $p/2$ (т. е. в каждой паре располагается один процессор из первой группы, и один процессор из второй группы);

— на второй итерации каждого этапа балансировки описанное выше правило первой итерации применяется для каждой группы процессоров, сформированных на предшествующей итерации, т. е. каждая группа первой итерации снова разделяется на две части и в процессорные пары объединяются процессоры, номера которых различаются уже на величину $p/4$ и т. д. Всего на каждом этапе балансировки выполняется $\lfloor \log_2 p \rfloor + 1$ итерация.

Проведенные вычислительные эксперименты показывают, что предложенные схемы балансировки улучшают распределение нагрузки за счет больших дополнительных накладных расходов. Таким образом, вопрос балансировки вычислений требует проведения дополнительных исследований.

Глава 5

Параллельные вычисления для многомерной многоэкстремальной оптимизации при редукции размерности с использованием кривых Пеано

5.1. Общая схема подхода

Рассмотрим многомерную задачу глобальной оптимизации вида

$$\varphi(y^*) = \min\{\varphi(y)\colon y \in D,\ g_j(y) \leqslant 0,\ 1 \leqslant j \leqslant m\}, \tag{5.1}$$

$$D = \{y \in R^N\colon -2^{-1} \leqslant y_i \leqslant 2^{-1},\ 1 \leqslant i \leqslant N\}. \tag{5.2}$$

Данная постановка охватывает большой класс задач, так как любой гиперинтервал S,

$$S = \{s \in R^N\colon a_i \leqslant s_i \leqslant b_i,\ 1 \leqslant i \leqslant N\},$$

можно с помощью линейного преобразования координат привести к единичному гиперкубу.

Способ редукции размерности, речь о котором пойдет ниже, использует отображение многомерной области поиска на одномерный интервал с помощью *кривых Пеано*. Итальянским математиком Пеано было установлено, что может быть построено однозначное непрерывное отображение $y(x) = (y_1(x), y_2(x))$ отрезка $[0, 1]$ на единичный квадрат:

$$\{y \in R^2\colon -2^{-1} \leqslant y_1,\ y_2 \leqslant 2^{-1}\} = \{y(x)\colon 0 \leqslant x \leqslant 1\}.$$

Этот результат был обобщен на многомерный случай, т. е. было доказано существование кривых $y(x)$, заданных непрерывными координатными функциями $y_i(x)$, $x \in [0, 1]$, $1 \leqslant i \leqslant N$, однозначно отображающих отрезок $[0, 1]$ на N-мерный гиперкуб D, т. е.

$$D = \{y \in R^N\colon a_i \leqslant y_i \leqslant b_i,\ 1 \leqslant i \leqslant N\} = \{y(x)\colon 0 \leqslant x \leqslant 1\}.$$

Такие кривые, называемые также *развертками Пеано*, позволяют свести многомерную задачу условной минимизации в области D к одномерной задаче условной минимизации на отрезке $[0,1]$

$$\varphi\big(y(x^*)\big) = \min\{\varphi(y(x))\colon x \in [0,1],\ g_j(y(x)) \leqslant 0,\ 1 \leqslant j \leqslant m\}. \quad (5.3)$$

Рассматриваемая схема редукции размерности сопоставляет многомерной задаче с липшицевой минимизируемой функцией и липшицевыми ограничениями одномерную задачу, в которой соответствующие функции удовлетворяют равномерному условию Гёльдера [42, 67], т. е.

$$\begin{gathered} |g_j(y(x')) - g_j(y(x''))| \leqslant K_j |x' - x''|^{1/N}, \\ x',\, x'' \in [0,1], \quad 1 \leqslant j \leqslant m+1, \end{gathered} \quad (5.4)$$

где N — размерность исходной многомерной задачи, а коэффициенты K_j связаны с константами Липшица L_j исходной задачи соотношениями $K_j \leqslant 4L_j\sqrt{N}$.

Исходную задачу (5.1) мы будем рассматривать в предположении, что функции задачи $g_j(y)$, $1 \leqslant j \leqslant m+1$, могут быть определены лишь частично. Это означает, что они определены и вычислимы лишь на множествах

$$Q_1 = [0,1], \quad Q_{j+1} = \{x \in Q_j\colon g_j(y(x)) \leqslant 0\}, \quad 1 \leqslant j \leqslant m. \quad (5.5)$$

Эти условия позволяют ввести классификацию точек $x \in [0,1]$ в соответствии с числом $\nu = \nu(x)$ ограничений, вычисленных в этой точке. Этот индекс $\nu(x)$ также можно определить с помощью условий

$$g_j(y(x)) \leqslant 0, \quad 1 \leqslant j \leqslant \nu - 1, \quad g_\nu(y(x)) > 0, \quad (5.6)$$

где последнее неравенство несущественно, если $\nu = m+1$.

Поэтому проведение итерации в точке $x \in [0,1]$ сводится к последовательному вычислению величин $g_j(y(x))$, $1 \leqslant j \leqslant \nu = \nu(x)$, т. е. последующее значение $g_{j+1}(y(x))$ вычисляется лишь в том случае, когда $g_j(y(x)) \leqslant 0$. Процесс вычислений завершается либо в результате установления неравенства $g_j(y(x)) > 0$, либо в результате достижения значения $\nu(x) = m+1$.

Таким образом, описанная процедура, называемая *испытанием* в точке $x \in [0,1]$, автоматически приводит и к определению индекса ν этой точки, ибо условия (5.6) эквивалентны условиям

$$g_j(y(x)) \leqslant 0, \quad 1 \leqslant j \leqslant \nu - 1, \quad g_\nu(y(x)) > 0,$$

где последнее неравенство несущественно, если $\nu{=}m{+}1$. Пара значений

$$z = g_\nu(y(x)), \quad \nu = \nu(x),$$

порожденная испытанием в точке $x \in [0,1]$, называется *результатом испытания*.

Кроме точного решения y^* из (5.1) будем рассматривать также *ε-резервированное решение* задачи (5.1), определяемое условиями

$$\varphi(y_\varepsilon) = \min\{\varphi(y)\colon\ y \in D,\ g_j(y) \leqslant -\varepsilon_j,\ 1 \leqslant j \leqslant m\}, \tag{5.7}$$

где $\varepsilon_1, \ldots, \varepsilon_m$ — положительные числа («резервы» по каждому ограничению). Введем в рассмотрение также множество

$$Y_\varepsilon = \{y \in D\colon\ g_j(y) \leqslant 0,\ 1 \leqslant j \leqslant m,\ \varphi(y) \leqslant \varphi(y_\varepsilon)\} \tag{5.8}$$

всех допустимых точек задачи (5.1), которые не хуже (по значению целевой функции), чем ε-резервированное решение. Рисунок 5.1 иллюстрирует введенное определение.

Существование ε-резервированного решения задачи (5.1) можно интерпретировать как аналог условий регулярности в классических задачах нелинейного программирования. Следует также отметить прикладную роль этого условия. Даже если точное решение y^* известно, его практическая реализация возможна лишь как некоторое приближение к y^*. Поэтому важно, чтобы существовали близкие к y^* (по координатам и значению) точки из допустимой области Q. Существование ε-резервированного решения гарантирует наличие таких точек. При этом область Y_ε может играть роль множества подходящих приближений.

Перед тем, как перейти к подробному описанию разверток и алгоритма, рассмотрим некоторые свойства редуцированной задачи (5.3), которых не было в чисто одномерном случае. Во-первых, как уже говорилось, функции $g_j(y(x))$, $1 \leqslant j \leqslant m+1$, не будут липшицевыми. Они будут удовлетворять условию Гёльдера

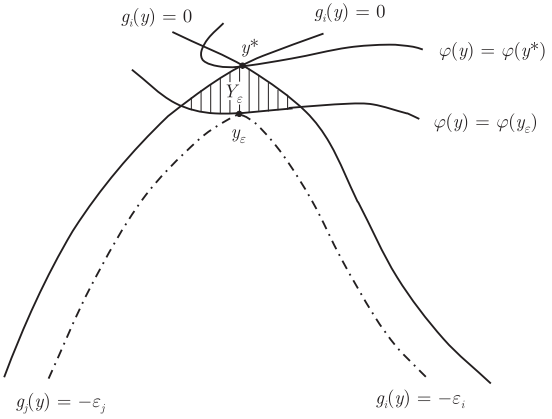

Рис. 5.1. ε-резервированное решение

и, следовательно, все длины интервалов, участвующие в правилах алгоритма поиска, должны быть заменены на длины в метрике

$$\rho(x', x'') = (|x' - x''|)^{1/N}, \quad x',\ x'' \in [0, 1].$$

Во-вторых, вообще говоря, множества (5.5), будучи прообразами N-мерных подмножеств с нелинейной границей $g_j(y) = 0$, $y \in D$, $1 \leqslant j \leqslant m$, не являются объединением конечного числа отрезков. Следовательно, учет ε-резервированных решений играет значительно более важную роль для сходимости в редуцированной одномерной задаче (5.3), чем в чисто одномерном случае.

Вопросы численного построения отображений типа кривой Пеано и соответствующая теория подробно рассмотрены в работах [42, 67]. Здесь же опишем основные шаги построения развертки, отображающей единичный отрезок вещественной оси $[0, 1]$ на гиперкуб D:

$$D = \{y \in R^N\colon\ -2^{-1} \leqslant y_i \leqslant 2^{-1},\ 1 \leqslant i \leqslant N\}.$$

1. Гиперкуб D, длина ребра которого равна 1, разделяется координатными гиперплоскостями на 2^N гиперкубов первого разбиения (с длиной ребра, равной 1/2), которые нумеруются числами z_1

от 0 до $2^N - 1$, причем гиперкуб первого разбиения с номером z_1 условимся обозначать через $D(z_1)$.

Далее, каждый гиперкуб первого разбиения также разбивается на 2^N гиперкубов второго разбиения (с длиной ребра, равной 1/4) гиперплоскостями, параллельными координатным осям и проходящими через серединные точки ребер гиперкуба, ортогональных к этим гиперплоскостям. При этом гиперкубы второго разбиения, входящие в гиперкуб $D(z_1)$, нумеруются числами z_2 от 0 до $2^N - 1$, причем гиперкуб второго разбиения с номером z_2, входящий в $D(z_1)$, обозначается через $D(z_1, z_2)$. Случай $N = 2$ изображен на рис. 5.2 для $m = 1$, $m = 2$.

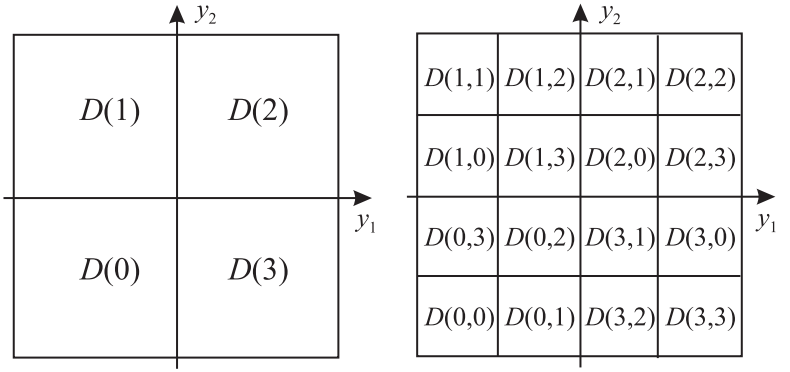

Рис. 5.2. Разбиение N-мерного гиперкуба

Продолжая указанный процесс, можно построить гиперкубы любого m-го разбиения с длиной ребра, равной $(1/2)^m$, которые обозначаются как $D(z_1, \ldots, z_m)$, причем $D(z_1) \supset D(z_1, z_2) \supset \ldots \supset$ $\supset D(z_1, \ldots, z_m)$ и $0 \leqslant z_j \leqslant 2^N - 1$, $1 \leqslant j \leqslant m$.

2. Теперь осуществим деление отрезка $[0, 1]$ на 2^N равных частей, каждую из которых также разделим на 2^N равных частей и так далее, причем элементы каждого разбиения нумеруются слева направо числами z_j от 0 до $2^N - 1$, где j — номер разбиения. При этом интервалы m-го разбиения обозначим как $d(z_1, \ldots, z_m)$, где, например, $d(z_1, z_2)$ обозначает интервал второго разбиения с но-

мером z_2, являющийся частью интервала $d(z_1)$ первого разбиения с номером z_1.

Можно отметить, что $d(z_1) \supset d(z_1, z_2) \supset \ldots \supset d(z_1, \ldots, z_m)$, и длина интервала $d(z_1, \ldots, z_m)$ равна $(1/2)^{mN}$. Предполагается, что интервал $d(z_1, \ldots, z_m)$ содержит свой левый конец. Он содержит правый конец тогда и только тогда, когда $z_1 = z_2 = \ldots = z_m = 2^N - 1$. Случай $N = 2$ изображен на рис. 5.3 для $m = 1$ и $m = 2$.

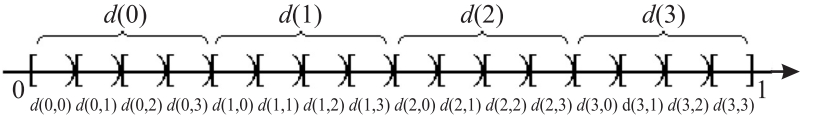

Рис. 5.3. Разбиение одномерного отрезка

3. Примем, что точка $y(x) \in D$, соответствующая точке $x \in [0, 1]$, при любом $m \geqslant 1$ содержится в гиперкубе $D(z_1, \ldots, z_m)$, если x принадлежит интервалу $d(z_1, \ldots, z_m)$, т. е.

$$x \in d(z_1, \ldots, z_m) \to y(x) \in D(z_1, \ldots, z_m).$$

Построенное соответствие $y(x)$ является однозначным.

4. Чтобы построенное отображение было еще и непрерывным, наложим требования на порядок нумерации гиперкубов каждого разбиения. Так как 2^{mN} центров $y(z_1, \ldots, z_m)$ гиперкубов m-го разбиения $D(z_1, \ldots, z_m)$ образуют равномерную ортогональную сетку в области D (причем шаг этой сетки по любой координате равен 2^{-m}), то можно установить следующую нумерацию узлов этой сетки. Пронумеруем слева направо по i все интервалы, составляющие m-е разбиение отрезка $[0, 1]$, т. е.

$$d(z_1, \ldots, z_m) = [x_i, x_{i+1}), \quad 0 \leqslant i < 2^{mN} - 1,$$

где через x_i обозначен левый конец интервала, имеющего номер i.

Будем считать, что центр гиперкуба $D(z_1, \ldots, z_m)$ имеет тот же номер i, что и соответствующий этому гиперкубу интервал $d(z_1, \ldots$ $\ldots, z_m)$, т. е.

$$y_i = y(z_1, \ldots, z_m), \quad 0 \leqslant i < 2^{mN} - 1.$$

При этом центры y_i и y_{i+1} соответствуют смежным гиперкубам, имеющим общую грань. Подробнее о нумерации гиперкубов см. в работах [42, 44].

Рассмотрим отображение $l(x)$ отрезка $[0,1]$ в гиперкуб D, определяемое выражениями

$$l(x) = y_i + (y_{i+1} - y_i)\left(\frac{w(x) - x_i}{x_{i+1} - x_i}\right),$$
$$x_i \leqslant w(x) \leqslant x_{i+1},$$
$$w(x) = x(1 - 2^{-mN}), \quad 0 \leqslant x \leqslant 1.$$

Образ любого подынтервала отрезка $[0,1]$ вида

$$\left[x_i\left(1 - 2^{-mN}\right)^{-1},\ x_{i+1}\left(1 - 2^{-mN}\right)^{-1}\right],$$
$$0 \leqslant i < 2^{mN} - 1$$

при соответствии $l(x)$ является линейным отрезком, соединяющим узлы y_i и y_{i+1}.

Таким образом, мы построили кусочно-линейную кривую $l_m(x)$, $0 \leqslant x \leqslant 1$, соединяющую узлы y_i, $0 \leqslant i < 2^{mN} - 1$, в порядке их нумерации. Эта построенная численно кривая является приближением к теоретической кривой Пеано с точностью, не хуже 2^{-m} по каждой координате (параметр m называется *плотностью раз-*

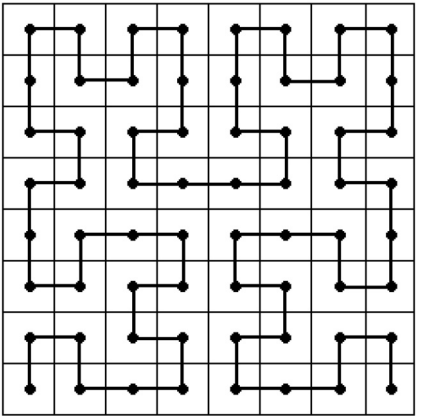

Рис. 5.4. Кусочно-линейная развертка

вертки). Для иллюстрации на рис. 5.4 изображен образ отрезка $[0, 1]$ при соответствии $l(x)$ для случая $N = 2$, $m = 3$ (узлы сетки отмечены темными кружками).

5.2. Многомерный обобщенный алгоритм глобального поиска

Рассматриваемая схема редукции размерности сопоставляет многомерной задаче с липшицевой минимизируемой функцией и липшицевыми ограничениями одномерную задачу, в которой соответствующие функции удовлетворяют равномерному условию Гёльдера (5.4). Как результат, все длины интервалов, участвующие в правилах алгоритма поиска, должны быть заменены на длины в новой метрике, в которой расстояние определяется выражением

$$\Delta_i = |x_i - x_{i-1}|^{1/N}.$$

Также при редукции размерности задачи с помощью развертки Пеано $y(x)$, $0 \leqslant x \leqslant 1$, *испытание* в точке $x^k \in [0, 1]$, осуществляемое на k-й итерации алгоритма, будет состоять в следующей последовательности действий.

1. Определить образ $y^k = y(x^k)$ при соответствии $y(x)$.

2. Вычислить величины $g_1(y^k), \ldots, g_\nu(y^k)$, где значения индекса $\nu \leqslant m$ определяются условиями

$$g_j(y^k) \leqslant 0, \quad 1 \leqslant j < \nu, \quad g_\nu(y^k) > 0, \quad \nu \leqslant m.$$

Выявление первого нарушенного ограничения прерывает испытание в точке y^k. В случае, когда точка y^k допустима, т. е. когда $y(x^k) \in \in Q_{m+1}$, испытание включает вычисление значений всех функций задачи, при этом значение индекса принимается равным величине $\nu = m + 1$. Пара значений

$$\nu = \nu(x^k), \quad z^k = g_\nu(y(x^k)) \tag{5.9}$$

является *результатом испытания* в точке x^k.

Таким образом, можно легко модифицировать исходный одномерный алгоритм из разд. 3.3 и получить на его основе *много-*

мерный обобщенный индексный алгоритм глобального поиска, к изложению которого мы сейчас переходим.

Первое испытание осуществляется в произвольной внутренней точке $x^1 \in (a, b)$. Выбор точки x^{k+1}, $k \geqslant 1$, любого последующего испытания определяется следующими правилами.

Правило 1. Перенумеровать точки $x^1, \ldots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т. е.

$$a = x_0 < x_1 < \ldots < x_i < \ldots < x_k < x_{k+1} = b, \tag{5.10}$$

и сопоставить им значения $z_i = g_\nu(y(x_i))$, $\nu = \nu(x_i)$, $1 \leqslant i \leqslant k$, из (5.9), вычисленные в этих точках; точки $x_0 = a$ и $x_{k+1} = b$ введены дополнительно (значения z_0 и z_{k+1} не определены) для удобства последующих обозначений.

Правило 2. Провести классификацию номеров i, $1 \leqslant i \leqslant k$, точек из ряда (5.10) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$I_\nu = \{i\colon 1 \leqslant i \leqslant k,\ \nu = \nu(x_i)\}, \quad 1 \leqslant \nu \leqslant m+1,$$

содержащих номера всех точек x_i, $1 \leqslant i \leqslant k$, имеющих индексы, равные одному и тому же значению ν. Граничные точки $x_0 = a$ и $x_{k+1} = b$ интерпретируются как имеющие нулевые индексы, и им сопоставляется дополнительное множество $I_0 = \{0, k+1\}$.

Определить максимальное значение индекса

$$M = \max\{\nu = \nu(x_i),\ 1 \leqslant i \leqslant k\}.$$

Правило 3. Вычислить текущие нижние границы

$$\mu_\nu = \max\left\{\frac{|z_i - z_j|}{(x_i - x_j)^{1/N}},\quad i, j \in I_\nu,\ i > j\right\} \tag{5.11}$$

для неизвестных констант Гёльдера K_ν функций g_ν, $1 \leqslant \nu \leqslant m+1$. Если множество I_ν содержит менее двух элементов или если μ_ν из (5.11) оказывается равным нулю, то принять $\mu_\nu = 1$.

Правило 4. Для всех непустых множеств I_ν, $1 \leqslant \nu \leqslant m+1$, вычислить оценки

$$z_\nu^* = \begin{cases} -\varepsilon_\nu, & \nu < M, \\ \min\{g_\nu(x_i)\colon i \in I_\nu\}, & \nu = M, \end{cases}$$

где вектор $\varepsilon_R = (\varepsilon_1, \dots, \varepsilon_m)$ является заданным вектором с неотрицательными компонентами (вектор резервов).

Правило 5. Для каждого интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k+1$, вычислить *характеристику* $R(i)$, где

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r_\nu^2 \mu_\nu^2 \Delta_i} - 2\frac{(z_i + z_{i-1} - 2z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_{i-1}) = \nu(x_i),$$

$$R(i) = 2\Delta_i - 4\frac{(z_i - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_i) > \nu(x_{i-1}),$$

$$R(i) = 2\Delta_i - 4\frac{(z_{i-1} - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_{i-1}) > \nu(x_i),$$

$$\Delta_i = (x_i - x_{i-1})^{1/N}.$$

Величины $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$, являются параметрами алгоритма. Подходящий выбор значений r_ν позволяет использовать произведение $r_\nu \mu_\nu$ как оценку константы Гёльдера K_ν, $1 \leqslant \nu \leqslant m+1$.

Правило 6. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i)\colon 1 \leqslant i \leqslant k+1\}. \tag{5.12}$$

Правило 7. Провести очередное испытание в серединной точке интервала (x_{t-1}, x_t), если индексы его концевых точек не совпадают, т. е.

$$x^{k+1} = \frac{x_t + x_{t-1}}{2}, \quad \nu(x_{t-1}) \neq \nu(x_t).$$

В противном случае провести испытание в точке

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \operatorname{sign}(z_t - z_{t-1}) \frac{1}{2r_\nu} \left[\frac{|z_t - z_{t-1}|}{\mu_\nu}\right]^N,$$
$$\nu = \nu(x_{t-1}) = \nu(x_t).$$

Описанные правила можно дополнить условием остановки, прекращающим испытания, если

$$(x_t - x_{t-1})^{1/N} \leqslant \varepsilon,$$

где t из (5.12) и $\varepsilon > 0$ — заданная покоординатная точность решения задачи.

Условия сходимости алгоритма. Данные условия являются частным случаем теоремы о сходимости, которая будет доказана в п. 5.3.3 для параллельного алгоритма при его использовании с единственной разверткой на одном процессоре.

Пример 5.1. В качестве иллюстрации рассмотрим задачу минимизации функции

$$\varphi(y_1, y_2) = -1{,}5y_1^2 \exp\{1 - y_1^2 - 20{,}25(y_1 - y_2)^2\} - \\ - [0{,}5(y_1 - 1)(y_2 - 1)]^4 \exp\{2 - [0{,}5(y_1 - 1)]^4 - (y_2 - 1)^4\}$$

в области поиска $0 \leqslant y_1 \leqslant 4$, $-1 \leqslant y_2 \leqslant 3$, при ограничениях

$$g_1(y_1, y_2) = 0{,}01[(y_1 - 2{,}2)^2 + (y_2 - 1{,}2)^2 - 2{,}25] \leqslant 0,$$
$$g_2(y_1, y_2) = 100[1 - (y_1 - 2)^2/1{,}44 - (0{,}5y_2)^2] \leqslant 0,$$
$$g_3(y_1, y_2) = 10[y_2 - 1{,}5 - 1{,}5\sin(6{,}283(y_1 - 1{,}75))] \leqslant 0.$$

На рис. 5.5 изображена область поиска — квадрат, где обозначены точки 1098 испытаний, полученные при применении индексного метода со следующими параметрами: параметр надежности $r_1 = \ldots = r_4 = 2$, ε-резервирование отсутствует (т. е. $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0$), точность решения $\varepsilon = 10^{-3}$, плотность построения развертки $m = 12$.

Отметим, что весьма часто в прикладных задачах оценка значений функций ограничений и целевой функции требует заметных вычислительных ресурсов. Результаты проведенного эксперимента подтверждают экономичность используемой индексной схемы учета ограничений, так как при этом значения функций g_1, g_2, g_3, $\varphi = g_4$ вычислялись соответственно $k_1 = 1098$, $k_2 = 623$, $k_3 = 392$ и $k_4 = 152$ раз. Если бы поиск решения проводился на равномерной сетке, то для достижения такой же точности $\varepsilon = 10^{-3}$ потребо-

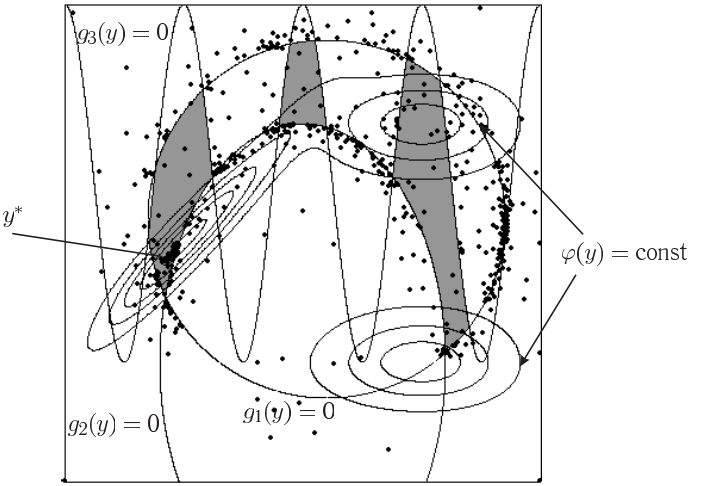

Рис. 5.5. Результаты решения задачи из примера 5.1

валось бы $k_1 = 1{,}6 \cdot 10^7$, $k_2 = 7 \cdot 10^6$, $k_3 = 3 \cdot 10^6$ и $k_4 = 1{,}4 \cdot 10^6$ вычислений значений функций g_1, g_2, g_3, $\varphi = g_4$ соответственно.

5.3. Параллельные многомерные многоэкстремальные методы на основе множественных кривых Пеано

5.3.1. Использование множественных отображений

Редукция многомерных задач к одномерным с помощью разверток имеет такие важные свойства, как непрерывность и сохранение равномерной ограниченности разностей функций при ограниченности вариации аргумента. Однако при этом происходит потеря части информации о близости точек в многомерном пространстве, так как точка $x \in [0, 1]$ имеет лишь левых и правых соседей, а соответствующая ей точка $y(x) \in R^N$ имеет соседей по 2^N направлениям. Как результат, при использовании отображений типа кривой Пеано близким в N-мерном пространстве образам y', y'' могут соответствовать достаточно далекие прообразы x', x'' на от-

резке $[0, 1]$. Данное свойство приводит к избыточным вычислениям, так как нескольким предельным точкам x', x'' последовательности испытаний, порождаемой индексным методом на отрезке $[0, 1]$, может соответствовать единственная предельная точка y в N-мерном пространстве.

Возможным способом преодоления этого недостатка является использование множественных отображений

$$Y_L(x) = \{y^0(x), y^1(x), \ldots, y^L(x)\} \tag{5.13}$$

вместо применения единственной кривой Пеано $y(x)$ [45, 67].

Рассмотрим семейство гиперкубов

$$D_l = \{y \in R^N\colon -2^{-1} \leqslant y_i + 2^{-l} \leqslant 3 \cdot 2^{-1},\ 1 \leqslant i \leqslant N\}, \quad 0 \leqslant l \leqslant L, \tag{5.14}$$

где гиперкуб D_{l+1} получается путем «сдвига» гиперкуба D_l вдоль главной диагонали на шаг 2^{-l} по каждой координате. На рис. 5.6 изображены гиперкубы $D_0, \ldots, D_3$ для случая $L = 3$.

Примем, что развертка $y^0(x)$ типа кривой Пеано отображает отрезок $[0, 1]$ на гиперкуб D_0 из (5.14), т. е.

$$D_0 = \{y^0(x)\colon x \in [0, 1]\}. \tag{5.15}$$

Тогда развертки $y^l(x) = \{y_1^l(x), \ldots, y_N^l(x)\}$, координаты которых определяются условиями

$$y_i^l(x) = y_i^{l-1}(x) + 2^{-l}, \quad 1 \leqslant i \leqslant N, \quad 1 \leqslant l \leqslant L, \tag{5.16}$$

отображают отрезок $[0, 1]$ на соответствующие гиперкубы D_l, $1 \leqslant l \leqslant L$ (на рис. 5.6 ломаной линией изображен образ отрезка $[0, 1]$, который получается при использовании развертки $y^0(x)$, $x \in [0, 1]$). Поскольку гиперкуб D из (5.2) входит в общую часть семейства гиперкубов (5.14) (граница гиперкуба D выделена на рис. 5.6), то, введя дополнительную функцию ограничения

$$g_0(y) = \max\{|y_i| - 2^{-1}\colon 1 \leqslant i \leqslant N\}, \tag{5.17}$$

исходный гиперкуб D можно представить в виде

$$D = \{y^l(x)\colon x \in [0, 1],\ g_0(y^l(x)) \leqslant 0\}, \quad 0 \leqslant l \leqslant L, \tag{5.18}$$

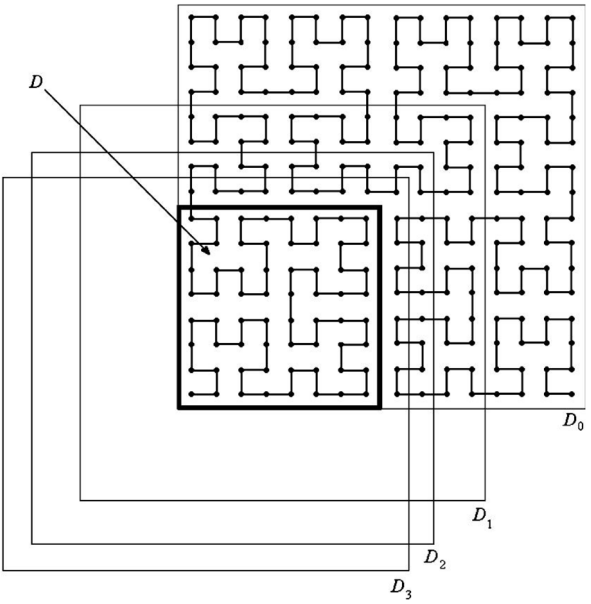

Рис. 5.6. Множественная развертка

т. е. $g_0(y) \leqslant 0$, если $y \in D$, и $g_0(y) > 0$ в противном случае. Следовательно, любая точка $y \in D$ имеет свой прообраз $x^l \in [0,1]$ при каждом соответствии $y^l(x)$, $0 \leqslant l \leqslant L$.

Таким образом, каждая развертка $y^l(x)$, $0 \leqslant l \leqslant L$, порождает свою задачу вида (5.3), характеризуемую своей расширенной по сравнению с D областью поиска D_l и дополненную ограничением с левой частью из (5.17):

$$\min\{\varphi(y^l(x)) \colon x \in [0,1],\ g_j(y^l(x)) \leqslant 0,\ 0 \leqslant j \leqslant m\}, \quad 0 \leqslant l \leqslant L. \tag{5.19}$$

При этом задачам (5.19) соответствуют область $Q_0 = [0,1]$ и области Q^l_{j+1}, $0 \leqslant j \leqslant m$, определяемые вторым выражением из (3.3) для соответствующих разверток $y^l(x)$. Применение множественного отображения определяет следующую связь окрестностей в многомерных и одномерных областях поиска.

Теорема 5.1. *Пусть y^* — произвольная точка из области поиска D, принадлежащая отрезку с концевыми точками y', $y'' \in D$, различающимися значениями единственной координаты, и пусть*

$$|y'_j - y''_j| \leqslant 2^{-p}, \quad y'_i = y''_i = y^*_i, \quad 1 \leqslant i \leqslant N, \quad i \neq j,$$

где p — целое число, $1 \leqslant p \leqslant L$, а j — номер координаты, значения которой для точек y^*, y', y'' являются различными. Тогда существует хотя бы одно соответствие $y^l(x)$, $0 \leqslant l \leqslant L$, и прообразы x^*, x', $x'' \in [0,1]$, такие что

$$y^* = y^l(x^*), \quad y' = y^l(x'), \quad y'' = y^l(x''),$$
$$\max\{|x'-x^*|,\ |x''-x^*|,\ |x'-x''|\} \leqslant 2^{-pN}.$$

Доказательство данной теоремы изложено в работе [67].

Замечание. Условия теоремы выделяют специфическую окрестность точки y^*. Эта окрестность включает в себя лишь такие точки, которые могут быть получены путем смещения y^* параллельно одной из осей координат на расстояние, не превышающее 2^{-p}. Изменяя значение j, $1 \leqslant j \leqslant N$, в условиях теоремы, можно выделить ближайших «соседей» точки y^* по N координатным направлениям. Согласно утверждению, близость точек в N-мерном пространстве по конкретному направлению будет отражена близостью их прообразов в одной из одномерных подзадач. Сохранение информации о близости точек приводит, во-первых, к более точной оценке констант Липшица, и, во-вторых, к увеличению характеристик интервалов, образы граничных точек которых являются близкими в N-мерном пространстве.

5.3.2. Организация параллельных вычислений

Использование множественных отображений позволяет решать исходную задачу (5.1) путем параллельного решения индексным методом $L+1$ задач вида (5.19) на наборе отрезков $[0,1]$. Каждая одномерная задача решается на отдельном процессоре. Результаты испытания в точке x^k, полученные конкретным процессором для решаемой им задачи, интерпретируются как результаты испытаний

во всех остальных задачах (в соответствующих точках x^{k0}, x^{k1}, ..., x^{kL}). При таком подходе испытание в точке $x^k \in [0,1]$, осуществляемое в s-й задаче, состоит в последовательности действий:

1) определить образ $y^k = y^s(x^k)$ при соответствии $y^s(x)$;

2) вычислить величину $g_0(y^k)$. Если $g_0(y^k) \leqslant 0$, то есть $y^k \in D$, то проинформировать остальные процессоры о начале проведения испытания в точке y^k (*блокирование* точки y^k);

3) вычислить величины $g_1(y^k), \ldots, g_\nu(y^k)$, где значения индекса $\nu \leqslant m$ определяются условиями

$$g_j(y^k) \leqslant 0, \quad 1 \leqslant j < \nu, \quad g_\nu(y^k) > 0, \quad \nu \leqslant m.$$

Выявление первого нарушенного ограничения прерывает испытание в точке y^k. В случае, когда точка y^k допустима, т. е. когда $y^s(x^k) \in \in Q_{m+1}$, испытание включает вычисление значений всех функций задачи, при этом значение индекса принимается равным величине $\nu = m+1$. Тройка

$$y^s(x^k), \quad \nu = \nu(x^k), \quad z^k = g_\nu(y^s(x^k)) \tag{5.20}$$

является результатом испытания в точке x^k;

4) если $\nu(x^k) > 0$, то есть $y^k \in D$, то определить прообразы $x^{kl} \in [0,1]$, $0 \leqslant l \leqslant L$, точки y^k и интерпретировать испытание, проведенное в точке $y^k \in D$ как проведение испытаний в $L+1$ точке

$$x^{k0}, \ x^{k1}, \ \ldots, \ x^{kL}, \tag{5.21}$$

с одинаковыми результатами

$$\nu(x^{k0}) = \nu(x^{k1}) = \ldots = \nu(x^{kL}) = \nu(x^k),$$
$$g_\nu(y^0(x^{k0})) = g_\nu(y^1(x^{k1})) = \ldots = g_\nu(y^L(x^{kL})) = z^k.$$

Проинформировать остальные процессоры о результатах испытания в точке y^k.

В случае если $\nu(x^k)$=0, то есть $y^k \notin D$, результат испытания относится только к s-й задаче.

Каждый процессор имеет свою копию программных средств, реализующих вычисление функций задачи, и решающее правило алгоритма. Для организации взаимодействия на каждом процессоре

создается $L+1$ очередь, в которые процессоры помещают информацию о выполненных итерациях в виде троек: точка очередной итерации, индекс и значение из (5.9), причем индекс заблокированной точки полагается равным -1, а значение функции в ней не определено. Связи между процессорами посредством очередей Q_{ls}, где l, $0 \leqslant l \leqslant L$, — номер передающего процессора и s, $0 \leqslant s \leqslant L$, — номер принимающего процессора, проиллюстрированы на рис. 5.7.

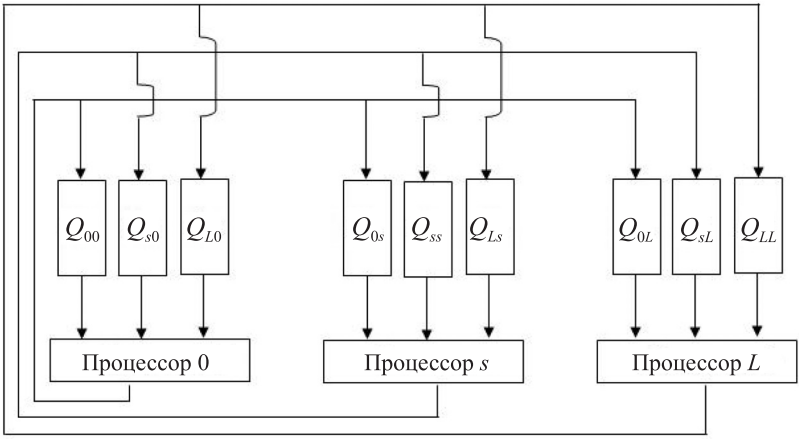

Рис. 5.7. Связи между процессорами

Предлагаемая схема не содержит какого-либо единого управляющего процессора, что увеличивает надежность выполняемых вычислений.

Решающие правила предлагаемого параллельного алгоритма в целом совпадают с правилами последовательного алгоритма (кроме способа проведения испытания). Однако для целей более простого освоения материала схема изложена полностью.

Алгоритм выбора точек итераций для всех процессоров одинаков. Начальная итерация осуществляется в заданной точке $x^1 \in (0,1)$ (начальные точки для всех процессоров задаются различными). Выбор точки x^{q+1}, $q \geqslant 1$, любого последующего испытания определяется следующими правилами.

Правило 1. Изъять из всех очередей, закрепленных за данным процессором, записанные для него результаты, включающие множество $Y_q = \{y^{q_i}\colon 1 \leqslant i \leqslant s_q\}$ точек итераций в области (5.2) и вычисленные в них значения индекса и величин из (5.9). Определить множество $X_q = \{x^{q_i}\colon 1 \leqslant i \leqslant s_q\}$ прообразов точек множества Y_q при соответствии $y^l(x)$.

Правило 2. Точки множества итераций

$$\{x_1\} \cup X_1 \cup \ldots \cup X_q$$

перенумеровать нижними индексами в порядке увеличения значений координаты

$$0 = x_0 < x_1 < \ldots < x_i < \ldots < x_k < x_{k+1} = 1, \tag{5.22}$$

где $k = 1 + s_1 + \ldots + s_q$, и сопоставить им значения $z_i = g_\nu(x_i)$, $\nu = \nu(x_i)$, $1 \leqslant i \leqslant k$, вычисленные в этих точках. При этом индекс блокированной точки x_i (т. е. точки, в которой начато проведение испытания другим процессором) полагается равным -1, т. е. $\nu(x_i) = -1$, значение z_i является неопределенным. Точки x_0, x_{k+1} введены дополнительно для удобства последующего изложения, индекс данных точек полагается равным -2, т. е. $\nu(x_0) = \nu(x_{k+1}) = -2$, а значения z_0, z_{k+1} являются неопределенными.

Правило 3. Провести классификацию номеров i, $1 \leqslant i \leqslant k$, точек из ряда (5.22) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$\begin{aligned} I_{-2} &= \{0, k{+}1\}, \\ I_{-1} &= \{i\colon 1 \leqslant i \leqslant k,\ \nu(x_i) = -1\}, \\ I_\nu &= \{i\colon 1 \leqslant i \leqslant k,\ \nu(x_i) = \nu\}, \quad 0 \leqslant \nu \leqslant m+1, \end{aligned}$$

содержащих номера всех точек x_i, $1 \leqslant i \leqslant k$, имеющих индексы, равные одному и тому же значению ν. Определить максимальное значение индекса

$$M = \max\{\nu = \nu(x_i), \quad 1 \leqslant i \leqslant k\}.$$

Правило 3. Вычислить текущие нижние границы

$$\mu_\nu = \max\left\{\frac{|z_i - z_j|}{(x_i - x_j)^{1/N}}\colon\ j < i,\ j \in I_\nu,\ i \in I_\nu\right\} \tag{5.23}$$

для относительных разностей функций g_ν, $1 \leqslant \nu \leqslant m+1$. Если множество I_ν содержит менее двух элементов или если μ_ν из (5.23) оказывается равным нулю, то принять $\mu_\nu = 1$. Из (5.23) следует, что оценки μ_ν являются неубывающими, начиная с момента, когда (5.23) порождает первое положительное значение μ_ν.

Правило 4. Для всех непустых множеств I_ν, $1 \leqslant \nu \leqslant m+1$, вычислить оценки

$$z_\nu^* = \begin{cases} -\varepsilon_\nu, & \nu < M, \\ \min\{g_\nu(x_i)\colon\ i \in I_\nu\}, & \nu = M, \end{cases} \tag{5.24}$$

где вектор $\varepsilon_R = (\varepsilon_1, \ldots, \varepsilon_m)$ является заданным вектором с неотрицательными компонентами (вектор резервов).

Правило 5. Для каждого интервала (x_{i-1}, x_i), $1 \leqslant i \leqslant k+1$, вычислить характеристику $R(i)$, где

$$\begin{aligned} R(i) &= \Delta_i + \frac{(z_i - z_{i-1})^2}{r_\nu^2 \mu_\nu^2 \Delta_i} - 2\frac{(z_i + z_{i-1} - 2z_\nu^*)}{r_\nu \mu_\nu}, \\ &\qquad \nu = \nu(x_{i-1}) = \nu(x_i), \\ R(i) &= 2\Delta_i - 4\frac{(z_i - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu(x_{i-1}) < \nu(x_i) = \nu, \\ R(i) &= 2\Delta_i - 4\frac{(z_{i-1} - z_\nu^*)}{r_\nu \mu_\nu}, \quad \nu = \nu(x_{i-1}) > \nu(x_i), \\ &\qquad \Delta_i = (x_i - x_{i-1})^{1/N}. \end{aligned} \tag{5.25}$$

Величины $r_\nu > 1$, $1 \leqslant \nu \leqslant m+1$, являются параметрами алгоритма.

Правило 6. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = \max\{R(i)\colon\ 1 \leqslant i \leqslant k+1\}. \tag{5.26}$$

Правило 7. Провести очередное испытание в серединной точке интервала (x_{t-1}, x_t), если индексы его концевых точек не совпа-

дают, т. е.

$$x^{q+1} = \frac{x_t + x_{t-1}}{2}, \quad \nu(x_{t-1}) \neq \nu(x_t). \tag{5.27}$$

В противном случае провести испытание в точке

$$x^{q+1} = \frac{x_t + x_{t-1}}{2} - \operatorname{sign}(z_t - z_{t-1}) \frac{1}{2r_\nu} \left[\frac{|z_t - z_{t-1}|}{\mu_\nu} \right]^N, \tag{5.28}$$
$$\nu = \nu(x_{t-1}) = \nu(x_t).$$

В случае если $\nu(x^{q+1}) = 0$, т. е. $y^q \notin D$, то результаты испытания занести только в очередь, закрепленную за данным процессором на нем самом. Если же $\nu(x^{q+1}) > 0$, т. е. $y^q \in D$, то результаты испытания занести во все очереди, закрепленные за данным процессором. Увеличив q на единицу, перейти к новой итерации.

Описанные правила можно дополнить условием остановки, прекращающим испытания, если

$$\Delta_t \leqslant \Delta,$$

где t из (5.26) и $\Delta > 0$ имеет порядок желаемой покоординатной точности в задаче.

5.3.3. Условия сходимости алгоритма

Достаточные условия сходимости рассматриваемого параллельного алгоритма могут быть сформулированы в виде следующей теоремы [45].

Теорема 5.2. *Пусть выполнены следующие условия.*

1. *Задача* (5.1) *имеет ε-резервированное решение y_ε из (5.7).*

2. *Каждая функция $g_j(y)$, $1 \leqslant j \leqslant m+1$, допускает липшицево с константой L_j продолжение $G_j(y)$ на всю допустимую область D из* (5.2), *т. е.*

$$g_j(y(x)) = G_j(y(x)), \quad x \in Q_j, \quad 1 \leqslant j \leqslant m+1, \tag{5.29}$$

где множества Q_j из (5.5) *и $y(x)$ — развертка из* (5.3).

3. *Параметры метода ε_ν,$1 \leqslant \nu \leqslant m$, используемые в правиле (5.24), имеют значения соответствующих координат вектора резервов ε_R из* (5.7).

4. *Если некоторым точкам из ряда* (5.22) *соответствуют значения индекса, равные* ν, *и при достаточно больших значениях* k *для соответствующей величины* μ_ν *из* (5.23) *справедливо неравенство*

$$r_\nu \mu_\nu > 8\sqrt{N}\, L_\nu, \quad 1 \leqslant \nu \leqslant m+1. \tag{5.30}$$

Тогда любая предельная точка $\overline{y}$ *последовательности испытаний* $\{y^k\}$, *порождаемой индексным алгоритмом для задачи* (5.1), *принадлежит множеству* Y_ε *из* (5.8) *и удовлетворяет условиям*

$$\varphi(\overline{y}) = \inf\{\varphi(y^k)\colon g_j(y^k) \leqslant 0,\ 1 \leqslant j \leqslant m,\ k = 1, 2, \ldots\} \leqslant \varphi(y_\varepsilon). \tag{5.31}$$

Прежде чем приступить к доказательству теоремы, сформулируем и докажем несколько вспомогательных утверждений.

Лемма 5.1 (о последовательности стягивающихся интервалов). *Пусть выполнены условия теоремы. Тогда для каждой предельной точки* $\overline{y}$ *последовательности* $\{y^k\}$ *существует бесконечная последовательность интервалов*

$$\{(x_{t-1}, x_t)\colon t = t(q^p)\}, \quad p = 1, 2, \ldots, \tag{5.32}$$

удовлетворяющих условиям

$$\overline{x} = \bigcap_{p=1}^{\infty} [x_{t-1}, x_t],$$

$$\lim_{p\to\infty} \Delta_t = 0, \tag{5.33}$$

$$R(t(q^p)) > 0, \quad p = 1, 2, \ldots, \tag{5.34}$$

$$\lim_{p\to\infty} R(t(q^p)) = 0, \tag{5.35}$$

где $q^1 < q^2 < \ldots$; $R(t)$, Δ_t *и* t — *соответственно из правил* (5.25), (5.26).

Доказательство леммы 5.1. Поскольку $\overline{y}$ является предельной точкой, должен существовать такой номер l, $0 \leqslant l \leqslant L$, что l-й процессор порождает последовательность точек x^{q^l}, сходящуюся к прообразу $\overline{x}^l$ точки $\overline{y} = y^l(\overline{x}^l)$. Следовательно, существует по-

следовательность номеров испытаний q^1, q^2, ..., удовлетворяющая условиям

$$\overline{x}^l \in [x_{t-1}, x_t], \quad t = t(q), \quad q \in \{q^p\}, \quad p = 1, 2, \ldots \ . \tag{5.36}$$

Эти условия отражают то обстоятельство, что точка $(q+1)$-го испытания на l-м процессоре попадет в интервал $[x_{t-1}, x_t]$, содержащий точку $\overline{x}^l$ на шаге q, т. е. $t = t(q)$.

Из (5.27), (5.28), (5.23) и условия (5.36) следует, что

$$\max\{x_t - x^{ql},\ x^{ql} - x_{t-1}\} \leqslant \gamma(x_t - x_{t-1}), \tag{5.37}$$

где $\gamma = (r+1)/2r < 1$, $r = \min\{r_\nu\colon 1 \leqslant \nu \leqslant m+1\} > 1$. При этом свойство (5.33) является следствием неравенства (5.37).

На каждом шаге $k \geqslant 1$ существует интервал (x_{i-1}, x_i): $i = i(k)$, на одном из концов которого достигается значение z_V^*, где V — максимальное значение индекса. Если при этом $\nu(x_{i-1}) \neq \nu(x_i)$, то, согласно правилам (5.25), $R(i) = 2\Delta_i > 0$. В случае когда $\nu(x_{i-1}) = \nu(x_i) = \omega$, из (5.23) следует оценка

$$z_i + z_{i-1} - 2z_V^* \leqslant \mu_V \Delta_i,$$

которая в сочетании с (5.25) дает неравенство $R(i) = (1 - r^{-1})\Delta_i > 0$. Таким образом, на любом шаге поиска $k \geqslant 1$ существует интервал, имеющий строго положительную характеристику. Отсюда с учетом (5.26) получаем, что интервалы из системы (5.32) также должны иметь положительные характеристики, ибо они подвергаются дроблению попадающими в них точками испытаний; т. е. условия (5.34) справедливы.

Из правил выбора величин z_ν^* следует, что

$$z_j = g_\nu(x_j) \geqslant z_\nu^*, \quad \nu = \nu(x_j), \quad 1 \leqslant j \leqslant k.$$

Тогда из правил вычисления характеристики (5.25), учитывая условия (5.29) и неравенство (5.34), выводим справедливость (5.35).

Лемма 5.2 (о существовании допустимой точки). *Пусть выполнены условия теоремы. Тогда существует такой номер испытания $h \geqslant 1$, при котором*

$$\nu(x^h) = m + 1. \tag{5.38}$$

Доказательство леммы 5.2. По определению (5.7), точка y_ε является внутренней точкой допустимого множества. Следовательно, для любого из отображений $y^l(x)$, $0 \leqslant l \leqslant L$, существует гиперкуб $D(m)$ некоторого m-го разбиения, принадлежащий допустимому множеству и содержащий ε-резервированное решение y_ε. Значит, также существует интервал-прообраз $d(m)$ гиперкуба и этот интервал содержит прообраз x_ε точки y_ε, т. е. $y_\varepsilon = y^l(x_\varepsilon)$. На любом шаге поиска $q > 1$ существует интервал $[x_{j-1}, x_j]$, $j = j(q)$, содержащий указанную точку x_ε.

Из свойства (5.4) гельдеровости функций и условия (5.29) теоремы следует, что

$$z_j = g_\nu(y^l(x_j)) \leqslant g_\nu(y^l(x_\varepsilon)) + 4L_\nu\sqrt{N}\,(x_j - x_\varepsilon)^{1/N}, \quad \nu = \nu(x_j), \tag{5.39}$$

$$z_{j-1} = g_\nu(y^l(x_{j-1})) \leqslant g_\nu(y^l(x_\varepsilon)) + 4L_\nu\sqrt{N}\,(x_\varepsilon - x_{j-1})^{1/N}, \quad \nu = \nu(x_{j-1}). \tag{5.40}$$

Кроме того, из условия (5.7) следует, что

$$g_\nu(y(x_\varepsilon)) \leqslant -\varepsilon_\nu, \quad 1 \leqslant \nu \leqslant m. \tag{5.41}$$

Если условие (5.38) не выполнено ни на одном из шагов поиска, то

$$\nu = \max\{\nu(x_{j-1}), \nu(x_j)\} \leqslant m,$$

и из условий (5.25), (5.30), (5.39)–(5.41) следует, что при достаточно больших q

$$R(j(q)) \geqslant \Delta_j\left(1 - \frac{8L_\nu\sqrt{N}}{r_\nu\mu_\nu}\right) + 4\frac{z_\nu^* + \varepsilon_\nu}{r_\nu\mu_\nu} > 0, \quad \nu(x_j) = \nu(x_{j-1}),$$

$$R(j(q)) \geqslant 2\Delta_j\left(1 - \frac{4L_\nu\sqrt{N}}{r_\nu\mu_\nu}\right) + 4\frac{z_\nu^* + \varepsilon_\nu}{r_\nu\mu_\nu} > 0, \quad \nu(x_j) \neq \nu(x_{j-1}).$$

Из этих неравенств, предела (5.35), а также правила (5.26) вытекает необходимость проведения испытания в рассматриваемом интервале $[x_{j-1}, x_j]$ на некотором шаге поиска, что приведет к его разделению на два подынтервала, хотя бы один из которых будет содержать точку x_ε. Повторение этого процесса при $q \to \infty$ порождает бесконечную последовательность подынтервалов, стягивающу-

юся к точке x_ε. В начале доказательства леммы было показано, что существует интервал $d(m)$, содержащий точку x_ε и все точки этого интервала являются внутренними точками допустимого множества, т. е. их индексы равны $m+1$. Следовательно, некоторый интервал $[x_{j-1}, x_j]$ из последовательности стягивающихся к x_ε интервалов будет принадлежать интервалу $d(m)$, что доказывает справедливость (5.38).

Лемма 5.3 (о допустимости любой предельной точки). *Любая предельная точка $\overline{y} = y^l(\overline{x}^l)$ последовательности испытаний $\{y^k\}$ является допустимой, т. е.*

$$\nu(\overline{x}^l) = m + 1. \tag{5.42}$$

Доказательство леммы 5.3. Предположим, что предельная точка $\overline{y}$ не является допустимой, т. е. существует такое целое $\nu \geqslant 1$, что

$$g_i(y(\overline{x}^l)) \leqslant 0, \quad 1 \leqslant i \leqslant \nu - 1, \quad g_\nu(y(\overline{x}^l)) > 0.$$

Тогда точка $\overline{y}$ имеет окрестность $U_\rho(\overline{y})$ радиуса $\rho = \varepsilon_\nu / 2L_\nu$ такую, что все точки этой окрестности удовлетворяют условию

$$g_\nu(y) > -\varepsilon_\nu/2, \quad y \in U_\rho(\overline{y}). \tag{5.43}$$

Если целое число M удовлетворяет неравенству $2^{-M}\sqrt{N} < \rho$, то существует некоторый подкуб $D(M)$, содержащий точку $\overline{y}$. Если $\overline{y}$ является внутренней точкой $D(M)$, то прообраз $d(M)$ этого подкуба содержит все прообразы точки $\overline{y}$, следовательно, $\overline{x}$ является внутренней точкой $d(M)$. Если $\overline{y}$ — вершина подкуба $D(M)$, то существует 2^N подкубов M-го разбиения в $U_\rho(\overline{y})$ и их прообразы содержат все прообразы точки $\overline{y}$. Значит, $\overline{x}^l$ принадлежит одному из прообразов. Обозначим этот интервал как $d'(M)$. Если $\overline{x}^l$ является концевой точкой $d'(M)$, то она также является концевой точкой другого интервала $d''(M)$. Следовательно, объединение $d'(M) \cup d''(M)$ содержит $\overline{x}^l$ как внутреннюю точку. Таким образом, в любом случае существует интервал $(\alpha, \beta) \subset [0, 1]$, содержащий $\overline{x}^l$ как внутреннюю точку, и образы $y = y(x)$ всех точек интервала (α, β) удовлетворяют условию (5.43).

Так как $\overline{x}^l$ — предельная точка, то интервалы из стягивающейся к ней последовательности $[x_{t-1}, x_t]$, $t = t(q)$ должны удовлетворять

условию $[x_{t-1}, x_t] \subset (\alpha, \beta)$ при достаточно больших q. Таким образом, из правила (5.24) и утверждения леммы 2 (5.38) следует, что при любом $q \geqslant h$

$$z_\nu^* = -\varepsilon_\nu, \quad 1 \leqslant \nu \leqslant m, \tag{5.44}$$

и из (5.25), (5.43) и (5.44) получаем оценку

$$R(t(q)) \leqslant \Delta_i + (z_i - z_{i-1})^2 / r_\nu^2 \mu_\nu^2 \Delta_i - 2\varepsilon_\nu / r_\nu \mu_\nu.$$

Следовательно, в соответствии с (5.33) получаем при достаточно больших q

$$R(t(q)) < -\varepsilon_\nu / r_\nu \mu_\nu < 0. \tag{5.45}$$

Последнее неравенство противоречит с условием (5.34), что в свою очередь приводит к противоречию с предельным характером точки $\overline{x}$.

Доказательство теоремы 5.2.

1. Для обоснования левого равенства в (5.31) докажем справедливость неравенств

$$\varphi(y(\overline{x})) \leqslant z_{m+1}^* = \min\{z_i \colon i \in I_{m+1}\} \tag{5.46}$$

для допустимой предельной точки $\overline{x}$ последовательности $\{x^k\}$. Предположим, что последовательность вложенных интервалов (5.32) при любом $p \geqslant 1$ не удовлетворяет условиям

$$\max\{\nu(x_{t-1}), \nu(x_t)\} = m + 1, \quad t = t(q^p). \tag{5.47}$$

Тогда из неравенств

$$g_\nu(y(x_i)) > 0, \quad \nu = \nu(x_i), \quad i = t{-}1, t,$$

правил (5.25) и соотношения (5.44) следует выполнимость неравенства (5.45) при достаточно больших q, что противоречит условию положительности характеристики (5.34). Значит, условие (5.47) выполняется при достаточно больших q.

Если допустить, что (5.46) не имеет места, то для некоторого положительного числа δ должны выполняться неравенства

$$z_i = g_{m+1}(y(x_i)) > z_{m+1}^* + \delta, \quad \nu(x_i) = m + 1, \quad i = t, t{-}1, \quad t = t(q^p).$$

Из этих неравенств, правил вычисления характеристики (5.25), условия (5.47) и предела (5.35) следует оценка для характеристики

$$R(t(q^p)) \leqslant 2\Delta_t - 4\delta/r_{m+1}\mu_{m+1} < -4\delta/r_{m+1}L_{m+1} < 0,$$

если число p достаточно велико. Но полученная оценка противоречит положительности характеристики (5.34), и, следовательно, должно иметь место свойство (5.46).

2. Для доказательства правого неравенства в (5.31) покажем, что отношение

$$\varphi(y_\varepsilon) \leqslant z^*_{m+1} - \beta$$

невыполнимо при достаточно больших значениях числа k, если $\beta > 0$.

При доказательстве леммы 5.2 был рассмотрен интервал $[x_{j-1}, x_j]$, $j = j(q)$, содержащий прообраз x_ε точки y_ε, для которого при достаточно больших q выполняется условие (5.47). Тогда из условия

$$\varphi(y_\varepsilon) = g_{m+1}(y(x_\varepsilon)) \leqslant z^*_{m+1} - \beta,$$

а также из (5.25), (5.39), (5.40) следует, что

$$R(j(q)) \geqslant \Delta_j \left(1 - \frac{8L_{m+1}\sqrt{N}}{r_{m+1}\mu_{m+1}}\right) + \frac{4\beta}{r_{m+1}\mu_{m+1}}, \quad \nu(x_{j-1}) = \nu(x_j),$$

$$R(j(q)) \geqslant 2\Delta_j \left(1 - \frac{4L_{m+1}\sqrt{N}}{r_{m+1}\mu_{m+1}}\right) + \frac{4\beta}{r_{m+1}\mu_{m+1}}, \quad \nu(x_{j-1}) \neq \nu(x_j).$$

Из этих неравенств, свойства (5.4) гельдеровости функций задачи и условия теоремы (5.30) получаем справедливую для достаточно больших q оценку

$$R(j(q)) > 4\beta/r_{m+1}\mu_{m+1} > \beta/2r_{m+1}L_{m+1}\sqrt{N} > 0.$$

С одной стороны, согласно полученному неравенству и правилу (5.26), интервал $[x_{j-1}, x_j]$ должен стягиваться в точку, а с другой — стягивающийся интервал должен удовлетворять условию (5.35), что противоречит существованию предела. Следовательно, нижняя грань z^*_{m+1}, совпадающая со средней частью

в отношении (5.31), не может превышать значение $\varphi(y_\varepsilon)$. *Теорема доказана.*

Пример 5.2. Рассмотрим задачу из примера 5.1 и решим ее с использованием параллельного индексного алгоритма со следующими параметрами: параметр надежности $r_1 = \ldots = r_4 = 1{,}4$, точность решения $\varepsilon = 10^{-3}$, плотность построения развертки $m = 12$, параметры резервирования $\varepsilon_\nu = 0$. Число процессоров и соответственно число разверток, варьировалось от 2 до 6. Во всех экспериментах была получена одна и та же оценка оптимума $(y_1^*, y_2^*) = (0{,}942, 0{,}945)$.

На рис. 5.8 изображена область поиска — квадрат, где выделена допустимая область задачи и обозначены точки испытаний, полученные при применении параллельного индексного метода с шестью развертками.

В таблицах 5.2–5.4 приведены подробные результаты экспериментов (табл. 5.1 соответствует последовательному алгоритму и приведена для сравнения).

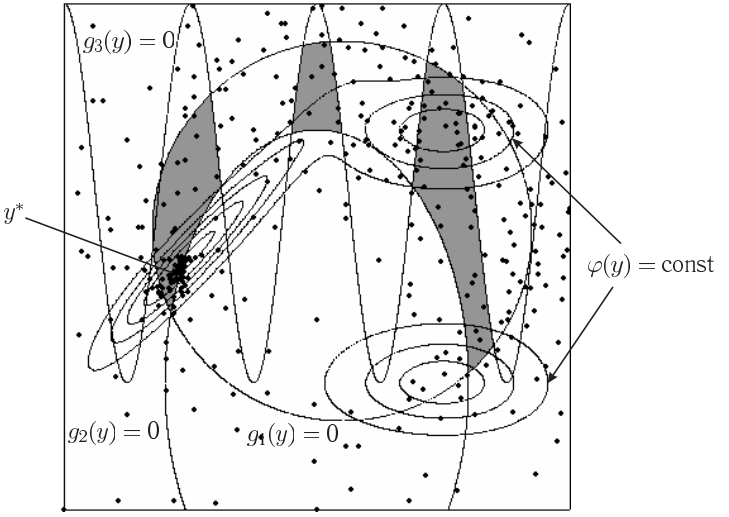

Рис. 5.8. Результаты решения задачи с использованием шести процессоров

Таблица 5.1. Результаты решения задачи с использованием одного процессора

l	Вычислено значений функции			
	g_1	g_2	g_3	$\varphi = g_4$
1	1098	623	392	152

Таблица 5.2. Результаты решения задачи с использованием двух процессоров

l	Вычислено значений функции				
	g_0	g_1	g_2	g_3	$\varphi = g_4$
1	636	567	331	212	104
2	632	327	193	125	61
Всего	1268	894	524	337	165

Таблица 5.3. Результаты решения задачи с использованием четырех процессоров

l	Вычислено значений функции				
	g_0	g_1	g_2	g_3	$\varphi = g_4$
1	585	508	311	206	88
2	579	262	157	86	36
3	579	293	174	119	41
4	579	284	178	116	48
Всего	2322	1347	820	527	213

В таблице 5.5 приведена оценка ускорения, возникающего при использовании параллельного алгоритма. Ускорение определялось как отношение числа итераций (которое совпадает с числом проверок первого ограничения g_1), выполненных последовательным алгоритмом, и наибольшего числа итераций, выполненных параллельным алгоритмом на одном из процессоров. Данный способ определения ускорения актуален для задач, в которых оценка значений функций требует заметных вычислительных ресурсов.

Таблица 5.4. Результаты решения задачи с использованием шести процессоров

l	Вычислено значений функции				
	g_0	g_1	g_2	g_3	$\varphi = g_4$
1	354	289	165	105	44
2	346	90	5 5	28	10
3	346	113	66	40	10
4	346	160	87	48	25
5	346	224	119	68	23
6	346	274	159	97	37
Всего	2084	1150	651	386	149

Таблица 5.5. Результаты численных экспериментов

Число процессоров	Ускорение
1	—
2	1,93
4	2,16
6	3,8

Результаты проведенных экспериментов наглядно демонстрируют эффект ускорения (уменьшение количества трудоемких вычислений значений функций задачи).

5.4. Параллельные вычисления на основе новых схем построения множественных кривых Пеано

5.4.1. Схема построения вращаемых разверток

Использование описанной в п. 5.3.1 схемы построения множественных разверток (далее будем называть их сдвиговыми развертками или С-развертками) позволяет сохранить информацию о близости точек в многомерном пространстве и тем самым обеспечить более точную (по сравнению с единственным отображением) оценку константы Липшица в процессе поиска. Однако у данного под-

хода есть серьезные ограничения, которые сужают применимость параллельных алгоритмов, построенных на основе С-разверток.

Ввиду того, что сдвиговая развертка строится в результате сдвига вдоль главной диагонали гиперкуба D, причем шаг сдвига уменьшается в 2 раза при построении каждого следующего отображения, количество таких сдвигов ограниченно плотностью развертки m. В случае если сдвиг будет на величину, меньшую 2^{-m}, то следующая развертка совпадет с предыдущей. Таким образом, на количество применяемых разверток L, а как следствие, и число используемых процессоров, накладывается ограничение

$$L \leqslant m,$$

где m — плотность развертки.

Также из алгоритма построения сдвиговой развертки следует, что решение исходной задачи в области D сводится к решению набора задач в расширенных областях D_l из (5.14) при наличии дополнительного ограничения (5.17) на попадание в область D. Как результат — усложняется структура области поиска, имеет место экспоненциальное уменьшение объема области поиска D по отношению к объемам областей D_l с ростом размерности задачи.

Преодолеть указанные недостатки, сохранив информацию о близости точек в N-мерном пространстве, позволяет новая схема построения множественных отображений. Отличительной чертой предлагаемой схемы является посторенние множества кривых Пеано не с помощью сдвига вдоль главной диагонали гиперкуба, а поворотом развертки вокруг начала координат [47]. При этом найдется отображение $y^i(x)$, которое точкам многомерного пространства y', y'', которым при исходном отображении соответствовали достаточно далекие прообразы на отрезке [0, 1], будет сопоставлять более близкие прообразы x', x''.

Развертки, порождаемые в соответствии с новой схемой, будем называть вращаемыми развертками или В-развертками.

Для иллюстрации на рис. 5.9 изображены две кривые, являющиеся приближением к развертке Пеано для случая $N = 2$. Узлы сетки в N-мерном пространстве отмечены темными кружками,

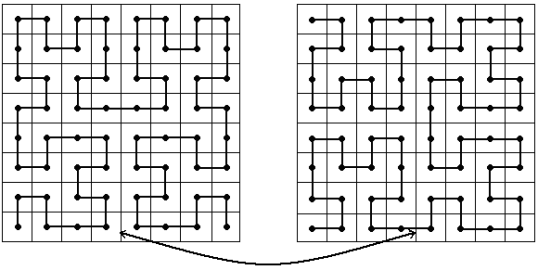

Рис. 5.9. Близкие/далекие точки при использовании вращаемых разверток

стрелкой указана одна из пар точек, прообразы которых далеки на одномерной оси при одном отображении и близки при другом.

Максимальное число различных поворотов развертки, отображающей N-мерный гиперкуб на одномерный отрезок, составляет 2^N. Использование всех поворотов может оказаться избыточным, в этом случае можно выбрать лишь часть из всех возможных вариантов.

В качестве возможного подхода можно предложить получение новых разверток при помощи поворота исходной развертки на угол $\pm\pi/2$ в каждой из координатных плоскостей. В качестве примера на рис. 5.10 приведены матрицы поворота в плоскости (y_2, y_4) при $N = 5$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Рис. 5.10. Матрицы поворота

Число подобных пар поворотов определяется числом координатных плоскостей пространства, которое равно $C_N^2 = N(N-1)/2$, а общее число преобразований будет равно $N(N-1)$. Учитывая исходное отображение, приходим к заключению, что данный способ позволяет строить до $N(N-1)+1$ разверток для отображения

N-мерной области на соответствующие одномерные отрезки. При этом дополнительное ограничение $g_0(y) \leqslant 0$, где $g_0(y)$ — из (5.17), которое возникало при построении сдвиговых разверток, отсутствует. В случае необходимости данный способ построения множества отображений может быть легко «отмасштабирован» для получения большего (вплоть до 2^N) числа разверток.

Использование множества отображений $Y_L(x)=\{y^1(x),\ldots, y^L(x)\}$ приводит к формированию соответствующего множества одномерных многоэкстремальных задач

$$\min\{\varphi(y^l(x)) \colon x \in [0,1],\ g_j(y^l(x)) \leqslant 0,\ 1 \leqslant j \leqslant m\}, \quad 1 \leqslant l \leqslant L. \tag{5.48}$$

Каждая задача из данного набора может решаться независимо, при этом любое вычисленное значение $z = g_\nu(y')$, $y' = y^i(x')$ функции $g_\nu(y)$ в i-й задаче может интерпретироваться как вычисление значения $z = g_\nu(y')$, $y' = y^s(x'')$ для любой другой s-й задачи без повторных трудоемких вычислений функции $g_\nu(y)$. Подобное информационное единство позволяет решать исходную задачу (5.1) путем параллельного решения индексным методом L задач вида (5.48) на наборе отрезков $[0,1]$.

Решающие правила параллельного алгоритма с использованием B-разверток полностью совпадают с правилами последовательного алгоритма, кроме способа проведения испытания. А именно испытание в точке $x^k \in [0,1]$, осуществляемое в s-й задаче в случае использования B-разверток, состоит в последовательности действий:

1) Определить образ $y^k = y^s(x^k)$ при соответствии $y^s(x)$;

2) Проинформировать остальные процессоры о начале проведения испытания в точке y^k (*блокирование* точки y^k);

3) Вычислить величины $g_1(y^k), \ldots, g_\nu(y^k)$, где значения индекса $\nu \leqslant m$ определяются условиями

$$g_j(y^k) \leqslant 0, \quad 1 \leqslant j < \nu, \quad g_\nu(y^k) > 0, \quad \nu \leqslant m.$$

Выявление первого нарушенного ограничения прерывает испытание в точке y^k. В случае, когда точка y^k допустима, испытание включает вычисление значений всех функций задачи, при этом значение

индекса принимается равным величине $\nu = m + 1$. Тройка

$$y^s(x^k), \quad \nu = \nu(x^k), \quad z^k = g_\nu\big(y^s(x^k)\big) \tag{5.49}$$

является результатом испытания в точке x^k;

4) Определить прообразы $x^{kl} \in [0,1]$, $1 \leqslant l \leqslant L$, точки y^k и интерпретировать испытание, проведенное в точке $y^k \in D$, как проведение испытаний в L точках

$$x^{k1}, \ \ldots, \ x^{kL} \tag{5.50}$$

с одинаковыми результатами

$$\nu(x^{k1}) = \ldots = \nu(x^{kL}) = \nu(x^k),$$
$$g_\nu(y^1(x^{k1})) = \ldots = g_\nu(y^L(x^{kL})) = z^k;$$

5) Проинформировать остальные процессоры о результатах испытания в точке y^k.

5.4.2. Сравнение эффективности алгоритмов

Пример 5.3. Для сравнения эффективности алгоритмов в очередной раз применим описанную ранее в п. 2.2.2 методику построения операционных характеристик. Будем использовать генератор двумерных функций G в соответствии со схемой (2.103).

Эксперименты проводились для описанных выше алгоритмов со сдвиговыми (С) и вращаемыми (В) развертками с плотностью $m = 12$, число разверток $L = 2$. При этом использовались параметры $r_\nu = 2{,}1$, $\varepsilon = 0{,}01$ и резервы $\varepsilon_\nu = 0{,}05$, $0 \leqslant \nu \leqslant 1$, из (5.24). Данные параметры являются минимальными для соответствующих методов, при которых достигается решение 100 % задач из выборки.

Операционные характеристики методов, полученные на выборке, порожденных генератором G, представлены на рис. 5.11. При этом нижняя разрывная кривая характеризует метод со сдвиговыми развертками, средняя разрывная кривая — метод с единственной разверткой, а верхняя непрерывная линия — метод с вращаемыми развертками. Указанное расположение кривых показывает, что алгоритм с В-развертками обеспечивает в среднем более быстрое получение оценок, лежащих в заданной окрестности решения,

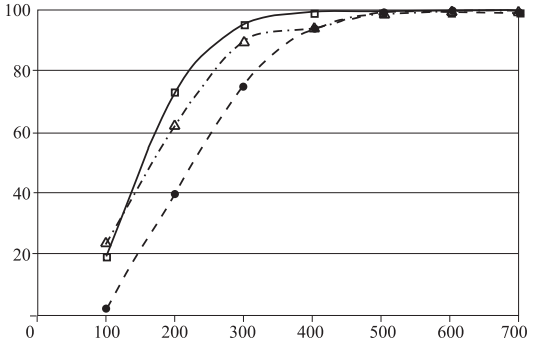

Рис. 5.11. Операционные характеристики

чем алгоритм с С-развертками или же алгоритм с единственной разверткой.

Пример 5.4. С целью иллюстрации использования параллельного алгоритма с множеством В-разверток при решении существенно многомерных задач рассмотрим задачу минимизации функции из примера 4.4

$$\varphi(y) = \sum_{i=1}^{N} \left(y_i^2 - \cos(18y_i^2)\right), \quad -1{,}5 \leqslant y_i \leqslant 1{,}5, \quad 1 \leqslant i \leqslant N, \quad N = 6,$$

где минимальное значение $\varphi(y^*) = -N$ достигается в точке $y^* = 0$.

Для решения данной задачи использовались последовательный и параллельный методы с вращаемой разверткой с плотностью $m = 10$ использовался параметр метода $r = 2{,}0$ и $\varepsilon = 0{,}05$ в критерии остановки. Число разверток $L = 30$ соответствовало числу процессоров, задействованных при решении задачи. И последовательный, и параллельный алгоритмы нашли решение с требуемой точностью, при этом последовательный алгоритм затратил 173 116 итераций, а параллельный — 8 535 (максимальное число итераций на одном процессоре). Ускорение по числу итераций составило 20,28, а ускорение по времени — 7,48.

Пример 5.5. Рассмотрим теперь задачу минимизации функции из предыдущего примера при $-1 \leqslant y_i \leqslant 1$, $1 \leqslant i \leqslant N$, $N = 11$,

15, 20. В данном эксперименте оценим ускорение по времени и эффективность нахождения глобального оптимума при одинаковом условии остановки по числу испытаний. В таблице 5.1 приведены результаты решения указанных задач.

Таблица 5.6. Результаты минимизации многомерных функций Растригина

	Кол-во испытаний	Последовательный алгоритм глобального поиска		Параллельный алгоритм глобального поиска с использованием B-разверток		
		Время, с	Оценка φ^*	Время, с (процессов)	Оценка φ^*	Ускорение по времени
$N = 11$	5 000 000	544 068,8	$-10,997543$	9 321,5(110)	$-10,995598$	**112,01**
$N = 15$	3 000 000	605 788,4	$-14,835308$	7 414,6(100)	$-14,925409$	**81,7**
$N = 20$	3 000 000	605 956,7	$-19,989635$	6 654,7(100)	$-19,048833$	**91,06**

Из таблицы 5.1 видно, что индексный метод с вращаемой разверткой, запущенный на исполнение на более чем 100 вычислительных узлах показывает близкое к линейному ускорение по времени выполнения вычислений, что доказывает его высокую эффективность.

Пример 5.6. Продемонстрируем теперь эффективность параллельного индексного метода при решении условных задач глобальной оптимизации. В качестве целевой функции рассмотрим модифицированную функцию Растригина, а ограничения выберем так, чтобы глобальный минимум находился внутри допустимой области. Сформированная таким образом задача имеет вид

$$\varphi(y) = \sum_{i=1}^{N} (y_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot y_i)) \to \min,$$

$$g_1(y) = \sum_{i=1}^{N} y_i - 0{,}5 \leqslant 0,$$

$$g_2(y) = \sum_{i=1}^{N} y_i^2 - 1{,}0 \leqslant 0,$$

$$g_3(y) = \sum_{i=1}^{N} \sin(i \cdot \pi \cdot y_i) - 0{,}3 \leqslant 0$$

при $-1 \leqslant y_i \leqslant 1$, $1 \leqslant i \leqslant N$, $N = 8$. Результаты решения данной задачи последовательным и параллельным индексным методом приведены в табл. 5.7. В обоих случаях метод завершал работу по достижении заданной точности $\varepsilon = 0{,}0035$.

Таблица 5.7. Сравнение алгоритмов при решении задачи с ограничениями

	Последовательный, 56 разверток	Параллельный, 56 процессоров всего (максимум на один процесс)
Количество испытаний	309 640	244 356 (4891)
Количество вычислений критерия	33 874	22 722 (789)
Ускорение по испытаниям		63,3
Ускорение по вычислениям критерия		42,93
Время вычислений	40006.6 с	626,8 с
Ускорение по времени		63,83
Оценка φ^*	0,016412	0,022182

Из таблицы 5.2 видно, что параллельный алгоритм имеет ускорение 63,3 по числу испытаний, 42,93 — по количеству вычислений критерия и 63,83 — по времени.

Отметим, что в задачах большей размерности сравнение последовательного и параллельного алгоритмов усложняется, поскольку в последовательном случае вся поисковая информация сосредоточена на одном вычислительном узле. Последовательному алгоритму, запущенному на одном узле, не хватает ресурсов оперативной памяти, в результате чего включается подкачка. В параллельном алгоритме все данные распределяются по процессам, что позволяет не использовать внешнюю память для хранения поисковой информации.

Глава 6

Суперкомпьютерные параллельные вычисления при решении прикладных оптимизационных задач

6.1. Оптимизация профиля колеса для рельсовых видов транспорта

Рассматриваемая задача поиска оптимального профиля колеса для рельсовых видов транспорта (ж/д транспорт, метро, трамвай, и т. п.) была решена в рамках совместного научно-исследовательского проекта РФФИ (№ 04-01-89002-НВО_а) и NWO (Netherlands Organisation for Scientific Research, № 047.016.014) «Быстрые вычисления в глобальной оптимизации: последовательные и параллельные среды», выполненного ННГУ им. Н. И. Лобачевского и Технологическим университетом г. Делфт, Нидерланды (TUDelft).

6.1.1. Постановка задачи

Подробно постановка задачи оптимизации профиля рельсового колеса изложена в работах [69, 70]; здесь приведем лишь краткое описание данной задачи. Поскольку колесо имеет коническую форму, центр колесной пары совершает синусоидальное движение. Данный процесс проиллюстрирован на рис. 6.1. Параметры контакта колеса с рельсом, такие как радиус вращения, угол контакта и угол наклона колесной пары, варьируются при поперечном — относительно рельса — смещении колесной пары. Зависимость между этими изменениями и поперечной позицией колесной пары определяется профилем колеса и рельса.

Важной характеристикой контакта колеса и рельса является радиус вращения колеса в контактной точке. Фактически радиус может быть разным для правого и левого колес, так как колесная

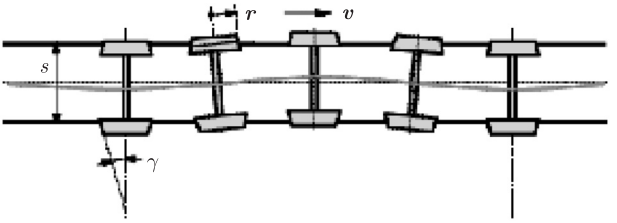

Рис. 6.1. Смещение колесной пары

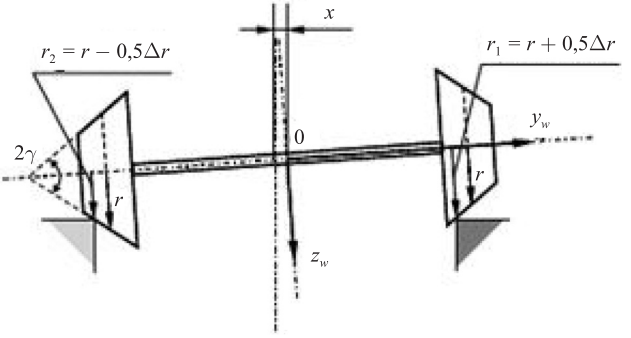

Рис. 6.2. Разница радиусов вращения

пара смещается по рельсу (радиусы r_1 и r_2 соответственно на рис. 6.2).

Когда колесная пара находится в центральной позиции, радиусы вращения левого и правого колес совпадают, т. е. $r_1 = r_2 = r$. Отличие между радиусами вращения левого и правого колес может быть определено как функция бокового смещения у колесной пары по отношению к ее центральной позиции $\Delta r(x) = r_1(x) - r_2(x)$.

Математическая модель данной задачи, построенная в Технологическом университете Делфта, состоит в следующем. Профиль колеса описывается с помощью B-сплайна, для построения которого выбирается множество точек на кромке, основании кромки и поверхности качения колеса (рис. 6.3). Положение этих точек

может варьироваться с целью изменения профиля. Для уменьшения трудоемкости задачи, оптимизации положения точек на верхней части кромки и конической части профиля фиксируются, так как эти части колесного профиля не участвуют в контакте с рельсом.

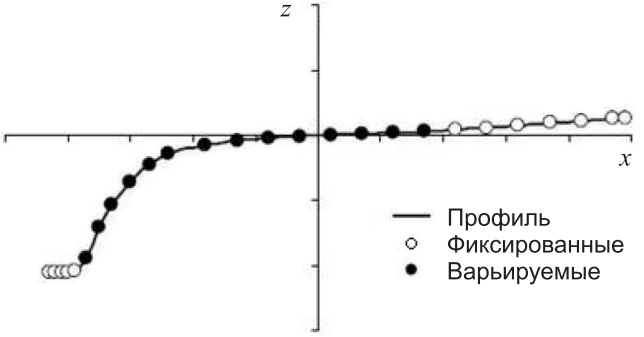

Рис. 6.3. Моделирование профиля колеса

В качестве компонент вектора y параметров задачи оптимизации выбраны ординаты z_i подвижных точек сплайна, т. е.

$$y = [z_1, \dots, z_N],$$

при этом абсциссы данных точек фиксированы.

В рассматриваемой задаче число подвижных точек, а, следовательно, и число переменных, составляет $N = 11$, границы изменения параметров $[-1, 1]$; минимизируется разность радиусов вращения $\Delta r(x)$, а ограничения вводятся из соображений устойчивости (например, одно из ограничений наложено на максимальный угол наклона колесной пары).

6.1.2. Результаты численных экспериментов

Таким образом, возникает задача оптимизации с числом параметров $N = 11$, числом ограничений $m = 6$, в которой целевая функция и функции ограничений — многоэкстремальные. Для иллюстрации на рис. 6.4 приведено одномерное сечение целевой функции.

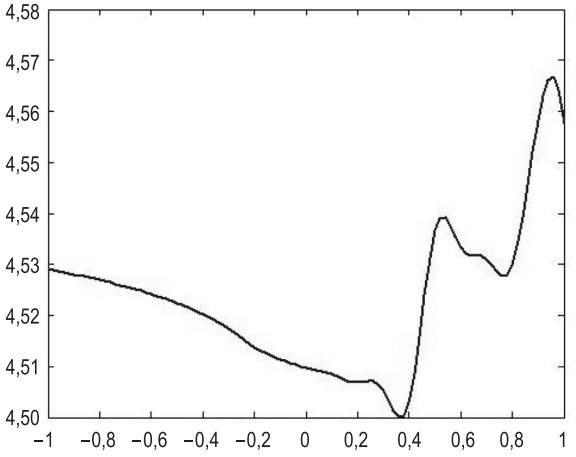

Рис. 6.4. Сечение целевой функции

Ограничения аналитической формы не имеют и заданы в виде вычислительных процедур в пакете MATLAB. Вычисление значений, входящих в задачу функций для одного набора значений параметров, занимает более десяти 20 секунд (Pentium IV, 3 ГГц).

Итак, задача сформулирована, теперь попробуем оценить, какие ресурсы нужны для ее решения в зависимости от выбранного метода.

Если оптимум искать, используя полный перебор на сетке выбранной точности, получим следующее. Взяв десять вариантов значения каждого из параметров (напомним, всего параметров 11), получим 10^{11} вычислений, что дает порядка 10^{12} секунд, то есть 3 года непрерывного счета на суперкомпьютере с 10 000 процессоров. Таким образом, сложность задачи такова, что параллельные вычисления не только являются необходимыми, но и должны в обязательном порядке дополняться эффективными численными методами решения для получения результата в обозримых временных рамках.

Описанная задача решалась авторами с помощью параллельного индексного метода со сдвиговой множественной разверткой на кла-

стере из 4 компьютеров в Технологическом университете г. Делфта. Покоординатная точность поиска решения составляла 2^{-10}, то есть 1024 точки по каждому параметру. Время поиска оценки оптимума составило 27 часов, при этом значения функций задачи было посчитаны $4297 + 4415 + 4236 + 4266 = 17214$ раз.

Расчеты, проведенные специалистами TUDelft для колеса оптимизированного профиля, показали, что его ресурс (т. е. межпроверочный интервал) возрос до 120 000 км пробега (более чем в пять раз по сравнению с колесом оригинального профиля), а максимально допустимая скорость увеличилась с 40 до 60 м/с.

6.2. Решение обратной задачи химической кинетики

В этом разделе отражены результаты совместной научно-исследовательской работы, выполненной в ННГУ им. Н. И. Лобачевского и Институте нефтехимии и катализа РАН (ИНК РАН) [23].

6.2.1. Постановка задачи

Построение математических моделей сложных химических реакций предполагает наличие в них неизвестных кинетических параметров (константы скоростей, энергии активации и частота столкновений реагирующих молекул), которые можно найти, решая задачу минимизации отклонения между расчетными данными (*прямая задача*) и данными натурных экспериментов. Таким образом, возникает задача идентификации математической модели (*обратная задача химической кинетики*), которая в общем случае является задачей глобальной оптимизации.

Многие физико-химические задачи предполагают значительный объем вычислений, обеспечивающих тем не менее достаточно низкую точность. Именно к таким задачам относятся обратные задачи химической кинетики, в процессе решения которых требуется решать множество прямых задач — систем дифференциальных и алгебраических уравнений. Для кинетики сложных химических реакций характерно наличие быстро и медленно меняющихся параметров (так как стадии реакций протекают с различными скоро-

стями), поэтому решение прямых кинетических задач осложняется жесткостью систем дифференциальных уравнений, описывающих механизмы этих реакций.

Прямая кинетическая задача для изотермической нестационарной модели в закрытой системе представляет собой задачу Коши для системы обыкновенных дифференциальных уравнений:

$$\frac{dx_i}{dt} = F_i, \quad i = 1, \ldots, M; \quad F_i = \sum_{j=1}^{N} S_{ij} w_j;$$
$$w_j = k_j \prod_{i=1}^{M} (x_i)^{|\alpha_{ij}|} - k_{-j} \prod_{i=1}^{M} (x_i)^{|\beta_{ij}|} \tag{6.1}$$

с начальными условиями $t = 0$, $x_i(0) = x_i^0$, где x_i — концентрации веществ (*мольные доли*), участвующих в реакции; M — количество веществ; N — количество стадий; S_{ij} — стехиометрическая матрица; w_j — скорость j-й стадии (1/ч); k_j, k_{-j} — приведенные константы скорости прямой и обратной реакции (1/ч) соответственно; α_{ij} — отрицательные элементы S_{ij}, β_{ij} — положительные элементы S_{ij}.

Поскольку часть констант k_j, k_{-j}, как правило, неизвестна, возникает задача идентификации математической модели или обратная кинетическая задача, которая представляет собой задачу минимизации функции отклонения между расчетными и экспериментальными данными:

$$F = \sum_{i=1}^{n} \sum_{j=1}^{M} |x_{ij}^{\text{р}} - x_{ij}^{\text{э}}| \to \min, \tag{6.2}$$

где $x_{ij}^{\text{р}}$ — расчетные значения концентраций наблюдаемых веществ, мольные доли; $x_{ij}^{\text{э}}$ — экспериментально полученные значения концентраций наблюдаемых веществ, мольные доли; n — количество точек эксперимента.

Для нахождения энергии активации и частоты столкновений реагирующих в элементарной стадии молекул используется уравнение Аррениуса:

$$k = Ae^{-Ea/RT}, \tag{6.3}$$

где k — приведенная константа скорости элементарной стадии, 1/ч; E — энергия активации, Дж/моль; R — универсальная газовая постоянная, Дж/(моль·К); A — частота столкновений реагирующих молекул; T — температура, К.

Так как при поиске констант скоростей элементарных стадий они могут попасть в область, где система дифференциальных уравнений, описывающая реакцию, может оказаться жесткой, для решения прямой задачи используется метод Мишельсена с автоматическим выбором шага [50]. А поскольку выходные данные модели x_{ij}^{p} зависят нелинейно от значений констант k_j, k_{-j}, задача (6.3) является задачей многоэкстремальной оптимизации.

6.2.2. Результаты численных экспериментов

Традиционно для решения задачи (6.3) в ИНК РАН применялись методы, основанные на идеях случайного поиска. При этом характерное время расчетов составляло несколько суток машинного времени. В рамках выполнения совместных исследований ИНК РАН и ННГУ было предложено использовать эффективные параллельные методы глобальной оптимизации, описанные в главе 5.

Для применения новой методики идентификации была выбрана математическая модель реакции каталитического карбоалюминирования олефинов и ацетиленов с помощью триалкилаланов в присутствии комплексов переходных металлов. Эта реакция получила применение в лабораторной практике ИНК РАН как эффективный способ построения новых Me–C (метил–углерод), Et–C (этил–углерод) и C–C (углерод–углерод) связей [60]. В ИНК РАН были выполнены натурные эксперименты по проведению реакции каталитического карбоалюминирования олефинов и ацетиленов при различных температурах и фиксированных начальных концентрациях. В ходе эксперимента наблюдались концентрации пяти веществ и выдавались их процентные соотношения.

Для математического описания реакции в ИНК РАН был предложен ряд схем (содержащих параметры, которые требуется идентифицировать), описывающих реакции каталитического карбоалюминирования олефинов и ацетиленов. В качестве приоритетной задачи идентификации неизвестных параметров требовалось опреде-

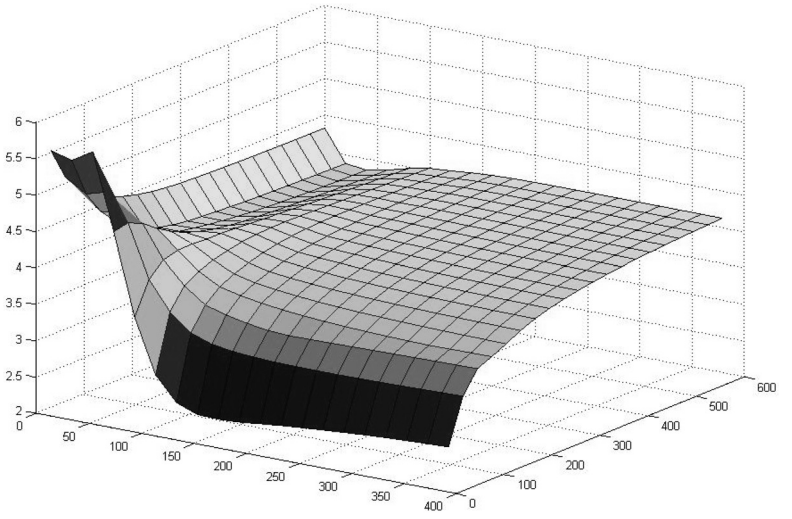

Рис. 6.5. Двумерное сечение целевой функции

лить константы скоростей элементарных стадий для предложенных схем реакции при различных температурах, а также найти энергии активации элементарных стадий при различных катализаторах. Подробно указанные схемы и задача идентификации описаны в [23]. Здесь же отметим, что возникающая при этом задача минимизации зависит от 10 параметров и является безусловной (функциональные ограничения отсутствуют); вычисление значения функции в первой точке требует порядка 3 с машинного времени (Intel Xeon 3.2 GHz). Целевая функция является многоэкстремальной, что подтверждается ее двумерным сечением, приведенным на рис. 6.5.

В результате расчетов были получены оценки констант скоростей элементарных стадий для 10- и 12-стадийных схем реакции карбоалюминирования в присутствии катализатора Cp_2ZrCl_2. Значение минимизируемой функции в полученной точке составило $F = 3{,}040$. Достичь значения функции, близкого к нулю, не удалось, поскольку данные натурных экспериментов содержали погрешности. Тем не менее модель, идентифицированная с по-

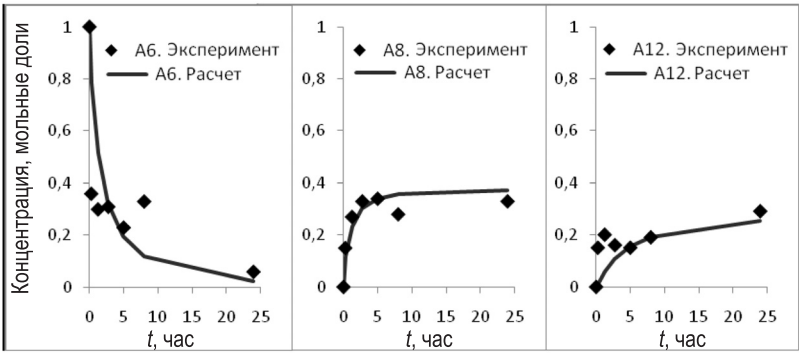

Рис. 6.6. Сопоставление расчетных и экспериментальных данных

мощью параллельного метода глобальной оптимизации, позволяет получать хорошее согласование расчетных и экспериментальных данных. В качестве иллюстрации на рис. 6.6 представлены графики сопоставления расчетных и экспериментальных концентраций различных веществ, участвующих в реакции.

6.3. Идентификация динамической балансовой нормативной модели региональной экономики

В этом разделе приведены результаты, полученные при выполнении проекта РФФИ «Параллельные методы глобальной оптимизации в идентификации динамических балансовых нормативных моделей региональной экономики» № 11-07-97017-р_Поволжье_а, выполненного совместно коллективом исследователей из ННГУ им. Н. И. Лобачевского и ВЦ им. А. А. Дородницына РАН [13].

6.3.1. Постановка задачи

Объектом исследования является региональная экономика (области, края, республики и т. п.). Для математического описания экономических процессов, происходящих на уровне одного региона страны, ВЦ РАН предложена динамическая балансовая норматив-

ная модель экономики [34]. Данная модель является общей, при этом она содержит значительное число параметров, зависящих от конкретного региона. Найти эти параметры (*идентифицировать модель*) можно, решив задачу минимизации отклонения между расчетными временными рядами макроэкономических показателей и соответствующими им статистическими историческими данными. Функции, входящие в постановку задачи, являются нелинейными, следовательно, задача идентификации математической модели экономики является задачей глобальной оптимизации.

Идентифицированную модель можно использовать для проверки тех или иных сценариев возможного развития экономики региона на основе численных экспериментов с моделью. Выводы и предсказания, полученные с помощью идентифицированной математической модели, в принципе, могут быть экспериментально опровергнуты. Это позволяет в дальнейшем корректировать модель, и на постоянно совершенствующейся модели получать более обоснованные прогнозы.

Задача идентификации представленной здесь многосекторной модели региональной экономики решается на данных по Нижегородской области. Как уже было отмечено, нормативные модели содержат большое число неизвестных параметров (нормативов распределения продуктов и нормативов распределения финансовых средств). Для их идентификации предлагается использовать эффективные параллельные методы глобальной оптимизации, рассматриваемые в настоящей книге. Использование данных алгоритмов на суперкомпьютерных системах позволяет повысить уровень сложности математических моделей региональной экономики, выраженный числом независимых параметров. Проведение сценарных расчетов на основе идентифицированных моделей региональной экономики даст возможность более точного прогнозирования экономических последствий тех или иных стратегических решений.

6.3.2. Описание модели региональной экономики

Экономику Нижегородской области можно сагрегировать в три основных сектора, примерно равной мощности:

1) отрасли инфраструктуры, производства и распределения сырья (сельское хозяйство, электроэнергетика, строительство, транспорт, госуправление, образование, здравоохранение);

2) обрабатывающие отрасли (машиностроение и металлообработка, химическая и нефтехимическая промышленность, топливная промышленность, черная и цветная металлургия, промышленность строительных материалов, лесная, деревообрабатывающая и целлюлозно-бумажная, легкая, пищевая, химико-фармацевтическая, микробиологическая, мукомольно-крупяная и комбикормовая, полиграфическая промышленность, научный комплекс);

3) отрасли услуг (торговля, операции с недвижимым имуществом, финансовая деятельность, предоставление услуг).

При построении модели выделим следующих экономических агентов: Правительство области, Производители в лице трех выделенных секторов, Банковская система, Домашние хозяйства области, Внешние потребители и поставщики.

Несмотря на отсутствие на территории Нижегородской области богатых природных ресурсов, ее экономика является одним из наиболее развитых в промышленном отношении регионов Российской Федерации. Основу промышленности Нижегородской области составляют обрабатывающие отрасли, мощный военно-промышленный комплекс; хорошо развит фундаментальный научный комплекс. Здесь имеются хорошие возможности для развития наукоемких производств и инновационных отраслей.

В качестве основы при построении математической модели экономики региона был использован трехсекторный вариант модели общего равновесия с запасами продуктов, факторов производства и денег при налогообложении и наличии теневого оборота [35].

Напомним, что производители в модели экономики Нижегородской области представлены тремя секторами: 1) инфраструктурный комплекс (X); 2) комплекс обрабатывающих отраслей (Y); 3) комплекс отраслей услуг, торговли, недвижимости, финансов (Z). Производители (секторы региональной экономики X, Y, Z), используют в производстве труд, капитал и промежуточную продукцию смежных секторов. Производители поставляют продукцию на внутренние рынки и внешний рынок. Домашние хозяйства L

предлагают труд и потребляют конечную продукцию. Торговый посредник T занят перераспределением материальных и финансовых потоков. Банковская система B выдает кредиты производителям с целью извлечения банковской прибыли. Правительство региона G собирает налоги с производителей и домашних хозяйств и регулирует расходы бюджета. Считается, что свои цены формируются на каждом рынке каждой продукции и изменение цен обратно пропорционально изменению запасов соответствующих продуктов.

Для учета теневого оборота предполагается, что произведенный продукт производители делят на открытый и теневой, последний не облагается налогами. В результате у производителя оказывается два вида денег — «белые» и «черные». «Черные» деньги могут отмываться, а их запас подвергается штрафным санкциям. У потребителя все деньги считаются «белыми», а свой доход потребитель делит по заданным нормам потребления легальных и теневых продуктов всех секторов. Производственные секторы $m = X, Y, Z$ платят налог на прибыль n_1, налог на добавленную стоимость n_2, акцизы на валовой выпуск n_3, единый социальный налог на фонд заработной платы n_4, таможенные платежи на экспорт n_5. Домашние хозяйства L в модели оплачивают таможенные платежи с импорта n_6, подоходный налог с зарплаты n_7.

Показатели и параметры модели снабжены верхними и нижними индексами, причем верхние индексы используются для агентов, а нижние для благ. Распределение запаса каждого блага производится по нормативу: a_i^{nm} — доля запаса блага i, идущая от агента n к агенту m. Распределение денег производится также по некоторому нормативу: b_i^{nm} — доля запаса денег агента m, идущая агенту n за продукт i. Коэффициенты фондоемкости также задаются нормативами: c_i^m — норма затрат продукта i на создание единицы фондообразующего продукта агента m.

Идентификация параметров проводилась на сравнении выходных временных рядов переменных модели с доступными статистическими временными рядами 2000–2008 гг. В качестве критерия близости расчетного и статистического временных рядов использовался индекс несовпадения Тэйла, который вычисляется в соответствии

с формулой

$$U = \frac{\sqrt{\sum (x_i - y_i)^2}}{\sqrt{\sum x_i^2} + \sqrt{\sum y_i^2}};$$

чем ближе он к нулю, тем ближе сравниваемые ряды.

6.3.3. Результаты численных экспериментов

Подробное описание построенной модели выходит за рамки этой книги, дополнительная информация может быть получена в [13]. Здесь же отметим, что возникающая в процессе идентификации задача глобальной оптимизации имеет 60 независимых параметров, 86 нелинейных ограничений и многоэкстремальную целевую функцию. В качестве иллюстрации на рис. 6.7 приведено двумерное сечение функции, подтверждающее ее существенную многоэкстремальность.

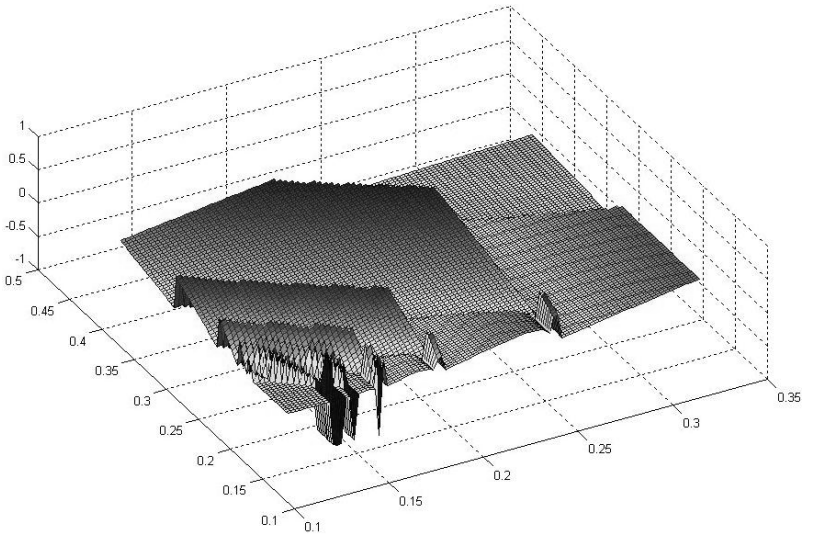

Рис. 6.7. Двумерное сечение целевой функции

Данная задача была решена на кластере ННГУ им. Н. И. Лобачевского с помощью параллельного индексного метода с вращаемой разверткой (использовалось 8 процессоров). Оценка оптимума была получена после 300 000 итераций, время получения оценки составило 26 минут, общее время счета — 3 часа.

С помощью идентифицированной математической модели Нижегородской области в ВЦ РАН были проведены сценарные расчеты, отражающие динамику экономических показателей региона при различной экономической политике (пессимистический, базовый и оптимистический сценарии).

Глава 7

Программные системы глобальной оптимизации

7.1. Программная система АБСОЛЮТ для исследований и изучения методов глобальной оптимизации

Система АБСОЛЮТ представляет собой интегрированную программную систему для проведения вычислительных экспериментов с методами глобального поиска для *изучения*и*исследования* основных понятий, подходов и развитых на их основе методов в области ***глобальной оптимизации***. Рассматриваемые в рамках теории глобального поиска *многоэкстремальные задачи оптимизации* являются предметом проведения интенсивных научных исследований и широко используются в практических приложениях при автоматизированном проектировании, решении задач идентификации и др.

АБСОЛЮТ принадлежит к классу систем для проведения вычислительных экспериментов (*систем имитации объектов и явлений на ЭВМ*), разработка которых является важным направлением создания новых технологий исследований в областях науки и техники.

Отличительной чертой системы АБСОЛЮТ является наличие всех необходимых средств для проведения вычислительных экспериментов. Используя АБСОЛЮТ, пользователь системы (*оптимизатор*) может осуществить постановку задачи оптимизации, выбрать оптимизационный алгоритм из числа реализованных в системе методов глобального поиска, установить графические индикаторы для наблюдения за процессом глобального поиска, выполнить различные эксперименты, получить и проанализировать результаты выполненных экспериментов. Эта полнота и ориентированность на

новые технологии обучения отличает АБСОЛЮТ от других систем (таких, например, как LGO [61], СИМОП [11]).

Возможными сферами использования системы АБСОЛЮТ являются:

— *учебное применение* студентами и преподавателями вузов для исследований и изучения методов глобального поиска в рамках лабораторного практикума по учебным дисциплинам оптимизационного цикла (методы оптимизации, системный анализ, исследование операций и др.). При этом АБСОЛЮТ может использоваться для демонстрации ряда важных понятий математики и вычислительной математики (последовательности, предельные точки, сходимость численного метода и др.). АБСОЛЮТ может также служить примером комплексного подхода к разработке системы имитации на ЭВМ;

— *научное использование* специалистами в области теории многоэкстремальной оптимизации для исследования эффективности разрабатываемых алгоритмов глобального поиска. При этом система может использоваться как средство подготовки и проведения научных демонстраций численного решения оптимизационных задач различными методами;

— *прикладное применение* техническими и научными работниками в качестве обучающей компоненты сложных профессиональных систем оптимизации.

Пользователи, начинающие знакомиться с проблематикой глобального поиска, найдут систему АБСОЛЮТ полезной для освоения методов многоэкстремальной оптимизации, опытные оптимизаторы могут использовать систему для оценки эффективности новых разрабатываемых алгоритмов глобального поиска.

Внешний вид главного окна программной системы АБСОЛЮТ представлен на рис. 7.1.

Возможности системы. АБСОЛЮТ является интегрированной программной средой для проведения вычислительных экспериментов с методами глобального поиска. Пользователю системы предоставляется возможность:

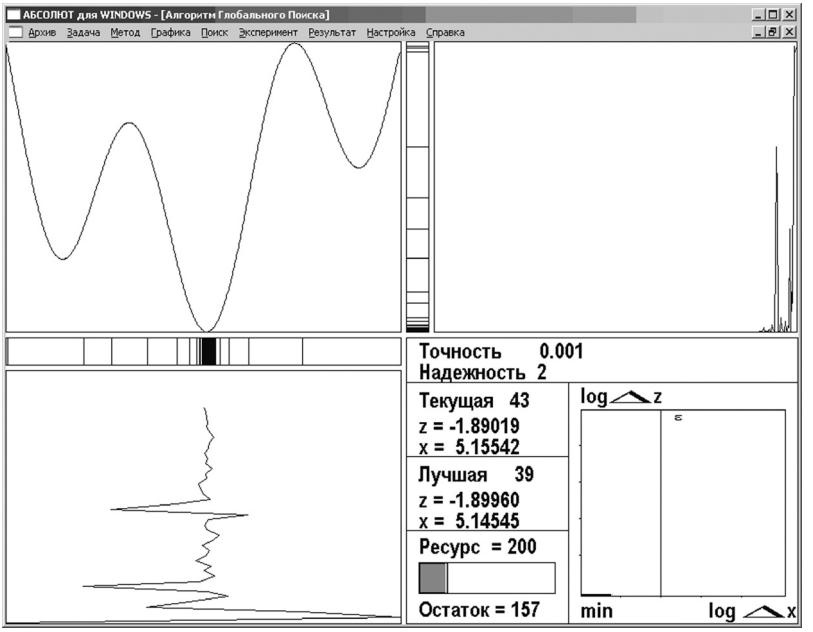

Рис. 7.1. Интерфейс системы АБСОЛЮТ

— *осуществить постановку задачи оптимизации* (минимизируемая функция может быть выбрана из стандартного набора, сгенерирована при помощи случайного датчика, задана формульным описанием или сформирована при помощи графического редактора);

— *выбрать алгоритм глобального поиска* из реализованных в системе методов оптимизации (всего АБСОЛЮТ содержит 10 разных многоэкстремальных алгоритмов, наиболее известных в теории и практике глобального поиска); в системе имеется возможность расширения набора методов внутренними средствами без использования универсальных алгоритмических языков программирования;

— *установить графические индикаторы* для наблюдения за процессом глобального поиска; эти индикаторы позволяют демон-

стрировать график минимизируемой функции или ее кусочно-линейной аппроксимации, распределение, плотность и последовательность точек итераций и значений функции в этих точках. Наблюдение за индикаторами в процессе поиска глобального минимума обеспечивает лучшее понимание теории глобальной оптимизации и развивает необходимую интуицию для практического применения и дальнейшего развития методов многоэкстремальной оптимизации;

— *выполнить различные эксперименты* последовательно или одновременно. Последний случай позволяет наглядно сравнивать динамику поиска различными методами и реализуется в режиме разделения времени. Проведение серийных экспериментов, требующих длительных вычислений, может осуществляться в автоматическом режиме с возможностью запоминания результатов поиска для организации последующего анализа полученных данных. Выполнение экспериментов может осуществляться и в режиме ручного поиска, при котором оптимизатор указывает точки итераций самостоятельно (данный режим позволяет пользователю проверить различные предположения, которые могли бы быть положены в основу схем глобального поиска);

— *накапливать и анализировать результаты выполненных экспериментов*. АБСОЛЮТ позволяет оценивать операционные характеристики методов и обеспечивает ведение *журнала* экспериментов для записи результатов оптимизации. Накапливаемые данные могут быть представлены в различных обобщенных формах (таблицы, графики, диаграммы), удобных для проведения анализа. Результаты вычислений могут быть записаны в архив системы АБСОЛЮТ, напечатаны или скопированы в буфер обмена системы Windows в текстовой, графической или табличной формах и перемещены далее в текстовый редактор Word, систему электронных таблиц Excel и другие программы системы Windows для последующей обработки (в т.ч. для использования в подготавливаемых статьях, отчетах и т. п.).

Подробная информация о системе АБСОЛЮТ представлена в [12].

7.2. Программная система ПараЛаб для изучения и исследования методов параллельных вычислений

Программная система Параллельная Лаборатория (сокращенное наименование *ПараЛаб*) обеспечивает возможность проведения вычислительных экспериментов с целью изучения и исследования параллельных алгоритмов решения сложных вычислительных задач. Система может быть использована для организации лабораторного практикума по различным учебным курсам в области параллельного программирования, в рамках которого обеспечивается возможность:

- *моделирования многопроцессорных вычислительных систем*с различной топологией сети передачи данных;
- *получения визуального представления* о вычислительных процессах и операциях передачи данных, происходящих при параллельном решении разных вычислительных задач;
- *построения оценок эффективности* изучаемых методов параллельных вычислений.

Проведение такого практикума может быть организовано на «обычных» однопроцессорных компьютерах, работающих под управлением операционных систем семейства MS Windows (режим многозадачной имитации параллельных вычислений). Кроме режима имитации, в системе ПараЛаб может быть обеспечен удаленный доступ к многопроцессорной вычислительной системе для выполнения экспериментов в режиме «настоящих» параллельных вычислений для сопоставления результатов имитации и реальных расчетов.

В целом система ПараЛаб представляет собой *интегрированную среду для изучения и исследования параллельных алгоритмов*решения сложных вычислительных задач. Широкий набор имеющихся средств визуализации процесса выполнения эксперимента и анализа полученных результатов позволяет изучить эффективность использования тех или иных алгоритмов на разных вычислительных системах, сделать выводы о масштабируемости алгоритмов и определить возможное ускорение процесса параллельных вычислений.

Реализуемые системой ПараЛаб процессы изучения и исследований ориентированы на активное освоение основных теоретических положений и способствуют формированию у обучающихся своих собственных представлений о моделях и методах параллельных вычислений путем наблюдения, сравнения и сопоставления широкого набора различных визуальных графических форм, демонстрируемых в ходе выполнения вычислительного эксперимента.

Основной сферой использования системы ПараЛаб является *учебное применение* студентами и преподавателями вузов для исследования и изучения параллельных алгоритмов решения сложных вычислительных задач в рамках лабораторного практикума по различным учебным курсам в области параллельного программирования. Система ПараЛаб может использоваться также и при проведении *научных исследований* для оценки эффективности параллельных вычислений. Пользователи, начинающие знакомиться с проблематикой параллельных вычислений, найдут систему ПараЛаб полезной для освоения методов параллельного программирования, опытные вычислители могут использовать систему для оценки эффективности новых разрабатываемых параллельных алгоритмов.

Внешний вид главного окна программной системы ПараЛаб, в которой запущена тестовая задача оптимизации, представлен на рис. 7.2.

Возможности системы. ПараЛаб — программная система, которая позволяет проводить как реальные параллельные вычисления на многопроцессорной вычислительной системе, так и имитировать такие эксперименты на одном последовательном компьютере с визуализацией процесса параллельного решения сложной вычислительной задачи.

При проведении имитационных экспериментов ПараЛаб предоставляет возможность для пользователя:

- *определить топологию* параллельной вычислительной системы для проведения экспериментов, *задать число процессоров* в этой топологии, *установить производительность* процессоров, выбрать *характеристики коммуникационной среды* и *способ коммуникации*;

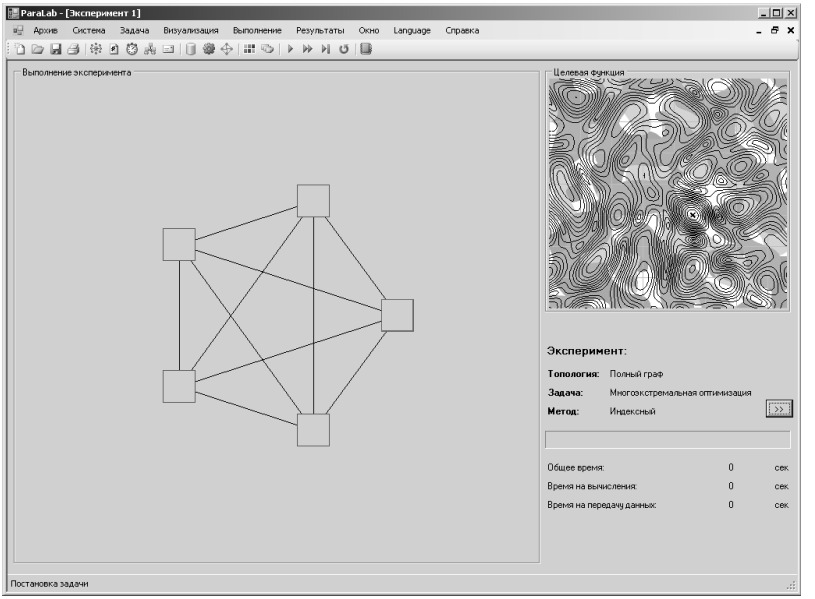

Рис. 7.2. Интерфейс системы ПараЛаб

- *осуществить постановку вычислительной задачи*, для которой в составе системы ПараЛаб имеются реализованные параллельные алгоритмы решения, *выполнить задание параметров* задачи;
- *выбрать параллельный метод* для решения выбранной задачи;
- *установить параметры визуализации* для выбора желаемого темпа демонстрации, способа отображения пересылаемых между процессорами данных, степени детальности визуализации выполняемых параллельных вычислений;
- *выполнить эксперимент*для параллельного решения выбранной задачи; при этом в системе ПараЛаб может быть сформировано несколько различных *заданий для проведения экспериментов*с отличающимися типами многопроцессорных систем, задач или методов параллельных вычислений, для которых выполнение экспе-

римента может происходить одновременно (в режиме разделения времени); одновременное выполнение эксперимента для нескольких заданий позволяет наглядно сравнивать динамику решения задачи различными методами, на разных топологиях, с разными параметрами исходной задачи. При выполнении *серии экспериментов*, требующих длительных вычислений, в системе имеется возможность их проведения в автоматическом режиме с запоминанием результатов для организации последующего анализа полученных данных,

- *накапливать и анализировать результаты выполненных экспериментов*; по запомненным результатам в системе имеется возможность построения графиков, характеризующих параллельные вычисления зависимостей (*времени решения*, *ускорения*, *эффективности*) от параметров задачи и вычислительной системы.

Одной из важнейших характеристик системы является возможность выбора способов проведения экспериментов. Эксперимент может быть выполнен в *режиме имитации*, т. е. проведен на одном процессоре без использования каких-либо специальных программных средств типа библиотек передачи сообщений. Кроме того, в рамках системы ПараЛаб обеспечивается возможность проведения *реального вычислительного эксперимента*.

При построении зависимостей временных характеристик от параметров задачи и вычислительной системы для экспериментов, выполненных в режиме имитации, используются теоретические оценки в соответствии с имеющимися моделями параллельных вычислений (см., например, [10]). Для реальных экспериментов на многопроцессорных вычислительных системах зависимости строятся по набору результатов проведенных вычислительных экспериментов.

Важно отметить, что в системе ПараЛаб обеспечена возможность запоминания результатов проведенных экспериментов в специальной области памяти. Запомненные результаты позволяют выполнить анализ полученных данных; по имеющейся информации любой из проведенных ранее экспериментов может быть восстановлен для повторного выполнения или продолжения расчетов.

Реализованные таким образом процессы изучения и исследований позволят освоить теоретические положения и помогут формированию представлений о методах построения параллельных алгоритмов, ориентированных на решение конкретных прикладных задач. Подробная информация о системе ПараЛаб представлена в [14].

7.3. Программная система GlobalExpert для параллельного решения задач глобальной оптимизации

Система GlobalExpert представляет собой интегрированную программную систему для решения задач глобальной оптимизации на параллельных вычислительных системах. Используя GlobalExpert, исследователь может осуществить постановку задачи оптимизации, выбрать оптимизационный алгоритм из числа реализованных в системе методов глобального поиска, установить графические индикаторы для наблюдения за процессом глобального поиска, выполнить расчет, получить и проанализировать результаты.

Система GlobalExpert ориентирована на прикладное применение техническими и научными работниками в качестве эффективного средства решения многомерных задач глобальной оптимизации. Ключевой характеристикой прикладных задач является значительное время вычисления оптимизируемой функции в одной точке, которая может достигать десятков секунд, поэтому в системе GlobalExpert решена проблема эффективного использования информации, накопленной в процессе поиска.

Система состоит из двух основных модулей: расчетного и интерфейсного. Интерфейсный модуль реализован как MS .NET-приложение (язык разработки — Managed C++) и предназначен для управления расчетами и отображения результатов. Используя интерфейсный модуль, исследователь получает возможность:

1) сформировать задачу оптимизации (задать функции, выбрать из них критерий и ограничения, определить область изменения переменных);

2) провести предварительное исследование задачи (построить сечения и/или линии уровня функций задачи, выделить допустимую области задачи при фиксированных параметрах);

3) выбрать метод оптимизации и задать его параметры;

4) определить способ запуска расчетной части (последовательный, параллельный на одном компьютере, параллельный в локальной сети);

5) в реальном времени наблюдать за процессом поиска;

6) вносить изменения в постановку задачи и/или параметры метода и решать задачу в новой постановке с использованием накопленной ранее поисковой информации.

Внешний вид главного окна интерфейсного модуля представлен на рис. 7.3.

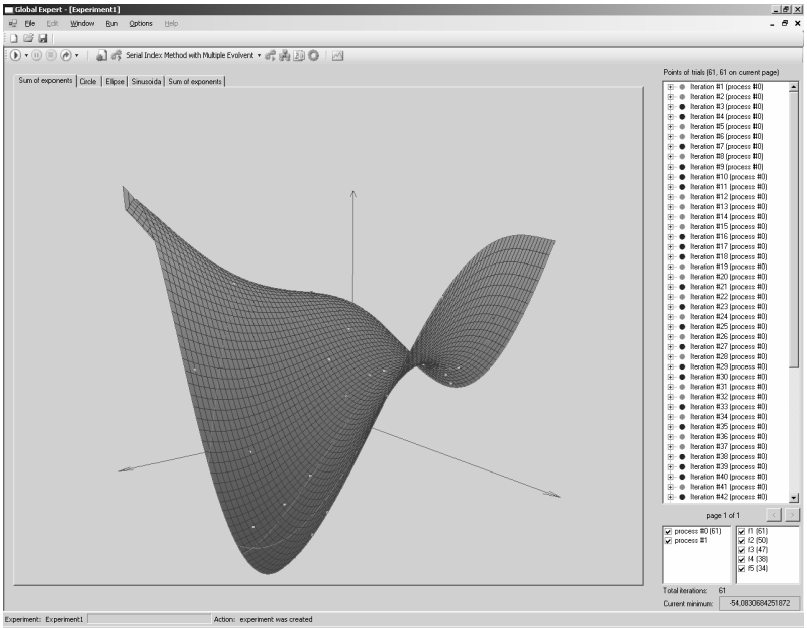

Рис. 7.3. Интерфейс системы GlobalExpert

Расчетная часть реализована как в виде исполняемого файла, так и в виде динамической библиотеки (язык разработки — C++). В виде динамической библиотеки она может быть подключена к существующей программной системе в качестве «решателя» задач оптимизации. В виде исполняемого файл расчетная часть может запускаться либо независимо (результаты отображаются в консольном окне), либо под управлением интерфейсного модуля (результаты передаются в интерфейсный модуль для визуализации и обработки).

При проведении масштабных расчетов рекомендуется следующий порядок работы. Сначала провести предварительное исследование с помощью интерфейсной части, убедиться в правильности постановки задачи и корректности результатов при запуске на одном компьютере или в локальной сети. А для непосредственного проведения расчетов на кластерной системе запустить консольное приложение независимо, через систему управления кластером. С этой целью в интерфейсной части предусмотрена генерация командной строки со всеми параметрами, необходимыми для успешного запуска на кластере.

Возможности системы. GlobalExpert является интегрированной программной средой для решения задач глобальной оптимизации, что позволяет исследователю:

— менять постановку решаемой задачи без потери накопленной поисковой информации (изменение набора ограничений, смена минимизируемой функции, расширение/сужение области поиска);

— использовать ряд информационно-статистических алгоритмов глобального поиска (GlobalExpert содержит несколько последовательных и параллельных алгоритмов, с наилучшей стороны зарекомендовавших себя на практике);

— в удобной форме представлять визуальную информацию о решаемой задаче. Исследователь может строить различные сечения функций задачи, отображать линии уровня минимизируемых функций, распределение точек итераций на графике и/или линиях уровня функции, выделение точек итераций с разных процессоров и многое другое.

— накапливать и анализировать результаты выполненных экспериментов. GlobalExpert обеспечивает ведение журнала экспериментов для записи результатов оптимизации, при этом результаты вычислений включают в себя данные со всех процессоров, на которых была запущена система. Накапливаемые данные представляются в табличной форме, удобной для проведения анализа и последующей обработки.

Отметим, что все результаты экспериментов, приведенные в главе 6, были получены с помощью GlobalExpert. Подробное описание системы представлено в работе [49].

Список литературы

1. *Базара М., Шетти К.* Нелинейное программирование. Теория и алгоритмы. — М.: Мир, 1982.
2. *Баркалов К.А., Стронгин Р.Г.* Метод глобальной оптимизации с адаптивным порядком проверки ограничений // Ж. вычисл. матем. и матем. физ. 2002. Т. 42, № 9. С. 1338–1350.
3. *Васильев Ф.П.* Численные методы решения экстремальных задач. — М.: Наука, 1982.
4. *Гергель А.В, Гнатюк Д.В.* Адаптивные параллельные алгоритмы для многомерной многоэкстремальной оптимизации // Материалы всероссийской конференции «Высокопроизводительные параллельные вычисления на кластерных системах». — Владимир: Изд-во ВлГУ, 2009. С. 92–96.
5. *Гергель В.П. Гришагин В.А., Гергель А.В.* Адаптивные параллельные вычисления для многомерной многоэкстремальной оптимизации // Труды Всероссийской суперкомпьютерной конференции (21–26 сентября 2009 г., г. Новороссийск). — М.: Изд-во МГУ, 2009. С. 417–421.
6. *Гергель А.В.* Адаптивные параллельные вычисления для многомерной многоэкстремальной оптимизации // Известия ВУЗов. Приборостроение. 2009. Т. 52, № 10. С. 74–80.
7. *Гергель В.П. Гришагин В.А., Гергель А.В.* Многомерная многоэкстремальная оптимизация на основе адаптивной многошаговой редукции размерности // Вестник ННГУ. Математическое моделирование и оптимальное управление. — Нижний Новгород: Изд-во Нижегородского гос. ун-та, 2010. № 1. С. 163–170.
8. *Гергель В.П.* Алгоритм глобального поиска с использованием производных // Динамика систем: Межвуз. тематич. сб. науч. тр. / Под ред. Ю. И. Неймарка. — Горький: ГГУ, 1992. С. 161–178.

9. *Гергель В.П.*Об одном способе учета значений производных при минимизации многоэкстремальных функций // Ж. вычисл. матем. и матем. физ. 1996. Т. 36, № 6. С. 51–57.
10. *Гергель В.П.* Теория и практика параллельных вычислений — М.: Бином. Лаборатория знаний, 2007.
11. *Гергель В.П., Стронгин Р.Г.* Система многоэкстремальной оптимизации // Алгоритмы и алгоритмические языки. Пакеты прикладных программ. Программное обеспечение оптимизационных задач. — М.: Наука, 1987. С. 39–50.
12. *Гергель В.П., Стронгин Р.Г.* АБСОЛЮТ. Программная система для исследований и изучения методов глобальной оптимизации. Н.Новгород: Изд. Нижегород. ун-та, 1998.
13. *Гергель В.П., Горбачев В.А., Оленев Н.Н., Рябов В.В., Сидоров С.В.* Параллельные методы глобальной оптимизации в идентификации динамической балансовой нормативной модели региональной экономики // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. 2011. № 25 (242). С. 4–15.
14. *Гергель В.П., Лабутина А.А.* ПараЛаб — среда для проведения вычислительных экспериментов для изучения и исследования параллельных методов решения сложных вычислительных задач // Вестник ННГУ. Информационные технологии. — Нижний Новгород: Изд-во Нижегородского гос. ун-та, 2012. № 3 (2). С. 239–247.
15. *Гермейер Ю.Б.* Введение в теорию исследования операций. — М.: Наука, 1971.
16. *Гнеденко Б.В.* Курс теории вероятностей. — М: Наука, 1988.
17. *Городецкий С.Ю.* Сходимость и асимптотические оценки поведения для одного класса методов поиска // Динамика систем. Динамика и управление. — Горький: Изд-во ГГУ, 1984.
18. *Городецкий С.Ю., Гришагин В.А.* Нелинейное программирование и многоэкстремальная оптимизация. — Н.Новгород: Изд-во ННГУ, 2007.
19. *Гришагин В.А.* Операционные характеристики некоторых алгоритмов глобального поиска // Проблемы случайного поиска. — Рига: Зинатне, 1978. № 7. С. 198–206.
20. *Гришагин В.А.* Сравнительное исследование динамики поиска двух алгоритмов глобальной оптимизации // Динамика систем. Вып. 18 (3). — Горький: Изд-во ГГУ, 1979. С. 76–89.

21. *Гришагин В.А.* Об условиях сходимости для одного класса алгоритмов глобального поиска // Тезисы докладов III Всесоюзного семинара «Численные методы нелинейного программирования». — Харьков: Изд-во ХГУ, 1979. С. 82–84.
22. *Гришагин В.А., Стронгин Р.Г.* Оптимизация многоэкстремальных функций при монотонно-унимодальных ограничениях // Изв. АН СССР. Техническая кибернетика. 1984. № 4. С. 203–208.
23. *Губайдуллин И.М., Рябов В.В., Тихонова М.В.* Применение индексного метода глобальной оптимизации при решении обратных задач химической кинетики // Вычислительные методы и программирование: новые вычислительные технологии. 2011. Т. 12, № 1. С. 137–145.
24. *Ермаков С.М.* Метод Монте-Карло и смежные вопросы. — М.: Наука, 1971.
25. *Жилинскас А.Г.* Одношаговый байесовский метод поиска экстремума функций одной переменной // Кибернетика. 1975. № 1. С. 139–144.
26. *Жилинскас А.Г.* Глобальная оптимизация. Аксиоматика статистических моделей, алгоритмы, применение. — Вильнюс: Мокслас, 1988.
27. *Жиглявский А.А., Жилинскас А.Г.* Методы поиска глобального экстремума. — М.: Наука, 1991.
28. *Иванов В.В.* Об оптимальных алгоритмах минимизации функций некоторых классов // Кибернетика, 1972. Т. 4. С. 81–94.
29. *Карр Ч., Хоув Ч.* Количественные методы принятия решений в управлении и экономике. — М.: Мир, 1964.
30. *Леонов В.В.* Метод покрытий для отыскания глобального максимума функции от многих переменных // Исследования по кибернетике. М.: Советское радио, 1970.
31. *Маркин Д.Л., Стронгин Р.Г.* Метод решения многоэкстремальных задач с невыпуклыми ограничениями, использующий априорную информацию об оценках оптимума // Ж. вычисл. матем. и матем. физ. 1987. Т. 27, № 1. С. 52–61.
32. *Моцкус Й.Б.* О байесовских методах поиска экстремума // Автоматика и вычислительная техника, 1972. № 3. С. 53–62.
33. *Немировский А.С., Юдин Д.Б.* Сложность задач и эффективность методов оптимизации. — М.: Наука, 1979.
34. *Оленев Н.Н., Автухович Э.В., Гуриев С.М., Петров А.А., Поспелов И.Г., Шананин А.А., Чуканов С.В.* Математическая модель экономики переходного периода // Вычислительный центр РАН. М., 1999.

35. *Оленев Н.Н.* Модель оценки инновационного потенциала региональной экономики // Экономика депрессивных регионов: Проблемы и перспективы развития региональных экономик: Труды международной научно-практической конференции / Под ред. В. И. Беляева, И. Н. Дубины, О. П. Мамченко. Барнаул: Изд-во Алтайского гос. ун-та, 2007. С. 178–188.
36. *Пиявский С.А.* Один алгоритм отыскания абсолютного экстремума функции // Ж. вычисл. матем. и матем. физ. 1972. Т. 12, № 4. С. 888–896.
37. *Растригин Л.А.* Статистические методы поиска. — М.: Наука, 1968.
38. *Сергеев Я.Д.* Одномерный детерминированный алгоритм глобальной минимизации // Ж. вычисл. матем. и матем. физ. 1995. Т. 35, № 5. С. 705–717.
39. *Сергеев Я.Д., Квасов Д.Е.* Диагональные методы глобальной оптимизации. — М.: Физматлит, 2008.
40. *Сергеев Я.Д., Стронгин Р.Г.* Алгоритм глобальной минимизации с параллельными итерациями // Ж. вычисл. матем. и матем. физ. 1989. Т. 29, № 3. С. 332–345.
41. *Соболь И.М., Статников Р.Б.* Выбор оптимальных параметров в задачах со многими критериями. — М.: Наука, 1981.
42. *Стронгин Р.Г.* Численные методы в многоэкстремальных задачах. — М.: Наука, 1978.
43. *Стронгин Р.Г., Маркин Д.Л.* Минимизация многоэкстремальных функций при невыпуклых ограничениях // Кибернетика. 1986. № 4. С. 63–69.
44. *Стронгин Р.Г.* Поиск глобального оптимума. — М.: Знание, 1990.
45. *Стронгин Р.Г.* Параллельная многоэкстремальная оптимизация с использованием множества разверток // Ж. вычисл. матем. и матем. физ. 1991. Т. 31, № 8. С. 1173–1185.
46. *Стронгин Р.Г., Баркалов К.А.* О сходимости индексного алгоритма в задачах условной оптимизации с ε-резервированными решениями // Математические вопросы кибернетики. — М.: Наука, 1999. С. 273–288.
47. *Стронгин Р.Г., Гергель В.П., Баркалов К.А.* Параллельные методы решения задач глобальной оптимизации // Известия высших учебных заведений. Приборостроение. Т. 52, № 10, 2009. С. 25–32.
48. *Сухарев А.Г.* Оптимальный поиск экстремума. — М.: Изд-во МГУ, 1975.

49. *Сысоев А.В.* Программный комплекс параллельных вычислений в задачах выбора глобально-оптимальных решений // Вестник ННГУ. Математическое моделирование и оптимальное управление. — Нижний Новгород: Изд-во Нижегородского гос. ун-та, 2012. № 5(2).
50. *Тихонова М.В., Губайдуллин И.М., Спивак С.И.* Численное решение прямой кинетической задачи методами Розенброка и Мишельсена для жестких систем дифференциальных уравнений // Ж. Средневолжского математического общества. 2010. Т. 12, № 2. С. 26–33.
51. *Фихтенгольц Г.М.* Основы математического анализа. — М.: Наука, 1968.
52. *Galperin E.A.* The cubic algorithm // J. Math. Analys. and Appl. 1985. V. 112. P. 635–640.
53. *Grishagin V.A., Sergeyev Ya.D., Strongin R.G.* Parallel characteristical global optimization algorithms // Journal of Global Optimization. 1997. V. 10, No. 2. P. 185–206.
54. *Hansen P., Jaumard B., Lu S.-H.* Global Optimization of Univariate Lipschitz Functions: 2. New Algorithms and Computaional Comparison // Math. Programming. 1989. V. 55. P. 273–293.
55. *Hill J.D.* A search technique for multimodal surfaces // IEEE Trans. Systems Sci. and Cybernetics. 1969. V. 5, No. 1. P. 2–8.
56. *Kumar V., Grama A., Gupta A., Karypis G.* Introduction to Parallel Computing. — The Benjamin/Cummings Publishing Company, Inc., 1994 (2nd edn. 2003).
57. *Kushner H.* A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise // Trans. ASME, ser. D. 1964. V. 86. P. 97–105.
58. *Lucidi S., Piccioni M.* Random tunneling by means of acceptance-rejection sampling for global optimization // Journal of Optimization Theory and Applications. 1989. V. 62(2). P. 255–277.
59. *MacLagan D., Sturge T., Baritompa W.* Equivalent methods for global optimization // State of the Art in Global Optimization / Ed. by C. A. Floudas, P. M. Pardalos. Dordrecht: Kluwer Academic Publishers, 1996. P. 201–212.
60. *Parfenova L.V., Gabdrakhmanov V.Z., Khalilov L.M., Dzhemilev U.M.* On study of chemoselectivity of reaction of trialkylalanes with alkenes, catalyzed with Zr π-complexes // Journal of Organometallic Chemistry. 2009. V. 694, No. 23. P. 3725–3731.

61. *Pinter J.* Global Optimization in Action. — Dordrecht: Kluwer Academic Publishers. The Netherlands, 1996.
62. *Sergeyev Ya.D.* Global one-dimensional optimization using smooth auxiliary functions // Math. Program. 1998. V. 81, No. 1. P. 127–146.
63. *Sergeyev Ya.D.* On Convergence of «Divide the Best» Global Optimization Algorithms // Optimization, 1998. Vol. 44. P. 303–325.
64. *Sergeyev Ya.D., Grishagin V.A.* A Parallel Method for Finding the Global Minimum of Univariate Functions // Journal of Optimization Theory and Applications. 1994. Vol. 80(3). P. 513–536.
65. *Sergeyev Ya.D., Grishagin V.A.* Parallel asynchronous global search and nested optimization scheme // J. Comput. Anal. Appl. 2001. V. 3, No. 2. P. 123–145.
66. *Strongin R.G., Sergeyev Y.D.* Global multidimensional optimization on parallel computer // Parallel Computing, 1992. Vol. 18, No. 11. P. 1259–1273.
67. *Strongin R.G., Sergeyev Ya.D.* Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000.
68. *Shekel J.* Test functions for multimodal search technique // Proc. 5th Princeton Conf. Inform. Sci. Systems. Princeton Univ. Press. 1971. P. 354–359.
69. *Shevtsov I.Y., Markine V.L., Esveld C.* Optimal design of wheel profile for railway vehicles // Proc. 6th International Conference on Contact Mechanics and Wear of Rail/Wheel Systems. Gothenburg, Sweden, June 10–13, 2003. P. 231–236.
70. *Shevtsov I.Y., Markine V.L., Esveld C.* An inverse shape design method for railway wheel profiles. Structural and Multidisciplinary Optimization. Vol. 33, №. 3. Berlin; Heidelberg: Springer, 2007. P. 243–253.

Lobachevsky State University of Nizhni Novgorod

Strongin, R. G., Gergel, V. P., Grishagin, V. A.,
Barkalov, K. A.

Parallel Computing for Global Optimization Problems

Monograph

This monograph is devoted to problematic of complex multiextremal decision-making models which is traditional for professor Strongin's scientific school having worldwide recognition in this field. The book contains necessary theoretical apparatus for construction and analysis of parallel methods. The apparatus stands on information statistical approach to optimization methods' construction, characteristic theory of convergence and efficiency of extremum search methods, methodology of investigated models complexity reduction. A description of optimization methods' computational schemes is given along with theoretic justification and examples of practical application. The tutorial includes an overview of a software system which implements the discussed methods of globally optimal solutions search. This system is used in complex applied problems and also serves as environment for laboratory works and research in educational process.

The book is intended for a wide audience of students (including Ph.D. students) and professionals willing to study and apply parallel methods of global optimization to computationally intensive applied problems.

Recommended by the Academic Board of Faculty of Computational Mathematics and Cybernetics of Lobachevsky State University of Nizhni Novgorod for use in educational process.

This book was created within the Russian Federation President Committee project for modernization and technological development of Russian economy "Development of training system for highly qualified specialists in the area of supercomputer technologies and specialized software".

Contents

Introduction 11

Chapter 1. Global search: from one-dimensional to multi-dimensional problems through dimension reduction **16**

1.1. Global optimization problem 16
1.2. Computational optimization techniques 21
1.3. Multiextremal problems and complexity reduction 32

Chapter 2. Global search algorithms for one-dimensional multiextreamal optimization problems **36**

2.1. Information statistical approach to global search 36
2.2. Basic global search algorithm 41
2.3. Characteristic scheme of global search algorithm representability 49
2.4. Global search acceleration using additional information 54
2.5. From efficient sequential global search algorithms to parallel computing 77

Chapter 3. Global search algorithms for one-dimensional multiextremal optimization problems with non-linear constraints **111**

3.1. Problem statement 111
3.2. Partial computability and index scheme for constraints consideration 111
3.3. Index algorithm of global search 116
3.4. Index algorithm which considers existence of e-reserved solutions 121
3.5. Index method with adaptive order of constraints check 129
3.6. Parallel computing for multiextremal optimization problems with non-linear constraints 141

Chapter 4. Parallel computing for multidimensional multiextremal optimization based on nested optimization scheme **151**

4.1. General outline of nested optimization scheme 151
4.2. Properties of one-dimensional sub-problems of nested optimization scheme 157
4.3. Parallel computing for nested optimization scheme 166
4.4. General adaptive multistep dimension reduction 176
4.5. Adaptive parallel computing for multidimensional multiextremal optimization 195

Chapter 5. Parallel computing for multidimensional multiextremal optimization using Peano curves based dimension reduction **209**

5.1. General scheme of the approach 209
5.2. Multidimensional generalized algorithm of global search 216
5.3. Parallel multidimensional multiextremal methods based on multiple Peano curves 220
5.4. Parallel computing based on new schemes of multiple Peano curves construction 237

Chapter 6. Supercomputer parallel computing for applied optimization problems solution **245**

6.1. Wheel contour optimization for rail transport 245
6.2. Solution of chemical kinetics inverse problem 249
6.3. Identification of dynamic balance normative model of regional economy 253

Chapter 7. Software systems for global optimization **259**

7.1. Software system Absolut for global optimization methods research and study 259
7.2. Software system ParaLab for parallel computing methods research and study 263
7.3. Software system GlobalExpert for parallel solution of global optimization problems 267

References **271**

Учебное издание

Роман Григорьевич Стронгин, Виктор Павлович Гергель,
Владимир Александрович Гришагин,
Константин Александрович Баркалов

Параллельные вычисления в задачах глобальной оптимизации

Редактор *М. С. Сафарова*
Оформление переплета
П. А. Брызгалов, Ю. Н. Симоненко
Корректор *М. С. Сафарова*
Верстка *П. Л. Поляков*

Подписано в печать 30.11.2012
Формат 60×84/16. Гарнитура Times.
Офсетная печать. Бумага офсетная № 1.
Усл. печ. л. 16,28. Тираж 1800 экз.
Изд. № 9910. Заказ №

Ордена «Знак Почета»
Издательство Московского университета
125009, Москва, ул. Б. Никитская 5/7.

Тел.: (495) 629-50-91. *Факс:* (495) 697-66-71
(495) 939-34-93 *(отдел реализации)*
E-mail: secretary-msu-press@yandex.ru

Сайт Издательства МГУ:
www.msu.ru/depts/MSUPubl2005

Интернет-магазин: http://msupublishing.ru

Адрес отдела реализации:
Москва, ул. Хохлова, 11 (Воробьевы горы, МГУ).
E-mail: izd-mgu@yandex.ru Тел.: (495) 939-34-93

Отпечатано в типографии МГУ.
119991, ГСП-1, Москва,
Ленинские горы, д. 1, стр. 15