Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

И. Г. Черноруцкий

МЕТОДЫ ОПТИМИЗАЦИИ

Санкт-Петербург
2012

УДК 681.3.06
ББК 22.18
Ч-49

Черноруцкий И.Г. **Методы оптимизации:** – учеб. пособие / И.Г. Черноруцкий .

Основная направленность данного учебного издания – пользовательский, прикладной аспект. Излагаемые теория, методы и алгоритмы позволят читателю овладеть принципами корректного и обоснованного применения существующих программных систем поддержки принятия решений, а также создавать новые системы.

Учебное пособие предназначено для обучения студентов высших учебных заведений по направлению подготовки магистров «Информатика и вычислительная техника», «Программная инженерия», «Системный анализ и управление».

1. МЕТОДЫ ОПТИМИЗАЦИИ

1.1.ВВЕДЕНИЕ В ПРОБЛЕМУ ОПТИМИЗАЦИИ

1.1.1. Постановка задачи оптимизации

Математически проблема оптимизации описывается следующим образом.

Рассматривается некоторое множество элементов (вообще говоря, произвольной природы) U, называемое множеством допустимых элементов. Пусть задана функция J, отображающая множество U в множество вещественных чисел (такие функции называются функционалами).

Задача 1. Требуется найти такой элемент u* множества U, которому соответствует минимальное значение J(u):

$$u^* \in U, \quad J(u^*) \le J(u), \quad \forall u \in U. \qquad (1.1.1)$$

Если заменить J на (–J), то задача минимизации трансформируется в задачу максимизации и обратно. Везде далее, если не оговорено противное, будем говорить о задачах минимизации. Элемент u* (не обязательно единственный), удовлетворяющий соотношению (1.1.1), называется точкой оптимума или минимизатором J(u) на U. Часто используются следующие обозначения:

$$u^* = \arg\min_U J(u),$$

$$u^* \in \operatorname{Arg}\min_{u \in U} J(u) = \left\{ u \in U \middle| u = \arg\min_U J(u) \right\}.$$

При формулировке задачи (1.1.1) предполагается ограниченность снизу функции J на U. Однако даже в этом случае задача (1.1.1) может не иметь решения, т.е. минимизаторы u* могут отсутствовать среди

элементов множества U. Например, функция $J(u) = e^{u}$, $U = R^1$ (вещественная прямая) ограничена снизу, но не достигает минимума ни в одной из точек на U.

Целесообразно поэтому рассматривать более общую задачу, которая для ограниченных снизу функций всегда имеет решение.

Задача 2. Требуется найти последовательность $\{u^k\}$ элементов (точек) множества U, удовлетворяющую предельному соотношению

$$\lim_{k\to\infty} J(u^k) = \inf_{U} J(u). \qquad (1.1.2)$$

В обеих задачах функционал J называется целевым функционалом. Последовательности, для которых выполняется условие (1.1.2) называются минимизирующими для J(u) на U.

Если существует минимизатор u* для J(u) на U и для минимизирующей последовательности $\{u^k\}$ справедливо соотношение

$$\lim_{k\to\infty} u^k = u^*, \qquad (1.1.3)$$

то минимизирующая последовательность $\{u^k\}$ называется сходящейся.

На **Ошибка! Источник ссылки не найден.** показан пример расходящейся минимизирующей последовательности $\{u^k\} = \{k\}$: При выполнении последнего условия говорят, что имеет место сходимость по функционалу. Для сходящихся минимизирующих последовательностей имеем еще и сходимость по аргументу.

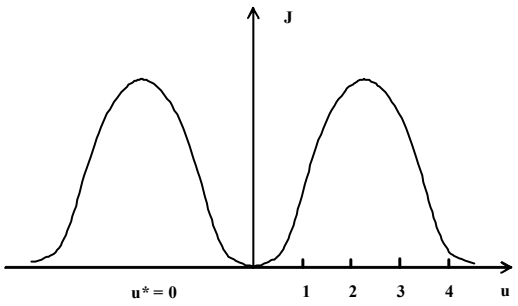


Рис. 1.1.1

Как правило, численные методы оптимизации позволяют строить минимизирующие последовательности, и лишь в том случае, если они

оказываются сходящимися в смысле (1.1.3), можно рассчитывать на получение достаточно хорошего приближения к минимизатору u*.

При практических приложениях теории оптимизации принято различать «задачи аппроксимации» (аппроксимационные задачи оптимизации) и «задачи идентификации» (идентификационные задачи оптимизации). Приведем соответствующие примеры.

Пример 1 (задачи аппроксимации).

Дана система обыкновенных дифференциальных уравнений

$$\frac{dx}{dt} = f(x,p,t), \qquad (1.1.4)$$

где t – время; p – k-мерный вектор параметров. Предполагается, что система (7.1.4) адекватно описывает процессы в некоторой реальной системе. Задача заключается в поиске наилучшего по затратам памяти способа хранения информации о наборе из m экспериментальных кривых, характеризующих реальную систему

$$x^1(t), x^2(t), ..., x^m(t), \ t \in [0,T]. \qquad (1.1.5)$$

Каждая кривая задана набором из N точек вида

$$x^i(t_j), \quad j = 1,2,\ldots,N; \ t_j \in [0,T].$$

Таким образом, всего требуется хранить $m\ell N$ чисел, где ℓ –размерность вектора x (порядок системы (1.1.4)). Предположим, что вектор $\bar{p}_i$ доставляет достаточно малое значение функционалу

$$J(p) = \sum_{j=1}^{N} \left\| x^i(t_j) - x^i(p,t_j) \right\|^2 \quad i = 1,2,...,m, \qquad (1.1.6)$$

где x (p,t) – численное решение (1.1.3), полученное при текущем выборе p. Тогда для хранения заданной экспериментальной информации достаточно хранить m векторов $\bar{p}_i$, что составит mk вещественных чисел. При

необходимости восстановления информации (1.1.5) в этом случае достаточно численно решить систему (1.1.4) при $p = \bar{p}$ и соответствующих начальных условиях.

Если $mk << m\ell N$ или $k << \ell N$, то рассмотренный способ сжатия информации может оказаться оправданным. Совершенно ясно, что в данном случае нас не интересует сходимость по аргументу функционала J(p). Важно лишь обеспечить достаточную точность аппроксимации, т.е. получение достаточно малых значений J(p).

Многочисленные примеры оптимизационных задач, где требуется лишь сходимость по функционалу, дают не только задачи аппроксимации, но также теория и практика оптимального проектирования. В таких задачах целевой функционал J часто отражает качество проекта, а аргументом является некоторый вектор конструктивных параметров. При этом по-прежнему часто оказывается неважным, с помощью какого вектора параметров (из допустимой области) удалось обеспечить заданное качество.

Пример 2 (задачи идентификации).

Рассмотрим теперь несколько иную ситуацию. Пусть по-прежнему задана система (1.1.4), адекватно описывающая некоторую реальную систему, и набор экспериментальных кривых (1.1.5). Допустим теперь, что система (1.1.4) описывает некоторый химико-технологический процесс полимеризации, протекающий в лабораторных условиях, а вектор $\bar{p}$ является вектором скоростей элементарных реакций, принимающих при постоянной температуре конкретные, но неизвестные значения. Основная наша цель при решении задачи (1.1.6) минимизации J(p) состоит в получении достаточно точных оценок этих скоростей, имеющих конкретный физический смысл. Далее полученный вектор $\bar{p}$ будет

использован при моделировании и оптимизации в промышленных условиях режима управления каскадом реакторов по производству полимера (например, ударопрочного сополимера стирола с каучуком). Предполагается, что сам химико-технологический процесс производства полимера описывается другими математическими моделями, отличными от (1.1.4):

$$\frac{dz}{dt} = \varphi(z, p, q, \ldots, t).$$

Здесь p – найденный ранее (на этапе идентификации модели (1.1.4)) вектор оценок скоростей элементарных реакций; t – время; z – вектор фазовых переменных процесса; q – вектор управляющих (режимных) параметров, определяемый как решение некоторой дополнительной задачи оптимизации вида

$$F(q) = \sum_{k=1}^{S} \left\| z(t_k, q) - \bar{z}(t_k) \right\|^2 \to \min_q, \qquad (1.1.7)$$

где – желаемые значения вектора фазовых переменных в заданный момент времени; z (t_k, q) – соответствующие расчетные значения.

Ясно, что если при решении задачи (1.1.6) нас интересует сходимость по аргументу, то в случае (1.1.7) мы должны обеспечить лишь достаточно быструю сходимость по функционалу, как в примере 1.

1.1.2. Терминологические замечания. Классификация задач.

В данном разделе мы рассмотрим некоторые частные случаи сформулированной в разд. 1.1.1 общей задачи поиска минимизатора в абстрактном множестве U (задача 1).

Делая разумные предположения о свойствах функционала J и множества U (например, выпуклость), можно получать полезные

теоретические результаты даже в такой общей постановке вопроса, как задача 1 из разд. 1.1.1. Это один уровень изучения проблемы оптимизации. Можно наделять указанные объекты J, U дополнительными свойствами, используя все более конкретные (частные) математические структуры. В результате мы приходим к целой иерархии теорий, изучающих проблему с различной степенью подробности. Если обратиться к множеству U, то можно наделить его, скажем, структурой банахова или даже гильбертова пространства и строить соответствующую теорию минимизации функционалов в банаховых или гильбертовых пространствах. В данной книге мы пойдем еще дальше и будем считать множество U некоторым подмножеством D конечномерного евклидова пространства R^n над множеством вещественных чисел. Такого типа конечномерные задачи оптимизации называются задачами математического программирования (МП). Итак, МП-задачи – это задачи поиска минимума вещественной функции (функционала) от n вещественных переменных, определенной в некотором допустимом множестве $D \subset R^n$:

$$\begin{gathered} J : D \to R^1 \\ J(x) \to \min_{x \in D}, x = (x_1, x_2, ..., x_n). \end{gathered} \qquad (1.1.5)$$

В данном случае термин «программирование» является неудачным переводом соответствующего английского аналога. Более точный перевод означал бы «планирование» или что-то подобное. Однако терминология в русскоязычной литературе устоялась и нам придется применять общепринятые названия.

Любая терминология вводится для каких-то целей. В данном случае с прикладной точки зрения МП-задачи позволяют оставить в стороне так называемые бесконечномерные задачи оптимизации, изучаемые в основном в рамках теории оптимального управления.

Простейшая задача оптимального управления может быть сформулирована, например, следующим образом.

Управляемый процесс описывается системой обыкновенных дифференциальных уравнений:

$$\frac{dx}{dt} = f(x,u) \quad x(t_0) = x^0, \qquad (1.1.9)$$

где x(t) – n-мерный вектор состояний системы; x^0 – заданное начальное состояние; u(t) – r-мерный вектор управлений: $u(t) = [u_1(t), \ldots, u_r(t)]$, где $u_i(t)$ – функции времени, принадлежащие некоторому заданному множеству функций Ф.

Допустим, что существуют управления $u \in$ Ф, переводящие управляемый процесс из заданного начального состояния x^0 в предписанное конечное состояние $x(t_1) = x^1$ (момент времени t_1 не фиксируется). Такие управления будут составлять некоторое множество $D \subset$ Ф допустимых управлений.

Требуется отыскать такое управление $u \in D$, чтобы минимизировать функционал

$$J(u) = \int_{t_0}^{t_1} F[x(t), u(t), t]dt \to \min_{u \in D}, \qquad (1.1.10.)$$

где F – заданная функция своих аргументов.

В этом случае система (1.1.9) играет роль ограничений на управления u(t) и фазовые переменные x(t). Иногда здесь вводятся еще дополнительные ограничения на допустимые функции u(t).

Сформулированная задача минимизации, очевидно, не является конечномерной, так как функции $u_i(t)$ не определяются, вообще говоря, конечным набором вещественных чисел. Для решения бесконечномерных задач развиты специальные методы и принципы исследования (такие, как принцип оптимальности Беллмана, принцип максимума Понтрягина),

однако в некоторых случаях удается свести бесконечномерную задачу к конечномерной. В качестве примера рассмотрим принцип параметризации. Он состоит в том, что оптимальные управления u(t) ищутся в классе функций, представленных в виде:

$$u(t) = \sum_{i=1}^{N} \alpha_i \varphi_i(t),$$

где $\{ \varphi_1, \varphi_2, \dots, \varphi_N \}$ − заданная система базисных функций.

В этом случае функционал (1.1.10) допускает представление:

$$J(u) = J(\sum_{i=1}^{N} \alpha_i \varphi_i(t)) = J_1(\alpha),$$

т.е. задача минимизации J(u) заменяется конечномерной задачей минимизации $J_1(\alpha)$ в N-мерном пространстве весовых коэффициентов $\alpha = (\alpha_1, \dots, \alpha_N)$.

При реализации такого подхода основные трудности заключаются в выборе базисной системы функций, наиболее соответствующей решаемой задаче. В частности при этом возникает проблема уменьшения размерности вектора α.

Далее мы будем рассматривать только МП-задачи, т.е. конечномерные задачи оптимизации.

Принято различать следующие виды задач математического программирования.

Нелинейное программирование. В задачах нелинейного программирования (НП) предполагается, что множество допустимых значений аргументов D минимизируемой функции (функционала) J задается с помощью систем равенств и неравенств. Таким образом, имеем следующее общее представление для НП-задач:

$$J(x) \to \min_{x \in D},$$
$$D = \left\{ x \in R^n \,\middle|\, g_i(x) \le 0, i = 1,\dots, l; h_j(x) = 0, j = 1,\dots, s \right\}, \qquad (1.1.6)$$

где J, g_i, h_j – функционалы, реализующие отображения $R^n \to R$.

Линейное программирование. Если целевой функционал и функционалы ограничений могут быть заданы с помощью линейных функций, то такие задачи НП называются задачами линейного программирования (ЛП).

Так называемая основная задача ЛП (ОЗЛП) ставится следующим образом.

Требуется найти минимизатор линейной функции

$$J(x) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \sum_{i=1}^{n} c_i x_i = \langle c, x \rangle$$

в допустимом множестве D, элементы которого удовлетворяют ограничениям:

$Ax = b$ (ограничения-равенства); $x_i \geq 0$, $i = 1, \ldots, n$ (ограничения-неравенства), где A – (m × n)-матрица вещественных чисел; b – m-вектор; $\langle \cdot , \cdot \rangle$ – знак скалярного произведения.

Кратко ОЗЛП может быть записана в виде $\min\{cx \mid x \geq 0, Ax = b\}$, где cx – краткая запись скалярного произведения вектора c на вектор x.

Существует несколько эквивалентных форм ЛП-задач. Например, все следующие виды ЛП-задач эквивалентны в смысле сводимости друг к другу:

$\min \{cx \mid x \geq 0,\ Ax = b\}$, (ОЗЛП)

$\min \{cx \mid x \geq 0,\ Ax \geq b\}$,

$\min \{cx \mid\ Ax \geq b\}$,

$\max \{cx \mid x \geq 0, Ax = b\}$,

$\max \{cx \mid x \geq 0, Ax \leq b\}$,

$\max \{cx \mid\ Ax \leq b\}$.

Таким образом, любой метод решения одной из представленных задач может быть преобразован для решения всех остальных. (Существуют и другие эквивалентные формулировки, основанные, в частности, на теореме двойственности.)

ЛП-задачи образуют важный класс НП-задач по крайней мере в двух отношениях. Во-первых, такие задачи характерны для многих практических (особенно экономических) приложений. Во-вторых, для решения ЛП-задач созданы специальные и достаточно эффективные методы, позволяющие в большом числе случаев получать решение за конечное число шагов.

Кроме того, существуют вычислительные технологии решения общих НП-задач на основе их последовательной аппроксимации соответствующими ЛП-задачами.

Выпуклое программирование. Задачи выпуклого программирования, так же как и ЛП-задачи оказываются достаточно удобными для точного математического анализа основных ситуаций. Так называются задачи, в которых целевой функционал и допустимое множество D являются выпуклыми. Существуют свои специфические для данного класса задач методы минимизации, например, методы возможных направлений, линеаризации, двойственные и др. Мы не будем далее рассматривать эти методы, отметим только, что уже такая простая задача, как задача оценки скалярного параметра a скалярного дифференциального уравнения

$$\frac{dx}{dt} = ax, \quad x(0) = x_0$$

по известному экспериментальному значению $x(t_1) = x_{Э1}$ из условия минимума функции $J(a) = [x(t_1, a) - x_{Э1}]^2 \to \min\limits_{a}$ оказывается невыпуклой. Здесь x(t,a) – решение дифференциального уравнения при заданном

произвольном значении параметра a.

Существует еще несколько «программирований», например, «квадратичное», «недифференцируемое» и т.д. Мы не будем здесь приводить соответствующие формулировки, имея в виду представленный выше общий случай НП-задач.

В заключение отметим важный класс задач нелинейного программиро-вания, допускающих весьма специфические и принципиально важные методы решения. Это еще одно «программирование» – динамическое. Методы динамического программирования ориентированы на НП-задачи, в которых целевой функционал имеет специальную сепарабельную структуру, что позволяет трактовать задачу построения минимизатора как некоторую многоэтапную процедуру, в которой результирующее значение целевого функционала является суммой его значений на отдельных этапах. На методах динамического программирования основаны, в частности, многие сетевые методы, когда, например, требуется отыскать минимальный путь на графе и т.п. Общая теория динамического программирования имеет гораздо более широкий спектр приложений.

1.2. ОСНОВНЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОПТИМИЗАЦИИ.

1.2.1. Общая проблема оптимизации произвольной системы.

Математические модели оптимизируемых систем будем характеризовать конечной совокупностью числовых параметров, которые условно можно разделить на три группы: внутренние, внешние и выходные.

Под внутренними параметрами понимаются параметры отдельных элементов, составляющих оптимизируемую систему. Так, при оптимальном проектировании электронной схемы в дискретном исполнении внутренними параметрами являются электрические параметры типа сопротивлений, емкостей, индуктивностей, токов и напряжений источников либо определяющие их другие величины. При расчете интегральных микросхем внутренними параметрами являются не только электрические, но также геометрические и физико-структурные параметры.

Внешние параметры характеризуют влияние внешней среды на оптимизируемый объект. Примерами внешних параметров могут служить параметры входных сигналов, температура окружающей среды, случайные факторы, определяющие шумовое воздействие среды на объекты, и т. д.

Важнейшее значение при описании объектов имеют выходные параметры, отражающие основные свойства и характеристики оптимизируемой системы. В качестве примера выходных параметров некоторой технической системы можно указать потребляемую мощность, быстродействие, габариты, стоимость, оценки точности аппроксимации заданных характеристик, например амплитудно-частотных, и т. д.

Обозначим через $v = (v_1, v_2, \ldots, v_k)$, $w = (w_1, w_2, \ldots, w_i)$ и $y = (y_1, y_2, \ldots, y_m)$

векторы внутренних, внешних и выходных параметров соответственно. Компоненты векторов v, w являются независимыми переменными, определяющими значения зависимых выходных параметров. Таким образом, существует функциональная связь

$$y = \varphi (v, w), \qquad (1.2.1)$$

описываемая некоторым набором алгоритмов, составляющих математическое описание или математическую модель объекта оптимизации. Только в простейших случаях функция (1.2.1) может быть задана с помощью системы явных выражений. В большинстве же случаев связь y с v и w оказывается алгоритмической, что исключает применение аналитических методов исследования.

Не все внутренние параметры являются равноправными: обычно только часть из них может варьироваться в процессе оптимизации. Изменяемые внутренние параметры называются управляемыми параметрами или параметрами оптимизации и образуют вектор $x=(x_1,x_2,\ldots,x_n)$, являющийся подвектором вектора v.

При отсутствии факторов неопределенности все внешние и неизменяемые внутренние параметры принимают известные, заранее заданные значения. В результате от функции (1.2.1) приходим к основной для целей алгоритмической оптимизации зависимости

$$y = F (x), \qquad (1.2.2)$$

осуществляющей связь между вектором параметров оптимизации и основными выходными характеристиками объекта. Выражения, реализующие зависимость (1.2.2), называются детерминированной моделью объекта оптимизации. Весьма часто, однако, оптимизация осуществляется при наличии различных типов неопределенностей. Довольно распространена в задачах оптимизации неопределенность («неопределенность обстановки»), приводящая к усложнению зависимости

(1.2.2) в результате введения в математическую модель некоторого, вообще говоря, неизвестного вектора $z=(z_1, \dots, z_s)$:

$$y=F(x,z). \qquad (1.2.3)$$

При этом отдельные компоненты вектора z могут быть как случайными, так и неслучайными величинами.

Например, составляющие вектора z_i могут описывать влияние:

- случайных отклонений в технологическом процессе при массовом производстве каких-либо технических устройств (технологический разброс параметров);
- изменяющихся, как правило, неслучайным образом условий функционирования оптимизируемой системы, таких, как температура, влажность, вибрация, уровень радиации (при оптимизации технических систем) или уровень спроса на продукцию, параметры фондового рынка (при оптимизации финансовых и экономических систем);
- старения или износа оборудования и т.д.

Все перечисленные причины приводят к тому, что в действительности будем иметь систему, параметры и характеристики которой отличаются от расчетных. Методы учета неопределенности обстановки, применяемые в задачах оптимизации, призваны, с одной стороны, определить степень влияния z на основные характеристики системы, а с другой стороны, по возможности ослабить это влияние.

При случайном характере компонент вектора z модель (1.2.3) называется вероятностной (стохастической) или моделью оптимизации в условиях риска.

Основные принципы однокритериальной оптимизации в условиях неопределенности обстановки были, по существу, изложены в разд. 1.3 «Теории принятия решений» в контексте общей задачи принятия решений.

Непосредственно на основе математического описания оптимизируемой системы решается важнейшая задача оптимизации – задача анализа. Задача анализа заключается в вычислении вектора выходных параметров по заданным значениям всех остальных переменных при фиксированной функционально-структурной модели системы. Методы решения задач анализа определяются природой оператора F, задающего математическую модель объекта оптимизации (функцию реализации – в терминологии разд.1.3 ТПР). Как уже указывалось, оператор F может быть задан в неявной форме и для его реализации необходимо решить одну или несколько стандартных для данного уровня моделирования задач численного анализа. Наиболее часто возникают задачи, связанные с необходимостью решения систем алгебраических и дифференциальных, а также смешанных уравнений. Соответствующие вопросы весьма полно представлены в литературе по численному анализу и теории математического моделирования.

Важная область применения теории и методов оптимизации связана с задачами оптимального проектирования объектов и систем. Под проектированием понимается процесс создания математического описания еще не существующего объекта с целью его последующего изготовления или последующей реализации (воплощения). Используя ранее введенные обозначения, можно сказать, что задача оптимального проектирования по существу распадается на два этапа. На этапе «структурного синтеза» происходит формирование структуры, определяющей элементный (компонентный) состав будущей системы и связи между элементами. Иначе говоря, на этапе структурного синтеза конкретизируется вид функции реализации F в выражении (1.2.3). Задача выбора структуры в настоящее время решается, как правило, неформальными методами. И хотя в отдельных областях применения могут быть развиты регулярные методы

структурного синтеза, в общем виде проблема, по-видимому, неразрешима. Обычно при проектировании систем исходные структуры формируются исходя из имеющегося банка типовых структурных решений, обобщающих предшествующий опыт разработок в соответствующей области. Некоторые обнадеживающие результаты по формализации процесса структурного синтеза устройств с существенно неоднородным базисом проектирования получены при использовании «избыточных структур», для которых ищется частное структурное решение с помощью удаления «лишних» элементов.

Большой опыт проектирования, накопленный в различных областях деятельности, обычно позволяет существенно сузить множество (класс) возможных структурных решений для каждой реальной задачи и поэтому основные трудности возникают на этапе «параметрического синтеза». Задача параметрического синтеза состоит в выборе такого вектора x параметров оптимизации (здесь – параметров проектирования), при которых все выходные характеристики проектируемой системы удовлетворяют требованиям технического задания (списку спецификаций). Эти требования обычно содержат ограничения на входные и выходные параметры; основные типы ограничений рассмотрены ниже.

Как правило, речь идет об оптимальном параметрическом синтезе, так как вектор x выбирается не только из условий правильности функционирования системы, но и из условий обеспечения оптимальности по принятым критериям качества. Задача оптимального параметрического синтеза часто называется задачей параметрической оптимизации или задачей оптимального проектирования.

Задача оптимального параметрического синтеза в конечном счете основана на последовательном решении большого числа задач анализа при различных пробных значениях параметров элементов проектируемой системы. Число необходимых

анализов в среднем оценивается как (100 – 200) n, где n – число варьируемых параметров. Таким образом, уже при n = 10 необходимо выполнить анализ примерно 1000 вариантов проектируемой системы. Это вынуждает выдвигать достаточно жесткие ограничения как на трудоемкость методов анализа, так и на размерность n вектора x. Существенные вычислительные трудности возникают, в частности, если алгоритм анализа содержит процедуру решения одной или нескольких жестких систем дифференциальных уравнений.

Решение задач конечномерной оптимизации вообще и оптимального проектирования в частности (в зависимости от интерпретации) производится с учетом имеющихся ограничений.

Наиболее простой вид имеют так называемые прямые ограничения на компоненты вектора управляемых параметров:

$$a_i \le x_i \le b_i, \qquad (1.2.4)$$

где $[a_i, b_i]$ – заданный допустимый интервал изменения параметра x_i. В более общем случае границы интервалов могут быть функциями от других управляемых параметров, например

$$a_i (x_j) \le x_i \le b_i (x_i) \quad (i \ne j). \qquad (1.2.5)$$

Такие ограничения тоже будем относить к классу прямых.

Ограничения (1.2.4), (1.2.5) часто бывают вызваны причинами, связанными с условиями физической и практической реализуемости, например, с требованиями неотрицательности емкостей, сопротивлений, индуктивностей при проектировании электронных схем или геометрических параметров. Прямые ограничения вытекают также из технологических возможностей производства, определяющих предельно допустимые значения управляемых параметров.

На выходные параметры накладывается два типа ограничений. Функциональные ограничения включают в себя условия работоспособности, имеющие принципиальное значение при оценке правильности функционирования оптимизируемой системы (реальной или

проектируемой) исходя из целей оптимизации и выполнения системой своего функционального назначения. Эти ограничения обычно задаются в виде системы равенств и неравенств

$$y_i \le t_i;\ y_j \ge t_j;\ y_k = t_k, \qquad (1.2.6)$$

где t_i, t_j, t_k – заданные числовые параметры. Если в качестве примера снова обратиться к задачам проектирования, то к функциональным ограничениям могут относиться следующие требования: рассеиваемая в элементах проектируемой электронной схемы мощность должна быть меньше предписанного порогового значения; полюсы передаточной функции фильтра должны лежать в левой полуплоскости; коэффициент обратной связи в схеме генератора должен быть больше критического значения; порог срабатывания ждущей релаксационной схемы должен находиться в заданных пределах и т. д.

Вторая группа ограничений накладывается на выходные параметры, имеющие смысл частных критериев оптимальности и характеризующие качество объекта оптимизации. Наличие подобных частных критериев, по существу, отражает ту неопределенность целей, которая присутствует при оптимизации любой сколько-нибудь сложной системы. Каждый из критериальных выходных параметров желательно максимизировать или минимизировать:

$$y_i \to \max_x;\quad y_j \to \min_x. \qquad (1.2.7)$$

Однако в процессе оптимизации требования (1.2.7) могут быть изменены или дополнены с помощью следующих соотношений, называемых критериальными ограничениями:

$$y_i \ge t_i;\ y_j \le t_j. \qquad (1.2.7)$$

Примерами критериальных ограничений при оптимизации технических систем являются ограничения на стоимость изделия, помехоустойчивость, быстродействие, точность аппроксимации

характеристик, нагрузочную способность, степень устойчивости, время срабатывания и т. д.

Соотношения (1.2.7), (1.2.8) не исключают друг друга. Напротив, как правило, оптимизационные задачи (1.2.7) решаются с учетом ограничений (1.2.8), отражающих требования к характеристикам качества, подлежащие безоговорочному выполнению.

Критериальные ограничения принципиально отличаются от функциональных. Выполнение критериальных ограничений отражает стремление получить оптимальный или близкий к оптимальному вариант системы среди систем, заведомо удовлетворяющих функциональным ограничениям, т. е. функционирующих правильно, в соответствии с предъявляемыми требованиями. В этом смысле можно сказать, что критериальные ограничения по своей сути являются менее жесткими, чем функциональные. Важно, однако, понимать, что грань между функциональными и критериальными ограничениями может быть весьма условной и зависеть от конкретных условий оптимизации.

На критериальные выходные параметры могут одновременно накладываться и функциональные ограничения. Например, в задаче максимизации запаса по устойчивости некоторой системы весьма естественным оказывается требование его неотрицательности.

Список прямых, функциональных и критериальных ограничений составляет основную часть требований на оптимизацию системы. Кроме этого, указываются условия работы системы: характеристики возможных помех и факторов неопределенности среды, диапазоны температуры, давлений, влажности, диапазоны изменения параметров рынка (при оптимизации экономических систем) и т. д. Эти условия формулируются как ограничения на допустимые диапазоны изменения компонент вектора внешних параметров.

Рассмотрим основные методы формальной постановки задачи оптимизации при различных предположениях об условиях и целях оптимизации.

Достаточно общая детерминированная задача оптимизации формулируется следующим образом:

$$f_i(x) \to \min_x; \quad x \in D \quad (i = \overline{1,k}); \tag{1.2.9}$$

$$D = \{x \in R^n \mid g_i(x) \le 0 \ (i = \overline{1,m}); \ g_i(x) = 0$$

$$(i = \overline{m+1,s}); \ a_j \le x_j \le b_j \ (j = \overline{1,q})\}.$$

Множество D называется множеством допустимых значений. В пределах этого множества выполняются прямые, функциональные и критериальные ограничения, представленные в виде общей системы неравенств и равенств. Прямые ограничения удобно указывать в явном виде, так как они учитываются обычно отдельно от других типов ограничений, имеющих более сложную структуру.

Задача (1.2.9) может рассматриваться как формальное представление основных требований к оптимизируемой системе. Предполагается, что каждый из критериальных выходных параметров f_i необходимо минимизировать. Это не ограничивает общности, так как максимизация функции $\varphi(x)$ эквивалентна минимизации $-\varphi(x)$. Кроме этого, нужно учитывать, что замена знака у левых частей неравенств $p(x) \ge 0$ меняет знаки неравенств на противоположные и приводит их к стандартному виду $g(x) \le 0$, где $g(x) = -p(x)$.

Задача (1.2.9) не является стандартной для базовых методов численного анализа из-за наличия векторного критерия оптимальности. Поэтому приобретают важное значение различные приемы ее сведения к однокритериальным задачам, допускающим эффективное численное решение обычными средствами. Такое сведение не является однозначным

и обычно вызывает известные трудности. Вид и форма окончательной постановки задачи во многом определяются конкретными целями оптимизации, а также имеющимися в распоряжении алгоритмическими и программными средствами.

Существующие методы и технологии решения многокритериальных задач оптимизации были, по существу, рассмотрены в разд. 1.2 ТПР

1.2.2. Методы преобразования и учета ограничений.

Рассмотрим методы исключения и преобразования ограничений в общей задаче (8.1.9).

Наиболее просто снимаются прямые ограничения $a_i \le x_i \le b_i$ или $a_i(x_j) \le x_i \le b_i(x_j)$, $i \ne j$. Для этого достаточно заменить переменные по одной из формул, указанных в табл. 1.2.1, где z_i означают новые независимые переменные. Далее будем предполагать, что прямые ограничения отсутствуют.

Существует два класса методов оптимизации, ориентированных на решение однокритериальных задач с ограничениями. Первый класс составляют алгоритмы, реализующие методы проекции градиента, отсечения, а также различные варианты метода возможных направлений. Эти алгоритмы дают возможность на каждой итерации свести исходную задачу к формально более простой задаче с ограничениями, например к задаче линейного программирования.

Во вторую группу входят методы штрафных функций и модифицированных функций Лагранжа, основанные на учете ограничений непосредственно в конструкции критерия оптимальности с последующим использованием алгоритмов безусловной оптимизации. В сложных практических задачах оптимизации чаще применяется второй подход.

Основная идея метода штрафных функций состоит в следующем. Рассмотрим задачу нелинейной оптимизации вида

$$J(x) \to \min_{x \in D}; \quad D = \{x \in R^n \,|\, h_i(x) = 0 \quad (i = \overline{1,q})\}, \qquad (1.2.10)$$

не содержащую ограничений в виде неравенств. Тогда вместо (1.2.10) решается последовательность задач безусловной минимизации однопараметрического семейства функционалов $\{J_k\}$, где

$$J_k(x) = J(x) + \sigma_k \sum_{i=1}^{q} h_i^2(x) \quad (\sigma_k \to \infty, k \to \infty). \qquad (1.2.11)$$

Таблица 1.2.1

Ограничение	Преобразование
$x_i > a_i$	$x_i = a_i + \exp(z_i)$
$x_i \geq a_i$	$x_i = a_i + z_i^2$
$x_i \geq x_j, \quad i \neq j$	$x_j = z_j, \quad x_i = z_j + z_i^2$
$a_i \leq x_i \leq b_i$	$x_i = b_i + (a_i - b_i) \sin^2 z_i$ $x_i = 0{,}5\,(a_i + b_i) + 0{,}5\,(b_i - a_i) \sin z_i$
$a_i < x_i < b_i$	$x_i = b_i + (a_i - b_i)\,\frac{1}{\pi}\,\mathrm{arcctgz}_i$ $x_i = b_i + (a_i - b_i) \exp(z_i) / [1 + \exp(z_i)]$
$a \leq x_i \leq b$ $a \leq x_j \leq b$ $x_i \geq x_j, \quad i \neq j$	$x_j = b + (a - b) \sin^2 z_j$ $x_i = b + (a - b) \sin^2 z_j \sin^2 z_i$
$a < x_i < b$ $a < x_j < b$ $x_i > x_j, \quad i \neq j$	$x_j = b + (a - b)\,\frac{1}{\pi}\,\mathrm{arcctgz}_j$ $x_i = b + (a - b)\,\frac{1}{\pi^2}\,\mathrm{arcctgz}_i\ \mathrm{arcctgz}_j$
$a_i(x_j) \leq x_i \leq b_j(x_j)$ $i \neq j$	$x_j = z_j$ $x_i = b_i(z_j) + [a_i(z_j) - b_i(z_j)] \sin^2 z_i$

$a_i(x_k) \le x_i \le b_i(x_j)$ $i \ne j, \quad i \ne k$	$x_j = z_j$ $x_k = z_k$ $x_i = b_i(z_j) + [a_i(z_k) - b_i(z_j)] \sin^2 z_i$

Второе слагаемое в (1.2.11) имеет смысл «штрафа» за нарушение ограничений, что и определяет название метода. Справедлива следующая теорема.

Теорема 1. Пусть задача (1.2.10) имеет единственное решение x^*; функции J, h_i непрерывны в R^n; для любого $k = 1, 2, \ldots$ существует $x^k = \arg\min J_k(x) \in D \subset R^n$, где D – ограниченное замкнутое множество; $\sigma_k \to \infty$, $k \to \infty$. Тогда $\lim x^k = x^*$, $k \to \infty$.

Доказательства различных утверждений, аналогичных сформулированной теореме, содержатся во многих работах по математическому программированию.

Наиболее распространенный вариант метода штрафных функций для решения общей задачи математического программирования

$$J(x) \to \min; \quad x \in D; \tag{1.2.8}$$

$$D = \{x \in R^n \mid g_i(x) \le 0 \quad (i = \overline{1, m});$$

$$g_i(x) = 0 \quad (i = \overline{m+1, s})\}$$

состоит в применении вспомогательных функционалов

$$J_k(x) = J(x) + \sigma_k \sum_{i=1}^{s} (g_i^+(x))^p, \tag{1.2.13}$$

где

$$g_i^+ = \begin{cases} \max\{g_i(x); 0\} & (i = \overline{1, m}); \\ |g_i(x)| & (i = \overline{m+1, s}). \end{cases}$$

Если функционалы J, g_i являются r раз непрерывно дифференцируемыми на некотором множестве D, то при любом $p > r$ этим же свойством будут обладать функционалы $J_k(x)$.

Основной недостаток метода штрафных функций заключается в ухудшении обусловленности вспомогательных задач при больших σ_k. Соответствующие вопросы рассмотрены далее.

Наиболее перспективным общим методом учета ограничений считается метод модифицированных функций Лагранжа. Применительно к задаче (1.2.10) он формулируется следующим образом.

Введем в качестве обобщенного критерия оптимальности функционал

$$M(x,\xi,\sigma) = J(x) + \langle \xi, h(x) \rangle + 0{,}5\sigma \|h(x)\|^2, \qquad (1.2.14)$$

где $\sigma > 0$ – параметр метода. Тогда алгоритм оптимизации сводится к итерационному процессу

$$x^{k+1} = \arg\min_x M(x, \xi^k, \sigma); \qquad (1.2.15)$$

$$\xi^{k+1} = \xi^k + \sigma h(x^{k+1}); \quad h(x) = [h_1(x), ..., h_q(x)],$$

обобщающему методы штрафных функций и множителей Лагранжа. Основная особенность сформулированного алгоритма по сравнению с методом штрафных функций заключается в отсутствии неограниченно растущего штрафного коэффициента σ, который в данном случае влияет лишь на скорость сходимости, но не на сам факт сходимости последовательности $\{x^k\}$ к оптимуму x^*. При решении практических задач значение σ целесообразно подбирать в интерактивном режиме, так как надежные методы априорного задания σ в настоящее время отсутствуют.

Теоремы о сходимости методов модифицированных функций Лагранжа, а также вычислительные схемы решения общей задачи (1.2.12) рассматривать не будем и отсылаем читателя к списку литературы, представленному в конце настоящего издания.

Иногда при оптимизации возникает необходимость преобразования ограничений-равенств в неравенства и обратно. Неравенства могут быть

сведены к равенствам в результате расширения списка управляемых параметров. Например, ограничение $g(x) \le t$ эквивалентно двум ограничениям $h(\bar{x}) = t,\ x_{n+1} \ge 0$, где $h(\bar{x}) = g(x) + x_{n+1},\ \bar{x} = (x_1, \ldots, x_{n+1})$. Дополнительное прямое ограничение $x_{n+1} \ge 0$ устраняется заменой переменных.

Ограничение-равенство $h(x) = 0$ эквивалентно двум неравенствам $g_1(x) = h(x) \le 0,\ \ g_2(x) = -h(x) \le 0.$

1.2.3. Оптимизация систем в условиях неопределенности.

Нижеследующие сведения в известном смысле дополняют результаты из разд. 3 первой части. Здесь больше внимания уделено алгоритмическим аспектам при решении общих «бесконечномерных» задач (множества изменений аргументов функции реализации могут быть бесконечными).

Рассмотрим математическую модель объекта оптимизации, заданную в виде функции реализации

$$y=F(x,z), \qquad (1.2.9)$$

включающей неопределенный вектор z. В этих условиях, как уже указывалось, методологически целесообразно различать три основные ситуации:

1) z – случайный вектор с известным законом распределения; вектор выходных параметров y реализуется многократно для различных значений вектора z (оптимизация в условиях риска);

2) вектор y реализуется однократно при заранее неизвестном векторе z;

3) выходные параметры y реализуются многократно; вектор z изменяется неизвестным образом, но не является случайным, либо распределение вероятностей z оказывается неизвестным.

Первый случай, например, характерен для анализа технологического разброса параметров при массовом производстве каких-либо изделий. Проектирование уникальных изделий происходит в условиях неопределенности второго типа. Третий вариант возникает при моделировании влияния неконтролируемых параметров внешней среды, определяющих условия эксплуатации и функционирования объекта или системы. Приведенные примеры, разумеется, не исчерпывают все возможные случаи, когда оказывается оправданным следовать приведенным предположениям.

На практике возникают и более сложные ситуации. Например, часть компонент вектора z может иметь случайный характер с известными законами распределения, а некоторые компоненты могут меняться непредсказуемым образом, но не обладать свойством статистической устойчивости.

Обратимся к методам оптимизации при наличии неопределенности первого типа. Наиболее простой путь заключается в переходе от выражения (1.2.16) к зависимости

$$y=F(x,z_M), \qquad (1.2.10)$$

где $z_M=M\{z\}$ – математическое ожидание случайного вектора z. В модели (1.2.17) неопределенность формально отсутствует и могут применяться методы исследования детерминированных моделей. Однако получаемые при этом результаты должны интерпретироваться с учетом вероятностной природы вектора z. При решении задачи (1.2.17) гарантируется лишь оптимальность «в среднем» для достаточно большого числа реализаций z.

Второй возможный подход основан на использовании представления

$$y=M\{F(x,z)\}, \qquad (1.2.11)$$

что соответствует критерию математического ожидания из разд. 1.3.2. ТПР

Понятно, что переход от модели (1.2.16) к (1.2.17) или (1.2.18) является неформальным актом и построение окончательной математической модели должно опираться на дополнительную информацию о задаче и на эвристические представления исследователя о действительных целях оптимизации.

При использовании модели (1.2.18) однокритериальная задача оптимизации может быть сформулирована следующим образом:

$$F_0(x)=M\{J(x,z)\}\to\min,\ x\in D, \qquad (1.2.12)$$

$$D=\{x\in R^n \mid F_i(x)=M\{g_I(x,z)\}\le 0 \quad (i=\overline{1,l})\}.$$

Предполагается, что исходная многокритериальная задача предварительно редуцирована к одной или нескольким задачам (1.2.19) со скалярным критерием качества.

Задача (1.2.19) является задачей стохастического программирования. В отличие от детерминированной постановки функционалы задачи (1.2.19) не заданы в явном виде и для их вычисления необходимо проводить усреднение по z, что, вообще говоря, как уже указывалось, связано с вычислением многомерных интегралов. Последнее приводит к большим вычислительным затратам в случае прямого применения детерминистских методов нелинейного программирования. Более эффективными в ряде случаев оказываются процедуры стохастического программирования, основанные на информации о конкретных реализациях функционалов J (x,z), g_i (x,z), отвечающих различным значениям вектора z.

Один из вариантов подобных методов сводит задачу (1.2.19) к последовательности детерминистских задач:

$$J(x,z^k)\to\min,\ x\in D_k \qquad (1.2.13)$$

$$D_k=\{x\in R^n \mid g_i(x,z^k)\le 0,\ i=1,\ldots,l\}$$

при фиксированных значениях случайного вектора $z = z^k$, $k = 1, 2, ...$ Эти значения должны вырабатываться датчиком случайных чисел в соответствии с заданной плотностью распределения z. Решение задачи (1.2.20), отвечающее вектору z^k, обозначим через x^k. Тогда последовательность векторов $\{\tilde{x}^k\}$, сходящаяся к решению исходной задачи (1.2.19), строится следующим образом: $\tilde{x}^{k+1} = \tilde{x}^k + \alpha_k (x^{k+1} + \tilde{x}^k)$, $\alpha_k > 0$, где $\tilde{x}^2 = x^1 + \alpha_1 (x^2 - x^1)$. Для сходимости процесса необходимо выполнение условий

$$\alpha_k \underset{k\to\infty}{\to} 0; \quad \sum_{k=1}^{\infty} \alpha_k = \infty; \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Этим требованиям удовлетворяет, например, последовательность $\{\alpha_k = 1/k\}$. На практике, однако, возникают проблемы, связанные с эффективным выбором α_k для повышения скорости сходимости метода.

Рассмотрим второй и третий типы неопределенности. Информация о статистических свойствах вектора z, даже если она и имеется, здесь уже не может быть эффективно использована. В указанных условиях целесообразно производить расчет «на наихудший случай», используя принцип гарантированного результата (см. разд. 1.3.3 ТПР) и дополнительную информацию вида $z \in G_z$, где G_z – некоторое ограниченное множество. Соответствующая задача оптимизации формулируется следующим образом:

$$F_0(x) = \max_{z\in G_z} J(x, z) \to \min_{x\in D} \qquad (1.2.14)$$

$$D = \{x \in R^n \,|\, F_i(x) = \max_{z\in G_z} g_i(x, z) \le 0 \quad (i = \overline{1, l})\}.$$

Из выражения (1.2.21) видно, что вычисление функционалов, задающих критерий и ограничения, сопряжено с решением вспомогательных задач оптимизации. В результате трудоемкость

процедуры в целом оказывается достаточно высокой. Аналогичные замечания справедливы и для других критериев, применяемых в условиях данного типа неопределенностей, например для критериев Гурвица (см. разд. 1.3.3 ТПР).

Как видно из изложенного, регулярный учет случайных факторов в процессе решения общей задачи оптимизации (когда множества допустимых значений x и z оказываются бесконечными) алгоритмически достаточно сложен и в настоящее время ограничен лишь относительно простыми ситуациями. Поэтому в сложных случаях чаще вначале решается задача детерминистской оптимизации при фиксированных, например средних, значениях z. Далее в отношении наилучшего детерминистского результата проводится тот или иной вид статистического анализа . Более детально методы статистических расчетов изложены в специальных работах.

1.2.4. Декомпозиция задач оптимизации больших систем.

Под методами разделения или декомпозиции понимаются способы сведения исходной «сложной» задачи к нескольким «простым», поддающимся решению стандартными методами математического программирования. Рассмотрим два характерных подхода.

Алгоритмически наиболее простым является метод введения макроописания объекта оптимизации. Он допускает следующее формальное представление.

Исходная задача формулируется в виде

$$J(x) \to \min_{x \in D}. \qquad (1.2.15)$$

Далее вводятся агрегированные характеристики

$$z_i = \varphi_i (x_1, x_2, ..., x_n), \qquad (1.2.16)$$

где вектор $z = (z_1, z_2, ... , z_s)$ должен иметь существенно меньшую по сравнению с x размерность. Функции φ_i строятся таким образом, чтобы:

1) критерий (1.2.22) был представим в виде суперпозиции отображений

$$J(x) = F[\varphi (x)] = F(z); \; z \in D_z = \varphi (D); \qquad (1.2.17)$$

2) существовали обратные отображения φ_i^{-1}, позволяющие для $\forall z \in D_z$ достаточно эффективно вычислять $x = \varphi^{-1}(z) \in D$.

Приведем одну из возможных «электронных» интерпретаций изложенной формальной конструкции, известную в теории оптимизации технических объектов как метод «аппроксимации и реализации».

Пусть, например, требуется спроектировать электронную схему, имеющую заданную амплитудно-частотную характеристику (АЧХ). Тогда на этапе аппроксимации строится дробно-рациональная передаточная функция W(p, z) комплексного переменного p и вектора $z=(z_1, z_2, ... , z_s)$ коэффициентов полиномов в числителе и знаменателе. Вектор z вычисляется как решение задачи

$$F(z) \to \min_{z \in D_z}, \qquad (1.2.18)$$

где F(z) характеризует близость расчетной и желаемой АЧХ, а множество D_z задается условиями физической реализуемости функции W (p, z). На этапе реализации по найденным z_i^* выбирается структурная схема устройства, конкретизирующая вид функций φ_i в соотношениях (1.2.23). А далее определяется одно из решений системы уравнений

$$\varphi_i (x_1, x_2, ..., x_n) = z_i^* \quad (i = \overline{1, s}). \qquad (1.2.19)$$

Разрешимость системы (1.2.26) следует из выполненных на этапе аппроксимации условий физической реализуемости.

Такое разбиение процесса проектирования на два этапа обычно

мотивируется соображениями удобства, позволяющими на этапе аппроксимации не рассматривать конкретные объекты и получать некоторые общие результаты. Однако не менее важная особенность такого подхода связана с идеей декомпозиции. Обратимся к предыдущему примеру. Пусть трудоемкость решения задачи минимизации функционала линейно зависит от размерности n вектора x и приближенно оценивается числом kn. Допустим также, что основная работа выполняется при вычислении «расстояния» между аппроксимируемой и аппроксимирующей АЧХ. Иначе говоря, коэффициенты z_i по заданным x_i рассчитываются относительно просто, и, напротив, реализация зависимостей $F(z)$, $J(x) = F[\varphi(x)]$ как функций z и x оказывается достаточно трудоемкой.

В этом случае трудоемкость прямого решения задачи $J(x) \to \min$ без разбиения ее на этапы аппроксимации и реализации оценивается числом kn, а трудоемкость этапа аппроксимации – числом ks. Пренебрегая вычислительными затратами на этапе реализации, связанными с получением значений z_i по формулам (1.2.23), получим выигрыш во времени оптимизации в результате декомпозиции приблизительно в n/s раз.

Второй известный принцип декомпозиции связан с сокращением множества D потенциально возможных решений. Это может быть выполнено с помощью введения вектора вспомогательных частных критериев $u(x)=[u_1(x),...,u_m(x)]$. Предполагается, что критерий J(x) удовлетворяет следующему условию монотонности: для любых двух точек $x', x'' \in D$ из системы неравенств $u_i(x') \geq u_i(x'')$, $i = \overline{1,m}$ следует $J(x') \geq J(x'')$. Таким образом, если решение x'' оказывается более предпочтительным, чем x' по векторному критерию u(x), то оно будет

более предпочтительным и с позиций скалярного критерия J(x).

При выполнении условий монотонности исходная задача (1.2.22) может быть заменена следующей

$$J(x) \to \min_{x \in P_u(D)}, \qquad (1.2.20)$$

где $P_u(D)$ – множество решений, эффективных (Парето-оптимальных) по векторному критерию u на множестве D. Очевидно, $P_u(D) \subset D$. Если достигнутое сужение множества D значительно, то задача (1.2.27) оказывается проще исходной и цель декомпозиции считается достигнутой.

Техническая сторона изложенного подхода заключается в следующем. Предполагается, что критерии $u_i(x)$ в отличие от J(x) оказываются эффективно вычислимыми на компьютере с относительно малыми затратами машинного времени. Решая последовательность оптимизационных задач вида

$$\max_i \alpha_i u_i(x) \to \min_{x \in D}; \quad \alpha = (\alpha_1, \alpha_2, ..., \alpha_m) \in \alpha^\varepsilon,$$

где α^ε – некоторая дискретная сетка в множестве

$$A = \{\alpha = (\alpha_1, ..., \alpha_m) \,|\, \alpha_i > 0; \quad \sum_{i=1}^{m} \alpha_i = 1\},$$

мы фактически реализуем некоторую функцию $x(\alpha)$, определяющую точки x, в которых необходимо вычислять исходный глобальный критерий $J[x(\alpha)]$. Число таких точек определяется числом узлов сетки α^ε. Существенно, что размерность пространства векторов α может оказаться значительно меньше (на несколько порядков) размерности исходного пространства векторов x.

Для построения α^ε-сетки может быть использован метод зондирования пространства векторов α, основанный на построении известных из теории математического моделирования $ЛП_\tau$-

последовательностей, обладающих свойством равномерного заполнения заданной многомерной области.

Приведенные подходы далеко не исчерпывают все известные методы декомпозиции. Дальнейшие результаты можно почерпнуть из списка литературы.

1.2.5. Особенности оптимизационных задач.

Возникающие на практике оптимизационные задачи обладают особенностями по сравнению с общей постановкой задачи нелинейного программирования, что необходимо учитывать и использовать при проведении реальных вычислений. Основные характерные черты заключаются в следующем.

1. Алгоритмическое задание функционалов, задающих критерии и ограничения, существенно увеличивает трудоемкость их вычисления и вынуждает ограничиваться методами оптимизации, не использующими в явном виде выражения для производных.

2. Критерии оптимальности, применяемые в реальных задачах оптимизации, часто имеют характерную структуру, позволяющую, например, строить специальные методы оптимизации второго порядка, использующие упрощенные выражения для вторых производных.

3. Однократное вычисление функционалов задачи связано с достаточно сложным и трудоемким решением соответствующей задачи анализа. Наиболее эффективными целесообразно считать алгоритмы, которые в процессе оптимизации наименьшее число раз обращаются к вычислению значений минимизируемых функционалов и ограничений для получения решений с требуемой точностью.

4. Если оптимизируется сложная многопараметрическая система, то ее обычно можно представить как некоторую совокупность связанных подсистем меньшей размерности. Учет подобной структуры системы позволяет строить более рациональные методы оптимизации по сравнению с традиционными универсальными алгоритмами нелинейного программирования.

5. Невыпуклая структура минимизируемых функционалов существенно понижает эффективность обычных методов нелинейной оптимизации, особенно если такой структуре сопутствует описываемая ниже овражная ситуация.

6. Как свидетельствует практика оптимизационных расчетов, возникающие оптимизационные задачи являются, как правило, плохо обусловленными. Это определяет характерную овражную структуру поверхностей уровня минимизируемых функционалов и вызывает резкое замедление сходимости стандартных методов оптимизации.

Указанные характерные черты оптимизационных задач определяют конкретные требования к практическим методам оптимизации и использующим их программным системам оптимизации. С позиций существующего аппарата нелинейной оптимизации наиболее существенными оказываются особенности, отмеченные в пп. 4, 5. Основная трудность состоит в том, что математически проблема невыпуклости минимизируемого функционала оказывается неразрешимой в силу сложности класса невыпуклых оптимизационных задач. Основной вывод заключается в том, что даже для гладких одноэкстремальных функционалов в задачах с не очень малой размерностью пространства управляемых параметров скорость сходимости любого метода (равномерно по всем задачам) безнадежно мала и попытка построить общий метод, эффективный для всех задач с гладкими невыпуклыми

целевыми функционалами, заранее обречена на неудачу.

Однако с позиций специалиста по реальным вычислениям представляет интерес задача построения методов оптимизации, вырабатывающих эффективные направления поиска в точках пространства, где стандартные процедуры оказываются неработоспособными. В последующих главах книги эта проблема решается на основе построения методов, которые как в выпуклой, так и в невыпуклой и одновременно овражной ситуации локально (для квадратичной модели) дают существенно более удовлетворительные по скорости убывания функционала результаты по сравнению с традиционными методами.

О правомерности развиваемого подхода и целесообразности использования соответствующих алгоритмов можно судить только по результатам решения реальных задач. Создание новых методов будет оправданно, если их применение окажется эффективным для заведомо непустого множества практических ситуаций, вызывающих трудности для известных поисковых процедур. В данном случае такое множество можно указать заранее – это множество задач с целевыми функционалами, близкими к кусочно-квадратичным, не обязательно выпуклым зависимостям. Типичность подобных функций подтверждается практикой решения реальных задач. Речь, следовательно, идет не о замене традиционных методов новыми, а о некотором существенном расширении уже имеющегося арсенала методов и алгоритмов оптимизации.

Подтверждаемая экспериментально достаточно высокая работоспособность рассматриваемых методов никак не противоречит тезису о безнадежной трудности невыпуклых задач. Дело, по-видимому, заключается в том, что на практике достаточно редко реализуются те специальные структуры невыпуклых задач, которые и приводят к

пессимистическим теоретическим оценкам. Заметим, что именно так обстоит дело со знаменитым симплекс-методом линейного программирования: строгий теоретический анализ показывает (вопреки практике) его полную непригодность.

1.2.6. Некоторые стандартные схемы оптимизации.

На основе сделанных выше замечаний в зависимости от реальной ситуации могут формироваться различные вычислительные схемы оптимизации. Ниже рассмотрены некоторые варианты таких схем, охватывающие значительное число практических задач.

Задачи аппроксимации. В большом числе случаев задача оптимизации некоторой системы состоит в реализации заданной зависимости некоторой величины W от непрерывной переменной s, например частоты или времени. В качестве таких зависимостей при оптимизации реальных объектов могут выступать амплитудно-частотные, фазочастотные, переходные и другие характеристики. Необходимо подобрать вектор управляемых параметров таким образом, чтобы «расстояние» между заданной и расчетной характеристиками было минимальным.

К задачам аппроксимации относятся также многочисленные задачи идентификации, т. е. задачи выбора параметров моделей реальных объектов по экспериментально полученным характеристикам.

Критерии оптимальности в этих случаях чаще всего формируются одним из следующих способов:

$$J_1(x) = \sum_{i=1}^{N} \alpha_i^2 [W(x, s_i) - W^*(s_i)]^2 \to \min_{x \in D}; \qquad (1.2.21)$$

$$J_2(x) = \max_{i=1,N} \alpha_i \left| W(x, s_i) - W^*(s_i) \right| \to \min_{x \in D}, \qquad (1.2.22)$$

где D – множество допустимых значений x_i; α_i – весовые коэффициенты, определяющие необходимую точность аппроксимации в отдельных точках диапазона изменения независимой переменной s; s_i, $i = \overline{1, N}$ – дискретная сетка значений s, при которых происходит сравнение заданной W*(s) и расчетной W(x,s) характеристик.

Каждый из приведенных критериев имеет свои особенности, существенные для организации вычислительного процесса. Функционал J_1 достаточно прост и обладает свойством «гладкости». Именно, если функция W(x,s) является дважды непрерывно дифференцируемой функцией x, то этим же свойством будет обладать зависимость $J_1(x)$, что существенно облегчает последующую процедуру оптимизации. Основная характерная черта J_1 заключается в ограниченной точности аппроксимации отдельных слагаемых. Иначе говоря, плохая точность аппроксимации в некоторых точках при больших значениях N может компенсироваться хорошей точностью в других точках. Иногда эта ситуация не соответствует смыслу решаемой задачи Эта особенность отсутствует в критерии (1.2.29), однако он не сохраняет характеристики гладкости функции W(x,s), что требует привлечения специальных методов оптимизации.

Как показывает практика, достаточно простой и надежный способ решения задач аппроксимации заключается в использовании гладких среднестепенных аппроксимаций минимаксного критерия J_2. Согласно этому подходу вместо решения задачи (1.2.29) ищется минимум функционала со среднестепенной структурой:

$$J_3(x) = \sum_{i=1}^{N} \varphi_i^{\nu}(x) \to \min_{x \in D} \quad (\nu = 2, 3, ...), \qquad (1.2.23)$$

где $\varphi_i(x) = \alpha_i \left| W(x, s) - W^*(s_i) \right|$.

При достаточно больших значениях ν решения задач (1.2.29), (1.2.30) будут почти совпадать. Действительно, справедливо предельное соотношение

$$\left(\sum_{i=1}^{N} \varphi_i^{\nu}\right)^{1/\nu} \to \max_i \varphi_i, \quad \nu \to \infty,$$

где $\varphi_i \geq 0$, $i = \overline{1,N}$ – произвольные числа. Кроме этого можно показать, что операция извлечения корня ν-й степени не влияет на локализацию точки минимума.

Функционал J_3 совмещает в себе особенности функционалов J_1, J_2. Являясь гладким подобно J_1, J_3 не допускает значительных отклонений точности аппроксимации в отдельных точках. Иногда такой подход оказывается наиболее адекватным истинным целям моделирования.

При решении практических задач на основе критерия J_3 целесообразно пошаговое увеличение параметра ν, начиная с $\nu = 2$. Таким способом обычно удается избежать переполнения разрядной сетки компьютера при возведении первоначально больших значений локальных ошибок аппроксимации φ_i в высокую степень ν. Кроме этого, проводя в интерактивном режиме оценку получаемых в процессе увеличения ν решений, можно вовремя прервать процесс, если получены удовлетворяющие разработчика результаты. Заранее задать оптимальное значение ν обычно трудно. Как правило, при практических расчетах значение ν не превышает 10 –15.

Рассмотренный подход очевидным образом распространяется на вектор-функцию W. При этом в качестве функций φ_i (x) могут использоваться зависимости $\varphi_i(x) = \alpha_i \|W(x, s_i) - W^*(s_i)\|$..

Задачи решения систем неравенств. Задачи моделирования и оптимизации реальных или проектируемых систем часто могут быть

представлены как задачи решения систем неравенств. Так, например, к системам неравенств приводят формализации задач проектирования, цель решения которых заключается в увеличении процента создаваемых при массовом производстве годных изделий в условиях статистического разброса параметров. Пусть технические требования (ТТ) к проектируемому устройству выражаются системой функциональных и критериальных ограничений

$$y_i(x) \le t_i \quad (i = \overline{1,m}). \qquad (1.2.24)$$

Годным считается изделие, выходные параметры y_i которого удовлетворяют соотношениям (1.2.31).

Основная трудность заключается в том, что номинальные значения x^*_i, т.е. значения, на которые настроен соответствующий технологический процесс, могут быть выбраны так, что ТТ выполняются. Однако из-за случайных отклонений параметров в процессе производства некоторые из ТТ могут оказаться нарушенными. Ставится задача такого выбора вектора x^*, чтобы случайные отклонения в технологическом процессе, а также в условиях эксплуатации устройства в наименьшем числе случаев приводили бы к нарушению ТТ.

Известен достаточно простой и эффективный подход к решению сформулированной задачи, которая может иметь и иную интерпретацию. Потребуем, чтобы ТТ выполнялись с некоторыми запасами

$$y_i(x) + \delta_i \le t_i; \quad \delta_i > 0, \qquad (1.2.25)$$

где δ_i характеризует рассеяние i-го выходного параметра в результате статистических вариаций внутренних и внешних параметров. Требование (1.2.32) эквивалентно неравенству

$$z_i(x) = \{[t_i - y_i(x)]/\delta_i\} - 1 \ge 0. \qquad (1.2.26)$$

Величина z_i называется «запасом работоспособности» по i-му

выходному параметру. В результате имеем многокритериальную задачу.

$$z_i(x) \to \max_{x \in D'} \quad (i = \overline{1,m}). \qquad (1.2.27)$$

Здесь D' – множество, в котором выполняются прямые ограничения на управляемые параметры. Предполагается, что функциональные и критериальные ограничения учтены в результате расширения системы неравенств (1.2.31). С помощью соответствующей замены переменных или перевода прямых ограничений в ранг функциональных можно избавиться от ограничений и принять $D' = R^n$.

В практике оптимизации получила распространение максиминная свертка векторного критерия (1.2.34), приводящая к следующему обобщенному показателю качества:

$$J_4(x) = \min_{i=1,m} z_i(x) \to \max_{x \in R^n}. \qquad (1.2.28)$$

Иногда в выражение (1.2.35) вводятся весовые коэффициенты α_i:

$$J_4(x) = \min \alpha_i z_i(x) \to \max.$$

Их роль заключается во введении некоторого стабилизирующего фактора. Действительно, увеличение некоторого весового коэффициента α_i усиливает влияние запаса z_i на результирующую целевую функцию. В результате уже незначительное нарушение соответствующего неравенства приводит к существенному ухудшению целевой функции. С другой стороны, уже при незначительных положительных значениях z_i имеем запас в выполнении i-го неравенства, сравнимый с запасами по остальным выходным параметрам.

Выбор параметров δ_i по существу определяет единицы измерения разностей $t_i - y_i(x)$. Этот выбор обычно облегчается конкретным физическим смыслом δ_i, которые, как правило, могут характеризоваться как характеристики рассеяния. Для их определения может проводиться

статистический анализ в окрестности текущей точки x, что позволяет говорить о превращении показателя (1.2.35) в статистический критерий. Значения δ_i обычно имеют смысл трехсигмовых допусков, которые периодически уточняются в процессе оптимизации. Весьма часто величины δ_i задаются как исходные данные на основе априорной информации, что заметно сокращает трудоемкость процедуры оптимизации.

Функционал J_4, так же как и J_2, не является гладким, что существенно усложняет ситуацию и требует применения специальных оптимизирующих процедур. Ниже излагается альтернативный подход, основанный на процедуре сглаживания исходного функционала с последующим обращением к методам гладкой оптимизации.

Очевидно, $\arg\min_i z_i = \arg\max_i [\exp(-z_i)]$. Поэтому задача (1.2.35) эквивалентна задаче

$$\max_i [\exp(-z_i)] \to \min_x . \qquad (1.2.29)$$

Для задачи (1.2.36) применима среднестепенная свертка (1.2.30), если принять $\varphi_i(x) = \exp[-z_i(x)]$. В результате приходим к следующему критерию оптимальности:

$$J_5(x) = \sum_{i=1}^{m} \exp[-\nu z_i(x)] \to \min_x \quad (\nu = 1, 2, ...). \qquad (1.2.30)$$

Как показывает вычислительная практика, трудности оптимизации систем по критериям типа минимального запаса работоспособности (1.2.35) часто возникают из-за негладкости критериев, приводящей к преждевременной остановке поисковой процедуры. Целесообразно сразу обращаться к модифицированным критериям (1.2.37) с применением на первом этапе простейших алгоритмов оптимизации типа метода простого покоординатного спуска.

Использование среднестепенных критериев оптимальности в задачах оптимизации, где, по существу, необходим минимаксный подход, оправдано также с позиций рассмотренного явления плохой обусловленности. Развитая в настоящее время техника решения негладких оптимизационных задач достаточно сложна и в невыпуклой овражной ситуации многие алгоритмы теряют эффективность. В то же время излагаемые далее методы позволяют получать удовлетворительные результаты для невыпуклых овражных функционалов при условии их гладкости. При этом удается использовать структурные особенности функционалов (1.2.30), (1.2.37) для увеличения эффективности соответствующих вычислительных процедур.

1.3. ПРОБЛЕМА ПЛОХОЙ ОБУСЛОВЛЕННОСТИ

1.1.2. Явление овражности

В этой главе анализируются часто возникающие на практике случаи получения неудовлетворительных результатов с помощью стандартных методов конечномерной оптимизации, применяемых в задачах моделирования, численного эксперимента и других задач системного анализа (см. пример во введении ТПР). Как правило, это выражается в резком увеличении затрат машинного времени, а в некоторых случаях в невозможности получения приемлемых результатов из-за полной остановки алгоритма задолго до достижения оптимальной точки.

Возникновение подобных трудностей связывается далее со специальной формой плохой обусловленности матрицы вторых производных минимизируемых целевых функционалов, приводящей к характерной овражной структуре поверхностей уровня критерия оптимальности.

Рассмотрим критерий оптимальности, зависящий от двух управляемых параметров x_1, x_2:

$$J(x_1, x_2) = g_0^2 (x_1, x_2) + \sigma g_1^2 (x_1, x_2) \to \min_x, \qquad (1.3.1)$$

где σ – достаточно большое положительное число. Рассмотрим также уравнение

$$g_1 (x_1, x_2) = 0, \qquad (1.3.2)$$

определяющее в простейшем случае некоторую зависимость $x_2 = f(x_1)$. Тогда при стремлении параметра σ к бесконечности значение функционала J в каждой точке, где $g_1 (x_1, x_2) \neq 0$, будет неограниченно возрастать по абсолютному значению, оставаясь ограниченным и равным $g_0^2 (x_1, x_2)$ во

всех точках на кривой $x_2 = f(x_1)$. То же самое будет происходить с нормой вектора градиента $J'(x)=(\partial J/\partial x_1, \partial J/\partial x_2)$,где

$$\partial J/\partial x_1 = 2g_0(x_1,x_2)\partial g_0/\partial x_1 + 2\sigma g_1(x_1,x_2)\partial g_1/\partial x_1 ,$$

$$\partial J/\partial x_2 = 2g_0(x_1,x_2)\partial g_0/\partial x_2 + 2\sigma g_1(x_1,x_2)\partial g_1/\partial x_2 .$$

Линии уровня $J(x) = \text{const}$ для достаточно большого σ представлены на рис. 1.3.1. Там же стрелками показано векторное поле антиградиентов, определяющее локальные направления наискорейшего убывания $J(x)$.

Ясно, что минимальные значения $J(x)$ следует искать вдоль зависимости $x_2 = f(x_1)$, определяющей так называемое дно оврага. Из формулы (9.1.1) следует, что изменение $J(x)$ вдоль дна задается выражением $g_0^2(x_1, x_2)$ и не зависит от значения параметра σ. Таким образом, задача минимизации $J(x)$ сводится к минимизации функционала $g_0^2(x_1, f(x_1))$ от одной переменной x_1 . В общем случае уравнение $x_2 = f(x_1)$ обычно неизвестно.

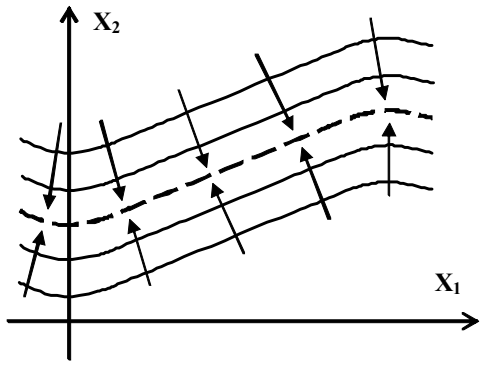


рис. 1.3.1

Приведенный пример овражной структуры критерия оптимальности является достаточно простым, хотя и из него уже видны принципиальные трудности, связанные, например, с применением широко распространенных методов спуска по антиградиенту.

Действительно, из рис. 1.3.1. следует, что направления поиска, задаваемые антиградиентами, оказываются неэффективными. Приводя достаточно быстро процесс поиска на дно оврага, эти направления в окрестности дна начинают

осциллировать, оставаясь почти перпендикулярными направлению, указывающему в точку минимума J (x).

Возможны различные усложнения и обобщения рассматриваемой ситуации. Например, уравнение (1.3.2) может, вообще говоря, определять не одно, а несколько решений $x_2 = f(x_1)$, каждое из которых определяет свой овраг. Более существенным обобщением является предположение о наличии многомерного дна оврага. Чтобы проиллюстрировать это явление, обратимся к следующему примеру:

$$J(x) = g_0^2(x) + \sigma\sum_{i=1}^{m} g_i^2(x); \quad x \in R^n; \quad n > 2; \quad m < n. \qquad (1.3.3)$$

В этом случае дно оврага задается системой уравнений

$$g_i(x) = 0, \quad (i = \overline{1, m}), \qquad (1.3.4)$$

что в принципе позволяет выразить m параметров x_i через оставшиеся n – m переменных. Предположим, не ограничивая общности, что уравнения (1.3.4) определяют следующие зависимости:

$$x_1 = f_1(x_{m+1}, \ldots, x_n);$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$x_m = f_m(x_{m+1}, \ldots, x_n).$$

Аналогично предыдущему случаю устанавливаем, что задача минимизации (1.3.3) при достаточно больших σ эквивалентна минимизации функции $g_0^2(f_1, \ldots, f_m, x_{m+1}, \ldots, x_n)$ от $r = n - m$ переменных $x_{m+1}, \ldots, x_n$. Число r называется размерностью дна оврага. Легко представить, что, например, для n=3 поверхности уровня одномерного оврага имеют характерный «сигарообразный» вид, а для двухмерного оврага они будут близки к деформированным дискам.

В любом случае для овражной ситуации определяющим фактором является специальная структура поверхностей уровня J(x), сильно

отличающаяся от сферической. Характерно наличие некоторой области притяжения (дно оврага) $Q \subset R^n$, содержащей оптимальную точку x*=argminJ(x). При этом норма вектора градиента J'(x) для $x \in Q$, как правило, существенно меньше, чем в остальной части пространства.

Овражную структуру могут иметь не только функционалы вида (1.3.1), (1.3.3), явно содержащие большой параметр σ. Можно привести следующий пример квадратичного функционала:

$$f(x_1, x_2) = 0{,}250025\, x_1^2 + 0{,}49995\, x_1 x_2 + 0{,}250025\, x_2^2 - x_1 - x_2 . \qquad (1.3.5)$$

Линии уровня f (x) = const функционала (1.3.5) представляют семейство подобных эллипсоидов с центром в точке (1;1). Длины полуосей эллипсоидов относятся при этом как 1:100. Указать малый параметр в выражении (1.3.5) нельзя, хотя овражная ситуация налицо и так же, как и в предыдущих случаях, явно выделяется дно оврага (прямая ab на рис 1.3.1), описываемое уравнением $x_2 = -x_1 + 2$. Подставляя выражение для x_2 в (1.3.5), снова приходим к эквивалентной задаче меньшей размерности

$$f_1(x_1) = 10^{-4}(x_1 - 1)^2 - 1 \to \min.$$

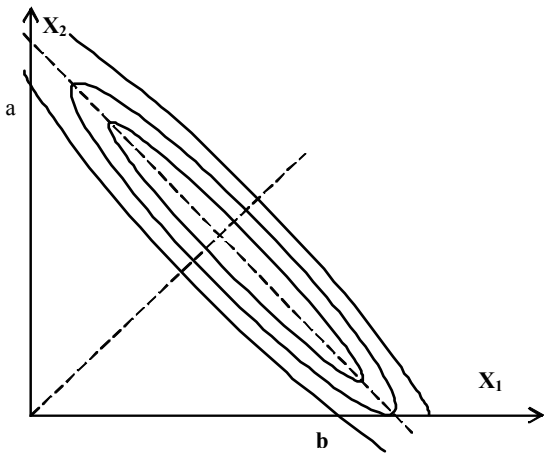


рис 1.3.1

Необходимость выделения овражных оптимизационных задач в отдельный класс обусловлена, с одной стороны, значительными вычислительными трудностями при их решении стандартными для практики моделирования и вычислительного эксперимента методами, а с другой стороны, бесспорным фактом важности данного класса задач для большинства приложений и особенно для задач оптимального выбора параметров как

реально существующих систем – технических, экономических, экологических, так и вновь создаваемых систем, находящихся в стадии проектирования.

Существуют различные методы, ориентированные на решение рассматриваемых оптимизационных задач, однако и в настоящее время проблема минимизации овражных функционалов является актуальной. Особенно важно решить вопрос минимизации овражных и одновременно невыпуклых функционалов, так как именно в этой ситуации «отказывает» большинство из известных методов конечномерной оптимизации.

1.3.2. Формальное определение. Критерии овражности целевого функционала.

Пусть решается задача $J(x) \to \min; J \in C^2(D); x \in D \subset R^n$. Будем предполагать далее, что функционал J(x) ограничен снизу на множестве D.

Траектория наискорейшего спуска (ТСН) $x(\tau)$ функционала J(x) задается известным векторным дифференциальным уравнением

$$dx / d\tau = -J'(x);$$

$$J'(x) = (\partial J / \partial x_1, \ldots, \partial J / \partial x_n). \qquad (1.3.6)$$

Эти траектории обладают специфическими чертами. Например, для функционала (1.3.5) имеем

$$x(\tau) = \sum_{i=1}^{2} [\alpha_i^* + (\alpha_i^0 - \alpha_i^*) \exp(-\lambda_i \tau)] u_i, \qquad (1.3.7)$$

где

$$x(0) = \sum_{i=1}^{2} \alpha_i^0 u_i; \; x^* = (1;1) = \sum_{i=1}^{2} \alpha_i^* u_i;$$

$$\lambda_1 = 1; \lambda_2 = 10^{-4}; u_1 = 1/\sqrt{2}(1;1); \quad u_2 = 1/\sqrt{2}(-1;1).$$

Из выражения (1.3.7) видно, что ввиду наличия быстро затухающей и медленно затухающей экспонент отчетливо выделяются два участка с существенно различным поведением решения. Первый, сравнительно непродолжительный, характеризуется большими значениями производных $dx_i(\tau)/d\tau$ и означает спуск на дно оврага. На дне выполняются условия типа (1.3.2) и норма вектора градиента, а с ней и производные $dx_i(\tau)/d\tau$ становятся относительно малыми. Поэтому для второго участка характерно относительно плавное изменение переменных x_i. Таким образом прослеживается полная аналогия с поведением решений так называемых жестких систем обыкновенных дифференциальных уравнений. В связи с этим Ю.В. Ракитским было предложено следующее общее определение.

Определение 1. Функционал J(x) называется овражным, если отвечающая ему система дифференциальных уравнений (1.3.6) – жесткая.

Однако при решении задач оптимизации более конструктивным оказывается приведенное ниже определение овражного функционала. Это определение не содержит, в частности, таких неестественных для задач оптимизации требований, как необходимость задания промежутка интегрирования уравнения (1.3.6), что предполагается в теории жестких систем.

Определение 2. Функционал $J(x) \in C^2(D), D \in R^n$ называется овражным (жестким) в множестве $Q \subset D$, если найдутся такие числа $\delta > 0$, $\sigma >> 1$, что

$$1)\ \forall x \in Q_\delta:\quad \lambda_1[J''(x)] \geq \sigma \left|\lambda_n[J''(x)]\right|;$$

$$2)\ \forall x \in Q:\quad \operatorname{Arg} \min_{x' \in X_\delta(x)} J(x') \subset X_\delta(x) \bigcap Q; \qquad (1.3.8)$$

$$3)\ \forall x \in Q:\quad L[X_\delta(x) \bigcap Q] \leq \sigma^{-1} L[X_\delta(x)],$$

где $C^2(D)$ – множество дважды непрерывно дифференцируемых на D функционалов;

$$X_\delta(x) = \left\{x' \in R^n \middle| \|x' - x\| \le \delta\right\}; \qquad Q_\delta = \bigcup_{x \in Q} X_\delta(x);$$

$\lambda_i(A)$ – собственные числа матрицы вторых производных $A=J''(x)$, упорядоченные по убыванию: $\lambda_1(A) \ge \lambda_2(A) \ge \ldots \ge \lambda_n(A)$; $L(S)$ – минимальная константа Липшица в соотношении

$$\|J'(x') - J'(x)\| \le L(S)\|x' - x\|; \forall x', x \in S \subset R^n.$$

Основным в определении 2 является первое условие, констатирующее резко несимметричное расположение спектра матрицы $J''(x)$ относительно начала координат: $\lambda_i \in [-m, M]$, $M >> m > 0$. Второе и третье условия необходимы для описания свойства устойчивости множества Q: можно показать, что все ТСН, начинающиеся в любой точке $x \in Q_\delta$ быстро попадают в достаточно малую окрестность Q_ε ($\varepsilon << \delta$) множества Q и остаются там до выхода из множества Q_δ.

Как правило, оказывается достаточной более грубая модель явления овражности (жесткости), когда предполагается, что собственные числа матрицы вторых производных можно отчетливо разделить на две группы, в одну из которых входят собственные числа, по модулю намного превосходящие элементы второй группы. Будет использоваться следующее определение.

Пусть в $D \subset R^n$ задана r-мерная поверхность (конфигурационное пространство)

$$Q = \left\{x \in D \middle| g_i(x) = 0 (i = \overline{1, n-r})\right\}; \qquad g_i \in C^2(D).$$

Определение 3. Функционал $J(x)\in C^2(D)$, $D\subset R^n$ называется овражным (жестким) в множестве Q, если найдутся такие числа $\delta>0$, $\sigma >>1$, что

1) $\forall x \in Q_\delta$: $\lambda_1[J''(x)] \geq .... \geq \lambda_{n-r}[J''(x)] \geq \sigma|\lambda_{n-r+1}[J''(x)]| \geq ... \geq \sigma|\lambda_n[J''(x)]|$;

2) $\forall x \in Q$: $\text{Arg} \min\limits_{x'\in X_\delta} J(x') \subset Q$; (1.3.9)

3) $\forall x \in Q$: $L[X_\delta(x) \bigcap Q] \leq \sigma^{-1} L[X_\delta(x)]$.

Число r называется размерностью (дна) оврага Q.

Пример. Рассмотрим квадратичный функционал

$f(x)=1/2\langle Ax,x\rangle - \langle b,x\rangle + c$; $c=\text{const}$. (1.3.10)

Пусть собственные числа λ_i матрицы $A=f''(x)$ удовлетворяют неравенствам (1.3.8), а u_i означают соответствующие собственные векторы. Предположим, что $\det A\neq 0$ и обозначим через x* решение уравнения Ax=b. Примем

$$Q = \left\{ x \in R^n \middle| \langle x - x^*, u_i \rangle = 0 \left(i = \overline{1, n-r} \right) \right\}; \quad (1.3.11)$$

$Q_\delta = R^n$; $\sigma \cong \lambda_1 / |\lambda_{n-r+1}|$.

Можно доказать, что все три условия будут выполнены.

Таким образом, для квадратичных функционалов при сдвинутом в точку x* начале координат дно оврага Q совпадает с линейной оболочкой (1.3.11) собственных векторов, отвечающих малым собственным числам. Это согласуется с интуитивными представлениями, развитыми в разд.1.5.

На этом примере можно проиллюстрировать значение отдельных условий в определениях 2 и 3. Действительно, для квадратичного функционала (1.3.10) первое и второе условия могут выполняться для всего пространства $Q=R^n$ и любого $\delta >0$. Необходимая линейная оболочка

собственных векторов может быть выделена только при дополнительном требовании, эквивалентном третьему условию. В то же время требования 1, 3 также оказываются недостаточными. В этом случае сдвиг линейной оболочки Q, являющейся дном оврага, вдоль любого из не вошедших в оболочку собственных векторов не приведет к нарушению условий 1 и 3, а условие 2 при этом нарушится.

Рассмотренные выше модели явления овражности не являются исчерпывающими. Однако они описывают наиболее существенные стороны большинства практических ситуаций, связанных с задачами оптимизации реальных и проектируемых систем.

Определение 4. Пусть $\forall x \in Q$, $\det J''(x) \neq 0$. Наименьшее из чисел σ, удовлетворяющих определению 2, называется степенью овражности J(x) в Q и обозначается $\eta(Q)$. Отношение $\eta(x) = \lambda_1(x) / \left|\min_i \lambda_i(x)\right|$, $x \in Q$ называется локальной степенью овражности J(x) в точке x. Для вырожденных матриц J''(x) величина $\eta(x)$ принимается равной бесконечности.

Если $J''(x) > 0$, то $\eta(x) = \mathrm{cond}[J''(x)] = \max_i \lambda_i(x) / \min_i \lambda_i(x)$. В общем случае справедливо неравенство $1 \le \eta \le \mathrm{cond}\ (J'')$. При наличии больших по модулю отрицательных собственных чисел $\lambda_i(x)$ (т.е. при отсутствии овражной ситуации) возможно неравенство $\eta\ (x) << \mathrm{cond}\ [J''(x)]$. Из высокой степени овражности J(x) в точке x следует плохая обусловленность матрицы J''(x); обратное неверно. Действительно, пусть спектр матрицы J''(x) расположен в множестве $[-M, m] \cup [m, M]$, $M >> m > 0$, включая граничные значения. Тогда $\eta(x) = 1$, а $\mathrm{cond}[J''(x)] = M/m >> 1$. Данный функционал не будет относиться к классу овражных, что

естественно, ибо трудностей при его минимизации, например методом наискорейшего спуска, не возникает. Отличие между двумя характеристиками $\eta(x)$ и $\mathrm{cond}\,[J''(x)]$ функционала $J(x)$ часто игнорируется и овражными называют функционалы с большим числом $\mathrm{cond}\,(J'')$, что не оправдывается с позиций основных вычислительных трудностей, возникающих при решении задач оптимизации. Однако, учитывая указанную выше связь между $\eta(x)$ и $\mathrm{cond}\,(J'')$, овражные задачи, т.е. задачи минимизации овражных или жестких функционалов, далее будут называться плохо обусловленными задачами оптимизации.

В каждом конкретном случае различные значения $\eta(x)$ следует считать большими. Здесь существует полная аналогия с понятием плохой обусловленности матрицы. В большинстве случаев все определяется точностью вычислений и типом применяемого алгоритма оптимизации. Традиционно принято классифицировать задачу, как плохо обусловленную, если

$$\log_2\eta > t, \qquad (1.3.12)$$

где t – длина применяемой разрядной сетки компьютера. Однако и при меньших значениях η для многих алгоритмов могут возникать значительные вычислительные трудности, особенно если овражная структура сопровождается отсутствием выпуклости $J(x)$.

Дополнительным фактором, характеризующим степень сложности оптимизационной задачи и затрудняющим применение традиционных алгоритмов минимизации, является наличие многомерных оврагов с $r > 1$. В указанной ситуации целый ряд методов, специально ориентированных на решение плохо обусловленных задач, становятся неэффективными.

Рассмотрим практические методы распознавания овражной ситуации, играющие роль критериев овражности. Наиболее существенной характеристикой оказывается значение показателя η в допустимой области изменения управляемых параметров.

Своеобразным индикатором может служить метод простого градиентного спуска (ПГС), реализуемый по схеме

$$x^{k+1} = x^k - hJ'(x^k) \qquad (1.3.13)$$

с постоянным шагом $h \in R^1$.

Принадлежность J (x) к классу овражных в этом случае проявляется в необходимости применения относительно малых значений h. Попытки увеличения h вызывают потерю свойства релаксационности (монотонного убывания) последовательности $\{J(x^k)\}$ и значения $J(x^k)$ начинают резко возрастать. Если для некоторого фиксированного h (наибольшего из возможных) удалось заставить процесс (1.3.13) протекать без полной остановки, то можно количественно оценить величину η. Для этого процесс (1.3.13) продолжается до тех пор, пока отношение $\| J'(x^{k+1}) \| / \| J'(x^k) \|$ не установится около некоторого значения μ. Тогда справедливо равенство

$$\eta \cong 2 / | 1 - \mu |. \qquad (1.3.14)$$

Соотношение (1.3.14) является основным для грубой практической оценки степени овражности минимизируемого функционала в окрестности текущей точки. Доказательство соотношения (1.3.14) дано в разделе, посвященном градиентным схемам оптимизации.

В силу вышеизложенного можно рекомендовать процесс оптимизации начинать с помощью метода ПГС. Если задача простая и степень овражности функционала невелика, то уже этот стартовый метод достаточно быстро приведет в малую окрестность оптимума. В противном случае будет получено значение η, что позволит правильно оценить ситуацию и выбрать наиболее рациональный алгоритм.

Другой метод оценки η сводится к вычислению матрицы Гессе функционала и решению для нее полной проблемы собственных значений.

Тогда на основе непосредственной проверки выполнения неравенства (1.3.8) для вычисленных собственных чисел делается вывод о значении η. При этом определяется также размерность дна оврага r. Главный недостаток такого подхода заключается в существенных вычислительных трудностях принципиального характера, возникающих при определении малых собственных значений. Известно, что абсолютная погрешность $|d\lambda_i|$ представления любого собственного значения матрицы A за счет относительного искажения δ ее элементов удовлетворяет неравенству $|d\lambda_i| \le n\delta|\lambda_1|$, где $|\lambda_1| = \max_i |\lambda_i|$. Принимая $\delta = \varepsilon_м = 2^{-t}$, где $\varepsilon_м$ – относительная машинная точность (машинное эпсилон), а t – длина разрядной сетки мантиссы числа, получим оценку для абсолютных искажений собственных чисел из-за ошибок округления:

$$| d\lambda_i | \le n\varepsilon_м | \lambda_1 |. \qquad (1.3.15)$$

Параметр $\varepsilon_м$ известен для каждого компьютера. Из последнего неравенства можно сделать следующее заключение. Если все вычисленные собственные числа матрицы $A = J''(x)$ достаточно велики, т.е. $| \lambda_i | \ge n\varepsilon_м | \lambda_1 |$, то параметр η может быть вычислен непосредственно. Если же некоторые из вычисленных собственных чисел удовлетворяют неравенству $| \lambda_i | < n\varepsilon_м | \lambda_1 |$, то все они должны быть отнесены к блоку малых собственных чисел, а для η имеем границу снизу:

$$\eta \ge 1 / (n\, \varepsilon_м).$$

Качественным признаком плохой обусловленности оптимизационной задачи может служить существенное различие в результатах оптимизации, например методом ПГС при спуске из различных начальных точек. Получаемые результирующие точки обычно расположены достаточно далеко друг от друга и не могут рассматриваться как приближения к единственному решению или конечной совокупности

решений (при наличии локальных минимумов). Описанная ситуация, как правило, означает наличие оврага, а точки остановки применяемой поисковой процедуры трактуются как элементы дна оврага Q.

1.3.3. Основные причины возникновения. овражных целевых функционалов.

Несмотря на то, что типичность овражной ситуации может считаться установленным экспериментальным фактом, определенный интерес представляет выяснение основных причин появления оврагов в задачах моделирования и численной оптимизации.

Естественная причина появления овражной ситуации может быть связана с наличием некоторых неучтенных устойчивых связей между управляемыми параметрами, определяемых внутренними законами функционирования моделируемой системы. Если уравнения связей известны, то часть управляемых параметров может быть исключена из рассмотрения. В этом случае наличие овражной ситуации целесообразно трактовать как следствие некоторой избыточности в математическом описании объекта. С позиций приведенных в предыдущем разделе определений эти неизвестные соотношения между управляемыми параметрами можно трактовать как уравнения дна оврага.

Таким образом, поверхности уровня отдельных критериальных характеристик системы могут иметь овражный характер в силу естественных, в известном смысле не зависящих от исследователя причин.

Фактор агрегированности аргументов минимизируемого функционала. Выбор множества управляемых параметров (аргументов функционала) обычно производится по результатам анализа их влияния на основные характеристики оптимизируемой системы, а также исходя из

реальных возможностей изменения этих параметров в нужных пределах. Результатом такого вполне естественного подхода является ситуация, когда вектор выходных характеристик y оптимизируемой системы в действительности зависит от s агрегатов:

$$y = \Phi (z_1, z_2, \dots ,z_s), \qquad (1.3.16)$$

где $z_i = \varphi_i (x)$; $i=\overline{1,s}$; $s < n$.

Подтверждением сказанному служит широко применяемый на практике метод декомпозиции, связанный с выделением этапов аппроксимации и реализации в процессе оптимального выбора параметров x_i (см. разд. 1.2). Прямое решение задачи $J (x) \to \min$ в пространстве параметров x в данном случае связано с наличием овражной ситуации. Действительно, значение $J (x)$ мало меняется на множестве, определяемом равенствами $\varphi_i (x)=z_i^*$, $i=\overline{1,s}$. Поэтому последние уравнения фактически являются уравнениями дна оврага. Следовательно, фактор агрегированности естественным образом указывает на овражную ситуацию.

Во многих случаях соотношения (1.3.16) неизвестны, хотя агрегированные переменные z_i полностью определяющие выходные параметры объекта оптимизации, по-прежнему существуют. Поэтому применение изложенного в разд. 1.2 метода декомпозиции, в определенной степени исключающего проблему овражности, невозможно.

Отметим также возможную причину появления овражной ситуации в результате возникновения агрегатов при оптимизации динамических систем, описываемых жесткими системами обыкновенных дифференциальных уравнений.

Рассмотрим пример.

Пусть функционирование системы описывается вектор-функцией $u(t)=[u_1(t), u_2(t)]$, являющейся решением жесткой дифференциальной системы, где

$$\begin{cases} u_1(t)=[x_1 \exp(-At) + (x_1+x_2)\exp(-at)]t & (A>>a) \\ u_2(t)=[x_2 \exp(-Bt) + (x_1+x_2)\exp(-bt)]t & (B>>b) \end{cases} \qquad (1.3.17)$$

Требуется получить оптимальные значения параметров x_1, x_2 из условия наилучшего совпадения $u(t)$ с заданной вектор-функцией $\bar{u}(t)$ для $t\in[t_0, T]$. Если предположить, что $t_0 > \tau_{\text{п.с., где}}$ $\tau_{\text{п.с.}}$ – длина пограничного слоя, определяющего почти полное затухание экспонент $\exp(-At)$, $\exp(-Bt)$, то из соотношений (9.3.2) видно, что поведение решения $u(t)$ для $t\in[t_0, T]$ будет определяться агрегатом $z = x_1 + x_2$. Если продолжать считать x_1 и x_2 независимыми параметрами, то степень овражности, например, следующей оптимизационной задачи

$$J(x) = [u_1(t_1) - \bar{u}_1(t_1)]^2 + [u_2(t_1) - \bar{u}_2(t_1)]^2 \to \min_x,$$

где $t_1 = 1 \in [t_0, T]$; $\bar{u}_i = u_i(t_i)$, равна $\eta \cong 1{,}8 \cdot 10^8$ при $a = 1$; $b = 1{,}5$; $A = 20$; $B = 15$;

$\eta \cong \exp(-\min\{a; b\}/\exp(-\max\{A,B\})$.

Таким образом, до тех пор, пока не разработаны регулярные методы выделения агрегатов для последующей декомпозиции исходной задачи оптимизации, необходимо работать в пространстве переменных x_i в условиях отчетливо выраженной овражной ситуации.

Методы учета ограничений. Овражная ситуация может быть внесена в задачу оптимизации при учете ограничений с помощью построения обобщенного критерия оптимальности. Рассмотрим эффект овражности, возникающий при использовании методов штрафных функций и модифицированных функций Лагранжа. На почти обязательное наличие овражной ситуации в подобных случаях указывается во многих работах.

Рассмотрим в качестве примера ограничения в виде равенств $g_i(x) = 0$, $j=\overline{1,p}$. Тогда согласно методу штрафных функций задача сводится к минимизации вспомогательных функционалов

$$J_0(x,\sigma) = J(x) + \sigma\sum_{j=1}^{p} g_j^2(x) \to \min$$

с достаточно большим положительным коэффициентом σ. При этом структура расширенного критерия J_0, содержащего большой параметр σ, как правило, оказывается овражной, даже если исходный функционал $J(x)$ этим свойством не обладает.

Пример. Пусть требуется найти x_1 и x_2, минимизирующие квадратичный функционал $f(x) = x_1^2 + x_2^2$ при условии $x_1=2$. Задача имеет очевидное решение $x_1^* = 2$, $x_2^* = 0$. Поступим формально и составим вспомогательный функционал согласно общей рецептуре метода штрафных функций:

$$J_0(x) = x_1^2 + x_2^2 + \sigma (x_1 - 2)^2 = [(x_1 - b_1)^2/a_1^2] + [(x_2 - b_2)^2/a_2^2] + d,$$

где $a_1 = \sqrt{1+\sigma}$; $a_2 = 1$; $b_1 = 2\sigma / (\sigma+ 1)$; $b_2 = 0$; $d = 4\sigma / (\sigma+ 1)$. Уравнение линии уровня $J_0(x)=const$ является уравнением эллипса с центром в точке (b_1, b_2) и длинами полуосей, относящихся как $a_2 / a_1 = (1 + \sigma)^{1/2}$. При больших значениях σ, обеспечивающих относительно точное выполнение ограничения $x_1 = 2$, линии уровня оказываются сильно вытянутыми (рис. 1.3.2). Чем точнее выполняются ограничения, тем ярче выражен эффект овражности. В данном случае степень овражности равна $\eta = 1 + \sigma$. Заметим, что линии уровня исходного функционала являются сферами и явление овражности отсутствует.

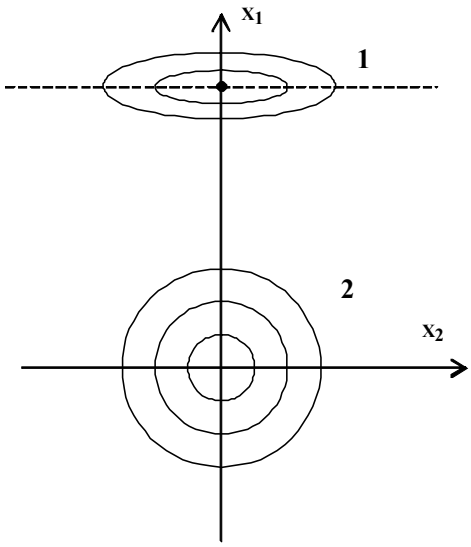


рис. 1.3.2

На практике метод штрафных функций широко используется на начальных этапах оптимизации, при этом применяются такие

возможно большие значения σ, для которых удается достигнуть относительно быстрого убывания J_0 (x) при достаточно точном выполнении ограничений. Для последующего уточнения решения привлекаются более тонкие стратегии, которые, как правило, оказываются и существенно более трудоемкими. Кроме того, метод штрафных функций до сих пор не имеет разумных альтернатив в некоторых критических ситуациях, характерных для реальных задач оптимизации.

Например, при наличии вырожденного минимума оптимизационной задачи, когда нарушается условие линейной независимости градиентов $g_j'(x)$, могут потерять работоспособность все методы учета ограничений, основанные на обычной и модифицированной функциях Лагранжа, а также на линеаризации ограничений. Метод же штрафных функций в указанной ситуации применим. Он оказывается наименее чувствительным ко всем формам вырождения.

Второй критической ситуацией, возникающей в практике оптимизации, является неоправданное завышение функциональных требований к объекту оптимизации, которое приводит к пустому множеству D допустимых значений управляемых параметров. В подобном случае наиболее целесообразно применять метод штрафных функций, позволяющий получить такое решение задачи $\|g(x)\|^2 \to \min$, для которого значение J(x) минимально. Другие методы либо теряют смысл, либо заведомо не будут сходящимися.

В силу вышеизложенного наличие алгоритмов оптимизации, сохраняющих работоспособность при достаточно высокой степени овражности минимизируемых функционалов, оказывается чрезвычайно желательным. Этот вывод подтверждается также тем фактором, что и при использовании модифицированных функций Лагранжа сталкиваемся с овражной ситуацией, хотя и в ослабленной форме. Например, как

известно, повышая скорость сходимости итераций (1.2.15) за счет выбора достаточно больших коэффициентов σ, мы снова приходим к плохо обусловленной задаче минимизации функционалов (1.2.14).

Объединение конфликтных выходных параметров. Учет противоречивых требований к многокритериальному объекту оптимизации с помощью единого критерия оптимальности является важнейшим фактором, обусловливающим возникновение овражной ситуации. Используемые методы построения Парето-оптимальных решений приводят к двум основным видам сверток: линейной и минимаксной (максиминной). Остановимся в качестве примера на минимаксной свертке двух частных критериев:

$$J(x) = \max \{a_1 J_1(x), a_2 J_2(x)\}; \quad a_i>0; \quad a_1 + a_2 = 1.$$

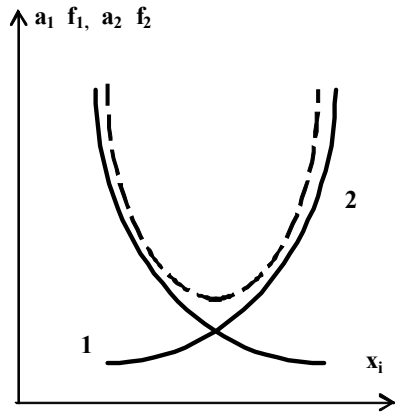


рис. 1.3.3

При этом отдельные критериальные выходные параметры как функции от параметров оптимизации могут иметь монотонный, существенно неовражный характер. Однако их объединение почти неизбежно приводит к овражной ситуации. При этом крутые склоны оврага характеризуют доминирующее влияние на обобщенный критерий какого-то одного из частных критериев. Как следует из рис. 1.3.3, объединение критериев J_1 и J_2 приводит к образованию сложной «клювообразной» зависимости, порождающей в многомерном случае овраг с крутыми склонами. Характерно, что движение по любой из поверхностей a_iJ_i в отдельности с помощью любых методов оптимизации обычно не вызывает затруднений.

Широко применяемый на практике метод наименьших квадратов связан с минимизацией функционалов вида

$$J(x) = \sum_{i=1}^{n} \alpha_i \varphi_i^2(x) \to \min. \qquad (1.3.18)$$

Конструкция J (x) в принципе может рассматриваться как линейная свертка многокритериальной задачи $\varphi_i^2(x) \to \min$, $i = \overline{1,n}$. На типичность овражной ситуации при решении задач (1.3.18) указано во многих как теоретических, так и экспериментальных исследованиях. Появление оврагов выражается, в частности, в известном факте плохой обусловленности соответствующей системы нормальных уравнений.

1.3.4. Некоторые стандартные методы. конечномерной оптимизации.

Рассмотрим возможности некоторых традиционных методов с позиций их практической применимости.

Методы сопряженных градиентов (СГ). Методы СГ позволяют строить достаточно эффективные вычислительные процедуры и поэтому занимают важное место в арсенале средств конечномерной оптимизации. Главный недостаток методов СГ, существенно ограничивающий область их рационального использования, заключается в понижении скорости сходимости для плохо обусловленных задач оптимизации. Оценка скорости сходимости этих методов показывает, что стандартные схемы СГ сходятся по закону геометрической прогресии со знаменателем t, близким к единице: $t \cong 1 - 2/\sqrt{\eta}$, где η – степень

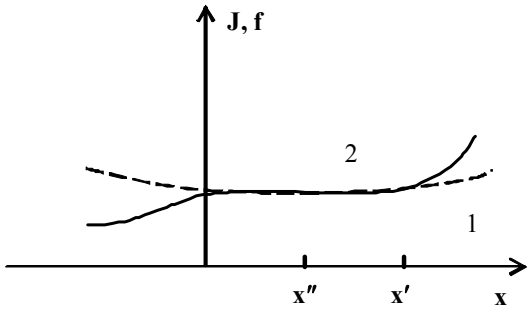


рис. 1.3.4

овражности минимизируемого функционала. В некоторых работах имеются указания на достаточно высокую скорость сходимости метода СГ по функционалу независимо от значения η:

$$J(x^k) - J(x^*) = L\,\|x^0 - x^*\|^2 / [2(2k+1)^2] \qquad (L = \text{const}). \qquad (1.3.19)$$

Однако оценки типа (1.3.19) получены в предположении строгой положительной определенности матрицы $J''(x^*)$. Кроме этого, согласно (9.4.1) достаточно эффективно получаются значения функционала порядка $J(x^*)$, где x^* – точка минимума аппроксимирующего квадратичного функционала $f(x)$, что в общем случае не решает задачу (рис. 1.3.4).

Отрезок [x', x"] при высоких значениях η методом СГ будет преодолеваться с малым шагом по аргументу, хотя $f(x') \cong f(x'') \cong f(x^*)$, где $x^* = \operatorname{argmin} f(x)$. Предположение о невыпуклости $J(x)$ вносит дополнительные трудности. Как известно, в этих условиях метод СГ эквивалентен классическому методу наискорейшего спуска со всеми вытекающими отсюда последствиями.

Эти выводы подтверждаются опытом практической работы. Кроме того, следует учесть, что основные преимущества методов СГ перед методами второго порядка ньютоновского типа в значительной степени теряются, если используются специальные экономичные методы вычисления вторых производных, обсуждаемые ниже.

Ньютоновские методы. Классическая формула метода Ньютона имеет вид

$$x^{k+1} = x^k - h_k[J''(x^k)]^{-1} J'(x^k) \quad (h_k \in R^1). \qquad (1.3.20)$$

Предполагается, что все матрицы $J''(x^k)$ положительно определены, что гарантирует разрешимость задачи вычисления x^{k+1}. Известны различные модификации метода $x^{k+1} = x^k \; h_k[J(x^k)]^1 J(x^k) \quad (h_k \; R^1)$.

(1.3.20), построенные с целью его обобщения − на ситуации, когда матрица $J''(x^k)$ оказывается вырожденной или не обладает свойством положительной определенности. При этом вместо матрицы $J''(x^k)$ в схеме метода начинает фигурировать некоторая другая, положительно определенная, матрица G_k. Один из таких методов рассмотрен в разд. 1.5.2 под названием метода. Левенберга. Общий недостаток этих алгоритмов заключается в том, что изменения в схему (1.3.20) вносятся с единственной целью сделать осмысленными все вычислительные операции независимо от определенности матрицы $J''(x^k)$. В то же время полезная информация о рельефе минимизируемого функционала, содержащаяся в матрице $J''(x^k)$, почти не используется.

Важный подкласс ньютоновских методов составляют методы Гаусса-Ньютона (ГН), аппроксимирующие метод Ньютона и его модификации на основе использования информации о структуре минимизируемого функционала. В классических вариантах методов ГН предполагается, что $J(x)$ имеет вид суммы квадратов. В результате вычисление матрицы $J''(x)$ с достаточной точностью сводится к вычислению только первых производных от составляющих $J(x)$ функций. Соответствующие вопросы рассмотрены в разд.3.2 ТПР. Обсуждавшиеся выше недостатки ньютоновских методов сохраняются в методах ГН.

В разд. 1.4, 1.5 излагаются методы, также использующие аппроксимации матрицы $J''(x)$ на основе первых производных. В отличие от методов ГН их вычислительные схемы ориентированы на общую, невыпуклую ситуацию, что оказывается более рациональным при решении реальных задач оптимизации.

Квазиньютоновские методы (КН). КН-методы имеют структуру

$$x^{k+1} = x^k - h_k H_k J'(x^k) \qquad (h_k \in R^1), \tag{1.3.21}$$

где H_k – (n × n)-мерная матрица, пересчитываемая на каждом шаге с помощью одной из известных рекуррентных формул; h_k – длина шага, выбираемая, например, из условия минимума J(x) в направлении $p^k = -H_k J'(x^k)$. Существуют и другие стратегии назначения величины h_k.

Обычно КН-методы (1.3.21) обладают свойством $H_n = [J''(x)]^{-1}$ при минимизации сильно выпуклых квадратичных функционалов. Начальная матрица H_0 выбирается симметричной и положительно определенной. Тогда этими же свойствами будут обладать последующие матрицы H_k. Поэтому направления p^k будут указывать в сторону убывания J(x), независимо от выпуклости J(x).

Таким образом, в КН-методах аппроксимация матрицы, обратной матрице Гессе, осуществляется с помощью первых производных. Это определяет высокую эффективность КН-методов при решении широкого класса задач.

Доказано, что большинство вариантов КН-методов при минимизации сильно выпуклых квадратичных функционалов приводит к одной и той же траектории поиска, вырождаясь в методы СГ. Поэтому для них характерно аналогичное замедление сходимости в овражной ситуации. Кроме этого, КН-методы построены и исследованы в расчете на выпуклые задачи; нарушение свойства выпуклости и особенно неудачное масштабирование могут приводить к вырождению матриц, а также к необходимости работы с бесконечно большими шагами, что существенно снижает эффективность этих методов.

Для решения «больших» задач оптимизации традиционные процедуры КН-методов в отличие от методов СГ не применяются, так как их вычислительные схемы предполагают хранение и пересчет

заполненных (n×n)-мерных матриц H_k, что требует больших затрат памяти[1].

Несмотря на отмеченные недостатки, КН-методы считаются достаточно эффективными и широко применяются в задачах практической оптимизации.

«Метод оврагов» Гельфанда – Цетлина. Недостатки этого эвристического метода, ориентированного на решение плохо обусловленных задач, достаточно полно изложены во многих работах. Основной недостаток заключается в полной непригодности метода для ситуации многомерного оврага. По существу это обстоятельство ограничивает возможное число аргументов минимизируемого функционала на уровне n = 2. Метод должен применяться для специальных классов задач, структура которых определяет наличие только одномерных оврагов.

Аналогичными недостатками обладает известный метод вращения осей Розенброка, кратко рассмотренный в разд. 1.4.

Методы поиска глобального оптимума. Все рассмотренные выше методы, за исключением «метода оврагов», являются локальными, так как с их помощью может быть найден один из локальных минимумов функционала J(x). Представляет практический интерес поиск глобального минимума, т.е. такого локального минимума, где значение критерия оптимальности оказывается наименьшим.

Трудность вопроса заключается в том, что для произвольного функционала J(x) задача глобальной оптимизации неразрешима с

[1] В последнее время появились некоторые новые варианты КН-методов, ориентированные на задачи большой размерности.

помощью вычислений J(x) в любом сколь угодно большом, но конечном числе точек. Поэтому алгоритмы глобальной оптимизации должны развиваться для достаточно узких классов задач на основе имеющейся априорной информации.

В настоящее время известен один довольно общий класс критериев оптимальности, для которых обеспечивается возможность локализации глобального минимума за обозримое машинное время. Речь идет о классе функционалов, удовлетворяющих условиям Липшица:

$$|J(x') - J(x'')| \le L \, \| x' - x'' \| \qquad (L = \text{const}, \; L \ge 0).$$

Существующие для таких функционалов методы глобальной оптимизации достаточно подробно изложены в литературе. Недостатком этих методов является требование знания константы Липшица для всей области изменения x. Неправильное назначение L может резко уменьшить скорость сходимости метода либо привести к потере глобального минимума.

Реальная ситуация в области глобальной оптимизации в настоящее время расценивается как неблагоприятная. Существующие методы поиска глобального экстремума, особенно в овражной ситуации, не могут рассматриваться как исчерпывающие при решении задач достаточно высокой размерности.

Наиболее распространенный и достаточно эффективный эвристический метод заключается в задании некоторой грубой сетки начальных точек в допустимом множестве с последующим применением методов локальной оптимизации. Для построения таких сеток целесообразно применять широко известные в практике математического моделирования $ЛП_\tau$-последовательности, обладающие свойством равномерного заполнения многомерной области. При этом в качестве начальных точек для локальных процедур спуска могут использоваться только некоторые точки сетки, которым отвечают меньшие значения функционала.

Таким образом, в настоящее время основным инструментом практической оптимизации продолжают оставаться локальные методы.

В заключение отметим, что в некоторых случаях проблема многоэкстремальности возникает в результате определенного непонимания реальной ситуации. Регистрируемые на практике многочисленные локальные экстремумы в действительности оказываются точками остановки применяемых поисковых процедур (см. разд. 1.4). Можно утверждать, что наличие многих локальных минимумов в практических задачах встречается значительно реже, чем об этом принято говорить (за исключением многочисленных специально сконструированных тестовых многоэкстремальных задач).

1.4.ПОКООРДИНАТНЫЕ СТРАТЕГИИ КОНЕЧНОМЕРНОЙ ОПТИМИЗАЦИИ.

1.4.1. Методы покоординатного спуска.

Пусть задан функционал $J(x) \in C^1(R^n)$. Решается задача построения минимизирующей последовательности $\{x^k\}$ для $J(x)$.

Часто используемый метод решения поставленной задачи состоит в применении покоординатной стратегии исследования пространства поиска.

Переход от вектора x^i к вектору x^{i+1} с помощью метода покоординатного спуска (ПС) происходит следующим образом: для $l = \overline{1,n}$ компонента x_l^{i+1} определяется из условия минимума:

$$J(x_1^{i+1}, x_2^{i+1}, \ldots, x_{l-1}^{i+1}, x_l^{i+1}, x_{l+1}^{i}, \ldots, x_n^{i}) = \min_{x \in R^1} J(x_1^{i+1}, x_2^{i+1}, \ldots, x_{l-1}^{i+1}, x, x_{l+1}^{i}, \ldots, x_n^{i}). \quad (1.4.1)$$

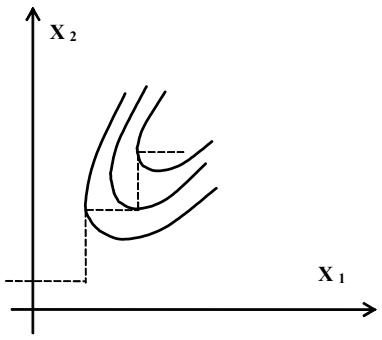


рис. 1.4.1

В овражной ситуации этот метод применим лишь в редких случаях ориентации оврагов вдоль координатных осей. Трудности применения подобных процедур проиллюстрированы на рис. 1.4.1 Продвижение к точке минимума становится замедленным при наличии сильно вытянутых поверхностей уровня. Обычно имеет место ситуация «заклинивания», вызываемая дискретным характером представления информации в вычислительной машине. Рассмотрим этот вопрос подробнее.

Числа, представимые в компьютере, расположены на вещественной оси дискретно (рис. 1.4.1). Аналогичным образом плоскость (x_1, x_2) в памяти компьютера аппроксимируется также конечным множеством точек, лежащих на пересечении соответствующих прямых (рис. 1.4.2).

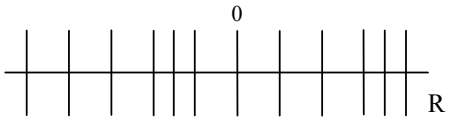


рис. 1.4.1

В результате любая точка А плоскости R^2 может оказаться точкой «заклинивания» метода ПС, если в ее окрестности линии уровня J(x) имеют овражную структуру (рис. 1.4.4).

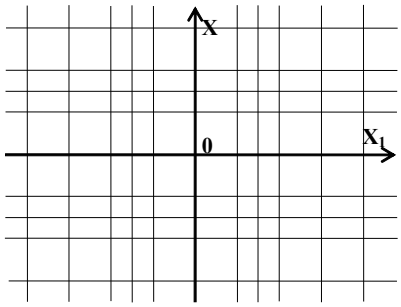


рис. 1.4.2

Из рисунка видно, что если процесс попадает в точку А или в любую другую точку, расположенную на дне оврага, то ближайшие доступные точки В, В′ и С, С′ будут соответствовать большему значению функционала и убывания J не будет ни в одном из координатных направлений. Подобная ситуация может возникнуть и на очень больших расстояниях от точки минимума. Вероятность заклинивания возрастает при использовании негладких функционалов типа максиминных критериев минимального запаса при решении систем неравенств, обсуждавшихся в разд.1.2.

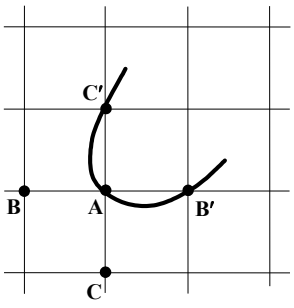


рис. 1.4.4

Типичный случай представлен на рис. 1.4.5. Линии уровня имеют изломы, поэтому метод теряет работоспособность уже в относительно простых задачах. Такого типа ситуации, по существу, не являются следствием высокой степени овражности и могут быть устранены

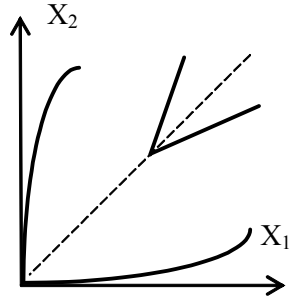


рис. 1.4.5

переходом к гладким аппроксимациям, построенным по методике, изложенной в разд.1.2.

Несмотря на отмеченную низкую эффективность метода ПС в овражной ситуации, его включение в библиотеку алгоритмов целесообразно при решении любого класса оптимизационных задач, по крайней мере как стартового алгоритма, с целью получения разумного начального приближения для последующих процедур. Причина этого заключается в высокой надежности метода по отношению к различным сбойным ситуациям, а также в простоте процесса подготовки задачи к решению на компьютере. Метод имеет нулевой порядок, т.е. не требует включения в вычислительную схему информации о производных минимизируемого функционала. При реализации метода могут быть использованы стандартные способы одномерного поиска минимума типа золотого сечения, квадратичной аппроксимации и другие, известные из численного анализа. Однако это приводит к заметному и часто неоправданному усложнению алгоритма. Легко можно привести примеры, когда точный поиск минимума вдоль координатных направлений не только не обязателен, но даже вреден. Поэтому часто применяются более простые стратегии выбора шагов. Рассмотрим в качестве примера один простой вариант, оказывающийся вполне приемлемым для практических вычислений.

Задается вектор начальных шагов $h = (h_1,\ldots,h_n)$ продвижений из точки x в направлении ортов $e_1, e_2, \ldots, e_n$. Далее шаги h_i модифицируются от итерации к итерации. Если выполняется неравенство $J(x+h_ie_i)\le J(x)$, то текущая точка x заменяется на $x+h_ie_i$, а величина h_i утраивается : $h_i:=3h_i$. После этого осуществляется переход к следующему номеру i. Если

$J(x+h_ie_i)>J(x)$, то производится умножение h_i на $-0,5$ и также осуществляется переход к следующему координатному орту.

Таким образом, алгоритм адаптируется к конкретным условиям оптимизации в результате изменения значений и знаков шагов. Если начальные значения шагов были выбраны неудачно, то они быстро скорректируются до необходимых значений.

Указанный метод выбора координатных шагов реализован в алгоритме GZ1. Это, по-видимому, простейшая из возможных реализаций метода покоординатного спуска.

Алгоритм GZ1.

Шаг 1. Ввести начальную точку $x = (x_1, \ldots, x_n)$ и шаг s, принять $F := J(x)$.

Шаг 2. Принять $h_i := s, \ i = \overline{1,n}$.

Шаг 3. Принять $m := 1$.

Шаг 4. Принять $x_m := x_m + h_m$; вычислить $F_1 = J(x)$.

Шаг 5. Если $F_1 \le F$, принять $h_m := 3h_m, \ F := F_1$ и перейти к шагу 7; иначе – перейти к шагу 6.

Шаг 6. Принять $x_m := x_m - h_m, \ h_m := -0,5h_m$.

Шаг 7. Принять $m := m+1$. Если $m \le n$, перейти к шагу 4; иначе – к шагу 3.

Выход из алгоритма осуществляется после достижения заданного числа N вычислений J(x). Обычно программа составляется таким образом, чтобы обеспечить возможность продолжения работы с прерванного места после повторных входов в GZ1. Таким образом, в этом случае мы отказываемся от применения каких-либо внутренних критериев сходимости процесса оптимизации и обрываем его после заранее

обусловленного числа шагов. При необходимости такой анализ сходимости может выполняться во внешней программе путем сравнения результатов, полученных при двух последовательных обращениях к GZ1.

При каждом новом входе в алгоритм счетчик числа вычислений функционала зануляется, следовательно, разрешается еще N обращений к подпрограмме вычисления J(x). Задавая различные значения N, можно регулировать частоту выходов из GZ1 во внешнюю программу для оценки получаемых результатов, например для снятия выходных характеристик, соответствующих текущим значениям компонент вектора настраиваемых параметров. Кроме этих вспомогательных действий во внешней программе должна быть организована требуемая процедура вывода окончательных результатов.

Из-за рассмотренного выше явления «заклинивания» Х. Розенброк в 1960 г. был вынужден модифицировать процедуру ПС. Подробное описание полученного алгоритма содержится во многих работах. Основная идея заключается в организации процесса покоординатного спуска не вдоль фиксированных координатных ортов, а вдоль осей специальным образом выбираемой системы координат. При этом одна из осей должна составлять достаточно малый угол с образующей дна одномерного оврага. В результате смещения по этой оси совпадают с продвижением вдоль дна оврага в направлении точки минимума. Схема метода Розенброка сводится к трем основным этапам. Пусть x^{m-1} и x^m – две соседние точки в минимизирующей последовательности $\{x^k\}$, построенной рассматриваемым методом. Тогда переход к x^{m+1} осуществляется следующим образом:

1) Выбрать новую систему координат, первая ось которой направлена вдоль вектора $x^m - x^{m-1}$, а остальные дополняют ее до ортонормированного базиса («поворот осей»).

2) В новой системе координат для поиска x^{m+1} осуществить алгоритм GZ1 до выполнения условий поворота осей.

3) Возвратиться к старой системе координат и перейти к шагу 1.

Точка x^0 задается, а x^1 получается из x^0 с помощью алгоритма GZ1 в естественной системе координат.

Различные модификации метода отличаются друг от друга способом организации одномерного поиска вдоль координатных ортов, способом построения ортогонального дополнения к оси $x^m - x^{m-1}$ (методом ортогонализации), а также выбором условия окончания процесса спуска для перехода к очередному повороту осей.

К недостаткам всех вариантов метода Розенброка следует отнести невозможность продолжения процесса оптимизации, если в качестве начальной точки выбрана «точка заклинивания» метода покоординатного спуска. Другим более существенным недостатком является то обстоятельство, что метод применим лишь к задачам оптимизации с одномерными оврагами. При наличии многомерных оврагов метод теряет эффективность, так как в нем не принимается специальных мер для погружения необходимого числа координатных осей (равного размерности оврага) в пространство, образующее дно оврага. В результате целенаправленное изменение ориентации лишь одной из координатных осей не позволяет эффективно продвигаться по многомерному дну.

1.4.2. Методы обобщенного покоординатного спуска.

Пусть решается задача $J(x) \to \min$, $x \in R^n$, $J \in C^2(R^n)$ с овражным (по определению 3, см. 1.3.2) функционалом. Таким образом, существует некоторое подмножество $Q \subset R^n$, и при $\forall x \in Q$ для собственных чисел матрицы $J''(x)$ справедливы неравенства

$$\lambda_1 \geq \ldots \geq \lambda_{n-r} >> |\lambda_{n-r+1}| \geq \ldots \geq |\lambda_n| . \qquad (1.4.2)$$

Число r малых собственных значений определяет размерность оврага (дна оврага). Основной процедурой при реализации рассматриваемого далее класса методов обобщенного покоординатного спуска (ОПС) является процедура приведения матрицы $J''(x)$ к главным осям, т.е. процедура диагонализации, с последующим покоординатным спуском вдоль собственных векторов матрицы. Целесообразность такого подхода вытекает из того, что оси наиболее рациональной системы координат при минимизации квадратичных функционалов (независимо от их выпуклости) методом покоординатного спуска совпадают с собственными векторами матрицы вторых производных. Эта идея неоднократно высказывалась в литературе и даже строились соответствующие алгоритмы. Однако опубликованные результаты численных экспериментов показали низкую эффективность такого подхода. Может быть предложено и объяснение этих результатов. Оно основано на том, что при определении собственных векторов, соответствующих близким или кратным собственным значениям, возникают принципиальные вычислительные трудности. Аналогичные трудности, связанные с ограниченной точностью задания исходной информации, а также последующих вычислений, наблюдаются и при диагонализации плохо обусловленных матриц, имеющих относительно малые по модулю спектральные составляющие. Указанные обстоятельства, по-видимому, явились основным сдерживающим фактором, не позволившим внедрить изучаемые ниже методы в вычислительную практику. Однако из дальнейшего изложения следует, что неудачи при численном экспериментировании были вызваны особенностями реализации метода, рассчитанной на минимизацию выпуклых функционалов. Как показано ниже, при минимизации как выпуклых, так и невыпуклых функционалов достаточно вычислить произвольный

ортонормированный базис в инвариантном подпространстве, отвечающем каждой изолированной группе собственных значений. При этом отдельные собственные векторы могут быть вычислены со значительными погрешностями. Можно показать, что отвечающие этим базисам линейные оболочки с высокой точностью совпадают с истинными подпространствами, определяемыми невозмущенной диагонализируемой матрицей.

Эти выводы в известной степени подтверждаются следующей теоремой.

Теорема 9. Пусть A – (n × n)-мерная симметричная матрица; $[u_i\}$ – ортонормированные собственные векторы; $\{\lambda_i\}$ – собственные значения. Тогда при $\lambda_i \neq \lambda_j$ с точностью до величин второго порядка малости имеем

$$\langle u_i + du_i, u_j \rangle = (\lambda_i - \lambda_j)^{-1} \langle (dA)\, u_i, u_j \rangle, \qquad (1.4.3)$$

где dA – возмущение матрицы A; du_i – соответствующее возмущение вектора u_i;

Доказательство. Отбрасывая величины второго порядка малости, из равенства $Au_i = = \lambda_i u_i$ получим $(dA)\, u_i + Adu_i = \lambda_i du_i + d\lambda_i u_i$. Отсюда $\langle (dA)u_i, u_j \rangle + \langle Adu_i, u_j \rangle = \lambda_i \langle du_i, u_j \rangle + + d\lambda_i \langle u_i, u_j \rangle$. Из равенства $\langle u_i, u_j \rangle = 0$; $\langle Adu_i, u_j \rangle = \lambda_j (du_i, u_j \rangle$ получим $(\lambda_i - \lambda_j) \langle du_i, u_j \rangle = = \langle (dA)\, u_i, u_j \rangle$, откуда следует равенство (10.2.2). Теорема доказана.

Пусть теперь $M_1 = \sum_{i=1}^{n-r} a_i u_i$; $M_2 = \sum_{j=n-r+1}^{n} a_j u_j$ есть два линейных многообразия, порожденных непересекающимися системами собственных векторов

$$\left\{u_i, i = \overline{1, n-r}\right\}; \qquad \left\{u_j, j = \overline{n-r+1, n}\right\}$$

матрицы A. Если соответствующие множества собственных значений

$$\left\{\lambda_i, i = \overline{1, n-r}\right\}; \qquad \left\{\lambda_j, j = \overline{n-r+1, n}\right\}$$

строго разделены: $|\lambda_i| >> |\lambda_j|$, то из равенства (1.4.3) следует $\langle u_i + du_i, u_j \rangle \approx 0$ при достаточно малом значении $||dA|| / |\lambda_i|$. Это означает, что все собственные векторы под действием возмущения dA изменяются только в пределах своих линейных многообразий, сохраняя с высокой точностью свойство ортогональности к векторам из дополнительных многообразий. При этом сами вариации векторов при близких $\lambda_i \approx \lambda_j$ собственных значениях в пределах фиксированного линейного многообразия, как это следует из равенства (1.4.3), могут быть весьма значительными.

Вышеизложенное позволяет в качестве модели программ, реализующих различные методы диагонализации матрицы A, использовать оператор $\Lambda(A)$, ставящий в соответствие произвольной симметричной матрице A ортогональную матрицу V, отличную, вообще говоря, от истинной матрицы U, состоящей из собственных векторов матрицы A. Оператор Λ характеризуется тем, что если спектр матрицы A разделяется на p групп

$$\lambda_i^{\sigma}(A); \quad \sum_{\sigma=1}^{p} k_\sigma = n \qquad (i = \overline{1, k_\sigma})$$

близких между собой собственных чисел, то каждой группе σ соответствует набор столбцов $\{v_i^{\sigma}\}$ матрицы V, задающий точное линейное многообразие, порожденное соответствующими столбцами $\{u_i^{\sigma}\}$ точной матрицы U.

Рассмотрим квадратичную аппроксимацию

$$f(x) = 1/2 \langle Ax, x\rangle - \langle b, x\rangle + c \qquad (1.4.4)$$

исходного функционала J(x) в окрестности точки $x \in Q$. Допустим, что известны матрица A и ортогональная матрица U, приводящая ее к

диагональному виду $U^TAU = \mathrm{diag}\,(\lambda_i)$. Тогда замена переменных $x = Uy$ приводит квадратичный функционал к сепарабельному виду

$$f(x) = f(Uy) = \sum_{i=1}^{n} f_i(y_i), \qquad (1.4.5)$$

где f_i – квадратичные функции одной переменной. Таким образом, локально достигается полная декомпозиция исходной задачи и последняя сводится к n независимым оптимизационным задачам. В результате поиск оптимального вектора y^* может осуществляться покомпонентно, ибо связь между аргументами y_i фактически исчезает. В указанной ситуации явление заклинивания невозможно, и все вычислительные трудности при применении покоординатных стратегий поиска оптимума, связанные с большими значениями η, полностью устраняются.

В действительности бывает задана не матрица A, а возмущенная матрица A + dA, где dA отражает как неопределенность задания исходной матрицы A, так и последующие ошибки округления при проведении собственно процесса диагонализации. В связи с этим вместо точной матрицы U оказывается доступной некоторая матрица $V = \Lambda(A)$. Свойства оператора Λ были рассмотрены выше. Замена переменных $x = Vy$ уже не приводит функционал к виду (10.2.4). Для изучения создающейся ситуации важное значение имеет следующая теорема.

Теорема 10. Пусть собственные значения $\{\lambda_i\}$ и отвечающие им ортонормированные собственные векторы $\{u_i\}$, $i = \overline{1,n}$, некоторой симметричной матрицы A разделены произвольным образом на p групп λ_i^σ; u_i^σ; $\sum_{\sigma=1}^{p} k_\sigma = n$ $(i = \overline{1,k_\sigma}$, $\sigma = \overline{1,p})$ так, что $u_i^\sigma \neq u_j^s$; $\sigma \neq s$ $(i = \overline{1,k_\sigma}$, $j = \overline{1,k_s})$, где λ_j^σ, u_j^σ – j-е собственное число и соответствующий

собственный вектор группы σ. Тогда, если в каждом линейном многообразии M_σ размерности k_σ с базисом $\{u_i^\sigma\}$, $i=\overline{1,k_\sigma}$ задать иной ортонормированный базис $\{w_i^\sigma\}$, $i=\overline{1,k_\sigma}$, связанный с исходным базисом линейным соотношением $\omega_i^\sigma=\sum_{m=1}^{k_\sigma}\alpha_{mi}^\sigma u_m^\sigma$ $(i=\overline{1,k_\sigma},\alpha_{mi}^\sigma\in R^1)$, то существует такая матрица P перестановок столбцов, что:

1) преобразование подобия $\overline{W}^T\overline{AW}$, $W=[\omega_1^1,\ldots,\omega_{k_1}^1,\ldots,\omega_{k_p}^p]$, $\overline{W}=WP$ приводит матрицу A к блочно-диагональному виду $\overline{W}^T\overline{AW}=\mathrm{diag}\,(A_1,A_2,\ldots,A_p)$; $\overline{W}^T=\overline{W}^{-1}$ с квадратными $(k_\sigma\times k_\sigma)$-мерными матрицами A_σ на главной диагонали;
2) собственные значения матрицы A_σ есть λ_i^σ, $i=\overline{1,k_\sigma},\sigma=\overline{1,p}$.

Доказательство. Первое утверждение проверяется непосредственно с учетом ортонормированности векторов базиса $\{u_i\}$. Для доказательства второго утверждения достаточно заметить, что вид и расположение матрицы A_m при фиксированном многообразии M_m не зависят от способа задания остальных многообразий M_σ, $\sigma\neq m$. Поэтому, предположив, что все $k_\sigma=1$ при $\sigma\neq m$, получим

$$\overline{W}^T\overline{AW}=\mathrm{diag}\,(\lambda_1^1,\lambda_1^2,\ldots,A_m,\ldots,\lambda_1^p).$$

Учитывая, что преобразование подобия не изменяет спектр матрицы, приходим к требуемому заключению. Теорема доказана.

Теорема 11. Пусть $V=\Lambda(A)$, тогда:

1) замена переменных $x=Vy$ с точностью до нумерации компонент вектора y приводит функционал f(x) вида (10.2.3) к блочно-сепарабельному виду

$$f_s(y)=f(Vy)=\sum_{\sigma=1}^{p}f_\sigma(y^\sigma), \qquad (1.4.6)$$

где $y = (y_1, \dots, y_n) = (y^1, \dots, y^p)$; $y^\sigma = (y_1^\sigma, \dots, y_{k_\sigma}^\sigma)$;

$f_\sigma(y^\sigma) = 1/2\langle A_\sigma y^\sigma, y^\sigma\rangle - \langle b^\sigma, y^\sigma\rangle + c_\sigma,\ c_\sigma \in R^1$,

2) собственные значения матрицы f_σ'' равны $\lambda_i^\sigma(A)$, $i = \overline{1, k_\sigma}$, $\sigma = \overline{1, p}$.

Доказательство. Имеем $V = \overline{W}P$, где P – некоторая матрица перестановок столбцов. Поэтому $f(x) = 1/2\ \langle V^T A V y, y\rangle - \langle V^T b, y\rangle + c = 1/2\ \langle P^T \overline{W}^T A \overline{W} P y, y\rangle - \langle P^T \overline{W}^T b, y\rangle + c = 1/2(\langle \overline{W}^T A \overline{W} z, z\rangle - \langle \overline{W}^T b, z\rangle + c$, $z \underset{=}{\Delta} Py$. Согласно теореме 10 матрица $\overline{W}^T A \overline{W}$ имеет блочно-диагональную структуру, что и доказывает первое утверждение. Второе утверждение есть прямое следствие второго утверждения теоремы 10.

Следствие. Пусть собственные числа матрицы A удовлетворяют неравенствам (10.2.1). Тогда:

1) замена переменных $x = Vy$, $V = \Lambda(A)$, где $V = (v_1^1, \dots, v_{n-r}^1, v_1^2, \dots, v_r^2)^T$; $v_i^1 = \sum_{m=1}^{n-r} \alpha_{mi}^1 u_m$, $v_i^2 = \sum_{m=1}^{r} \alpha_{mi}^2 u_{n-r+m}$, с точностью до нумерации компонент вектора y приводит f(x) к виду

$$f_s(y) = f_1(y^1) + f_2(y^2); \quad y = (y^1, y^2), \qquad (1.4.7)$$

где $y^1 = (y_1, \dots, y_{n-r})$; $y^2 = (y_{n-r+1}, \dots, y_n)$;

2) $\eta_1 << \eta$, $\eta_2 << \eta$, где η_i – показатели овражности функционалов f_i.

Таким образом, исходная оптимизационная задача локально может быть сведена к двум эквивалентным задачам с существенно меньшими числами η_i. Представление (1.4.7) является аналогом идеализированного соотношения (1.4.5).

Если собственные числа матрицы квадратичного функционала разделяются более чем на две группы, то будет справедливо представление

(1.4.7), содержащее соответствующее число слагаемых.

Согласно соотношению (1.4.7) появляется возможность независимого решения не связанных между собой оптимизационных задач для функционалов f_i с невысокими показателями овражности.

Полученные результаты носят локальный характер и справедливы в рамках квадратичной аппроксимации исходного функционала J(x). Для неквадратичных функционалов приближенное выполнение соотношений типа (1.4.7) позволяет говорить о существенном ослаблении связей между различными группами переменных, что определяет достаточно высокую эффективность покоординатного спуска и в общем случае.

Исследование сходимости представленных выше алгоритмов в предположении точной линейной оптимизации вдоль направляющих ортов может быть основано на следующем общем подходе к исследованию алгоритмов нелинейного программирования.

Пусть решается задача $J(x) \to \min$, $x \in R^n$, $J(x) \in C^1(R^n)$. Рассмотрим произвольный алгоритм A, строящий последовательность точек $\{x^k\}$, причем каждая точка x^{k+1} получается последовательной минимизацией функционала J(x) вдоль направлений $d_1, \dots, d_n$, начиная из точки x^k. Предполагается, что матрица $D = (d_1, \dots, d_n)$ может зависеть от номера k, являясь при любом k ортогональной. Легко видеть, что метод ПС, метод Розенброка, а также методы ОПС описываются приведенной общей схемой.

Теорема 12. Пусть: 1) $J(x) \in C^1(R^n)$; 2) множество решений, определяемое как $X_* = \{x \in R^n \mid J'(x) = 0\}$, непусто; 3) множество $\{ x \in R^n \mid J(x) \le J(x^0)\}$, где x^0 – заданная начальная точка, ограничено и замкнуто в R^n ; 4) минимум функционала J(x) вдоль любой прямой в R^n единственен; 5) если $J'(x^k) = 0$, то алгоритм останавливается в x^k.

Тогда каждая предельная точка последовательности $\{x^k\}$, построенной алгоритмом A, принадлежит множеству X_*.

Доказательство мы здесь не приводим.

1.4.3. Реализация методов обобщенного покоординатного спуска.

Методы вычисления производных. Для функционалов, заданных аналитически, можно воспользоваться аналитическим методом вычисления производных. Однако в действительности такой подход имеет ограниченное применение, так как реальные функционалы имеют достаточно сложную алгоритмическую структуру, не позволяющую воспользоваться аналитическим дифференцированием. Поэтому наиболее часто на практике применяются полуаналитические и численные методы.

Полуаналитические методы занимают промежуточное положение между аналитическими и численными методами построения производных. Эти методы основаны на использовании специальной структуры минимизируемых функционалов и в этом смысле оказываются менее универсальными, чем численные методы.

Рассмотрим характерные (см. разд. 10) для практических задач конечномерной оптимизации функционалы специального вида:

$$J_1(x) = \nu^{-1} \sum_{k=1}^{m} \varphi_k^{\nu}(x) \quad (\nu = 2, 3, \ldots); \tag{1.4.8}$$

$$J_2(x) = \nu^{-1} \sum_{k=1}^{m} \exp[-z_k(x)\nu] \quad (\nu = 1, 2, \ldots), \tag{1.4.9}$$

где φ_k, z_k − алгоритмически заданные функции вектора управляемых параметров.

Имеем следующие выражения для составляющих вектора градиента:

$$\partial J_1(x) / \partial x_i = \sum_{k=1}^{m} \varphi_k^{\nu-1}(x) \partial \varphi_k(x) / \partial x_i; \qquad (1.4.10)$$

$$\partial J_2(x) / \partial x_i = -\sum \exp(-z_k \nu) \partial z_k(x) / \partial x_i \qquad (i = \overline{1,n}). \qquad (1.4.11)$$

Естественный способ получения вторых производных состоит в линеаризации функций φ_k, z_k вблизи текущей точки x':

$$\varphi_k(x) \approx \varphi_k(x') + \langle \partial \varphi_k(x')/\partial x,\ x - x' \rangle;$$

$$z_k(x) \approx z_k(x') + \langle \partial z_k(x')/\partial x,\ x - x' \rangle. \qquad (1.4.12)$$

Используя (1.4.12), получаем

$$\partial^2 J/\partial x_i \partial x_j \approx (\nu - 1) \sum_{k=1}^{m} \varphi_k^{\nu-2}(x)\ (\partial \varphi_k/\partial x_i)\ (\partial \varphi_k/\partial x_j) \qquad (\nu = 2, 3, \ldots); \quad (1.4.13)$$

$$\partial^2 J/\partial x_i \partial x_j \approx \nu \sum_{k=1}^{m} \exp\ [-z_k(x)\nu]\ (\partial z_k/\partial x_i)\ (\partial z_k/\partial x_j) \qquad (\nu = 1, 2, \ldots). \quad (1.4.14)$$

Из выражений (1.4.13), (1.4.14) следует, что вычисление вторых производных минимизируемого функционала может быть сведено к вычислению первых производных функций φ_k, z_k. Последняя задача для многих практических ситуаций решается относительно просто. В частности, возможен численный подход для вычисления $\partial \varphi_k/\partial x_i$, $\partial z_k/\partial x_i$, что значительно проще прямого численного вычисления гессовой матрицы (отсюда название – полуаналитический метод).

Дальнейшая детализация структуры решаемой задачи обычно позволяет построить эффективный алгоритм получения указанных производных первого порядка. Например, хорошо известны методы построения производных решений дифференциальных уравнений по параметрам.

Еще одним примером использования специфики задачи оптимизации для вычисления первых производных может служить известный в теории электрических цепей метод, основанный на теореме Теллегена и

позволяющий находить частные производные реакций схемы по параметрам компонентов. При этом анализ чувствительности может проводиться как во временной, так и в частотной областях.

Обратимся теперь к численным методам вычисления производных. Достоинством численного подхода кроме его универсальности является низкая стоимость подготовки задачи к компьютерному моделированию. От пользователя требуется лишь написание программы для вычисления значения J(x) при заданном x. Реализованные на основе численных производных методы оптимизации оказываются по существу прямыми методами (так называются методы, не использующие в своей схеме производные функционала J(x)). Действительно, заменяя $\partial J/\partial x_i$, например, конечно-разностным отношением $[J(x+ se_i) - J(x)]/s$, где $e_i = (0, \dots , 1, \dots , 0)$, мы фактически используем лишь значения J, вычисленные при определенных значениях аргумента.

Все рассмотренные в этой книге методы оптимизации строятся на основе использования локальной квадратичной модели минимизируемого функционала, получаемой из общего разложения в ряд Тейлора. Естественно поэтому при выборе формул численного дифференцирования также руководствоваться идеей локальной квадратичной аппроксимации. Исходя из этого целесообразно вместо формул с односторонними приращениями (подобно только что рассмотренным) применять двусторонние конечно-разностные аппроксимации производных, оказывающиеся точными для квадратичных функционалов. Действительно, легко проверить, что если $f(x) = =1/ 2 \langle Ax,x\rangle - \langle b, x\rangle$, то равенство

$$\partial f/\partial x_i = [f(x + se_i) - f(x - se_i)]/(2s) \qquad (1.4.15)$$

оказывается точным при любом $s \neq 0$.

Точно так же точным оказывается следующее представление для

вторых производных квадратичного функционала:

$$\partial^2 f/\partial x_i \partial x_j = [f(x + se_i + se_j) - f(x - se_i + se_j) -$$
$$- f(x + se_i - se_j) + f(x - se_i - se_j)] / (4s^2). \qquad (1.4.16)$$

При использовании соотношений (1.4.15), (1.4.16) вычислительные затраты характеризуются числом обращений к вычислению значений J(x): для вычисления градиента – 2n, для вычисления матрицы Гессе – $(2n^2 + 1)$ обращений, где n – размерность вектора x.

Для излагаемых далее методов оптимизации достаточно определять J′(x) и J′′(x) с точностью до множителя, поэтому при реализации формул (1.4.15), (1.4.16) деление соответственно на 2s и $4s^2$ не производится. Это позволяет устранить известные трудности вычислений, если значение s оказывается относительно малым. Если необходимо работать с различными шагами s_i по отдельным компонентам x_i вектора x, то можно надлежащим образом ввести масштабы независимых переменных, оставляя без изменения стандартную программу вычисления производных.

Методы диагонализации. В качестве основной процедуры приведения симметричной матрицы к главным осям может быть выбран широко известный метод Якоби, несмотря на наличие конкурирующих и, вообще говоря, более эффективных вычислительных схем. Этот выбор обусловлен следующими обстоятельствами. Во-первых, вычисленные методом Якоби собственные векторы всегда строго ортонормальны с точностью, определяемой точностью компьютера даже при кратных собственных числах. Последнее весьма существенно при использовании этих векторов в качестве базиса, так как предотвращается возможность вырождения базиса, существующая, например, в методе Пауэлла. Во-вторых, многие вычислительные схемы имеют преимущество перед методом Якоби лишь при решении частичной проблемы собственных значений. В нашем же случае всегда решается полная проблема и поэтому

выигрыш во времени оказывается несущественным при существенно более сложных вычислительных схемах. В-третьих, алгоритмы, основанные на методах Якоби, часто оказываются наиболее доступными, так как соответствующие программы имеются в большинстве вычислительных лабораторий. И наконец, определенное влияние на выбор алгоритма оказала простота логики метода Якоби, что приводит к компактности реализующих его программ.

В задачах большой размерности по сравнению с методом Якоби более предпочтительным по объему вычислительных затрат оказывается метод, использующий преобразование Хаусхолдера для приведения матрицы к трехдиагональной форме с последующим обращением к QR-алгоритму определения собственных векторов симметричной трехдиагональной матрицы .

В методе Якоби исходная симметричная матрица A приводится к диагональному виду с помощью цепочки ортогональных преобразований вида

$$A_{k+1} = U^T_k A_k U_k;\ A_0 = A\ (k = 1, 2, \ldots), \qquad (1.4.17)$$

являющихся преобразованиями вращения. В результате надлежащего выбора последовательности $\{U_k\}$ получаем $\lim A_k = D = U^T A U$, $k \to \infty$, где $D = \mathrm{diag}(\lambda_i)$ – диагональная матрица; $U = U_0 U_1 U_2 \ldots$ – ортогональная матрица. Так как (10.3.10) есть преобразование подобия, то на диагонали матрицы D расположены собственные числа матрицы A; столбцы матрицы U есть собственные векторы матрицы A.

Элементарный шаг (1.4.17) процесса Якоби заключается в преобразовании посредством матрицы $U_k = \{u_{i\ j}\}$, отличающейся от единичной элементами $u_{pp} = u_{qq} = \cos\varphi$, $u_{pq} = -u_{q\,p} = \sin\varphi$. Угол вращения φ выбирается таким образом, чтобы сделать элемент a_{pq} матрицы A нулем. Вопросы сходимости различных численных схем, реализующих метод

Якоби, подробно исследованы в литературе.

За основу может быть взят алгоритм jacobi (из известной коллекции алгоритмов Уилкинсона и Райнша, записанных на языке Алгол), реализующий так называемый частный циклический метод Якоби. В этом методе аннулируются все элементы верхней треугольной части матрицы A с применением построчного выбора. При таком выборе индексы элементов a_{pq} пробегают последовательность значений (1, 2), (1, 3), … , (1, n); (2, 3), (2, 4), … ... , (2, n); … ; (n − 1, n). Затем начинается новый цикл перебора элементов в том же порядке.

Эмпирическая оценка трудоемкости процесса построения матрицы $\Lambda(A)$ методом jacobi позволяет выразить необходимое время работы процессора T через размерность n решаемой задачи. Известно, что для матриц до 50-го порядка и длин машинных слов от 32 до 48 двоичных разрядов общее число циклов в процессе вращений Якоби в среднем не превышает 6 −10 (под циклом понимается любая последовательность из $(n^2 - n)/2$ вращений). При этом $T = kn^3$, где коэффициент k определяется быстродействием компьютера и приблизительно равен 40 t_y, где t_y – время выполнения операции умножения.

Полученная оценка, а также опыт практической работы, показывают, что при умеренных значениях n время реализации оператора Λ для многих практических случаев невелико и сравнимо с временем однократного вычисления значения минимизируемого функционала. Упоминающаяся выше комбинация метода Хаусхолдера и QR алгоритма оказывается приблизительно в полтора-два раза быстрее, что может иметь значение только при достаточно больших n.

1.4.4. Алгоритмы обобщенного покоординатного спуска.

Процедура вычисления производных может быть организована пользователем, например, в соответствии с рекомендациями, приведенными в разд. 3.2 ТПР. Однако в качестве основной процедуры для вычисления матрицы Гессе далее выбран метод конечно-разностных соотношений с переменным шагом дискретности s. Последний может определяться автоматически, например, в зависимости от продвижения в пространстве переменных x, задаваемого нормой $|| x^i - x^{i-1} ||$. Чем большее значение s используется, тем шире предполагаемая область справедливости локальной квадратичной модели исходного функционала. Наибольшая точность вычислений по формуле (1.4.16) применительно к квадратичным зависимостям достигается при работе с максимально возможным s, так как в этом случае вклад погрешностей задания значений J в окончательный результат становится наименьшим. Поэтому чем дальше удалось продвинуться на основе построенной квадратичной аппроксимации функционала, тем, по-видимому, большие значения s целесообразно выбирать для вычисления производных на следующем этапе поиска. Возможны и другие стратегии регулировки параметра s. Например, сама подпрограмма, реализующая метод оптимизации, может быть настроена на работу с постоянным шагом дискретности. Изменения s в этом случае осуществляются во внешней программе в зависимости от получаемых результатов.

Собственные векторы матрицы не зависят от скалярного множителя, поэтому, как уже указывалось, деление на $4s^2$ в формуле (1.4.16) не производится.

Полученные в результате диагонализации матрицы Гессе новые координатные орты используются далее для реализации базового

алгоритма покоординатного спуска с процедурой выбора шагов продвижения по осям, применяемой в алгоритме GZ1. Переход к новым осям координат целесообразно осуществлять после того, как текущие оси «исчерпали себя» и дальнейшего существенного улучшения ситуации не ожидается. В предлагаемых алгоритмах обновление осей координат происходит после того, как по каждому из координатных направлений вслед за успешным продвижением последовала неудача – возрастание значения J(x). Разумеется, могут существовать и другие критерии для определения момента изменения координатных ортов.

Укрупненное описание алгоритма, реализующего метод ОПС, сводится к следующей последовательности шагов.

Алгоритм SPAC1.

Шаг 1. Ввести данные: x, s.

Шаг 2. Вычислить матрицу $B = \{b_{ij}\}$ по формулам $b_{ij} = J(x + se_i + se_j) - J(x - se_i + se_j) - J(x + se_i - se_j) + J(x - se_i - se_j)$, $i, j = \overline{1, n}$; $e_i=(0,\ldots, 1, \ldots, 0)$.

Шаг 3. С помощью процедуры jacobi построить ортогональную матрицу U, приводящую матрицу B к диагональному виду U^TBT.

Шаг 4. В осях $\{u_i\}$, совпадающих со столбцами матрицы U, реализовать процесс покоординатного спуска из точки x до выполнения условия поворота осей; присвоить x полученное лучшее значение, модифицировать s и перейти к шагу 2.

Процесс заканчивается по исчерпании заданного числа обращений к процедуре вычисления J(x). Так же, как и в алгоритме GZ1, должна быть предусмотрена возможность повторных входов в алгоритм и продолжения вычислений с прерванного места. Приводимые в разд. 20 результаты работы SPAC1 соответствуют автоматическому выбору шагов

дискретности для численного дифференцирования, исходя из равенства $s_{i+1} = 0{,}1 \| x^{i+1} - x^i \|$.

Построенный алгоритм имеет простую структуру, однако его эффективность может быть достаточно высокой, несмотря на необходимость построения матрицы B. Соответствующие примеры приведены в разд. 20. В некоторых случаях более эффективной оказалась модификация метода ОПС, реализованная в алгоритме SPAC2.

В алгоритме SPAC2 матрица вторых производных вычисляется в текущих осях $\{u_i\}$, без возврата к единичному исходному базису $\{e_i\}$. В результате информация о последнем используемом базисе не теряется, что позволяет иногда сократить трудоемкость решения задачи.

Алгоритм SPAC2.

Шаг 1. Ввести исходные данные: x, s.

Шаг 2. Принять U = E, где E – единичная матрица; в качестве координатных векторов взять столбцы $\{u_i\}$ матрицы U.

Шаг 3. Построить матрицу $B = \{b_{ij}\}$ по формулам $b_{ij} = J(x + su_i + su_j) - J(x - su_i + su_j) - J(x + su_i - su_j) + J(x - su_i - su_j)$, $(i, j = \overline{1, n;})$.

Шаг 4. Принять U := UT, где T – ортогональная матрица, приводящая матрицу B к диагональному виду $T^T BT$.

Шаг 5. В осях $\{u_i\}$ реализовать процесс покоординатного спуска из точки x до выполнения условия поворота осей; присвоить x лучшее полученное значение. Модифицировать s и перейти к шагу 3.

Окончание процесса и выбор шагов дискретности такие же, как и в алгоритме SPAC1.

Дадим необходимые пояснения к алгоритму, касающиеся построения матрицы U на шаге 4.

Выбор в качестве координатных направлений столбцов $\{u_i\}$ некоторой ортогональной матрицы u очевидно эквивалентен замене переменных x = Uy. В этом случае изменения компонент y_i вектора y приводят в исходном пространстве к смещениям вдоль одноименных векторов u_i.

Функционал J(Uy) = I(y) как функция y имеет матрицу Гессе вида $I''(y)=U^TJ''U$. Действительно, $I'(y) = U^TJ'(x)$; $I''(y) = \partial[U^TJ'(Uy)]/\partial y = U^TJ''(x)U$.

Таким образом, если необходимо работать с функционалом, имеющим матрицу Гессе вида $U^TJ''U$, то для этого достаточно в качестве базиса взять столбцы матрицы U. Если требуется изменить матрицу Гессе и привести ее к виду $T^T(U^TJ''U)T = U^T{}_1JU_1$, где T – новая ортогональная матрица; $U_1 = UT$, то в качестве базисных векторов достаточно выбрать столбцы матрицы $U_1 = UT$. Указанная процедура и реализована на шаге 4 сформулированного алгоритма.

Остановимся на некоторых принципиальных отличиях алгоритма SPAC2 от SPAC1. Во-первых, если минимизируемый функционал близок к квадратичному и матрица Гессе меняется относительно мало, то повторное ее построение в осях $\{u_i\}$ опять приведет к диагональной матрице и поэтому число якобиевых циклов вращений при последующей диагонализации вновь полученной матрицы J′′ будет сведено к минимуму. В результате матричная поправка T к матрице U, вычисляемая на шаге 4, оказывается близкой к единичной матрице. В алгоритме SPAC1 в указанных условиях весь процесс диагонализации должен каждый раз целиком повторяться; то, что оси $\{u_i\}$ фактически меняются мало при переходе к следующей точке x^i, никак не используется.

Во-вторых, если на каком-то этапе поиска минимума шаг

дискретности s оказывается меньше, чем, скажем, $\min_i |\varepsilon_M x_i|$, где ε_M – машинное эпсилон, то это приведет к получению нулевой матрицы на этапе построения матрицы B. Подпрограмма, реализующая алгоритм jacobi, выполнена таким образом, что при этом в качестве диагонализирующей матрицы T получим единичную матрицу. Если указанная ситуация возникает в процессе работы SPAC2, то на шаге 4 не произойдет изменения матрицы U и процесс покоординатного спуска будет продолжен в текущих осях координат. Таким образом, будет сохраняться возможность медленного продвижения, скорость которого затем может вновь возрасти. При использовании алгоритма SPAC1 нулевая матрица B автоматически приведет к получению матрицы U = E, что эквивалентно возврату к исходному единичному базису.

В результате вероятность заклинивания на участках медленного продвижения для SPAC1 оказывается существенно большей, чем для SPAC2.

Отмеченные особенности SPAC2 не позволяют, однако, полностью отказаться от применения SPAC1. Это объясняется, во-первых, тем, что трудоемкость вычисления матрицы B в SPAC2 оказывается заметно выше, чем в SPAC1, так как вектор x варьируется в направлениях u_i, отличных, вообще говоря, от единичных векторов e_i. Кроме этого, важное отличие заключается в том, что алгоритм SPAC2 предполагает почти единственно возможный способ построения матриц Гессе, основанный на формулах (1.4.16). При применении же SPAC1 пригоден любой из описанных в разд. 1.4.3 методов. Последнее обстоятельство часто оказывается решающим при выборе метода.

1.4.5. Реализация методов обобщенного покоординатного спуска на основе рекуррентных алгоритмов оценивания.

В процессе работы алгоритма ОПС получается последовательность векторов $\{x^k\}$ и отвечающая им последовательность значений минимизируемого функционала $\{J_k\}$. В указанных алгоритмах используются только те точки x^k, которые приводят к монотонному убыванию J(x), а «неудачные» точки отбрасываются и далее никак не участвуют в процессе поиска. Ниже показано, что соответствующая информация может быть эффективно использована для построения квадратичной модели функционала J(x) с целью последующего вычисления собственных векторов матрицы J′′(x) в качестве новых направлений поиска.

Рассмотрим последовательности $\{x^k\}$, $\{J_k\}$, получаемые методом ОПС в текущей системе координат. В этом случае имеются в виду «полные» последовательности, включающие в себя как удачные, так и неудачные шаги. Опишем процедуру, позволяющую по этой информации вычислить матрицу Гессе аппроксимирующего квадратичного функционала. Задачу квадратичной аппроксимации будем решать на основе метода наименьших квадратов:

$$F_N(c) = \sum_{i=1}^{N} [J(x^i) - f(x^i, c)]^2 \to \min_{c \in R^m}; \qquad (1.4.18)$$

$x^i = (x_1^i, x_2^i, \ldots, x_n^i)$; $c = (c_1, c_2, \ldots, c_m)$;

$m = n^2/2 + 3n/2 + 1$; $N \geq m$.

Здесь f(x, c) означает аппроксимирующий функционал с неизвестными коэффициентами c_i:

$$f(x, c) = c_1 x_1^2 + c_2 x_1 x_2 + c_3 x_1 x_3 + \ldots + c_n x_1 x_n +$$

$$+ c_{n+1} x_2^2 + c_{n+2} x_2 x_3 + \ldots + c_{2n-1} x_2 x_n +$$

. . . . . .

$$+ c_{n^2/2+n} x_n^2 + c_{n^2/2+n+1} x_1 + \ ... \ + c_{m-1} x_n + c_m. \qquad (1.4.19)$$

Полагая $y(x) = (x_1^2, x_1x_2, ..., x_1x_n, x_2^2, ..., x_n^2, x_1, ..., x_n, 1)$, представим (1.4.19) в виде $f(x, c) = \langle c, y\rangle = y^T c$, что позволяет говорить о линейной регрессионной задаче оценки параметров $c_1,...,c_m$. Система нормальных уравнений, отвечающая МНК-функционалу (1.4.18), имеет вид

$$Y_N Y^T{}_N c = \left(\sum_{k=1}^{N} y_k y_k^T\right) c = Y_N J^N = \sum_{k=1}^{N} J_k y_k, \qquad (1.4.20)$$

где $Y_N = (y_1, y_2, ..., y_N)$ – $(m \times N)$-мерная матрица; $y_k = y(x^k)$; $J^N = (J_1, J_2, ..., J_N)^T$; $J_i = J(x_i)$. Заметим, что так как квадратичный функционал $F_N(c) \geq 0$ и $F''_N(c) = YY^T$, то матрица YY^T неотрицательно определена. Можно рассчитать коэффициенты квадратичной модели непосредственно из системы линейных алгебраических уравнений (10.5.3). Однако более рациональным оказывается другой подход, позволяющий избежать решения линейных систем.

Известно, что для псевдообратной матрицы $(\cdot)^+$ выполняется соотношение

$$(Y^T)^+ = \lim (\delta E + YY^T)^{-1} Y \qquad (\delta \to 0, \ \delta > 0).$$

Отсюда следует, что вместо системы уравнений

$$Y_r Y_r^T c = Y_r J^r \qquad (1.4.21)$$

можно рассматривать систему

$$(\delta E + Y_r Y_r^T)\, c = Y_r J^r, \qquad (1.4.22)$$

решение которой при $\delta \to 0$ сходится к решению (1.4.21) с минимальной нормой среди всех векторов, минимизирующих величину

$$\| Y_r^T c - J^r \|^2 \to \min_c,$$

что при $r = N$ совпадает с выражением (1.4.18). Ниже на основе известных рекуррентных алгоритмов оценивания, соответствующих случаю $\delta = 0$,

будут построены методы решения регуляризованных систем (1.4.22) при конечных (малых) значениях параметра δ. В этом случае δ играет роль параметра регуляризации, обеспечивая устойчивость получаемых решений к ошибкам округления. При δ=0 решение системы (1.4.21) может наталкиваться на существенные вычислительные трудности, так как при N < m матрицы $Y_N Y_N^T$ будут вырождены, а при N>m – плохо обусловлены.

Введем обозначение

$$P_r^{-1} = \sum y_k y_k^T + \delta E = P_{r-1}^{-1} + y_r y_r^T \qquad (1.4.23)$$

Используя так называемую вторую лемму об обращении матриц (эквивалентную формуле Шермана-Моррисона-Вудбери)

$$(K^{-1} + B^T R^{-1} B)^{-1} = K - KB^T (BKB^T + R)^{-1} BK,$$

получаем рекуррентное соотношение

$$P_r = (P_{r-1}^{-1} + y_r y_r^T)^{-1} = P_{r-1} - P_{r-1} y_r (y_r^T P_{r-1} y_r + 1)^{-1} y_r^T P_{r-1}, \ (r = \overline{1.N}) \qquad (1.4.24)$$

Так как, очевидно, $P_1^{-1} = y_1 y_1^T + \delta E$, то из (1.4.23) следует, что необходимо принять $P_0^{-1} = \delta E$ или $P_0 = \delta^{-1} E$.

Матрица P_0^{-1} симметрична и положительно определена. Предположим, что P_{r-1}^{-1} симметрична и положительно определена, и покажем, что P_r^{-1} обладает этими же свойствами. Последнее согласно теории симметричных возмущений немедленно следует из представления (1.4.23), так как матрица $y_r y_r^T$ симметрична и неотрицательно определена. Поэтому по принципу индукции все матрицы P_r^{-1}, а вместе с ними и P_r будут симметричны и положительно определены. Таким образом, все обратные матрицы в соотношении (1.4.24) существуют.

Построим рекуррентное соотношение для определения оценок c_i^r. Уравнение (1.4.22) имеет вид

$$P_r^{-1} c^r = \sum_{k=1}^{r} J_k y_k, \qquad (1.4.25)$$

откуда

$$P_r^{-1} c^r = \sum_{k=1}^{r} J_k y_k + y_r J_r = P_{r-1}^{-1} c^{r-1} + y_r J_r . \qquad (1.4.26)$$

Здесь c^r означает оценку вектора c по r вычислениям функционала J(x). Прибавляя и вычитая $y_r y_r^T c^{r-1}$ в правой части (1.4.26), получаем

$$P_r^{-1} c^r = P_r^{-1} c^{r-1} + y_r (J_r - y_r^T c^{r-1}). \qquad (1.4.27)$$

Из (10.5.10) имеем окончательное выражение

$$c^r = c^{r-1} + P_r y_r (J_r - y_r^T c^{r-1}). \qquad (1.4.28)$$

По формуле (1.4.28) может быть вычислена новая оценка c^r вектора параметров при условии, что известна предыдущая оценка c^{r-1}, матрица P_r и вновь полученные значения J_r, $y_r = y(x^r)$.

Последовательный метод оценки параметров квадратичной модели функционала J позволяет заменить процедуру обращения матрицы полной нормальной системы уравнений (1.4.22) операцией вычисления скаляра, обратного заданному $y_r^T P_{r-1} y_r + 1$, выполняемой на каждом шаге итерационного процесса (1.4.24).

Непосредственно из построения уравнений видно, что результат c^r для r=N, полученный согласно выражениям (1.4.24), (1.4.28), приводит к оценке, которая получается из решения полной системы (1.4.22). При этом нужно принять $c^0 = 0$. Последнее следует из соотношений (1.4.25), (1.4.28), записанных для r = 1.

Действительно, согласно (1.4.25) имеем оценку $c^1 = P_1 J_1 y_1$, полученную в результате решения системы (10.5.5). Из выражения (1.4.28) следует $c^1 = c^0 + P_1 y_1 (J_1 - y_1^T c^0)$. Поэтому для совпадения обеих оценок c^1, а значит и последующих оценок, необходимо и достаточно принять $c^0 = 0$.

На основе вычисленных оценок c_i^N, $i = \overline{1,(n^2+n)/2}$, задающих аппроксимацию матрицы J′′(x), может быть реализована процедура ОПС. При этом возможны различные стратегии применения изложенного

общего подхода, конкретизирующие способ выбора числа «измерений» y_r, J_r, участвующих в коррекции текущей оценки, а также самих точек y_r. Целесообразно после каждого поворота осей обновлять процесс и вновь начинать процедуру построения аппроксимации. Такая тактика позволяет не учитывать «устаревшие» значения J, расположенные достаточно далеко от текущей точки.

Изложенная процедура обладает определенными свойствами адаптируемости по локализации окрестности текущей точки, в которой строится аппроксимирующая квадратичная модель. Действительно, если норма результирующего вектора продвижения в текущих осях достаточно велика, то исходный функционал заменяется квадратичным в достаточно широкой области пространства поиска. Если же оси выбраны неудачно и продвижение мало, то автоматически на формирование квадратичной модели оказывают влияние только близко лежащие точки и тем самым область предполагаемой «квадратичности» функционала J(x) сжимается.

Опыт применения такого типа алгоритмов для целей оптимизации в настоящее время недостаточен. Однако можно ожидать, что в некоторых случаях будут возникать трудности, связанные с рациональным выбором δ, определяющим, в частности, погрешности промежуточных вычислений и их влияние на результат. В этом смысле подбор δ необходимо начинать с относительно больших значений, позволяющих с достаточной точностью получать «малые разности больших величин» при реализации соотношений (1.4.24). Кроме этого необходимо учитывать, что реализация методов ОПС на основе рекуррентного оценивания коэффициентов квадратичной модели приводит к увеличению емкости необходимой памяти компьютера.

1.4.6. Результаты численных экспериментов.

Основным подходом к сравнению алгоритмов с точки зрения их последующих приложений является тестирование, то есть проверка алгоритмов при решении определенного класса тестовых задач оптимизации, построенных таким образом, чтобы внести в процесс оптимизации те характерные трудности, которые возникают при решении реальных задач.

Однако выбор класса тестовых функционалов, подлежащих минимизации, еще не решает проблемы. Остается актуальным вопрос об оценке вычислительных затрат алгоритма при решении конкретной тестовой задачи. Наиболее естественен подход, связанный с анализом реального машинного времени, затраченного на поиск минимума функционала с заданной точностью. Однако на практике целесообразно применять метод, основанный на оценке трудоемкости в специальных единицах – Горнерах. В один Горнер (1 Г) оценивается трудоемкость операции однократного вычисления значения минимизируемого функционала. Эффективность алгоритма оптимизации, затратившего на получение результата с заданной точностью наименьшее число Горнеров, считается наивысшей.

Использование метода оценки трудоемкости в Горнерах базируется на экспериментально оправданном предположении, что наиболее трудоемкой при решении реальных задач является процедура вычисления J(x). В задачах оптимизации реальных систем эта процедура связана с решением соответствующей задачи анализа, что требует существенных вычислительных затрат.

При решении тестовых задач, имеющих простую алгоритмическую структуру, основное время расходуется на реализацию собственно

стратегии оптимизации, а не на вычисление значений J(x). В силу этой оценки сравнительной эффективности на основе анализа времени работы машины имеют ограниченную ценность.

Далее приводятся результаты численных экспериментов по проверке рассмотренных алгоритмов оптимизации. Рассмотрены семь характерных примеров, относительно которых накоплен достаточно богатый опыт. Кроме этого, выбирались такие задачи, которые оказались неразрешимыми хотя бы одним из обычно применяемых методов оптимизации. В результате работа алгоритмов проверялась на задачах минимизации следующих тестовых функций.

1 Квадратичная функция простой структуры

$F_1(x) = (x_1 - x_2)^2 + (x_1 + x_2 - 10)^2/9$; $x^0 = (0; 1)$; $x^* = (5; 5)$; $F_1(x^*) = 0$.

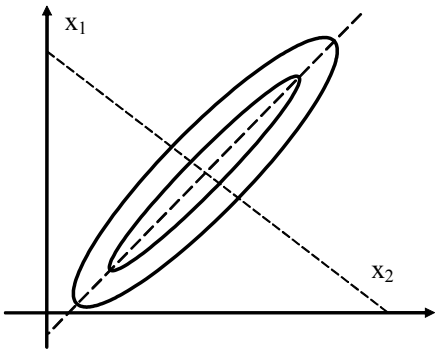


рис. 1.4.6

Приведенная функция моделирует ситуацию, когда никаких вычислительных трудностей не возникает и оказываются пригодными почти все методы. Результаты минимизации F_1 могут быть использованы при отладке соответствующих программ. Линии уровня F_1 изображены на рис. 1.4.6

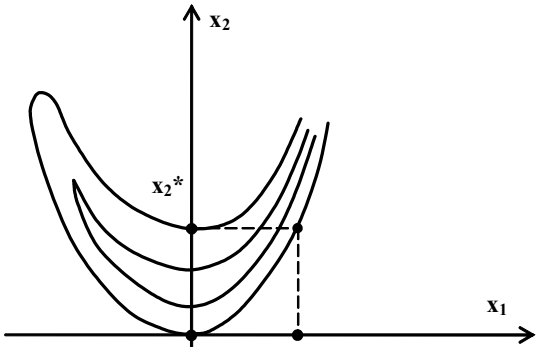


рис. 1.4.7

2. Функция Розенброка

$F_2 = 100\,(x_1^2 - x_2)^2 + (1 - x_1)^2$;

$x^0 = (-1, 2; 1)$; $x^* = (1; 1)$; $F_2(x^*) = 0$.

Эта функция является часто предлагаемым тестом, используемым во многих работах по методам конечномерной оптимизации. Линии уровня F_2 имеют ярко выраженную

овражную структуру с криволинейным дном оврага, расположенным вдоль параболы $x_2 = x_1^2$ (рис. 1.4.7). Хотя степень овражности в этом случае не очень высока ($\eta \approx 2500$), работа многих алгоритмов затруднена из-за значительного уменьшения скорости сходимости. В частности, непригодными оказываются алгоритмы покоординатного спуска, а также классические методы спуска по антиградиенту.

3. «Асимметричная долина»

$F_3 = [(x_1 - 3)/100]^2 - (x_2 - x_1) + \exp[20\,(x_2 - x_1)]$;

$x^0 = (0; -1)$; $x^* = (3; 2{,}850214)$; $F_3(x^*) = 0{,}199786$.

Минимизация F_3 сопряжена с известными трудностями, так как это пример «неквадратичной» задачи. Линии уровня F_3, представленные на рис. 1.4.8, показывают, что даже локально функция F_3 заметно отличается от квадратичной зависимости.

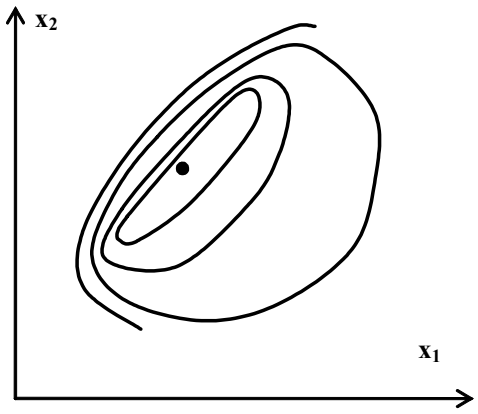


рис. 1.4.8

Большинство алгоритмов позволяют достаточно быстро получить значения функции около 0,2, но соответствующее значение x оказывается неудовлетворительным.

4. Функция Пауэлла

$$F_4 = (x_1 + 10\,x_2^2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4;$$

$x^0 = (3; -1; 0; 1)$; $x^* = (0; 0; 0; 0)$; $F_4(x^*) = 0$.

В точке минимума x* матрица F''_4 вырождена, а в окрестности этой точки – плохо обусловлена, что затрудняет применение методов ньютоновского типа. Дополнительные экспериментальные данные, касающиеся применения различных методов минимизации к этой функции, содержатся в многочисленных публикациях по методам

нелинейного программирования.

5. Функция Зангвилла

$F_5 = (x_1 - x_2 + x_3)^2 + (x_2 - x_1 + x_3)^2 + (x_1 + x_2 - x_3)^2$;

$x^0 = (0{,}5;\ 1;\ 0{,}5)$; $x^* = (0;\ 0;\ 0)$; $F_5(x^*) = 0$.

Функция является примером, для которого неприменим первоначальный вариант известного метода Пауэлла, обычно имеющего достаточно высокую эффективность.

6. Пример, связанный с оценкой экспериментальных данных методом наименьших квадратов:

$$F_6 = 10^4 \sum_{i=1}^{7} \left[\left(\frac{x_1^2 + x_2^2 a_i + x_3^2 a_i^2}{1 + x_4^2 a_i} - b_i \right) / b_i \right]^2,$$

$x^0 = (2{,}7;\ 90;\ 1500;\ 10)$; $x^* = (2{,}714;\ 140{,}4;\ 1707;\ 31{,}51)$; $F_6^* = 318{,}57$.

Значения констант a_i, b_i являются компонентами следующих заданных векторов:

$a = 10^{-3}\ (0;\ 0{,}428;\ 1;\ 1{,}61;\ 2{,}09;\ 3{,}48;\ 5{,}25)$;

$b = (7{,}391;\ 11{,}18;\ 16{,}44;\ 16{,}20;\ 22{,}2;\ 24{,}02;\ 31{,}32)$.

Согласно сведениям, приведенным в книге Д.Химмельблау «Прикладное нелинейное программирование» (М.: Мир, 1975), эта функция приводит к определенным трудностям при работе многих алгоритмов минимизации. Из рассмотренных в работе одиннадцати алгоритмов шесть оказались неспособными построить приемлемое приближение к x*. При этом значения, близкие по функционалу к F_6^*, получались относительно быстро. Среди этих алгоритмов находятся: метод вращения осей Розенброка, метод конфигураций Хука – Дживса, симплексный метод Нелдера – Мида. Последний метод в руководствах по практической оптимизации наиболее часто рекомендуется применять в овражной ситуации. Однако при решении задачи минимизации F_6,

наиболее приближенной к реальным условиям из всех рассмотренных здесь задач, он оказался неприемлемым.

7. Квадратичный функционал с высокой степенью овражности $\eta = 10^{12}$:

$$F_7 = 1/2\sum_{i=1}^{4}\lambda_i\langle x, u_i\rangle^2 - \langle b, x\rangle; \quad b = (1;\ 1;\ 1;\ 1).$$

Собственные числа матрицы Гессе F''_7 равны: $\lambda_1=10^8$, $\lambda_2=\lambda_4=10^{-4}$, $\lambda_3=10^6$. Собственные векторы u_i есть: $u_1 = 1/\sqrt{3}$ (1; –1; 1; 0); $u_2 = 1/\sqrt{6}$ (1; 2; 1; 0); $u_3 = 1/\sqrt{3}$ (1; 0; –1; 1); $u_4 = 1/\sqrt{6}$ (1; 0; –1; –2); x^0=(0;0;0;0); $F_7(x^*) \approx -16\ 667$; $x^* \cong$ (3333,3; 13 333; 10 000; 6666,7).

Точные значения компонент вектора x* задаются выражением

$$x^* = (u_1; u_2; u_3; u_4)\,(\lambda_1^{-1}\langle b, u_1\rangle; \lambda_2^{-1}\langle b, u_2\rangle; \lambda_3^{-1}\langle b, u_3\rangle; \lambda_4^{-1}\langle b, u_4\rangle)^T \approx$$

$$\approx (u_1; u_2; u_3; u_4) \times (0; 10^4\langle b, u_2\rangle; 0; 10^4\langle b, u_4\rangle)^T.$$

Отсюда видно, что компоненты x* в основном определяются малыми собственными числами λ_2, λ_4, и уже небольшая погрешность в их представлении приводит к большой ошибке в результате. Необходимо отметить, что при написании тестовой программы, осуществляющей вычисление значений F_7, следует использовать вышеприведенное представление функции в виде суммы. Применение для этой цели обычного выражения квадратичного функционала $F_7(x) = 1/2\ \langle Ax, x\rangle - (b, x\rangle$, содержащего в явном виде матрицу $A = F_7''$, недопустимо, так как из-за ограниченной точности представления элементов a_{ij} матрицы в памяти компьютера информация о малых собственных числах λ_2, λ_4 теряется на фоне больших λ_1, λ_3.

Указанное обстоятельство приводит к резкой потере эффективности методов ньютоновского типа, основанных на существенном использовании информации о малых собственных числах при явном представлении

аппроксимации матриц Гессе минимизируемого функционала.

Другая особенность функции F_7 заключается в наличии двухмерного оврага, дно которого совпадает с многообразием вида $\{x^* + \alpha u_2 + \beta u_4\}$. Это вносит дополнительные трудности для методов, рассчитанных на минимизацию функционалов с одномерными оврагами.

Тестированию подвергались алгоритмы GZ1, SPAC1, SPAC2, SPACR, SPAC5, SIMPL. Алгоритмы GZ1, SPAC1, SPAC2 были описаны ранее, SPACR реализует метод вращения осей Розенброка. Алгоритм SPAC5 является модификацией SPAC1, использующей для вычисления производных соотношения (1.4.13), (1.4.14). Он применяется для функционалов, имеющих структуру (1.4.8), (1.4.9). При этом первые производные получаются численно на основе двухсторонних приращений с шагом дискретности, вычисляемым на каждом шаге согласно базовому алгоритму SPAC1. В программе SIMPL реализован симплексный метод Нелдера – Мида на основе алгоритма, приведенного в уже упоминавшейся книге Д.Химмельблау.

Начальный шаг дискретности s везде принимался равным 0,1. Он же определял начальные смещения по осям координат при организации процедуры ОПС, а также начальные размеры симплекса в алгоритме SIMPL. После каждого поворота осей в методах SPAC1, SPAC2, SPAC5 настройка шагов по отдельным направлениям спуска осуществлялась исходя из ранее полученных для одноименных осей значений.

Вычисления производились с одинарной точностью.

Таблица 1.4.1

Метод	F	F	F	F	F	F	F
GZ1	6	Н	Н	Н	7	Н	Н
SPACR	4	1	1	5	6	Н	Н
SPAC1	6	3	7	2	1	3	1
SPAC5	3	2	–	1	7	1	–
SIMPL	4	2	–	–	–	Н	Н

В табл.1.4.1 представлены выраженные в горнерах трудоемкости получения результата с погрешностью $\delta \le 3$ %. В качестве δ выбиралась величина

$$\delta = \max \{\delta_x; \delta_F\},$$

где $\delta_x = \max_i \{\delta_i\}$;

$$\delta_i = \begin{cases} (|x_i^* - x_i| / |x_i^*|) \cdot 100 & (x_i^* \neq 0); \\ | x_i^* - x_i | \cdot 100 & (x_i^* = 0). \end{cases}$$

Погрешность по функционалу δ_F вычислялась аналогично δ_i.

Значок «Н» означает, что метод не позволил получить указанный результат за приемлемое машинное время, прочерк – вычисления не проводились.

Полученные результаты позволяют сделать следующие замечания.

Метод ПС, реализованный в алгоритме GZ1, эффективен для относительно простых задач, например для F_1, F_5. Для более сложных функций F_2, F_3 требуются большие затраты машинного времени. В некоторых случаях, особенно для функций типа F_4, F_6, F_7, реализуется ситуация заклинивания на дне оврага.

Алгоритм SPACR эффективен для задач, где размерность дна оврага не превышает единицы. Вместе с алгоритмом SIMPL алгоритм SPACR оказался наиболее эффективным из всех рассмотренных здесь методов при минимизации F_2. Алгоритм SPACR позволил получить удовлетворительное приближение к точному решению для функций F_3, F_4, F_5. Неудовлетворительные результаты были получены при решении шестого и седьмого примеров, хотя и удалось продвинуться к точке

минимума несколько дальше, чем с помощью метода ПС. Причина неэффективности SPACR в данном случае заключается в наличии многомерных оврагов. Таким образом, имеющиеся экспериментальные данные подтверждают сделанные ранее предположения.

Алгоритмы SPAC1, SPAC2, реализующие две различные модификации метода ОПС, оказались приблизительно одинаковыми по затратам машинного времени, поэтому результаты работы SPAC2 отдельно не приводятся. Все рассматриваемые здесь задачи были ими решены с достаточно высокой точностью, хотя для второго и пятого примеров методом Розенброка были получены лучшие результаты. Вариант метода ОПС, реализованный в алгоритме SPAC5, оказался более эффективным, чем SPAC1, при минимизации функций F_1, F_2, F_4, F_6. При решении шестого теста он был наилучшим и по точности определения минимума.

Симплексный метод оказался эффективным при решении первой и второй задачи, однако он не позволил найти минимумы F_6, F_7.

По результатам тестирования, а также основываясь на опыте решения реальных задач, могут быть сделаны следующие выводы.

Наиболее универсальными методами для решения задач, имеющих отмеченные в разд. 1.2.5 особенности, среди всех рассмотренных алгоритмов оказались алгоритмы типа SPAC1, SPAC2. В частности, как показывает практика, они сохраняют работоспособность там, где квазиньютоновский метод Давидона – Флетчера – Пауэлла регистрирует ситуацию локального минимума при фактическом отсутствии последнего.

При указанных в разд. 1.4 подходах к формализации задачи оптимизации приходим к функционалам типа (1.4.8), (1.4.9). В этих случаях целесообразно основывать вычисления на алгоритмах, аналогичных SPAC5.

При оценке полученных результатов тестирования необходимо иметь в виду, что некоторые из приводимых в литературе методов оптимизации используют процедуры одномерной, как правило, квадратичной экстраполяции, дающие им односторонние преимущества при решении задач минимизации функций двух ($n = 2$) переменных. Автор не является сторонником такого подхода, так как эти преимущества немедленно теряются для более реальных ситуаций, когда $n > 2$ и увеличивается вероятность появления многомерных оврагов.

Например, метод Пауэлла в модификации Брента, реализованный в диалоговой системе оптимизации (ДИСО) ВЦ РАН и снабженный указанным механизмом экстраполяции, позволяет получить значение $F_2 \approx 1{,}3 \cdot 10^{-5}$ за $g = 125$ вычислений функции. При этом счет ведется с двойной точностью. В то же время «чистый» алгоритм Пауэлла дает для F_2 результаты, аналогичные полученным с помощью алгоритма SPAC1. Не лучшие результаты получаются устанавливаемым «по умолчанию» в системе ДИСО алгоритмом АР3, реализующим один из вариантов метода сопряженных градиентов.

В заключение отметим, что при решении реальных задач всегда возникает проблема определения момента окончания вычислений. Основная трудность заключается в том, что точно установить степень приближения решения к искомой оптимальной точке почти невозможно, и поэтому необходимо применять какие-либо косвенные критерии сходимости, позволяющие обрывать вычислительный процесс на основе анализа доступной информации. Далее предполагается, что в качестве такой информации используются свойства последовательностей $\{x^k\}$, $\{J_k\}$, генерируемых методом оптимизации. Требуется определить момент завершения вычислений, который, вообще говоря, может просто являться моментом перехода к другой более перспективной вычислительной процедуре. При этом обычно отказываются от часто рекомендуемого анализа последовательности градиентов $\{J'_k\}$ и главным образом по причине ее малой информативности. Как показано в разд. 1.3.1, при решении плохо обусловленных задач градиент может быть достаточно

малым в некоторых точках дна оврага, однако процесс оптимизации в указанных условиях целесообразно продолжать. При этом изменения $||\Delta x||$ и $|\Delta J|$ могут быть весьма значительными. Кроме этого градиент прямо зависит от выбранных масштабов измерения J.

Наиболее надежное решение проблемы окончания процесса оптимизации может быть получено в интерактивном (диалоговом) режиме работы с программой, когда пользователь может всесторонне оценить получаемое решение из соображений, вообще говоря, жестко не связанных с соответствующим значением критерия J.

Например, при решении задачи аппроксимации некоторой функции методами оптимизации может быть проанализирована сама функция, соответствующая параметрам x^k, а не только величина $J(x^k)$, характеризующая усредненную ошибку аппроксимации. Такой наиболее естественный способ оценки результатов позволяет исходить из действительных целей оптимизации, которые подчас не удается адекватно отобразить при формировании единого или даже векторного критерия оптимальности.

Вышеизложенное не исключает необходимости построения критериев остановки, позволяющих автоматически завершать вычисления или сигнализировать о целесообразности перехода от медленно сходящихся оптимизирующих процедур к процедурам, более приспособленным для решения данной задачи. Важность такого типа критериев очевидна, в частности, при разработке сложных оптимизирующих программных систем, обладающих средствами автоматического перехода от одного метода оптимизации к другому с целью выбора наиболее перспективного алгоритма.

Разумной альтернативой всем применяемым критериям остановки является условие, сочетающее требования как на относительную, так и на абсолютную точность: $\left|J_{k+1} - J_k\right| \le \varepsilon_1 \left|J_k\right| + \varepsilon_2, \varepsilon_1 > \varepsilon_M$. Здесь ε_1, ε_2 – заданные значения относительной и абсолютной точности локализации

оптимума по функционалу.

Как ранее было показано, наиболее часто встречающиеся в приложениях функционалы характеризуются медленным изменением при продвижении по дну оврага. Поэтому в этом случае необходимо осуществлять проверку по аргументу $\left|x_i^{k+1} - x_i^k\right| \le \varepsilon_1 \left|x_i^k\right| + \varepsilon_2, i = \overline{1,n}$ или использовать комбинированное условие.

В сложных случаях целесообразно прекращать работу программы только после исчерпания выделенного ресурса времени работы, независимо от свойств последовательностей $\{x^k\}$, $\{J_k\}$, $\{J'_k\}$.

1.5. ГРАДИЕНТНЫЕ СТРАТЕГИИ КОНЕЧНОМЕРНОЙ ОПТИМИЗАЦИИ.

1.5.1. Общая схема градиентных методов. Понятие функции релаксации.

Классические градиентные схемы, основанные на применении антиградиентов в качестве направлений спуска, непосредственно не приспособлены для минимизации овражных функционалов и по этой причине не позволяют получить решение сколько-нибудь сложной задачи конечномерной оптимизации. Однако ниже будет показано, что на их основе могут быть построены методы, обладающие существенно лучшими показателями сходимости в овражной ситуации независимо от характера выпуклости минимизируемых функционалов. С учетом представленных в разд. 1.3 моделей явления овражности эффективность алгоритмов оценивается по свойствам специально вводимых для этих целей функций релаксации, полностью определяющих локальные характеристики методов.

Вычислительные аспекты оптимизации систем с большим числом управляемых параметров рассматриваются в разд. 1.5.5.

Пусть решается задача

$$J(x) \to \min_{x};\quad x \in R^n;\quad J \in C^2(R^n). \qquad (1.5.1)$$

Рассмотрим класс матричных градиентных методов вида

$$x^{k+1} = x^k - H_k(A_k, h_k)J'(x^k) \quad (h_k \in R^1), \qquad (1.5.2)$$

где $A_k = J''(x^k)$, H_k – матричная функция A_k. Предполагается, что в некоторой ζ_k-окрестности $\{x \in R^n | \ ||x - x^k|| \le \zeta_k\}$ точки x^k функционал $J(x)$

достаточно точно аппроксимируется квадратичным функционалом

$$f(x) = 1/2\ \langle A_k x, x\rangle - \langle b_k, x\rangle + c_k, \qquad (1.5.3)$$

где A_k – симметричная, не обязательно положительно определенная матрица. Без существенного ограничения общности можно считать, что $b_k = 0$, $c_k = 0$. Действительно, принимая $\det A_k \neq 0$, $x = x^* + z$, где $x^* = A_k^{-1} b_k$, получим представление

$$f_1(z) = f(x^* + z) = 1/2\ \langle A_k z, z\rangle + \blacksquare. \qquad (1.5.4)$$

При этом константа $\bar{c}_k = c_k - 1/2\left\langle A_k x^*, x^*\right\rangle$ может не учитываться, как не влияющая на процесс оптимизации.

Формула (1.5.2) обладает свойством инвариантности относительно смещения начала координат. Будучи записанной для f(x), она преобразуется в аналогичное соотношение для $f_1(z)$. Для f(x) имеем:

$$x^{k+1} = x^k - H_k(A_k x^k - b_k). \qquad (1.5.5)$$

Принимая $z^k = x^k - x^*$, получаем из (1.5.5) : $z^{k+1} = z^k - H_k A_k z^k$. А это есть запись метода (1.5.2) для функционала f_1.

Ставится задача построения таких матричных функций H_k, чтобы выполнялись условия релаксационности процесса $f(x^{k+1}) < f(x^k)$ и чтобы при этом норма $\|x^{k+1} - x^k\|$ ограничивалась сверху только параметром ζ_k, характеризующим область справедливости локальной квадратичной модели (1.5.3). В настоящее время ситуация такова, что при высокой степени овражности $\eta(x^k)$ для большинства классических схем поиска имеем $\|x^{k+1} - x^k\| << \zeta_k$, что в результате приводит к медленной сходимости.

Определение 5. Скалярная функция $R_h(\lambda) = 1 - H(\lambda, h)\lambda$; $\lambda, h \in R^1$ называется функцией релаксации метода (1.5.2), а ее значения $R_h(\lambda_i)$ на спектре матрицы A_k – множителями релаксации для точки x^k.

В некоторых случаях для сокращения записей индекс «h» в

выражении функции релаксации будет опускаться.

Здесь H (λ, h) означает скалярную зависимость, отвечающую матричной функции H (A_k,h_k) в представлении (1.5.2).

Напомним, что если A – симметричная матрица и

$$A = T\ \mathrm{diag}\ (\lambda_1, \lambda_2, \ldots, \lambda_n)\ T^T,$$

где T – ортогональная матрица, столбцы которой есть собственные векторы матрицы A, то

$$F(A) = T\ \mathrm{diag}\ [F(\lambda_1), F(\lambda_2), \ldots, F(\lambda_n)]\ T^T.$$

Матричная функция имеет смысл, если скалярная функция F(λ) определена в точках $\lambda_1, \lambda_2, \ldots, \lambda_n$.

Теорема 13. Для выполнения условия

$$f(x^{k+1}) \le f(x^k) \qquad (1.5.6)$$

при $\forall x^k \in R^n$ необходимо и достаточно, чтобы

$$|R(\lambda_i)| \ge 1 \quad (\lambda_i < 0); \quad |R(\lambda_i)| \le 1 \quad (\lambda_i > 0) \qquad (1.5.7)$$

для всех собственных чисел λ_i, $i = \overline{1,n}$ матрицы A_k.

Доказательство. Пусть $\{u_i\}$ – ортонормированный базис, составленный из собственных векторов матрицы A_k. Тогда, разлагая x^k по векторам базиса, получаем

$$x^k = \sum_{i=1}^{n} \xi_{i,k} u_i; \quad x^{k+1} = x^k - H_k(A_k, h_k) f'(x^k) = (E - H_k A_k) x^k =$$

$$= \sum_{i=1}^{n} \xi_{i,k} [1 - H_k(\lambda_i, h_k)\lambda_i] u_i = \sum_{i=1}^{n} \xi_{i,k} R(\lambda_i) u_i.$$

Из сравнения выражений

$$f(x^k) = \frac{1}{2} \sum_{i=1}^{n} \xi_{i,k}^2 \lambda_i;$$

$$f(x^{k+1}) = \frac{1}{2} \sum_{i=1}^{n} \xi_{i,k+1}^2 \lambda_i = \frac{1}{2} \sum_{i=1}^{n} \xi_{i,k}^2 \lambda_i R^2(\lambda_i) \qquad (1.5.8)$$

следует, что при выполнении (1.5.7) каждое слагаемое суммы в представлении $f(x^k)$ не возрастает. Достаточность доказана. Докажем необходимость. Пусть существует такой

индекс $i = i_0$, для которого $\lambda_{i_0} < 0, \left|R(\lambda_{i_0})\right| < 1$. Выберем $x^k = u_{i_0}$. Тогда $f(x^k) = 0{,}5\lambda_{i_0} < f(x^{k+1}) = 0{,}5\lambda_{i_0} R^2$, что противоречит условию релаксационности (1.5.2). Аналогично рассматривается второе неравенство (11.1.7). Теорема доказана.

Замечания:

1) Для строгого выполнения неравенства (1.5.6) необходимо и достаточно, кроме выполнения условий (1.5.7) потребовать, чтобы существовал такой индекс $i = i_0$, для которого $\xi_{i_0 k} \neq 0$, и соответствующее неравенство (1.5.7) было строгим.
2) Выражения (1.5.8) позволяют оценить скорость убывания функционала f в зависимости от «запаса», с которым выполняются неравенства (1.5.7). Действительно, обозначим через λ_i^+, λ_i^- положительные и отрицательные собственные числа матрицы A_k. Эти же индексы присвоим соответствующим собственным векторам. Суммирование по соответствующим i будем обозначать Σ^+, Σ^-. Тогда

$$2\,|f(x^k) - f(x^{k+1})| = \Sigma^+ \xi^2_{i,k} \lambda_i^+ [1 - R^2(\lambda_i^+)] + \Sigma^- \xi^2_{i,k} |\lambda_i^-| [R^2(\lambda_i^-) - 1].$$

Из полученного выражения следует, что наибольшее подавление будут испытывать слагаемые, для которых значение множителя релаксации наиболее существенно отличается от единицы [при выполнении условий (1.5.7)].

Далее будут рассматриваться в основном зависимости $R_h(\lambda)$, обладающие свойством

$$R_h(\lambda) \to 1 \quad (h \to 0). \tag{1.5.9}$$

В этом случае из равенства

$$\| x^{k+1} - x^k \| = \sum_{i=1}^{n} \xi^2_{i,k} [R_{h_k}(\lambda_i) - 1]^2 \tag{1.5.10}$$

следует, что для $\forall \zeta_k \in R^1$ всегда можно выбрать такой h_k, что $||x^{k+1} - x^k|| \le \zeta_k$. Таким образом, с помощью параметра h_k можно регулировать норму вектора продвижения в пространстве управляемых параметров с целью предотвращения выхода из области справедливости локальной квадратичной модели (1.5.3).

Иногда для ограничения нормы (1.5.10) параметр h может вводиться в схему оптимизации как множитель в правой части (1.5.2):

$$x^{k+1}(h) = x^k - hH_k J'(x^k), \quad h \in [0,1]. \qquad (1.5.11)$$

При этом $||x^{k+1}(h) - x^k|| = h\,||x^{k+1}(1) - x^k||$, а второе равенство (1.5.10) трансформируется к виду

$$f(x^{k+1}) = 1/2 \sum_{i=1}^{n} \xi_{i,k}^2 \lambda_i \overline{R^2}(\lambda_i),$$

где $\overline{R}(\lambda_i) = (1-h) + hR(\lambda_i)$. Таким образом, новые множители релаксации $\overline{R}(\lambda_i)$ принимают промежуточные значения между 1 и $R(\lambda_i)$, что и требуется для обеспечения свойства релаксационности, определяемого требованиями (1.5.7).

Введенное понятие функции релаксации позволяет с единых позиций оценить локальные свойства различных градиентных схем поиска. Удобство такого подхода заключается также в возможности использования наглядных геометрических представлений.

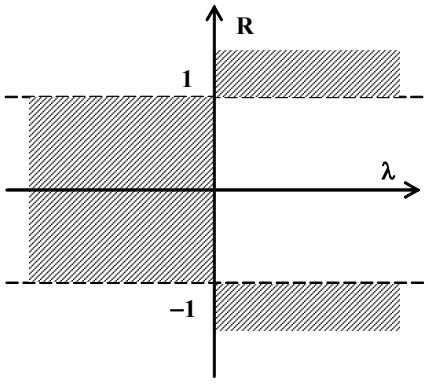


рис. 1.5.1

Подобно областям устойчивости методов численного интегрирования обыкновенных дифференциальных уравнений, построенных на основе тестового (линейного скалярного) уравнения, можно для любого метода (1.5.2) построить функцию релаксации, характеризующую область его релаксационности в множестве собственных чисел. При этом роль тестового функционала играет

квадратичная зависимость (1.5.3). Требуемый характер функции релаксации представлен на рис. 1.5.1рис. 1.5.; заштрихована запрещенная область, где условия релаксационности (1.5.7) не выполняются.

Очень важное свойство функций релаксации заключается в возможности использования соответствующих представлений для синтеза новых процедур из класса (1.5.2), обладающих некоторыми желательными свойствами при решении конкретных классов задач оптимизации.

1.5.2. Классические градиентные схемы.

Рассмотрим некоторые конкретные методы (1.5.2) и отвечающие им функции релаксации.

Простой градиентный спуск (ПГС). Формула метода ПГС имеет вид

$$x^{k+1} = x^k - h\,J'(x^k) \quad (h = \text{const}). \qquad (1.5.12)$$

Соответствующая функция релаксации

$$R(\lambda) = 1 - h\lambda \qquad (1.5.13)$$

линейна и представлена на рис. 11.2.1.

Пусть собственные значения матрицы A_k расположены в замкнутом интервале [m, M], причем $0<m\le M$, так что $\eta(x) = M/m >>1$. В этом случае условие (11.1.9), очевидно, выполняется, а неравенства (11.2.9) сводятся к требованию $\left|R(\lambda_i)\right| \le 1, i = \overline{1,n}$ или

$$\left|1 - h\lambda_i\right| \le 1 \qquad (i = \overline{1,n}). \qquad (1.5.14)$$

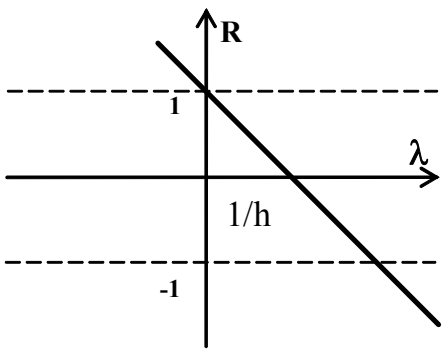


рис. 1.5.1

Из неравенства (1.5.14) следует $h \le 2/M$, $R(m) = 1 - hm \approx 1$. Точка

пересечения прямой $R(\lambda)$ с осью абсцисс есть точка $\lambda = 1/h$ и, чтобы при $\lambda \in [m,M]$ зависимость $R(\lambda)$ находилась в разрешенной области, необходимо выполнение неравенства $1/h \geq M/2$ (рис. 1.5.1). При этом ординаты функции релаксации характеризуют соответствующие множители релаксации, которые в окрестности $\lambda = m$ будут тем ближе к единице, чем больше отношение M/m. Будем считать, что для собственных чисел матрицы A_k выполняются неравенства $\lambda_1 \geq ... \geq \lambda_{n-r} \geq \sigma\,|\lambda_{n-r+1}| \geq ... \geq \sigma\,|\lambda_n|$, $\sigma >> 1$, характерные для овражной ситуации. Тогда для точки $x^k \in Q$, где Q – дно оврага, имеем согласно (1.5.10)

$$\left\| x^{k+1} - x^k \right\| \approx 4 \sum_{i=n-r+1}^{n} \xi_{i,k}^2 \left(\lambda_i / \lambda_1\right)^2 \leq 4\sigma^{-2} \left\| x^k \right\|^2,$$

что может быть существенно меньше ζ_k.

В результате соответствующие малым собственным значениям из окрестности $\lambda = 0$ слагаемые в выражении (1.5.8) почти не будут убывать, а продвижение будет сильно замедленным. Это и определяет низкую эффективность метода (1.5.12).

В области $\lambda < 0$ функция (1.5.13) удовлетворяет условиям релаксации при любом значении параметра h. Параметр h в методе ПГС выбирается из условия монотонного убывания функционала на каждом шаге итерационного процесса. При отсутствии убывания величина h уменьшается до восстановления релаксационности процесса. Существуют различные стратегии выбора h, однако при больших η все эти методы, включая и метод наискорейшего спуска $x^{k+1} = \arg\min_{h \geq 0} J[x^k - hJ'(x^k)]$, малоэффективны, даже при минимизации сильно выпуклых функционалов. Так же как и в методе покоординатного спуска здесь возможна ситуация заклинивания, представленная на рис. 1.5.2.

Как указывалось в разд. 1.3.2, метод ПГС представляет определенный интерес как средство оценки локальной степени овражности в окрестности точки замедления алгоритма. Выведем соответствующие соотношения.

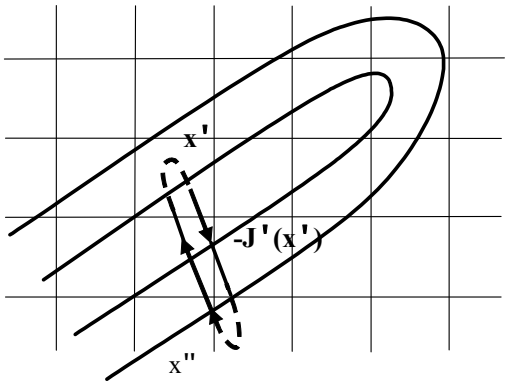


рис. 1.5.2

Пусть замедление метода ПГС при минимизации некоторого функционала J(x) произошло в окрестности некоторой точки x^0. Тогда можно предположить, что достаточно длинный отрезок последовательности $\{x^k\}$, построенный из точки x^0, будет оставаться в области $\|x - x^0\| \le \zeta_0$ и для J(x) справедлива квадратичная аппроксимация

$$f(x) = 1/2\,\langle Ax, x\rangle - \langle b, x\rangle. \qquad (1.5.15)$$

Метод (1.5.12) для аппроксимации (1.5.15) примет вид

$$x^{k+1} = x^k - h(Ax^k - b) = Bx^k + g, \qquad (1.5.16)$$

где $B = E - hA; \quad g = hb.$

Записывая (1.5.16) для двух последовательных номеров k и вычитая полученные равенства, приходим к соотношению

$$y^k = x^{k+1} - x^k = B(x^k - x^{k-1}) = B^k(x^1 - x^0) = B^k y^0. \qquad (1.5.17)$$

Согласно степенному методу определения максимального собственного числа симметричной матрицы в результате проведения процесса (1.5.17) может быть получена оценка максимального собственного числа матрицы B.

$$\left\|y^{k+1}\right\| / \left\|y^k\right\| \to \max_i |\lambda_i(B)| \qquad (k \to \infty). \qquad (1.5.18)$$

Пусть шаг h в итерационном процессе (1.5.16) выбирается из условия релаксационности, тогда можно заключить, что $h \approx 2/M$. Пусть $\lambda_i(A) \in [-m, M]$, $M > m > 0$. В результате имеем $\max |\lambda_i(B)| = 1 + mh$.

Следовательно, для достаточно больших k

$$\left\|y^{k+1}\right\| / \left\|y^{k}\right\| = \left\|J'(x^{k+1})\right\| / \left\|J'(x^{k})\right\| = \mu_k \approx 1 + mh. \qquad (1.5.19)$$

Приходим к требуемой оценке степени овражности функционала J(x) в окрестности точки x^0:

$$\eta(x^0) = M/m \approx 2/(\mu_k - 1).$$

Рассуждая аналогично для случая $\lambda_i(A) \in [m, M]$, получаем вместо (1.5.19) равенство $\mu_k \approx 1 - mh < 1$ и соответствующую оценку $\eta\ (x^0) \approx 2/(1 - \mu_k\rangle$. Общая оценка может быть записана в виде $\eta\ (x^0) \approx 2/\ |1 - \mu_k|$, причем, сравнивая μ_k с единицей, можно установить характер выпуклости J(x) в окрестности точки x^0, что дает дополнительную полезную информацию.

Метод Ньютона основан на построении квадратичной аппроксимации функционала J(x) в окрестности текущей точки x^k:

$$J(x) \approx f(x) = J\ (x^k) + \langle x - x^k, J'\ (x^k)\rangle + 1/2\ \langle J''(x^k)\ (x - x^k), x - x^k\rangle.$$

В качестве x^{k+1} выбирается точка, удовлетворяющая уравнению $f'(x) = 0$. В результате приходим к формуле

$$x^{k+1} = x^k - h_k\, A_k^{-1}\, J'\,(x^k); \qquad A_k = J''(x^k). \qquad (1.5.20)$$

С помощью дополнительно введенного параметра h_k, как указывалось ранее, осуществляется регулировка нормы вектора продвижения $\| x^{k+1} - x^k\|$.

Таким образом, по построению метод Ньютона является оптимальным для квадратичных функционалов при условии положительной определенности матрицы A_k. Минимум сильно выпуклого квадратичного функционала находится за один шаг при $h_k = 1$.

Главный недостаток метода заключается в следующем. Если функционал J(x) не является выпуклым в окрестности точки x^k, то это может послужить причиной расходимости. Действительно, допустим, что функционал J(x) аппроксимируется невыпуклым квадратичным

функционалом. В этом случае ньютоновское направление $p^k = A_k^{-1} J'(x^k)$ указывает на точку x', удовлетворяющую условию $f'(x') = 0$. Такой точкой может оказаться, например, седловая точка, а не точка минимума, как в выпуклой ситуации. В результате направление p^k будет указывать «наверх», а не «вниз» (рис. 1.5.3). Заметим, что полная потеря работоспособности алгоритма Ньютона в невыпуклых экстремальных задачах происходит независимо от наличия или отсутствия овражной ситуации. Такая особенность методов ньютоновского типа существенно ограничивает область их практического применения, так как уже простейшие задачи оптимизации могут приводить к невыпуклым критериям.

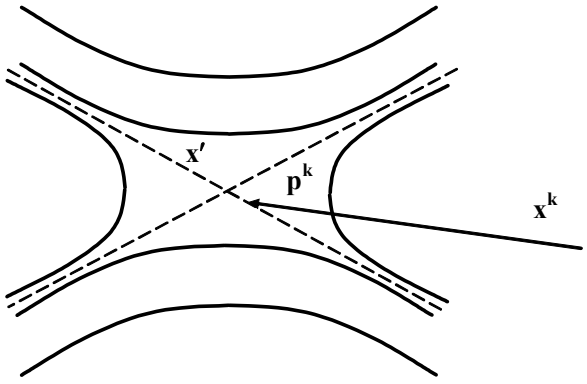


рис. 1.5.3

На языке функций релаксации это обстоятельство проявляется в попадании графика функции релаксации метода (1.5.20)

$$R(\lambda) = 1 - H(\lambda, h_k)\,\lambda = 1 - h_k$$

в запрещенную область, где не выполняются неравенства (1.5.7). Действительно, при $h_k = 1$, что соответствует классическому варианту метода Ньютона без регулировки шага, имеем $R(\lambda) \equiv 0$ при $\forall\lambda \neq 0$. И аналогично при любых значениях h_k прямая релаксации $1 - h_k$ параллельна оси абсцисс и захватывает запрещенную область либо при

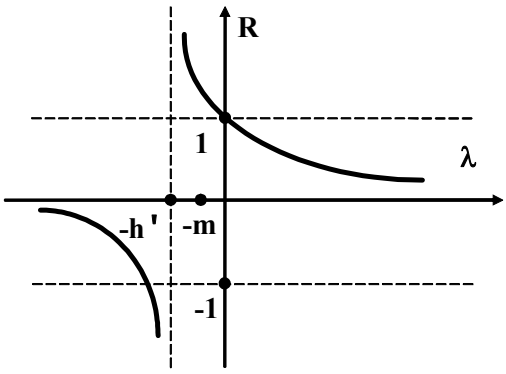


рис. 1.5.4

$\lambda > 0$, либо при $\lambda < 0$. Положение ее при $h = 0$ соответствует остановке процесса. В указанных условиях эффективный выбор h_k оказывается затруднительным.

Аналогичным недостатком обладают многие из аппроксимирующих метод Ньютона квазиньютоновских алгоритмов и совпадающих с ними при минимизации квадратичных функционалов методов сопряженных направлений. Все эти методы по эффективности приближаются к методу ПГС, если функционал $J(x)$ в окрестности x^k не является выпуклым.

Традиционное возражение против метода Ньютона, связанное с необходимостью выполнения операции вычисления вторых производных целевого функционала, для реальных задач оптимизации оказывается менее существенным.

Метод Левенберга. Если известно, что собственные значения матрицы A_k расположены в интервале $[-m, M]$, где $M>>m$, то можно построить метод, имеющий нелинейную функцию релаксации (рис. 1.5.4)

$$R(\lambda) = h' / (h' + \lambda) \qquad (h' > 0,\ h = 1/h'), \qquad (1.5.21)$$

удовлетворяющую требованиям (1.5.7) при $\forall\lambda \in [-m, M]$, если $h' > m$.

Соответствующий метод предложен Левенбергом и имеет функцию

$$H(\lambda, h) = [1 - R(\lambda)] / \lambda = (h' + \lambda)^{-1}.$$

Схема метода с указанной функцией H имеет вид

$$x^{k+1} = x^k - [h'E + J''(x^k)]^{-1} J'(x^k). \qquad (1.5.22)$$

Скаляр h' на каждом шаге итерационного процесса подбирается так, чтобы матрица $h'E+J''(x^k)$ была положительно определена и чтобы $\left\|x^{k+1} - x^k\right\| \le c_k$, где c_k может меняться от итерации к итерации.

Из последнего выражения следует, что мы имеем, по существу, классическую регуляризованную форму метода Ньютона с параметром регуляризации h'. Данная форма метода Ньютона применялась

Левенбергом для решения задач метода наименьших квадратов. Существенно позже данный метод применялся Маркуардтом для решения общих задач нелинейной оптимизации и иногда встречается его описание в литературе под названием «метод Маркуардта».

Реализация метода (1.5.22) сводится к решению на каждом шаге линейной алгебраической системы

$$\left[h'E + J''(x^k)\right]\Delta x^k = -J'(x^k) \qquad \left(\Delta x^k \underline{\underline{\Delta}} x^{k+1} - x^k\right). \tag{1.5.23}$$

Главный недостаток метода заключается в необходимости достаточно точного подбора параметра h', что сопряжено с известными вычислительными трудностями. Значение m, как правило, неизвестно и не может быть вычислено с приемлемой точностью. При этом оценка для m существенно ухудшается при возрастании размерности n. Лучшее, что обычно можно сделать на практике, это принять

$$h' \geq \max\left\{\varepsilon_M n\|J_k''\|, |\min\lambda_i(J_k'')|\right\}. \tag{1.5.24}$$

Правая часть неравенства (1.5.24) обусловлена тем, что, как уже указывалось в разд. 1.3.2, абсолютная погрешность представления любого собственного числа матрицы J''_k ввиду ограниченности разрядной сетки равна

$$|d\lambda_i| \leq n\,\lambda_1\,\varepsilon_M \approx n\,||J''_k||\,\varepsilon_M.$$

При невыполнении условия $h' > m$ система (1.5.23) может оказаться вырожденной. Кроме этого слева от точки $\lambda = -h'$ функция релаксации быстро входит в запрещенную область и метод может стать расходящимся. Попытки использования алгоритмического способа более точной локализации h' приводят к необходимости многократного решения плохо обусловленной линейной системы (1.5.23) с различными пробными значениями h'.

Легко видеть, что число обусловленности матрицы $h'E + J''(x^k)$ может

превышать cond[$J''(x^k)$]. Действительно, потребуем, например, чтобы R (–m) = 10 для обеспечения заданной скорости убывания J(x). Определим необходимое значение параметра h'. Имеем 1/(1–hm) = 10 или $h' = 1/h = m/0{,}9$. В этом случае $\lambda_{min}(h'E + J''_k) = -m + m/0{,}9 = m/9 > 0$. Принимая $\lambda_{max}(h'E + J''_k) \approx \lambda_{max}(J''_k)$, получаем, что $\text{cond}(h'E + J''_k) \approx 9\,\lambda_{max}(J''_k)/|\lambda_{min}(J''_k)$.

При выборе заведомо больших значений h', что реализуется, например, когда определяющим в (1.5.24) является первое выражение в скобках, имеем $m << h'$ и $|R(-m)| \approx 1$, что приводит к медленной сходимости. Ограничение h' снизу не позволяет также уменьшить до желаемого значения множитель релаксации для $\lambda > 0$.

Эти трудности возрастают при аппроксимации производных конечными разностями, так как при малых значениях $J'(x^k)$ для точек x^k, расположенных на дне оврага, приходим к необходимости получать компоненты вектора градиента, как малые разности относительно больших величин порядка $J(x^k)$. В результате компоненты вектора Δx^k будут находиться с большими относительными погрешностями порядка $\eta(x^k)|\varepsilon_M J(x^k)| / \|J'(x^k)\|$. Коэффициент овражности η в этом случае играет роль своеобразного коэффициента усиления погрешности. Для метода Ньютона справедливо аналогичное замечание. В то же время для метода ПГС точность задания $J'(x^k)$ может оказаться достаточной для правильного указания направления убывания J(x).

Методы, рассмотренные в последующих разделах этой главы, также используют в своей схеме производные, которые вычисляются с теми же погрешностями. Однако их структура такова, что в соответствующих вычислительных схемах не используются окончательные результаты решения плохо обусловленных линейных алгебраических систем. Например, в методах с экспоненциальной релаксацией на участках выпуклости J(x) решение эквивалентной линейной системы выполняется

итеративно, причем каждая итерация имеет «физический» смысл, что дает возможность непрерывно контролировать точность вычислений и прерывать процесс, когда накопленная ошибка начинает превышать допустимый уровень.

Несмотря на отмеченные недостатки, метод (1.5.23) часто оказывается достаточно эффективным и его присутствие в библиотеке методов оптимизации следует признать весьма желательным. Дополнительное положительное свойство соответствующих алгоритмов связано с возможностью их обобщения на решение «больших» задач оптимизации, рассмотренных в разд. 1.5.5.

1.5.3. Методы с экспоненциальной функцией релаксации.

Изучаемые в этом разделе методы могут быть построены по принципу «непрерывных методов» из общей теории численного интегрирования жестких систем обыкновенных дифференциальных уравнений с помощью специальных «системных» алгоритмов численного интегрирования, предложенных Ю.В.Ракитским. В настоящей работе использован другой подход к построению алгоритмов, основанный на понятии функции релаксации.

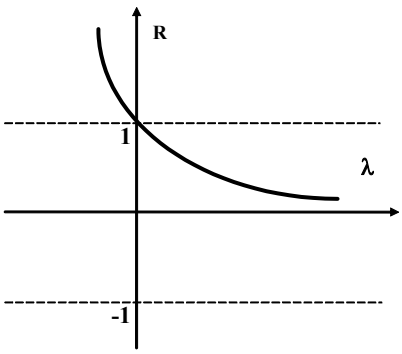


рис. 1.5.5

Исходя из вышеизложенных требований к функциям релаксации, естественно рассмотреть экспоненциальную зависимость вида (рис. 1.5.5)

$$R(\lambda) = \exp(-\lambda h) \quad (h > 0), \qquad (1.5.25)$$

для которой условие (1.5.7) выполняется при любых значениях параметра h. Кроме того, реализуется предельное соотношение (1.5.9), что позволяет эффективно регулировать норму вектора продвижения независимо от

расположения спектральных составляющих матрицы A_k на вещественной оси λ.

Функция (1.5.25) обобщает ранее рассмотренные функции релаксации и является в определенном смысле оптимальной. Действительно, разлагая экспоненту в ряд Тейлора и ограничиваясь двумя первыми членами разложения, получаем $\exp(-\lambda h) = 1/\exp(\lambda h) \approx 1/(1+\lambda h)$, что совпадает с выражением (1.5.21). И аналогично, принимая $\exp(-\lambda h) \approx 1-\lambda h$, приходим к зависимости (1.5.13). Для достаточно больших значений параметра h имеем $\exp(-\lambda h) \approx 0$ при любых $\lambda \geq m > 0$, что позволяет говорить о вырождении метода в классический метод Ньютона без регулировки шага.

Для построения метода в виде (1.5.2) необходимо определить соответствующую функцию $H(\lambda,h)$. Имеем $\lambda H(\lambda,h) = 1 - R(\lambda) = 1 - \exp(-\lambda h)$.

Принимая $\lambda \neq 0$, получаем

$$H(\lambda,h) = \lambda^{-1}\left[1 - \exp(-\lambda h)\right] = \int_0^h \exp(-\lambda\tau)d\tau . \qquad (1.5.26)$$

Доопределяя $H(0,h)$ из условия непрерывности, получаем $H(0,h) = h$. В результате схема метода с экспоненциальной релаксацией (ЭР) примет вид

$$x^{k+1} = x^k - H(A_k, h_k)\, J'(x^k), \qquad A_k = J''(x^k); \qquad (1.5.27)$$

$$H(A,h) = \int_0^h \exp(-A\tau)d\tau . \qquad (1.5.28)$$

Параметр h_k определяется равенством

$$h_k = \arg\min_{h\geq 0} J\left[x^k - H(A_k,h)J'(x^k)\right], \qquad (1.5.29)$$

однако возможны и другие способы выбора h_k.

Принципиальная схема метода ЭР была получена исходя из анализа

локальной квадратичной модели минимизируемого функционала. Представляет интерес выяснение возможностей метода в глобальном смысле, без учета предположений о квадратичной структуре J(x).

Можно доказать, что алгоритм (1.5.27), (1.5.28) сходится почти при тех же ограничениях на минимизируемый функционал, что и метод наискорейшего спуска, имея в определенных условиях существенно более высокую скорость сходимости.

Следующая теорема устанавливает факт сходимости метода ЭР для достаточно широкого класса невыпуклых функционалов в предположении достижимости точки минимума (второе условие) и отсутствия точек локальных минимумов (третье условие).

Теорема 14. Пусть:

1) $J(x) \in C^2(R^n)$;

2) множество $X_* = \{x^* \mid J(x^*) = = \min J(x)\}$ непусто;

3) для любого $\varepsilon>0$ найдется такое $\delta > 0$, что $\|J'(x)\| \geq \delta$, если $x \notin S(X_*)$, где $S(X_*) = \{x|d(x, X_*) \leq \varepsilon\}$; $d(x, X_*) = \min\limits_{x^* \in X_*} \|x - x^*\|$;

4) для любых $x, y \in R^n$ имеем $\|J'(x + y) - J'(x)\| \leq l \|y\|$ $(l > 0)$;

5) собственные числа матрицы $J''(x)$ заключены в интервале $[-M, M]$, где $M > 0$ не зависит от x.

Тогда независимо от выбора начальной точки x^0 для последовательности $\{x^k\}$, построенной согласно выражениям (1.5.27), (1.5.28), выполняются предельные соотношения:

$$\lim d(x^k, X_*) = 0; \qquad (1.5.30)$$

$$\lim J(x^k) = J(x^*), \quad k \to \infty. \qquad (1.5.31)$$

Доказательство. Используются соотношения

$$J(x + y) = J(x) + \int_0^1 \langle J'(x + \vartheta y), y\rangle d\vartheta;$$

$$\left|\int_0^1 \langle x(\vartheta), y(\vartheta)\rangle d\vartheta\right| \le \int_0^1 \|x(\vartheta)\|\|y(\vartheta)\| d\vartheta$$

Обозначая $J_k = J(x^k)$, $J'_k = J'(x^k)$, $A_k = J''(x^k)$, получаем

$$J_k - J\left[x^k - H(A_k, h)J'_k\right] = \int_0^1 \left\langle J'\left[x^k - \vartheta H(A_k, h)J'_k\right], H(A_k, h)J'_k\right\rangle d\vartheta = \left\langle H(A_k, h)J'_k, J'_k\right\rangle -$$

$$- \int_0^1 \left\langle J'_k - J'(x^k - \vartheta H(A_k, h)J'_k), H(A_k, h)J'_k\right\rangle d\vartheta \ge \left\langle H(A_k, h)J'_k J'_k\right\rangle - l\left\|H(A_k, h)J'_k\right\|^2 \int_0^1 \vartheta d\vartheta =$$

$$= \left\langle H(A_k, h)J'_k, J'_k\right\rangle - 0{,}5l\left\|H(A_k, h)J'_k\right\|^2 \ge \rho\left\|J'_k\right\|^2 - 0{,}5l\left\|J'_k\right\|^2 R^2 = \alpha\left\|J'_k\right\|^2;$$

$$\alpha = \rho - (1/2)R^2. \qquad (1.5.32)$$

При этом использованы неравенства

$$\rho\|y\|^2 \le \left\langle H(A_k, h)y, y\right\rangle \le R\|y\|^2, \qquad (1.5.33)$$

где ρ, R – минимальное и максимальное собственные числа положительно определенной матрицы $H(A_k, h)$.

Левое неравенство (1.5.33) следует из представления минимального собственного числа λ любой симметричной матрицы B в виде $\lambda = \min_{x \ne 0}\left[\langle Bx, x\rangle / \langle x, x\rangle\right]$, а правое – из условия согласования $\|Bx\| < \|B\| \, \|x\|$ сферической нормы вектора $\|x\| = \sqrt{\langle x, x\rangle}$ и спектральной нормы симметричной матрицы $\|B\| = \max_i |\lambda_i(B)|$, где $\lambda_i(B), i = \overline{1, n}$ – собственные числа матрицы B. Для значений ρ и R получим:

$$\rho = \min_i \lambda_i[H(A, h)] = \min_i \int_0^h \exp[-\lambda_i(A)\tau]d\tau;$$

$$R = \max_i \int_0^h \exp[-\lambda_i(A)\tau]d\tau.$$

Согласно пятому условию имеем

$$\int_0^h \exp(-M\tau)d\tau \le \int_0^h \exp[-\lambda_i(A)\tau]d\tau \le \int_0^h \exp(M\tau)d\tau,$$

поэтому

$$\rho \ge \int_0^h \exp(-M\tau)d\tau = M^{-1}[1-\exp(-Mh)];$$

$$R \le \int_0^h \exp(M\tau)d\tau = M^{-1}[\exp(Mh)-1];$$

$$\alpha = \rho - 0{,}5lR^2 \ge M^{-1}[1-\exp(-Mh)] - 0{,}5lM^{-2}[\exp(Mh)-1]^2.$$

Принимая h=1/M и считая без ограничения общности, что M > le(e–1)/2, где e – основание натуральных логарифмов, получаем

$$\alpha \ge (e-1)[2M - le(e-1)]/(2M^2e) > 0. \qquad (1.5.34)$$

Из соотношений (1.5.29), (1.5.32), (1.5.34) следует

$$J_k - J_{k+1} \ge J_k - J\left[x^k - H(A_k, M^{-1})J_k\right] \ge \alpha\left\|J_k\right\|^2. \qquad (1.5.35)$$

Следовательно, последовательность $\{J_k\}$ является монотонно невозрастающей и ограниченной снизу величиной J(x*), поэтому она имеет предел и $J_{k+1} - J_k \to \infty$ при $k \to \infty$. Из выражения (1.5.35) следует $\|J'_k\|^2 \le \alpha^{-1}(J_k - J_{k+1})$, поэтому $\|J'_k\| \to 0$ при $k \to \infty$. А так как по условию $\|J'_k\| \ge \delta$ при $x^k \notin S(X_*)$, то найдется такой номер N, что $x^k \in S(X_*)$ при $k \ge N$ и, следовательно, справедливо утверждение (1.5.30).

Обозначим через $\bar{x}^k$ проекцию x^k на множество X_*. Тогда по теореме о среднем

$$J_k - J(\bar{x}^k) = \left\langle J'(x^{kc}), x^k - \bar{x}^k \right\rangle,$$

где

$$x^{kc} = \bar{x}^k + \lambda_k(x^k - \bar{x}^k), \ \lambda_k \in [0,1].$$

Учитывая, что $J'(\bar{x}^k) = 0,$ получаем

$$J_k - J(\bar{x}^k) = \left\langle J'(x^{kc}) - J'(\bar{x}^k), x^k - \bar{x}^k \right\rangle \le$$

$$\left\|J'(x^{kc}) - J'(\bar{x}^k)\right\| \cdot \left\|x^k - \bar{x}^k\right\| \le ld^2(x^k, X_*).$$

Из соотношения (1.5.30) получаем (1.5.31). Теорема доказана.

Замечания:

1) Утверждения теоремы, очевидно, выполняются, если h_k выбирать не из условия (1.5.29), а из условия

$J[x^k - H(A_k, h_k)J'_k] = \min_{h\in[0,\bar{h}]} J[x^k - H(A_k, h)J'_k]$, где $\bar{h} > 0$ – произвольное число. Действительно, легко видеть, что неравенство (1.5.35) только усилится, если брать любое другое значение h_l (может быть даже большее чем 2/[le(e – 1)]) с меньшим значением функционала, чем при h = 1/M, и в то же время, если при h = 1/M сходимость имеет место, то она сохраняется и при меньших значениях h. Последнее следует из возможности выбора сколь угодно больших значений M при установлении сходимости.

2) Соотношения (1.5.30), (1.5.31) сохраняются также при замене условия (1.5.29)на следующее:

$$J_{k+1} = J[x^k - H(A_k, h_k)J'_k] \le (1-\gamma_k)J'_k + \gamma_k \min_{h>0} J[x^k - H(A_k, h)J'_k]$$

$$(0 < \gamma \le \gamma_k \le 1). \qquad (1.5.36)$$

Действительно, из условия (1.5.36) будем иметь

$$J_k - J_{k+1} \ge \gamma_k \{J_k - \min_{h>0} J[x^k - H(A_k, h)J'_k]\} \ge \gamma_k \{J_k - J[x^k - H(A_k, h)J'_k]\}$$

и согласно (1.5.32)

$$J_k - J_{k+1} \ge \gamma_k \alpha \|J'_k\|^2 = \bar{\alpha}\|J'_k\|^2, \quad \bar{\alpha} > 0.$$

Получено неравенство, аналогичное (1.5.35), и далее доказательство проводится по той же схеме с заменой α на $\bar{\alpha}$.

При сильной выпуклости функционала J(x) удается получить оценку скорости сходимости.

Теорема 15. Пусть: 1) $J(x) \in C^2(R^n)$; 2) для любых $x,y \in R^n$ выполняются условия $\lambda\|y\| \le \langle J''(x)y, y\rangle \le \Lambda\|y\|^2$, $\|J''(x+y) - J''(y)\| \le L\|x\|$, $\lambda > 0$, $L \ge 0$. Тогда независимо от выбора начальной точки x^0 для метода

(1.5.27) справедливы соотношения (1.5.30), (1.5.31) и оценка скорости сходимости

$$||x^{k+1} - x^k|| \le (\Lambda/\lambda)^{1/2} L ||x^k - x^*||^2 / (2\lambda).$$

Доказательство здесь не приводится.

Таким образом, установлена квадратичная скорость сходимости, характерная для методов ньютоновского типа.

1.5.4. Реализация и область применимости методов с экспоненциальной функцией релаксации.

Алгоритм вычисления матричных функций (1.5.28) может быть основан на использовании известного рекуррентного соотношения

$$H(A,2h) = H(A,h) [2E - AH(A,h)]. \qquad (1.5.37)$$

Так как все рассматриваемые матричные функции симметричны и, следовательно, обладают простой структурой, то для доказательства (1.5.37) достаточно проверить его для соответствующих скалярных зависимостей, что тривиально.

Формула (1.5.37) используется в вычислительной практике также для получения обратной матрицы A^{-1}, так как выполняется предельное соотношение $H(A,h) \to A^{-1} \quad (h \to \infty)$.

Это еще раз указывает на связь метода ЭР с методом Ньютона, который является предельным вариантом рассматриваемого алгоритма при условии положительной определенности матрицы A.

Выбор параметра h при известной матрице A_k или ее аппроксимации может осуществляться различными способами. В каждом из них приближенно реализуется соотношение (1.5.29). Наиболее простой прием заключается в следующем.

Задают некоторую малую величину h_0, такую, чтобы матрицу

$H(A_k, h_0)$ можно было заменить отрезком соответствующего степенного ряда

$$H(A_k, h_0) \approx h_0 \sum_{i=1}^{m} (-A_k h_0)^{i-1} / i! \qquad (1.5.38)$$

Далее последовательно наращивают h с помощью соотношения (1.5.37), вычисляя каждый раз значение $J[x^k - H(A_k, 2^q h_0)J'_k]$, $q = 0, 1, ...$ Процесс продолжается до тех пор, пока функция убывает. Точка с минимальным значением J принимается за x^{k+1}. При этом вместо точной реализации соотношения (1.5.29) оптимальный шаг выбирается на дискретной сетке значений $h_q = 2^q h_0$, $q = 0, 1, 2, ...$ Как правило, предельное значение q не превышает 30. Рассмотренная реализация метода ЭР носит название «системного алгоритма оптимизации».

Во многих случаях более эффективной оказалась реализация метода с элементами адаптации, в которой значение J не вычислялось для всех промежуточных значений q. Функционал вычислялся только для трех значений q: $q^* - 1$, q^*, $q^* + 1$, где q^* – оптимальное значение q, полученное на предыдущей итерации по k. На первой итерации для определения q^* необходимо вычислить все значения J.

С целью более точной локализации минимума на каждом шаге по k могут использоваться процедуры одномерного поиска по h, например, метод золотого сечения. Для этого вышеизложенным грубым способом определяется промежуток $[h_{min}, h_{max}]$, содержащий оптимальное в смысле условия (1.5.43) значение h_*. Примем

$$\varphi(h) = J[x^k - H(A_k, h)J'(x^k)].$$

Тогда

$$h_{min} = 2^{q'} h_0, h_{max} = 2^{q'+2} h_0, h_* = 2^{q'+1} h_0,$$

причем предполагается, что

$$\varphi(h_{min}) > \varphi(h_*), \quad \varphi(h_{max}) > \varphi(h_*).$$

Фиксируя число пересчетов q', получаем, что, выбирая $h'_0 \in [h_0, 4h_0]$, имеем $h = 2^{q'} h'_0 \in [h_{min}, h_{max}]$. Далее можно принять $\varphi(h) = \psi(h'_0)$, и задача сводится к стандартной задаче минимизации функции одной переменной $\psi(h'_0)$ на заданном промежутке.

Для приближенного вычисления матрицы A_k вторых производных функционала $J(x)$ могут применяться любые методы, изложенные в разд. 1.4.3. Рассмотрим наиболее универсальный алгоритм, основанный на конечно-разностных соотношениях. В результате вычисления по формулам (1.4.15), (1.4.16) приходим к матрице $A_k = D_k/\beta_k^2$ и вектору $J'_k = f_k/\beta_k$, где $\beta_k=2s_k$, s_k – шаг дискретности. Как уже говорилось, производить деление матрицы D_k на β_k^2 или вектора f_k на β_k с целью получения A_k и J'_k нецелесообразно. Поэтому далее принципиальная схема метода ЭР будет преобразована к виду, удобному для непосредственного применения D_k и f_k вместо A_k и J'_k.

Имеем

$$H(D_k, h) = \int_0^h \exp(-D_k\tau)d\tau = \beta_k^{-2}\int_0^h \exp(-A_k\beta_k^2\tau)d\beta_k^2\tau = \beta_k^{-2}\int_0^{\beta_k^2 h} \exp(-A_k t)dt$$

или

$$\beta_k^2 H(\beta_k^2 A_k, h) = H(A_k, h_k); \; h_k = \beta_k^2 h. \qquad (1.5.39)$$

С учетом (1.5.39) основное соотношение (1.5.27) приводится к виду

$$\begin{aligned} x^{k+1} &= x^k - H(A_k, h_k)J'(x^k) = \\ &= x^k - \beta_k^2 H(D_k, h_k/\beta_k^2)f_k/\beta_k = \\ &= x^k - 2s_k H(D_k, h)f_k; \end{aligned} \qquad (1.5.40)$$

$h = h_k/4s_k^2$.

Имеем также

$$H(D_k, 2h)=H(D_k, h)[2E - D_kH(D_k,h)].$$

Оптимальное значение h в (1.5.40) находится непосредственно из соотношения

$$J(x^{k+1}) = \min_{h>0} J(x^{k+1} - 2s_k H(D_k, h) f_k].$$

При использовании разностного уравнения (1.5.40) упрощенная вычислительная схема метода с экспоненциальной релаксацией может быть сведена к следующей последовательности действий.

Алгоритм RELAX.

Шаг 1. Ввести исходные данные x^0,s.

Шаг 2. Принять x := x^0; J := J(x); x^l = x; J_l := J.

Шаг 3. Вычислить матрицу D = $\{d_{ij}\}$ и вектор f = $\{f_i\}$ в точке x по формулам

$$d_{ij} = J(x + se_i + se_j) - J(x - se_i + se_j) - J(x + se_i - se_j) + J(x - se_i - se_j)$$

$$(i, j = \overline{1, n}); \qquad (1.5.41)$$

$$f_i = J(x + se_i) - J(x - se_i) \qquad [i = \overline{1, n}; e_i = (0,\ldots,1,\ldots,0)]; \qquad (1.5.42)$$

принять h_0 := 0,1/ ||D||.

Шаг 4. Принять k := 0. Вычислить матрицу H = H(D,h_0):

$$H = \sum_{i=1}^{7} (-D)^{i-1} h_0^i / i! \qquad (1.5.43)$$

Шаг 5. Принять x^t :=x – 2sHf; J_t := J(x^t); k := k+1.

Шаг 6. Если $J_t < J_l$, принять $x^l := x^t$; $J_l := J_t$.

Шаг 7. Если k > 20, перейти к шагу 8, иначе положить H := H(2E – DH) и перейти к шагу 5.

Шаг 8. Проверить условия окончания процесса оптимизации в целом; если они выполняются, остановить работу алгоритма; в противном случае принять x := x^l; J := J_l и перейти к шагу 3.

Заметим, что выбор на шаге 3 параметра $h_0 = 0{,}1 /||D||$ эквивалентен при $A = J''(x)$ равенству $h_0 = 0{,}1/||A||$ в исходной схеме алгоритма. А последнее равенство, как это следует из результатов, полученных при доказательстве теоремы 14, гарантирует убывание функционала:

$$J[x^k - H(A_k, h_0)\, J'_k] < J(x^k). \qquad (1.5.44)$$

Действительно, как было показано ранее, для выполнения неравенства (1.5.44) достаточно принять $h_0 \le 1/M$, где $M > le(e - 1)/2 \approx 2{,}3l$. Для квадратичного функционала, аппроксимирующего $J(x)$ в окрестности точки x, имеем $l = ||A||$, где $A = J''(x)$. Поэтому можно выбрать h_0 из условия $h_0 \le 1/(2{,}3\ ||A||) \approx 0{,}4/||A||$. Замена коэффициента 0,4 на 0,1 позволяет более точно реализовать шаг 4 алгоритма, одновременно гарантируя выполнение (1.5.44).

Параметр s_k может меняться от итерации к итерации в зависимости, например, от значения нормы $||x^k - x^{k-1}||$ так, как это было описано в разд. 1.4.3. Возможны и другие способы регулировки шага.

Обратимся к анализу влияния погрешностей вычислений при реализации методов ЭР.

Рассмотрим итерационный процесс, определяемый рекуррентным соотношением

$$x^{k+1} = x^k - H(A_k,h_k)A_k x^k = g(A_k)x^k, \qquad (1.5.45)$$

где $g(A) = E - H(A,h)A$. Такой процесс является упрощенной моделью метода ЭР, характеризуя его локальные свойства. Оценим влияние погрешностей в представлении матрицы A_k на характеристики релаксационности последовательности $\{f(x^k)\}$.

Кроме предположения о квадратичном характере $J(x)$ в окрестности точки x^k, неявно введено еще одно допущение. Именно, заменяя в соотношении (1.5.45) матрицу A_k на возмущенную матрицу $A + dA$ (индекс «k» у матрицы далее будем опускать), предполагаем, что ошибки в вычислении J'' и J' определенным образом согласованы. В действительности эквивалентное возмущение dA матрицы, определяющей

градиент Ax^k, может не совпадать с возмущением матрицы J'', так как J' и J'' вычисляются раздельно. Однако с позиций последующего анализа это отличие не является принципиальным.

Предположим, что собственные числа матрицы A разделены на две группы

$$\lambda_1 \geq ... \geq \lambda_{n-r} >> |\lambda_{n-r+1}| \geq ... \geq |\lambda_n|. \qquad (1.5.46)$$

Возмущение dA матрицы A приводит к появлению возмущений $d\lambda_i$ для собственных чисел и возмущений du_i для отвечающих им собственных векторов. Согласно результатам, полученным в разд. 1.4.2, будем считать, что вариации собственных векторов происходят в пределах линейных многообразий

$$M_1 = \sum_{i=1}^{n-r} \alpha_i u_i; \qquad M_2 = \sum_{j=n-r+1}^{n} \alpha_j u_j,$$

порожденных собственными векторами $\{u_i, i = 1,\ldots, n-r\}$ и $\{u_j, j = n-r+1,\ldots, n\}$ исходной невозмущенной матрицы A. В этом случае матрицы A и A + dA одновременно не приводятся к главным осям, что вносит дополнительный элемент сложности в анализ влияния погрешностей.

Пусть

$$U^T A U = \mathrm{diag}\,(\lambda_i); \quad U = (u_1, u_2, ..., u_n); \qquad (1.5.47)$$

$$\omega^T (A + dA)\omega = \mathrm{diag}\,(\lambda_i + d\lambda_i), \quad \omega = (\omega_1, \omega_2, ..., \omega_n).$$

Имеем теперь

$$g(A) = E - \omega D_1 \omega^T \omega D_2 \omega^T = E - \omega D_3 \omega^T,$$

где

$$D_1 = \mathrm{diag}\left\{\int_0^h \exp[(-\lambda_i - d\lambda_i)\tau]d\tau\right\};$$
$$D_2 = \mathrm{diag}(\lambda_i + d\lambda_i);$$
$$D_3 = D_1 D_2.$$

Таким образом, матрица g(A) имеет собственные векторы ω_i и соответствующие им собственные числа $\lambda_i(g) = 1 - \lambda_i(D_3)$.

Принимая

$$x^k = \sum_{i=1}^n \xi_{i,k}\omega_{i;} \quad \omega_i = \sum_{j=1}^n \alpha_{ji} u_i,$$

получаем

$$f(x^k) = 1/2\left\langle Ax^k, x^k\right\rangle = 1/2\sum_{i=1}^n \xi_{i,k}^2\left(\sum_{j=1}^n \alpha_{ji}\right)^2 \lambda_i.$$

Аналогично имеем

$$x^{k+1} = g(A)x^k = \sum_{i=1}^n \xi_{i,k}\lambda_i(g)\omega_i = \sum_{i=1}^n \xi_{i,k+1}\omega_i;$$
$$f(x^{k+1}) = 1/2\sum_{i=1}^n \xi_{i,k+1}^2\left(\sum_{j=1}^n \alpha_{ji}\right)^2 \lambda_i = 1/2\sum_{i=1}^n \xi_{i,k}^2\left(\sum_{j=1}^n \alpha_{ji}\right)^2 \lambda_i\lambda_i^2(g),$$

где

$$\lambda_i(g) = 1 - (\lambda_i + d\lambda_i)\int_0^{h_k} \exp[(-\lambda_i - d\lambda_i)\tau]d\tau = \exp[(-\lambda_i - d\lambda_i)h_k]. \quad (1.5.48)$$

Для выполнения неравенства $f(x^{k+1}) \le f(x^k)$ согласно теореме 13 должны выполняться условия релаксационности

$$|\lambda_i(g)| \le 1 \ (\lambda_i > 0); \quad |\lambda_i(g)| \ge 1 \quad (\lambda_i < 0). \quad (1.5.49)$$

Теперь легко видеть, что если возмущение $d\lambda_i$ таково, что собственное число меняет знак:

$$\mathrm{sgn}\,(\lambda_i) \ne \mathrm{sgn}\,(\lambda_i + d\lambda_i), \quad (1.5.50)$$

то условия (1.5.49), вообще говоря, нарушаются. Это приводит к резкому

замедлению сходимости, а в некоторых случаях к полной остановке процесса оптимизации.

Пусть вариация dA матрицы A вызывается только погрешностями округления. Тогда неравенство (1.5.50) невозможно, если все малые собственные числа по модулю ограничены снизу величиной $n\lambda_1\varepsilon_M$. Действительно, в этом случае $\operatorname{sgn}(\lambda_i) = \operatorname{sgn}(\lambda_i + d\lambda_i)$, так как $|d\lambda_i| \le n\lambda_1\varepsilon_M \le |\lambda_i|$. Отсюда имеем следующее ограничение степени овражности функционалов, эффективно минимизируемых методами ЭР:

$$\eta(x^k) \le 1/(n\varepsilon_M). \qquad (1.5.51)$$

Проведенный анализ показывает, что погрешности вычислений при достаточно больших значениях η могут приводить к случайному характеру множителей релаксации для малых собственных чисел, что определяет резкое снижение эффективности метода. Из соотношения (1.5.51) следует, что трудности возрастают при увеличении размерности n решаемой задачи и уменьшении длины разрядной сетки компьютера. Вычисления с двойной точностью приводят к оценке $\eta(x^k) \le 1/(n\varepsilon_M^2)$ и позволяют решать существенно более широкий класс задач.

Все вышеизложенное подтверждается экспериментально. Например, с помощью алгоритма RELAX при вычислениях с обычной точностью нельзя решить задачу минимизации функции F_7, приведенной в разд. 1.4.6. Для F_7 имеем $\eta=10^{12} > 1/(n\varepsilon_M) \approx 2{,}5\cdot10^6$, где $n = 4$, $\varepsilon_M \approx 10^{-7}$, т.е. неравенство (1.5.51) нарушено.

Как показывает практика, наибольшую эффективность методы типа RELAX имеют при решении задач с удовлетворяющим неравенству (1.5.51) значением η при кусочно-квадратичном характере зависимости J(x). Характер выпуклости J(x) при этом безразличен.

При минимизации функционалов типа (1.4.8), (1.4.9) целесообразно

вычислять производные по формулам (1.4.10), (1.4.11), (1.4.13), (1.4.14) так же, как и в методах ОПС. Для тестовых функций, приведенных в разд. 1.4.6, этим методом были получены следующие результаты:

F	$F_1 \approx 1{,}1 \cdot 10^{-12}$	$F_2 \approx 7{,}7 \cdot 10^{-5}$	$F_4 \approx 2{,}5 \cdot 10^{-4}$	$F_5 \approx 8{,}2 \cdot 10^{-16}$	$F_6 \approx 319{,}7$
g	25	280	150	19	160

Таким образом, в отличие от методов ОПС матричные градиентные схемы типа RELAX оказываются менее универсальными. Однако там, где они применимы, может быть получен заметный вычислительный эффект. Кроме того, как показано ниже, на базе градиентных методов могут быть построены алгоритмы оптимизации с большим числом n управляемых параметров. Следовательно, рассматриваемые классы методов взаимно дополняют друг друга, не позволяя выделить какой-то один наилучший подход.

1.5.5. Методы оптимизации больших систем.

Под большими системами будем понимать системы, описываемые моделями с большим числом управляемых параметров. Если степень овражности соответствующих критериев оптимальности достаточно высока, то стандартные вычислительные средства оказываются неэффективными в силу изложенных ранее причин. Методы ОПС, а также методы ЭР неприменимы, так как их вычислительные схемы содержат заполненные матрицы размерности $n \times n$, что при больших (около 1000) n определяет чрезмерные требования к объему необходимой памяти компьютера. Это же замечание справедливо для квазиньютоновских алгоритмов типа метода Давидона – Флетчера – Пауэлла, Бройдена, Пирсона, Мак-Кормика, Бройдена – Флетчера – Гольдфарба – Шенно,

Пауэлла – Бройдена и т.д.

Наиболее часто в указанной ситуации рекомендуется применять различные нематричные формы метода сопряженных градиентов (СГ). Однако, как показано далее, их возможности также весьма ограничены. Это вызвано тем, что доступное число итераций оказывается меньше размерности и в результате гарантируется сходимость со скоростью геометрической прогрессии с показателем, близким к единице:

$$||x^k - x^*|| \le 2t^k ||x^0 - x^*||, \qquad (1.5.52)$$

где $x^* = \operatorname{argmin} f(x)$, $x \in R^n$, $t \approx (1 - 2/\sqrt{\eta})$; η – коэффициент овражности минимизируемого сильно выпуклого квадратичного функционала[2]. Таким образом, конечность метода СГ при решении задач минимизации квадратичных функционалов в этом случае не играет роли.

Далее показано, что в рамках класса матричных градиентных схем (1.5.2) могут быть построены алгоритмы, более эффективные для рассматриваемых задач, чем методы СГ.

Пусть оптимизируемая большая система может быть представлена как совокупность q взаимосвязанных подсистем меньшей размерности. Пусть также требования к выходным параметрам системы могут быть сформулированы в виде неравенств

$$y_j(x^j, x^q) \le t_j \quad (j = \overline{1, q-1}); \quad y_q(x^q) \le t_q, \qquad (1.5.53)$$

где x^j – n_j-мерный частный вектор управляемых параметров; x^q – n_q-мерный вектор управляемых параметров, влияющий на все q выходных параметров и осуществляющий связь отдельных подсистем оптимизируемой системы. Размерность полного вектора управляемых параметров $x = (x^1, x^2, \dots, x^q)$ равна

[2] В литературе оценка (11.5.1) часто приводится с ошибкой в множителе.

$$n = \sum_{i=1}^{q} n_i . \qquad (1.5.54)$$

Используя технику построения целевых функционалов, представленную в разд. 1.2.6, можно привести задачу решения системы неравенств (1.5.53) к виду

$$J(x) = \sum_{j=1}^{q} \phi_j(x^j, x^q) \to \min_{x \in R^n}, \qquad (1.5.55)$$

где критерий (1.5.55) является сглаженным вариантом минимаксного критерия.

Функционалы (1.5.55) возникают и при других постановках задач оптимизации, не основанных непосредственно на минимаксных критериях. Поэтому задача (1.5.55) имеет достаточно общий характер.

Далее будут рассмотрены методы решения задачи (1.5.55) при следующих предположениях.

1) Критерий J(x) обладает относительно высокой степенью овражности, а его выпуклость гарантируется только в окрестности точки минимума.
2) Размерность (1.5.54) полного вектора управляемых параметров x велика, что, с одной стороны, затрудняет применение стандартных методов оптимизации из-за ограниченной емкости доступной памяти компьютера, а с другой – не позволяет реализовать предельно возможные характеристики сходимости алгоритмов.
3) Решение задачи анализа оптимизируемой системы требует значительных вычислительных затрат. Поэтому в процессе оптимизации требуется минимизировать число обращений к вычислению значений J(x).

4) Коэффициент заполнения γ матрицы $A(x) = J''(x)$ достаточно мал. Обычно можно считать $\gamma \sim 1/q$.

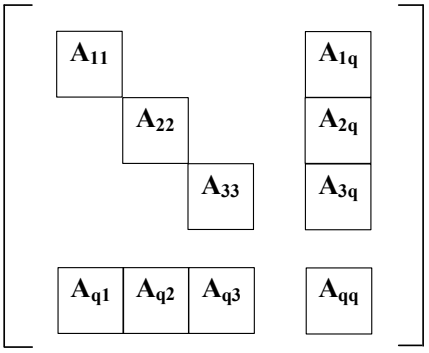
$\mathbf{A(x)} =$ 

рис. 1.5.6

При сделанных предположениях структура матрицы A(x) не зависит от точки x (рис. 1.5.6). Подматрица A_{ij} имеет размеры $n_i \times n_j$, а общее число ненулевых элементов равно

$$\sum_{i=1}^{q} n_i^2 + 2n_q \sum_{i=1}^{q-1} n_i .$$

Таким образом, учитывая симметричность матрицы A(x), в памяти компьютера необходимо хранить

$$\sum_{i=1}^{q} \frac{n_i^2 + n_i}{2} + n_q \sum_{i=1}^{q-1} n_i$$

ненулевых элементов. Необходимые сведения о схемах хранения разреженных матриц широко представлены в литературе.

Из изложенных в предыдущих разделах методов только методы Ньютона и Левенберга могут рассматриваться при оптимизации больших систем с достаточно высокими показателями овражности η. Однако неприменимость метода Ньютона в невыпуклой ситуации и отмеченные в разд. 1.5.2 недостатки метода Левенберга не позволяют считать вопрос решенным.

Обратимся снова к классу матричных градиентных схем (1.5.2).

В силу приведенных выше предположений и сформулированных в разд. 1.5.1 требований к функциям релаксации наиболее рациональный метод должен иметь функцию релаксации, значения которой резко снижаются от R = 1 при λ=0, оставаясь малыми во всем диапазоне [0, M]. И напротив, при $\lambda < 0$ функция $R(\lambda)$ должна интенсивно возрастать. Кроме

того, отвечающая $R(\lambda)$ матричная функция H должна строиться без матричных умножений для сохранения свойства разреженности матрицы $A_k = J''(x^k)$.

Покажем, что в качестве такой $R(\lambda)$ с точностью до множителя могут быть использованы смещенные полиномы Чебышева второго рода $P_s(\lambda)$, удовлетворяющие следующим соотношениям:

$$P_1(\lambda) = 1; \quad P_2(\lambda) = 2(1 - 2\lambda); \quad P_{s+1}(\lambda) = 2(1 - 2\lambda)\, P_s(\lambda) - P_{s-1}(\lambda). \qquad (1.5.56)$$

Графики зависимостей $P_s(\lambda)/s$ для $s = 3, 4, 5$ представлены на рис. 1.5.7.

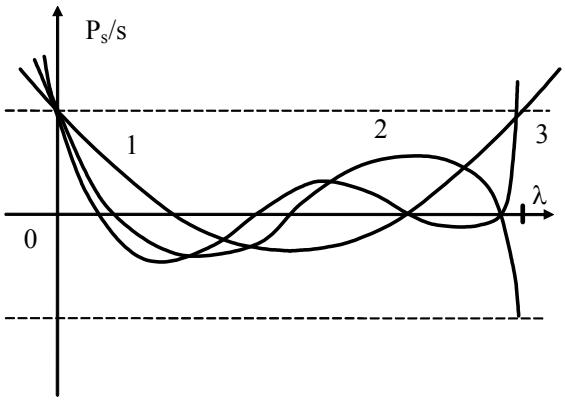


рис. 1.5.7

Действительно, полагая $R(\lambda) = P_L(\lambda)/L$ при достаточно большом значении L, получим сколь угодно быструю релаксацию любого слагаемого в представлении (см. разд. 1.5.1)

$$f(x^{k+1}) = 1/2 \sum_{i=1}^{n} \xi_{i,k}^2 \lambda_i R^2(\lambda_i), \qquad (1.5.57)$$

где

$$x^k = \sum_{i=1}^{n} \xi_{i,k} u_{i.}$$

Это утверждение вытекает из известного факта равномерной сходимости последовательности $\{P_s(\lambda)/s\}$ к нулю при $s \to \infty$ на промежутке (0; 1). Далее будем предполагать, что собственные числа матрицы A_k нормированы к промежутку (0; 1). Для этого достаточно вместо матрицы J'' рассматривать матрицу $J''/\|J''\|$, а вместо вектора J' – вектор $J'/\|J''\|$.

Отвечающая принятой $R(\lambda)$ зависимость $H(\lambda)$ имеет вид

$$H(\lambda) = [1 - R(\lambda)] / \lambda = [1 - P_L(\lambda) / L] / \lambda. \qquad (1.5.58)$$

Построение методов (1.5.2) непосредственно с функцией (1.5.58) возможно, но приводит, как и в методе Левенберга, к необходимости решения на каждом шаге по k больших линейных систем уравнений с разреженной матрицей. Ниже показано, что существуют более эффективные приемы реализации.

Действительно, из выражений (1.5.58) следует, что $H(\lambda)$ является полиномом степени $L - 2$, в то время как $R(\lambda)$ имеет степень $L - 1$. Поэтому для реализации матричного градиентного метода с указанной функцией $H(\lambda)$ нет необходимости решать линейные системы. Метод будет выглядеть следующим образом:

$$x^{k+1} = x^k - (\alpha_1 E + \alpha_2 A_k + \ldots + \alpha_{L-1} A_k^{L-2})\, J'(x^k) =$$

$$= x^k - H(A_k) J'(x^k), \quad A_k = J''(x^k). \qquad (1.5.59)$$

Реализация метода (1.5.59) может быть основана на методах вычисления коэффициентов α_i для различных степеней L. При этом число L должно выбираться из условия наиболее быстрого убывания $J(x)$. Альтернативный, более предпочтительный подход основан на других соображениях.

Для функции

$$H_s(\lambda) = \alpha_1 + \alpha_2 \lambda + \ldots + \alpha_{s-1} \lambda^{s-2}, \quad s = 2, 3,$$

из (1.5.56) можно получить рекуррентное соотношение

$$(s+1) H_{s+1} = 2s(1 - 2\lambda) H_s - (s-1) H_{s-1} + 4s, \qquad (1.5.60)$$

$$(H_1 = 0, \; H_2 = 2, \; s = \overline{2, L-1}).$$

Из соотношения (1.5.60) имеем

$$x^{k+1}[s+1] = x^k - H_{s+1} J'_k = x^k - \frac{2s}{s+1}(E - 2A_k) \cdot H_s J'_k + \frac{s-1}{s+1} H_{s-1} J'_k - \frac{4s}{s+1} J'_k$$

$$(s = \overline{2, L-1})$$

или

$$\vartheta_{s+1} = x^{k+1}[s+1] - x^k = \frac{2s}{s+1}(E - 2A_k)\vartheta_s - \frac{s-1}{s+1}\vartheta_{s-1} - \frac{4s}{s+1}J'_k$$

$$(\vartheta_1 = 0, \vartheta_2 = -2J'_k, s = \overline{2, L-1}). \qquad (1.5.61)$$

Здесь $x^{k+1}[s]$ есть s-е приближение к вектору $x^{k+1} = x^{k+1}[L]$.

Таким образом, при фиксированной квадратичной аппроксимации f(x) функционала J(x) в окрестности $x = x^k$ мы имеем возможность переходить от P_s к P_{s+1} в результате одного умножения матрицы $E - 2A_k$ на вектор ϑ_s, в полной мере используя свойство разреженности матрицы $A_k = J''_k$ и не прибегая к дополнительным вычислениям градиента. Эффективность алгоритма (1.5.61) при больших значениях η определяется множителями релаксации для малых собственных значений матрицы A_k. Рассмотрим положительную часть спектра ($\lambda > 0$), что особенно важно в окрестности оптимума, где матрица $J''(x)$ положительно определена. Основное достоинство метода с функцией релаксации вида $R_s(\lambda) = P_s(\lambda) / s$ состоит в том, что уже при малых s происходит заметное подавление слагаемых из (1.5.57) в широком диапазоне значений λ. Ниже представлены значения R_s для внутреннего максимума $R_s(\lambda)$ и границы диапазонов $\alpha_s \le \lambda \le \beta s$, где $|R_s(\lambda)| \le R_s$:

Таблица 1.5.1

s	3	4	5	6	7	8
R_s	0,333	0,272	0,250	0,239	0,233	0,230
α_s	0,147	0,092	0,061	0,044	0,033	0,025
β_s	0,853	0,908	0,939	0,956	0,967	0,975
$-R'_s(0)$	5,30	10,0	16,0	23,3	32,0	42,0

Можно показать, что значения α_s, β_s для $s > 8$ могут быть вычислены

по асимптотической формуле

$$\alpha_s=1{,}63/s^2; \quad \beta_s=1-\alpha_s; \qquad (1.5.62)$$

при этом $R_s<0{,}23$. В левой части спектра ($\lambda<0$) имеем $R_s(\lambda) > 1+R'_s(0)\lambda$, поэтому значения производных $R'_s(0)$ характеризуют множители релаксации для отрицательных слагаемых в (1.5.57).

Упрощенная схема алгоритма, построенного на основе соотношения (1.5.61), может быть реализована с помощью следующей последовательности шагов.

Алгоритм RELCH.

Шаг 1. Задать начальную точку x; вычислить J:= J(x); задать L.

Шаг 2. Вычислить J' := J'(x), J'':= J''(x); принять J' := J' / ||J''||, J'':= J'' / ||J''||, α := 1.

Шаг 3. По формуле (11.5.10) построить ϑ_L; принять $x^t := x + \vartheta_L$.

Шаг 4. Вычислить $J_t := J(x^t)$. Если $J_t > J$, перейти к шагу 5; иначе – к шагу 6.

Шаг 5. Принять $\alpha := \alpha/2$, $x^t := x + \alpha\vartheta_L$ и перейти к шагу 4.

Шаг 6. Принять $x := x^t$, $J := J_t$ и перейти к шагу 2.

Критерий окончания процесса здесь не указан. Как правило, вычисления заканчиваются по исчерпании заданного числа вычислений функционала либо при явной остановке алгоритма. Число пересчетов L по формуле (1.5.61) является параметром, задаваемым пользователем. Согласно (1.5.62) первоначально целесообразно принять $L \approx 1{,}3\sqrt{\eta}$, $\eta \approx 1/\alpha_L$, где η – оценка степени овражности минимизируемого функционала. При таком выборе L множители релаксации в положительной части спектра будут гарантированно меньше 0,23. При конструировании алгоритмических способов задания L необходимо учитывать, что последовательность $\{J_s\}$, где $J_s = J(x^k + \vartheta_s)$ не будет при

$s \to \infty$ убывать монотонно. На шаге 5 алгоритма применена регулировка нормы вектора продвижения с целью предотвращения выхода из области справедливости локальной квадратичной модели функционала.

Дадим оценку эффективности метода (1.5.61) по сравнению с методом сопряженных градиентов (СГ), наиболее конкурентоспособным из стандартных методов решения больших задач оптимизации.

Важная особенность алгоритмов типа RELCH заключается в том, что соответствующие множители релаксации будут определяться только числом итераций L и степенью η обусловленности задачи независимо от размерности n. В то же время в схемах методов СГ для завершения каждого цикла спуска требуется порядка n итераций; в противном случае согласно (1.5.52) скорость сходимости может быть очень малой. Кроме того, каждая итерация метода СГ даже для квадратичной функции требует нового вычисления градиента, т.е. дополнительных вычислительных затрат по анализам функционирования оптимизируемой системы.

Будем далее полагать, что алгоритм RELCH реализован с постоянным $L = \sqrt{\eta}$, имея в области $\lambda > 0$ множители релаксации, не превышающие значения 0,23.

Рассмотрим задачу минимизации квадратичного функционала $f(x)=1/2\langle Ax, x\rangle$ с положительно определенной матрицей A. Оценим количество вычислений f(x), требуемое для достижения контрольного вектора x' с нормой $||x'|| \le 0{,}23$, методом СГ и алгоритмом RELCH из начальной точки x^0 с $||x_0|| = 1$. По достижении точки x' вся ситуация повторяется, поэтому полученные ниже сравнительные оценки эффективности имеют достаточно общий характер.

Будем предполагать также, что вместо экономичных формул (1.4.11), (1.4.14) для вычисления производных применяются двусторонние конечно-

разностные соотношения (1.4.15), (1.4.16).

Для достижения вектора x' по алгоритму RELCH требуется вычислить в точке x^0 слабо заполненную матрицу Гессе с заполненной главной диагональю и вектор градиента $f'(x^0)$. При коэффициенте заполнения γ для этого потребуется около $2\gamma n^2$ вычислений f. Далее выполняется $L = 1{,}3\sqrt{n}$ итераций по формуле (1.5.61), не требующих дополнительных анализов функционирования.

Чтобы получить вектор x' по методу СГ, потребуется N итераций, где число N определяется из условия (1.5.52): $||x^N|| = 2t^N = 0{,}23$, т.е. $N \approx -2{,}2 / \ln t$. Для выполнения каждой итерации необходимо обновление вектора градиента, что связано с 2n вычислениями f(x). Общее число вычислений f равно $-4{,}4n / \ln t$. Относительный выигрыш в количестве вычислений f методом RELCH по сравнению с методом СГ задается функцией $\psi(\eta) \approx -2{,}2/(\gamma n \ln t)$. Очевидно, при $\eta \to \infty$ имеем $t(\eta) \to 1$, $\psi(\eta) \to \infty$. Характерные значения ψ для $\gamma = 0{,}01$ и n= 1000 даны ниже:

Таблица 1.5.2

η	100	1000	1500	10^4	10^5
ψ	1,0	3,4	4,0	11,0	35,0

Таким образом, для получения сравнимых результатов при $\eta = 10^4$ по алгоритму RELCH потребуется приблизительно в 11 раз меньше вычислений f, чем по методу СГ. Следует однако учитывать, что при увеличении η возрастает число L пересчетов по формуле (1.5.61). Это может приводить к возрастанию вычислительных погрешностей при вычислении ϑ_s с большими номерами s.

Важным дополнительным преимуществом алгоритма RELCH по сравнению с методом СГ является его достаточно высокая эффективность при решении задач с невыпуклыми функционалами, так как функция

релаксации метода в левой полуплоскости целиком расположена в разрешенной области и множители релаксации для $\lambda > 0$ быстро растут по абсолютному значению при переходе от ϑ_s к ϑ_{s+1}. Характеристики роста были приведены ранее.

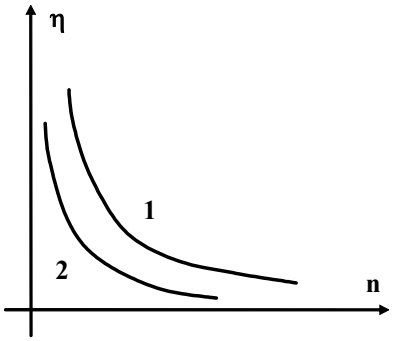


рис. 1.5.9

Так же как в методе ЭР, можно показать, что эффективность рассматриваемого подхода сохраняется при степенях овражности, удовлетворяющих неравенству $\eta < 1/(n\varepsilon_M)$. Области работоспособности алгоритмов RELAX, RELCH в плоскости n, η представлены на рис. 1.5.9. Ясно, однако, что при малых размерностях n более эффективными, вообще говоря, оказываются алгоритмы типа RELAX. Они позволяют за меньшее число N_y операций умножения матрицы на вектор получать заданные значения множителей релаксации. При больших η это приводит к существенному уменьшению накопленной вычислительной погрешности.

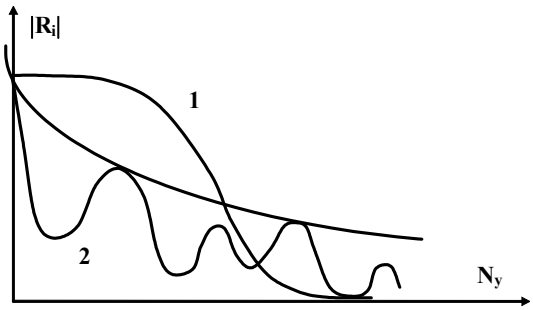


рис. 1.5.8

Для подтверждения данного замечания достаточно проанализировать характер изменения множителей релаксации при применении формул пересчета (1.5.37) и (1.5.61). Характерные зависимости для рассмотренных случаев (для фиксированного $\lambda_i>0$) представлены на рис. 1.5.8. Из рис. 1.5.10 видно в то же время, что если область локальной квадратичности функционала J(x) невелика (ζ_k мало), то $|R_i| \approx 1$ и более эффективными могут оказаться методы типа RELCH.

2. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кини Р.Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения/Под ред. И.Ф. Шахнова. – М.: Радио и связь, 1981.
2. Вентцель Е.С. Исследование операций. – М.: Советское радио, 1972.
3. Борисов А.Н., Вилюмс Э.Р., Сукур Л.Я. Диалоговые системы принятия решений на базе МИНИ-ЭВМ: Информационное, математическое и программное обеспечение. – Рига: Зинатне, 1986.
4. Вязгин В.А., Федоров В.В. Математические методы автоматизированного проектирования: Учеб. пособие для втузов. – М.: Высш. шк., 1989.
5. Моисеев Н.Н. Математические задачи системного анализа. – М.: Наука, 1981.
6. Растригин Л.А. Современные принципы управления сложными объектами. – М.: Сов. радио, 1980.
7. Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г. Численные методы решения жестких систем. – М.: Наука, 1979.
8. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. – М.: Высш. шк., 1989.
9. Подиновский В.В., Ногин В.Д. Парето–оптимальные решения многокритериальных задач. – М.: Наука, 1982.
10. Экспертные системы. Принципы работы и примеры/ А. Брукинг, П.Джонс, Ф. Кокс и др.; Под. ред. Р. Форсайта. – М.: Радио и связь, 1987.

11. Розен В.В. Цель – оптимальность – решение. – М.: Радио и связь, 1982.
12. Черноруцкий И.Г. Методы принятия решений: Учеб. пособие. – Л.: Изд-во Ленингр. Политехн. Ин-та, 1990.
13. Федоренко Р.П. Приближенное решение задач оптимального управления. – М.: Наука, 1978.
14. Табак Д., Куо Б. Оптимальное управление и математическое программирование. – М.: Наука, 1975.
15. Таха Х. Введение в исследование операций: В 2-х книгах. Кн. 2. – М.: Мир, 1985.
16. Черноруцкий И.Г. Оптимальный параметрический синтез: электротехнические устройства и системы. – Л.: Энергоатомиздат, 1987.
17. Льюс Р.Д., Райфа Х. Игры и решения. М.: Изд-во иностр. лит-ры, 1961.
18. Ногин В.Д., Чистяков С.В. Применение линейной алгебры в принятии решений: Учеб. пособие. - СПб.: Изд-во СПбГТУ, 1998.
19. Поспелов Г.С. Искусственный интеллект – основа новой информационной технологии. – М.: Наука, 1988.
20. Черноруцкий И.Г. Методы оптимизации: Учеб. пособие. – СПб., Изд-во СПбГТУ, 1998.
21. Миркин Б.Г. Проблема группового выбора. М.: Наука, 1974.
22. Нейлор К. Как построить свою экспертную систему. М.: Энергоатомиздат, 1991.
23. Дэннис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений. – М.: Мир, 1988.

24. Дубов Ю.А., Травкин С.И., Якимец В.Н. Многокритериальные модели формирования и выбора вариантов систем. – М.: Наука, 1986.
25. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987.
26. Построение экспертных систем: Пер. с англ./ Под ред. Ф. Хейеса-Рота, Д. Уотермана, Д. Лената. – М.: Мир, 1987.
27. Хованов Н.В. Анализ и синтез показателей при информационном дефиците. – СПб.: Изд-во СпбГУ, 1996.
28. Глухов В.В., Медников М.Д., Коробко С.Б. Математические методы и модели для менеджмента. СПб.: Изд-во «Лань», 2000.
29. Змитрович А.И. Интеллектуальные информационные системы. Мн: НТООО «ТетраСистемс», 1997.
30. Малыхин В.И. Финансовая математика: Учеб. пособие для вузов. – М.: ЮНИТИ-ДАНА, 2000.
31. Уотшем Т.Дж., Паррамоу К. Количественные методы в финансах: Учеб. пособие для вузов. – М.: Финансы, ЮНИТИ, 1999.
32. Интеллектуальные системы принятия проектных решений/А.В.Алексеев, А.Н.Борисов, Э.Р.Вилюмс, Н.Н.Слядзь, С.А.Фомин. – Рига: Зинатне, 1997.
33. Шрейдер Ю.А. Равенство, сходство, порядок. – М.: Наука, 1971.
34. Химмельблау Д. Прикладное нелинейное программирование. – М.: Мир, 1975.
35. Подиновский В.В. Многокритериальные задачи с упорядоченными по важности однородными критериями//Автоматика и Телемеханика, 1976. - № 11. – С. 118 – 127.