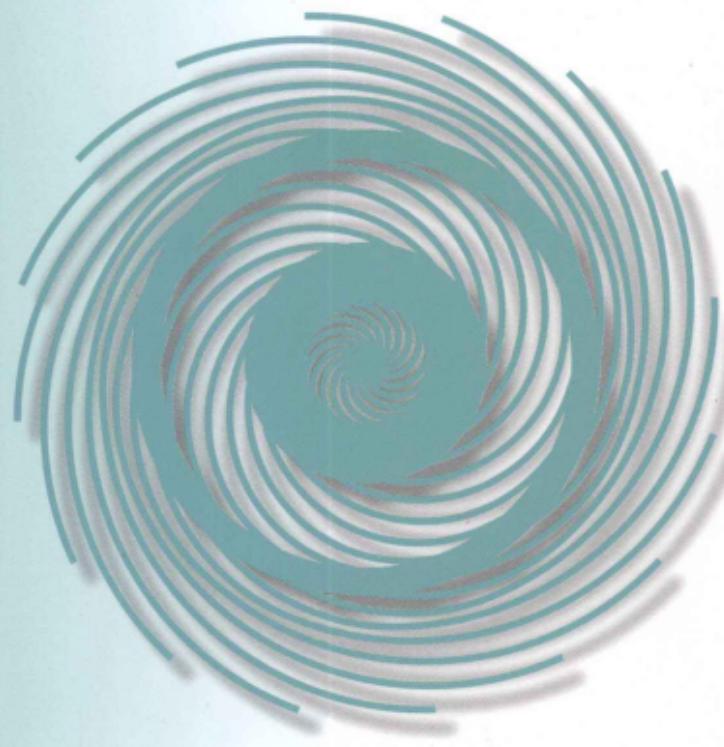


刘红英 夏 勇 周水生 编



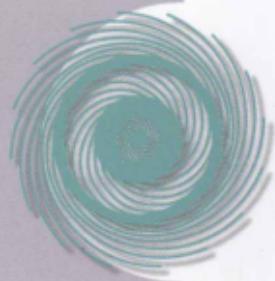
高等学校研究生教材

数学规划基础



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

GAODENG XUEXIAO
YANJIUSHENG
JIAOCAI



上架建议：数 学

ISBN 978-7-5124-0912-5



9 787512 409125 >

策划编辑：白 航

定价：39.00元

高等学

0221/25

数学规划基础

刘红英 夏 勇 周水生 编

北京航空航天大学出版社

内 容 简 介

本书以数学规划中最基本的问题为对象,从理论、算法和计算三方面介绍了线性规划、无约束非线性规划和约束非线性规划等优化问题。其中,线性规划主要包括基本理论、单纯形法、网络流问题和整数线性规划等;无约束非线性规划主要包括一维搜索、最速下降法和牛顿法、共轭梯度法和拟牛顿法及其在最小二乘问题中的应用;约束非线性规划主要包括最优化条件、积极集法、罚函数法、逐步二次规划法和内点法等。

本书可作为应用数学、计算数学、运筹学与控制论、管理科学与工程、工业工程、系统工程、信息工程及计算机科学等专业的研究生和高年级本科生的教材,也可以作为其他需要利用数学规划方法进行建模和求解实际问题的各学科领域的科研人员、工程技术人员的参考书。

图书在版编目(CIP)数据

数学规划基础 / 刘红英, 夏勇, 周水生编. -- 北京

北京航空航天大学出版社, 2012. 10

ISBN 978 - 7 - 5124 - 0912 - 5

I. ①数… II. ①刘… ②夏… ③周… III. ①数学规划—高等学校—教材 IV. ①O221

中国版本图书馆 CIP 数据核字(2012)第 191269 号

版权所有, 侵权必究。

数学规划基础

刘红英 夏 勇 周水生 编

责任编辑 徐金凤 罗小霞 杨淑媚

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: bhpress@263.net 邮购电话:(010)82316936

北京时代华都印刷有限公司印装 各地书店经销

*

开本: 787×1 092 1/16 印张: 18.25 字数: 467 千字

2012 年 10 月第 1 版 2012 年 10 月第 1 次印刷 印数: 3 000 册

ISBN 978 - 7 - 5124 - 0912 - 5 定价: 39.00 元

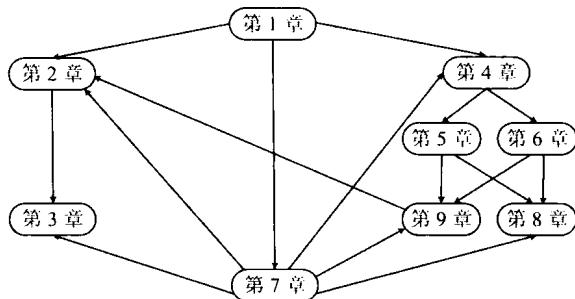
若本书有倒页、脱页、缺页等印装质量问题, 请与本社发行部联系调换。联系电话:(010)82317024

前　　言

数学规划是计算数学与运筹学的交叉学科,广泛应用于经济、金融、工程和管理等许多领域,如模式识别中的统计方法、资源分配中的效用最大化和调度问题、数据拟合中出现的各种优化问题等都可划归为数学规划范畴.

本书是为数学系高年级本科生和工科研究生的“最优化方法”课程而编写的教材. 内容上以数学规划中最基本的问题为对象, 从理论、算法和计算三方面介绍了求解线性规划、无约束非线性规划和约束非线性规划等问题. 在编写时既考虑到与本科生最基本的数学知识相衔接, 也考虑到夯实那些将来准备投身科学的研究和工程计算的学生的理论基础.

本教材在注重理论基础的同时, 尽可能给出各种理论的几何直观描述和方法的动机, 辅以典型应用和例子, 并配备层次分明、内容丰富的习题; 为了加强学生对应用和前沿性研究成果的了解, 本书在相关章节介绍了如网络流问题及其应用、半定规划和路径跟踪算法等相关内容. 本书各章节内容之间的关系如右图所示, 读者可以根据需要选读. 值得一提的是, 对关注最优化理论与方法的读者而言, 无论怎样重视第7章的最优化条件都不为过.



作者想借此机会对我们的同事, 北京航空航天大学的陆启韶教授和高宗升教授表示衷心的感谢, 该书得以出版与他们的鼓励与帮助是分不开的.

同时, 刘红英和周水生也想借此机会对自己的研究生导师——西安电子科技大学的刘三阳教授表示衷心的感谢, 因为今天取得的任何点滴成绩都源自上学时老师的悉心指导和谆谆教诲.

北京航空航天大学出版社的责任编辑为本书的出版付出了辛勤的劳动, 本书也得到了北京航空航天大学研究生精品课项目的资助, 在此表示感谢.

由于编者的水平有限, 经验不足, 缺点和疏漏在所难免, 敬请批评指正. 任何意见、建议或其他反馈都可以发送至 liuhongying@buaa.edu.cn, 在此深表感谢.

刘红英　夏勇　周水生
2012年6月于北京

目 录

第1章 引言	1
1.1 数学描述与例子	1
1.2 优化问题的分类	3
1.3 优化算法	5
1.4 数学基础	6
1.5 评注和参考	9
习题1	9
第2章 线性规划：基本理论与方法	11
2.1 基本性质	11
2.1.1 标准形	13
2.1.2 基本可行解	15
2.1.3 基本定理	16
2.1.4 几何直观	17
2.2 单纯形法	19
2.2.1 既约费用系数	20
2.2.2 基本可行解的改进	21
2.2.3 计算过程	22
2.2.4 退化与循环	25
2.2.5 初始基本可行解	27
2.2.6 修正单纯形法	29
2.2.7 单纯形法的效率	33
2.3 对偶	34
2.3.1 对偶问题	35
2.3.2 对偶定理	36
2.3.3 对偶问题与单纯形法的关系	37
2.3.4 灵敏度与互补	40
2.3.5 对偶单纯形法	41
2.4 评注与参考	44
习题2	44
第3章 线性规划：扩展及其应用	54
3.1 网络单纯形法	54
3.1.1 问题的表述	54
3.1.2 生成树与基	55

3.1.3 网络单纯形法	57
3.2 最小费用流问题的应用	60
3.2.1 运输问题和指派问题	60
3.2.2 最大流问题	62
3.2.3 最短路问题	64
3.3 整数线性规划	66
3.3.1 简介	66
3.3.2 对偶理论	69
3.4 整数规划的典型方法	70
3.4.1 Gomory 割平面法	71
3.4.2 分枝定界法	73
3.5 评注与参考	77
习题 3	78
第 4 章 无约束优化: 基础	80
4.1 极小点的条件	80
4.1.1 局部极小点的条件	80
4.1.2 凸性与全局极小点	82
4.2 算法概述	84
4.2.1 概述	84
4.2.2 线搜索法	86
4.3 非精确线搜索	87
4.3.1 一维搜索的终止准则	88
4.3.2 下降方法的稳定性	90
4.4 线搜索子问题的算法	92
4.5 评注与参考	98
习题 4	98
第 5 章 无约束优化: 线搜索法	100
5.1 基本方法	100
5.1.1 最速下降法	100
5.1.2 牛顿法	103
5.2 共轭梯度法	106
5.2.1 扩展子空间定理	106
5.2.2 基本的共轭梯度法	107
5.2.3 收敛速度与预条件	113
5.3 拟牛顿法	116
5.3.1 拟牛顿条件	116
5.3.2 DFP 法和 BFGS 法	117
5.3.3 DFP 法和 BFGS 法的性质	120

5.3.4 SR1 法	122
5.4 最小二乘	124
5.4.1 线性最小二乘	124
5.4.2 非线性最小二乘	125
5.5 评注与参考	128
习题 5	130
第 6 章 无约束优化:信赖域法	136
6.1 原型算法	136
6.2 信赖域子问题	140
6.2.1 解的刻画	140
6.2.2 求解子问题的牛顿法	142
6.3 求解子问题的近似方法	145
6.3.1 柯西点	145
6.3.2 Dog-leg 法	146
6.3.3 Steihaug 共轭梯度法	147
6.4 实用信赖域法	149
6.5 评注与参考	150
习题 6	150
第 7 章 约束优化:理论	153
7.1 概述	153
7.2 Lagrange 乘子	155
7.3 一阶条件	160
7.4 二阶条件	164
7.5 凸规划	167
7.6 凸规划和 Lagrange 乘子	168
7.7 对偶	171
7.8 半定规划	174
7.8.1 半定规划的对偶理论	175
7.8.2 最大割问题的 0.878 近似算法	177
7.8.3 半定规划的其他应用	179
7.9 评注与参考	181
习题 7	182
第 8 章 约束优化:线性约束规划	186
8.1 等式约束二次规划	186
8.2 积极集法	191
8.3 线性等式约束规划	194
8.4 线性不等式约束规划	197
8.5 锯齿现象	199

8.6 评注与参考	201
习题 8	202
第 9 章 约束优化: 非线性约束规划	205
9.1 惩罚和障碍函数	205
9.1.1 Courant 罚函数	206
9.1.2 障碍函数	211
9.2 乘子罚函数	212
9.3 ℓ_1 精确罚函数	218
9.4 逐步二次规划法	223
9.4.1 Lagrange-Newton 法	223
9.4.2 基本逐步二次规划法	223
9.4.3 价值函数	227
9.4.4 实用逐步二次规划法	230
9.5 线性规划的路径跟踪算法	234
9.5.1 障碍函数子问题和中心路径	234
9.5.2 用牛顿法求解障碍函数子问题	235
9.5.3 理论分析	236
9.6 评注与参考	239
习题 9	239
附录 A 基础知识	242
A.1 集合	242
A.2 矩阵	242
A.3 空间	243
A.4 特征值与二次型	245
A.5 拓扑概念	246
A.6 函数	247
A.7 矩阵分解	250
A.7.1 高斯消元法与 LU 分解	250
A.7.2 Cholesky 分解	253
A.7.3 QR 分解	254
A.7.4 奇异值分解	255
A.8 其他	255
A.8.1 标量方程求根	255
A.8.2 误差分析和浮点计算	256
A.8.3 条件数和稳定性	257
附录 B 阅读材料	259
B.1 KKT 条件和对偶理论的应用实例	259
B.1.1 KKT 条件的力学解释	259

B. 1.2 KKT 条件的应用实例	260
B. 1.3 对偶理论的应用实例	262
B. 2 MAX-2-SAT 问题的半定规划松弛	263
参考文献	266
索 引	269
一 画	269
二 画	269
三 画	270
四 画	270
五 画	271
六 画	272
七 画	274
八 画	274
九 画	276
十 画	277
十一画	278
十二画	278
十三画	279
十四画	279
十五画	279
十六画	279
其 他	279

第1章 引言

优化是以数学的方式来刻画和找出问题最优解的一门学科,在数学领域称为数学规划 (mathematical programming),在管理领域属于运筹学 (operational research) 范畴. 它是解决大量实际应用问题的有力工具,特别是科学决策或物理系统分析的重要工具,比如: 航空公司通过合理安排机组人员及航班来极小化成本; 投资者选择那些避免承担过多风险,同时又达到高收益的证券进行投资; 制造者选取合适的设计使得他们生产过程中的生产效率最大化; 物理系统趋向于具有最小能量的状态; 处于孤立化学系统的分子相互作用,直到系统中电子的总势能达到最小; 光线沿着用时最短的路径传播(费马原理)等.

为了解决这些问题,需要建立目标 (objective), 即待研究系统性能的某种度量. 目标可以是利润、时间、势能,或者是任何数量,也可以是多个量组合出来的某种度量,甚至可以有多个目标. 称决定目标的系统参数为变量 (variables) 或未知数 (unknowns). 优化的目的是求出使该目标达到最优的变量的值. 这些变量通常是以某种方式受限制的,或是带约束的 (constrained), 比如表示分子中电子密度的量和贷款利率的量必须是非负的.

优化过程的第一步是建立给定问题的恰当的目标、变量和约束,这一过程称为数学建模 (modeling), 这一步具有相当的统筹性和折衷的艺术性. 模型太简单,将难以给实际问题提供有用的决策支持; 模型太复杂,则很难求解,或者在要求的时间内不能求解.

建立了模型,需要设计相应的优化算法进行求解. 通常模型和算法不会太简单,一般需要借助于计算机来求解. 虽然没有通用优化算法(直观上越通用,效率越低),但存在许多针对特定类型优化问题的高效求解算法,用户可以借鉴和有选择地使用. 算法的设计常常依赖于最优性的必要条件.

应用优化算法求解模型之后,要判断算法是否完成了求解任务. 在许多情况下,可以用最优性的充分条件来检验所得到的解是否是问题的解. 用户通常不会只满足于找到问题的解,还需要应用灵敏度分析 (sensitivity analysis) 等技术改进模型,灵敏度分析揭示了最优解对模型中参数变化或误差扰动的敏感程度.

本书仅结合少数应用案例来说明一些典型的优化问题,其余内容不再涉及优化问题的应用背景. 在本书中,对优化问题进行了适当分类,并指出各类优化问题所适用的算法. 读者据此可以把自己面临的问题恰当归类,并设计相应算法求解,或者选用适当的优化软件求解.

下面介绍优化的基本模型、优化问题的各种分类、优化算法概述和部分常用的数学概念.

1.1 数学描述与例子

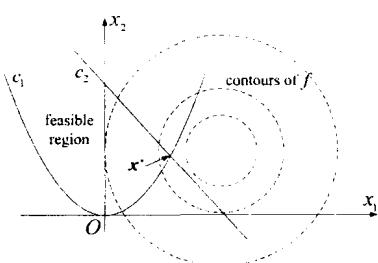
本书统一默认列向量 x 为变量, 函数 $f(x)$ 是目标函数, $c(x)$ 是变量 x 必须满足的约束向量值函数, 其分量个数即是对变量所施加的约束个数, 目标是在满足约束的前提下最小化 $f(x)$. 这样, 优化问题一般可以表述为

$$\begin{aligned}
 & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\
 & \text{subject to} \quad c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} \\
 & \quad c_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{I}
 \end{aligned} \tag{1.1.1}$$

这里 f 和 c_i 是 \mathbf{x} 的标量函数, \mathcal{E} 和 \mathcal{I} 为有限指标集. 根据 $\mathbf{x}, f(\mathbf{x})$ 和 $c(\mathbf{x})$ 的不同特性和结构, 可以对优化问题进一步细分, 这将在 1.2 节中介绍.

例 1.1.1 考虑下面的问题

$$\begin{aligned}
 & \text{minimize} \quad (x_1 - 2)^2 + (x_2 - 1)^2 \\
 & \text{subject to} \quad x_1^2 - x_2 \leq 0 \\
 & \quad x_1 + x_2 \leq 2
 \end{aligned}$$



令 $f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$, $c_1(\mathbf{x}) = x_1^2 - x_2$, $c_2(\mathbf{x}) = x_1 + x_2 - 2$, $\mathcal{I} = \{1, 2\}$, $\mathcal{E} = \emptyset$, 可将此问题写成式(1.1.1)的形式.

图 1.1.1 显示了目标函数的等高线(由函数值相同的点所构成的集合)、可行域(即所有满足约束条件的点集)和问题的解 \mathbf{x}^* .

在应用中, 我们通常需要将优化问题进行适当的转化, 才能表示成式(1.1.1)的形式. 此外, 当我们要

求 $\max f$, 而非 $\min f$ 时, 在式(1.1.1)中将 f 改为 $-f$ 即可. 许多优化软件须将问题按标准形式——式(1.1.1)输入.

例 1.1.2 (田忌赛马) “田忌赛马”的故事大家都很熟悉. 田忌是战国时期齐国的一位将军, 和齐威王赛马, 他们各自有上中下三等马, 但是齐威王每个级别的马都比田忌的强, 所以第一次比赛的时候, 上马对上马, 中马对中马, 下马对下马, 田忌零比三大败而归. 第二次比赛的时候, 田忌采用了孙膑的计策, 即用上马对中马, 中马对下马, 下马对上马, 结果二比一赢了.

若用数学建模的办法来还原孙膑的策略, 首先, 将马数字化为 1、2、3 等, 分别对应上、中、下, 田忌有如下的 3×3 的收益矩阵 (payoff matrix)

$$\mathbf{C} = (c_{ij}) = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

第 (i, j) 元素表示田忌的第 i 等马与齐威王的第 j 等马单独比赛的结果, 1 表示田忌赢, -1 表示田忌输.

孙膑的决策变量是

$$x_{ij} = \begin{cases} 1, & \text{田忌的第 } i \text{ 等马对齐威王的第 } j \text{ 等马} \\ 0, & \text{其他} \end{cases} \quad i, j = 1, 2, 3$$

比赛规则是每等马必须上场, 且只能上一场. 这样, 求解让田忌赢得最多的策略可以建模为

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^3 \sum_{j=1}^3 c_{ij} x_{ij} \\
 & \text{subject to} && \sum_{j=1}^3 x_{ij} = 1, \quad i = 1, 2, 3 \\
 & && \sum_{i=1}^3 x_{ij} = 1, \quad j = 1, 2, 3 \\
 & && x_{ij} \in \{0, 1\}, i, j = 1, 2, 3
 \end{aligned} \tag{1.1.2}$$

该问题中目标函数和约束都是线性函数,但是变量取整数,所以是整数线性规划问题. 进一步,如果把整数约束 $x_{ij} \in \{0, 1\}$ 放松为 $x_{ij} \geq 0$,便得到线性规划松弛问题. 我们将在 3.2.1 小节中介绍这个特殊结构的优化问题——指派问题,并说明求解线性规划松弛问题可以求得它的解.

例 1.1.3 (曲线拟合) 在实际应用中,许多问题需要借助实验数据来拟合曲线. 比如,图 1.1.2 给出信号在时刻 t_1, t_2, \dots, t_m 处的测量值 y_1, y_2, \dots, y_m ,可以推测该信号具有某种类型的指数衰减和振荡行为,从而选择函数 $\phi(t; \mathbf{x}) = x_1 + x_2 e^{-(x_3 - t)^2/x_4} + x_5 \cos(x_6 t)$ 作为信号的模型,这里实数 x_1, x_2, \dots, x_6 是模型的参数. 希望选择这些参数使得模型值 $\phi(t_j; \mathbf{x})$ 尽可能好地拟合观测值 y_j . 为了表述成优化问题,定义余量或残差为 $r_j(\mathbf{x}) = y_j - \phi(t_j; \mathbf{x}) (j = 1, 2, \dots, m)$,它表示模型与观测数据之间的差异. 通过求解

$$\underset{\mathbf{x} \in \mathbb{R}^6}{\text{minimize}} \quad f(\mathbf{x}) = r_1^2(\mathbf{x}) + \dots + r_m^2(\mathbf{x}) \tag{1.1.3}$$

可给出参数 \mathbf{x} 的估计值. 这是一个非线性最小二乘问题,是一种特殊的无约束优化问题,将在 5.4.2 小节中详细讨论求解这种问题的方法.

此外,如果将目标函数,如式(1.1.3)所示,改为残差绝对值之和,即

$$\underset{\mathbf{x} \in \mathbb{R}^6}{\text{minimize}} \quad f(\mathbf{x}) = |r_1(\mathbf{x})| + \dots + |r_m(\mathbf{x})| \tag{1.1.4}$$

称之为最小绝对偏差曲线拟合(least absolute deviations curve-fitting),其“稳健性”比最小二乘好,其不足之处是目标函数不再光滑;但若所有的 $r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_m(\mathbf{x})$ 都是线性函数,利用习题 2.2 的方法可以将该问题表述成一个等价的线性规划问题.

1.2 优化问题的分类

根据式(1.1.1)中的一般形式,目标和约束函数的性质(线性、非线性、是否凸),变量数目(多或少)和函数的光滑性(是否可微)等均可对优化问题进行分类.

最重要的区分是变量有约束与没有约束,据此可分为无约束优化与约束优化. 无约束(unconstrained)优化问题一方面源自许多实际应用,或者忽略某些关于变量的自然约束对最优解没有影响;另一方面,对约束优化问题,在目标中引入对约束的惩罚项,可将约束问题转化为无约束问题(详见第 9 章). 约束(constrained)优化问题源自对变量有明确约束的模型. 这些约束

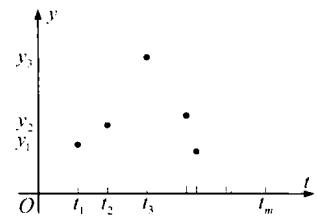


图 1.1.2 曲线拟合问题的数据

可以是诸如 $0 \leq x_1 \leq 100$ 之类的简单界约束, 或者更一般的 $\sum x_i \leq 1$ 之类的线性约束, 或者代表变量之间复杂关系的非线性不等式或等式. 当目标函数和所有约束都是 x 的线性函数时, 称为**线性**(linear)规划问题. 线性模型在管理科学和运筹学中具有广泛的应用. **非线性**(nonlinear)规划问题的约束与目标中至少有一个是非线性函数, 大部分来源于物理科学和工程应用, 更贴近实际问题, 且在管理和经济学中的应用变得越来越广泛.

连续(continuous)优化一般指变量在实向量空间的某个子集内连续取值的优化问题, 是本书的主要研究内容. 连续优化又分**光滑**(smooth)优化与**非光滑**(non-smooth)优化. 本书只考虑光滑优化, 而非光滑优化问题可以通过光滑优化技术来近似求解. 当函数光滑时, 可以利用函数在特定点 x 的信息来推断 x 邻近点的函数值.

离散(discrete)优化指从有限多个候选解中寻求最优解, 比如指派问题(1.1.2). 连续优化的可行域是无限集, 但这并不意味着连续优化一定比离散优化困难. 一方面, 对连续优化有微积分, 特别是有 **Taylor** 公式这个强有力的工具可用; 也有相对比较成熟的理论, 比如最优化条件中的必要性条件可以帮助我们排除很多非最优解, 甚至有时候(比如一些**严格凸**规划问题)只剩下唯一的候选解. 另一方面, 一些连续优化问题, 比如凹规划问题, 总是可以在可行域的顶点上取到最值, 所以可以限制为找最优顶点这个“离散”优化问题(如第2章介绍的线性规划). 对于一般的离散优化, 相邻两个可行解的距离很大, **Taylor** 近似常常失效. 因此相邻的可行解对应的函数值可能会显著不同. 此外, 可行域通常很大(比如指数多个可行解), 设计得再巧妙的穷举法(比如第3章介绍的分枝定界法)也很难求解大规模的离散优化问题. 求解离散优化一个明显的策略就是首先忽略整数要求, 当成连续变量来简化问题, 然后将所有分量舍入到最近的整数. 然而该策略一般不能保证给出的解会与最优解很近, 所得解甚至不可行, 参见例3.4.1. 还有一种模型, 一部分变量取连续值, 另一部分变量取整数值, 这样的问题为**混合整数**规划问题. 本书主要介绍连续优化, 只在第3章中简单介绍整数规划. 需要强调的是, 本书描述的连续优化算法对于离散优化很重要, 后者经常要求解一系列连续优化子问题. 例如, 求解整数线性规划的分枝定界法要花费很多时间求解“松弛的”线性规划子问题, 通常利用**对偶单纯形法**求解这些子问题, 详见3.4.2小节.

局部优化算法仅能找到一个**局部**(local)解, 即该点的目标函数值比其某个邻域中其他可行点的值小, 甚至常常连局部解也找不到. 通常很难找到所有极小点中最好的, 即**全局**(global)解. 在有些应用中必须(或者至少有很高的期望)找到全局解, 但是全局解通常不易识别、难于找到. 一种很重要的特殊情况是**凸**(convex)规划, 它的所有局部解也是全局解. 线性规划属于凸规划. 然而, 一般的非线性规划包括有约束的和无约束的, 常常属于**非凸**规划, 都可能有不是全局解的局部解. 著名的优化专家 **Rockefeller** 指出, 优化问题的**难易**与**否**不在于线性与非线性, 而在于凸与非凸. 本书的着重点是局部解的刻画和计算, 它们是优化领域研究的核心. 此外, 许多成功的全局优化算法也要求解一系列局部优化问题. 这时将用到本书描述的许多算法. 可参考 **Floudas** 和 **Pardalos** 的专著.

在一些优化问题中, 建模时会涉及一些随机的参数或者变量. 例如, 式(2.1.1)描述的运输问题中, 通常很难精确定顾客需求量 b_j . 许多经济和金融计划模型也具有该特征, 这些领域的问题经常与未来的利率和经济的发展趋势有关. 然而, 建模者经常能够以某种置信度预测或估计未知量. 例如, 他们可以对未知量的取值设定一些可能的状态, 甚至为每种状态指定概率.

在运输问题中,因为季节因素或经济条件不同,对应的需求可能会有不同的模式,零售市场的经理或许能够基于顾客以前的行为估计出需求模式.随机(stochastic)优化算法针对这些随机变量确定使优化模型达到平均性能的解.本书对随机优化不作进一步的考虑.重点考虑确定(deterministic)优化,即模型是完全确定的.许多随机优化问题可以表述成一个或多个确定子问题(比如取数学期望或者考虑鲁棒问题),用本书所描述的算法求解每个子问题.更多关于随机优化的信息,可参考 Birge 和 Louveaux、Kall 和 Wallace 的专著.

1.3 优化算法

本书叙述的方法都是迭代法,即从最优解的某个初始猜测 $x^{(0)}$ 出发,产生逐步改进的、可能收敛于解 x^* 的估计序列 $x^{(0)}, x^{(1)}, \dots$,记为 $\{x^{(k)}\}$,其中上标表示迭代次数.众多算法的主要区别在于从一个迭代点移动到下一个迭代点的策略不同.大多数策略都利用目标函数 f 和约束 c 的信息,可能还有这些函数的一阶和二阶导数信息.有些算法收集以前迭代的信息,而有的仅利用从当前点得到的局部信息.

对于一个算法,主要考虑两个方面:算法的可靠性,以及是否有足够的证据或理由表明算法能以适当快的速度收敛于问题的解.所有好的算法应该具有下面的性质:

- 稳健性(robustness).对大部分同类问题而言,只要合理选择初始点 $x^{(0)}$,算法都执行得很好.
- 有效性(efficiency).算法不用花费太多的计算时间和存储空间.
- 精确性(accuracy).算法能够得到一定精度的解,对误差(包括数据的误差或计算机执行算法时产生的算术舍入误差)不太敏感.

当然,这些目标可能是相互矛盾的.比如,一个非线性规划的快速收敛算法对于大规模问题可能需要过大的存储开销.另一方面,稳健的方法可能是最慢的.诸如收敛速度和存储要求之间的折衷、稳健性和收敛速度之间的折衷等是数值优化的中心论点.

最优化的数学理论既可用来刻画最优解,也可为大多数算法提供理论基础.没有厚实的理论支持,就不可能深刻理解数值优化.与此相应地,本书对最优化条件和收敛性分析进行了深刻的论述,其中最优化条件分局部和全局两种刻画方式,而收敛性分析揭示了一些重要算法的优点和缺点.

在上述大背景下,本书的内容编排按无约束优化问题和约束优化问题两大类来安排.其中,无约束优化问题不仅具有其自身的重要性,而且还是求解约束优化问题的重要工具,无约束优化的一些概念可以直接适用于约束优化.此外,无约束优化中还讨论了最小二乘这一重要特例.约束优化问题中,鉴于线性规划的重要性和特殊性,我们分基础与扩展两章介绍相关内容;此外,讨论了一阶和二阶最优化条件、凸性和对偶理论,以及求解约束优化的三类重要方法(罚函数法、逐步二次规划法和内点法).

本书线性规划部分的主要参考资料是 Luenberger 的 *Linear and nonlinear programming*^[17] 和 Vanderbei 的 *Linear programming: foundations and extensions*^[19];非线性规划部分主要参考 Fletcher 的名著 *Practical methods of optimization*^[14], Nocedal 和 Wright 合著的 *Numerical Optimization*^[18].本书内容侧重于那些已被应用,并且数值计算结果令人满意的实

用方法. 如果读者需要寻求一种求解自己特定问题的有效算法, 我们建议首先建立合适的优化模型, 然后根据具体问题的结构和特性(比如函数值和梯度值的计算难易程度)选择或设计针对性的算法.

1.4 数学基础

本书很多地方需要读者具备数值线性代数的基本概念和技巧, 这些内容可参阅附录 A. 本书中若不作特殊说明, 向量均用粗体小写字母表示(比如 \mathbf{a}), 矩阵用粗体大写字母表示(比如 \mathbf{B}), 即

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

本书中行向量写为 $\mathbf{b} = (b_1, b_2, \dots, b_n)$, 为了节省空间, 也将列向量写为 $\mathbf{a} = (a_1, a_2, \dots, a_n)^T$, 这里用上标 T 表示转置, 即 \mathbf{a}^T 表示行向量, 而 $\mathbf{a}^T \mathbf{z}$ 表示标量积/内积, 即 $\mathbf{a}^T \mathbf{z} = \mathbf{z}^T \mathbf{a} = \sum_{i=1}^n a_i z_i$. 除非特别指出, 向量均指列向量.

本书中用 \mathbb{R}^n 来表示 n 维向量空间, \mathbb{R}^1 略作 \mathbb{R} . \mathbb{R}^n 中的点 \mathbf{x} 用向量 $(x_1, x_2, \dots, x_n)^T$ 表示, 其中 x_1 是其在第一个坐标方向的分量, 其余依此类推. 优化技术中, 直线的概念相当重要, 若给定 \mathbf{x}' , $\mathbf{p} \in \mathbb{R}^n$, 则它是点集

$$\{\mathbf{x} (= \mathbf{x}(\alpha)) = \mathbf{x}' + \alpha \mathbf{p} : \alpha \in \mathbb{R}\} \quad (1.4.1)$$

当仅允许 $\alpha \geq 0$ 时, 得到以 \mathbf{x}' 为端点(对应于 $\alpha=0$), 以向量 \mathbf{p} 为方向的射线. 以 $(2, 2)^T$ 为端点, 以 $(3, 1)^T$ 为方向的射线见图 1.4.1, 将射线反向延伸即得对应的直线.

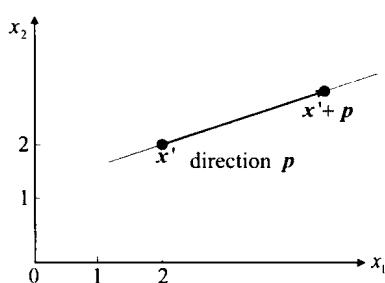


图 1.4.1 二维空间中的直线

优化方法中一个经常涉及的重要概念是多元函数(记为 $f(\mathbf{x})$)及其微分. 对于二元函数, 利用函数的等高线/等值线(contours)(比如函数取值为 γ 的等值线为集合 $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) = \gamma\}$)对函数可以进行直观的认识. 图 1.4.2 给出了著名的 Rosenbrock“香蕉”函数

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (1.4.2)$$

的等值线, 它得名于在原点的弯曲方式. 该函数之所以有名, 最主要的原因是, 用大多数优化方法寻找极小点时很慢, 从而它是检验优化算法的试金石. 它有唯一的极小点 $\mathbf{x}^* = (1, 1)^T$, 且 $f(\mathbf{x}^*) = 0$. Matlab 优化工具箱中的 Demo 给出了初始点 $\mathbf{x}^{(0)} = (-1.9, 2)^T$ 时, 用 BFGS、DFP、Steepest、Simplex、GN 以及 LM(除 Simplex 这一直接根据函数值进行优化的方法外, 其余方法后面均有介绍)求解该问题的演示实验, 各种方法的统计信息见表 1.4.1.

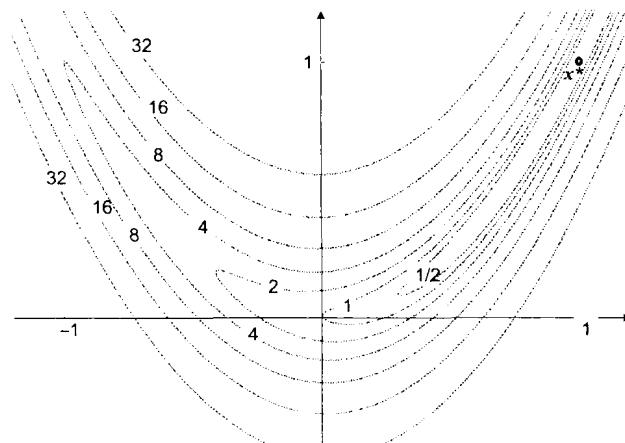


图 1.4.2 Rosenbrock 函数的等值线

表 1.4.1 各种方法用于 Rosenbrock 函数的数值结果

方法	迭代	计算函数值	方法	迭代	计算函数值
BFGS	34	45	Simplex	109	201
DFP	45	64	GN	11	48
Steepest	56	250	LM	18	82

通常假定所讨论的函数是光滑的,即连续 Fréchet-可微的(或称 C^1),这时函数在任一点存在一阶偏导数,称列向量 $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T$ 为 $f(\mathbf{x})$ 的梯度 (gradient). 这里 ∇ 表示梯度算子,经常将梯度向量记为 $\mathbf{g}(\mathbf{x})$. 如果 $f(\mathbf{x})$ 是二次连续可微的(C^2), 定义 $\nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]$ 是二阶偏导数矩阵或者 Hessian 阵(以德国数学家 Hesse 命名), 记为 $\nabla^2 f(\mathbf{x})$ 或者 $\mathbf{G}(\mathbf{x})$. 矩阵 $\nabla^2 f(\mathbf{x})$ 是对称的. 由于它的第 j 列为 $\nabla \frac{\partial f}{\partial x_j}$, 故 $\nabla^2 f(\mathbf{x})$ 也可以表示为 $\nabla(\nabla f)^T$. 例如, 对于式(1.4.2)有

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} -400x_1(x_2 - x_1^2) + 2x_1 - 2 \\ 200(x_2 - x_1^2) \end{bmatrix} \quad (1.4.3)$$

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} 1 & 200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 & 0 \end{bmatrix} \quad (1.4.4)$$

该例表明 ∇f 与 $\nabla^2 f$ 通常是 \mathbf{x} 的函数. 将点 $\mathbf{x}' = (0, 0)^T$ 代入式(1.4.3)和式(1.4.4), 有 $\mathbf{g}(\mathbf{x}') = (-2, 0)^T$, $\mathbf{G}(\mathbf{x}') = \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix}$.

利用梯度可以得出 f 沿由式(1.4.1)表示的任意直线 $\mathbf{x}(\alpha)$ 的导数. 为此, 令 $\phi(\alpha) = f(\mathbf{x}(\alpha))$, 根据链式法则

$$\frac{d\phi}{d\alpha} = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{dx_i(\alpha)}{d\alpha} = \sum_{i=1}^n p_i \frac{\partial f}{\partial x_i} = \mathbf{p}^T \nabla f(\mathbf{x}(\alpha))$$

从而 f 沿直线在任意点 $\mathbf{x}(\alpha)$ 的斜率 (slope) 是

$$\frac{d\phi}{d\alpha} = \nabla f(\mathbf{x}(\alpha))^T \mathbf{p} = \mathbf{p}^T \nabla f(\mathbf{x}(\alpha)) \quad (1.4.5)$$

同样, f 沿该直线的曲率 (curvature) 是

$$\frac{d^2\phi}{d\alpha^2} = \frac{d}{d\alpha} \frac{d\phi}{d\alpha} = \mathbf{p}^T \nabla (\nabla f(\mathbf{x}(\alpha))^T \mathbf{p}) = \mathbf{p}^T \nabla^2 f(\mathbf{x}(\alpha)) \mathbf{p} \quad (1.4.6)$$

考虑 Rosenbrock 函数在点 $\mathbf{x}' = (0, 0)^T$ 沿方向 $\mathbf{p} = (1, 0)^T$ 的斜率是 $\mathbf{p}^T \nabla f(\mathbf{x}') = -2$, 曲率是 $\mathbf{p}^T \nabla^2 f(\mathbf{x}') \mathbf{p} = 2$. 需要指出的是, 称 f 沿直线在点 \mathbf{x}' 处 (对应 $\alpha = 0$) 的斜率为 f 在点 \mathbf{x}' 处沿方向 \mathbf{p} 的方向导数, 记为 $f_{\mathbf{p}}(\mathbf{x}')$.

设 $y = f(x)$ 表示平面曲线 C . 如果 f 二次可微, 则曲线 C 在点 x 处的 (局部) 曲率 $k = \frac{f''(x)}{(1 + f'^2(x))^{3/2}}$. 本书中考虑优化问题, 仅利用曲率的符号, 因此曲率指的是曲线的二阶导数. 此外, 斜率与曲率的上述定义与向量 \mathbf{p} 的大小 (长度) 有关, 为避免概念上的模糊不清, 通常要求 $\|\mathbf{p}\| = 1$, 这里 $\|\mathbf{p}\|$ 表示范数 (norm), 用来度量 \mathbf{p} 的大小; 常用的 2-范数定义为 $\|\mathbf{p}\|_2 = \sqrt{\mathbf{p}^T \mathbf{p}}$. 若用 \mathbf{g}' 表示 $\nabla f(\mathbf{x}')$, 则在所有的单位向量 $\|\mathbf{p}\| = 1$ 中, f 沿方向 $\mathbf{g}'/\|\mathbf{g}'\|$ 的斜率最大, 沿 $-\mathbf{g}'/\|\mathbf{g}'\|$ 的斜率最小, 且 \mathbf{g}' 正交于 f 在 \mathbf{x}' 的等值线的切平面 (见图 1.4.3 与习题 1.8).

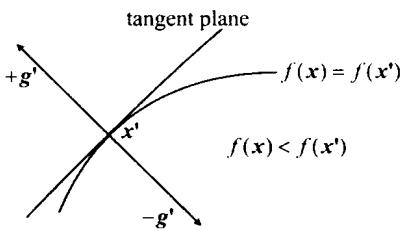


图 1.4.3 梯度向量的性质

下面介绍两种常用的多元函数: 线性函数和二次函数. 一般的线性 (linear) 函数表示为 $f(\mathbf{x}) = \sum_{i=1}^n a_i x_i + b = \mathbf{a}^T \mathbf{x} + b$, 其中 \mathbf{a}, b 是常量. 对于线性函数而言, $\nabla f = \mathbf{a}$ 是常向量, $\nabla^2 f$ 是零矩阵. 一般的二次 (quadratic) 函数表示为

$$q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} - \mathbf{b}^T \mathbf{x} + c \quad (1.4.7)$$

其中 \mathbf{G} 为对称矩阵 (如果不对称, 可以用 $1/2(\mathbf{G} + \mathbf{G}^T)$, 替代 \mathbf{G}), \mathbf{b} 为常向量, c 为常数.

当 \mathbf{u} 和 \mathbf{v} 均是 \mathbf{x} 的函数时, 利用函数乘积的求导法则, 有

$$\nabla(\mathbf{u}^T \mathbf{v}) = (\nabla \mathbf{u}^T) \mathbf{v} + (\nabla \mathbf{v}^T) \mathbf{u} \quad (1.4.8)$$

取 $\mathbf{u} = \mathbf{x}, \mathbf{v} = \mathbf{Gx}$, 并利用 \mathbf{G} 的对称性得 $\nabla q(\mathbf{x}) = \frac{1}{2} (\mathbf{G} + \mathbf{G}^T) \mathbf{x} - \mathbf{b} = \mathbf{Gx} - \mathbf{b}$. 类似可得 $\nabla^2 q(\mathbf{x}) = \mathbf{G}$. 因此二次函数的梯度是线性函数, Hessian 阵是常矩阵.

处理一般的光滑函数必不可少的工具是 Taylor 级数 (Taylor series). 对于一元函数, 它是无穷级数

$$\phi(\alpha) = \phi(0) + \phi'(0)\alpha + \frac{1}{2}\phi''(0)\alpha^2 + \dots \quad (1.4.9)$$

在计算中, 常用的 Taylor 公式形如

$$\phi(\alpha) = \phi(0) + \phi'(0)\alpha + \frac{1}{2}\phi''(0)\alpha^2 + \dots + \frac{1}{p!}\phi^{(p)}(0)\alpha^p + R_p(\alpha) \quad (1.4.10)$$

其中 $R_p(\alpha)$ 是余项 (remainder), 称 $\phi(0) + \phi'(0)\alpha + \frac{1}{2}\phi''(0)\alpha^2 + \dots + \frac{1}{p!}\phi^{(p)}(0)\alpha^p$ 是 p 阶 Taylor 多项式 (Taylor polynomial). 当 $R_p(\alpha) = o(\alpha^p)$ 时, 称式 (1.4.10) 是带 Peano 型余项的 Taylor 公式; 当 $R_p(\alpha) = \frac{1}{p!} \int_0^{\alpha} \phi^{(p+1)}(t)(\alpha - t)^p dt$ 时, 称式 (1.4.10) 是带积分型余项的 Taylor 公式; 当

$R_p(\alpha) = \frac{1}{(p+1)!} \phi^{(p+1)}(\xi) \alpha^{p+1}$ 时, 其中 $\xi \in (0, \alpha)$, 称式(1.4.10)是带 Lagrange 型余项的 Taylor 公式. 这 3 种 Taylor 公式的相同之处是考虑用多项式函数逼近一般函数, 不同的是 Peano 型余项只是定性地告诉我们: 当 $\alpha \rightarrow 0$ 时, 逼近误差是较 α^p 高阶的无穷小量; 而 Lagrange 型余项和积分型余项是以定量的形式给出逼近误差, 这便于对逼近误差进行具体的计算或者估计. 需要指出的是, 本书仅用到 $p=1, 2$ 的情况. 用 $\phi(\alpha) = f(\mathbf{x}(\alpha))$ 表示多元函数 f 沿直线 $\mathbf{x}(\alpha)$ 的取值, 并把式(1.4.5)和式(1.4.6)代入式(1.4.9), 有

$$f(\mathbf{x}' + \alpha \mathbf{p}) = f(\mathbf{x}') + \alpha \mathbf{p}^T \nabla f(\mathbf{x}') + \frac{1}{2} \alpha^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}') \mathbf{p} + \dots$$

如果令 $\mathbf{h} = \alpha \mathbf{p}$, 则得

$$f(\mathbf{x}' + \mathbf{h}) = f(\mathbf{x}') + \mathbf{h}^T \nabla f(\mathbf{x}') + \frac{1}{2} \mathbf{h}^T \nabla^2 f(\mathbf{x}') \mathbf{h} + \dots \quad (1.4.11)$$

这是两种最常用的多元函数的 Taylor 公式. 由于 Hessian 阵 $\nabla^2 f$ 的第 j 列是 $\nabla(\partial f(\mathbf{x}) / \partial x_j)$, 故将 $\nabla(\partial f(\mathbf{x}) / \partial x_j)$ 代入式(1.4.11), 可以得到梯度 $\nabla f(\mathbf{x})$ 的 Taylor 展式

$$\nabla f(\mathbf{x}' + \mathbf{h}) = \nabla f(\mathbf{x}') + \nabla^2 f(\mathbf{x}') \mathbf{h} + \dots \quad (1.4.12)$$

当 $\mathbf{h} \rightarrow \mathbf{0}$ 时, 可以忽略关于 \mathbf{h} 的高次项, 上式即简化成式(1.4.14). 该事实表明一般函数在点 \mathbf{x}' 的某个充分小的邻域内, 它的特性近似于二次函数.

特别地, 利用 Taylor 公式, 式(1.4.7)的二次函数 $q(\mathbf{x})$ 可以表示成

$$q(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^T \mathbf{G} (\mathbf{x} - \mathbf{x}^*) + q(\mathbf{x}^*) \quad (1.4.13)$$

其中 \mathbf{x}^* 是使得 $\nabla q(\mathbf{x}) = \mathbf{0}$ 的点, 即 $\mathbf{G} \mathbf{x}^* = \mathbf{b}$. 此外, 给定点 $\mathbf{x}', \mathbf{x}''$, 记 $\mathbf{g}' = \nabla q(\mathbf{x}')$, $\mathbf{g}'' = \nabla q(\mathbf{x}'')$, 则有

$$\mathbf{g}'' - \mathbf{g}' = \mathbf{G} (\mathbf{x}'' - \mathbf{x}') \quad (1.4.14)$$

即 Hessian 阵 \mathbf{G} 把自变量空间的差变换为梯度空间的差. 这些结果广泛地应用于各种优化技术中.

这里给出的是一些最常用的数学概念, 希望能对后面有关内容的理解有帮助. 后面的某些章节中, 对某些重要结论进行严格论证时, 可能还要不加解释地应用某些稍微复杂的数学知识(参见附录 A), 但是跳过这些内容并不影响读者对内容的整体理解.

1.5 评注和参考

优化可以追溯到变分、Euler 和 Lagrange 的工作. 在 20 世纪 40 年代, 线性规划的发展拓宽了领域, 并且在过去的 70 年里促进了现代优化理论和应用的诸多进展.

本书对建模并没花费太多的笔墨. 事实上, 建模相当重要, 理论上等价的不同模型在计算中有可能差异巨大, 所以需要对各种模型及相应的算法有所了解, 来进行综合设计. 关于各种应用领域的建模技术可以参见 Dantzig、Ahuja、Magnanti 和 Orlin、Fourer、Gay 的专著.

习题 1

1.1 市场上 5 种价格不同的合金, 其价格以及金属 A 和 B 的含量如下:

合 金	1	2	3	4	5
A 的含量/ (%)	10	25	50	75	95
B 的含量/ (%)	90	75	50	25	5
单价/(万元/吨)	5	4	3	2	1.50

可以将这些合金进行适当组合来生产想要的合金. 某制造商希望生产一种金属 A 和 B 含量各为 30% 和 70% 的合金. 制造商希望确定出满足 A 和 B 含量要求, 同时组合费用最低的各种合金的数量. 将此问题建模为线性规划问题.

- 1.2 一个原油精练场有 800 万桶原油 A 和 500 万桶原油 B 用于安排下个月的生产. 可用这些资源来生产售价为 38 元/桶的汽油, 或者生产售价为 33 元/桶的民用燃料油. 有 3 种生产过程可供选择, 各自的生产参数如下:

参 数	过程 1	过程 2	过程 3
输入原油 A/桶	3	1	5
输入原油 B/桶	5	1	3
输出汽油/桶	4	1	3
输出民用燃料油/桶	3	1	4
成本/元	51	11	40

例如, 对于第一个过程而言, 利用 3 桶原油 A 和 5 桶原油 B 可以生产 4 桶汽油和 3 桶民用燃料油. 表格中的成本指总的成本(即原油成本和生产过程的成本). 将此问题建模成线性规划, 其能使管理者极大化下个月的净利润.

- 1.3 请利用优化软件求解问题

$$\begin{aligned} & \text{minimize} && (x_1 - 2)^2 + (x_2 - 1)^2 \\ & \text{subject to} && x_1^2 - x_2 \leq 0 \\ & && x_1 + x_2 \leq 2 \end{aligned}$$

- 1.4 给出 n 元函数的梯度向量和 Hessian 阵:

(a) $\mathbf{a}^T \mathbf{x}$: \mathbf{a} 是常向量;

(b) $\mathbf{x}^T \mathbf{A} \mathbf{x}$: \mathbf{A} 是非对称的常矩阵;

(c) $\frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x}$: \mathbf{A} 是对称的常矩阵, \mathbf{b} 是常向量;

(d) $\mathbf{r}^T \mathbf{r}$: $\mathbf{r} = (r_1(\mathbf{x}), \dots, r_m(\mathbf{x}))^T$ 是依赖于 \mathbf{x} 的 m 维向量, 记 $\nabla \mathbf{r}^T$ 为 \mathbf{A}^T , 它一般不是常量.

- 1.5 写出函数 $\cos(1/x)$ 在 $x \neq 0$ 处的二阶 Taylor 公式; 写出函数 $\cos x$ 在任一点 x 的三阶 Taylor 公式; 写出函数 $\cos x$ 在 $x=1$ 处的二阶 Taylor 多项式.

- 1.6 写出 m 维向量值函数 $\mathbf{f}(\mathbf{x})$ 在 \mathbf{x}' 的 Taylor 展式, 用 \mathbf{A}^T 表示 $\nabla \mathbf{f}^T$.

- 1.7 假设在点 \mathbf{x}' 有 $\mathbf{g}' \neq \mathbf{0}$, 证明在所有单位向量 $\mathbf{p}^T \mathbf{p} = 1$ 中, 函数沿方向向量 $\mathbf{p} = \mathbf{g}' / \|\mathbf{g}'\|_2$ 的斜率最大. 称该方向是函数的最速上升(steepest ascent)方向.

- 1.8 假设在点 \mathbf{x}' 有 $\mathbf{g}' \neq \mathbf{0}$, 证明向量 $\pm \mathbf{g}'$ 与过点 \mathbf{x}' 的等值线的切平面正交.

第 2 章 线性规划：基本理论与方法

二战期间,美国军方为了保证士兵的健康,对每餐食品(牛奶、黄油、胡萝卜等)中的营养成份(蛋白质、脂肪、维生素等)有定量的规定. 不同的食物提供不同比例的营养成份. 由于战争条件的限制,在一盒套餐中,如何寻求最佳的配餐方案,即决定各种食品的数量,使得既能满足营养成份的需要,又可以降低成本,就是摆在美国空军管理部统计控制战斗分析处年轻的 Dantzig 面前的管理问题之一. Dantzig 利用线性目标和线性不等式建立了一个数学模型(当时称为 programming in a linear structure, 后来由 Koopmans 建议改为 linear programming), 并发明了后来被称为 20 世纪最成功的 10 种算法之一的求解线性规划的单纯形法.

本章先给出线性规划的几个典型实例,之后介绍线性规划标准形和基本可行解等基本概念. 在这些内容的基础上,将进一步学习线性规划的基本性质、著名的单纯形法和对偶理论.

2.1 基本性质

配餐问题. 假定共有 n 种不同的食品, 第 j 种食品的单价是 c_j . 另外, 有 m 种营养成份, 假设每单位第 j 种食品含 a_{ij} 单位的第 i 种营养成份. 为了保证健康, 规定每人每天至少应摄取 b_i 单位的第 i 种营养. 问题是如何确定每人每天食用各种食品的数量, 才能既保证基本的营养需求, 又使成本最少? 用 x_j 表示每人每天食用第 j 种食品的数量(即 x_j 单位), 此问题需要确定 x_j , 使得在满足营养需求

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\geq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\geq b_m \end{aligned}$$

和非负约束 $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$ 的同时, 让总成本 $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ 达到最小.

运输问题. 假设某种产品有 m 个产地 A_1, A_2, \dots, A_m , 它们的产量分别为 a_1, a_2, \dots, a_m . 该产品又有 n 个销售地 B_1, B_2, \dots, B_n , 它们的需求量分别是 b_1, b_2, \dots, b_n . 把产品从第 i 个产地运到第 j 个销售地的单位运价是 c_{ij} . 这些数据用表格表示为

	B_1	B_2	\cdots	B_n	产 量
A_1	c_{11}	c_{12}	\cdots	c_{1n}	a_1
A_2	c_{21}	c_{22}	\cdots	c_{2n}	a_2
\vdots	\vdots	\vdots		\vdots	\vdots
A_m	c_{m1}	c_{m2}	\cdots	c_{mn}	a_m
销量	b_1	b_2	\cdots	b_n	

进一步假定产销平衡, 即 $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. 问如何安排从 A_i 到 B_j 的运输量 x_{ij} , 才能既满足各销售地的需求, 又使总运费最少?

可将此问题表述为

$$\begin{aligned} \text{minimize}_{\mathbf{x}} \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^n x_{ij} = a_i, \quad i = 1, 2, \dots, m \\ & \sum_{i=1}^m x_{ij} = b_j, \quad j = 1, 2, \dots, n \\ & x_{ij} \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \end{aligned} \quad (2.1.1)$$

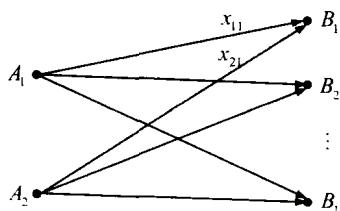


图 2.1.1 运输问题示意图

除了物资运输问题以外, 其他一些实际问题也可以表述成上述模型. 由于这类问题最早是从物资运输中归结出来的, 因此称为运输问题. 图 2.1.1 是运输问题的示意图, 其中 $m=2, n=12$.

运输问题显然是具有 mn 个变量的线性规划问题. 如果将运输问题表述成常用的矩阵形式, 得到的系数矩阵是仅由 0 和 1 组成的 $(m+n) \times (mn)$ 阶矩阵. 鉴于该问题的重要性和特殊性, 有求解它的专用高效算法, 详见

3.2.1 小节的讨论.

制造问题. 假设有一台设备, 可以从事 n 种不同的生产活动, 每种活动均可以生产不同数量的 m 种商品. 生产时间 $x_i \geq 0$, 单位生产时间的耗费是 c_i 元, 单位生产时间内可生产 a_{ij} 单位的第 j 种商品. 如果 m 种商品的需求量分别是 b_1, b_2, \dots, b_m , 希望以最小的成本来安排生产, 则得到线性规划问题

$$\begin{aligned} \text{minimize} \quad & c_1 x_1 + c_2 x_2 + \dots + c_n x_n \\ \text{subject to} \quad & a_{11} x_1 + a_{21} x_2 + \dots + a_{n1} x_n = b_1 \\ & a_{12} x_1 + a_{22} x_2 + \dots + a_{n2} x_n = b_2 \\ & \vdots \\ & a_{1m} x_1 + a_{2m} x_2 + \dots + a_{nm} x_n = b_m \\ & x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \end{aligned}$$

库存问题. 考虑对某种库存的商品进行买或卖, 使某些时段内的利润最大. 假定仓库的固定容量是 C , 在一个时段中每单位商品的库存费用是 r . 商品的价格在不同的时段(比如几个月)有波动, 但同一时段的买进价格和卖出价格是不变的. 开始时, 仓库是空的, 并要求在最后一个时段仓库也是空的.

为了表述该问题, 每个时段均需要引入变量. 特别地, 设 x_i 是第 i 时段开始时的商品库存量. 设 u_i 和 s_i 分别表示在第 i 时段买进和卖出的数量, 第 i 时段的卖出价格 p_i 给定. 如果有 n 个时段, 问题表示为

$$\begin{aligned}
 & \text{maximize} \quad \sum_{i=1}^n (p_i s_i - r x_i) \\
 & \text{subject to} \quad x_{i+1} = x_i + u_i - s_i, \quad i = 1, 2, \dots, n-1 \\
 & \quad x_n + u_n - s_n = 0 \\
 & \quad x_1 = 0, x_i \leq C, \quad i = 2, 3, \dots, n \\
 & \quad x_i \geq 0, u_i \geq 0, s_i \geq 0, \quad i = 2, 3, \dots, n
 \end{aligned}$$

对 $n=3$ 的情况写出约束的显式表示, 即

$$\begin{aligned}
 -u_1 + s_1 + x_2 &= 0 \\
 -x_2 - u_2 + s_2 + x_3 &= 0 \\
 x_2 &\leq C \\
 -x_3 - u_3 + s_3 &= 0 \\
 x_3 &\leq C
 \end{aligned}$$

请注意可以根据不同时段将系数矩阵剖分成块. 仅对角线及紧靠其上的块具有非零元. 涉及时段的问题通常都具有这种典型结构.

2.1.1 标准形

在上述各种典型问题中, 变量和约束具体形式表现不一. 可将任何线性规划问题统一成如下的标准形 (standard form)

$$\begin{aligned}
 & \text{minimize} \quad c_1 x_1 + c_2 x_2 + \dots + c_n x_n \\
 & \text{subject to} \quad a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1 \\
 & \quad a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2 \\
 & \quad \vdots \\
 & \quad a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m \\
 & \quad x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0
 \end{aligned}$$

其中 b_i, c_i 和 a_{ij} 是固定的常数, 实数 x_i 是决策变量. 用更紧凑的矩阵向量记号, 标准形可以表示为

$$\begin{aligned}
 & \text{minimize} \quad \mathbf{c}^T \mathbf{x} \\
 & \text{subject to} \quad \mathbf{A} \mathbf{x} = \mathbf{b} \\
 & \quad \mathbf{x} \geq \mathbf{0}
 \end{aligned} \tag{2.1.2}$$

这里, \mathbf{x}, \mathbf{c} 是 n 维向量, \mathbf{A} 是 $m \times n$ 矩阵, \mathbf{b} 是 m 维向量. 向量不等式 $\mathbf{x} \geq \mathbf{0}$ 表示 \mathbf{x} 的每个分量都是非负的. 下面说明如何将各种其他形式的线性规划问题转化成标准形.

考虑问题

$$\begin{aligned}
 & \text{minimize} \quad c_1 x_1 + c_2 x_2 + \dots + c_n x_n \\
 & \text{subject to} \quad a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \leq b_1 \\
 & \quad a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n \leq b_2 \\
 & \quad \vdots \\
 & \quad a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n \leq b_m \\
 & \quad x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0
 \end{aligned} \tag{2.1.3}$$

其约束集完全由线性不等式确定. 该问题又可表示为

$$\begin{aligned}
 & \underset{\mathbf{x}, \mathbf{y}}{\text{minimize}} \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
 & \text{subject to} \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n + y_1 = b_1 \\
 & \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n + y_2 = b_2 \\
 & \quad \vdots \\
 & \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n + y_m = b_m \\
 & \quad x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0, y_1 \geq 0, y_2 \geq 0, \dots, y_m \geq 0
 \end{aligned}$$

称将不等式约束转化成等式约束所引入的非负变量 y_i 为松弛(slack)变量. 这样将原问题转化为具有 $n+m$ 个未知数 $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m$ 的标准形问题. 这个新问题的等式约束系数矩阵具有特殊形式 $[\mathbf{A} \quad \mathbf{I}]$ (可将列剖分成两个集合: 前 n 列由原来的系数矩阵 \mathbf{A} 组成, 后 m 列是 m 阶单位矩阵).

如果上述问题中的某不等式是反向的, 比如设 $a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n \geq b_i$, 易见其等价于 $a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n - y_i = b_i, y_i \geq 0$. 此时称将“ \geq ”型不等式转化成等式的变量 y_i 为盈余(surplus)变量. 有时盈余变量和松弛变量不作区分而统称为松弛变量.

如果线性规划中有一个或多个变量没有非负要求, 即自由(free)变量, 则问题可通过以下两种方法之一转化成标准形. 假定在式(2.1.2)中, $x_1 \geq 0$ 没有出现, 即 x_1 无符号限制.

第一种方法是消元(elimination), 即利用约束方程中的一个方程消去 x_1 . 如在式(2.1.2)的 m 个方程中任取一个 x_1 的系数非零的方程

$$a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n = b_i$$

其中 $a_{i1} \neq 0$, 则 x_1 可以表示成其他变量的线性组合加一个常数, 即

$$x_1 = \frac{1}{a_{i1}} [b_i - (a_{i2} x_2 + \cdots + a_{in} x_n)] \quad (2.1.4)$$

将式(2.1.4)代入式(2.1.2), 则得到一个仅有 x_2, x_3, \dots, x_n 的标准形问题. 因为 x_1 的符号没有限制, 所以将非负变量 x_2, x_3, \dots, x_n 代入式(2.1.4)后即可得到原问题的一个可行解. 利用这种方法, 得到有 $m-1$ 个等式约束和 $n-1$ 个变量的线性规划标准形. 求解该问题, 然后将解代入式(2.1.4)可以得到变量 x_1 的值. 若有多个自由变量, 则可依次反复执行上述消元过程, 直到不再有自由变量为止. 例如

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} \quad x_1 + 3x_2 + 4x_3 \\
 & \text{subject to} \quad x_1 + 2x_2 + x_3 = 5 \\
 & \quad 2x_1 + 3x_2 + x_3 = 6 \\
 & \quad x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

因为 x_1 是自由变量, 由第一个约束得 $x_1 = 5 - 2x_2 - x_3$, 代入原目标和约束, 得到等价问题(从目标中减去常数 5 并不影响解)

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} \quad x_2 + 3x_3 \\
 & \text{subject to} \quad x_2 + x_3 = 4 \\
 & \quad x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

这是一个标准形, 求解它得 $x_2 = 4, x_3 = 0$, 进一步可得 $x_1 = -3$.

第二种方法是采用变量替换, 令 $x_1 = u_1 - v_1, u_1 \geq 0, v_1 \geq 0$. 在式(2.1.2)中以 $u_1 - v_1$ 代替

x_1 , 则所有的变量都要求非负, 得到的新问题有 $n+1$ 个变量 $u_1, v_1, x_2, x_3, \dots, x_n$. 因为给 u_1, v_1 增加一个常数不改变 x_1 (即对 x_1 的这种表示不唯一), 这显然存在某种冗余度, 但并不妨碍利用单纯形法求解它.

2.1.2 基本可行解

考虑标准形式(2.1.2)的线性约束系统

$$Ax = b, \quad x \geq 0 \quad (2.1.5)$$

为避免方程组 $Ax = b$ 无解或者有唯一解这些平凡情况, 对系数矩阵 A 统一进行行满秩假定: $m \times n$ 矩阵 A 满足 $m \leq n$, 且 m 个行向量线性无关.

首先需要说明该假定不失一般性. 如果行不满秩, 则线性等式系统 $Ax = b$ 或者无解, 或者至少有一个方程是冗余的, 冗余时删除所有冗余的方程便等价转化到行满秩的情形. 行满秩隐含了 $m \leq n$. 如果 $m = n$, 那么 $Ax = b$ 解唯一, 从而线性系统(2.1.5)或者无解, 或者有唯一解. 需要指出的是, 满秩假定并不能保证系统(2.1.5)一定有解. 判断该系统是否有解的方法将在 2.2.5 小节中讨论.

本质上线性方程组 $Ax = b$ 有解表明 b 可以用 A 的 n 个列线性表示. 在行满秩假定下, 该线性方程组有无穷多组解. 任意选定 A 的 m 个线性无关列作为 \mathbb{R}^m 的一个基, 则 b 可由这 m 列唯一地线性表示, 此时的表示系数 x 称为基本(basic)解, 与这 m 列对应的变量称为基(basic)变量, 另外 $n-m$ 个被置为 0 的变量称为非基变量. 因为非基变量取值恒为零, 本书后面提及基本解时, 通常仅指明基变量的取值.

例 2.1.1 (基本解) 考虑

$$A = \begin{bmatrix} 1 & -1 & 2 & 1 & 0 \\ 0 & 1 & 6 & 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 12 \end{bmatrix}$$

比如取 A 的前两列作为基, 即令 $x_3 = x_4 = x_5 = 0$, 得 $x_1 - x_2 = 8, x_2 = 12$, 解得 $x_1 = 20, x_2 = 12$. 这样可得基本解 $x_1 = 20, x_2 = 12$, 其中前 2 个变量是基变量, 后 3 个是非基变量.

基本解 x 可以更为直接地定义成: 满足 $Ax = b$, 且所有非零分量对应的 A 的列线性无关. 进一步, 如果基本解中有一个或多个基变量的值为零, 则称其为退化的(degenerate)基本解, 否则称为非退化的基本解. 显然, 从非退化基本解中能够识别出基变量, 而退化基本解中取零值的基变量和非基变量经常能够互换. 请读者结合习题 2.5 思考可以互换的充分必要条件是什么. 很容易把这些定义推广到一般的线性系统(2.1.5).

定义 2.1.1 称满足线性约束系统(2.1.5)的向量 x 是可行解. 称式(2.1.5)的非负基本解是基本(basic)可行解; 如果这个非负的基本解是退化的, 称为退化的基本可行解, 否则称为非退化的基本可行解.

举个例子来说明上面一系列的定义. 假定 A 的前 m 列线性无关, 它们组成 $m \times m$ 的非奇异矩阵 B , 称为基矩阵(简称为基). 则 $B^{-1}b$ 是关于基 B 的基本解. 它退化与否取决于 $B^{-1}b$ 是否含零分量. 进一步, 如果 $B^{-1}b \geq 0$, 则 x 是基本可行解; 如果 $B^{-1}b > 0$, 则 x 是非退化的基本可行解. 关于基本可行解, 有如下重要的定理.

定理 2.1.1 (存在性) 如果线性约束系统(2.1.5)有可行解, 则必有基本可行解.

证明 记 A 的列为 a_1, a_2, \dots, a_n . 假定 $x = (x_1, x_2, \dots, x_n)^\top$ 是可行解. x 不为零向量, 否则它已

经是基本可行解. 不失一般性, 假定 x 仅有前 p 个分量大于零, 则有

$$x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \cdots + x_p \mathbf{a}_p = \mathbf{b} \quad (2.1.6)$$

分 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性无关和线性相关两种情况进行讨论.

如果 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性无关(必有 $p \leq m$), 显然 x 是基本可行解. 特别地, 若 $p < m$, 则 x 是退化的基本可行解; 若 $p = m$, 则 x 是非退化的基本可行解. 现假定 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性相关, 则存在不全为零的常数 y_1, y_2, \dots, y_p , 其中至少有一个是正的(否则全体反号), 使得

$$y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + \cdots + y_p \mathbf{a}_p = \mathbf{0} \quad (2.1.7)$$

由式(2.1.6)和式(2.1.7)得到

$$(x_1 - \epsilon y_1) \mathbf{a}_1 + (x_2 - \epsilon y_2) \mathbf{a}_2 + \cdots + (x_p - \epsilon y_p) \mathbf{a}_p = \mathbf{b}$$

对任意的 ϵ 成立. 记 $\mathbf{y} = (y_1, y_2, \dots, y_p, 0, 0, \dots, 0)^\top$, 上面即证明了对任意的 ϵ , $x - \epsilon \mathbf{y}$ 是方程组的解, 但 $x - \epsilon \mathbf{y} \geq \mathbf{0}$ 不一定成立. $\epsilon = 0$ 对应初始可行解 x . 当 ϵ 从零开始增加时, 依据 y_i 是负的、正的或零, $x - \epsilon \mathbf{y}$ 的各分量将会增大、减小或保持不变. 因为至少有一个 y_i 是正的, 所以 ϵ 增加时, 至少有一个分量会减小. 增加 ϵ 直到一个或多个分量首次同时变成零. 具体地, 置

$$\epsilon' = \min \left\{ \frac{x_i}{y_i} : y_i > 0, i = 1, 2, \dots, m \right\} > 0 \quad (2.1.8)$$

易见 $x - \epsilon' \mathbf{y}$ 可行, 且最多有 $p-1$ 个正分量. 若有必要, 重复该过程, 逐步消去正分量, 直到得到一个可行解, 其非零分量对应的列线性无关. 然后由第一种情况可证明该定理. ■

2.1.3 基本定理

下面的线性规划基本定理说明基本可行解在求解线性规划问题时的重要性. 定理的证明方法在许多方面与结论本身同样重要, 因为它代表单纯形法推演的开始. 定理说明线性规划的最优值总可以在某个基本可行解处达到, 故找最优解时仅需考虑基本可行解.

考虑线性规划标准形, 如果问题(2.1.2)的某个最优解 x 还是基本解, 则称 x 是最优基本解.

定理 2.1.2 (线性规划基本定理) 如果线性规划问题(2.1.2)有最优解, 则必有最优基本解.

证明 设 $x = (x_1, x_2, \dots, x_n)^\top$ 为最优解, 类似定理 2.1.1 的证明, 假定恰有 p 个正分量 x_1, x_2, \dots, x_p , 也存在两种情况.

若 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性无关, x 即为最优基本可行解; 若 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性相关, 类比定理 2.1.1 的证明, 可得 $x - \epsilon' \mathbf{y}$ 为可行解, 且多一个零分量. 下面证明 $x - \epsilon' \mathbf{y}$ 仍是最优解. 因为对充分小的 $\epsilon > 0$, $x - \epsilon \mathbf{y}$ 与 $x + \epsilon \mathbf{y}$ 都是可行的, 由 x 的最优性可得 $\mathbf{c}^\top \mathbf{y} = 0$ (否则若 $\mathbf{c}^\top \mathbf{y} \neq 0$, 可以确定小的 ϵ 值使 $\mathbf{c}^\top x - \epsilon \mathbf{c}^\top \mathbf{y}$ 或者 $\mathbf{c}^\top x + \epsilon \mathbf{c}^\top \mathbf{y}$ 比 $\mathbf{c}^\top x$ 小, 且保持可行性, 这与 x 是最优解矛盾). 于是 $x - \epsilon' \mathbf{y}$ 也是最优解.

若有必要, 重复上述过程, 逐步消去正分量, 直到得到一个最优解, 其非零分量对应的列线性无关, 即为最优基本解. ■

本定理将求解线性规划这一连续问题转化成在所有的基本可行解中进行搜索的组合优化问题. 注意, n 个变量、 m 个约束的线性规划问题最多存在 $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ 个基本可行解(证明留给读者). 在 2.2 节中将介绍以基本定理为基石的一种有效方法——单纯形法.

2.1.4 几何直观

本小节借助于凸集的概念及性质给出基本可行解的几何直观,进而加深对基本定理的理解.

定义 2.1.2 称 \mathbb{R}^n 中的集合 C 是凸的(convex),如果任给 $x, y \in C$,对任意 $\theta \in (0, 1)$,点 $\theta x + (1-\theta)y \in C$,即连接集合中任意两点的线段也包含于该集合.进一步,若对任意的 $\alpha > 0$ 和 $x \in C$ 有 $\alpha x \in C$,则称 C 是凸锥(convex cone).

这里定义的凸锥是以原点为顶点的.此外,由凸集的定义,读者不难自行验证如下一些事实.

命题 2.1.1 (凸集的基本性质) (i) 任意多个凸集的交是凸集.

(ii) 如果 C 和 D 是凸集,则集合 $C + D = \{x: x = c + d, c \in C, d \in D\}$ 是凸集.

(iii) 如果 C 是凸集且 β 是实数,则集合 $\beta C = \{x: x = \beta c, c \in C\}$ 是凸集.

下面给出一些重要的凸集,其中 a 是非零向量, γ 是实数.

定义 2.1.3 称集合 $H = \{x \in \mathbb{R}^n: a^T x = \gamma\}$ 是 \mathbb{R}^n 中的超平面(hyperplane), a 是超平面的法向量(normal vector).称 $H_+ = \{x: a^T x \geq \gamma\}$ ($H_- = \{x: a^T x \leq \gamma\}$)是由超平面 $H = \{x: a^T x = \gamma\}$ 确定的正(负)闭半空间(positive (negative) closed half spaces).

定义 2.1.4 称有限个闭半空间的交集是多面集(polytope).

极点定义为不能位于连接该集合中其他两点的开线段上的点,它在多面集的表示理论中极其重要.特别地,空间中多面体的顶点就是它的极点.

定义 2.1.5 凸集 C 中的一点 x 称为 C 的极点(extreme point),如果存在 C 中两点 y, z 及某一 $\theta \in (0, 1)$ 满足 $x = \theta y + (1-\theta)z$,则必有 $y = z$.

有了上面这些定义,接下来说明标准形的基本可行解和它的可行域的极点是一回事.只要注意到,二者都不能表示成两个不同的解(点)的凸组合,就很容易理解该事实.

定理 2.1.3 (极点和基本可行解的等价性) 设 C 是由线性约束系统(2.1.5)的所有可行解组成的集合,即 $C = \{x: Ax = b, x \geq 0\}$,则向量 x 是 C 的极点当且仅当 x 是系统(2.1.5)的基本可行解.

证明 首先假定 x 是系统(2.1.5)的一个基本可行解,则它至少有 $n-m$ 个元素是零.不妨设 $x = (x_1, x_2, \dots, x_m, 0, 0, \dots, 0)^T$,则 $x_1 a_1 + x_2 a_2 + \dots + x_m a_m = b$,其中 a_1, a_2, \dots, a_m 是 A 的前 m 列且线性无关.假设 x 能表示成 C 中其他两个点 y 和 z 的凸组合,即 $x = \theta y + (1-\theta)z$, $0 < \theta < 1$.因为 x, y 和 z 的所有分量非负,且 $0 < \theta < 1$,由此立即可得 y 和 z 的后 $n-m$ 个分量为零.这样就有 $y_1 a_1 + y_2 a_2 + \dots + y_m a_m = b$ 和 $z_1 a_1 + z_2 a_2 + \dots + z_m a_m = b$.因为向量 a_1, a_2, \dots, a_m 线性无关,所以有 $y_i = z_i = x_i$, $i = 1, 2, \dots, m$.因此 x 是 C 的极点.

反之,假定 x 是 C 的极点,且 x 的前 p 个分量非零,则 $x_1 a_1 + x_2 a_2 + \dots + x_p a_p = b$.为了证明 x 是基本可行解,必须证明向量 a_1, a_2, \dots, a_p 线性无关.假设 a_1, a_2, \dots, a_p 线性相关,则存在它们的非平凡线性组合为零,即存在不全为零的系数 y_1, y_2, \dots, y_p 使得 $y_1 a_1 + y_2 a_2 + \dots + y_p a_p = 0$.定义 n 维向量 $y = (y_1, y_2, \dots, y_p, 0, 0, \dots, 0)^T$.因为 $x_i > 0$, $1 \leq i \leq p$,故可以选 $\epsilon > 0$ 充分小使得 $x + \epsilon y \geq 0$, $x - \epsilon y \geq 0$.然后有 $x = (x + \epsilon y)/2 + (x - \epsilon y)/2$,即 x 可表示为 C 中两个不同向量的凸组合,这与 x 为 C 的极点矛盾.这样 a_1, a_2, \dots, a_p 线性无关,所以 x 是一个基本可行解(如果 $p < m$,则 x 是退化的基本可行解). ■

极点和基本可行解之间的这种对应,使得我们能够证明由线性规划问题的约束条件所定义的凸多面集的某些几何性质.

推论 如果与线性系统(2.1.5)对应的凸集非空,则它有且仅有有限多个极点.如果线性规划问题(2.1.2)有最优解,则至少有一个极点是最优解.

例 2.1.2 考虑 \mathbb{R}^3 中由 $x_1 + x_2 + x_3 = 1, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ 定义的约束集,该集合如图 2.1.2(a) 所示. 它有 3 个极点,与 $x_1 + x_2 + x_3 = 1$ 的 3 个基本解对应.

例 2.1.3 考虑 \mathbb{R}^3 中由 $x_1 + x_2 + x_3 = 1, 2x_1 + 3x_2 = 1, x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ 定义的约束集. 该集合如图 2.1.2(b) 所示. 它有两个极点,与两个基本可行解对应. 注意方程组本身有 3 个基本解 $(2, -1, 0)^T, (1/2, 0, 1/2)^T$ 和 $(0, 1/3, 2/3)^T$, 其中第一个是不可行的.

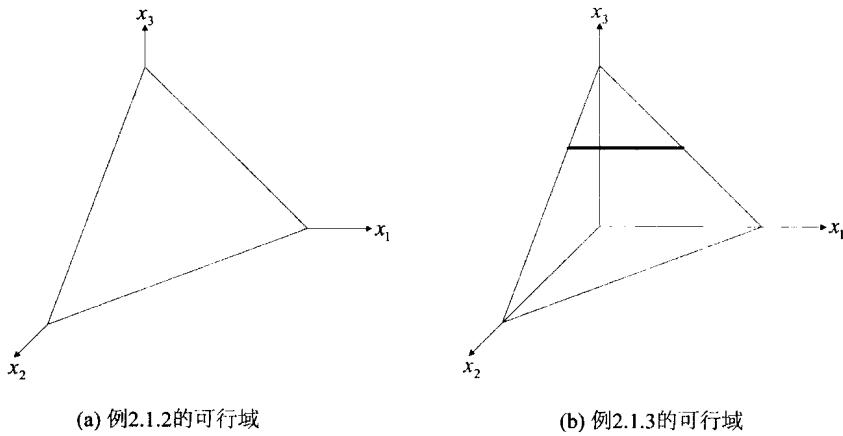


图 2.1.2 极点与基本可行解

例 2.1.4 考虑 \mathbb{R}^2 中由不等式

$$x_1 + \frac{8}{3}x_2 \leq 4$$

$$x_1 + x_2 \leq 2$$

$$2x_1 \leq 3$$

$$x_1 \geq 0, x_2 \geq 0$$

定义的约束集. 该集合如图 2.1.3(a) 所示. 易验证它有 5 个极点. 为了将此例与一般结论相比, 引入松弛变量产生 \mathbb{R}^5 中的等价集合

$$x_1 + \frac{8}{3}x_2 + x_3 = 4$$

$$x_1 + x_2 + x_4 = 2$$

$$2x_1 + x_5 = 3$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0$$

通过置任意的两个变量为零, 然后关于剩下的 3 个变量求解所得方程组即可得该系统的基本解. 如图 2.1.3(a) 所示, 图的每条边对应的点有一个变量取零, 极点对应的点有两个变量取零. 该例说明: 即使没有把问题化成标准形, 由线性规划约束条件所定义集合的极点也是与基本可行解一一对应的(见习题 2.6). 这可以通过在图中显示目标函数进行更直接的说明.

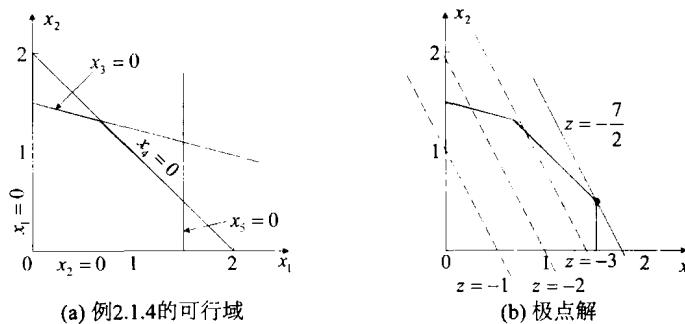


图 2.1.3 极点与极点解

假如在该例中,设要极小化的目标函数是 $-2x_1 - x_2$. 对于固定的 z ,满足 $-2x_1 - x_2 = z$ 的点集是一条直线. 随着 z 的改变,可以得到如图 2.1.3(b)所示的平行线. 线性规划问题的最优值是相应直线与可行域有公共交点的 z 的最小值. 至少在二维时,显然总会有一个极点解. 本例中极点解是 $(3/2, 1/2)^\top$,此时 $z = -7/2$.

2.2 单纯形法

线性规划的基本定理说明,只要穷举有限多个基本可行解,就一定能找到最优解. 由于基本可行解的数目一般是指数多个,为避免盲目搜索,需要一种有效的搜索机制,比如本节将要介绍的单纯形法. 它的搜索机制是从一个基本可行解迭代到相邻的一个基本可行解,并使得目标函数值有所下降,直到找到最优解或者判定问题无界. 这里先用一个小例子来直观地理解这种方法.

例 2.2.1 (单纯形法) 考虑

$$\begin{aligned} & \text{minimize} && x_1 + 2x_2 + 3x_3 + 4x_4 \\ & \text{subject to} && x_1 + x_2 + x_3 + x_4 = 1 \\ & && x_1 + x_3 - 3x_4 = \frac{1}{2} \\ & && x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

先由第二个方程得到 $x_1 = 1/2 - x_3 + 3x_4$,代入第一个方程得到 $x_2 = 1/2 - 4x_4$. 进一步,将 x_1 , x_2 代入目标函数,得到 $z = 3/2 + 2x_3 - x_4$. 这样,得到仅利用变量 x_3, x_4 表述的且与原问题等价的问题,即

$$\begin{aligned} & \text{minimize} && z = \frac{3}{2} + 2x_3 - x_4 \\ & \text{subject to} && x_2 = \frac{1}{2} - 4x_4 \\ & && x_1 = \frac{1}{2} - x_3 + 3x_4 \\ & && x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

由这个等价问题可以得到基本可行解 $(1/2, 1/2, 0, 0)^T$, 且易于判断如果 x_1 从 0 开始逐渐增大, 则目标函数会减小; 但 x_1 一旦大于 $1/8$, 变量 x_2 将变成负的. 这样, 利用变量 x_2, x_3 表述原问题, 得到

$$\begin{aligned} \text{minimize} \quad z &= \frac{11}{8} + \frac{1}{4}x_2 + 2x_3 \\ \text{subject to} \quad x_4 &= \frac{1}{8} - \frac{1}{4}x_2 \\ x_1 &= \frac{7}{8} - \frac{3}{4}x_2 - x_3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

由该表述得新的基本可行解 $(7/8, 0, 0, 1/8)^T$; 此外, 因为所有的可行解都非负, 因而所有可行解处的目标值不会比这个新的基本可行解处的小. 这样可得问题的解.

2.2.1 既约费用系数

考虑标准形问题(2.1.2). 为方便起见, 先进行非退化假设: 问题(2.1.2)的每个基本可行解都是非退化的. 将在 2.2.4 小节中交待如何处理该假定不满足的情形.

不妨设已经获得了基变量为 x_1, x_2, \dots, x_m 的基本可行解, 则系数矩阵 A 的前 m 列线性无关, 即 $A = [B \ N]$, 其中 $B \in \mathbb{R}^{m \times m}$ 非奇异, 是与该基本可行解对应的基. 相应记 $x^T = (x_B^T, x_N^T)$, 给 $Ax = b$ 左乘 B^{-1} , 可以得到等价的约束条件

$$x_B + B^{-1}N x_N = B^{-1}b \quad (2.2.1)$$

也可以写出与之对应的基本可行解为 $x_B = B^{-1}b, x_N = 0$, 目标值 $z_0 = c_B^T x_B$, 其中 $c_B^T = (c_1, c_2, \dots, c_m)$. 式(2.2.1)称为规范形(canonical form), 用方程组形式写出来即为

$$\left. \begin{array}{l} x_1 + y_{1,m+1}x_{m+1} + \dots + y_{1n}x_n = y_{10} \\ x_2 + y_{2,m+1}x_{m+1} + \dots + y_{2n}x_n = y_{20} \\ \vdots \qquad \vdots \\ x_m + y_{m,m+1}x_{m+1} + \dots + y_{mn}x_n = y_{m0} \\ x_1, x_2, \dots, x_n \geq 0 \end{array} \right\} \quad (2.2.2)$$

其中 $y_{i,m+1}$ 是 $B^{-1}N$ 的第 i 行、第 j 列的元素, $y_{i0} = (B^{-1}b)_i$. 相应的基本可行解是 $x_1 = y_{10} > 0, x_2 = y_{20} > 0, \dots, x_m = y_{m0} > 0, x_{m+1} = 0, \dots, x_n = 0$. 方程组(2.2.2)的系数矩阵的一般形式如表 2.2.1 所列.

表 2.2.1 方程组(2.2.2)的系数矩阵

x_1	x_2	\dots	x_m	x_{m+1}	x_{m+2}	\dots	x_n	$B^{-1}b$
1	0	\dots	0	$y_{1,m+1}$	$y_{1,m+2}$	\dots	y_{1n}	y_{10}
0	1	\dots	0	$y_{2,m+1}$	$y_{2,m+2}$	\dots	y_{2n}	y_{20}
		\ddots				\vdots		\vdots
0	0	\dots	1	$y_{m,m+1}$	$y_{m,m+2}$	\dots	y_{mn}	y_{m0}

因为非基变量 $x_{m+1}, x_{m+2}, \dots, x_n$ 是自由变量, 一旦给定它的一组取值, 由式(2.2.2)可以求得其余变量

$$\left. \begin{aligned} x_1 &= y_{10} - \sum_{j=m+1}^n y_{1j} x_j \\ x_2 &= y_{20} - \sum_{j=m+1}^n y_{2j} x_j \\ &\vdots \\ x_m &= y_{m0} - \sum_{j=m+1}^n y_{mj} x_j \end{aligned} \right\} \quad (2.2.3)$$

将式(2.2.3)代入问题(2.1.2)的目标函数,消去 x_1, x_2, \dots, x_m ,得

$$z = \mathbf{c}^\top \mathbf{x} = z_0 + (c_{m+1} - z_{m+1})x_{m+1} + \dots + (c_n - z_n)x_n \quad (2.2.4)$$

其中

$$z_j = y_{1j}c_1 + y_{2j}c_2 + \dots + y_{mj}c_m, \quad m+1 \leq j \leq n \quad (2.2.5)$$

进一步,读者不难自行验证问题(2.1.2)(在最优值相等且最优解一一对应的意义下)等价于如下既约(reduced)线性规划

$$\begin{aligned} \text{minimize} \quad & r_{m+1}x_{m+1} + \dots + r_nx_n + z_0 \\ \text{subject to} \quad & (x_1 =) y_{10} - \sum_{j=m+1}^n y_{1j}x_j \geq 0 \\ & \vdots \\ & (x_m =) y_{m0} - \sum_{j=m+1}^n y_{mj}x_j \geq 0 \\ & x_{m+1} \geq 0, \dots, x_n \geq 0 \end{aligned} \quad (2.2.6)$$

其中 $r_j = c_j - z_j$ 是既约费用系数(reduced cost coefficients),或者称为相对费用系数(relative cost coefficients).显然基变量的既约费用系数是零.需要注意的是,这里的既约线性规划问题(2.2.6)是待求解问题(2.1.2)相对于当前基本可行解的一个等价表述.

定理 2.2.1 (最优性判别) 在基本可行解 \mathbf{x} (设基变量指标为 $1, 2, \dots, m$)处,如果对所有非基变量指标 $j = m+1, \dots, n$,有 $r_j = c_j - z_j \geq 0$ 成立,则该解是最优的.

证明 基本可行解 \mathbf{x} 对应着既约线性规划问题(2.2.6).如果对于所有非基变量指标 $j = m+1, \dots, n$ 有 $r_j = c_j - z_j \geq 0$,则由变量的非负性可知, z_0 是既约线性规划最优目标值的一个下界.此外,零向量是该既约线性规划的一个可行解,对应的目标函数值为 z_0 ,这说明下界可以取到,从而零向量是该既约线性规划的最优解.于是,基本可行解 \mathbf{x} 是最优的. ■

2.2.2 基本可行解的改进

对于当前基本可行解 \mathbf{x} (基变量指标为 $1, 2, \dots, m$),设非基变量指标 q 使得 $r_q = \min\{r_i : i = m+1, \dots, n\}$.在2.2.1小节中已经看到, $r_q \geq 0$ 蕴含着最优性.否则, $r_q = c_q - z_q < 0$,回到问题(2.2.6),不难发现,若固定 \mathbf{x} 的其他变量而只让 x_q 从0开始逐渐增加,必然导致该问题的目标函数值相应减小 $|r_q| x_q$.

令 $\mathbf{y}_q = (y_{1q}, y_{2q}, \dots, y_{mq})^\top$.如果 $\mathbf{y}_q \leq \mathbf{0}$,则 x_q 增加至无穷大仍然是问题(2.2.6)的可行解,相应的目标函数值趋向负无穷大,这说明该问题无界.否则,至少有一个指标 $i \in \{1, 2, \dots, m\}$,使得 $y_{iq} > 0$,此时 x_q 若过度增加则不再是问题(2.2.6)的可行解,这是因为 $(x_i =) y_{i0} - y_{iq}x_q \geq 0$ 等价于要求 $x_q \leq y_{i0}/y_{iq}$.由于使 $y_{iq} > 0$ 的指标 i 未必唯一,所以要求

$$x_q \leq \min \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0, i = 1, 2, \dots, m \right\} \quad (2.2.7)$$

设上式最小值在指标 p 处取到, 即 $y_{p0}/y_{pq} = \min\{y_{i0}/y_{iq} : y_{iq} > 0, i = 1, 2, \dots, m\}$. 因问题(2.2.6)的目标函数随 x_q 增加而递减, 且 x_q 有如式(2.2.7)所示上界, 故置 x_q 为这一正的上界, 即 $x_q = y_{p0}/y_{pq}$ 不再是非基变量, 而 x_1, x_2, \dots, x_m 按照 $x_i = y_{i0} - y_{iq}x_q$ 的方式相应变动, 且 $x_p = y_{p0} - y_{pq}x_q = 0$, x_p 不再是基变量. 此时称非基变量 x_q 进基, 而基变量 x_p 出基. 用 x_q 替换 x_p , 这样便得到一个相邻的新的基本可行解.

为顺利进入下一次迭代, 还需要给出与新的基本可行解对应的规范形, 注意到新旧基本可行解的继承性(即只有两个变量不一样), 对表 2.2.1 进行一系列初等行变换可以求得新的规范形. 具体地, 给第 p 行除以 y_{pq} 使得在第 p 个方程中 x_q 的系数为 1; 然后, 从其余的每一行减去第 p 行的适当倍数, 使得 x_q 在其他方程中的系数为零. 执行这些行变换后, 表格的第 q 列除过第 p 个元素(等于 1)外, 其余元素均为零. 记新规范形的系数为 y'_{ij} , 则

$$y'_{ij} = \begin{cases} y_{ij} - \frac{y_{pi}}{y_{pq}}y_{iq}, & i \neq p \\ \frac{y_{pi}}{y_{pq}}, & i = p \end{cases} \quad (2.2.8)$$

式(2.2.8)是线性规划单纯形法中频繁出现的转轴公式, 元素 y_{pq} 称为转轴元(pivot element).

结束本小节之前, 我们给一个经济解释, 说明在单纯形法中扮演重要角色的既约费用系数 $r_j = c_j - z$, 为什么也被称为相对费用系数. 这里将问题(2.1.2)解释为配餐问题, 其中营养需求必须精确满足. A 的列给出了 1 单位特定食物的营养等价量. 对于给定的基(假定是 A 的前 m 列), 对应的规范形的数据说明如何由基中食物的线性组合来构造任一种食物(或更精确地说是任一种食物的营养成分). 例如胡萝卜不在基中, 我们能够利用规范形的数据, 将基中的食物进行适当的线性组合, 构造一种合成胡萝卜, 它的营养等效于胡萝卜.

为了检验由当前基表示的解是否最优, 考虑某一种不在基中的食物, 假定为胡萝卜, 并确定让其进基是否会更有利. 考虑胡萝卜的费用与合成胡萝卜的费用. 如果胡萝卜是食物 j , 则 1 单位胡萝卜的费用是 c_j ; 另一方面, 1 单位合成胡萝卜的费用 $z_j = \sum_{i=1}^m c_i y_{ij}$. 如果 $r_j = c_j - z_j \leq 0$, 则用真的胡萝卜代替合成胡萝卜更有利. 因此, 应该让胡萝卜进基. 需要注意的是: 这里的合成费用 z_j 是相对于当前的基, 它会随着基的改变而发生变化.

一般地, 可将每个 z_j 看作由当前基中的食物构造 1 单位第 j 种合成食物(a_j 所代表)的费用. 每列的直接价格与合成价格之差决定了该列是否应该进基.

2.2.3 计算过程

2.2.2 小节的理论和诸多技术建立了详细推演单纯形法的必要条件. 这里给出完整的计算过程, 并用几个例子进行说明.

首先将基本可行解对应的规范形数据列出, 并在底端附加上既约费用系数和当前基本可行解的**目标值的相反数**, 如表 2.2.2 所列. 称表 2.2.2 是与该基本可行解对应的单纯形表(simplex tableau). 与该表对应的基本解

$$x_i = \begin{cases} y_{i0}, & 1 \leq i \leq m \\ 0, & m+1 \leq i \leq n \end{cases}$$

对应的目标值是 z_0 . 假设其是可行的, 即 $y_{i0} \geq 0$, $i=1, 2, \dots, m$. 既约费用系数 r_j 表明 x_j 进基后, 目标值增大还是减小. 如果既约费用系数全是非负的, 则表明解已经是最优的. 如果有某些是负的, 则让相应的分量进基可以改善目标(在非退化假设下). 若有多于一个的既约费用系数是负的, 则在确定转轴列时可以选取它们中的任何一个. 通常的做法是选取既约费用系数最小的.

表 2.2.2 单纯形表(其中基变量依次为 x_1, x_2, \dots, x_m)

x_1	...	x_p	...	x_m	x_{m+1}	x_{m+2}	...	x_q	...	x_n	$B^{-1}b$
1	...	0	...	0	$y_{1,m+1}$	$y_{1,m+2}$...	y_{1q}	...	y_{1n}	y_{10}
					⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	...	1	...	0	$y_{p,m+1}$	$y_{p,m+2}$...	y_{pq}	...	y_{pn}	y_{p0}
					⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	...	0	...	1	$y_{m,m+1}$	$y_{m,m+2}$...	y_{mq}	...	y_{mn}	y_{m0}
r^T	0	...	0	...	0	r_{m+1}	r_{m+2}	...	r_q	...	r_n
											$-z_0$

下面深入讨论单纯形表的最后一行. 将 z 看作额外的变量, 将

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n - z = 0$$

看作另一个方程, 其与原来的等式约束一起形成增广系统. 增广系统的基本解有 $m+1$ 个基变量, 但要求 z 总是基变量. 因此, 与 z 对应的列总是 $(0, 0, \dots, 0, -1)^T$, 所以可以忽略不写. 这样, 开始时将 c_j 和右端项 0 组成的行向量附加到规范形的阵列后表示附加方程. 利用标准转轴运算将该行中与基变量对应的元素化为零, 这等价于将附加方程表示为

$$r_{m+1} x_{m+1} + r_{m+2} x_{m+2} + \dots + r_n x_n - z = -z_0 \quad (2.2.9)$$

则式(2.2.9)与式(2.2.4)是等价的, 因此所得 r_j 为既约费用系数. 这样, 最后一行可从 $(c^T, 0)$ 开始, 通过初等行变换将与基变量对应的元素化为零后, 在运算上与其他行是一样的.

在选取第 q 列进行转轴后, 对第 q 列的正元素 y_{iq} , $i=1, 2, \dots, m$, 计算比值 y_{i0}/y_{iq} , 然后选取比值最小的对应元素 y_{pq} . 以该元素转轴既能保持解的可行性, 同时又能使(假设非退化)目标函数的值减小. 如果有多个元素取到最小值, 则可利用取到最小值的任一元素. 如果该列不存在正元素, 则问题是无界的. 当以 y_{pq} 作为转轴元更新整个表格, 并用像处理其他所有行(除第 q 行外)的方式处理完最后一行后, 表格右下角的元素是新基本可行解处目标值的相反数, 最后一行的其余元素是与新基对应的既约费用系数.

这样, 我们可以给出算法的伪码, 即算法 2.2.1(Algorithm 2.2.1). 通过前面的推演, 从本质上已经说明该方法可以求解问题(再一次假设非退化, 即所有基本可行解都是非退化的). 仅当最优化满足或发现问题无界时, 过程才会终止. 在给定的基本可行解处, 如果两个条件都不成立, 则目标值可以严格减小. 因为基本可行解是有限的, 而目标值逐次严格减小, 所以基不会重复, 故算法必在满足两个终止条件之一的某个基处终止.

Algorithm 2.2.1 Simplex method for linear programming problem (2.1.2)

```

1: Initially form a tableau corresponding to a basic feasible solution;
2: find the reduced cost coefficients with row reduction;
3: while 1 do
4:   if each  $r_j \geq 0$  then
5:     return the current basic feasible solution is  $x^*$ .
6:   else
7:     select  $q$  such that  $r_q < 0$  to determine which nonbasic variable is to become basic;
8:     calculate the ratio  $y_{i0}/y_{iq}$  for  $y_{iq} > 0, i=1, \dots, m$ ;
9:     if no  $y_{iq} > 0$  then
10:    return the problem is unbounded.
11:   else
12:     select  $p$  as the index  $i$  corresponding to the minimum positive ratio;
13:     pivot on the  $(p, q)$ -th element, updating all rows including the last;
14:   end if
15: end if
16: end while

```

例 2.2.2 (单纯形法) 用单纯形法求解下面问题

$$\begin{aligned}
 & \text{maximize} \quad 3x_1 + x_2 + 3x_3 \\
 & \text{subject to} \quad 2x_1 + x_2 + x_3 \leq 2 \\
 & \quad \quad \quad x_1 + 2x_2 + 3x_3 \leq 5 \\
 & \quad \quad \quad 2x_1 + 2x_2 + x_3 \leq 6 \\
 & \quad \quad \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

为了使用单纯形法, 必须将问题转化成标准形. 将目标函数乘以 -1 使极大化变成极小化, 然后引入 3 个松弛变量 x_4, x_5, x_6 , 得初始表格/第 1 张表格

a_1	a_2	a_3	a_4	a_5	a_6	b
2	1	1	1	0	0	2
1	2	3	0	1	0	5
2	2	1	0	0	1	6
r^T	-3	-1	-3	0	0	0

该问题已经是 3 个松弛变量作为基变量的规范形. 因为松弛变量的费用是零, 此时有 $r_j = c_j - z_j = c_j$. 选择转轴列的准则说明前 3 列中的任一列进基都将产生目标值减小的解. 在这些列中, 通过计算比值 y_{i0}/y_{ij} , 并选取正比值中最小的, 可以确定恰当的转轴元. 3 个允许的转轴元在表格中均加框来标记. 仅需确定一个允许转轴元, 常规上讲, 没有必要计算出所有允许的转轴元. 然而, 对于此种手算规模的问题, 希望检查所有可允许的转轴元, 然后选取一个所要求的除法运算最少的(在当前转轴中). 对此例我们选取 1 作为转轴元, 执行转轴运算后, 得第 2 张表格

x_1	x_2	x_3	x_4	x_5	x_6	$B^{-1}b$
2	1	1	1	0	0	2
-3	0	1	-2	1	0	1
-2	0	-1	-2	0	1	2
r^T	-1	0	-2	1	0	2

注意,目标函数(原始值的负值)已经从0减小到-2.再次以1为转轴元,转轴后得第3张表格

x_1	x_2	x_3	x_4	x_5	x_6	$B^{-1}b$
5	1	0	3	-1	0	1
-3	0	1	-2	1	0	1
-5	0	0	-4	1	1	3
r^T	-7	0	-3	2	0	4

目标值现在已经减小到-4,可以在第1列或第4列进行转轴,得第4张表格

x_1	x_2	x_3	x_4	x_5	x_6	$B^{-1}b$
1	$\frac{1}{5}$	0	$\frac{3}{5}$	$-\frac{1}{5}$	0	$\frac{1}{5}$
0	$\frac{3}{5}$	1	$-\frac{1}{5}$	$\frac{2}{5}$	0	$\frac{8}{5}$
0	1	0	-1	0	1	4
r^T	0	$\frac{7}{5}$	0	$\frac{6}{5}$	$\frac{3}{5}$	$\frac{27}{5}$

因为最后一行没有负元素,从而与第4张表格对应的解是最优的.这样 $x_1 = 1/5, x_3 = 8/5, x_6 = 4, x_2 = x_4 = x_5 = 0$ 是最优解,对应的(负)目标值是-27/5.

2.2.4 退化与循环

前面介绍的单纯形法总是强调一个前提,这就是非退化假设,即要求问题的所有基本可行解是非退化的.因为基本可行解是有限的,此假设的目的是保证转轴后基本可行解处的目标值比转轴前的解处的目标值严格减小,从而使算法中出现的基本可行解不会重复,这样单纯形法在有限步迭代后必然终止.

例 2.2.3 (退化的极点) 考虑

$$\begin{array}{ll}
 \text{minimize} & -x_1 - 2x_2 - 3x_3 \\
 \text{subject to} & x_1 + 2x_3 \leq 3 \\
 & x_2 + 2x_3 \leq 2 \\
 & x_1, x_2, x_3 \geq 0
 \end{array}
 \quad
 \begin{array}{ll}
 \text{minimize} & -x_1 - 2x_2 - 3x_3 \\
 \text{subject to} & x_1 + 2x_3 \leq 2 \\
 & x_2 + 2x_3 \leq 2 \\
 & x_1, x_2, x_3 \geq 0
 \end{array}$$

该问题的几何直观如图 2.2.1 所示.其中前一个问题如图 2.2.1(a) 所示,是非退化的,每个极点是 3 个平面的交点;后一个问题如图 2.2.1(b) 所示,标出的极点是 4 个平面的交点,是退化的,这个点对应标准形问题的基本可行解 $(0, 0, 1, 0, 0)^T$,这里 x_3 和其余任何一个变量都可作

为基变量,该基本可行解与多个基对应.对于标准形而言,当一个基本可行解与一个基对应时,是非退化的;当与多个基对应时,是退化的.

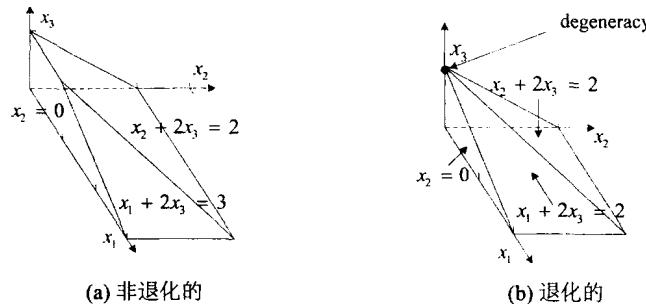


图 2.2.1 退化的极点/基本可行解

例 2.2.4 (退化步) 假设有如下极小化问题的单纯形表

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$B^{-1} b$
0	2	1	-3	0	2	0	0	1
0	-1	0	3	1	-1	0	0	4
1	0	0	0	0	2	0	0	3
0	0	0	1	0	-1	1	0	2
0	1	0	-1	0	0	0	1	0
r^1	0	-5	0	-4	0	-1	0	-6

这里的基变量 $x_8 = 0$, 所以对应的基本可行解是退化的. 若选 x_4 或 x_6 进基, 则得到的是非退化步, 即转轴后目标函数值严格减小; 若选 x_2 进基, 则要求 x_8 出基, 得到的是退化步 (degenerate step), 即转轴后目标函数值保持不变.

上例说明,对于退化问题,在单纯形法的迭代过程中,有可能出现退化的基本可行解;此时,有可能出现退化步;这样,目标值将保持不变,新的基本可行解也是退化的.可以想像,该过程可能会持续若干步,最终又回到初始的退化解. 称这种现象是循环 (cycle). Beale 给出了一个单纯形法产生循环的例子,见习题 2.21. 该例如果采用既约费用系数最小者进基、最小正比率检验中最小下标者出基,则经过 6 次转轴迭代后回到初始单纯形表.

为了避免求解退化问题时出现这种循环, Charnes 于 1952 年提出了摄动法, Dantzig, Orden 和 Wolfe 于 1954 年提出了字典序法. Bland 于 1977 年提出了一种简单的新规则确定进基出基变量, 称为 Bland 法则. 该法则可避免用单纯形法在求解退化问题时出现循环. Bland 法则的做法是: 在迭代过程中,如果有多个既约费用系数是负的,则选取下标最小的既约费用系数对应的变量进基;如果最小正比率在多个指标处取到,取下标最小者对应的变量出基. 实际应用中,很少出现循环现象,但这些简单的反循环规则在保证问题的收敛性理论方面是非常成功的. 为了安全起见,许多程序都会采取某种防止循环的措施.

2.2.5 初始基本可行解

单纯形法是从一个基本可行解到另一个基本可行解的迭代法,从而需要从一个基本可行解开始.定理2.1.1保证问题只要可行,就必定存在基本可行解.有些线性规划问题的基本可行解很容易得到,比如,约束形式如 $Ax \leq b, x \geq 0$ 且其中 $b \geq 0$ 的问题.引入的松弛变量自动成为相应标准形问题的基本可行解,从而可以启动单纯形法,比如例2.2.2就是这种类型的.但是一般情形下,找一个初始基本可行解没有这么直观.为此,这里介绍一种方法:预先构造一个具有显见基本可行解的辅助线性规划问题,利用刚介绍的单纯形法求解这个辅助问题,以得到初始基本可行解.

考虑如下线性系统

$$Ax = b, \quad x \geq 0 \quad (2.2.10)$$

其中 $b \geq 0$ (注意:若有某方程右端项 $b_i < 0$,则该方程两边同时乘以-1).为了求得系统(2.2.10)的基本可行解,考虑辅助问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^n, y \in \mathbb{R}^m}{\text{minimize}} \quad \sum_{i=1}^m y_i \\ & \text{subject to} \quad Ax + y = b \\ & \quad x \geq 0, \quad y \geq 0 \end{aligned} \quad (2.2.11)$$

其中 $y = (y_1, y_2, \dots, y_m)^\top$ 是人为添加的变量,称为人工变量(artificial variable). $x = 0, y = b$ 自动成为辅助线性规划问题(2.2.11)的基本可行解.又因为问题(2.2.11)有下界0,所以用单纯形法求解(2.2.11)时,必能得到最优解(\hat{x}, \hat{y}),最优值 $\hat{z} = \sum \hat{y}_i$.则有如下几种情况出现:

(a) 若 $\hat{z} > 0$,则系统(2.2.10)没有可行解.

(b) 若 $\hat{z} = 0$,则 \hat{x} 是系统(2.2.10)的可行解,且是潜在的基本可行解.若基变量中没有人工变量,则 \hat{x} 恰是系统(2.2.10)的基本可行解;否则,可用下面方法驱赶所有人工变量出基.具体地,设 y_q 为基变量,若当前单纯形表第 q 行的前 n 个数据(与原始变量 x 对应)全为零,则系统(2.2.10)中第 q 个方程是冗余的,可直接删除;否则,以任一非零元为转轴元转轴,得辅助问题的一个新的最优基本可行解,且基变量中少了一个人工变量.重复该过程,必得到与 \hat{x} 对应的基.

关于转轴的形式化描述见习题2.15和习题2.16.作为额外的收获,这实际上给出了一种判定线性系统(2.2.10)有无解的方法.利用上述方法来启动单纯形法可以求解一般的线性规划问题,称所得方法为两阶段法(two-phase method).

概括地说,方法由第I阶段和第II阶段组成.其中第I阶段如上引入人工变量,构造辅助问题,然后利用单纯形法求得一个基本可行解(或者确定问题没有可行解);第II阶段即从第I阶段得到的基本可行解出发,利用单纯形法求解原问题.当然,第I阶段中仅需对那些不含松弛变量的方程引入人工变量即可.

例2.2.5(两阶段法) 求解问题

$$\begin{aligned}
 & \text{minimize} && x_1 - x_2 \\
 & \text{subject to} && -x_1 + 2x_2 + x_3 = 2 \\
 & && -4x_1 + 4x_2 - x_3 = 4 \\
 & && -5x_1 + 6x_2 = 6 \\
 & && x_1 - x_3 = 0 \\
 & && x_1, x_2, x_3 \geq 0
 \end{aligned}$$

构造辅助问题,即

$$\begin{aligned}
 & \text{minimize} && y_1 + y_2 + y_3 + y_4 \\
 & \text{subject to} && -x_1 + 2x_2 + x_3 + y_1 = 2 \\
 & && -4x_1 + 4x_2 - x_3 + y_2 = 4 \\
 & && -5x_1 + 6x_2 + y_3 = 6 \\
 & && x_1 - x_3 + y_4 = 0 \\
 & && x_1, x_2, x_3 \geq 0, \quad y_1, y_2, y_3, y_4 \geq 0
 \end{aligned}$$

易得第1张单纯形表为

x_1	x_2	x_3	y_1	y_2	y_3	y_4	
-1	2	1	1	0	0	0	2
-4	4	-1	0	1	0	0	4
-5	6	0	0	0	1	0	6
1	0	-1	0	0	0	1	0
r^T	9	-12	1	0	0	0	-12

一次转轴迭代后,得到最优单纯形表

x_1	x_2	x_3	y_1	y_2	y_3	y_4	
$-\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	1
-2	0	-3	-2	1	0	0	0
-2	0	-3	-3	0	1	0	0
1	0	-1	0	0	0	1	0
r^T	3	0	7	6	0	0	0

因为辅助问题的最优值为0,故原始问题有可行解.但人工变量 y_2, y_3, y_4 没有出基,先采用转轴运算迫使 y_2 出基、 x_1 进基(也可选 x_3 进基,转轴时不用考虑既约费用系数)可得

x_1	x_2	x_3	y_1	y_2	y_3	y_4	
0	1	$\frac{5}{4}$	1	$-\frac{1}{4}$	0	0	1
1	0	$\frac{3}{2}$	1	$-\frac{1}{2}$	0	0	0
0	0	0	-1	-1	1	0	0
0	0	-$\frac{5}{2}$	-1	$\frac{1}{2}$	0	1	0

再考虑让 y_3 出基, 注意相应行中原变量 x_1, x_2, x_3 对应的系数都为 0, 故该行对应的方程是冗余的, 直接删除. 再让 y_4 出基、 x_3 进基, 则有

x_1	x_2	x_3	y_1	y_2	y_4
0	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$
1	0	0	$\frac{2}{5}$	$-\frac{1}{5}$	$\frac{3}{5}$
0	0	1	$\frac{2}{5}$	$-\frac{1}{5}$	$-\frac{2}{5}$

这样得到了原问题的基本可行解 $(0, 1, 0)^T$. 进一步, 删除 y_1, y_2, y_3 所在 3 列可以得到与这个基本可行解对应的规范形, 易得初始表格和单纯形表分别为

x_1	x_2	x_3		x_1	x_2	x_3	
0	1	0	1		0	1	0
1	0	0	0		1	0	0
0	0	1	0		0	0	1
c^T	1	-1	0		r^T	0	0

这也是最优单纯形表, 得到原始问题的解 $x = (0, 1, 0)^T$.

2.2.6 修正单纯形法

前面与单纯形法相关的转轴变换以表格的形式给出, 这样便于我们将注意力放在单个元素的表现形式上, 具有易于辨别、直观等特点. 然而大量经验表明, 应用单纯形法求解从各领域涌现出的问题(其中的 n 和 m 取各种各样的值)时, 通常会在大约 $2m \sim 3m$ 次转轴后找到最优解. 这样, 特别是当 m 比 n 小得多时, 整个寻优过程仅在很少的一部分列发生转轴. 没有显式用到的列很多, 对这些列的计算显得有些浪费. 修正单纯形法(revised simplex method)是一种避免无谓计算, 从而减小单纯形法所需计算量的实现方法. 事实上, 即使要求在所有列上转轴, 当 m 比 n 小得多时, 修正单纯形法也可以节省计算开销.

修正单纯形法基于单纯形表的矩阵表示. 这种矩阵表示不仅可以增强对单纯形法的理解, 而且在许多情况下, 还会具有相当大的计算获益. 此外, 矩阵表示也是讨论线性规划的对偶和其他与线性规划相关主题的一种自然背景.

假设知道一个基本可行解和与之相应的基 B . 与前文一致, 假定 B 由 A 的前 m 列组成. 然后将 A, x 和 c 相应地剖分为 $A = [B \ N]$, $x = (x_B^T, x_N^T)^T$, $c = (c_B^T, c_N^T)^T$, 线性规划标准形问题变成

$$\begin{aligned} & \text{minimize} && c_B^T x_B + c_N^T x_N \\ & \text{subject to} && B x_B + N x_N = b \\ & && x_B \geq 0, x_N \geq 0 \end{aligned} \tag{2.2.12}$$

在该表示中, 令 $x_N = \mathbf{0}$ 即可得到这个基本解的基变量的取值. 进一步, 对于 x_N 的任一取值, 由 $B x_B + N x_N = b$ 计算出 x_B 的必然取值为

$$x_B = B^{-1} b - B^{-1} N x_N \tag{2.2.13}$$

将表达式(2.2.13)代入目标函数,得

$$\begin{aligned} z &= \mathbf{c}_B^T (\mathbf{B}^{-1} \mathbf{b} - \mathbf{B}^{-1} \mathbf{N} \mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}) \mathbf{x}_N \end{aligned}$$

上式给出了标准形问题(2.2.12)的任一解用 \mathbf{x}_N 表示时的费用函数和相对费用向量(关于非基变量的) $\mathbf{r}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$, 利用该向量来确定哪个变量进基. 基于这些矩阵表示, 单纯形表可以表示成

$$\left[\begin{array}{c|c} \mathbf{B}^{-1} \mathbf{A} & \mathbf{B}^{-1} \mathbf{b} \\ \hline \mathbf{c}^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A} & -\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} \end{array} \right]$$

基于单纯形表的这种矩阵表示, 并定义 $\mathbf{B} \lambda^T = \mathbf{c}_B^T$ 的解 λ 为 Lagrange 乘子, 直接可以得到修正单纯形法的伪码描述, 即算法 2.2.2.

Algorithm 2.2.2 Revised simplex method for linear programming problem(2.1.2)

```

1: Initially get the inverse  $\mathbf{B}^{-1}$  of a current basis, and the current solution  $\mathbf{y}_0 = \mathbf{B}^{-1} \mathbf{b}$ ;
2: while 1 do
3:   calculate the simplex multiplier  $\lambda^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ ;
4:   calculate the reduced cost vector  $\mathbf{r}_N^T = \mathbf{c}_N^T - \lambda^T \mathbf{N}$ ;
5:   find  $q$  such that  $r_q = \min\{r_i : x_i \text{ is nonbasic}\}$ ;
6:   if  $r_q \geq 0$  then
7:     return the current basic feasible solution is  $\mathbf{x}^*$ 
8:   else
9:     determine vector  $\mathbf{a}_q$  to enter the basis;
10:    calculate  $\mathbf{y}_q = \mathbf{B}^{-1} \mathbf{a}_q$ ;
11:    if  $\mathbf{y}_q \leq 0$  then
12:      return the problem is unbounded.
13:    else
14:      calculate the ratio  $y_{i0}/y_{iq}$  for  $y_{iq} > 0, i = 1, \dots, m$ ;
15:      select  $p$  as the index corresponding to the minimum positive ratio;
16:      update  $\mathbf{B}^{-1}$ , the current simplex multiplier  $\lambda$ , and the current solution  $\mathbf{B}^{-1} \mathbf{b}$ ;
17:    end if
18:  end if
19: end while

```

要有效地实现修正单纯形法, 基矩阵的逆必须要有有效的更新公式. 为此考查相邻两次迭代中基的关系, 不妨设 $\mathbf{B} = [\mathbf{a}_1 \dots \mathbf{a}_p \dots \mathbf{a}_m]$, 对应的规范形见式(2.2.2). 此时以 y_{pq} 为转轴元进行转轴, 即 \mathbf{a}_q 进基、 \mathbf{a}_p 出基. 令 $\mathbf{E}_{pq} = [\mathbf{e}_1 \dots \mathbf{e}_{p-1} \mathbf{v} \mathbf{e}_{p+1} \dots \mathbf{e}_m]$, 其中 \mathbf{v} 的第 i 个分量定义为 $v_i = -\frac{y_{iq}}{y_{pq}}$, $i \neq p$ 且 $v_p = \frac{1}{y_{pq}}$, 即初等矩阵 \mathbf{E}_{pq} 的第 p 列完全由转轴列 \mathbf{y}_q 的元素确定. 以下结论成立.

定理 2.2.2 (数据的更新) 转轴后得到的新基

$$\hat{\mathbf{B}} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_{p-1} \ \mathbf{a}_q \ \mathbf{a}_{p+1} \ \cdots \ \mathbf{a}_m]$$

满足 $\hat{\mathbf{B}}^{-1} = \mathbf{E}_{pq}\mathbf{B}^{-1}$ ；转轴后的单纯形乘子 $\hat{\lambda}^T = \lambda^T + \frac{r_q}{y_{pq}}\mathbf{u}^p$ ，其中 \mathbf{u}^p 表示 \mathbf{B}^{-1} 的第 p 行。

证明 令 $\mathbf{C} = \mathbf{B}^{-1}\hat{\mathbf{B}}$ ，则 $\mathbf{C} = [\mathbf{e}_1 \cdots \mathbf{e}_{p-1} \ \mathbf{y}_q \ \mathbf{e}_{p+1} \cdots \mathbf{e}_m]$ 。易见 $\mathbf{C}^{-1} = \mathbf{E}_{pq}$ 。从而 $\hat{\mathbf{B}}^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1} = \mathbf{E}_{pq}\mathbf{B}^{-1}$ 。

此外，由单纯形乘子的定义有

$$\begin{aligned}\hat{\lambda} &= (\mathbf{c}_B^T + (0, \dots, 0, c_q - c_p, 0, \dots, 0))\mathbf{E}_{pq}\mathbf{B}^{-1} \\ &= \mathbf{c}_B^T\mathbf{E}_{pq}\mathbf{B}^{-1} + \left(0, \dots, 0, \frac{c_q - c_p}{y_{pq}}, 0, \dots, 0\right)\mathbf{B}^{-1} \\ &= \mathbf{c}_B^T(\mathbf{I} - [\mathbf{0} \ \cdots \ \mathbf{e}_p - \mathbf{v} \ \cdots \ \mathbf{0}])\mathbf{B}^{-1} + \left(0, \dots, \frac{c_q - c_p}{y_{pq}}, \dots, 0\right)\mathbf{B}^{-1} \\ &= \lambda^T + \left(0, \dots, -c_p + \mathbf{c}_B^T\mathbf{v} + \frac{c_q - c_p}{y_{pq}}, \dots, 0\right)\mathbf{B}^{-1}\end{aligned}$$

进一步，由向量 \mathbf{v} 和既约费用系数 r_q 的定义，有

$$\begin{aligned}&-c_p + \mathbf{c}_B^T\mathbf{v} + \frac{c_q - c_p}{y_{pq}} \\ &= -\frac{1}{y_{pq}}(c_1y_{1q} + c_2y_{2q} + \cdots + c_py_{pq} + \cdots + c_my_{mq}) + \frac{c_q}{y_{pq}} \\ &= \frac{r_q}{y_{pq}}\end{aligned}$$

因为给任何矩阵左乘 \mathbf{E}_{pq} 等价于给第 $i \neq p$ 行加上第 p 行的 v_i 倍，给第 p 行乘以 v_p ；此外，给 λ^T 加上 \mathbf{B}^{-1} 的第 p 行的 r_q/y_{pq} 倍即可得到 $\hat{\lambda}^T$ 。这显然等价于对修正单纯形表

变 量	\mathbf{B}^{-1}	\mathbf{x}_B	\mathbf{y}_q
i_1		y_{10}	y_{1q}
\vdots		\vdots	\vdots
i_p		y_{p0}	$\boxed{y_{pq}}$
\vdots		\vdots	\vdots
i_m		y_{m0}	y_{mq}
λ^T	$\lambda_1 \ \cdots \ \lambda_m$	z_0	$-r_q$

执行以 y_{pq} 为转轴元的转轴运算，即可得与新基对应的数据。下面用例子说明修正单纯形法的计算过程。

例 2.2.6 (修正单纯形法) 用修正单纯形法求解例 2.2.2，这里列出所给向量供参考

a_1	a_2	a_3	a_4	a_5	a_6	b
2	1	1	1	0	0	2
1	2	3	0	1	0	5
2	2	1	0	0	1	6

目标函数由 $c = (-3, -1, -3, 0, 0, 0)^T$ 确定. 初始基本可行解所对应的单纯形乘子和目标值分别是 $\lambda^T = (0, 0, 0)$, $B^{-1} = (0, 0, 0)$, $z = 0$. 然后计算 $r_N^T = c_N^T - \lambda^T N = (-3, -1, -3)$. 选取 a_1 进基, 给 a_1 左乘 B^{-1} 得到 y_1 , 这样有

变 量	B^{-1}			x_B	y_1
4	1	0	0	2	2
5	0	1	0	5	1
6	0	0	1	6	2
λ^T	0	0	0	0	3

像往常一样, 计算完正比值后, 选取所标记的元素为转轴元, 转轴后得到新表格

变 量	B^{-1}			x_B
1	$\frac{1}{2}$	0	0	1
5	$-\frac{1}{2}$	1	0	4
6	-1	0	1	4
λ^T	$-\frac{3}{2}$	0	0	-3

由 λ^T 和原始数据按公式 $r_j = c_j - \lambda^T a_j$ 计算后, 得 $r_2 = 1/2$, $r_3 = -3/2$, $r_4 = 3/2$. 选取 a_3 进基, 计算 $y_3 = B^{-1} a_3 = (1/2, 5/2, 0)^T$, 则有表格

变 量	B^{-1}			x_B	y_3
1	$\frac{1}{2}$	0	0	1	$\frac{1}{2}$
5	$-\frac{1}{2}$	1	0	4	5/2
6	-1	0	1	4	0
λ^T	$-\frac{3}{2}$	0	0	-3	$\frac{3}{2}$

利用所标记的转轴元转轴, 得到

变 量	B^{-1}			x_B
1	$\frac{3}{5}$	$-\frac{1}{5}$	0	$\frac{1}{5}$
3	$-\frac{1}{5}$	$\frac{2}{5}$	0	$\frac{8}{5}$
6	-1	0	1	4
λ^T	$-\frac{6}{5}$	$-\frac{3}{5}$	0	$-\frac{27}{5}$

现在 $r_2 = 7/5$, $r_4 = 6/5$, $r_5 = 3/5$. 因为所有的 r_i 都是非负的, 所以最优解 $\mathbf{x}^* = (1/5, 0, 8/5, 0, 0, 4)^T$.

2.2.7 单纯形法的效率

本小节讨论单纯形法的效率. 这里算法的效率指时间复杂度, 即算法求解给定问题所需要的时间. 有两种常用方式分析算法的复杂度. 一种是平均情况分析, 即求解典型问题需要多少时间. 这种研究方式从数学上讲很难, 通常只能基于经验或者数值实验来说明. 另一种是最坏情况分析, 即用算法求解最难的问题需要多少时间. 这种研究方式从数学上讲是可以处理的, 通常可以得到计算时间的一个上界. 该上界是问题规模的函数. 如果存在多项式函数作为算法计算时间的上界, 则称它是多项式时间 (polynomial-time) 算法; 否则, 称为指数时间 (exponential-time) 算法.

度量线性规划问题的常用量有: 约束的个数 m 和变量的个数 n ; 确定问题的数据个数 $m \times n$; 非零数据的个数; 把问题的数据输入计算机时所需二进制代码的长度, 即输入长度. 对于普通问题, 变量的个数通常起决定性作用; 对于大规模问题, 非零数据的个数通常起决定性作用. 算法所需要的时间通常依赖于: 迭代次数; 每次迭代的算术运算次数; 每次算术运算的时间 (依赖于硬件). 迭代次数通常起决定性作用.

几何上, 单纯形算法从多面体的一个顶点开始, 沿着多面体的边走到目标函数值得以改善的下一个顶点, 直至到达最优解为止. 虽然这个算法在实际中很有效, 在小心处理可能出现的“循环”后, 可以保证找到最优解, 但它的最坏情况可以很坏. Klee 和 Minty 构造出一族问题^[10], 利用单纯形法求解它们时会遍历所有的顶点. 为了获得感性认识, 我们先看 $n=3$ 的 Klee-Minty 问题, 即

$$\begin{aligned} & \text{maximize} && 4x_1 + 2x_2 + x_3 \\ & \text{subject to} && x_1 \leq 1 \\ & && 4x_1 + x_2 \leq 100 \\ & && 8x_1 + 4x_2 + x_3 \leq 10\,000 \\ & && x_1, x_2, x_3 \geq 0 \end{aligned} \tag{2.2.14}$$

其可行域见图 2.2.2, 它是一个稍微扭曲了的立方体, 其中每条边上的数字表示该边的长度. 利用 Dantzig 的单纯形法求解该问题时, 如果从 $\mathbf{x}^{(0)} = \mathbf{0}$ 开始, 算法将会遍历所有的顶点 (见习题 2.20). 一般的 Klee-Minty 问题为

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n 2^{n-i} x_i \\ & \text{subject to} && 2 \sum_{i=1}^{j-1} 2^{j-i} x_i + x_j \leq 100^{j-1}, \quad j = 1, 2, \dots, n \\ & && x_i \geq 0, \quad i = 1, 2, \dots, n \end{aligned}$$

它的可行域是超立方体 $\{\mathbf{x} \in \mathbb{R}^n : 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 100, \dots, 0 \leq x_n \leq 100^{n-1}\}$ 的稍微扭曲, 共有 2^n 个顶点. 单纯形法求解这类问题需要的计算时间是 $O(2^n)$. 当 $n=70$ 时, $2^{70} = 1.2 \times 10^{21}$. Klee-Minty 问题说明单纯形法的时间复杂度是指数的.

Klee-Minty 问题表明, 单纯形法虽然在实际应用中很有效, 但在理论上它还不是多项式

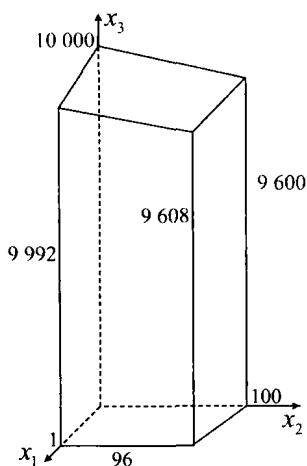


图 2.2.2 Klee-Minty 问题的可行域($n=3$)

时间算法,于是产生这样的问题:对于线性规划,能否找到多项式时间算法?事实上,一段时期内人们曾不能确定线性规划问题是 NP 完全问题,还是多项式时间可解的问题.基于线性规划基本定理,人们甚至一度相信线性规划问题不能用多项式时间的算法求解.

第一个在最坏情况具有多项式时间复杂度的线性规划算法由苏联青年数学家 Khachiyan 于 1979 年提出^[11],称为椭球法(ellipsoid method).事实上,椭球法最早由 Naum Shor 于 20 世纪 70 年代中期发明,并用于求解非线性规划.此后 Nemirovski (2003 年 von Neumann 运筹学理论奖得主) 和 Yudin 运用这种方法求解凸规划问题. Khachiyan 的贡献在于把该方法运用到线性规划上,并证明了它的多项式时间计算复杂度,从根本上改变了人们对线性规划的认识.在理论上,“椭球法”在最坏情况下所需要的计算量要比“单纯形法”增加得缓慢.但在实际应用上, Khachiyan 的椭球法

令人失望,根本无法与单纯形法的实际效率相匹敌.

理论上好,但实际表现差的椭球法激发了人们寻求求解线性规划的理论上好且实际表现也好的算法.之后,贝尔实验室的印度裔数学家 Karmarkar 于 1984 年提出了投影尺度法(又名 Karmarkar 算法)^[12].这是一种内点法,它摒弃了单纯形法的“边界趋近”观念,采用了“内部逼近”的路线.这是第一个在理论上和实际上都表现良好的算法,它的时间复杂度也是多项式的,且在大规模的实际问题中的表现可以与单纯形法相媲美.此后,内点法成为研究热点,很多内点法被提出来并进行分析.一个常用的内点法是 Mehrotra 于 1992 年提出的 predictor-corrector 法,它在实际应用中表现很出色.

线性规划的算法现状是:就线性规划的日常应用问题而言,如果算法的实现良好,基于单纯形法和内点法的算法之间的效率没有太大差别,只有在超大型线性规划中,顶点几乎成天文数字时,内点法才有机会领先单纯形法.

2.3 对偶

从分蛋糕难题开始.如何让 n 个人平分蛋糕,使得每个人都相信自己分得的一份至少为那块蛋糕的 $1/n$,这种方式称为无妒忌方法.美国科学家 Brains 和 Taylor 于 1995 年找到一个相当复杂的可行算法,但 $n=2$ 的情形很简单,只要采用“我分你选”的做法就可以实现公平的分配.从优化的角度:设甲分乙选,因为乙要先选,所以甲只能得到较少的一份;而甲来分,所以甲的优化目标是让这较少的一份达到最大,正好为 $1/2$.把这个思想运用到线性规划,每一个极小化的线性规划问题都存在一个线性表出的下界,让这个线性下界达到最大,便得到一个极大的线性规划问题,称为对偶问题.可以证明当目标函数的最优值有限时,二者是相等的.

对偶问题往往有一些经济上的解释,其变量也与单纯形法中的既约费用系数密切相关.这样,研究对偶性可以加深我们对单纯形法的理解,也有助于提出更多、更好的求解方法.另外,

同时从原始和对偶的观点考虑一个问题，经常会带来很多的计算效益和经济上的理解。

2.3.1 对偶问题

考虑线性规划问题(2.1.2)以及任意的 m 维向量 λ 和 n 维向量 $y \geq 0$ ，若 x 为问题(2.1.2)的可行解，必有 $c^T x + \lambda^T (b - Ax) - y^T x \leq c^T x$ 。从而可得上述线性规划问题的下界

$$\min_{x \in \mathbb{R}^n} \{ (c - A^T \lambda - y)^T x + \lambda^T b \}$$

读者不难验证这个下界为

$$\begin{cases} b^T \lambda, & c - A^T \lambda - y = 0 \\ -\infty, & c - A^T \lambda - y \neq 0 \end{cases}$$

若取这类下界中最大的一个，便得到如下线性规划问题

$$\begin{aligned} & \text{maximize} && b^T \lambda \\ & \text{subject to} && c - A^T \lambda - y = 0 \\ & && y \geq 0 \end{aligned}$$

进一步消去变量 y ，得到

$$\begin{aligned} & \text{maximize} && b^T \lambda \\ & \text{subject to} && \lambda^T A \leq c^T \end{aligned} \tag{2.3.1}$$

称这个问题为线性规划(2.1.2)的对偶(dual)问题。类似地，读者可以自行验证式(2.3.2)中左边线性规划的对偶问题正如右边所写。

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax \geq b \\ & && x \geq 0 \end{aligned} \quad \begin{aligned} & \text{maximize} && \lambda^T b \\ & \text{subject to} && \lambda^T A \leq c^T \\ & && \lambda \geq 0 \end{aligned} \tag{2.3.2}$$

其中 A 是 $m \times n$ 矩阵， x 和 c 是 n 维向量， b 和 λ 是 m 维向量。向量 x 是原始问题的变量，称为原始(primal)变量； λ 是对偶问题的变量，称为对偶变量。称问题对(2.3.2)为对称形式的对偶。

例 2.3.1 (配餐问题的对偶) 可将 2.1 节中的配餐问题看作是营养学家尝试选取食物组合以满足某些营养需求时，极小化费用所建立的优化问题，形如式(2.3.2)左边的问题。将它看作原始问题，可以给出其对偶问题的一种解释：想像有一个制药公司，生产被营养学家认为是重要营养的营养丸。制药公司试着说服营养学家购买营养丸来直接满足营养需求，而不是通过购买各种食物来满足。制药公司面对的问题是为营养丸确定单价，以便在与食物竞争的同时极大化收入。为了与真实的食物进行竞争，用从制药公司那里买来的纯粹营养成分合成的一单位食物 j 的费用 $\lambda^T a_j$ 不能比食物的市场价格 c_j 大。这样，对每种营养丸的定价必须满足 $\lambda^T a_j \leq c_j$ 。用矩阵形式表示为 $\lambda^T A \leq c^T$ 。因为要购买 b_i 单位的第 i 种营养，制药公司面临的是式(2.3.2)右边的问题，即配餐问题的对偶问题。

例 2.3.2 (运输问题的对偶) 运输问题是制造商在几个固定的源和目的地之间选择产品的运输模式，在满足需求的同时极小化费用。记与生产约束对应的对偶变量为 u_i ($i = 1, 2, \dots, m$)，与需求约束对应的为 v_j ($j = 1, 2, \dots, n$)。它的对偶问题是

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \\ & \text{subject to} && u_i + v_j \leq c_{ij}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \end{aligned}$$

现在来解释对偶问题,想像有一个自认头脑灵活的中间商,他来到制造商这里,在工厂现场(源)报价买入全部产品,然后贩卖到仓库(目的地).对制造商而言,也能完成其预定的运输目的,而中间商则是期望在这样的交易中谋取最大的差价(利润).假设中间商在 m 个源的单位报价(买入价)是 $-u_1, -u_2, \dots, -u_m$,在 n 个目的地的单位报价(卖出价)是 v_1, v_2, \dots, v_n . 中间商为了能从制造商那里顺利承包到该项任务,必须对所有的 i, j 满足 $u_i + v_j \leq c_{ij}$. 这是因为 $u_i + v_j$ 代表单位产品从源 i 到目的地 j 时中间商那里的差价,如果 $u_i + v_j > c_{ij}$, 制造商知道:任何一个人在源 i 以 $-u_i$ 的报价从制造商手里买入,再运输到目的地 j 卖出,都一定赚钱,聪明的制造商是不可能把这个甜头交给中间商的.这样,在这些约束条件下,中间商如何定价来极大化自己的利润,便得到了如上所给的对偶问题.

一般地,可以将任一线性规划问题化成标准形,从而得到对偶问题(2.3.1);也可以直接写出它的对偶问题.具体地,给定 $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$, 记 A 的第 i 行为 a^i , 第 j 列为 a_j . 将所给问题的约束分成 6 类,即普通约束“ $\geq, \leq, =$ ”和变量“ $\geq 0, \leq 0$, 无限制”,则互为对偶对的问题可表述为

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \begin{array}{ll} a^i x \geq b_i, & i \in M_1 \\ a^i x \leq b_i, & i \in M_2 \\ a^i x = b_i, & i \in M_3 \end{array} \\ & \begin{array}{ll} x_j \geq 0, & j \in N_1 \\ x_j \leq 0, & j \in N_2 \\ x_j \text{ 无限制,} & j \in N_3 \end{array} \end{array} \quad \begin{array}{ll} \text{maximize} & \lambda^T b \\ \text{subject to} & \begin{array}{ll} \lambda_i \geq 0, & i \in M_1 \\ \lambda_i \leq 0, & i \in M_2 \\ \lambda_i \text{ 无限制,} & i \in M_3 \end{array} \\ & \begin{array}{ll} \lambda^T a_j \leq c_j, & j \in N_1 \\ \lambda^T a_j \geq c_j, & j \in N_2 \\ \lambda^T a_j = c_j, & j \in N_3 \end{array} \end{array}$$

这里 M_i 和 N_i 分别表示约束和变量的指标集.下面是一个按这种一般性对偶原则写出的互为对偶问题的例子:

$$\begin{array}{ll} \text{minimize} & x_1 + 2x_2 + 3x_3 \\ \text{subject to} & \begin{array}{ll} -x_1 + 3x_2 = 5 \\ 2x_1 - x_2 + 3x_3 \geq 6 \\ x_3 \leq 4 \\ x_1 \geq 0 \\ x_2 \leq 0 \\ x_3 \text{ 无限制} \end{array} \end{array} \quad \begin{array}{ll} \text{maximize} & 5\lambda_1 + 6\lambda_2 + 4\lambda_3 \\ \text{subject to} & \begin{array}{ll} \lambda_1 \text{ 无限制} \\ \lambda_2 \geq 0 \\ \lambda_3 \leq 0 \\ -\lambda_1 + 2\lambda_2 \leq 1 \\ 3\lambda_1 - \lambda_2 \geq 2 \\ 3\lambda_2 + \lambda_3 = 3 \end{array} \end{array}$$

2.3.2 对偶定理

本小节将讨论原始、对偶问题之间更深层的关系,所给出的强对偶定理的证明方法依赖于点与闭凸集的分离定理,因此比前面的讨论复杂一些.之所以这样做是为了直接建立对偶理论的一般形式.随后给出另一种简单的证明方法.

本小节中,我们考虑具有标准形的原始问题(2.1.2)和它的对偶问题(2.3.1),且不必假定 A 是满秩的.读者不难发现,本小节开头对偶问题的引入,就已经蕴含了下面的定理,它给出了原始和对偶问题之间的重要关系.

定理 2.3.1 (弱对偶性) 如果 \hat{x} 和 $\hat{\lambda}$ 分别是问题(2.1.2)和问题(2.3.1)的可行解,则 $c^T \hat{x} \geq \hat{\lambda}^T b$.

证明 我们有 $\hat{\lambda}^T b = \hat{\lambda}^T A \hat{x} \leq c^T \hat{x}$, 最后一个不等式成立是因为 $\hat{x} \geq 0$ 且 $\hat{\lambda}^T A \leq c^T$. ■

定理 2.3.1 说明：由一个问题的可行解可得到另一个问题目标值的界. 如图 2.3.1 所示，原始问题任一可取的目标值不小于对偶问题任一可取的目标值. 由此我们有如下重要推论.



图 2.3.1 原始值与对偶值的关系

推论 1 设 \hat{x} 和 $\hat{\lambda}$ 分别是问题(2.1.2)和问题(2.3.1)的可行解, 如果有 $c^T \hat{x} = \hat{\lambda}^T b$, 则 \hat{x} 和 $\hat{\lambda}$ 分别是各自问题的最优解.

推论 2 如果问题(2.1.2)和问题(2.3.1)中某个问题无界, 则另一个问题没有可行解.

上面的推论 1 说明, 如果找到了原始-对偶问题的一对目标值相等的可行解, 则它们都是各自的最优解. 线性规划的强对偶定理表明逆也是真的, 事实上, 图 2.3.1 中的两个区域有一个公共点, 即不存在间隙(gap).

定理 2.3.2 (强对偶性) 如果问题(2.1.2)和问题(2.3.1)中一个有解, 则另一个也有解, 且二者的最优值相等.

证明 首先需要指出的是, 尽管原始-对偶问题的表述各异, 但是每个问题均可转换成标准形, 且原始-对偶的角色是可以互换的. 故证明命题时, 假定问题(2.1.2)有有限最优解, 然后证明对偶问题(2.3.1)具有一个目标值相同的解即可.

假设(2.1.2)有解 x^* , 最优值为 $z^* = c^T x^*$. 定义集合 $C = \{(r, v) \in \mathbb{R} \times \mathbb{R}^m : r = tz^* - c^T x, v = tb - Ax, x \geq 0, t \geq 0\}$. 将用反证法证明点 $(1, \mathbf{0})$ 不属于 C . 假设 $(1, \mathbf{0}) \in C$, 即存在 $t_0 \geq 0$, $x_0 \geq 0$ 使得

$$1 = t_0 z^* - c^T x_0, \quad t_0 b = Ax_0 \quad (2.3.3)$$

同时成立. 分两种情况来讨论.

(i) 当 $t_0 > 0$ 时, 由式(2.3.3)有 $b = A(x_0/t_0)$, 再加上 $x_0 \geq 0$, 可知 $\hat{x} = x_0/t_0$ 是问题(2.1.2)的可行解, 且此时 $z^* - c^T \hat{x} = 1/t_0 > 0$, 这与 x^* 的最优性矛盾.

(ii) 当 $t_0 = 0$ 时, 由式(2.3.3)有 $Ax_0 = \mathbf{0}$, 且有 $x_0 \geq 0, c^T x_0 = -1$. 易见 $\epsilon \geq 0$ 时 $x^* + \epsilon x_0$ 仍然可行, 且随着 ϵ 任意增大, 目标函数的值可以任意小. 这与 x^* 的最优性矛盾.

综合(i)和(ii)可知 $(1, \mathbf{0}) \notin C$. 现在, 由点与闭凸锥的分离定理, 即引理 7.3.5 可知, 存在超平面分离 $(1, \mathbf{0})$ 和 C , 即存在非零向量 $(s, \lambda^T) \in \mathbb{R}^{m+1}$ 和常数 γ 使得

$$s \cdot 1 + \lambda^T \mathbf{0} < \gamma \leq s \cdot r + \lambda^T v, \quad \forall (r, v) \in C$$

因为 $(0, \mathbf{0}) \in C$, 从而必有 $s < \gamma \leq 0$. 不失一般性, 可以假设 $s = -1$, 即对任意 $(r, v) \in C$, 有

$$\lambda^T v - r \geq \gamma \quad (2.3.4)$$

又因为 C 是锥, 如果存在 $(r, w) \in C$ 使得 $\lambda^T w - r < 0$, 则对充分大的 $\epsilon > 0, \epsilon(r, w) \in C$, 其必将违反不等式(2.3.4). 这样证明了存在 $\lambda \in \mathbb{R}^m$, 使得对所有 $(r, v) \in C$ 有 $\lambda^T v - r \geq 0$. 利用 C 的定义, 此即等价于对所有的 $x \geq 0, t \geq 0$ 有 $(c^T - \lambda^T A)x - tz^* + t\lambda^T b \geq 0$. 令 $t = 0$ 及 $x = e_i (i = 1, 2, \dots, m)$ 可得 $\lambda^T A \leq c^T$, 说明 λ 是对偶可行解. 令 $x = \mathbf{0}, t = 1$ 可得 $\lambda^T b \geq z^*$, 据此和定理 2.3.1 的推论 1 可判定 λ 是对偶问题的解. ■

2.3.3 对偶问题与单纯形法的关系

本小节利用单纯形法的特点证明强对偶定理, 这种证明方法的好处之一是: 一旦用单纯

形法求得原始问题的解后,易于得到对偶问题的解.考虑线性规划问题(2.1.2),假定基 \mathbf{B} 是最优基,即基本可行解 $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ 是最优解.我们将用 \mathbf{B} 确定对偶问题(2.3.1)的解.

将 $\mathbf{A} = [\mathbf{B} \quad \mathbf{N}]$, 则 $\mathbf{r}^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}$. 因为基本可行解 $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ 是最优的, 既约费用向量 \mathbf{r} 的每一个分量必须非负, 因此有 $\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} \leq \mathbf{c}_N^T$.

令 $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$. 下面说明如此定义的向量 $\boldsymbol{\lambda}$ 是对偶问题的解. 一方面, $\boldsymbol{\lambda}^T \mathbf{A} = (\boldsymbol{\lambda}^T \mathbf{B}, \boldsymbol{\lambda}^T \mathbf{N}) = (\mathbf{c}_B^T, \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}) \leq (\mathbf{c}_B^T, \mathbf{c}_N^T) = \mathbf{c}^T$, 所以 $\boldsymbol{\lambda}$ 是对偶问题的可行解. 另一方面, $\boldsymbol{\lambda}^T \mathbf{b} = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} = \mathbf{c}_B^T \mathbf{x}_B$, 这样对偶目标函数在 $\boldsymbol{\lambda}$ 处的值等于原始问题在基本可行解 $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ 处的值. 可将该讨论看作对偶定理的另一种证明.

定理 2.3.3 (与单纯形法的关系) 设线性规划问题(2.1.2)有最优基本可行解, 对应基是 \mathbf{B} , 则向量 $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ 是对偶问题(2.3.1)的最优解, 且两个问题的最优值是相等的.

现在开始讨论如何直接由原始问题最终的单纯形表得到对偶问题的解. 假定原始矩阵 \mathbf{A} 中嵌有 m 阶的单位矩阵, 比如添加 m 个松弛变量将不等式转化成等式时即如此. 则在最终的表格中, 初始的单位矩阵所在的位置即为矩阵 \mathbf{B}^{-1} . 进一步, 最后一行与这个单位矩阵对应的元素为 $\mathbf{c}_l^T - \mathbf{c}_B^T \mathbf{B}^{-1}$, 其中 \mathbf{c}_l 是 m 维向量, 代表初始单位矩阵的列所对应变量的费用系数. 从而将这些费用系数从最后一行的对应元素中减去, 即得 $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ 的负值. 特别地, 当 m 阶单位阵对应的变量为松弛变量时, $\mathbf{c}_l = \mathbf{0}$, 最后一行中在 \mathbf{B}^{-1} 下方的元素为单纯形乘子的相反数.

例 2.3.3 (单纯形乘子) 考虑原始问题

$$\begin{aligned} \text{minimize} \quad & -x_1 - 4x_2 - 3x_3 \\ \text{subject to} \quad & 2x_1 + 2x_2 + x_3 \leq 4 \\ & x_1 + 2x_2 + 2x_3 \leq 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

引入松弛变量, 可以利用单纯形法求解. 下面不加解释地给出求解过程中的单纯形表序列

	x_1	x_2	x_3	x_4	x_5	$\mathbf{B}^{-1}\mathbf{b}$
	2	2	1	1	0	4
	1	2	2	0	1	6
\mathbf{r}^T	-1	-4	-3	0	0	0

	x_1	x_2	x_3	x_4	x_5	$\mathbf{B}^{-1}\mathbf{b}$
	1	1	$\frac{1}{2}$	$\frac{1}{2}$	0	2
	-1	0	1	-1	1	2
\mathbf{r}^T	3	0	-1	2	0	8

	x_1	x_2	x_3	x_4	x_5	$\mathbf{B}^{-1}\mathbf{b}$
	$\frac{3}{2}$	1	0	1	$-\frac{1}{2}$	1
	-1	0	1	-1	1	2
\mathbf{r}^T	2	0	0	1	1	10

最优解 $x^* = (0, 1, 2)^T$. 相应的对偶问题是

$$\begin{aligned} & \text{maximize} && 4\lambda_1 + 6\lambda_2 \\ & \text{subject to} && 2\lambda_1 + \lambda_2 \leq -1 \\ & && 2\lambda_1 + 2\lambda_2 \leq -4 \\ & && \lambda_1 + 2\lambda_2 \leq -3 \\ & && \lambda_1 \leq 0, \lambda_2 \leq 0 \end{aligned}$$

从单纯形表最后一行与第1张表中单位矩阵出现的列对应的元素, 就可以直接得到对偶问题的最优解 $\lambda^* = (-1, -1)^T$.

例 2.3.4 (线性约束的对偶解释) 考虑

$$\begin{aligned} & \text{minimize} && 18x_1 + 12x_2 + 2x_3 + 6x_4 \\ & \text{subject to} && 3x_1 + x_2 - 2x_3 + x_4 = 2 \\ & && x_1 + 3x_2 - x_4 = 2 \\ & && x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{aligned}$$

图 2.3.2(a)给出原始问题系数矩阵的列向量和右端向量 b 的图示. 基本可行解代表由两个 a_i 的非负线性组合可以合成 b . 对偶问题是

$$\begin{aligned} & \text{maximize} && 2\lambda_1 + 2\lambda_2 \\ & \text{subject to} && 3\lambda_1 + \lambda_2 \leq 18 \\ & && \lambda_1 + 3\lambda_2 \leq 12 \\ & && -2\lambda_1 \leq 2 \\ & && \lambda_1 - \lambda_2 \leq 6 \end{aligned}$$

对偶问题的几何表示见图 2.3.2(b). 原始问题的每一列 a_i 定义了对偶问题的一个半空间约束. 该半空间的边界即 $\{\lambda: \lambda^T a_i = c_i\}$, 其与列向量 a_i 正交. 对偶目标在对偶可行域的某极点处取得最大值, 该点处恰好有两个对偶约束是积极的, 且这两个积极约束与原始问题的最优基相对应. 事实上, 定义对偶目标函数的系数向量 b 是这两个向量的正线性组合. 在这个具体的例子中, b 是 a_1 和 a_2 的正线性组合, 且组合系数是原始问题最优解中 x_i 的取值.

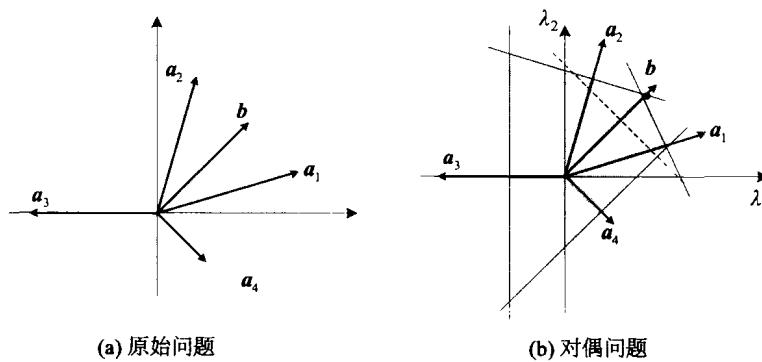


图 2.3.2 线性约束的对偶解释

最后, 我们给出单纯形法中向量 λ 的经济解释来结束本小节. 在单纯形法的每一步, 均可定义向量 $\lambda^T = c_b^T B^{-1}$, 它一般不是对偶问题的最优解, 除非 B 是原始问题的最优基. 然而, 它也有经济解释. 此外, 如在推演修正单纯形法时所见, 每一步可利用向量 λ 计算既约费用系数. 基

于此,通常称 $\lambda^T = c_B^T B^{-1}$ 是与基 B 对应的单纯形乘子(simplex multiplier).

与前文一致,记 A 的列为 a_1, a_2, \dots, a_n ,记 \mathbb{R}^m 中的 m 个单位向量为 e_1, e_2, \dots, e_m .显然, a_i 和 b 的分量直接说明了如何由 e_i 构造这些向量.给定任一由 A 的 m 个线性无关列组成的基 B ,通过这些向量的线性组合可以构造(合成)出 \mathbb{R}^m 中的任一向量.如果与基向量 a_i 对应的单价是 c_j ,则由基构造(合成)的向量的费用也可以通过计算得到.特别地, $\lambda^T = c_B^T B^{-1}$ 的第 i 个元素 λ_i 是由基构造第 i 个单位向量 e_i 时的费用.这样,可以将 λ_i 解释成单位向量的合成价格.

现在任一向量都可分两步由基表示:(i)用基表示单位向量;(ii)将想要的向量表示成单位向量的线性组合.这样,用基构造向量时相应的合成费用可以直接计算:(i)求出单位向量的合成价格;(ii)用这些价格求出单位向量线性组合的费用.这样,利用单纯形乘子可以快速求出任意的用单位向量来表示的向量的合成费用,这些向量的真正费用与合成费用之差是相对费用.有时将“用单纯形乘子计算一个向量关于给定基的合成费用”称为定价.

原始问题的最优性相于用基构造每个向量 a_1, a_2, \dots, a_n 所需的费用不超过每个向量自身的价格.这样,对所有的 j 有 $\lambda^T a_j \leq c_j$,或等价地,有 $\lambda^T A \leq c^T$.

2.3.4 灵敏度与互补性

上面说明可以将线性规划对偶问题的最优解解释为价格,现在进一步研究这种解释.假设线性规划问题(2.1.2)的最优基是 B ,对应解是 $x_B = B^{-1}b$.对偶问题的解是 $\lambda^T = c_B^T B^{-1}$.现在,假设这个最优解非退化,则对向量 b 进行微小的扰动时,最优基仍保持不变(请思考为什么).这样,与 $b + \Delta b$ 对应的最优解是 $x_B + \Delta x_B$,其中 $\Delta x_B = B^{-1} \Delta b$;费用函数的相应增量 $\Delta z = c_B^T \Delta x_B = \lambda^T \Delta b$,该等式说明 λ 给出最优费用关于向量 b 微小扰动的灵敏度.换句话说,如果求解一个将 b 替换为 $b + \Delta b$ 的新问题,目标函数最优值的改变量将是 $\lambda^T \Delta b$.

对偶向量 λ 的解释与它作为单纯形乘子的解释密切相关.因为 λ_i 是用基构造单位向量 e_i 时的价格,它直接度量了向量 b 的第 i 个分量的改变所引起费用的改变.当 b_i 变成 $b_i + \Delta b_i$ 时,最优值的改变量是 $\lambda_i \Delta b_i$.这样,可以把 λ_i 看作分量 b_i 的边际价格(marginal price)或影子价格(shadow price).原始-对偶问题的最优解还满足另一个具有经济解释的关系——互补性(complementarity).

定理 2.3.4 (互补性) 设 x 和 λ 分别是问题(2.1.2)和问题(2.3.1)的可行解.它们是各自最优解的充分必要条件是对所有的 j 有

- (i) $x_j > 0 \Rightarrow \lambda^T a_j = c_j$;
- (ii) $x_j = 0 \Leftrightarrow \lambda^T a_j < c_j$.

证明 如果所述条件成立,则显然有 $(\lambda^T A - c^T)x = 0$.这样 $\lambda^T b = c^T x$.由定理 2.3.1 的推论 1 可知,两个解均是各自的最优解.反之,如果两个解是最优的,由强对偶定理 2.3.2,必有 $\lambda^T b = c^T x$,因此有 $(\lambda^T A - c^T)x = 0$.因为 x 的每个分量非负,而 $\lambda^T A - c$ 的每个分量非正,所以条件(i)和(ii)成立. ■

定理 2.3.5 (对称形式的互补性) 设 x 和 λ 分别是式(2.3.2)中原始、对偶问题的可行解.它们是各自最优解的充分必要条件是对所有的 i 和 j 有

- (i) $x_j > 0 \Rightarrow \lambda^T a_j = c_j$;
- (ii) $x_j = 0 \Leftrightarrow \lambda^T a_j < c_j$;
- (iii) $\lambda_i > 0 \Rightarrow a^T x = b_i$;

$$(iv) \lambda_i = 0 \Leftrightarrow \mathbf{a}' \mathbf{x} > b_i.$$

其中 \mathbf{a}' 是 \mathbf{A} 的第 i 行.

对于该定理,可以将对称问题转换成标准形,利用前一个定理得到的结论,也可直接证明.互补条件有一个相当明显的经济解释.例如就配餐问题而言,它是对称形式对偶对中的原始问题,假设最优的食物供给方案中第 i 种营养的供给量大于 b_i 单位.这意味着营养学家不愿为第 i 种营养的少量变动作任何付出,因为它的微小改变并不能使最优配餐的费用减少.就上文将 λ_i 解释为边际价格的观点来看,这隐含着 $\lambda_i = 0$,恰好相应于定理 2.3.5 的(iv).读者可对其他条件作出类似的解释.这里给出对偶定理的一个简单应用.

例 2.3.5 (互补性) 问题

$$\begin{aligned} & \text{minimize} && 2x_1 + 3x_2 + x_3 \\ & \text{subject to} && 3x_1 - x_2 + x_3 \geq 1 \\ & && x_1 + 2x_2 - 3x_3 \geq 2 \\ & && x_1, x_2, x_3 \geq 0 \end{aligned}$$

不方便用图解法求解,但其对偶问题是

$$\begin{aligned} & \text{maximize} && \lambda_1 + 2\lambda_2 \\ & \text{subject to} && 3\lambda_1 + \lambda_2 \leq 2 \\ & && -\lambda_1 + 2\lambda_2 \leq 3 \\ & && \lambda_1 - 3\lambda_2 \leq 1 \\ & && \lambda_1, \lambda_2 \geq 0 \end{aligned}$$

这是两个变量的问题,可方便地用图解法得到最优解 $\lambda = (1/7, 11/7)^T$,进一步可以利用互补性求原问题的最优解.由于在 λ 处,对偶问题的第 3 个约束满足严格不等式,由定理 2.3.5 的结论(ii)得原问题的最优解处 $x_3 = 0$.而 λ 的分量都大于零,由定理 2.3.5 的结论(iii)知原问题的两个不等式约束在最优解处成立等式,于是有

$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1 \\ x_1 + 2x_2 - 3x_3 &= 2 \\ x_3 &= 0 \end{aligned}$$

解得原问题的最优解为 $\mathbf{x} = (4/7, 5/7, 0)^T$.

2.3.5 对偶单纯形法

在应用中经常出现的一种情况是:线性规划问题有一个很显然的基本解,它是不可行的,但既约费用系数是非负的,即单纯形乘子是对偶问题的可行解.在单纯形表格中,这种情况对应于最后一行无负元素,但最后一列有负元素.比如,当求出某一线性规划问题的解,然后通过改变向量 \mathbf{b} 构造一个新问题时,我们有对偶问题的基本可行解,因此用单纯形法求解对偶问题是再好不过的了.

与其针对对偶问题构造一个表格(如果原始问题是标准形,对偶问题涉及 m 个自由变量和 n 个非负的松弛变量),不如用原始问题的表格求解对偶问题,这样做会更有效一些.基于该想法的完美技术是对偶单纯形法.就原始问题而言,该过程是保持最优性这个前提条件来寻求可行解.

前面介绍的单纯形法实际上也可称为原始单纯形法,其中既约费用系数的正负刻画了对

偶约束的可行与否. 本小节讨论对偶单纯形法, 这也可以理解为将(原始)单纯形法应用于对偶线性规划问题. 具体地, 考虑标准形问题(2.1.2). 假设 \mathbf{B} 是已知的基, 且满足由 $\boldsymbol{\lambda}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ 定义的 $\boldsymbol{\lambda}$ 是对偶问题的可行解. 此时, 称原始问题的基本解 $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b}$ 是对偶可行的(dual feasible), 且显然满足 $\mathbf{c}_B^T \mathbf{x}_B = \boldsymbol{\lambda}^T \mathbf{b}$. 如果 $\mathbf{x}_B \geq \mathbf{0}$, 则这个解也是原始可行的, 因此必是最优的.

给定对偶可行的单纯形乘子向量 $\boldsymbol{\lambda}$, 即对 $j = 1, 2, \dots, n$, 满足 $\boldsymbol{\lambda}^T \mathbf{a}_j \leq c_j$. 假设基是 \mathbf{A} 的前 m 列, 则等式

$$(r_j =) c_j - \boldsymbol{\lambda}^T \mathbf{a}_j = 0, \quad j = 1, 2, \dots, m \quad (2.3.5)$$

和不等式

$$(r_j =) c_j - \boldsymbol{\lambda}^T \mathbf{a}_j > 0, \quad j = m+1, m+2, \dots, n \quad (2.3.6)$$

成立. 为了推演方便, 这里假设对偶问题是非退化的, 即非基变量的既约费用系数严格大于零. 现在确定新向量 $\hat{\boldsymbol{\lambda}}$ 使等式(2.3.5)中的一个变成不等式, 不等式(2.3.6)中的一个变成等式, 同时对偶问题的目标函数值增大. 这样新解中的 m 个等式确定了一个新基.

假设第 p 个变量出基. 下面记 \mathbf{B}^{-1} 的第 p 行为 \mathbf{u}^p . 由定理 2.2.2 知

$$\hat{\boldsymbol{\lambda}}^T = \boldsymbol{\lambda}^T - \epsilon \mathbf{u}^p \quad (2.3.7)$$

从而我们有 $\hat{r}_j = c_j - \hat{\boldsymbol{\lambda}}^T \mathbf{a}_j = r_j + \epsilon \mathbf{u}^p \mathbf{a}_j$. 因为 $\mathbf{u}^p \mathbf{a}_j$ 是单纯形表的第 (p, j) 个元素 y_{pj} , 从而

$$\left. \begin{array}{l} \hat{r}_p = \epsilon, \\ \hat{r}_j = 0, \quad j = 1, 2, \dots, m, j \neq p \\ \hat{r}_j = r_j + \epsilon y_{pj}, \quad j = m+1, m+2, \dots, n \end{array} \right\} \quad (2.3.8)$$

还有

$$\hat{\boldsymbol{\lambda}}^T \mathbf{b} = \boldsymbol{\lambda}^T \mathbf{b} - \epsilon y_{p0} \quad (2.3.9)$$

这样, 为了使对偶问题的目标函数值增大, 选取基本解中取负值的变量对应的列出基. 方法的伪码见算法 2.3.1.

Algorithm 2.3.1 Dual simplex method for linear programming problem(2.1.2)

```

1: Initially get a dual feasible basic solution  $\mathbf{x}_B$  ;
2: while 1 do
3:   find  $p$  such that  $y_{p0} = \min y_{pj}$  ;
4:   if  $y_{p0} \geq 0$  then
5:     return the current basic feasible solution is  $\mathbf{x}^*$  .
6:   else
7:     if for each  $j$   $y_{pj} \geq 0$  then
8:       return the primal is infeasible.
9:     else
10:    find

```

$$\hat{\epsilon} = \frac{r_q}{-y_{pq}} = \min_j \left\{ \frac{r_j}{-y_{pj}} : y_{pj} < 0, j = 1, 2, \dots, n \right\} \quad (2.3.10)$$

```

11:    form a new basis  $\mathbf{B}$  by replacing the  $p$ -th column in the basis with  $\mathbf{a}_q$  ;
12:    using this basis determine the corresponding basic dual feasible solution  $\mathbf{x}_B$  ;
13:  end if
14: end if
15: end while

```

算法收敛到最优解的证明在细节上与原始单纯形法类似. 基本的观察来源于：

- (a) 由式(2.3.10)中 q 的选取和式(2.3.8)知,新的基本解也是对偶可行的;
- (b) 由式(2.3.9)和 $y_{p0} < 0$ 知,对偶目标值会增大;
- (c) 过程不会终止在非最优解;
- (d) 因为存在有限个基,必在有限步达到最优.

一种经常出现的问题是极小化非负变量的正组合,约束是“大于等于”型不等式. 这些问题具备应用对偶单纯形法的自然条件.

例 2.3.6 (对偶单纯形法) 考虑

$$\begin{aligned} \text{minimize} \quad & 3x_1 + 4x_2 + 5x_3 \\ \text{subject to} \quad & x_1 + 2x_2 + 3x_3 \geq 5 \\ & 2x_1 + 2x_2 + x_3 \geq 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

通过引入盈余变量并改变不等式的符号,得到如下初始表格

	x_1	x_2	x_3	x_4	x_5	$B^{-1}b$
	-1	-2	-3	1	0	-5
	-2	-2	-1	0	1	-6
r^T	3	4	5	0	0	0

因为所有的 $c_j - z_j$ 都是非负的,所以该基本解是对偶可行的. 选取任一 $y_{10} < 0$,比如 $x_5 = y_{10} = -6$,让其出基. 为了在第 2 行找到合适的转轴元,计算比值 $\frac{r_j}{y_{2j}}$,然后找最小正比率,产生表中标记的转轴元. 第 2 张表格是

	x_1	x_2	x_3	x_4	x_5	$B^{-1}b$
	0	-1	$-\frac{5}{2}$	1	$-\frac{1}{2}$	-2
	1	1	$\frac{1}{2}$	0	$-\frac{1}{2}$	3
r^T	0	1	$\frac{7}{2}$	0	$\frac{3}{2}$	-9

第 3 张表格(亦即最终的表格)是

	x_1	x_2	x_3	x_4	x_5	$B^{-1}b$
	0	1	$\frac{5}{2}$	-1	$\frac{1}{2}$	2
	1	0	-2	1	-1	1
r^T	0	0	1	1	1	-11

由第 3 张表格可得原始问题的可行解,且是最优的. 这样,解 $x^* = (1, 2, 0)^T$.

最后特别指出的是,对于整数线性规划的 Gomory 割平面法和分枝定界法,以及一些灵敏度分析问题而言,有很显然的对偶可行基本解,因而在这些场合利用对偶单纯形法极为方便.

2.4 评注与参考

线性规划这一学科源于对线性不等式的研究. 相关研究可以上溯至法国数学家 Fourier 的工作. 从那时起,许多研究者各自证明了该学科中最重要的结论——对偶定理的各种特殊情况. 该学科的应用始于苏联数学家 Kantorovich. 他指出了一类线性规划在实际应用中的重要性,并给出求解这类问题的算法^[8]. 遗憾的是,有很多年 Kantorovich 的工作既未被西方学者所知,也未引起东方同行的关注.

美国数学家 Dantzig 于 1947 年在求解美国空军规划中的线性规划问题时提出了单纯形法,这为这门学科奠定了基础. Dantzig 也因此被称为线性规划之父,他的专著^[7]是该领域的重要参考文献. 在 Dantzig 发现单纯形法的同一年, Koopmans 证明线性规划是分析经典经济理论的恰当模型. Kantorovich 和 Koopmans 一起因“最优资源配置理论的贡献”荣获 1975 年诺贝尔经济学奖.

强调对偶定理背后所蕴含的思想可以追溯到 Dantzig 和美国数学家 von Neumann(计算机之父,博弈论之父)在 1947 年秋季的一次会谈. 但是一直没有显式的表述,直到 Gale 等的论文^[9]发表. 对偶理论的提出开创了线性规划的许多新的研究领域(博弈论),扩大了它的应用范围. 对偶单纯形法归功于 Lemke.

20 世纪 50 年代后,研究者对线性规划进行了大量的理论研究,并涌现出一大批新的算法. 苏联数学家 Khachiyan 提出解线性规划问题的椭球算法,并证明它是多项式时间算法^[11]. 刚出道不久的年轻的 Khachiyan 也因此一举成名. 但是椭球法的数值表现非常糟糕,远远没有单纯形法那么有效. 美国贝尔实验室的印度裔数学家 Karmarkar 于 1984 年提出解线性规划问题的新的多项式时间算法^[12],其在计算效率上可以和单纯形法相媲美,该算法的提出掀起了凸优化及内点法的研究与应用热潮,目前最有效的内点法是原始-对偶路径跟踪内点法.

线性规划的研究成果还直接推动了其他数学规划问题(包括整数规划、随机规划和非线性规划)的算法研究. 一方面,很多其他种类的最优化问题算法都可以分拆或者转化成线性规划子问题,然后求解. 另一方面,由线性规划引申出的很多概念启发了最优化理论的核心概念,诸如对偶、分解、凸性等.

习题 2

2.1 将下面的线性规划问题化成标准形,并求解第 3 个问题(c).

$$\begin{aligned}
 (a) \quad & \text{minimize} && x + 2y + 3z \\
 & \text{subject to} && 2 \leq x + y \leq 3 \\
 & && 4 \leq x + z \leq 5 \\
 & && x \geq 0, y \geq 0, z \geq 0 \\
 (b) \quad & \text{minimize} && x + y + z
 \end{aligned}$$

$$\begin{aligned}
 & \text{subject to} \quad x + 2y + 3z = 10 \\
 & \quad x \geq 1, y \geq 2, z \geq 1 \\
 (c) \quad & \text{minimize} \quad x_1 + 4x_2 + x_3 \\
 & \text{subject to} \quad x_1 - 2x_2 + x_3 = 4 \\
 & \quad x_1 - x_3 = 1 \\
 & \quad x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

2.2 将下面的问题化成线性规划

$$\begin{aligned}
 & \text{minimize} \quad |x| + |y| + |z| \\
 & \text{subject to} \quad x + y \leq 1 \\
 & \quad 2x + z = 3
 \end{aligned}$$

2.3 一类逐段线性函数 $f(\mathbf{x}) = \max\{\mathbf{c}_1^\top \mathbf{x} + d_1, \mathbf{c}_2^\top \mathbf{x} + d_2, \dots, \mathbf{c}_p^\top \mathbf{x} + d_p\}$, 其中 $\mathbf{c}_i \in \mathbb{R}^n, d_i \in \mathbb{R}$, $i = 1, 2, \dots, p$. 针对这样的函数, 考虑问题

$$\begin{aligned}
 & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\
 & \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & \quad \mathbf{x} \geq \mathbf{0}
 \end{aligned}$$

将此问题转化成线性规划.

2.4 给出例 2.1.4 的所有基本解.

2.5 考虑问题

$$\begin{aligned}
 & \text{minimize} \quad c_1 x_1 + c_2 x_2 + c_3 x_3 \\
 & \text{subject to} \quad x_1 + x_2 + x_3 \leq 4 \\
 & \quad x_1 \leq 2 \\
 & \quad x_3 \leq 3 \\
 & \quad 3x_2 + x_3 \leq 6 \\
 & \quad x_1, x_2, x_3 \geq 0
 \end{aligned}$$

注意系数 c_1, c_2, c_3 尚未确定. 表示成标准形 $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ 后, 其中

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 3 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 6 \end{bmatrix}$$

记 \mathbf{A} 的第 i 列为 \mathbf{a}_i .

- (a) 画出所给问题的可行域(三维空间中).
- (b) 点 $(0, 1, 3, 0, 2, 0, 0)^\top$ 是基本可行解吗?
- (c) 点 $(0, 1, 3, 0, 2, 0, 0)^\top$ 是退化基本可行解吗? 如果是, 找出可能的与其对应的基.

2.6 考虑问题(2.1.3)和给其添加松弛变量之后的线性规划问题, 一个在 \mathbb{R}^n , 另一个在 \mathbb{R}^{n+m} . 证明这两个问题的可行域的极点之间是一一对应的.

2.7 利用转轴运算求解方程组

$$\begin{array}{ll}
 \text{(a)} \quad 3x_1 + 2x_2 = 5 & \text{(b)} \quad x_1 + 2x_2 + x_3 = 7 \\
 5x_1 + x_2 = 9 & 2x_1 - x_2 + 2x_3 = 6 \\
 & x_1 + x_2 + 3x_3 = 12
 \end{array}$$

2.8 如果与每个非基变量 x_j 对应的既约费用系数 $r_j > 0$, 证明与其对应的基本可行解是唯一的最优解.

2.9 举例说明退化基本可行解不用满足所有 $r_j \geq 0$, 也可以是最优的.

2.10 将下面的问题转化成标准形, 用单纯形法求解, 然后画出问题在 x_1, x_2 空间的可行域, 并标明单纯形法的迭代路径.

$$\begin{array}{ll}
 \text{(a)} & \begin{array}{ll} \text{maximize} & -x_1 + x_2 \\ \text{subject to} & x_1 - x_2 \leq 2 \\ & x_1 + x_2 \leq 6 \\ & x_1 \geq 0, x_2 \geq 0 \end{array} \\
 \text{(b)} & \begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & -2x_1 + x_2 \leq 1 \\ & x_1 - x_2 \leq 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}
 \end{array}$$

2.11 利用单纯形法求解

$$\begin{array}{ll}
 \text{maximize} & 2x_1 + 4x_2 + x_3 + x_4 \\
 \text{subject to} & x_1 + 3x_2 + x_4 \leq 4 \\
 & 2x_1 + x_2 \leq 3 \\
 & x_2 + 4x_3 + x_4 \leq 3 \\
 & x_i \geq 0, i = 1, 2, 3, 4
 \end{array}$$

利用求解结果回答以下问题:

- 为使最优基保持不变, 给出 $b = (4, 3, 3)^T$ 中第 1 个元素的可变范围(其他的保持不变).
- 为使最优基保持不变, 给出 $c = (2, 4, 1, 1)^T$ 中第 1 个元素的可变范围(其他的保持不变). 同理给出第 4 个元素的可变范围.
- 对于 b 微小的改变, 最优解将发生怎样的改变?
- 对于 c 微小的改变, 最优值将发生怎样的改变?

2.12 考虑问题

$$\begin{array}{ll}
 \text{minimize} & x_1 - 3x_2 - 0.4x_3 \\
 \text{subject to} & 3x_1 - x_2 + 2x_3 \leq 7 \\
 & -2x_1 + 4x_2 \leq 12 \\
 & -4x_1 + 3x_2 + 3x_3 \leq 14 \\
 & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0
 \end{array}$$

- 找出一个最优解.
- 存在几个最优基本可行解?

(c) 证明：如果 $c_4 + \frac{1}{5}a_{14} + \frac{4}{5}a_{24} \geq 0$, 则以费用系数 c_4 和活动向量 $(a_{14}, a_{24}, a_{34})^T$ 能够引入另一个活动变量 x_4 , 使最优解保持不变.

2.13 与选取最负既约费用系数对应的变量进基的原则不同, 有人建议(当转入时)更好的准则或许是：选取使目标函数提高最多的变量进基. 证明该准则即选取极小化

$$\max_{i, y_{ik} > 0} r_k \frac{y_{i0}}{y_{ik}}$$

的指标 k 对应的变量 x_k 进基.

2.14 利用两阶段单纯形法求解问题

$$(a) \quad \begin{aligned} & \text{minimize} && -3x_1 + x_2 + 3x_3 - x_4 \\ & \text{subject to} && x_1 + 2x_2 - x_3 + x_4 = 0 \\ & && 2x_1 - 2x_2 + 3x_3 + 3x_4 = 9 \\ & && x_1 - x_2 + 2x_3 - x_4 = 6 \\ & && x_i \geq 0, i=1,2,3,4 \end{aligned}$$

$$(b) \quad \begin{aligned} & \text{minimize} && x_1 + 6x_2 - 7x_3 + x_4 + 5x_5 \\ & \text{subject to} && 5x_1 - 4x_2 + 13x_3 - 2x_4 + x_5 = 20 \\ & && x_1 - x_2 + 5x_3 - x_4 + x_5 = 8 \\ & && x_i \geq 0, i=1,2,3,4,5 \end{aligned}$$

2.15 说明在利用两阶段法求有基本可行解的问题时, 在第 I 阶段, 如果某个人工变量变成非基变量, 则它不必再变成基变量. 这样, 若人工变量变成非基变量, 则可以将其所在列从表格中删除.

2.16 在两阶段法的第 I 阶段, 假定给系统 $Ax = b, x \geq 0$ 的辅助问题应用单纯形法后, 所得表格(忽略费用行)形如

x_1	x_k	x_{k+1}	\cdots	x_n	y_1	\cdots	y_k	y_{k+1}	\cdots	y_m	
1								0	\cdots	0	b'_1
\ddots		R_1			S_1			\vdots		\vdots	\vdots
1								0	\cdots	0	b'_k
0	\cdots	0						1			0
\vdots	\vdots		R_2		S_2			\ddots		\vdots	
0	\cdots	0							1		0

即基变量中有 $m-k$ 个人工变量, 它们取零值.

- 证明 R_2 中的任何非零元素都可作为转轴元以消去人工基变量, 这样将产生一个类似的表格, 但 k 会增加 1.
- 重复(a)中的过程, 直到 $R_2 = 0$. 证明原始系统是冗余的, 并说明可以删除底端的这些行, 然后继续第 II 阶段.
- 利用上面的方法(即两阶段法)求解线性规划

$$\begin{aligned}
 & \text{minimize} && 2x_1 + 6x_2 + x_3 + x_4 \\
 & \text{subject to} && x_1 + 2x_2 + x_4 = 6 \\
 & && x_1 + 2x_2 + x_3 + x_4 = 7 \\
 & && x_1 + 3x_2 - x_3 + 2x_4 = 7 \\
 & && x_1 + x_2 + x_3 = 5 \\
 & && x_i \geq 0, i = 1, 2, 3, 4
 \end{aligned}$$

2.17 利用修正单纯形法找出下列系统的一个基本可行解

$$\begin{aligned}
 & x_1 + 2x_2 - x_3 + x_4 = 3 \\
 & 2x_1 + 4x_2 + x_3 + 2x_4 = 12 \\
 & x_1 + 4x_2 + 2x_3 + x_4 = 9 \\
 & x_i \geq 0, i = 1, 2, 3, 4
 \end{aligned}$$

2.18 可以将两阶段的单纯形法综合成大 M 法 (big-M method). 给定标准形式的线性规划

$$\begin{aligned}
 & \text{minimize} && \mathbf{c}^T \mathbf{x} \\
 & \text{subject to} && \mathbf{Ax} = \mathbf{b} \\
 & && \mathbf{x} \geq \mathbf{0}
 \end{aligned}$$

其中 $\mathbf{b} \geq \mathbf{0}$. 考虑逼近问题

$$\begin{aligned}
 & \text{minimize} && \mathbf{c}^T \mathbf{x} + M \sum_{i=1}^m y_i \\
 & \text{subject to} && \mathbf{Ax} + \mathbf{y} = \mathbf{b} \\
 & && \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}
 \end{aligned}$$

在此问题中, $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$ 是人工变量, M 是大的常数. 考虑将 $M \sum_{i=1}^m y_i$ 作为非零 y_i 的罚项, 并利用单纯形法求解该问题. 试证明下面结论:

- 如果求得一个最优解, 其中 $\mathbf{y} = \mathbf{0}$, 则对应的 \mathbf{x} 是原问题的最优基本可行解.
- 如果对每个 $M > 0$, 找到的最优解都有 $\mathbf{y} \neq \mathbf{0}$, 则原问题不可行.
- 如果对每个 $M > 0$, 近似问题是无界的, 则原问题或者无界, 或者不可行.
- 现在假定原问题的最优值 $V(\infty)$ 是有限的. 设 $V(M)$ 为近似问题的最优值, 证明 $V(M) \leq V(\infty)$.
- 证明对 $M_1 \leq M_2$, 有 $V(M_1) \leq V(M_2)$.
- 证明存在 M_0 使得对 $M \geq M_0$, 有 $V(M) = V(\infty)$. 因此断定: 对充分大的 M 值, 求解近似问题即可得到原问题的解, 此即求解一般线性规划的大 M 法.

2.19 下面的表格是用单纯形法求解极小化问题的一个中间表格

y_1	y_2	y_3	y_4	y_5	y_6	y_0
1	$2/3$	0	0	$4/3$	0	4
0	$-7/3$	3	1	$-2/3$	0	2
0	$-2/3$	-2	0	$2/3$	1	2
r^T	0	$8/3$	-11	0	$4/3$	0

(a) 确定下一个转轴元.

(b) 给定当前基的逆

$$\mathbf{B}^{-1} = [\mathbf{a}_1, \mathbf{a}_4, \mathbf{a}_6]^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & 2 \\ -1 & 2 & 1 \end{bmatrix}$$

和对应的费用系数 $\mathbf{c}_B = (c_1, c_4, c_6)^T = (-1, -3, 1)^T$. 确定原问题的数据 $(\mathbf{c}, \mathbf{A}, \mathbf{b})$.

2.20 利用单纯形法求解问题(2.2.15), 其中初始点 $\mathbf{x}^{(0)} = (0, 0, 0)^T$, 要求选既约费用系数最负的变量进基.

2.21 (Beale 问题) 考虑

$$\begin{aligned} \text{minimize} \quad & -\frac{3}{4}x_4 + 20x_5 - \frac{1}{2}x_6 + 6x_7 \\ \text{subject to} \quad & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0 \\ & x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0 \\ & x_3 + x_6 = 1 \\ & x_1 \geq 0, x_2 \geq 0, \dots, x_7 \geq 0 \end{aligned}$$

请以 x_1, x_2 和 x_3 为基变量的初始基本可行解为初始点, 利用单纯形法求解该问题. 要求选取既约费用系数最小的变量进基; 当有多个变量可以出基时, 选择指标最小的变量出基. 观察到了什么样的现象? 可以通过哪些措施来解决该问题?

2.22 详细验证问题(2.3.1)的对偶是问题(2.1.2).

2.23 找出

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{a} \end{aligned}$$

的对偶, 其中 \mathbf{a}, \mathbf{c} 和 \mathbf{A}, \mathbf{b} 是常量.

2.24 构造一个原始问题和对偶问题都没有可行解的例子.

2.25 设 \mathbf{A} 是 $m \times n$ 矩阵, \mathbf{b} 是 n 维向量. 证明 $\mathbf{A} \mathbf{x} \leq \mathbf{0}$ 蕴含着 $\mathbf{c}^T \mathbf{x} \geq 0$, 当且仅当存在 $\lambda \geq \mathbf{0}$ 使得 $\mathbf{c}^T + \lambda^T \mathbf{A} = \mathbf{0}$. 给出该结论的一个几何解释.

2.26 通常在优化理论和自由竞争之间有很强的联系, 这可通过经营实体选址的理想模型进行说明. 假设存在 n 种经营实体(各种工厂、公司、商场等), 准备将它们单独地设在 n 块不同的土地上. 设将实体 i 设在第 j 块上, 能够产生 s_{ij} 单位(元)的价值.

如果给实体指派土地的工作由专家来完成, 或许会通过使生产价值最大来作出决定. 换句话说, 要求指派极大化 $\sum_i \sum_j s_{ij} x_{ij}$, 其中

$$x_{ij} = \begin{cases} 1, & \text{若将实体 } i \text{ 指派到土地 } j \\ 0, & \text{其他} \end{cases}$$

更明确地说, 该方法需要求解优化问题

$$\text{maximize} \quad \sum_i \sum_j s_{ij} x_{ij}$$

$$\begin{aligned}
 \text{subject to} \quad & \sum_j x_{ij} = 1, \quad i = 1, 2, \dots, n \\
 & \sum_i x_{ij} = 1, \quad j = 1, 2, \dots, n \\
 & x_{ij} \geq 0, x_{ij} \in \{0, 1\}
 \end{aligned}$$

实际上,能够证明:通过约束定义的集合的任一极点是自动满足最终的要求($x_{ij} = 0$ 或1)的,所以通过线性规划的单纯形法能够找到最优指派.

现在从自由竞争的观点来考虑问题,假定不是由专家来确定指派,而是由各个实体对土地进行投标,从而建立价格机制.

- (a) 证明存在实体的价格 $p_i (i=1, 2, \dots, n)$ 和土地价格 $q_j (j=1, 2, \dots, n)$ 使得 $p_i + q_j \geq s_{ij}, i=1, 2, \dots, n, j=1, 2, \dots, n$. 如果一个最优解指派实体 i 到土地 j , 则上述不等式中等号成立.
- (b) 证明(a)蕴含着: 如果最优解指派实体 i 到土地 j , 且如果 j' 是任一其他的土地, 则有 $s_{ij} - q_j \geq s_{ij'} - q_{j'}$. 给出该结论的一个经济解释, 并以此为背景解释自由竞争和最优性之间的关系.
- (c) 假定每一个 s_{ij} 都是正的, 证明总可以假定价格是非负的.

- 2.27 博弈论(game theory)部分地涉及线性规划理论. 考虑如下博弈: 参与人甲可以在 m 种策略(strategy)中任选一种, 参与人乙可以在 n 种策略中任选一种. 如果甲选择策略 i , 乙选择策略 j , 则甲从乙处赢得的数量为 a_{ij} . 博弈通常会重复多次. 假设参与人甲设计了一种混合(mixed)策略, 记为向量 $\mathbf{x} = (x_1, x_2, \dots, x_m)^\top$, 其中分量 $x_i \geq 0$ 表示参与人选择策略 i 的概率, 要求 $\sum_{i=1}^m x_i = 1$. 同样, 乙的混合策略记为 $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$, 其中 $y_i \geq 0$, $\sum_{i=1}^n y_i = 1$, 这时乙给甲的平均支付 $P(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{A} \mathbf{y}$.

(a) 考虑线性规划问题

$$\begin{aligned}
 \text{maximize} \quad & \alpha \\
 \text{subject to} \quad & \sum_{i=1}^m x_i = 1 \\
 & \sum_{i=1}^m x_i a_{ij} \geq \alpha, \quad j = 1, 2, \dots, n \\
 & x_i \geq 0, \quad i = 1, 2, \dots, m
 \end{aligned}$$

设 (α, \mathbf{x}) 是该问题的可行解. 证明当参与人甲选择可行解中的 \mathbf{x} 作为自己的策略时, 不管乙选取何种策略 \mathbf{y} , 甲的支付至少为 α .

(b) 证明上面问题的对偶是

$$\begin{aligned}
 \text{minimize} \quad & \beta \\
 \text{subject to} \quad & \sum_{j=1}^n y_j = 1 \\
 & \sum_{j=1}^n a_{ij} y_j \leq \beta, \quad i = 1, 2, \dots, m \\
 & y_j \geq 0, \quad j = 1, 2, \dots, n
 \end{aligned}$$

- (c) 证明 $\max \alpha = \min \beta$ (称这个共同的值是博弈的值(value), 对应的解 \mathbf{x}^* 和 \mathbf{y}^* 是这个博弈的平衡点).
- (d) 考虑“匹配博弈”. 每个参与人选择正面或反面. 如果选择匹配, 即一个选正面的同时另外一个选反面, 则甲从乙那里赢得 1 元; 如果它们不匹配, 则乙从甲那里赢得 1 元. 求出该博弈的值和平衡点.
- (e) 考虑博弈, 其中每个参与人可以选择 1、2、3 中的某一个. 相应的支付规则是: 当参与人的数字相等时, 不发生支付; 当参与人的数字恰好比另一个参与人的数字大 1 时, 该参与人输 3 元; 当参与人的数字恰好比另一个参与人的数字大 2 时, 该参与人赢 1 元. 确定该博弈的值和平衡点.
- 2.28 考虑原始线性规划问题(2.1.2). 假设它和对偶问题(2.3.1)是可行的, 且已知对偶问题的最优解 λ^* .
- (a) 如果给原始问题的第 k 个方程乘以 $\mu \neq 0$, 确定这个新问题对应的对偶问题的最优解 \mathbf{w} .
- (b) 假设在原始问题中, 给第 r 个方程加上第 k 个方程的 μ 倍. 求新问题的对偶问题的最优解 \mathbf{w} .
- (c) 假设在原始问题中, 给 \mathbf{c}^T 加上 \mathbf{A} 的第 k 行的 μ 倍. 求新问题的对偶问题的最优解 \mathbf{w} .
- 2.29 一个公司可以制造 n 种不同的产品, 每种使用不同数量的 m 种有限资源. 每单位的产品 j 产生 c_j 元的利润, 并使用 a_{ij} 单位的第 i 种资源. 第 i 种资源的拥有量是 b_i . 为了极化利润, 公司通过求解

$$\begin{aligned} & \text{maximize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned}$$

来选择制造每种产品的数量 x_j . 单位利润 c_j 已经考虑到与制造每单位产品相关的可变费用. 除此费用之外, 公司承受一项额外的固定开销 H , 并且为了便于核算, 想将此项开销分配到它的每一种产品. 换句话说, 它想调整单位利润以便核算额外开销. 这样分配额外开销的机制必须满足两个条件: ① 因为不管产品组合怎样, H 是固定的, 所以分配额外开销的机制务必不改变最优解. ② 所有的额外开销必须被分配, 即使用修正费用系数的最优目标值必须比原始最优目标值 z 少 H 元.

- (a) 考虑根据 $\hat{\mathbf{c}}^T = \mathbf{c}^T - r \lambda_0^T \mathbf{A}$ 来修正单位利润的分配机制, 其中 λ_0 是原始问题的对偶问题的最优解, $r = H/z_0$ (假定 $H \leq z_0$).
- (i) 证明修正问题的最优解与原始问题的最优解相同, 且新的对偶最优解是 $\hat{\lambda}_0 = (1 - r)\lambda_0$.
- (ii) 说明这种方法可以彻底地分配 H .
- (b) 假设这项开销可以追溯到每一种资源约束. 设 $H_i \geq 0$ 是与第 i 种资源相关的开销, 其中 $\sum_{i=1}^m H_i \leq z_0$ 且 $r_i = H_i/b_i \leq \lambda_i^0, i = 1, 2, \dots, m$. 根据该信息提出了一种分配机制, 它修正单位利润使得 $\hat{\mathbf{c}}^T = \mathbf{c}^T - \mathbf{r}^T \mathbf{A}$.

- (i) 说明该修正问题的最优解与原始问题的最优解相同, 且对应的对偶最优解是
 $\hat{\lambda}_0 = \lambda_0 - r$.

- (ii) 说明这种机制可以彻底地分配 H .

2.30 考虑求解线性不等式组

$$\begin{aligned} -2x_1 + 2x_2 &\leq -1 \\ 2x_1 - x_2 &\leq 2 \\ -4x_2 &\leq 3 \\ -15x_1 - 12x_2 &\leq -2 \\ 12x_1 + 20x_2 &\leq -1 \end{aligned}$$

其中 x_1 和 x_2 没有非负限制. 该问题可以转化为在这些约束条件下极大化 $0 \cdot x_1 + 0 \cdot x_2$. 为此, 写出这个问题的对偶问题, 并利用单纯形法求解对偶问题, 由此得到原始问题的所有最优基本解. 这些最优基本解的所有凸组合即是待求的解集.

2.31 (a) 利用单纯形法求解

$$\begin{aligned} \text{minimize} \quad & 2x_1 - x_2 \\ \text{subject to} \quad & 2x_1 - x_2 - x_3 \geq 3 \\ & x_1 - x_2 + x_3 \geq 2 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

提示: 将所给问题转化成标准形后, $x = (2, 0, 0, 1, 0)^\top$ 是问题的一个基本可行解.

- (b) 对偶问题是什么? 对偶问题的最优解怎样?

2.32 (a) 利用单纯形法求解

$$\begin{aligned} \text{minimize} \quad & 2x_1 + 3x_2 + 2x_3 + 2x_4 \\ \text{subject to} \quad & x_1 + 2x_2 + x_3 + 2x_4 = 3 \\ & x_1 + x_2 + 2x_3 + 4x_4 = 5 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{aligned}$$

(b) 利用(a)中完成的工作和对偶单纯形法求解相同的问题, 不过方程组的右端项分别变成 1 和 8.

2.33 对于问题

$$\begin{aligned} \text{minimize} \quad & 5x_1 - 3x_2 \\ \text{subject to} \quad & 2x_1 - x_2 + 4x_3 \leq 4 \\ & x_1 + x_2 + 2x_3 \leq 5 \\ & 2x_1 - x_2 + x_3 \geq 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

- (a) 以 1 为转轴元, 仅转轴一次找到一个可行解.

- (b) 利用单纯形法求解问题.

- (c) 对偶问题是什么?

- (d) 对偶问题的解怎么样?

2.34 考虑

$$\begin{aligned}
 \text{minimize} \quad & 12x_1 + 8x_2 + 16x_3 + 12x_4 \\
 \text{subject to} \quad & -2x_1 - x_2 - 4x_3 + x_5 = -2 \\
 & -2x_1 - 2x_2 - 4x_4 + x_6 = -3 \\
 & x_i \geq 0, i = 1, 2, \dots, 6
 \end{aligned}$$

- (a) 利用对偶单纯形法求解问题.
 (b) 写出对偶问题, 并用图解法求解.
 (c) 写出(a)中与每张单纯形表对应的单纯形乘子, 并将这些点标在(b)中的图上. 写出你发现的事实.

2.35 给定标准形式的线性规划问题, 假设已知基 B , 以及对应(不必可行)的原始、对偶基本解 x 和 λ . 假定至少有一个既约费用系数 $c_j - \lambda^T a_j$ 是负的. 考虑辅助问题

$$\begin{aligned}
 \text{minimize} \quad & c^T x \\
 \text{subject to} \quad & Ax = b \\
 & \sum_{j \in T} x_j + y = M \\
 & x \geq 0, y \geq 0
 \end{aligned}$$

其中 $T = \{j : c_j - \lambda^T a_j < 0\}$, y 是松弛变量, M 是一个大的正常数. 证明如果 k 是原始解中与最负既约费用系数对应的指标, 则 $(\lambda^T, c_k - \lambda^T a_k)^T$ 是辅助问题的对偶可行解. 根据该观察, 研发针对对偶单纯形法的大 M 人工约束法.

2.36 纺织品公司生产 3 种产品, 假设产量分别为 x_1, x_2, x_3 . 它的下个月的生产计划必须满足约束

$$\begin{aligned}
 x_1 + 2x_2 + 2x_3 & \leq 12 \\
 2x_1 + 4x_2 + x_3 & \leq f \\
 x_1 & \geq 0, x_2 \geq 0, x_3 \geq 0
 \end{aligned}$$

其中第一个约束由设备的能力确定, 并且是固定的; 第二个约束由棉花的可用性确定. 除去棉花的费用和固定费用之外, 产品的净利润分别是 2, 3 和 3.

- (a) 求出影子价格 λ_2 作为棉花输入 f 的函数 $\lambda_2(f)$ (提示: 利用对偶单纯形法). 画出 $\lambda_2(f)$ 和除去棉花费用外的净利润 $z(f)$ 的图形.
 (b) 公司可以从一个开放市场以 $1/6$ 的价格购买棉花. 然而, 公司也可以 $1/12$ 的价格从一个经常购买其货物的供应商那里获得数量有限的棉花. 确定公司的净利润 π 作为新购买棉花数量 s 的函数 $\pi(s)$.

第3章 线性规划：扩展及其应用

作为线性规划的一个重要的应用,我们在2.1节中介绍了运输问题,实际中类似这样点到点搬运物资时极小化运输费用的问题,统称为最小费用网络流问题,泛见于运输、电子和通信网等领域.原则上,这类问题可以用一般的单纯形法来求解,但若能充分利用其特殊结构,相应的特殊单纯形法会变得更为简单.此外,许多重要的网络流问题又是最小费用网络流问题进一步的特例.

还有许多实际问题,当将其建模成线性规划时,额外还需要部分或全部变量取整数值.称这样的问题为整数线性规划问题,并在3.3节中介绍求解它的典型方法.

3.1 网络单纯形法

3.1.1 问题的表述

定义网络(network)为 $(\mathcal{N}, \mathcal{A})$,其中 \mathcal{N} 表示节点集, \mathcal{A} 表示弧集.节点与弧是网络中的两个最基本元素.单个节点(node)用一个小写英文字母或者数字来代替,比如节点*i*和节点*j*等,网络中节点的个数(即 \mathcal{N} 包含元素的个数)设为*m*;节点与节点之间由弧(arc)连接,弧是有方向的,通常用有序对来表示,比如将连接节点*i*和*j*的弧记为 (i, j) .弧集 \mathcal{A} 是所有可能弧的一个子集,即 $\mathcal{A} \subset \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$.事实上,在一般的网络中,每个节点仅与很少的邻近节点相连,因而集合 \mathcal{A} 中的元素常常很少.网络亦称图(graph)或有向图(digraph).图3.1.1(a)给出了一个由5个节点和8条弧组成的网络.

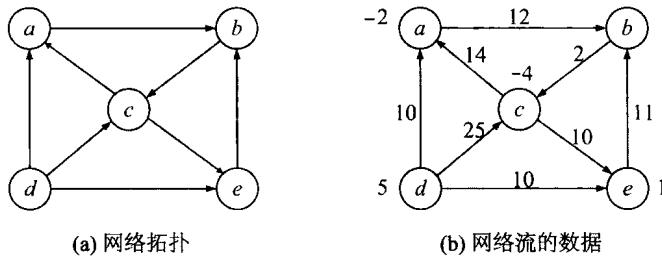


图3.1.1 一个网络流问题

最小费用网络流问题(minimum-cost network flow problem)定义为把供给点的物资运到各个需求点,在满足需求的同时总运费最小.一般设 b_i 为节点*i*处的物资供给量,其中 $b_i < 0$ 的节点为需求点,需求量是 $-b_i$.从节点*i*沿弧 (i, j) 运送一单位物资到节点*j*的费用记为 c_{ij} .图3.1.1(b)是网络流的数据信息.为使叙述清晰,在每个节点旁标出物资的供给量(未标出数字的节点默认供给量为0).作为基础,以下仅讨论产销平衡问题,即 $\sum_{i \in \mathcal{N}} b_i = 0$.

设沿弧 (i, j) 的运输量是 x_{ij} ,这样优化的目标是 $\min \sum_{(i, j) \in \mathcal{A}} c_{ij} x_{ij}$,这里的决策变量 x 需要满足一些约束.首先,它们必须使得每个节点处的流是守衡的.以节点 $k \in \mathcal{N}$ 为例,从节点 k 流出的量是 $\sum_{j: (k, j) \in \mathcal{A}} x_{kj}$,流入节点 k 的量是 $\sum_{i: (i, k) \in \mathcal{A}} x_{ik}$.前者减去后者必须等于这个节点处的供给量.因此,流平衡(flow conservation)约束为

$$\sum_{j: (k, j) \in \mathcal{A}} x_{kj} - \sum_{i: (i, k) \in \mathcal{A}} x_{ik} = b_k, \quad k \in \mathcal{N}$$

此外,每条弧上的流量必须是非负的(否则将它视为反方向的流),即 $x_{ij} \geq 0, (i, j) \in \mathcal{A}$.这样,最小费用网络流问题可以用矩阵记号表示成

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned}$$

这与一般的线性规划模型完全相同,但系数矩阵 \mathbf{A} 具有特殊的结构.以图3.1.1为例,决策变量 $\mathbf{x} = (x_{ab}, x_{bc}, x_{ca}, x_{ac}, x_{cd}, x_{dc}, x_{de}, x_{eb})^T$,费用向量 $\mathbf{c} = (12, 2, 14, 10, 10, 25, 10, 11)^T$.约束条件的系数矩阵和右端向量分别为

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & -1 & & & & \\ -1 & 1 & & & & & \\ & -1 & 1 & 1 & -1 & & \\ & & & & 1 & 1 & 1 \\ & & & & -1 & -1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -2 \\ 0 \\ -4 \\ 5 \\ 1 \end{bmatrix}$$

在最小费用网络流问题中,称这样的系数矩阵 \mathbf{A} 为点弧关联矩阵(node-arc incidence matrix),称变量 \mathbf{x} 为原始流(primal flows).

3.1.2 生成树与基

为了描述基矩阵的网络结构,先引进一些定义.对每条弧 (i, j) ,称 i 为尾(tail), j 为头(head).节点 i 的出度(outdegree)是以 i 为尾的弧数,入度(indegree)指以 i 为头的弧数.入度为零的节点称为源(source),出度为零的节点称为宿(sink).源仅发放物资,而宿仅接收物资.

定义网络中连接任何两个节点的一串邻接弧序列是路(path,路内弧的方向不必相同,但是路中的弧只能出现一次).称任意两个节点都有路连接的图是连通的(connected).图3.1.2(a)和(c)中的网络连通,图3.1.2(b)中的不连通.称第一个和最后一个节点重合的路为圈(cycle),称不含圈的图是无圈的(acyclic).图3.1.2(a)中的网络含圈,(b)和(c)中的网络不含圈.

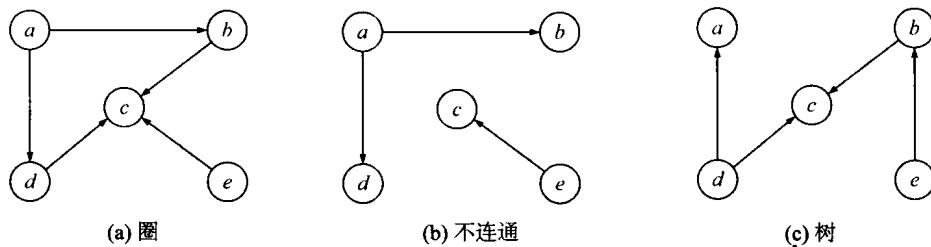


图3.1.2 网络基本概念

不含圈的连通图称为树(tree). 图 3.1.2(a)和(b)中的网络都不是树, 图 3.1.2(c)中的网络是树. 可以证明具有 m 个节点的树有 $m-1$ 条弧(习题 3.2). 在树中, 称出度与入度之和为 1 的节点为叶子(leaf)节点. 如果 $\tilde{\mathcal{N}} \subseteq \mathcal{N}$ 并且 $\tilde{\mathcal{A}} \subseteq \mathcal{A}$, 称网络 $(\tilde{\mathcal{N}}, \tilde{\mathcal{A}})$ 为网络 $(\mathcal{N}, \mathcal{A})$ 的子网络(subnetwork). 如果一个 $\tilde{\mathcal{N}} = \mathcal{N}$ 的子网络同时还是一棵树, 则称为生成树(spanning tree). 这时, 可以用弧集等价地刻画生成树. 以下简称生成树上的弧为树弧, 其余的为非树弧.

本章只考虑连通的网络. 给定一个网络流问题, 满足流平衡约束的解称为平衡(balanced)流. 非负的平衡流称为可行流. 给定一棵生成树, 非树弧上流量皆为 0 的平衡流称为树解(tree solution), 注意它对应着单纯形法中的基本解. 例如, 图 3.1.3(a)是图 3.1.1 所给问题的一个树解. 通过从某个叶子节点出发, 逆向依次求解流平衡方程可以得到树解, 比如

$$\begin{aligned} a \text{ 点流平衡: } -x_{da} &= -2 \Rightarrow x_{da} = 2 \\ d \text{ 点流平衡: } x_{da} + x_{dc} &= 5 \Rightarrow x_{dc} = 3 \\ c \text{ 点流平衡: } -x_{dc} - x_{bc} &= -4 \Rightarrow x_{bc} = 1 \\ b \text{ 点流平衡: } x_{bc} - x_{be} &= 0 \Rightarrow x_{be} = 1 \end{aligned}$$

读者可以自行证明求解过程的可行性. 只要注意, 每棵树至少有一个叶子节点, 且删除该叶子节点剩下的部分仍然构成一棵子树, 从而可以应用数学归纳法去证明.

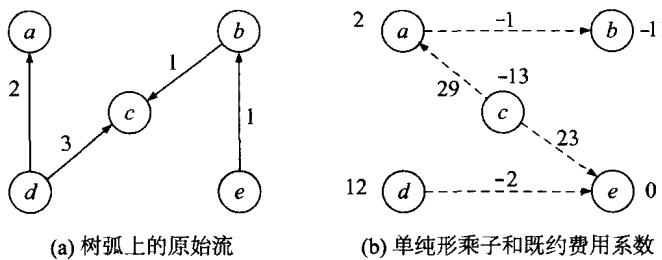


图 3.1.3 树解

上述简单的计算表明生成树恰对应着单纯形法中的基. 接下来深入阐述这一点. 实际上一个基对应着约束系数矩阵的一个可逆子方阵. 然而就关联矩阵 \mathbf{A} 而言, 并不存在这样的可逆子矩阵. 这是因为如果把 \mathbf{A} 的所有行相加, 由于 \mathbf{A} 的每列有且仅有一个 1 和一个 -1, 则会得到一个元素全为 0 的行向量. 对于任一子方阵也有这样的性质, 从而任一子方阵奇异. 事实上, 对于连通的网络, 流平衡约束中有且只有一个冗余的约束, 即 \mathbf{A} 的秩为 $m-1$.

(任意)选择一个叶子节点, 删除与之相对应的流平衡约束, 并称为根(root)节点. 对于图 3.1.1 所示的网络, 以下的内容均是在根节点为 e 的假设下说明的. 删除根节点之后的关联矩阵记为 $\tilde{\mathbf{A}}$, 相应的供需向量记为 $\tilde{\mathbf{b}}$, 则网络流问题有如下最重要的性质.

定理 3.1.1 $\tilde{\mathbf{A}}$ 的一个子方阵是基当且仅当其列对应的弧恰好组成一个生成树.

充分性的证明留给读者(习题 3.3). 必要性即要求对任一生成树证明其弧对应的 \mathbf{A} 的列线性无关. 前面用图 3.1.3 所示的生成树解释了证明的基本思想, 详细的证明过程留给读者. 令 \mathbf{B} 为该生成树的弧对应的 $\tilde{\mathbf{A}}$ 的各列组成的子方阵, 以下记这些树弧组成的集合为 \mathcal{T} . \mathbf{B} 可逆当且仅当以之为系数矩阵的线性方程组

$$\mathbf{Bx}_B = \tilde{\mathbf{b}} \quad (3.1.1)$$

有唯一解. 正是通过解这一方程组而得到树解的.

3.1.3 网络单纯形法

最小费用网络流问题作为线性规划问题,可以用单纯形法解之. 然而,基于其自身的特殊结构,可以不再使用传统的单纯形表,而代之以更直观的网络结构,称为网络单纯形法.

前面介绍了从叶子节点出发依次求解的方法. 其本质上是先将 \mathbf{B} 的行和列进行重新排列以得到一个下三角矩阵,然后再求解该下三角方程组. 具体到上面的计算,作如下的行列重排,即

$$\mathbf{PBQ}^T = \begin{matrix} (d,a)(d,c)(b,c)(e,b) \\ a \begin{bmatrix} -1 \\ 1 & 1 \\ c & -1 & -1 \\ b & 1 & -1 \end{bmatrix} \end{matrix}$$

重排之后得到一个下三角矩阵,这也验证了 \mathbf{B} 的可逆性,这里 \mathbf{P} 和 \mathbf{Q} 是置换矩阵. 因为 \mathbf{B} 的主对角线元素为 1 或者 -1 ,并且非对角线元素也是 1 或者 -1 ,因而解该方程组不需要进行任何除法和乘法运算,总之,只需要简单的加减法,就可以求解以 \mathbf{B} 为系数矩阵的线性方程组(3.1.1).

除了与基相关的原始解外,还可通过解方程组 $\mathbf{y}^T \mathbf{B} = \mathbf{c}_B^T$ 得到对应的单纯形乘子 \mathbf{y} . 由于 \mathbf{B} 的特殊性,有

$$y_i - y_j = c_{ij}, \quad (i, j) \in \mathcal{T} \quad (3.1.2)$$

因为一棵 m 个节点的生成树有 $m-1$ 条弧,这些等式给出了关于 m 个未知数的 $m-1$ 个方程. 由于原始问题在根节点处有一个冗余约束,比如第 m 个节点对应的约束,因此删除该冗余约束便得到关于 $m-1$ 个未知数的 $m-1$ 个方程. 于是,令与根节点对应的单纯形乘子为零,然后从根节点出发依次解方程(3.1.2)得到 \mathbf{y} . 以图 3.1.3 中的生成树为例,则

$$y_e = 0$$

$$\text{经过弧}(e, b) : y_e - y_b = 11 \Rightarrow y_b = -11$$

$$\text{经过弧}(b, c) : y_b - y_c = 2 \Rightarrow y_c = -13$$

$$\text{经过弧}(d, c) : y_d - y_c = 25 \Rightarrow y_d = 12$$

$$\text{经过弧}(d, a) : y_d - y_a = 10 \Rightarrow y_a = 2$$

求出单纯形乘子,那么与非树弧对应的既约费用系数为

$$r_{ij} = c_{ij} - (y_i - y_j), \quad (i, j) \notin \mathcal{T} \quad (3.1.3)$$

并把计算结果列在图 3.1.3(b) 中的非树弧上. 为了与树弧区别,非树弧是用虚线表示的.

如果所有的流量非负(即原始可行),且所有的既约费用系数非负(即对偶可行),那么当前的树解是最优的. 图 3.1.3 所示的树解满足第一个条件,但是不满足第二个,这意味着可以使用单纯形法. 单纯形法的基本思想是选择一个既约费用系数为负的非基变量(即非树弧)进基,然后通过转轴得到下一个基本可行解(可行树解). 把单纯形法的基本思想在网络中继承下来,就是网络单纯形法.

图 3.1.3 所示的树解中,非树弧 (a, b) 和 (d, e) 的既约费用系数是负的,选择一个最小的进基,即让 (d, e) 成为新的生成树的弧,称为入弧,然后进行转轴. 注意,转轴的根本目的是确定出基变量,同时产生新的基本可行解. 变量出基的标志是新的基本可行解中该分量为 0. 有了这些认识基础,可以来继续理解网络单纯形法了. 当前的生成树和入弧合在一起势必产生一

个圈,该圈上的流量受到入弧的影响会改变,而生成树的其他弧上的流量保持不变(如图 3.1.4 所示),入弧进入生成树产生圈,即圈“bcde”. 随着入弧上流量 t 的逐步增大,最后(当 $t=3$ 时)弧 (d, c) 上的流量降为 0. 因此,弧 (d, c) 要退出生成树,称为出弧. 令 $t=3$,并进行相应的流量更新.

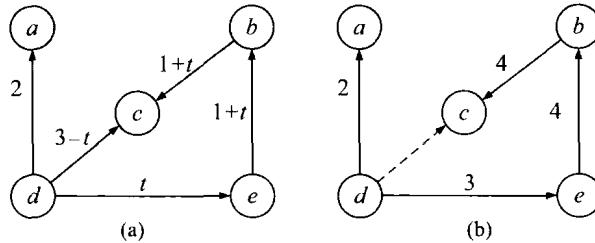


图 3.1.4 确定出弧

综上所述,在与入弧共圈的反方向弧中选流量最小的作为出弧. 同时,与出弧同方向的弧的流量减去该出弧的原有流量;与出弧反方向的弧的流量加上该出弧的原有流量. 注意,如果所有可能弧集合为空(即圈中所有的弧与入弧同向),则原始问题的最优值是 $-\infty$.

接下来应该计算新的既约费用系数,可以由单纯形乘子得到这些数据. 为了得到单纯形乘子,可以从根节点出发,求解新的方程(3.1.2),即从根节点出发沿着生成树依次计算,再按照式(3.1.3)得到非树弧的既约费用系数.

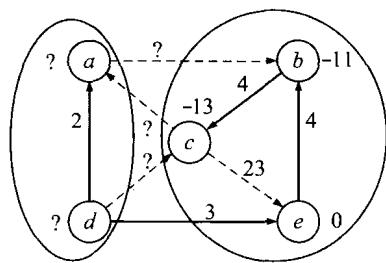


图 3.1.5 单纯形乘子和既约费用系数的快捷计算方法

下面介绍一种更新单纯形乘子和既约费用系数的快捷的计算方法. 如果删除出弧,则原生成树将一分为二,得到两棵子树. 在上述例子中,一棵子树包含节点 a 和 d ,而另一棵子树包含其余的节点,见图 3.1.5.

由于根节点的单纯形乘子恒为 0,因而包含根节点的子树上的单纯形乘子均保持不变,只需要更新另一棵子树上的单纯形乘子. 图 3.1.5 中,已标出的数据与迭代前相同,即保持不变,而需要更新的都以问号标出. 这里的“另一棵子树”由节点 a 和 d 组成,它们将分别增加相同的量,这是因为唯一的变化是桥接两棵子树的弧从出弧 (d, c) 换成入弧 (d, e) . 用带“~”的量来表示改变之后的量. 对节点 d 有

$$\bar{y}_d = c_{de} + \bar{y}_e = c_{de} + y_e$$

又因为

$$r_{de} = c_{de} - (y_d - y_e)$$

联立这两个方程,得到

$$\bar{y}_d = y_d + r_{de}$$

即节点 d 的单纯形乘子增加了 $r_{de} = -2$. 当然,这棵子树上的所有单纯形乘子都分别增加这个相同的量. 一般地,如果入弧从含根节点的子树指向不含根节点的子树,则不含根节点的子树上的各单纯形乘子分别减去入弧的原有既约费用系数;否则,这些单纯形乘子均加上入弧的原有既约费用系数.

有了单纯形乘子之后,就可以更新既约费用系数. 通过单纯形乘子的更新方式发现, 同一棵子树上的非树弧上的既约费用系数保持不变; 而对于那些桥接两棵子树的非树弧, 由于它们的终点和起点的某一个单纯形乘子发生改变而另一个保持不变, 所以它们的既约费用系数需要改变. 进一步, 以和入弧相同的方向桥接两棵子树的非树弧的既约费用系数减去 r_{de} , 而以和入弧相反的方向桥接两棵子树的非树弧的既约费用系数加上 r_{de} . 在这个例子中, 非树弧 (a, b) 和 (d, c) 桥接两棵子树, 且与入弧桥接方向相同, 它们都减去 -2 ; 而非树弧 (c, a) 桥接两棵子树, 但与入弧方向相反, 所以它要加上 -2 . 更新之后的树解如图 3.1.6 所示, 因为新的树解的既约费用系数都非负, 从而得到最优解.

方法的伪码见算法 3.1.1.

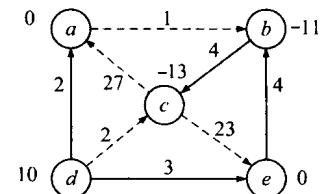


图 3.1.6 第一步完成后的树解

Algorithm 3.1.1 Simplex method for minimum-cost network flow problem

```

1: Compute a feasible tree solution  $\mathcal{T}$  with root node  $m$ ;
2: start from the root node and solve the system
    $y_m = 0,$ 
    $y_i - y_j = c_{ij}, (i, j) \in \mathcal{T}$ 
   to get the simplex multiplier  $y$ ;
3: compute the reduced cost coefficient on each nontree arc using the formula
    $r_{ij} = c_{ij} - (y_i - y_j), (i, j) \notin \mathcal{T}.$ 
4: while 1 do
5:   find the nontree arc  $(i, j)$  such that
    $(i, j) = \arg \min_{(i, j) \notin \mathcal{T}} r_{ij}.$ 
6:   if  $r_{ij} \geq 0$  then
7:     return the current tree solution is  $x^*$ 
8:   else
9:     add the entering arc  $(i, j)$  to the tree solution;
10:    with this arc added, there must be a cycle consisting of the entering arc and other tree arcs;
11:    if the arcs on the cycle are in the same direction then
12:      return the problem is unbounded.
13:    else
14:      the leaving arc  $(u, v)$  is chosen from those arcs on the cycle that go in the opposite direction from
         the entering arc and having the smallest flow among all such arcs;
15:    end if
16:    update primal flows, simplex multipliers and the reduced cost coefficients as was said before;
17:  end if
18: end while

```

最后, 给出网络单纯形法的一个特殊而重要的性质. 需要强调的是, 以上考虑的网络流问题都是无容量限制的, 即变量 x 没有上界约束. 对于这类问题, 有如下的整性定理(integrality

theorem) 成立.

定理 3.1.2 (整性定理) 对无容量限制的最小费用网络流问题:

- (i) 如果供给量 b_i 都是整数, 则每个基本可行解的分量都是整数.
- (ii) 如果费用系数 c_{ij} 都是整数, 则与每个树解对应的单纯形乘子的分量都是整数.

定理证明并不困难. 证明第一条时只要注意整数对加减法具有封闭性, 而前面讲过, 为得到树解而求解的线性方程组只需要加法和减法运算. 第二条的证明类似, 留给读者. 带整数约束的问题称为整数线性规划, 3.3 节将要进一步介绍, 它们通常很难求解, 然而这里对具有整性数据的最小费用流问题却可以通过网络单纯形法求解. 结合上面的整性定理以及网络单纯形法的计算步骤, 这一点是很自然的事实.

3.2 最小费用流问题的应用

3.2.1 运输问题和指派问题

沿着网络运输货物的网络流问题统称为转运(transshipment)问题. 再特殊一点, 考虑节点集合 N 只有源集 S 和宿集 T , 即

$$N = S \cup T, \quad S \cap T = \emptyset$$

这样, S 是 A 中弧的起点集合, T 是终点集合. 这里 S 中的节点可以看成源(供给)节点, T 中的节点称为宿(需求)节点. 称这样的图为二部图(bipartite graphs), 如图 3.2.1 所示, 其上的网络流问题称为运输(transportation)问题, 模型见式(2.1.1). 直观上讲, 运输问题是没有中转节点的转运问题. 它又分为两类: 一类是 3.1 节提到的满足产销平衡假定的问题, 称为平衡运输问题; 另一类则是不平衡运输问题, 引入辅助变量可以将这类问题转化为平衡运输问题.

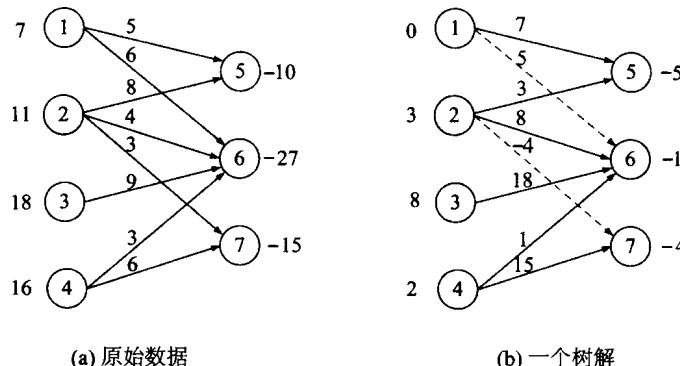


图 3.2.1 二部图: 一个运输问题的网络

如果还是沿用 3.1 节介绍的图示网络单纯形法来求解, 会遇到二部图的弧严重交叉的问题. 为解决这个交叉问题, 代之以表来描述, 称相应的单纯形法为表上作业法. 首先把图 3.2.1 重新绘制成图 3.2.2, 那里弧的尾是源, 弧的头是需求节点, 弧上标的是费用. 简化后见表 3.2.1, 其中“*”表示相应的弧不存在. 只要对源和宿的单纯形乘子, 以及弧上对应的原始流量和既约费用系数进行简单配置, 单纯形法的迭代可以很容易地在表中给出. 比如

图 3.2.1(b) 显示的树解的数据如表 3.2.2 所列, 其中方框内的数表示树解, y_i 表示单纯形乘子, 其余是与非树弧对应的既约费用系数. 当然读者可以验证它并不是最优解, 求该问题的过程也留给读者(习题 3.4).

表 3.2.1 运输问题的数据

供给/需求	10	27	15
7	5	6	*
11	8	4	3
18	*	9	*
16	*	3	6

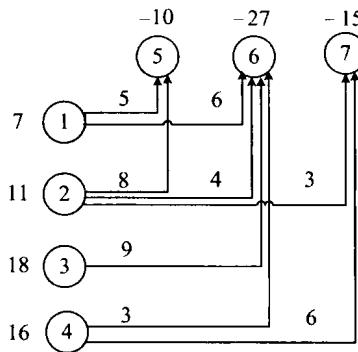


图 3.2.2 运输问题的矩阵表示

表 3.2.2 运输问题的树解

y_i/y_j	-5	-1	-4
0	7	5	*
3	3	8	-4
8	*	18	*
2	*	1	15

称所有供给点和需求点都有弧相连的为 **Hitchcock 运输问题**. 记供给量为 $s_i, i \in \mathcal{S}$, 需求量为 $d_j, j \in \mathcal{T}$, 该运输问题的模型可简化为

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{T}} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j \in \mathcal{T}} x_{ij} = s_i, \quad \forall i \in \mathcal{S} \\ & \sum_{i \in \mathcal{S}} x_{ij} = d_j, \quad \forall j \in \mathcal{T} \\ & x_{ij} \geq 0, \quad \forall i \in \mathcal{S}, j \in \mathcal{T} \end{aligned}$$

下面介绍指派问题, 也称为分配问题, 即给定 n 个人和 n 项任务, 第 i 个人完成任务 j 的费用为 c_{ij} . 指派(assignment)问题是指派每个人去做且只做一项任务, 且每项任务由且只由一个人去完成的费用最小的方案.

如果令

$$x_{ij} = \begin{cases} 1, & \text{若安排第 } i \text{ 个人完成任务 } j \\ 0, & \text{其他} \end{cases}$$

优化的目标为 $\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$, 而指派每个人只做一项任务的约束为 $\sum_{j=1}^n x_{ij} = 1, \forall i$, 每项任务只由一个人完成的约束是 $\sum_{i=1}^n x_{ij} = 1, \forall j$, 从而得线性指派问题的模型为

$$\begin{aligned}
 & \text{minimize} && \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
 & \text{subject to} && \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \\
 & && \sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \\
 & && x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n
 \end{aligned} \tag{3.2.1}$$

如果不要求变量 x_{ij} 取整数, 则指派问题退化为 Hitchcock 运输问题, 其中每个供应点(人)的供应量是 1, 每个需求点(任务)的需求量也是 1. 相应的 Hitchcock 运输问题因此也称为指派问题的线性规划松弛. 易见指派问题的可行解与线性规划松弛的整数可行解一一对应. 运用整性定理(定理 3.1.2), 在应用网络单纯形法求解该线性规划松弛时, 总是会产生一个整数最优解, 这不仅求解了线性规划松弛问题, 而且求解了指派问题本身.

关于指派问题有一个著名的算法, 称为匈牙利算法, 是 Kuhn 于 1955 年提出的(在第 7 章将学习 Kuhn 关于非线性规划基础性的工作——Karush-Kuhn-Tucker 条件), 复杂度为 $O(n^3)$. 其本质上是一种原始-对偶算法: 迭代时每步保持原始可行和对偶可行, 逐步让互补性得到满足.

3.2.2 最大流问题

本小节并不打算介绍最大流问题的一些高效算法, 而只介绍一个极为重要的最大流最小割定理(max-flow min-cut theorem).

设 $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ 是网络, 弧 (i, j) 的容量为 u_{ij} (流的上界), s 和 t 分别是源和宿. 可行流 x 指能使每个节点保持流平衡, 且弧 (i, j) 上的流 x_{ij} 在容量限制之内, 即 $0 \leq x_{ij} \leq u_{ij}$. 流的值 (value of the flow) 定义为 $|x| = \sum_{(s, j) \in \mathcal{A}} x_{sj}$, 它表示从源 s 到宿 t 的流量. 最大流问题(maximum-flow problem)就是找值最大的可行流, 即尽可能多地把流从源 s 路由到宿 t .

虚拟一条从宿 t 回到源 s 的人工弧 (t, s) , 费用是 -1 , 其余弧上的费用为零. 根据节点 s 处的流平衡方程 $x_s = |x|$, 最大流问题可表述为

$$\begin{aligned}
 & \text{maximize} && x_s \\
 & \text{subject to} && \mathbf{Ax} + (\mathbf{e}_t - \mathbf{e}_s)x_s = 0 \\
 & && 0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in \mathcal{A}
 \end{aligned} \tag{3.2.2}$$

其中 \mathbf{e}_i 表示单位矩阵的第 i 列. 注意这个网络流问题带有容量限制, 该线性规划问题的变量是 $x_{ij}, (i, j) \in \mathcal{A}$ 和 x_s . 这里省略了人工流 x_s 的非负约束, 请读者思考这样做的正确性.

\mathcal{G} 的一个 $s-t$ 割($s-t$ cut) $\mathcal{C} = (\mathcal{S}, \mathcal{T})$ 是满足 $s \in \mathcal{S}, t \in \mathcal{T}$ 的节点集 \mathcal{N} 的剖分, 即 $\mathcal{S} \cup \mathcal{T} = \mathcal{N}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$. \mathcal{C} 的割集(cut-set)是弧集 $\{(i, j) \in \mathcal{A} : i \in \mathcal{S}, j \in \mathcal{T}\}$, 若把 \mathcal{C} 的割集去掉, s 与 t 就不连通了. $s-t$ 割的容量定义为 $C(\mathcal{C}) := \sum_{i \in \mathcal{S}, j \in \mathcal{T}} u_{ij}$. 最小割就是极小化 $C(\mathcal{C})$, 即确定 \mathcal{S} 和 \mathcal{T} 使得 $s-t$ 割的容量是最小的.

考虑图3.2.3中的网络，其中弧上所标记的数字的分母表示弧的容量，分子是弧上的流。这个流是可行的，且流值是7。这里白色节点形成子集 S ，灰色节点形成子集 T ；它的割集是虚线边。因为 $s-t$ 割的容量是7，等于流值。下面的最大流最小割定理表明在这个网络中，流值和割的容量都是最优的。

定理3.2.1 (最大流最小割定理) 流的最大值等于 $s-t$ 割的最小容量。

证明 首先写出最大流问题的对偶问题，即

$$\begin{aligned} & \text{minimize} \quad \sum_{(i,j) \in \mathcal{A}} u_{ij} z_{ij} \\ & \text{subject to} \quad y_t - y_s = 1 \\ & \quad y_i - y_j + z_{ij} \geq 0, \quad \forall (i,j) \in \mathcal{A} \\ & \quad z_{ij} \geq 0, \quad \forall (i,j) \in \mathcal{A} \end{aligned} \quad (3.2.3)$$

其中的变量是 y 与 z ，这里 y_i 与最大流问题的第 i 个等式约束对应，而 z_{ij} 与约束 $x_{ij} \leq u_{ij}$ 相对应。对偶问题的第一个约束对应于人工流 x_s 。此外，由矩阵 \mathbf{A} 中与 x_{ij} 对应的列向量为 $\mathbf{e}_i - \mathbf{e}_j$ ，可以推导出约束 $y_i - y_j + z_{ij} \geq 0$ 。

首先由流平衡条件，对于任意的 $s-t$ 割 $\mathcal{C} = (\mathcal{S}, \mathcal{T})$ 和可行流，从 \mathcal{S} 到 \mathcal{T} 的流量等于从 \mathcal{T} 到 \mathcal{S} 的流量加上人工流 x_s ，即

$$x_s = \sum_{i \in \mathcal{S}, j \in \mathcal{T}} x_{ij} - \sum_{i \in \mathcal{T}, j \in \mathcal{S}} x_{ji}$$

这样，对任何可行流和任何 $s-t$ 割 \mathcal{C} 恒有 $x_s \leq C(\mathcal{C})$ ，即弱对偶性成立。

其次，令 $x_{ij}^*, (i,j) \in \mathcal{A}, x_s^*$ 为原始问题(3.2.2)的解， $y_i^*, i \in \mathcal{N}, z_{ij}^*, (i,j) \in \mathcal{A}$ 为对偶问题(3.2.3)的解。现在定义集合 $\mathcal{S}^* = \{i : y_i^* \leq y_s^*\}$ 和 $\mathcal{T}^* = \{j : y_j^* > y_s^*\}$ 。因为 $y_t = y_s^* + 1$ ，所以 $\mathcal{C}^* = (\mathcal{S}^*, \mathcal{T}^*)$ 是一个 $s-t$ 割。

考虑任意的正向弧 (i,j) ，即 $i \in \mathcal{S}^*, j \in \mathcal{T}^*$ 。这时有 $y_i^* \leq y_s^* < y_j^*$ ，从而由对偶可行性有 $z_{ij}^* > 0$ 。再由互补定理，得到 $x_{ij}^* = u_{ij}$ 。然后考虑任意的反向弧 (j,i) ，即 $i \in \mathcal{S}^*, j \in \mathcal{T}^*$ 。这时有 $y_i^* \leq y_s^* < y_j^*$ 。再由对偶可行性 $z_{ij}^* \geq 0$ 有 $y_j^* - y_i^* + z_{ij}^* > 0$ 。再次由互补定理有 $x_{ji}^* = 0$ 。这样就证明了

$$x_s^* = \sum_{i \in \mathcal{S}^*, j \in \mathcal{T}^*} x_{ij}^* - \sum_{i \in \mathcal{T}^*, j \in \mathcal{S}^*} x_{ji}^* = \sum_{i \in \mathcal{S}^*, j \in \mathcal{T}^*} u_{ij} = C(\mathcal{C}^*)$$

且上面的证明过程表明已经找到一个容量和流量相等的割。 ■

需要补充说明的是，最大流问题的对偶问题实际上正是最小割问题的等价模型。如上述证明所指出，对偶问题关于 y 具有平移不变性，从而不妨设 $y_s = 0, y_t = 1$ 。进一步消去变量 z ，对偶问题直接等价于

$$\begin{aligned} & \text{minimize} \quad \sum_{(i,j) \in \mathcal{A}} u_{ij} \max(y_t - y_i, 0) \\ & \text{subject to} \quad y_t = 1, y_s = 0 \end{aligned}$$

令最优解为 $y_i^*, i \in \mathcal{N}$ ，自然地有 $y_s^* = 0, y_t^* = 1$ 。定义指标集合

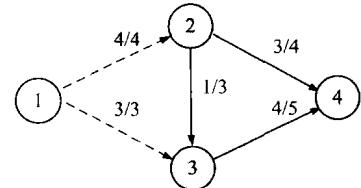


图3.2.3 流值等于一个 $s-t$ 割的容量的网络 ($s=1, t=4$)

$$\begin{aligned}
\mathcal{C}_0 &= \{i : y_i^* = 0\} \\
\mathcal{C}_1 &= \{i : y_i^* = \min_{k \notin \mathcal{C}_0} y_k^*\} \\
\mathcal{C}_2 &= \{i : y_i^* = \min_{k \notin \mathcal{C}_0 \cup \mathcal{C}_1} y_k^*\} \\
&\vdots \\
\mathcal{C}_r &= \{i : y_i^* = \min_{k \notin \mathcal{C}_0 \cup \dots \cup \mathcal{C}_{r-1}} y_k^*\}
\end{aligned}$$

即将所有指标按照最优变量值从小到大分成 $r+1$ 类, 显然 $r \leq m-1$ 且 \mathcal{C}_r 包含宿 t . 接下来证明 $r=1$. 将指标属于 \mathcal{C}_0 或 $\mathcal{C}_1 \cup \dots \cup \mathcal{C}_r$ 中的变量固定在其最优值上, 构造子问题

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j) \in \mathcal{A}} u_{ij} \max(y_j - y_i, 0) \\
\text{subject to} \quad & y_s = 1, y_s = 0 \\
& y_i = y_i^*, \forall i \in \mathcal{C}_0 \cup \mathcal{C}_1 \cup \dots \cup \mathcal{C}_r \\
& 0 \leq y_i = y_j \leq \min_{k \notin \mathcal{C}_0 \cup \mathcal{C}_1} y_k^*, \forall i, j \in \mathcal{C}_1
\end{aligned}$$

不难验证 $y_i^*, i \in \mathcal{C}_1$ 自然地满足人为添加的最后两行约束, 又由于 $y_i^*, i \in \mathcal{C}_1$ 的最优性, 所以它们仍然是这个子问题的最优解. 另一方面, 不难发现, 添加的最后一行约束将使得该问题退化成为一个一维带界约束的线性规划问题, 其最优解显然在(至少可以在)边界达到. 于是 \mathcal{C}_1 可以并入 \mathcal{C}_0 , 或者并入 \mathcal{C}_2 . 重复迭代上述过程, 最终必然得到 $r=1$, 即所有的指标集只有两类, 分别包含源 s 和宿 t , 于是所有的最优变量 y_i 取值为 0 或 1. 换言之, 上述对偶问题等价于

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j) \in \mathcal{A}} u_{ij} \max(y_j - y_i, 0) \\
\text{subject to} \quad & y_s = 1, y_s = 0 \\
& y_i \in \{0, 1\}, i \in \mathcal{N}
\end{aligned}$$

读者不难验证这正是最小割问题的精确模型, 形式上它是个线性整数规划问题, 而这里证明了它本质上是一个线性规划. 基于如上认识, 再由线性规划的对偶定理就可以更深刻地理解最大流最小割定理.

3.2.3 最短路问题

直观而言, 最短路问题是在网络 $(\mathcal{N}, \mathcal{A})$ 中找从节点 $s \in \mathcal{N}$ 到节点 $r \in \mathcal{N}$ 的最短路, 且要求路上连接前后相继节点的弧是同向的, 有时也称有向路. 为了确定最短路, 需要知道每个弧的长度, 记弧 (i, j) 的长度为 c_{ij} .

进一步定义**最短路问题**(shortest path problem)是为 \mathcal{N} 中所有节点找到 $r \in \mathcal{N}$ 的最短路. 当然, 如果这个问题得到解决, 那么上面求从给定节点 s 到给定节点 r 的最短路问题也就迎刃而解. 以下称 r 为**根节点**(root node).

对每个非根节点置一个单位的供给量, 对根节点赋予一个恰当的需求量以保持供需平衡, 即

$$b_i = \begin{cases} 1, & i \neq r \\ 1-m, & i = r \end{cases}$$

弧上的费用定义为该弧的长度. 则最短路问题可以看作最小费用网络流问题. 如果解决了这个网络流问题, 那么从 i 到 r 的最短路可以简单地通过沿着最优生成树追踪从节点 i 到节点 r 的树弧得到, 最短路长度为 $y_i^* - y_r^*$.

需要强调的是, 当应用网络单纯形法求解最短路问题时, 其效率可以进一步提高. 为方便描述, 记 i 到 r 的距离为 v_i , 称其为节点 i 的标号(label).

首先需要研究最短路的最优性条件. 显然 $v_r = 0$. 设从节点 i 出发的弧为 (i, j) , 如果已经得到从节点 j 到 r 的最短路, 那么从节点 i 到根节点 r 的距离是 $c_{ij} + v_j$. 因此, 需要选择从节点 i 出发, 使这个距离之和最小的弧, 即确定 j 使得

$$v_i = \min\{c_{ij} + v_j : (i, j) \in \mathcal{A}\}, \quad \forall i \neq r \quad (3.2.4)$$

称(3.2.4)为动态规划原理(principle of dynamic programming), 也称为 **Bellman** 方程. 在动态规划文献中, 将 v_i 看作节点 i 的函数, 称为值函数(value function).

Bellman 方程确定了总长最短的弧的集合: $\mathcal{T} = \{(i, j) \in \mathcal{A} : v_i = c_{ij} + v_j\}$. 求解这样的隐式方程组通常需要从一个初始点出发, 令其作为右端项, 进行迭代求解. 这种方法称为逐次近似(successive approximations)法. 对应到最短路问题, 其初始化标号为 $v_r^{(0)} = 0, v_i^{(0)} = +\infty, \forall i \neq r$. 由 **Bellman** 方程得到更新公式

$$v_i^{(k+1)} = \begin{cases} 0, & i = r \\ \min\{c_{ij} + v_j^{(k)} : (i, j) \in \mathcal{A}\}, & i \neq r \end{cases}$$

当所有 v_i 不再改变时, 算法终止. 由数学归纳法不难证明: 算法最多迭代 m 次就终止, 只要注意 $v_i^{(k)}$ 表示从 i 出发, 最多经过 k 条弧到达 r 的最短路的值. 每条最短路至多访问每个节点一次, 因此, 该方法最多迭代 m 次. 在每次迭代中只需要检查网络中的每条弧, 最多 n 次加法或比较运算, 于是总的(加法和比较次数)计算复杂度是 $O(nm)$.

从计算过程上看, 该方法从初始标号的估计出发, 迭代校正这些估计, 直至得到最优解, 从而也称为标号校正(label-correcting)算法. 另外, 由于该方法是 Ford 首先利用 **Bellman** 方程来设计的, 也称为 **Bellman-Ford** 法.

如果假定所有的弧长 c_{ij} 都非负, 则还有更快的 **Dijkstra** 算法, 也称为标号设置(label-setting)算法. 迭代过程中反复保存和调用的数据结构是一个已找到最短路的节点的集合 \mathcal{F} 和两个节点的数组. 第一个数组记录标号, 设为 $v_i, i \in \mathcal{N}$; 第二个数组的元素记为 $h_i, i \in \mathcal{N}$, 记录从节点 i 出发沿着最短路访问的下一个节点. 集合 \mathcal{F} 初始化为空集, 每迭代一步添加进一个节点, 且设置其标号为最短路的长度, 这也解释了“标号设置算法”名称的由来. 对已找到最短路的节点, 标号不作任何改变, 仍然保留为它们的长度; 对尚未找到最短路的节点, 标号是一个临时赋值, 表示从该节点经过 \mathcal{F} 中的部分或全部节点而到达根节点的最短路的长度, 若这样的路不存在, 临时标号置为 $+\infty$. 算法的伪码描述见算法 3.2.1.

算法一开始将根节点的标号置为 0, 其余节点的标号置为 $+\infty$; 节点集合 \mathcal{F} 置为空集. 只要还有未找到最短路的节点, 算法就会选择临时标号最小的一个节点 j , 将它添加进 \mathcal{F} , 对 j 的所有尚未找到最短路的相邻节点 i , 如果 $c_{ij} + v_j < v_i$ 成立, 就将 i 的标号更新为 $c_{ij} + v_j$, 对标号得到改变的相邻节点 i 置 $h_i = j$. **Dijkstra** 算法最多迭代 m 次, 每次迭代选择一个尚未找到最短路的节点.

Algorithm 3.2.1 Dijkstra algorithm

```

1: Initialize  $\mathcal{F} = \emptyset$ ,  $v_r = 0$  and  $v_j = +\infty$  for all  $j \neq r$ ;
2: while  $|\mathcal{F}| < m$  do
3:    $j = \arg \min\{v_k : k \notin \mathcal{F}\}$ ;
4:    $\mathcal{F} \leftarrow \mathcal{F} \cup \{j\}$ ;
5:   for  $i \in \mathcal{N}$  such that  $(i, j) \in \mathcal{A}$  and  $i \notin \mathcal{F}$  do
6:     if  $c_{ij} + v_j < v_i$  then
7:        $v_i = c_{ij} + v_j$ ;
8:        $h_i = j$ ;
9:     end if
10:   end for
11: end while

```

3.3 整数线性规划

整数线性规划(integer linear programming)是对线性规划模型作进一步的限制,即要求部分或全部变量取整数,用数学模型可以表述为

$$\begin{aligned}
& \text{minimize} && \mathbf{c}^T \mathbf{x} \\
& \text{subject to} && \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \\
& && x_i \text{ 取整数}, i \in \mathcal{I} \subseteq \{1, 2, \dots, n\}
\end{aligned} \tag{3.3.1}$$

特别地,如果最后一组整性约束换成 $x_i \in \{0, 1\}$,相应的问题称为**0-1线性规划**.指派问题(3.2.1)就是一个0-1线性规划问题.只是由于它非常特殊,所以能用网络单纯形法成功地解决.一般地,并没有类似的整性定理,本节将看到整数规划问题的求解要比线性规划困难得多.

3.3.1 简介

先介绍3个典型的整数规划问题,之后给出几个简单但常用的整数规划建模技巧,让大家体会整数规划模型的特点.

某移动通信公司将某地区划分成 m 个小区(cell).现在需要在这些小区中建立基站(base station),一个基站可以覆盖一个或者多个小区.现有 n 个备选地,已知在第 j 个备选地建站的费用是 c_j ,且知道这些基站可以覆盖的小区.如果基站 j 可以覆盖小区 i ,令 $a_{ij} = 1$;否则,令 $a_{ij} = 0$.现在该移动通信公司面临的决策是如何建立基站使该地区均被覆盖且总的建站费用最少.令 $x_j = 1$,表示在第 j 个备选地建站;否则,令 $x_j = 0$.于是问题可以建模为

$$\begin{aligned}
& \text{minimize} && \sum_{j=1}^n c_j x_j \\
& \text{subject to} && \sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, 2, \dots, m \\
& && x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n
\end{aligned}$$

该问题需要选择基站覆盖 m 个小区,故称为集合覆盖(set-covering)问题.

以简化版本的航空公司调度飞机为例.首先因为市场需要有一些特定的航段(flight leg),其中航段指的是航班从甲地在某时刻起飞,到某时刻在乙地降落.航线(route)是由某些首尾相接的航段组成.这里的航段可以是上午 8:30 离开上海到杭州的航班,也可以是下午 1:00 离开杭州到厦门的航班.一个重要的事实是,这些航段是由市场需求确定的,因此航空公司并不清楚如何将它们组合起来形成航线,使得现有的飞行器可以覆盖它们.根据市场需求分析,需要开通 m 个航段,共有 n 条可选航线,并且知道开通各条航线的费用以及航段和航线之间的隶属关系.现在航空公司面临的问题是,开通哪些航线以涵盖这些既定航段(而且每个航段只归属于某一航线),同时使得总成本最低.令 $x_j = 1$ 表示选择航线 j , $x_j = 0$ 表示不选择航线 j ,航线 j 的费用记为 c_j .如果航段 i 属于航线 j ,令 $a_{ij} = 1$;否则,令 $a_{ij} = 0$.则航空公司的航线选择问题可以建模为

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^n c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^n a_{ij} x_j = 1, \quad i = 1, 2, \dots, m \\ & x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n \end{aligned}$$

由于所有的航线被分割成不同的航段,该模型也称为集合分割(set-partitioning)问题.

一个旅行商从城市 1 出发,去城市 $2, 3, \dots, n$ 推销产品.他必须而且只能访问每个城市一次,最后返回城市 1.设城市两两之间的距离为 c_{ij} ,如何巧妙安排周游使总路程最短的问题称为旅行商问题(Traveling Salesman Problem, TSP).图 3.3.1(a)给出了一个周游.显然,一个周游可以由要经过的所有城市的一个排列给出,即 s_1, s_2, \dots, s_n ,其中 $s_1 = 1$.于是,所有可能的周游数目是 $(n-1)!$.这个数字即使对较小的 n 也是很大,比如 $50! \approx 3 \times 10^{64}$,因而枚举法是不现实的.为快速求解,考虑将它建模成整数规划问题.首先,为每个城市对 (i, j) 引进一个 $0-1$ 决策变量 x_{ij} .其中 $x_{ij} = 1$ 表示弧 (i, j) 在周游上,即从城市 i 出来后下一站访问城市 j ;否则, $x_{ij} = 0$.这样,目标是极小化 $\sum_i \sum_j c_{ij} x_{ij}$.把约束描述清楚需要一点技巧.有的约束描述较简单,如旅行商从城市 i 出来之后只允许进入一个城市,这可以描述为 $\sum_j x_{ij} = 1$.类似地,描述他只允许从城市 j 出来的约束是 $\sum_i x_{ij} = 1$.注意,这两组约束恰好是指派问题的约束.但这两组约束还不足以刻画一个周游,如图 3.3.1(b)所示,有两个甚至以上的子周游也可以不违背这些约束,故需要添加一些约束来排除子周游.

一种做法是,利用周游所含弧的数目和它的节点数目相同这个特点,添加约束簇

$$\sum_{i \in \mathcal{S}, j \in \mathcal{S}} x_{ij} \leq |\mathcal{S}| - 1, \forall \mathcal{S} \subseteq \{1, 2, \dots, n\}, 2 \leq |\mathcal{S}| \leq n - 1$$

来排除子周游,这里 $|\mathcal{S}|$ 表示集合 \mathcal{S} 中的元素个数.因此, n 个城市的旅行商问题可以用 n^2 个变量的整数规划模型表述,即

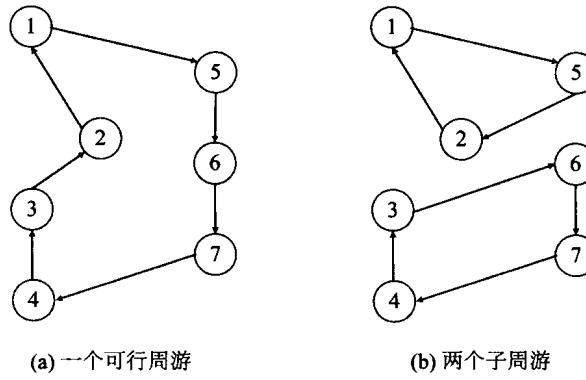


图 3.3.1 旅行商问题的周游

$$\begin{aligned}
 \text{minimize} \quad & \sum_i \sum_j c_{ij} x_{ij} \\
 \text{subject to} \quad & \sum_j x_{ij} = 1, \quad i = 1, 2, \dots, n \\
 & \sum_i x_{ij} = 1, \quad j = 1, 2, \dots, n \\
 & x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \\
 & \sum_{i \in \mathcal{S}, j \in \mathcal{S}} x_{ij} \leq |\mathcal{S}| - 1, \quad \forall \mathcal{S} \subseteq \{1, 2, \dots, n\}, 2 \leq |\mathcal{S}| \leq n - 1
 \end{aligned}$$

需要指出的是,尽管该模型最后一组约束的个数是 n 的指数量级,但是用 3.4.2 小节中的分枝定界法求解时,一开始不必把所有的约束放进去,而是在算法的迭代过程中逐步添加.

下面考虑另一种去除子周游的技巧. 假设给定周游 s_1, s_2, \dots, s_n , 其中 $s_1 = 1$. 令 t_i 为进入城市 i 之前已经经过的城市数目, 即 t_i 是 0 到 $n-1$ 之间的一个整数. 对于图 3.3.1(a) 中所示的周游, $t_1 = 0, t_5 = 1, t_6 = 2$ 等. 如果弧 (i, j) 在周游上, 即 $x_{ij} = 1$, 则有 $t_j = t_i + 1$. 如果 $x_{ij} = 0$, 则满足 $t_j \geq t_i - (n-1)$. 这些约束等价地写为(可参照该例后面的择一约束建模技巧)

$$t_j \geq t_i + 1 - n(1 - x_{ij}), \quad i \geq 1, j \geq 1, i \neq j, t_1 = 0$$

引进这些约束能避免产生不连通的周游. 若不然, 必有一个子周游不包含城市 1, 令 r 为该子周游上的城市对(弧)的数目, 显然 $r \geq 2$. 将该子周游上的所有弧 (i, j) 对应的不等式 $t_j \geq t_i + 1$ 相加, 简单整理可得矛盾式 $0 \geq r$. 因此, n 个城市的旅行商问题还可以用 $n^2 + n$ 个变量的整数规划表述, 即

$$\begin{aligned}
 \text{minimize} \quad & \sum_i \sum_j c_{ij} x_{ij} \\
 \text{subject to} \quad & \sum_j x_{ij} = 1, \quad i = 1, 2, \dots, n \\
 & \sum_i x_{ij} = 1, \quad j = 1, 2, \dots, n \\
 & x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \\
 & t_j \geq t_i + 1 - n(1 - x_{ij}), \quad i, j \geq 1, i \neq j \\
 & t_1 = 0, t_i \in \{1, 2, \dots, n-1\}, \quad i = 2, 3, \dots, n
 \end{aligned}$$

旅行商问题的建模表明, 使用不同的建模技巧所建立问题的规模和约束有很大差异. 在这

里介绍一些实用的整数规划建模技巧. 当两个约束条件择一适用时, 比如 $3x_1 + 2x_2 \leq 8$ 或者 $x_1 + x_2 \leq 6$, 容易验证这等价于

$$\begin{aligned} 3x_1 + 2x_2 &\leq 8 + My \\ x_1 + x_2 &\leq 6 + M(1 - y) \\ y &\in \{0, 1\} \end{aligned}$$

其中 M 是一个充分大的正数.

实际中目标函数有时候不是线性的, 而是带有固定成本的分段线性函数, 比如

$$c(x) = \begin{cases} 0, & x = 0 \\ K + cx, & x \geq 0 \end{cases}$$

如果还假设 x 有一个上界 u , 则上述函数 $c(x)$ 可以写为 $c(x) = Ky + cx$, 其中 y 是额外添加的 $0-1$ 变量. 此外, 还需要添加约束 $x \leq uy$ 和 $x \geq 0$. 对于多项式的非线性目标函数, 可以通过引进额外的变量来线性化. 比如 $x_i, x_j \in \{0, 1\}$, 且需要极小化的目标函数中的系数为正的交叉项 $x_i x_j$ 可以用 y_{ij} 来替代, 但需要 $y_{ij} \geq x_i + x_j - 1$ 和 $y_{ij} \geq 0$. 而系数为负的交叉项 $x_i x_j$ 可用 z_{ij} 来替代, 但需要 $z_{ij} \leq x_i$ 和 $z_{ij} \leq x_j$. 对于次数更高的项可以反复使用如上技巧. 对于更一般的非线性目标函数, 还可以用分段线性函数来逼近. 这里不再进行详细讨论.

3.3.2 对偶理论

考虑如下形式的整数线性规划

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \in X := \{\mathbf{x} : \mathbf{Dx} \geq \mathbf{d}, \mathbf{x} \text{ 各分量取整数}\} \end{aligned} \quad (3.3.2)$$

将这里的 m 个线性约束 $\mathbf{Ax} \geq \mathbf{b}$ 看成是难以处理的约束条件, 而 X 看作是容易处理的约束条件. 对任意 $\lambda \geq 0$, 整数线性规划问题(3.3.2)有下界

$$\varphi(\lambda) = \min_{\mathbf{x} \in X} [\mathbf{c}^T \mathbf{x} + \lambda^T (\mathbf{b} - \mathbf{Ax})] \quad (3.3.3)$$

称该问题是问题(3.3.2)的 Lagrange 松弛, λ 是 Lagrange 乘子, $\varphi(\lambda)$ 是对偶函数(dual function). 最大化这个下界, 得到 Lagrange 对偶问题, 即

$$\max_{\lambda \geq 0} \varphi(\lambda) \quad (3.3.4)$$

类似于线性规划的对偶理论, 有相应的弱对偶定理. 证明留给读者.

定理 3.3.1 (弱对偶性) 对问题(3.3.2)的任意可行解 \mathbf{x} 和问题(3.3.4)的任意可行解 $\lambda \geq 0$, 成立 $\mathbf{c}^T \mathbf{x} \geq \varphi(\lambda)$.

与线性规划对偶定理的一个显著不同之处在于, 建立如下强对偶时需要一些非常强的假设条件. 同样将证明留给读者.

定理 3.3.2 (强对偶性) 对问题(3.3.2)的最优解 \mathbf{x}^* 和问题(3.3.4)的最优解 $\lambda^* \geq 0$, 如果互补条件成立, 即 $\lambda_i^* (\mathbf{b} - \mathbf{Ax}^*)_i = 0, i = 1, 2, \dots, m$, 那么原问题(3.3.2)和对偶问题(3.3.4)的最优值相等.

需要注意的是, 一般的整数线性规划问题不一定满足强对偶定理. 但是, 由弱对偶定理知, 对偶问题总是给出原问题的一个下界, 而且往往易于求解.

例 3.3.1 (Lagrange 对偶) 考虑 0-1 线性规划问题

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & && \mathbf{x} \in \{0,1\}^n \end{aligned}$$

取 $X = \{0,1\}^n$, 对 $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ 进行松弛. 为此引入对偶变量 λ , 对偶函数

$$\begin{aligned} \varphi(\lambda) &= \min_{\mathbf{x} \in \{0,1\}^n} \{ \mathbf{c}^T \mathbf{x} + \lambda^T (\mathbf{b} - \mathbf{A}\mathbf{x}) \} \\ &= \lambda^T \mathbf{b} + \min_{\mathbf{x} \in \{0,1\}^n} (\mathbf{c} - \mathbf{A}^T \lambda)^T \mathbf{x} \\ &= \lambda^T \mathbf{b} + \sum_{i=1}^n \min_{x_i \in \{0,1\}} (\mathbf{c} - \mathbf{A}^T \lambda)_i x_i \\ &= \lambda^T \mathbf{b} + \sum_{i=1}^n \min((\mathbf{c} - \mathbf{A}^T \lambda)_i, 0) \end{aligned}$$

于是对偶问题可以转化为一个线性规划问题, 即

$$\begin{aligned} & \underset{\lambda \geq 0}{\text{maximize}} && \lambda^T \mathbf{b} + \sum_{i=1}^n t_i \\ & \text{subject to} && t_i \leq (\mathbf{c} - \mathbf{A}^T \lambda)_i, \quad t_i \leq 0, \quad i = 1, 2, \dots, n \end{aligned}$$

这样能够很快求解对偶问题, 得到原始问题的下界.

3.4 整数规划的典型方法

将整数线性规划问题(3.3.1)的整性约束去掉, 得到线性规划问题

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\ & && \mathbf{x} \geq 0 \end{aligned} \tag{3.4.1}$$

称式(3.4.1)是整数线性规划问题(3.3.1)的线性规划松弛(linear programming relaxation). 显然, 线性规划松弛总是给出原整数规划问题一个下界. 不难证明: 如果线性规划松弛不可行, 则原问题一定不可行; 如果线性规划松弛的一个最优解恰好是整数解, 那么它也是原整数规划问题的最优解. 指派问题便是一个范例.

如果线性规划松弛的最优解中有分量为分数, 则需要采取进一步的措施, 最简单的就是舍入或者称为圆整(rounding), 将所得非整数解(四舍五入)投影到整数解作为算法输出. 但不幸的是, 这种办法未必奏效, 而且有时候相差很远, 甚至舍入之后的解可能不可行. 这是因为线性规划松弛的最优解总可以在可行解区域的顶点取得, 一个反方向的舍入当然会跑到区域之外!

例 3.4.1 (线性规划松弛) 考虑整数规划问题

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_1 - x_2 \leq 1/2 \\ & && 5x_1 - 3x_2 \geq 5/2 \\ & && x_1, x_2 \text{ 是整数} \end{aligned}$$

图 3.4.1 中阴影部分表示线性规划松弛的可行域, 阴影部分的边界及内部的黑点表示(一部

分)可行的整数解. 易见线性规划松弛的唯一最优解 $x' = (1/2, 0)^T$. 把最优解舍入到邻近的整数点, 得到 $(0, 0)^T$ 或者 $(1, 0)^T$, 它们都不可行, 而真正的最优解 $x^* = (2, 2)^T$. 事实上, 由此例读者不难自行构造这样的例子: 整数线性规划的最优解与其线性规划松弛问题的最优解的距离可以任意大.

3.4.1 Gomory 割平面法

割平面法是求解整数线性规划的一种有效途径, 其基本思想是: 如果线性规划松弛问题的最优解 x' 不是整数解, 则构造一个线性约束, 添加到该松弛问题, 使得所有的整数可行解都满足该约束, 但是 x' 被排除在外, 即该约束会把 x' 割去.

例 3.4.2 (割平面) 考虑整数规划问题

$$\begin{aligned} & \text{minimize} && -x_2 \\ & \text{subject to} && 3x_1 + 2x_2 \leq 6 \\ & && -3x_1 + 2x_2 \leq 0 \\ & && x_1, x_2 \text{ 是非负整数} \end{aligned}$$

如图 3.4.2 所示, 该问题的可行域和线性规划松弛问题的解 $x' = (1, 3/2)^T$. 如果添加线性约束 $x_2 \leq 1$, 则可以割去 x' . 添加的约束称为割平面(cutting plane).

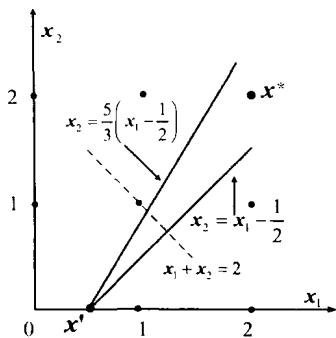


图 3.4.1 线性规划松弛

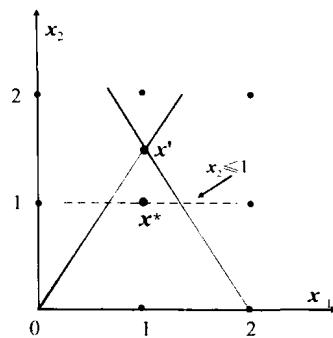


图 3.4.2 Gomory 割平面

求解添加割平面之后的线性规划问题, 得新的最优解, 如果还不是整数解, 继续添加合适的割平面, 迭代下去, 直到算法终止. 经典的 Gomory 割平面是由线性规划的单纯形表产生割平面. 它的优点是通用性, 即可以求解任何整数线性规划问题; 缺点是基于这种割平面的方法很慢.

下面介绍 Gomory 割平面法. 首先用单纯形法求解线性规划松弛问题, 得最优解 x' . 不妨假设最优基变量为 x_1, x_2, \dots, x_m , 相应的规范形为

$$x_i + \sum_{j=m+1}^n y_{ij} x_j = y_{i0}, \quad i = 1, 2, \dots, m \quad (3.4.2)$$

如果所有的 $y_{10}, y_{20}, \dots, y_{m0}$ 都是整数, 那么单纯形表给出了整数最优解; 否则, 不妨假设 y_{10} 不是整数. 根据式(3.4.2)中 $i=1$ 所得等式, 考虑不等式

$$x_1 + \sum_{j=m+1}^n \lfloor y_{1j} \rfloor x_j \leq \lfloor y_{10} \rfloor \quad (3.4.3)$$

其中 $\lfloor a \rfloor$ 表示不超过 a 的最大整数, 比如 $\lfloor 1.5 \rfloor = 1$, $\lfloor -1.5 \rfloor = -2$. 将式(3.4.2) ($i=1$ 时) 与式(3.4.3) 作差, 有

$$\sum_{j=m+1}^n (y_{1j} - \lfloor y_{1j} \rfloor) x_j \geq y_{10} - \lfloor y_{10} \rfloor \quad (3.4.4)$$

容易验证所有的整数可行解都自动满足式(3.4.4). 现在考虑 x' . 将 $x'_{m+1} = \dots = x'_n = 0$ 代入式(3.4.4), 左边为 0 而右边 $y_{10} - \lfloor y_{10} \rfloor > 0$. 故将它添加到线性规划松弛问题中, 一定会割去线性规划松弛的最优解 x' . 称这种类型的线性不等式约束是 Gomory 割平面.

例 3.4.3 (Gomory 割平面法) 用 Gomory 割平面法求解例 3.4.2. 首先添加松弛变量将松弛问题转化为标准形, 以变量 x_3 和 x_4 为基变量建立第 1 张单纯形表

	x_1	x_2	x_3	x_4	$B^{-1}b$
	3	2	1	0	6
	-3	2	0	1	0
r^T	0	-1	0	0	0

经过两次转轴, 得到最优单纯形表

	x_1	x_2	x_3	x_4	$B^{-1}b$
	1	0	$\frac{1}{6}$	$-\frac{1}{6}$	1
	0	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{2}$
r^T	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{2}$

得最优解 $x^{(0)} = (1, 3/2)^T$, 最优值 $z_0 = -3/2$. 因为 x_2 不是整数, 所以从单纯形表第二行导出 $x_2 \leq 1$ (如图 3.4.2 所示), 从而得 Gomory 割平面为 $\frac{1}{4}x_3 + \frac{1}{4}x_4 \geq \frac{1}{2}$. 添加盈余变量 x_5 , 得单纯形表

	x_1	x_2	x_3	x_4	x_5	$B^{-1}b$
	1	0	$\frac{1}{6}$	$-\frac{1}{6}$	0	1
	0	1	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{3}{2}$
	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	1	$-\frac{1}{2}$
r^T	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{3}{2}$

该单纯形表对应的基本解虽然不是原始可行的, 却是对偶可行的, 所以可以利用对偶单纯形法求解. 迭代一步(按对偶单纯形法转轴一次), 有

x_1	x_2	x_3	x_4	x_5	$\mathbf{B}^{-1} \mathbf{b}$
1	0	0	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
0	1	0	0	1	1
0	0	1	1	-4	2
r^T	0	0	0	0	1

得最优解 $\mathbf{x}^{(1)} = (3/2, 1)^T$, 最优值 $z_1 = -1$. 因为第一个分量不是整数, 所以从单纯形表第一行导出 $x_1 - x_4 \leq 0$, 从而得 Gomory 割平面为 $\frac{2}{3}x_4 + \frac{2}{3}x_5 \geq \frac{2}{3}$. 添加盈余变量 x_6 , 得单纯形表

x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{B}^{-1} \mathbf{b}$
1	0	0	$-\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{2}{3}$
0	1	0	0	1	0	1
0	0	1	1	-4	0	2
0	0	0	$-\frac{2}{3}$	$-\frac{2}{3}$	1	$-\frac{2}{3}$
r^T	0	0	0	0	1	0

利用对偶单纯形法求解, 迭代一步有

x_1	x_2	x_3	x_4	x_5	x_6	$\mathbf{B}^{-1} \mathbf{b}$
1	0	0	0	1	$-\frac{1}{2}$	1
0	1	0	0	1	0	1
0	0	1	0	-5	$\frac{3}{2}$	1
0	0	0	1	1	$-\frac{3}{2}$	1
r^T	0	0	0	0	1	0

得最优解 $\mathbf{x}^{(3)} = (1, 1)^T$. 此即整数规划问题的最优解, 最优值 $z^* = -1$.

3.4.2 分枝定界法

本小节简单介绍精确求解整数线性规划问题的主流算法——分枝定界法 (branch-and-bound method). 目前大部分求解整数规划问题的商业软件 (如 CPLEX 和 BARON) 都是基于分枝定界框架, 关于二者的简要介绍见 3.5 节.

分枝 以 0-1 线性规划为例, 对某 0-1 变量 x_i , 或者 $x_i = 0$ 或者 $x_i = 1$, 分别固定它们, 就得到两个不同的 $n-1$ 维的子问题. 不断地这样分下去, 就得到一个二叉树: 把原问题看作根节点, 把固定 m 个变量后的子问题看成是第 m 层的节点, 第 n 层的所有 2^n 个子问题便是叶子节点. 对于一般的整数线性规划, 某一整型变量 x_i 也可以类似二分叉, 比如 $x_i \leq 3$ 和 $x_i \geq 4$.

单纯只进行分枝过程,这便是枚举(enumeration)法.智能一点的枚举法还会执行剪枝操作:在某一(中间)节点处停止分枝,即该节点提前归为叶子节点.这种情况发生在:该节点处子问题不可行,或者该子问题的线性规划松弛的解满足整性要求.

分枝定界法又添加了定界(bounding)操作,即为当前分枝上的(极小化)整数线性规划问题确定一个下界.在前面的内容中我们学习了两种定界技巧:分别求解整数规划的对偶问题和线性规划松弛问题.定界的目的是为了更有效地剪枝(pruning),即若某节点子问题的下界大于或等于某已知的可行解的目标函数值,则停止分枝.这是因为,已知的可行解的目标值一定比这一枝上的最优值要更小一些,再枚举这一枝则是徒劳的.从这个意义上讲,分枝定界法是一种智能的枚举法.以如下的整数线性规划问题为例来说明.

例 3.4.4 (分枝定界法) 利用分枝定界法求解

$$\begin{aligned} & \text{minimize} \quad x_1 + 2x_2 \\ & \text{subject to} \quad 4x_1 + 2x_2 \geq 5 \\ & \quad x_1, x_2 \geq 0 \\ & \quad x_1, x_2 \text{ 为整数} \end{aligned}$$

用 P_i 表示节点 i 处的子问题的线性规划松弛问题.首先求解线性规划松弛问题,即 P_0 ,其可行域如图 3.4.3(a)所示,得最优解 $x' = (1.25, 0)^T$. 整性约束要求 $x_1 \leq 1$ 或者 $x_1 \geq 2$.下面分别考虑这两种情况:添加约束 $x_1 \leq 1$ 的子问题用节点 1 表示,另一个用节点 2 表示.二者的下界 $L = 1.25$,它们各自的线性规划松弛问题的可行域如图 3.4.3(b)所示.首先考察 P_1 .从图 3.4.3(b)中可以看出最优解是 $(1, 0.5)^T$,相应最优值是 2.此时需要对 x_2 进行分枝,即给节点 1 表示的问题分别添加约束 $x_2 \leq 0$ 和 $x_2 \geq 1$,用节点 3 和 4 表示相应的问题.二者的下界

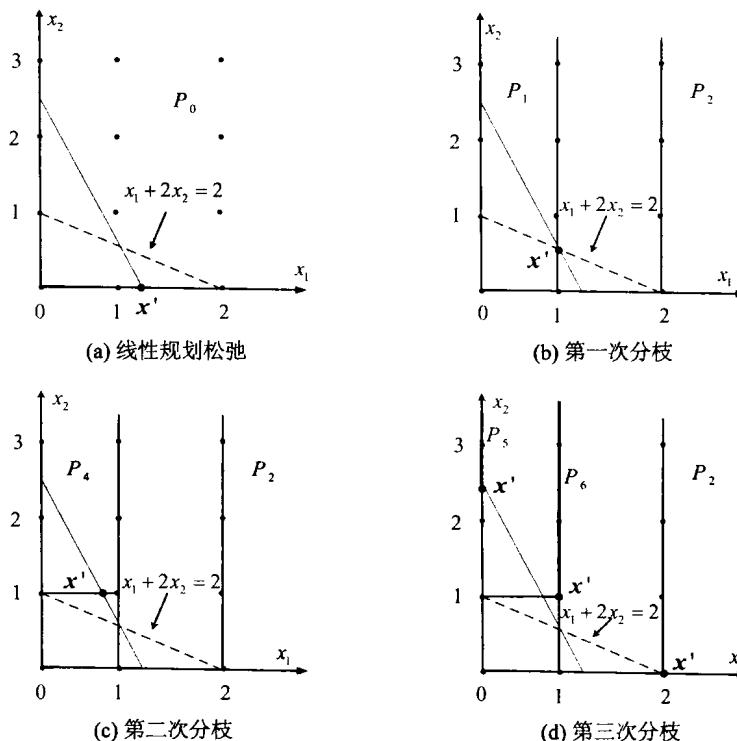


图 3.4.3 分枝定界法的分枝

$L=2$.

先考虑广度优先法(breadth-first search), 即优先求解树的同一层(即到根节点的长度相等的节点)对应的所有子问题. 对于该问题, 在节点2,3,4中优先考虑节点2. 为此考虑节点2对应的子问题. 由图3.4.3(b)得解 $x'=(2,0)^T$. 因为该松弛问题的最优解满足整性要求, 因而节点2表示的整数规划子问题已经求解, 停止分枝, 即在 P_2 处剪枝, 节点2成为叶子节点. 在图3.4.4中, 用正方形节点表示这种情况. 此时, 算法获得第一个可行解, 将其记录为当前最好解(best-so-far solution) \hat{x} 和最好值 \hat{f} . 当然, 后继分枝过程若产生更好的解, 则更新该记录. 然后讨论 P_3 . 因为它的下界大于等于当前最好值 \hat{f} , 执行剪枝, 从而节点3成为叶子节点; 同理剪掉节点4. 在图3.4.4中, 用浅灰色的圆形节点表示这种情况. 至此, 解完所有的子问题, 算法返回当前最好的解作为整数规划问题的最优解. 同时, 算法搭建了一个代表子问题的树, 称为枚举树(enumration tree), 如图3.4.4(a)所示. 节点旁标注的数字是为相应子问题确定的下界.

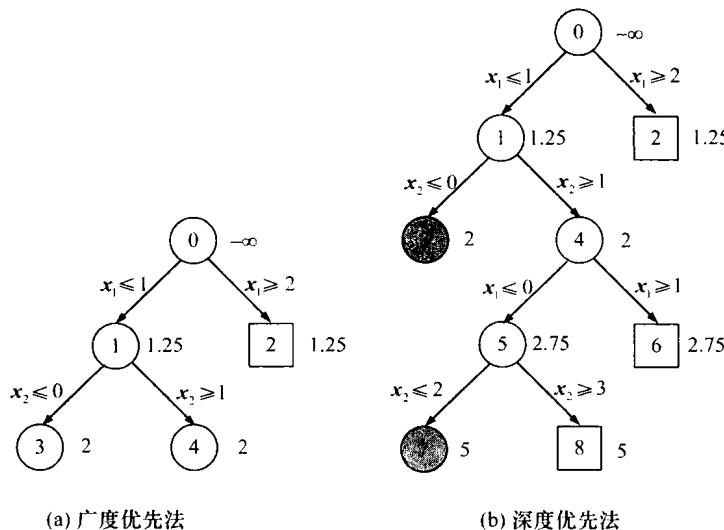


图3.4.4 分枝定界法的枚举树

下面考虑深度优先法(depth-first search), 即优先考虑底层子问题, 然后逐层向上回溯, 回溯过程中仍然保持底层优先. 对于该问题, 在节点2,3,4中考虑节点3. 在问题 P_3 中, 因为 $x_2 \leq 0$, 注意此时 $4x_1 + 2x_2 \leq 4 + 0 = 4$, 不可能再满足约束 $4x_1 + 2x_2 \geq 5$, 从而节点3代表的子问题不可行, 它也成为叶子节点, 不再分枝. 在图3.4.4(b)中, 用深灰色的圆形节点表示这种情况. 接下来回溯到同一层右边的子问题, 即节点4代表的子问题. P_4 的可行域如图3.4.3(c)所示, 得最优解 $(0.75, 1)^T$ 和最优值2.75. 注意此时没有当前最好解, 由深度优先法对 x_1 分枝, 左边的子问题用节点5表示, 它添加的是 $x_1 \leq 0$, 容易验证 P_5 的最优解是 $(0, 2.5)^T$, 最优值为5. 此时对 x_2 进行分枝, 分别用节点7和8表示添加约束 $x_2 \leq 2$ 和 $x_2 \geq 3$ 的子问题. 易见 P_7 不可行, 从而剪枝, 得到叶子节点. 现在考虑 P_8 , 解为 $(0, 3)^T$, 值为6. 此时由于 P_8 的解已经是整数点, 所以节点8成为叶子节点而不需要再分. 此外, 这是找到的第一个当前最好解, 把它记录下来. 这样又一次探索到最底层, 然后回溯到同一层右边的子问题, 即节点6, 它添加的是 $x_1 \geq 1$. 由图3.4.3(d)知 P_6 的最优解是 $(1, 1)^T$, 最优值为3. 因为 $3 < 6$, 所以需要将当前最好

解更新为 $(1,1)^T$, 对应的最好值为 3. 节点 6 已经成为叶子节点, 故而不需要再分. 逐层向上回溯, 下一个需要考虑节点 2. 求解问题 P_2 , 得最优解 $(2,0)^T$ 和最优值 2. 再一次将当前最好解更新为 $(2,0)^T$, 对应的最好值为 2. 因为节点 2 已经是叶子节点, 且所有的节点都考虑完了, 所以搜索过程结束, 返回当前最好解 $(2,0)^T$ 作为全局最优解.

在上例中发现广度优先法比深度优先法有优势, 但注意这只是一个个例. 读者可以发现: 对本例而言, 广度优先法的效果和执行右边分枝优先的深度优先法的效果一样. 另外也可以看到, 同一层之间先考虑左边的一枝还是先考虑右边一枝, 效果是不一样的. 一般而言, 二者没有好坏之分, 人们通常选用左边优先, 但这只是一个习惯.

在一般情况下, 深度优先法相对广度优先法显得更有优势, 这至少有 3 个理由. 首先, 一般整数解都处于枚举树的底层, 即距离根节点很远, 而尽早探测到整数解有两个好处: 一是能找到一个可行解(提前强制终止算法时, 有一个次优的解总比什么都没有强); 二是一旦确定一个可行整数解就可以帮助剪枝, 若后继节点的下界比当前最好解的目标值还要差, 该节点就不需要再分枝了. 第二个理由是深度优先法的程序相对容易编写. 第三个理由是最重要的, 在从枚举树的某一层到其下一层的迭代中, 新的线性规划子问题是通过针对上层线性规划问题的某个特殊的变量添加一个上界/下界而得到. 以 P_0 和 P_1 为例来具体说明. 对 P_0 的线性不等式约束添加盈余变量 $x_3 \geq 0$ 使 $4x_1 + 2x_2 - x_3 = 5$, 其最优单纯形表为(x_1 为基变量)

	x_1	x_2	x_3	$B^{-1}b$
	1	$\frac{1}{2}$	$-\frac{1}{4}$	$\frac{5}{4}$
r^T	0	$\frac{3}{2}$	$\frac{1}{4}$	$-\frac{5}{4}$

给 P_0 添加约束 $x_1 \leq 1$ 得到 P_1 . 对该约束引进一个松弛变量 $x_4 \geq 0$, 使得不等式约束变为 $x_1 + x_4 = 1$, 这样就得到 P_1 的单纯形表

	x_1	x_2	x_3	x_4	$B^{-1}b$
	1	$\frac{1}{2}$	$-\frac{1}{4}$	0	$\frac{5}{4}$
	0	$-\frac{1}{2}$	$\frac{1}{4}$	1	$-\frac{1}{4}$
r^T	0	$\frac{3}{2}$	$\frac{1}{4}$	0	$-\frac{5}{4}$

容易看出, 该表给出的解是对偶可行, 但不是原始可行的. 因此, 应用对偶单纯形法可以很快找到解. 根据对偶单纯形法, 变量 x_2 进基, 没有变量出基(因为基变量个数相应增加 1), 转轴之后得到单纯形表

	x_1	x_2	x_3	x_4	$B^{-1}b$
	1	0	0	1	1
	0	1	$-\frac{1}{2}$	-2	$\frac{1}{2}$
r^T	0	0	1	3	-2

这个表给出了 P_1 的最优解. 一般地, 对偶单纯形法可能需要不只一步的迭代, 但是, 还是期望很快会获得新问题的最优解.

需要说明的是: 对于规模非常小的问题, 可以用图示法; 对于规模大一点的问题, 需要设计算法编制程序交给计算机运行. 基于深度优先的求解整数规划的分枝定界法的伪码见算法 3.4.1. 最后要指出的是, 割平面可以加强该算法中步骤 18 的线性规划松弛下界, 因而分枝定界法可以与割平面法有效结合起来, 这称为分枝割 (branch-and-cut) 法.

Algorithm 3.4.1 Branch and bound method for integer programming problem

```

1: Initially  $\hat{f} = \infty$ , the continuous problem is put in the stack with  $L = -\infty$ ;
2: while there are problems in the stack do
3:   take the top problem from the stack;
4:   if  $L \geq \hat{f}$  then
5:     reject the problem;
6:   else
7:     try to solve the problem;
8:     if no feasible point exists then
9:       reject the problem;
10:    else
11:      let the solution be  $\mathbf{x}'$  with  $f'$ ;
12:      if  $f' \geq \hat{f}$  then
13:        reject the problem;
14:      else
15:        if  $\mathbf{x}'$  is integer feasible then
16:          set  $\hat{\mathbf{x}} = \mathbf{x}'$ ,  $\hat{f} = f'$ ;
17:        else
18:          select an integer variable  $i$  such that  $\lfloor x'_i \rfloor < x'_i$ , create two new problems by branching on  $x_i$ , place
           these on the stack with lower bound  $L = f'$  (or a tighter lower bound derived from  $f'$ );
19:        end if
20:      end if
21:    end if
22:  end if
23: end while
24: return  $\hat{\mathbf{x}}$  and  $\hat{f}$  as  $\mathbf{x}^*$  and  $f^*$ 

```

3.5 评注与参考

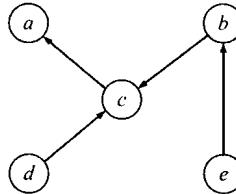
网络理论起源于图论. 最大流最小割定理是图论和网络流中最重要的结论之一, 由 Ford 和 Fulkerson 于 1956 年提出. 整数规划始于 1958 年, 是由 Gomory 提出割平面法之后形成独立分支而发展起来的. 因而 Gomory 被称为整数规划之父. 分枝定界法是由 Karp 在 20 世纪 60 年代发明的, 当时他用这种方法成功求解含有 65 个城市的旅行商问题, 打破当时的记录

(当时记录由 Dantzig、Fulkerson 和 Johnson 求解包含 49 个城市的旅行商问题保持).

CPLEX(IBM ILOG CPLEX optimization studio 的简称)是 IBM 公司开发的一种优化软件包, 其中的单纯形法是用 C 语言实现的, 并因此得名. BARON(Brand And Reduce Optimization Navigator)是一种致力于找到非凸优化问题全局最优解的计算系统^[31]. 它的特点是将区间分析和“既约集成库中的对偶性”与加强的分枝定界概念结合起来寻优, 这也是它名字的由来.

习题 3

3.1 考虑图 3.1.1 所示网络流问题, 令 $T = \{(b, c), (c, a), (d, c), (e, b)\}$, 即如下生成树



请以 e 为根节点, 完成以下工作:

- (a) 求每个树弧上的原始流量;
- (b) 求与每个节点对应的单纯形乘子;
- (c) 求与每个非树弧对应的既约费用系数.

3.2 证明 m 个节点的生成树有 $m-1$ 条弧.

3.3 假定 A 的一个子方阵非奇异, 证明子方阵的列对应的弧构建了一棵生成树.

3.4 考虑表 3.2.1 所给的运输问题.

- (a) 表 3.2.2 给出的树解可行吗? 对偶可行吗?
- (b) 求解该运输问题.

3.5 背包问题(knapsack problem). 一个背包远行的人要极大化他能带上物品的价值. 设第 j 件物品占用 a_j 单位的背包空间, 价值为 c_j ; 背包的容量记为 b . 问题模型为

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^n c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^n a_j x_j \leq b \\ & x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n \end{aligned}$$

这个问题可以说明分枝定界算法的计算复杂度是很高的. 简单起见, 假定所有的 $c_j = c$, 所有的 $a_j = 2$, 背包容量 $b = n$.

- (a) 当 n 为奇数和偶数时, 最优解分别是什么?
- (b) 当 n 为 3 和 5 时, 分枝定界算法需要考虑多少个子问题?

3.6 车辆调度问题(vehicle routing problem). 考虑一些车辆从配送中心出发沿着线路给一些零售商送货. 给定一个可行的线路集合和对应的成本, 讨论如何把本问题建模成集合

分割问题.

- 3.7 二次指派问题 (quadratic assignment problem). 令 \mathcal{F} 是 n 个工厂的集合, \mathcal{C} 是 n 个城市的集合. 要求在每一城市中设置且只设置一个工厂, 并要使工厂两两之间总的通信费用最小. 每对工厂 (i, k) 之间一定时间内通信的次数为 t_{ik} , 每对城市 (j, l) 之间的距离为 d_{jl} . 通信费用 $c_{ijkl} = t_{ik}d_{jl}$.
- (a) 试给该问题建立带非线性目标的整数规划模型.
- (b) 将上述模型中的非线性目标函数线性化.
- 3.8 考虑例 3.4.4 的枚举树上节点 2 的线性规划松弛问题 P_2 . 请从 P_0 的最优单纯形表开始, 利用对偶单纯形法求解 P_2 .
- 3.9 对线性规划

$$\begin{aligned} \text{minimize} \quad & -x_1 - 2x_2 \\ \text{subject to} \quad & -2x_1 + 2x_2 \leq 3 \\ & 2x_1 + 2x_2 \leq 9 \end{aligned}$$

分 (i) 无整数限制, (ii) x_1 为整数, (iii) x_1, x_2 均为整数 3 种情况, 用图解法求解相应的问题, 并给出用分枝定界法求解 (iii) 的过程, 画出枚举树.

第 4 章 无约束优化：基础

本书的第 4~6 章将集中考虑无约束极小化问题

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} f(\mathbf{x}) \quad (4.0.1)$$

其中目标函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

先看一个来自计算分子生物学的例子. 已知一个分子的化学式, 需要进一步确定该分子的几何结构. 人们这方面已经有一些实验技术, 比如 X 射线结晶、核磁共振等等. 这些实验会提供一些非常有用的信息, 比如该分子中全部或部分原子对的距离等. 接下来解析该分子的几何结构时, 往往需要借助于数值技术.

假定一共有 n 个原子, 第 i 个原子的三维坐标记为 $\mathbf{y}_i \in \mathbb{R}^3$, 它们组成向量 $\mathbf{x} \in \mathbb{R}^{3n}$, 赋予其一个势能 $f(\mathbf{x})$, 也称为构象能. 最小化势能函数便可以确定出分子中各原子的坐标, 该优化问题便是典型的无约束优化问题, 形如 $\min_{\mathbf{x} \in \mathbb{R}^{3n}} f(\mathbf{x})$. 势能有很多种不同的具体表达式, 分别对应不同的优化问题. 它们一般都是很多项的求和, 每一项对应于某个原子对, 常用的有键长公式

$$L_{ij}(\mathbf{y}_i, \mathbf{y}_j) = w_{ij}(\|\mathbf{y}_i - \mathbf{y}_j\| - d_{ij})^2$$

和 Van der Waals 公式

$$V_{ij}(\mathbf{y}_i, \mathbf{y}_j) = v_{ij} \left(\frac{\delta_{ij}}{\|\mathbf{y}_i - \mathbf{y}_j\|} \right)^6 - w_{ij} \left(\frac{\delta_{ij}}{\|\mathbf{y}_i - \mathbf{y}_j\|} \right)^{12}$$

这里的 v_{ij}, w_{ij} 和 δ_{ij} 都是已知的, 它们依赖于原子对 (i, j) ; $\|\mathbf{y}_i - \mathbf{y}_j\|$ 代表原子 i 到 j 的键长; d_{ij} 是实验测出来的键长.

就该例而言, 目标函数是很复杂的, 寻找其全局最优解到现在还在困扰着无数计算科学家. 因为全局最优解一定是局部最优解, 人们退而求其次, 寻求一个局部最优解. 需要注意的是, 学完本章大家会发现, 大部分算法迭代收敛的解甚至都不能保证是局部解, 而只能是某个更次的稳定点. 在想做和能做之间存在如此大的差距, 激发一批又一批的计算科学家和优化专家为之而不懈奋斗.

4.1 极小点的条件

本节的主要目的是指出并讨论局部极小点 \mathbf{x}^* 应该满足的一些简单条件.

4.1.1 局部极小点的条件

考察过 \mathbf{x}^* 的任一条直线 $\mathbf{x}(\alpha) = \mathbf{x}^* + \alpha \mathbf{p}$, 则 $\alpha = 0$ 必是函数 $\phi(\alpha) := f(\mathbf{x}(\alpha))$ 的局部极小点, 见图 4.1.1, 因此 $\phi(\alpha)$ 在 $\alpha = 0$ 有零斜率 (zero slope) 和非负曲率 (nonnegative curvature). 再由式 (1.4.5) 和式 (1.4.6) 知, 对所有的 \mathbf{p} 有, $\mathbf{p}^\top \mathbf{g}^* = 0$ 和 $\mathbf{p}^\top \mathbf{G}^* \mathbf{p} \geq 0$. 由于这些条件是 \mathbf{x}^* 为局部极小点时所蕴含的, 因此也是局部极小点的必要条件, 可以等价地表示为

$$\mathbf{g}^* = \mathbf{0} \quad (4.1.1)$$

和

$$\mathbf{G}^* \text{ 半正定} \quad (4.1.2)$$

式(4.1.1)成立是因为 $\mathbf{p}^T \mathbf{g}^* = 0$ 对所有 \mathbf{p} 成立当且仅当 $\mathbf{g}^* = \mathbf{0}$, 由于该条件仅包含一阶导数, 故称为一阶必要条件. 下面给出 \mathbf{x}^* 为局部极小点的一个充分条件.

定理 4.1.1 (二阶充分条件) 若在点 \mathbf{x}^* 处式(4.1.1)成立, 且 \mathbf{G}^* 为正定矩阵, 即

$$\mathbf{p}^T \mathbf{G}^* \mathbf{p} > 0, \quad \forall \mathbf{p} \neq \mathbf{0} \quad (4.1.3)$$

则点 \mathbf{x}^* 是严格局部极小点.

证明 用反证法. 假设 \mathbf{x}^* 不是严格局部极小点, 则 $\forall k, \exists \mathbf{x}^{(k)} \in N\left(\mathbf{x}^*, \frac{1}{k}\right), \mathbf{x}^{(k)} \neq \mathbf{x}^*$ 且满足

$$f^{(k)} \leq f^* \quad (4.1.4)$$

序列 $\left\{ \frac{\mathbf{x}^{(k)} - \mathbf{x}^*}{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|} \right\}$ 有界, 故有聚点. 不妨设 $\left\{ \frac{\mathbf{x}^{(k)} - \mathbf{x}^*}{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|} \right\}$ 收敛于 \mathbf{p} . 则 $\mathbf{p} \neq \mathbf{0}$ 且

$$f^{(k)} = f^* + \frac{1}{2}(\mathbf{x}^{(k)} - \mathbf{x}^*)^T \mathbf{G}^* (\mathbf{x}^{(k)} - \mathbf{x}^*) + o(\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2)$$

这里利用了 $\mathbf{g}^* = \mathbf{0}$. 上式两边同时除以 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2$, 再结合式(4.1.4), 有 $\mathbf{p}^T \mathbf{G}^* \mathbf{p} \leq 0$. 这与充分条件(4.1.3)矛盾, 因此假设不成立. 故 \mathbf{x}^* 是严格局部极小点. ■

该充分条件在数值上易于检验, 使用方便. 以 Rosenbrock 函数(1.4.2)为例, 在点 $\mathbf{x}^* = (1, 1)^T$ 处

$$\mathbf{g}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{G}^* = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}$$

可验证 \mathbf{G}^* 是正定矩阵(见下面的讨论), 由定理 4.1.1 知 \mathbf{x}^* 是严格局部极小点(事实上, 由于 $f^* = 0$, 而 $f(\mathbf{x}) \geq 0$, 所以 \mathbf{x}^* 还是全局极小点).

比较必要条件和充分条件, 因为存在使得曲率为零的向量 \mathbf{p} , 即 $\{\mathbf{p} : \mathbf{p} \neq \mathbf{0}, \mathbf{p}^T \mathbf{G}^* \mathbf{p} = 0\} \neq \emptyset$, 所以二者会存在间隙. 以函数 $f(\mathbf{x}) = x^3$ 和 $f(\mathbf{x}) = x^4$ 为例, 在 $\mathbf{x}^* = 0$ 处, 一阶条件均满足, 但都不满足充分条件. 显然 $\mathbf{x}^* = 0$ 是第二个函数的极小点, 但不是第一个函数的极小点. 从而对于满足二阶必要条件, 但不满足二阶充分条件的点, 需要具体函数具体分析.

正定矩阵的概念大家并不陌生, 然而条件(4.1.3)不易进行检验, 但是可借助下面的几个等价条件来检验.

- (i) \mathbf{G}^* 的所有特征值(或最小特征值)都大于零,
- (ii) \mathbf{G}^* 的 Cholesky 分解 $\mathbf{L}\mathbf{L}^T$ 存在, 且 $l_{ii} > 0$,
- (iii) \mathbf{G}^* 的 $\mathbf{L}\mathbf{D}\mathbf{L}^T$ 分解存在, 其中 $l_{ii} = 1$, 且 $d_{ii} > 0$,
- (iv) \mathbf{G}^* 的所有顺序主子式均大于零,

其中 \mathbf{L} 是下三角矩阵, $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n)$ 表示对角线元素依次为 d_1, d_2, \dots, d_n 的对角矩阵. 当 n 较小时(比如 $n \leq 3$), 条件(iv)最易于检验. 以上例中的 \mathbf{G}^* 为例, 有

$$\det(802) > 0, \det(\mathbf{G}^*) = 400 > 0$$

但一般来说, 条件(ii)和(iii)比较有效, 并且可以同时有效求解以 \mathbf{G}^* 为系数矩阵的线性方程组, 详见附录 A 中的算法 A.7.2.

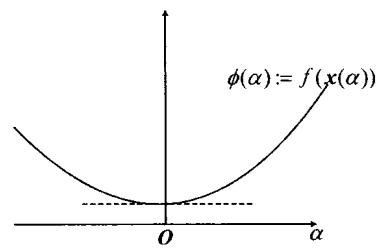


图 4.1.1 零斜率和非负曲率

事实上,多数优化方法在计算过程中使目标值逐渐减小,最后收敛于使 $\mathbf{g}(\mathbf{x}') = \mathbf{0}$ 的 \mathbf{x}' . 称这样的点是稳定点(stationary point),其有可能不是局部极小点. 图 4.1.2 给出了不同类型的稳定点,而图 4.1.3 给出了与这些稳定点所对应的函数的等值线. 用经过点 (\mathbf{x}^*, f^*) ,且垂直于 \mathbf{x} 平面的平面去截 3 个图形(如图 4.1.2 所示),所得曲线在 \mathbf{x}^* 处的斜率均为零,但曲率不尽相同. 其中,(a)中截图所得曲线的曲率为正,(b)中截图所得曲线的曲率为负,(c)中截图所得曲线的曲率有正有负. 因此,通常极小点处的 Hessian 阵是正定的,极大点处的 Hessian 阵是负定的,而鞍点处的 Hessian 阵是不定的,即特征值有正有负.

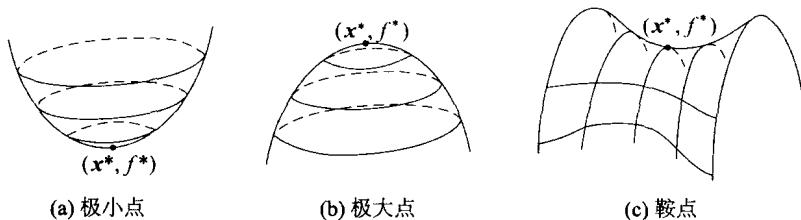


图 4.1.2 稳定点的类型

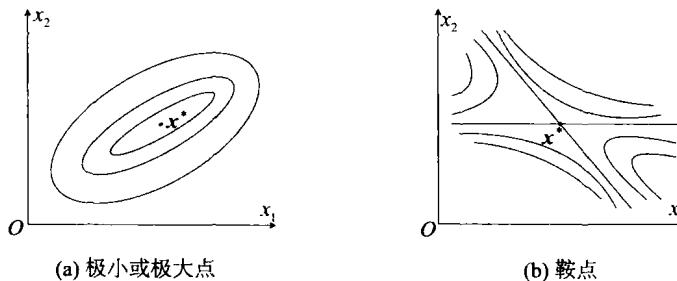


图 4.1.3 稳定点处的等值线

4.1.2 凸性与全局极小点

除了第 2 章介绍的凸集外,另一个关于凸性的基本概念是凸函数. 为简单起见,本章讨论仅限于定义在凸集 C 上的连续函数. 这种假设下,凸函数可定义为: 对任意 $\mathbf{x}_0, \mathbf{x}_1 \in C$ 应有

$$f_\theta \leqslant (1 - \theta)f_0 + \theta f_1, \quad \forall \theta \in (0, 1) \quad (4.1.5)$$

其中 f_θ 表示 $f(\mathbf{x}_\theta)$,而 \mathbf{x}_θ 定义为

$$\mathbf{x}_\theta = (1 - \theta)\mathbf{x}_0 + \theta\mathbf{x}_1 \quad (4.1.6)$$

考虑连接 $f(\mathbf{x})$ 的图形上的点 (\mathbf{x}_0, f_0) 与 (\mathbf{x}_1, f_1) 的弦,不等式(4.1.5)的几何意义是凸函数的图形总是位于弦的下方或弦上,如图 4.1.4(a)所示.

如果 C 为开集, $f(\mathbf{x})$ 在 C 上可微(C^1),其凸性的一个等价定义是,对所有 $\mathbf{x}_0, \mathbf{x}_1 \in C$ 有

$$f_1 \geqslant f_0 + (\mathbf{x}_1 - \mathbf{x}_0)^\top \nabla f_0 \quad (4.1.7)$$

这一定义的几何意义是, $f(\mathbf{x})$ 的图形必位于 $f(\mathbf{x})$ 在点 \mathbf{x}_0 处切线(面)的上方或上面,如图 4.1.4(b)所示. 因而称这样的切线(平面)是凸函数的支撑超平面(supporting hyperplane).

这里简要证明式(4.1.5)与式(4.1.7)对可微函数是等价的. 由式(4.1.5)可得

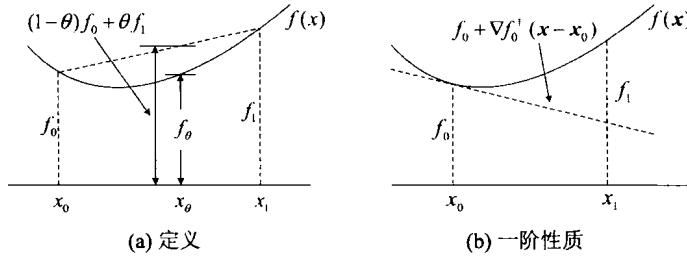


图 4.1.4 凸函数的几何直观

$$\frac{f_\theta - f_0}{\theta} \leq f_1 - f_0$$

把 x_θ 看作直线 $x_\theta = x_0 + \theta(x_1 - x_0)$ 上的一个点，并令 $\theta \downarrow 0$ ，即得式(4.1.7)。反之，若式(4.1.7)成立，则有 $f_1 \geq f_\theta + (x_1 - x_\theta)^\top \nabla f_\theta$ 和 $f_0 \geq f_\theta + (x_0 - x_\theta)^\top \nabla f_\theta$ 。于是得

$$(1 - \theta)f_0 + \theta f_1 \geq f_\theta + [(1 - \theta)(x_0 - x_\theta) + \theta(x_1 - x_\theta)]^\top \nabla f_\theta = f_\theta$$

此即式(4.1.5)。

从式(4.1.7)可以得到的重要结论是：可微凸函数的任意稳定点是全局极小点。进一步可推出结论：

$$(x_1 - x_0)^\top \nabla f_1 \geq f_1 - f_0 \geq (x_1 - x_0)^\top \nabla f_0 \quad (4.1.8)$$

该不等式说明凸函数沿任何直线的斜率都是单调非减的。最后，对二次可微凸函数及开凸集 C ，凸函数的另一个等价定义是

$$\nabla^2 f(x_0) \text{ 半正定, } \forall x_0 \in C \quad (4.1.9)$$

即凸函数具有非负曲率。事实上，令 $p \neq 0, x_1 = x_0 + \alpha p$ ，由 ∇f_1 的 Taylor 展式有

$$\nabla f(x_1) = \nabla f(x_0) + \alpha \nabla^2 f(x_0) p + o(\alpha)$$

代入式(4.1.8)后两边同时除以 α^2 ，再令 $\alpha \rightarrow 0$ ，得 $p^\top \nabla^2 f(x_0) p \geq 0$ ，此即式(4.1.9)。相反，将 f_1 在 x_0 处按 Taylor 级数展开，并结合式(4.1.9)得

$$f_1 = f_0 + (x_1 - x_0)^\top \nabla f_0 + \frac{1}{2} \alpha^2 p^\top \nabla^2 f(x_\theta) p \geq f_0 + (x_1 - x_0)^\top \nabla f_0$$

此即式(4.1.7)。

其他与凸函数有关的定义有：当对所有不同的 x_0, x_1 与 $\theta \in (0, 1)$ ，不等式(4.1.5)严格成立时，称 f 是严格(strict)凸函数；对于 C^1 函数，则不等式(4.1.7)对所有 $x_0 \neq x_1$ 严格成立是严格凸函数的等价定义；而对 C^2 函数，将式(4.1.9)替换成 $\forall x_0 \in C, \nabla^2 f(x_0)$ 正定，虽可推出 $f(x)$ 为严格凸函数，但其逆命题一般并不成立(反例： x^4 为严格凸函数，但在 $x=0$ 时的 Hessian 阵是 0)。若 $-f(x)$ 为凸函数，则称 $f(x)$ 为凹(concave)函数，因而凹函数的概念与非增的斜率及非正的曲率相联系。类似地，若 $-f(x)$ 是严格凸的，则 $f(x)$ 是严格凹的。

凸函数的例子包括：既是凸函数又是凹函数的线性函数；二次函数在其 Hessian 阵半正定时为凸函数，正定时为严格凸函数(注意，二次函数严格凸当且仅当其 Hessian 阵正定)；另一类凸函数是 $\|x\|$ (任何范数)。然而 $\|c(x)\|$ 除 $c(x)$ 为线性函数外一般不是凸函数，其中 $c(x)$ 是映 \mathbb{R}^n 到 \mathbb{R}^m 的多元向量值函数。下列引理给出了一个保持凸性的变换，从凸函数定义出发即可证明(留给读者)。

引理 4.1.1 若 $f_i(\mathbf{x}) (i=1, 2, \dots, m)$ 是凸集 C 上的凸函数, 且 $\alpha_i \geq 0$, 则 $\sum_{i=1}^m \alpha_i f_i(\mathbf{x})$ 仍为 C 上的凸函数.

凸性的一个迷人之处在于, 它提供了可以排除局部但非全局最优解存在的充分性假设, 具体表述如定理 4.1.2. 其推论给出的严格凸假设可以证明全局最优解是唯一的.

定理 4.1.2 设函数 f 是凸集 C 上的凸函数, 则 f 在 C 上的每个局部极小点都是全局极小点, 且全局极小点形成的集合 S 是凸集.

证明 设 \mathbf{x}^* 为局部但非全局的极小点, 则存在 $\mathbf{x}_1 \in C$ 使得 $f_1 < f^*$. 对于 $\theta \in (0, 1)$, 由 C 的凸性, $\mathbf{x}_\theta = (1-\theta)\mathbf{x}^* + \theta\mathbf{x}_1 \in C$. 由 f 的凸性, $f_\theta \leq (1-\theta)f^* + \theta f_1 = f^* + \theta(f_1 - f^*) < f^*$. 令 θ 充分小得出与 \mathbf{x}^* 的局部最优性矛盾, 因此局部极小点必为全局极小点. 又设 $\mathbf{x}_0, \mathbf{x}_1 \in S$, 由式(4.1.6)定义 \mathbf{x}_θ . 由 $\mathbf{x}_0, \mathbf{x}_1$ 的全局最优性有 $f_\theta \geq f_0 = f_1$, 但由凸性有 $f_\theta \leq (1-\theta)f_0 + \theta f_1 = f_1$. 因此 $\mathbf{x}_\theta \in S$, 即 S 为凸集. ■

推论 如果 $f(\mathbf{x})$ 在 C 上是严格凸的, 则全局极小点必唯一.

证明 设 $\mathbf{x}_0 \neq \mathbf{x}_1 \in S, \theta \in (0, 1)$, 同上, 利用严格凸性可得 $f_\theta < f_1$. 这与最优性 $f_\theta \geq f_1$ 矛盾. ■

4.2 算法概述

本节给出无约束优化迭代法的基本格式, 尤其是基于这些基本格式的算法的理想特征.

4.2.1 概述

对于一个实用算法, 应该具备的典型特征是迭代点列 $\mathbf{x}^{(k)}$ 能稳定地逼近局部极小点 \mathbf{x}^* 的邻域, 然后迅速收敛到点 \mathbf{x}^* ; 而且, 每次(或者固定的几次)迭代后 $f^{(k)} = f(\mathbf{x}^{(k)})$ 持续减小, 当人们预先设定的某种收敛准则满足时, 终止迭代. 算法的理论分析则比较复杂, 而且远远匹配不上算法的实际表现. 在进行了一系列的假设后, 往往还只能证明算法的某些收敛性质. 例如, 有时只能证明迭代序列 $\{\mathbf{x}^{(k)}\}$ 有一个聚点(收敛子列的极限点)是稳定点. 该结论虽不如整个序列收敛的结论强, 但在实际中也是完全可以接受的. 此外, 收敛性的证明中如果不要求初始点 $\mathbf{x}^{(0)}$ 充分接近 \mathbf{x}^* , 则称是大范围收敛的(global convergence).

当迭代点列进入局部极小点 \mathbf{x}^* 的某个邻域后, 为了分析和比较随后不同的收敛行为, 引入局部收敛(local convergence)的概念. 定义误差

$$e_k = \|\mathbf{x}^{(k)} - \mathbf{x}^*\| \quad \text{或者} \quad e_k = f^{(k)} - f^*$$

当 $e_k \rightarrow 0$ (收敛)时, 考察 $e_k \rightarrow 0$ 的速度, 即在 \mathbf{x}^* 的邻域内迭代收敛有多快的问题. 若误差满足 $e_{k+1}/e_k^p \rightarrow a$, 其中 $p > 0$, 则称序列是 p 阶收敛的, p 是收敛阶(order of convergence). 最重要的情况是 $p=1$ 和 $p=2$. 当 $p=1$ 时, 称收敛是线性(一阶)的(linear(first order)); 当 $p=2$ 时, 称收敛是二次(二阶)的(second quadratic(order)). 当上面的极限不存在时, 也可以用比值的界给出这些定义. 这时, 一阶收敛对应于 $e_{k+1}/e_k \leq a$, 或者 $\mathbf{h}^{(k+1)} = O(\|\mathbf{h}^{(k)}\|)$; 二阶收敛对应于 $e_{k+1}/e_k^2 \leq a$, 或者 $\mathbf{h}^{(k+1)} = O(\|\mathbf{h}^{(k)}\|^2)$, 其中 $\mathbf{h}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$.

实践中, 当线性收敛的速率常数(rate constant) $a=1$ 时, 称为次线性(sublinear)收敛, 这

是一种最差情况. 当 a 较小(比如 $a \leq 1/4$)时, 这种线性收敛才是令人满意的. 然而, 许多方法会比线性收敛更好, 它们的速率常数是零, 即超线性(superlinear)收敛, 其定义为

$$e_{k+1}/e_k \rightarrow 0 \quad \text{或者} \quad \mathbf{h}^{(k+1)} = o(\|\mathbf{h}^{(k)}\|)$$

值得注意的是, 收敛性和收敛阶的结论并不能保证算法在实际中一定有好的计算效果. 出现这种现象的原因是多方面的. 因为这些结论本身并不能保证算法具有好的特性, 且完全忽略了计算过程中十分重要的舍入误差的影响. 此外, 这些结论通常需要对函数 $f(\mathbf{x})$ 施加一些不易验证的假设, 而实际应用中的问题不一定满足这些条件. 因此, 开发优化方法还依赖于数值实验, 即将算法编制成程序来求解各种有代表性的测试函数(选取的这些测试函数代表了实际问题通常所具有的不同特征). 当然, 实验并不能代替严格的数学证明, 但经验表明: 适当的数值结果经常预示着可能的理论结果.

收敛性及收敛速度的证明通常根据特定算法进行非常具体的分析, 它本身并不能告诉我们如何更好地设计算法. 下面将叙述并讨论当前多数实用算法所具有的一些重要特征.

算法的另一个重要特征是停止迭代所需的准则, 最理想的应该是要求 $f^{(k)} - f^* \leq \epsilon$ 或 $|x_i^{(k)} - x_i^*| \leq \epsilon_i$, 其中 ϵ 和 ϵ_i 是预先给定的参数. 但是这些准则需要提前知道解 \mathbf{x}^* , 因而不实用. 一个不需要这种信息的准则是

$$\|\mathbf{g}^{(k)}\| \leq \epsilon \quad (4.2.1)$$

注意, 这种准则对类似于罚函数(见第9章)那样的坏条件数问题效果会很差. 对于一些预期收敛较快的算法, 准则

$$|x_i^{(k)} - x_i^{(k+1)}| \leq \epsilon_i, \forall i \quad (4.2.2)$$

或者

$$f^{(k)} - f^{(k+1)} \leq \epsilon \quad (4.2.3)$$

通常比较理想, 也很方便. 由于式(4.2.2)需要向量 ϵ , 也可替换为 $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}\| \leq \epsilon$ 的收敛测试准则, 此时仅需要一个标量 ϵ . 还有一个基于 f 的预测下降量的收敛测试准则是

$$\frac{1}{2} \mathbf{g}^{(k)^\top} \mathbf{H}^{(k)} \mathbf{g}^{(k)} \leq \epsilon \quad (4.2.4)$$

其中 $\mathbf{H}^{(k)} = \mathbf{G}^{(k-1)}$, 或者是 $\mathbf{G}^{(k-1)}$ 的近似(参见第5章的牛顿法和拟牛顿法). 已有数值结果表明, 对于拟牛顿法(式(4.2.3)或者式(4.2.4))效果很好. 对收敛较慢的算法, 如共轭梯度法, 准则(4.2.1)比较合适. 如果还知道目标函数的二阶导数, 就应考虑不能让算法终止于鞍点. 还有一个通常很有用的测试方法是: 当迭代次数达到预先指定的最大迭代次数后终止算法. 最后, 当靠近解时, 还应考虑舍入误差的影响, 如果确认舍入误差已经阻碍方法取得进展, 则应终止迭代. 上述的诸多终止测试中, 很难说哪一种是最好的, 经常会一起使用多种准则.

本书考虑的是光滑优化问题, 算法设计需要 $\mathbf{x}^{(k)}$ 处的梯度向量 $\mathbf{g}^{(k)}$, 实践中也证明这些一阶导数方法要比不用导数的方法可靠得多. 在不用导数的方法中, 通常用**向前差商**

$$g_i(\mathbf{x}) = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x})}{h} \quad (4.2.5)$$

或者**中心差商**

$$g_i(\mathbf{x}) = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h} \quad (4.2.6)$$

来近似导数, 其中 \mathbf{e}_i 是第 i 个单位向量, h 是步长. 另一方面, 即便对于导数可以计算的问题, 也可以在算法过程中利用式(4.2.5)和式(4.2.6)来系统地计算导数的估计值, 以验证计算导

数的程序,避免出错.类似地,也可以考虑二阶导数的近似.读者可自行推导如何用一阶导数的有限差商来近似二阶导数.

光滑非线性函数的极小化问题中,最简单的是二次函数极小化问题(除非另有说明,本书提到二次函数时均指它的 **Hessian 阵是正定的**),可以通过解线性方程组(即一阶最优性条件)得到二次函数的极小点.这样,设计极小化一般的非线性函数的迭代算法时,可以借助**二次模型**(quadratic model)完成一次迭代,过程有点类似微积分中“以直代曲”的思想.更重要的原因是:(i)对于一般函数而言,在局部极小点 x^* 附近可以用一个二次函数很好地近似;(ii)即使远离极小点,应用二次信息在很大程度上要比简单地放弃这些信息好.

本书描述的无约束优化方法几乎都基于二次模型.如果可以得到函数的一阶和二阶导数,一个显然的二次模型是二阶 Taylor 展式,基于它可以得到 5.1.2 小节中的经典牛顿法.即使二阶导数不可得,也可以利用各种方法来估计二阶导数,然后在算法中利用近似的二次模型.这类方法的典型代表是 5.3 节介绍的拟牛顿法.还有一种是 5.2 节介绍的共轭梯度法,它不是很直接地利用二次模型.最后,5.4 节描述的方法利用 $f(x)$ 的平方和结构很容易得到 Hessian 阵的一种近似.对于一阶导数不可得的极小化方法,已证明用差商估计一阶导数,再结合拟牛顿法是目前最有效的方法之一.

利用二次函数作目标函数的局部近似模型的好处之一是,在某些假定下,牛顿法是二阶收敛的,拟牛顿法是超线性收敛的.另一个好处是,算法通常具有**二次终止性**(quadratic termination),即算法在有限次迭代后能够确定二次函数的极小点.5.2 节将要介绍的**共轭方向法**(conjugate direction method)执行精确线搜索时就具有二次终止性.

4.2.2 线搜索法

线搜索法和**信赖域法**是两种基本的优化算法格式.线搜索法与最优化这门学科是一起出现和发展的,其基本思想是在当前迭代点处,根据收集到的关于优化问题的信息,确定一个有待进一步探测的方向;然后在这个方向上进行一维搜索得到合适的步长,进而得到下一个迭代点.各种不同的具体方法见第 5 章.信赖域法,就其思想萌芽而言,可以追溯到 20 世纪 40 年代人们关于最小二乘问题的研究,详见 5.4.2 小节.但是作为一种求解优化问题的系统方法,它出现于 20 世纪 40 年代,较快地发展于 20 世纪 80 年代.迄今,信赖域法仍然是非线性规划中一个比较活跃的研究领域.鉴于线搜索法中找下降方向和一维搜索的不匹配(一个是在 n 维空间找,另一个是在一维空间找),研究者在信赖域法中将它们合二为一,即直接构造一个带信赖域约束的模型函数,称为**信赖域子问题**.解之得到一个可能的改进步,最后根据一定的标准决定是否接受这个可能的改进步,详见第 6 章.本章的重点是线搜索法.

给定初始点 $x^{(0)}$.设 $x^{(k)}$ 处的梯度 $\mathbf{g}^{(k)} \neq \mathbf{0}$,则线搜索法(line search method)第 k 次迭代的基本格式为:

- (a) 确定 $x^{(k)}$ 处的**搜索方向**(search direction) $\mathbf{p}^{(k)}$.
- (b) (一维搜索, line search)求解关于 α 的极小化问题

$$\min \phi(\alpha) := f(x^{(k)} + \alpha \mathbf{p}^{(k)}) \quad (4.2.7)$$

得到 α_k .

- (c) 置 $x^{(k+1)} = x^{(k)} + \alpha_k \mathbf{p}^{(k)}$.

这里要求步骤(a)中确定的搜索方向满足**下降性**(descent property),即

$$\phi'(0) = \mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} < 0 \quad (4.2.8)$$

由 Taylor 定理, 对沿着 $\mathbf{p}^{(k)}$ 充分小的 α , 目标值必可减小, 故称满足该条件的方向 $\mathbf{p}^{(k)}$ 是下降方向. 在 4.3 节将看到, 在某些适当的条件下, 利用下降性即可证明方法的收敛性. 各种线搜索法的主要区别在于步骤(a)中选取 $\mathbf{p}^{(k)}$ 的方式, 本质上对应不同的二次模型的最优解

$$\mathbf{p}^{(k)} = \arg \min_{\mathbf{p} \in \mathbb{R}^n} \mathbf{g}^{(k)^\top} \mathbf{p} + \frac{1}{2} \mathbf{p}^\top \mathbf{B}^{(k)} \mathbf{p} \quad (4.2.9)$$

且只要 $\mathbf{B}^{(k)}$ 正定, $\mathbf{p}^{(k)}$ 就一定是下降方向. 在第 5 章将看到, 正是不同的 $\mathbf{B}^{(k)}$ 导致了不同的搜索方向 $\mathbf{p}^{(k)}$.

步骤(b)是线搜索子问题, 即求解(或者非精确地求解) n 元函数在一个一维子空间上的极小点. 由于每次迭代都要进行一维搜索, 所以一维搜索方法的好坏直接影响到非线性规划计算方法的效率.

精确(exact)线搜索是指精确求解问题(4.2.7), 即计算 $\alpha_k > 0$ 使得

$$\phi(\alpha_k) = \min_{\alpha > 0} f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)})$$

这种思想在概念上很重要, 且经常出现在一些理论的证明中, 尤其是精确线搜索要求 $d\phi/d\alpha$ 在 α_k 处必须是零这一重要性质. 该性质的几何直观如图 4.2.1 所示, 由式(1.4.5)可以将该性质显式表示为

$$\phi'(\alpha_k) = \nabla f(\mathbf{x}^{(k-1)})^\top \mathbf{p}^{(k)} = 0 \quad (4.2.10)$$

该条件对一些无约束优化方法的有限终止性起着关键作用; 而且由它可以得到二次函数(1.4.7)在由 $\mathbf{p}^{(k)}$ 确定的直线上的极小点

$$\alpha_k = \frac{-\mathbf{g}^{(k)^\top} \mathbf{p}^{(k)}}{\mathbf{p}^{(k)^\top} \mathbf{G} \mathbf{p}^{(k)}} \quad (4.2.11)$$

显然, 精确线搜索是一个具有特殊形式的一维优化问题, 它的本质是必须求解非线性方程 $d\phi/d\alpha = 0$. 其中, 计算 $\phi(\alpha)$ 的函数值和导数值实际上需要计算 n 元函数的函数值和它的方向导数. 当 n 很大时, 精确求解问题(4.2.7)的计算量是相当大的. 由于一维搜索只是多维优化方法每次迭代的一个子问题, 精确求解有可能降低整个方法的效率. 特别是当前迭代 $\mathbf{x}^{(k)}$ 远离原问题之解时, 精确求解子问题很可能得不偿失.

非精确(inexact)线搜索指不精确求解问题(4.2.7), 而仅求一个满足某些条件的 α_k . 详细讨论见 4.3 节.

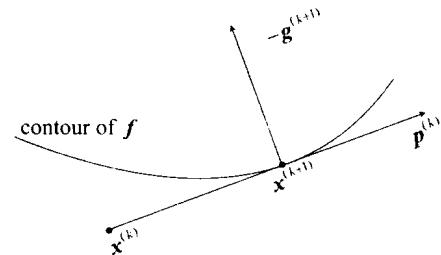


图 4.2.1 精确线搜索

4.3 非精确线搜索

引入线搜索增强了优化软件的可靠性. 早些时候, 人们通常选择精确线搜索, 即求解必要条件所确定的方程(4.2.10). 然而除去一些简单的情形, 一般执行精确线搜索的代价很大. 这促使一些研究者放宽了一维搜索的精度要求, 并仅利用下降性质来强迫每次迭代中目标函数

下降, 即 $f^{(k+1)} < f^{(k)}$. 该做法通常很有效, 但仅要求 f 下降并不能保证大范围收敛性.

例 4.3.1 目标函数 $f(x) = x^2$, 初始点 $x^{(0)} = 2$. 考虑以下情况:

(a) $p^{(k)} = (-1)^{k+1}, \alpha_k = 2 + \frac{3}{2^{k+1}}$, 这时迭代产生的点列如图 4.3.1(a) 所示. 方法产生的序列有两个聚点 1 和 -1.

(b) $p^{(k)} = -1, \alpha_k = \frac{1}{2^{k+1}}$, 迭代产生的序列如图 4.3.1(b) 所示. 方法产生的点列收敛于 1.

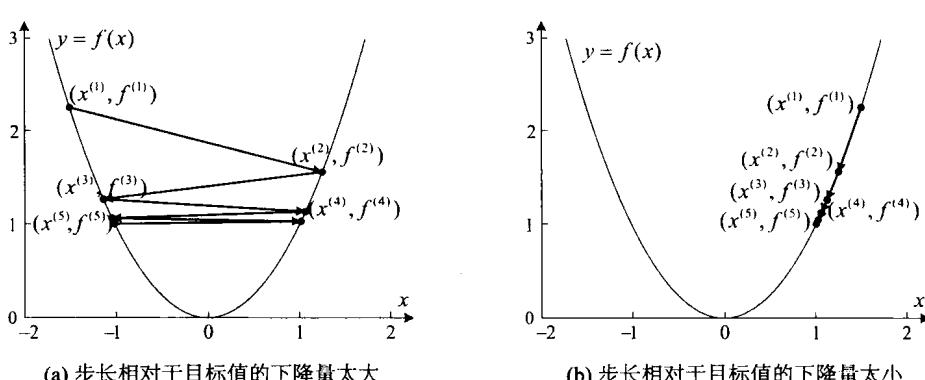


图 4.3.1 简单线搜索的失败

上例中方法产生的点列均未收敛到理想点 0. 因此人们对这种更有效方法的稳定性产生了怀疑. 这个事实和大量互异的线搜索代码的出现引发了人们对终止一维搜索时可接受条件的研究. 这些条件既允许低精度的一维搜索, 又强迫方法是大范围收敛的.

4.3.1 一维搜索的终止准则

用 $\bar{\alpha}$ 表示使得 $f(x^{(k)} + \alpha p^{(k)}) = f(x^{(k)})$ 成立的最小正数, 如图 4.3.2 所示. 例 4.3.1 说明. 并不是区间 $(0, \bar{\alpha})$ 中的每一个点都是理想的. 究其原因, 当 α_k 靠近 0 或者靠近 $\bar{\alpha}$ (比如例 4.3.1(a) 中靠近 $\bar{\alpha}$, (b) 中靠近 0) 时, 可能会出现 f 的下降量相对于精确线搜索时所获得的下降量可以忽略不计的情况. 这些讨论说明一维搜索的目的是确定使目标函数值 f 显著下降, 但同时不宜太大或者太小的步长. 为此, 确定的 α_k 从直观上讲不能太靠近区间 $(0, \bar{\alpha})$ 的端点; 而且相关的条件必须使得可接受点 (acceptable point) (满足条件的 α) 存在, 且可以在有限步内找到. 此外, 条件还不能排除具有正曲率二次函数 $q(x^{(k)} + \alpha p^{(k)})$ 的极小点, 该事实在证明有些方法的超线性收敛时很重要.

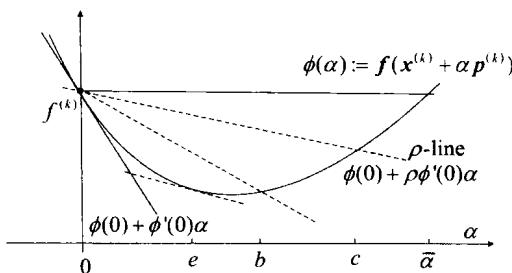


图 4.3.2 线搜索的稳定性条件

一个实用和流行的终止一维搜索的准则是 **Armijo 法则**. 该法则的本质思想是首先保证所选的 α_k 不太大, 其次也不应该太小. 考虑式(4.2.7)定义的函数 $\phi(\alpha)$, 并定义与之对应的线性函数

$$l(\alpha) = \phi(0) + \rho\phi'(0)\alpha$$

称这个线性函数为 ρ -线, 见图 4.3.2. 在 Armijo 法则中, 若 α 对应的函数值位于 ρ -线之

下,即

$$\phi(\alpha) \leq \phi(0) + \rho \phi'(0)\alpha \quad (4.3.1)$$

则认为 α 不太大,其中 $\rho \in (0, 1)$ 是参数,典型取值是 $\rho = 10^{-2}$,也称式(4.3.1)为 Armijo 条件.如果 α_k 满足式(4.3.1),则相应的 $f(\mathbf{x})$ 的下降量满足

$$f^{(k)} - f^{(k+1)} \geq -\rho \mathbf{g}^{(k)^T} \mathbf{s}^{(k)} \quad (4.3.2)$$

其中 $\mathbf{s}^{(k)} = \alpha_k \mathbf{p}^{(k)}$.为了确保 α 不太小,选取 $\gamma < 1$,且当

$$\phi(\alpha/\gamma) > \phi(0) + \rho \phi'(0)\alpha/\gamma$$

成立时,认为 α 不太小.这意味着将 α 扩大 $1/\gamma$ 倍后将不再满足条件(4.3.1).

在实践中,可以利用 Armijo 法则来设计一个简单的一维搜索技术,即伪码表示的算法 4.3.1.这里算法从一个初始的 $\bar{\alpha}$ 开始,如果它不满足式(4.3.1),则将 α 缩小 γ 倍,重复这个过程,直到满足条件(4.3.1).因为当 i 很大时, $\gamma^i \bar{\alpha}$ 会变得充分小,这样在有限步测试之后,能够找到一个满足条件(4.3.1)的步长(见图 4.3.2).最终接受的步长是形如

$$\bar{\alpha}, \gamma \bar{\alpha}, \gamma^2 \bar{\alpha}, \dots$$

的序列中第一个满足式(4.3.1)的值,所以这里的回溯策略避免了步长变得太小这种情况.实践中,允许在线搜索的每次迭代中改变缩减因子 γ .此外,初始步长 α 在牛顿法和拟牛顿法中选为 1,但在其他诸如最速下降法或共轭梯度法中可能取不同的值.

Algorithm 4.3.1 Backtracking-Armijo line search

```

1: Choose  $\alpha > 0, \gamma, \rho \in (0, 1)$ ; set  $\alpha = \bar{\alpha}$ ;
2: while  $\alpha$  doesn't satisfy inequality(4.3.1) do
3:   set  $\alpha = \gamma \alpha$ ;
4: end while
5: return  $\alpha$  as  $\alpha_k$ 

```

另一种常用的一维搜索终止准则是 Goldstein 测验.像 Armijo 法则一样,如果 Armijo 条件(4.3.1)对给定的 $\rho \in (0, 1/2)$ 成立,则认为 α 不太大.如果 α 满足

$$\phi(\alpha) \geq \phi(0) + (1 - \rho) \phi'(0)\alpha \quad (4.3.3)$$

则认为 α 不太小.换句话说, $\phi(\alpha)$ 必须位于图 4.3.2 中那两条过点 $(0, \phi(0))$ 的虚线之间.图 4.3.2 显示了 $\rho = 1/4$ 时由 Goldstein 测验所确定的可接受区间是 $[b, c]$.这里要求 $\rho < 1/2$ 是为了保证二次函数的极小点也满足条件.然而,当 $\phi(\alpha)$ 不是二次函数时,第二个条件(4.3.3)可能会排除 $\phi(\alpha)$ 的极小点,如图 4.3.2 所示.

基于此原因,当目标函数的导数可得时,可用关于斜率的测验条件(Wolfe 于 1968 年提出)

$$\phi'(\alpha) \geq \sigma \phi'(0), \quad \sigma \in (\rho, 1) \quad (4.3.4)$$

代替条件(4.3.3),这里 σ 是一个参数.该不等式蕴含着 $\mathbf{x}^{(k+1)}$ 满足

$$\mathbf{g}^{(k+1)^T} \mathbf{s}^{(k)} \geq \sigma \mathbf{g}^{(k)^T} \mathbf{s}^{(k)} \quad (4.3.5)$$

因为 $\sigma < 1$,利用这个条件可保证 α 不太小.同时同前面一样,利用条件(4.3.1)来保证 α 不太大.称式(4.3.1)和式(4.3.4)为 Wolfe 条件,图 4.3.2 中($\sigma = 1/2, \rho = 1/4$ 时)由该条件确定的可接受区间是 $[e, c]$,该区间包含真正的极小点.这里的限制条件 $\sigma \geq \rho$ 确保可接受点存在,且可在有限步内找到,详见定理 4.3.3 和定理 4.4.1.

满足式(4.3.1)和式(4.3.4)的点有可能离精确解较远. Powell 建议了一个更紧的关于斜率的双边条件

$$|\phi'(\alpha)| \leq -\sigma\phi'(0) \quad (4.3.6)$$

来代替条件(4.3.4). 通过减小 σ , 利用条件(4.3.6)有可能强迫可接受点跑到 $\phi(\alpha)$ 的局部极小点的一个充分小的领域内. 进一步, 若 $\sigma=0$, 则退化到精确线搜索的情形. 称式(4.3.1)和式(4.3.6)是强 Wolfe 条件, 该条件也确保可接受点是存在的.

在一种宽泛的意义下, Wolfe 条件是尺度不变的: 即给目标函数乘以某个常数, 或者对变量作仿射变换时, 它们保持不变. 可以在许多线搜索法中利用它们, 尤其是拟牛顿法.

实践中, $\sigma=0.9$ 相当于很弱(限制性不强)的一维搜索, 而 $\sigma=0.1$ 则对应相当精确的一维搜索. σ 值越小, 满足条件的步长区间越短, 一维搜索所需要的时间越长. 因此, 除了检验精确线搜索的理论假设外, 通常很少使用小的 σ 值. 当搜索方向 $\mathbf{p}^{(k)}$ 由牛顿法或拟牛顿法选定时, σ 的典型值是 0.9; 当 $\mathbf{p}^{(k)}$ 由共轭梯度法获得时, σ 的典型值是 0.1. 至于 ρ , 典型值是 10^{-2} . 最后, 需要指出的是, Armijo 条件也有可能让可接受区间位于真正极小点的左侧. 但实践中这种情况几乎不会出现, 而它的优点是排除了 $\alpha=\infty$ 这种麻烦的情况.

4.3.2 下降方法的稳定性

首先给出如下的满足 Armijo 条件的步长的存在性, 据此可以得到 Armijo 法则确定的步长的下界.

定理 4.3.1 (步长的存在性) 假设 $f \in C^1$, 且 $\mathbf{g}(\mathbf{x})$ 是 Lipschitz 连续的, Lipschitz 常数为 L . 设 $\mathbf{p}^{(k)}$ 是 $\mathbf{x}^{(k)}$ 处的下降方向. 如果 $0 < \rho < 1$, 则区间 $[0, \hat{\alpha})$ 内的所有值均满足 Armijo 条件(4.3.1), 其中 $\hat{\alpha} = \frac{(\rho-1)\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)}}{L \|\mathbf{p}^{(k)}\|^2}$. 进一步, 基于 Armijo 法则确定的步长大于等于 $\gamma\hat{\alpha}$.

证明 依次由中值定理和 $\mathbf{g}(\mathbf{x})$ 是 Lipschitz 连续的有

$$|f(\mathbf{x}^{(k)} + \alpha\mathbf{p}^{(k)}) - f^{(k)} - \alpha\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)}| \leq L \|\alpha\mathbf{p}^{(k)}\|^2$$

从而对 $\alpha \in (0, \hat{\alpha})$, 有

$$\begin{aligned} f(\mathbf{x}^{(k)} + \alpha\mathbf{p}^{(k)}) &\leq f^{(k)} + \alpha\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} + L\alpha^2 \|\mathbf{p}^{(k)}\|^2 \\ &\leq f^{(k)} + \alpha\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} + L\alpha \|\mathbf{p}^{(k)}\|^2 \hat{\alpha} \\ &= f^{(k)} + \alpha\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} + \alpha(\rho-1)\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \\ &= f^{(k)} + \rho\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \alpha \end{aligned}$$

这里的第一个等式利用了 $\hat{\alpha}$ 的定义. ■

因为一般不知道 L , 所以并不能利用该定理来计算 $\hat{\alpha}$, 它仅保证这个值的存在性. $\hat{\alpha}$ 的分子对应斜率, 分母对应曲率. 具体可将其解释如下: 如果曲率项大一些, 则可接受 α 的范围小一些. 类似地, 如果沿着搜索方向的梯度投影大一些, 则可接受 α 的范围大一些. 此外, 因为有上界 $\hat{\alpha}$, 所以步长肯定不会太大, 且当 $\langle \mathbf{p}^{(k)}, \mathbf{g}^{(k)} \rangle / \|\mathbf{p}^{(k)}\|^2$ 很小时, 步长只能很小. 该事实对成功终止一维搜索算法很重要.

定理 4.3.2 (大范围收敛) 考虑基于 Armijo 法则确定步长的线搜索法, 若 f 下方有界, 且 $\mathbf{g}(\mathbf{x})$ 在水平集 $\{\mathbf{x}: f(\mathbf{x}) < f(\mathbf{x}^{(0)})\}$ 上 Lipschitz 连续, 则

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)}\|}{\|\mathbf{p}^{(k)}\|} = 0 \quad (4.3.7)$$

证明 假设对所有 k 有 $\mathbf{g}^{(k)} \neq \mathbf{0}$. 再由 f 下方有界的假设, 必有 $\lim_{k \rightarrow \infty} f^{(k)} > -\infty$. 由 Armijo 条件 (4.3.1), 对所有 k 有 $f^{(k)} - f^{(k+1)} \geq -\rho \mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \alpha_k$. 对前 i 次的迭代求和, 得 $f^{(0)} - f^{(i+1)} \geq \rho \sum_{k=0}^i (-\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \alpha_k)$. 再次由 f 下方有界的假设, 该不等式左端有上界, 从而右端的正项级数也有上界, 故收敛, 因此有 $\lim_{k \rightarrow \infty} \mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \alpha_k = 0$. 再由定理 4.3.1 有 $\alpha_k \geq \hat{\gamma} \alpha$, 从而

$$\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)} \alpha_k \leq \frac{\gamma(\rho-1)}{L} \left(\frac{\mathbf{p}^{(k)^\top} \mathbf{g}^{(k)}}{\|\mathbf{p}^{(k)}\|} \right)^2 < 0$$

由夹逼定理即得所要结论. ■

定理 4.3.3 (步长的存在性) 假设 $f \in C^1$, $\mathbf{p}^{(k)}$ 是 $\mathbf{x}^{(k)}$ 处的下降方向, 且 f 沿着射线 $\{\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)} : \alpha > 0\}$ 有下界. 如果 $0 < \rho < \sigma < 1$, 则满足 Wolfe 条件和强 Wolfe 条件的可接受区间存在.

证明 因为 $\phi(\alpha) = f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)})$ 对所有 $\alpha > 0$ 有下界, 且 $0 < \rho < 1$, 所以 ρ -线 $l(\alpha)$ 与 $\phi(\alpha)$ 的图形至少相交一次. 设 $\hat{\alpha}$ 是最小的交点, 即 $\hat{\alpha} = \min\{\alpha > 0 : l(\alpha) = \phi(\alpha)\}$. 由 $\hat{\alpha}$ 的选取可知

$$f(\mathbf{x}^{(k)} + \hat{\alpha} \mathbf{p}^{(k)}) = f^{(k)} + \rho \phi'(0) \hat{\alpha} \quad (4.3.8)$$

且区间 $(0, \hat{\alpha})$ 中的任一 α 都满足式 (4.3.1). 再由中值定理, 存在 $\bar{\alpha} \in (0, \hat{\alpha})$ 满足

$$f(\mathbf{x}^{(k)} + \bar{\alpha} \mathbf{p}^{(k)}) - f^{(k)} = \phi'(\bar{\alpha}) \bar{\alpha} \quad (4.3.9)$$

由式 (4.3.8) 和式 (4.3.9) 有

$$\phi'(\bar{\alpha}) = \rho \phi'(0) > \sigma \phi'(0) \quad (4.3.10)$$

最后一个不等式是因为 $\rho < \sigma$ 且 $\phi'(0) < 0$. 因此 $\bar{\alpha}$ 满足 Wolfe 条件, 且其中的不等式是严格成立的. 因此, 由 f 的光滑性假设, 存在包含 $\bar{\alpha}$ 的区间, Wolfe 条件在该区间上成立. 此外, 因为式 (4.3.10) 的左边是负的, 所以强 Wolfe 条件在相同的区间上成立. ■

当搜索方向与最速下降方向趋于正交时, 有可能产生相对于 $f(\mathbf{x})$ 可以忽略不计的下降量. 因此, 在给出另外的大范围收敛结论时, 需要对确定的搜索方向有进一步的限制, 而不仅仅要求其为下降方向. 通常利用 $\mathbf{p}^{(k)}$ 满足 **夹角条件** (angle criterion) 来排除上述情况. 记 $\mathbf{p}^{(k)}$ 和 $-\mathbf{g}^{(k)}$ 之间的夹角为 θ_k . 所谓夹角条件, 是指 $\pi/2 - \theta_k$ 一致下方有界 (θ_k 一致上方有界), 即存在与 k 无关的数 $\mu \in (0, \pi/2)$ 使得

$$\theta_k \leq \frac{\pi}{2} - \mu, \quad \forall k \quad (4.3.11)$$

这里夹角 $\theta_k \in [0, \pi/2]$ 定义为

$$\cos \theta_k = \frac{-\mathbf{g}^{(k)^\top} \mathbf{p}^{(k)}}{\|\mathbf{g}^{(k)}\| \|\mathbf{p}^{(k)}\|} = \frac{-\mathbf{g}^{(k)^\top} \mathbf{s}^{(k)}}{\|\mathbf{g}^{(k)}\| \|\mathbf{s}^{(k)}\|} \quad (4.3.12)$$

有了这个限制, 就可以陈述大范围收敛性定理.

定理 4.3.4 (大范围收敛) 对于步长满足条件 (4.3.1) 和条件 (4.3.5) (或者条件 (4.3.6)) 的线搜索法, 如果 $\mathbf{g}(\mathbf{x})$ 在水平集 $\{\mathbf{x} : f(\mathbf{x}) < f(\mathbf{x}^{(0)})\}$ 上 Lipschitz 连续, 且对任意的 k 有式 (4.3.11) 成立, 则对某一 k 有 $\mathbf{g}^{(k)} = \mathbf{0}$, 或者 $f^{(k)} \rightarrow -\infty$, 或者 $\mathbf{g}^{(k)} \rightarrow \mathbf{0}$.

证明 假定对所有 k 有 $\mathbf{g}^{(k)} \neq \mathbf{0}$ (因此 $\mathbf{s}^{(k)} \neq \mathbf{0}$), 且 $\{f^{(k)}\}$ 有下界, 则可得到 $f^{(k)} - f^{(k+1)} \rightarrow 0$. 进一步由式 (4.3.1) 和式 (4.3.2) 有 $-\mathbf{g}^{(k)^\top} \mathbf{s}^{(k)} \rightarrow 0$. 假定 $\mathbf{g}^{(k)} \rightarrow \mathbf{0}$ 不成立, 则存在 $\epsilon > 0$ 和子序列使

得 $\|\mathbf{g}^{(k)}\| \geq \epsilon$. 这样, 由式(4.3.11)和式(4.3.12)有 $\mathbf{s}^{(k)} \rightarrow \mathbf{0}$. 这里为了书写简便, 假设子序列即为序列本身. 整理式(4.3.4), 有

$$-\mathbf{g}^{(k)\top} \mathbf{s}^{(k)} \leq \frac{(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})^\top \mathbf{s}^{(k)}}{1 - \sigma} \leq \frac{\|\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\| \|\mathbf{s}^{(k)}\|}{1 - \sigma} \quad (4.3.13)$$

Lipschitz 连续性假设蕴含着 $\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} = o(1)$, 因此由式(4.3.12)和式(4.3.13)及 $\|\mathbf{g}^{(k)}\| \geq \epsilon$ 有 $\cos \theta_k \leq o(1)$, 这与夹角条件(4.3.11)矛盾. 因此 $\mathbf{g}^{(k)} \rightarrow \mathbf{0}$. ■

类似地, 当步长满足条件(4.3.1)和条件(4.3.3)时, 线搜索法也是大范围收敛的.

通常由 $f(\mathbf{x})$ 的性质(有界性)可以排除 $f^{(k)} \rightarrow -\infty$ 这种可能性, 因此由上述定理立即得出, 充分大的 k 一定能使终止条件(4.2.1)满足, 这样迭代不可能收敛于非稳定点, 从而算法一定收敛到局部极小点或者鞍点. 由于每次迭代中 f 总在减小, 通常会收敛于局部极小点. 如果把 $\mathbf{g}(\mathbf{x})$ 是 Lipschitz 连续的假设放宽为 $f \in C^1$, 则可证明任一聚点是稳定点, 这样的结论稍弱一些.

最后, 考虑实用算法是否能满足夹角条件(4.3.11). 显然, 最速下降法中 $\mathbf{p}^{(k)} = -\mathbf{g}^{(k)}$, 从而 $\theta_k = 0$; 但不幸的是, 它在实践中的表现并不好, 只有迫不得已时才用该方法. 第 5 章将要讨论的拟牛顿法的搜索方向 $\mathbf{p}^{(k)} = -\mathbf{H}^{(k)} \mathbf{g}^{(k)}$, 其中 $\mathbf{H}^{(k)}$ 是对称矩阵. 这时, 使下降性质成立的一个充分条件是 $\mathbf{H}^{(k)}$ 是正定的(因为 $\mathbf{g}^{(k)} \neq \mathbf{0}$ 时, $\mathbf{g}^{(k)\top} \mathbf{p}^{(k)} = -\mathbf{g}^{(k)\top} \mathbf{H}^{(k)} \mathbf{g}^{(k)} < 0$). 此种情况下, 使夹角条件成立的一个充分条件是矩阵 $\mathbf{H}^{(k)}$ 的条件数 κ_k (等于最大与最小特征值之比 λ_1/λ_n) 一致有上界 κ . 此时, 由不等式 $\|\mathbf{H}\mathbf{g}\| \leq \lambda_1 \|\mathbf{g}\|$, $\mathbf{g}^\top \mathbf{H}\mathbf{g} \geq \lambda_n \mathbf{g}^\top \mathbf{g}$ 和 $\sin t \leq t$, 利用条件(4.3.11), 有

$$\theta_k \leq \frac{\pi}{2} - \frac{1}{\kappa_k} \leq \frac{\pi}{2} - \frac{1}{\kappa}$$

对于多数好算法, 目前还不能证明条件(4.3.11), 比如对拟牛顿法还不能证明 κ_k 有界. 鉴于收敛性证明的需要, 可以调整 $\mathbf{p}^{(k)}$ 来使式(4.3.11)对某预先指定的 μ 成立, 例如给方向 $\mathbf{p}^{(k)}$ 加上 $-\mathbf{g}^{(k)}$ 的适当倍数, 或者修正 $\mathbf{H}^{(k)}$ 使修正后的矩阵序列的条件数有界. 但这样做有可能使原来超线性收敛的算法退化成线性收敛. 因此, 采取这些特定的修正措施必须慎重, 否则得不偿失.

4.4 线搜索子问题的算法

一阶导数是否可用是设计算法时需要考量的基本问题. 当导数不可用时, 基本上没有太多的理论来指导如何终止一维搜索. 因此存在很多可能的做法, 本节最后考虑一些典型的方法. 然而一阶导数可得仍然是最重要的情况. 在这种情况下, 一种被广泛接受的终止准则是 4.3 节讨论的强 Wolfe 准则. 本节描述一个可以满足这些条件的一维搜索算法, 其中仅需要计算有限次(通常很少) $\phi(\alpha)$ 和 $\phi'(\alpha)$, 其中 $\phi(\alpha)$ 的定义见式(4.2.7), 同时也假定下降性质 $\phi'(0) < 0$ 成立.

一维搜索算法是迭代法, 当找到的迭代点满足一个可接受点的某些标准条件时会终止, 它包含两个不同的部分. 首先是划界阶段(bracketing phase), 其目标是找到一个覆盖(bracket), 即包含可接受点的一个非平凡区间 $[a_i, b_i]$. 紧接着的是分割阶段(secting phase), 将上面找到的覆盖进行分割以产生长度趋于零的区间序列 $[a_i, b_i]$. 该事实构成线搜索算法有限终止证明的基础. 此外, 鉴于找到接近 $\phi(\alpha)$ 的局部极小点的可接受点是更可取的, 采用某种形式的插值(interpolation)也是很有益的. 具体即先用关于 α 的二次或者三次多项式拟合已知数据, 选取

多项式(可能受某个覆盖限制)的极小点作为下一次迭代 α_{j+1} .

下面是当导数可用时一个被广泛使用的一维搜索算法, 虽然未必是最好的, 但是它与当前广泛使用的其他一维搜索算法非常相似, 同时也可以证明该算法的一些性质. 假设用户为 $\phi(\alpha)$ 提供的下界为 $\bar{\phi}$ (更精确地说, 即假定用户愿意接受 $\phi(\alpha)$ 的任何满足 $\phi(\alpha) < \bar{\phi}$ 的值). 例如, 在非线性最小二乘问题中 $\bar{\phi} = 0$ 很合适. 基于该假设, 线搜索可以限制在区间 $(0, \mu]$ 上, 其中

$$\mu = \frac{\bar{\phi} - \phi(0)}{\rho \phi'(0)} \quad (4.4.1)$$

是 ρ -线和直线 $\phi = \bar{\phi}$ 的交点.

在划界阶段, 迭代 α_i 以递增的跃度向右移动, 直到检测到 $\phi \leq \bar{\phi}$ 或者找到包含可接受点的区间. 初始化 $\alpha_0 = 1$, 给定 $\alpha_1 \in (0, \mu)$, 该阶段的算法用伪码描述为算法 4.4.1, 算法中的 τ_1 是预设的跃度的增加因子, 典型的取值是 $\tau_1 = 9$. 在第 19 步中, 可以用不同的方式选取 α_{i+1} , 但是一个合理的选择应该是在给定区间内极小化基于 $\phi(\alpha_i), \phi'(\alpha_i), \phi(\alpha_{i-1})$ 和 $\phi'(\alpha_{i-1})$ 的三次插值多项式.

Algorithm 4.4.1 Bracketing phase for Wolfe linear search

```

1: for  $i = 1, 2, \dots$  do
2:   evaluate  $\phi(\alpha_i)$ ;
3:   if  $\phi(\alpha_i) \leq \bar{\phi}$  then
4:     terminate.
5:   end if
6:   if  $\phi(\alpha_i) > \phi(0) + \rho \phi'(0) \alpha_i$  or  $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$  then
7:      $a_i = \alpha_{i-1}; b_i = \alpha_i$ ; terminate B;
8:   end if
9:   evaluate  $\phi'(\alpha_i)$ ;
10:  if  $|\phi'(\alpha_i)| \leq -\sigma \phi'(0)$  then
11:    terminate.
12:  end if
13:  if  $\phi'(\alpha_i) \geq 0$  then
14:     $a_i = \alpha_i; b_i = \alpha_{i-1}$ ; terminate B;
15:  end if
16:  if  $\mu \leq 2\alpha_i - \alpha_{i-1}$  then
17:     $\alpha_{i+1} = \mu$ ;
18:  else
19:    choose  $\alpha_{i+1} \in [2\alpha_i - \alpha_{i-1}, \min(\mu, \alpha_i + \tau_1(\alpha_i - \alpha_{i-1}))]$ ;
20:  end if
21: end for

```

为了描述算法的性质时方便, 引入恰当覆盖的定义.

定义 4.4.1 若覆盖满足如下性质:

- (i) a_i 是当前最好测试点 (ϕ 最小), 且满足条件 (4.3.1),
- (ii) 已经得到 $\phi'(a_i)$, 且满足 $(b_i - a_i)\phi'(a_i) < 0$, 但不满足条件 (4.3.6),
- (iii) b_i 满足 $\phi(b_i) > \phi(0) + \rho \phi'(0) b_i$ 或者 $\phi(b_i) \geq \phi(a_i)$, 或者二者都满足, 则称为恰当覆盖 (right bracket).

在算法 4.4.1 中, “terminate”表示一维搜索终止于一个可接受点 a_i , 或者满足 $\phi(a_i) \leq \bar{\phi}$ 的点. 当“terminate B”发生时, 已找到一个恰当覆盖(其记号既允许 $a_i < b_i$, 也允许 $b_i < a_i$). 对恰当覆盖, 如下结论成立.

引理 4.4.1 若 $\sigma \geq \rho$, 恰当覆盖包含满足强 Wolfe 条件的可接受区间.

证明 若 $b_i > a_i$, 考虑过点 $(a_i, f(a_i))$ 且斜率为 $\rho\phi'(0)$ 的直线 L (平行于 ρ -线). 令 \hat{a} 是直线 L 与 $\phi(a)$ 的图形在区间 (a_i, b_i) 内最靠近 a_i 的交点. 定义 4.4.1 的(ii)、(iii) 和连续性保证了 \hat{a} 的存在性. 仿照定理 4.3.3, 利用中值定理可证明结论. 若 $b_i < a_i$, 则考虑过点 $(a_i, f(a_i))$ 且斜率为 0 的直线 L , 类似可证. ■

该引理说明划界阶段已经找到包含一个可接受点的覆盖. 接下来的分割阶段将产生长度趋于零的覆盖序列 $[a_j, b_j], j = i, i+1, \dots$. 每次迭代挑选覆盖 $[a_j, b_j]$ 中的一个新的试探点 a_j ; 下一个覆盖是 $[a_j, a_j]$, 或者 $[a_j, a_j]$, 或者 $[a_j, b_j]$, 最终的选取结果要保证新的覆盖仍然是恰当覆盖. 在当前试探点 a_j 满足强 Wolfe 条件时终止分割阶段. 算法 4.4.2 是这个阶段的伪码.

在算法 4.4.2 中, τ_2 和 τ_3 是预设因子 ($0 < \tau_2 < \tau_3 \leq 1/2$), 用于避免 a_j 太靠近区间 $[a_j, b_j]$ 的端点. 可以得到

$$|b_{j+1} - a_{j+1}| \leq (1 - \tau_2) |b_j - a_j| \quad (4.4.2)$$

该事实可以保证区间长度趋于零. 典型值是 $\tau_2 = 1/10$ (建议 $\tau_2 \leq \sigma$) 和 $\tau_3 = 1/2$ (实践表明算法对这些具体数值不太敏感). 在第 2 步中可以用任何方式选取 a_j , 但是合理的选择应该极小化给定区间的基于 $\phi(a_j), \phi'(a_j), \phi(b_j)$ 和 $\phi'(b_j)$ 的二次或者三次插值多项式. 当算法 4.4.2 终止时, a_j 是待求的可接受点, 即所需要的步长 a_k . 下面的结论给出了分割阶段的收敛性质.

Algorithm 4.4.2 Sectioning phase for Wolfe linear search

```

1: for  $j = i, i+1, \dots$  do
2:   choose  $a_j \in [a_j + \tau_2(b_j - a_j), b_j - \tau_3(b_j - a_j)]$ ;
3:   evaluate  $\phi(a_j)$ ;
4:   if  $\phi(a_j) > \phi(0) + \rho\phi'(0)a_j$ , or  $\phi(a_j) \geq \phi(a_i)$  then
5:      $a_{j+1} = a_j$ ;  $b_{j+1} = a_j$ ;
6:   else
7:     evaluate  $\phi'(a_j)$ ;
8:     if  $|\phi'(a_j)| \leq -\sigma\phi'(0)$  then
9:       terminate.
10:    end if
11:    if  $(b_j - a_j)\phi'(a_j) \geq 0$  then
12:       $b_{j+1} = a_j$ ;
13:    else
14:       $b_{j+1} = b_j$ ;
15:    end if
16:     $a_{j+1} = a_j$ ;
17:  end if
18: end for

```

定理 4.4.1 若 $\sigma > \rho$ 且初始覆盖 $[a_i, b_i]$ 是恰当覆盖, 则分割算法 4.4.2 必定终止于满足强 Wolfe 条件的可接受点 a_i .

证明 假如算法没有终止, 则由式(4.4.2)有 $|a_i - b_i| \rightarrow 0$.

由算法 4.4.2 有

$$[a_{j+1}, b_{j+1}] \subset [a_j, b_j]$$

因此, 存在极限点 c 使得 $a_j \rightarrow c, b_j \rightarrow c$, 因此有 $a_j \rightarrow c$. 因为 a_j 满足条件(4.3.1), 但不满足式(4.3.6), 从而 c 满足条件(4.3.1)且

$$|\phi'(c)| \geq -\sigma\phi'(0) \quad (4.4.3)$$

设存在一个无限的覆盖子列满足 $b_j < a_j$. 由定义 4.4.1 的(iii), $b_j < a_j$ 以及 a_j 满足 Armijo 条件(4.3.1), 有 $\phi(b_j) - \phi(a_j) \geq 0$, 因此由中值定理和 c 的存在性有 $\phi'(c) \leq 0$. 但是由定义 4.4.1 的(ii), $\phi'(a_j)(b_j - a_j) < 0$ 蕴含着极限时 $\phi'(c) \geq 0$, 从而 $\phi'(c) = 0$, 这与不等式(4.4.3)矛盾. 从而对充分大的 j 有 $a_j \uparrow c$ 和 $b_j \downarrow c$, 因此仅考虑 $b_j > a_j$ 的情况. 由定义 4.4.1 的(iii)和 a_j 满足条件(4.3.1)可得

$$\phi(b_j) - \phi(a_j) \geq \rho\phi'(0)(b_j - a_j)$$

因此, 中值定理和 c 的存在性蕴含着 $\phi'(c) \geq \rho\phi'(0)$. 但是 $\phi'(a_j)(b_j - a_j) < 0$ 蕴含着 $\phi'(c) \leq 0$. 因为 $\sigma > \rho$, 这些不等式与式(4.4.3)矛盾. 从而算法必定终止于满足强 Wolfe 条件的可接受点 a_j . ■

从实用的目的讲, 定理 4.4.1 表明应该选取参数 $\sigma > \rho$, 这样保证分割算法 4.4.2 终止于可接受点.

通常在算法 4.4.1 和算法 4.4.2 中使用插值. 给定区间 $[0, 1]$, 数据 ϕ_0, ϕ'_0 和 ϕ'_1 (这里 $\phi_0 = \phi(0)$, 其余依此类推), 则唯一的二次插值函数是

$$q(z) = \phi_0 + \phi'_0 z + (\phi'_1 - \phi'_0 - \phi'_0)z^2$$

若 ϕ'_1 也已知, 则相应的(Hermite)三次插值多项式是

$$c(z) = \phi_0 + \phi'_0 z + \eta z^2 + \xi z^3$$

其中

$$\eta = 3(\phi_1 - \phi_0) - 2\phi'_0 - \phi'_1, \quad \xi = \phi'_0 + \phi'_1 - 2(\phi_1 - \phi_0)$$

算法 4.4.1 和算法 4.4.2 在区间 $[a, b]$ 而非 $[0, 1]$ 上操作, 且有可能 $a > b$. 为此, 利用变换

$$\alpha = a + z(b - a)$$

将 $[0, 1]$ 映射到 $[a, b]$. 根据链式法则 $\frac{d\phi}{dz} = \frac{d\phi}{d\alpha} \frac{d\alpha}{dz} = (b - a) \frac{d\phi}{d\alpha}$, 可以把一维搜索中得到的 $\frac{d\phi}{d\alpha}$ 的导数值映射到这里需要的导数值 ϕ'_0 和 ϕ'_1 . 需要检查稳定点处的值、区间端点的值和导数等, 才能得到 $q(z)$ 或者 $c(z)$ 在区间上的极小点. 下面用一个例子来说明这种一维搜索中的划界、分割和插值过程.

例 4.4.1 考虑 Rosenbrock 函数(1.4.2), 令 $\mathbf{x}^{(k)} = \mathbf{0}, \mathbf{p}^{(k)} = (1, 0)^T$. 则 $\phi(\alpha) = 100\alpha^4 + (1 - \alpha)^2, \phi'(\alpha) = 400\alpha^3 - 2(1 - \alpha)$. 使用参数 $\sigma = 0.1, \rho = 0.01, \tau_1 = 9, \tau_2 = 0.1$ 和 $\tau_3 = 0.5$. 一维搜索的过程见表 4.4.1.

表 4.4.1 使用一阶导数的一维搜索

Starting from $\alpha_1 = 0.1$				
α	0	0.1	0.2	0.160 948
$\phi(\alpha)$	1	0.82	0.8	0.771 111
$\phi'(\alpha)$	-2	-1.4	1.6	-0.010 423
Starting from $\alpha_1 = 1$				
α	0	1	0.1	0.19
$\phi(\alpha)$	1	100	0.82	0.786 421
$\phi'(\alpha)$	-2	—	-1.4	1.123 6
				-0.011 269

第 1 部分是初始猜测 $\alpha_1 = 0.1$ 时的结果. 这时初始区间不是覆盖, 因此划界算法要求在区间 $[0.2, 1]$ 中选下一个迭代. 映射 $[0, 0.1]$ 到 z 空间的 $[0, 1]$, 得到的三次拟合 $c(z) = 1 - 0.2z + 0.02z^3$, 且 $c(z)$ 在 $[2, 10]$ 中的极小值在 $z = 2$ 处取到. 这样, $\alpha_2 = 0.2$ 是下一次迭代. 这个迭代给出了恰当覆盖 $[0.2, 0.1]$. 紧接其后的分割算法在 $[0.19, 0.15]$ 中寻找新迭代. 映射 $[0.2, 0.1]$ 到 $[0, 1]$, 得到 $c(z) = 0.8 - 0.16z + 0.24z^2 - 0.06z^3$. 这个三次函数在区间 $[0.1, 0.5]$ 内的局部极小点 $z = 0.390 524$, 是局部无约束的; 这样, 新的迭代 $\alpha_3 = 0.160 948$. 经判断这个点是可接受的, 终止线搜索.

第 2 部分是一维搜索从 $\alpha_1 = 1$ 开始时的结果. 这时初始值 $\phi(1) > \phi(0)$, 因此这个点确定一个覆盖, 且不必计算 $\phi'(1)$ (这可避免无谓的计算梯度), 从而进入分割阶段. 二次插值得到 $q(\alpha) = 1 - 0.2\alpha + 99.2\alpha^2$, 它在区间 $[0.1, 0.5]$ 的极小点 $\alpha_2 = 0.1$. 这样, $[0.1, 1]$ 是新的恰当覆盖, 接下来的迭代在区间 $[0.19, 0.55]$ 中选取. 我们可以在 $\alpha = 0.1$ 和 $\alpha = 1$ 处进行二次插值, 或者在 $\alpha = 0$ 和 $\alpha = 0.1$ 处进行三次插值(表格中的结果是后者)得到这个新的迭代. 对这个例子来说, 这两种情况得到的极小点都是 $\alpha_3 = 0.19$, 从而得到新的覆盖 $[0.19, 0.1]$. 这个覆盖中的三次插值得到的 $\alpha_4 = 0.160 922$ 是一个可接受点.

上面算法的描述忽略了舍入误差的影响. 当 $x^{(k)}$ 接近 x^* 时, 因为舍入误差的影响可能会出现一些数值困难. 此时, 尽管 $\phi'(0) < 0$, 但是由于舍入误差的影响, 仍然有可能对所有 α 有 $\phi(\alpha) \geq \phi(0)$. 如果用户在其导数公式中引入误差, 也有可能出现这种情况. 因此, 通常建议在算法 4.4.2 的第 3 行后, 判断 $(\alpha_i - \alpha_j)\phi'(\alpha_i) \leq \epsilon$ 是否成立. 若成立, 则终止算法, 其中 ϵ 与式 (4.2.3) 中的相同, 是对目标函数的容许误差. 此时表明一维搜索没有取得进展, 从而能及时终止极小化算法.

注意 α_1 的初始选取也很重要. 若能够得到一维搜索中 f 下降量的估计 $\Delta f > 0$, 则基于 $\phi'(0)$ 和 Δf 得到的二次函数可以给出估计

$$\alpha_1 = \frac{-2\Delta f}{\phi'(0)} \quad (4.4.4)$$

对于极小化方法的首次迭代, 必须由用户提供 Δf . 以后令 $\Delta f = \max(f^{(k+1)} - f^{(k)}, 10\epsilon)$ 即可, 这是加上了恰当保护的上次迭代的下降量. 然而, 对于牛顿型的方法, $\alpha_k = 1$ 对最终的快速收敛很重要, 此时使用

$$\alpha_1 = \min\left(1, \frac{-2\Delta f}{\phi'(0)}\right)$$

下面介绍无导数方法。对于任意的3个 α 值 $\alpha_1 < \alpha_2 < \alpha_3$ ，当 $\phi_2 \leq \min(\phi_1, \phi_3)$ 时形成一个覆盖，见图4.4.1。一旦找到一个这样的覆盖，通过分割或者插值，或者两者的某种组合来缩短区间。

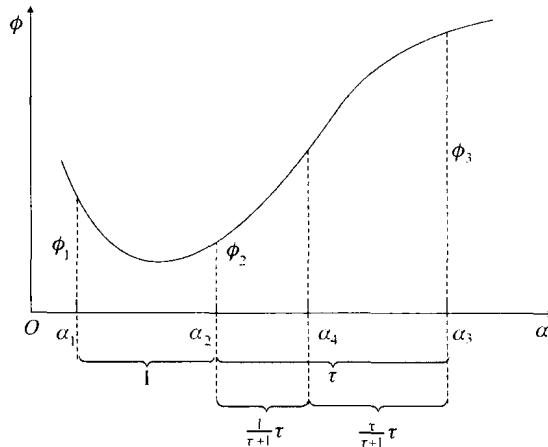


图4.4.1 黄金分割法

一个简单的纯粹分割的方法是**黄金分割法**(golden section search)，这里 $(\alpha_3 - \alpha_2) : (\alpha_2 - \alpha_1)$ 的值固定为 $\tau:1$ (或者 $1:\tau$)，其中 $\tau > 1$ 。当插入新点 α_1 来缩短区间时，要求两个潜在的新覆盖 $(\alpha_1, \alpha_2, \alpha_1)$ 和 $(\alpha_2, \alpha_3, \alpha_1)$ 形成长度相同的区间。在图4.4.1中，由对称性 $\alpha_3 - \alpha_1 = \alpha_2 - \alpha_1$ ，得到 τ 满足 $1 = \tau^2/(\tau+1)$ ，即为 $\tau^2 - \tau - 1 = 0$ 的正根，解得 $\tau = (1 + \sqrt{5})/2 \approx 1.618$ 。该方法需要计算新点处的 ϕ ，并选取包含极小点的那个新区间，在图中为 $(\alpha_1, \alpha_2, \alpha_1)$ 。重复该过程直到终止测试满足。黄金分割法每步区间缩短为原来的0.618，所以也称为0.618分割法。应用黄金分割法找函数

$$\phi(\alpha) = 1 - \alpha e^{-\alpha^2} \quad (4.4.5)$$

的极小点，要求保留一位有效数字的计算结果见表4.4.2。初始覆盖是 $(0, 0.618, 1)$ ，终止时的覆盖是 $(0.674, 0.708, 0.764)$ 。**Fibonacci分割法**(Fibonacci section search)也是利用类似的思想，主要区别在于搜索区间长度的缩短率不是固定的0.618，而是采用Fibonacci数列。人们能够证明它是此类一维分割法的最优策略，但要注意 n 趋于无穷大时，Fibonacci缩短率极限为0.618，在这个意义上，黄金分割法是近似最优方法。同时，由于黄金分割法简单实用，所以得到广泛应用。

表4.4.2 黄金分割法的例子

j	1	2	3	4	5	6	7	8
α	0	0.618	1.000	0.382	0.764	0.854	0.708	0.674
$\phi(\alpha)$	1	0.578	0.632	0.670	0.574	0.588	0.571	0.572

不幸的是，仅使用分割法有可能很无效(见习题4.9)。如果以某种方式使用插值，则会使分割法更有效。例如，给定覆盖 $(\alpha_1, \alpha_2, \alpha_3)$ ，基于 ϕ_1, ϕ_2 和 ϕ_3 可以得到二次插值多项式，将它的极小点选作下一个测试点。这样做通常会很有效，但是必须辅以某种分割的思想。在图4.4.1中，由二次插值得到的新测试点 $\alpha = \alpha_2$ 。然而，在无导数方法中并没有很好的机制将插值和分割结合起来。

若把黄金分割法推广到求解 n 维空间的优化问题, 则每次需要引进 $2^n - 1$ 个新点才能维持结构的继承性. 当 n 较大时, 相对于均匀洒点(需要 2^n 个点)已经没有优势, 所以一般不讲 n 维空间中的黄金分割法.

4.5 评注与参考

读者应该熟悉方程求根的二分法, 在该方法中需要初始覆盖的左右端点函数值异号, 每步迭代都加入一个中间点, 保留该中间点和一个端点以使得新区间两端点函数值仍然异号. 这样, 区间缩短率为 0.5, 故称二分法. 将求根与求极小点对应起来, 读者不难发现与二分法对应的分割方法正是黄金分割法. 不同之处是求极小点时需要高-低-高 3 个点, 才能保证区间内一定有一个极小点; 相同之处都是添加一个新内点, 删除一个端点, 且新内点的位置保证无论删除哪个端点之后剩下的区间长度都一致. 从这个角度来看, 读者也可以推导出黄金分割法.

注意, 精确线搜索本身是个一维最优化问题, 完全可以套用后面章节中所描述的牛顿法或拟牛顿法, 这项工作留给读者自己完成. 另外, 精确线搜索的大范围收敛性自然包含在定理 4.3.4 中, 这是因为它对应着强 Wolfe 条件在 $\sigma=0$ 的特例.

最后需要说明的是, 本章介绍的是单调线搜索. 它的主要弊端是有可能使迭代陷入局部极小点. 近年来研究者也在探讨使用非单调线搜索以避免这些问题.

习题 4

- 4.1 说明函数 $f(x) = 8x_1 + 12x_2 + x_1^2 - 2x_2^2$ 仅有一个稳定点, 且它既不是最小点, 也不是最大点, 而是一个鞍点. 画出 f 的等值线.
- 4.2 考虑问题 $\min_{x \in \mathbb{R}^n} \|Ax - b\|^2$, 其中 A 是 $m \times n$ 矩阵, b 是 m 维向量.
- 给出问题的几何解释.
 - 写出最优性的必要条件. 它也是一个充分条件吗?
 - 最优解唯一吗? 理由是什么?
 - 你能给出最优解的一种闭合形式(解析式)吗? 允许规定任何你所需的假设.
 - 求解该问题, 其中

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

- 4.3 对于二次函数 $q(x) = \frac{1}{2} x^T G x - b^T x$, 证明:

- 当且仅当 G 半正定且 $Gx = b$ 有解时, 极小点存在.
- 当且仅当 G 正定时, 极小点唯一.
- 如果 G 半正定, 则每一个稳定点都是全局极小点.

4.4 对标量 β 的每个值,找到函数

$$f(x, y) = x^2 + y^2 + \beta xy + x + 2y$$

的所有稳定点. 这些稳定点中的哪些是全局极小点?

4.5 针对下面的每一个问题,利用最优化条件充分解释你的答案.

(a) 说明二元函数 $f(x, y) = (x^2 - 4)^2 + y^2$ 有两个全局极小点和一个稳定点,该稳定点既不是局部极大点,也不是局部极小点.

(b) 找到二元函数 $f(x, y) = \frac{1}{2}x^2 + x \cos y$ 的局部极小点.

(c) 找到二元函数 $f(x, y) = \sin x + \sin y + \sin(x+y)$ 在集合 $\{(x, y) \mid 0 < x < 2\pi, 0 < y < 2\pi\}$ 上的所有局部极小点和局部极大点.

(d) 说明二元函数 $f(x, y) = (y - x^2)^2 - x^2$ 仅有一个稳定点,它既不是局部极大点,也不是局部极小点.

4.6 设 C 是 \mathbb{R}^n 中的凸集, $f: C \rightarrow \mathbb{R}$. 证明 f 是 C 上的凸函数当且仅当对任一整数 $k \geq 2$, $x_i \in C, \theta_i \geq 0, i=1, 2, \dots, k$, $\sum_{i=1}^k \theta_i = 1$ 蕴含着 $f(\sum_i \theta_i x_i) \leq \sum_i \theta_i f(x_i)$.

4.7 考虑以下序列

(a) $x^{(k)} = 1/k$,

(b) $x^{(k)} = (0.5)^{2^k}$,

(c) $x^{(k)} = 1/(k!)$,

确定它们的收敛阶;并说明为了使 $x^{(k)} < 10^{-1}$, 每个序列的 k 至少取多大?

4.8 考查函数 $f(x_1, x_2) = (x_1 + x_2^2)^2$ 在点 $x^{(k)} = (1, 0)^T$ 处的搜索方向 $p^{(k)} = (-1, 1)^T$, 证明 $p^{(k)}$ 是所给函数在 $x^{(k)}$ 处的下降方向,并求出一维极小化问题 $\min_{\alpha \geq 0} f(x^{(k)} + \alpha p^{(k)})$ 的所有极小点.

4.9 对于表 4.4.2 中的数据,给出由点 α_1, α_2 和 α_3 的二次插值函数得到的极小点 α 的估计值. 说明这个值在由黄金分割法所确定的最后一个区间内,但不能保证有相同的精度.

4.10 利用黄金分割法确定函数(4.4.5)的极小点(误差不超过 0.1). 验证 $\alpha_1 = 0, \alpha_2 = 1.236, \alpha_3 = 2$ 可以确定一个初始覆盖,由此出发确定函数的极小点(精确到小数点后 3 位). 再基于这 3 个点进行二次插值,给出所得到的结果.

4.11 对于函数(4.4.5),分别确定满足 Goldstein 条件、Wolfe 条件以及强 Wolfe 条件的 α 值的可接受区间,其中 $\sigma = \rho = 1/10$ 及 $\sigma = \rho = 1/4$. 对于后一种情况,以 $\alpha = 0$ 和 1 为初值,应用算法 4.4.1 和算法 4.4.2 得到一个可接受点.

4.12 使用以上两个问题中的一维搜索方法极小化函数 $1 - (5\alpha^2 - 6\alpha + 5)^{-1}$,且取相同的初始点. 验证黄金分割法能够得到很好的初值估计,但在缩小区间方面却相对较慢.

4.13 考虑连续可微的二次样条函数

$$\phi(\alpha) = \begin{cases} \frac{1}{2} \left(\frac{1 - \sigma^+}{t} \right) \alpha^2 - \alpha, & \alpha \leq t \\ \frac{1}{2} (\sigma^+ - 1) t - \sigma^+ \alpha, & \alpha \geq t \end{cases}$$

其中 σ^+ 远大于 σ 且 $\sigma < \rho < 1/2$. 说明它的图像与 ρ -线相交于 $\alpha = 1$,但没有点满足 Wolfe 条件.

第 5 章 无约束优化: 线搜索法

由第 4 章可知, 在确定搜索方向的二次模型(4.2.9)中选取不同的 $\mathbf{B}^{(k)}$ 可以得到不同的方法, 比如由 $\mathbf{B}^{(k)} = \mathbf{I}$ 可推导出最速下降法, 而若取 $\mathbf{B}^{(k)} = \mathbf{G}^{(k)}$ 则可得牛顿法. 这两种方法是最基本的优化方法, 是其他很多实用方法的基础. 其中, 最速下降法线性收敛; 牛顿法二次收敛, 但每步都需要计算 Hessian 阵, 计算量和存储量大.

如何设计存储量适中的求解超大规模的无约束优化问题是一个极具挑战性的问题. 5.2 节介绍的共轭梯度法具有二次终止性、内存需要量小、程序简单等优点, 它是无约束优化, 特别是大规模问题的一个重要方法.

如何设计避免使用二阶导数的方法是方法实用化的另一个基本要求. 5.3 节介绍的拟牛顿法仅使用一阶导数, 具有超线性收敛性, 是解决中小规模无约束优化问题的实用方法.

在应用中, 也可以依据问题的结构设计更有效的方法, 5.4 节介绍的最小二乘问题是一个很典型的例子.

5.1 基本方法

5.1.1 最速下降法

在确定搜索方向的二次模型(4.2.9)中, 最简单的选取方式是 $\mathbf{B}^{(k)} = \mathbf{I}$, 即单位矩阵, 此时 $\mathbf{p}^{(k)} = -\mathbf{g}^{(k)}$, 称 $-\mathbf{g}^{(k)}$ 为最速下降(steepest-descent)方向, 它求解问题

$$\begin{aligned} & \underset{\mathbf{p} \in \mathbb{R}^n}{\text{minimize}} \quad \mathbf{g}^{(k)^\top} \mathbf{p} \\ & \text{subject to} \quad \|\mathbf{p}\|_2 = 1 \end{aligned}$$

这样, 就单位方向而言, 负梯度方向是目标函数在 $\mathbf{x}^{(k)}$ 下降最快的方向. 注意, 最速下降方向与梯度方向平行, 永远不会正交. 这样搜索方向与负梯度方向的夹角 $\theta_k = 0$, 即 $\cos \theta_k = 1$. 由定理 4.3.2 和定理 4.3.4 立即可得如下的大范围收敛性.

定理 5.1.1 (大范围收敛) 假设 $f \in C^1$ 且 $\mathbf{g}(\mathbf{x})$ 是 Lipschitz 连续的, f 下方有界. 则对使用最速下降方向的线搜索法产生的迭代 $\{\mathbf{x}^{(k)}\}$ 而言, 无论步长满足 Wolfe 条件、强 Wolfe 条件, 还是由 Armijo 法则确定, 均有 $\lim_{k \rightarrow \infty} \mathbf{g}^{(k)} = \mathbf{0}$.

例 5.1.1 (最速下降法) 考虑 $\min f(\mathbf{x}) = \frac{1}{2} (x_1^2 + 5x_2^2 + 25x_3^2) + x_1 + x_2 + x_3$. 该问题的解 $\mathbf{x}^* = (-1, -1/5, -1/25)^\top$, 最速下降方向 $\mathbf{p}^{(k)} = -\mathbf{g}(\mathbf{x}^{(k)})$. 如果解的初始猜测 $\mathbf{x}^{(0)} = (0, 0, 0)^\top$, 使用精确线搜索法, 即步长由式(4.2.11)确定, 则迭代中的部分数据如表 5.1.1 所列.

从例 5.1.1 中可以看出: 即使从很好的初始点出发, 为了使 $\|\mathbf{g}^{(k)}\| < 10^{-8}$, 都需要迭代 200 多次. 下面探讨最速下降法的收敛速度, 将证明最速下降法是线性收敛的, 但是收敛因子高度

依赖于最优解 \mathbf{x}^* 处的 Hessian 阵 \mathbf{G}^* 的条件数。为了证明该结论，先考虑理想情况，即目标函数是正定二次函数且线搜索是精确的，这时可得最速下降法的很多性质。假设

$$q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{G} \mathbf{x} - \mathbf{b}^\top \mathbf{x} \quad (5.1.1)$$

则梯度 $\mathbf{g}(\mathbf{x}) = \mathbf{G}\mathbf{x} - \mathbf{b}$ ，且极小点 \mathbf{x}^* 是线性方程组 $\mathbf{G}\mathbf{x} = \mathbf{b}$ 的唯一解 $\mathbf{G}^{-1}\mathbf{b}$ 。利用精确步长(4.2.11)可得最速下降迭代

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)}}{\mathbf{g}^{(k)^\top} \mathbf{G} \mathbf{g}^{(k)}} \mathbf{g}^{(k)} \quad (5.1.2)$$

因为 $\mathbf{g}^{(k)} = \mathbf{G}\mathbf{x}^{(k)} - \mathbf{b}$ ，从而该等式给出用 $\mathbf{x}^{(k)}$ 表示 $\mathbf{x}^{(k+1)}$ 的显式形式。

表 5.1.1 例 5.1.1 的部分迭代数据

k	$\mathbf{x}^{(k)}$	$\ \mathbf{g}^{(k)}\ $	$f^{(k)}$	α_k
0	(0.000 0, 0.000 0, 0.000 0) ^T	1.732 1	0	0.096 8
1	-(0.096 8, 0.096 8, 0.096 8) ^T	1.759 8	-0.145 2	0.059 0
2	-(0.150 0, 0.127 2, 0.013 1) ^T	1.143 7	-0.236 5	0.102 9
⋮	⋮	⋮	⋮	⋮
216	-(1.000 0, 0.200 0, 0.040 0) ^T	9.009×10^{-3}	-0.620 0	—

为了量化收敛速率，引入加权范数 $\|\mathbf{x}\|_G^2 := \mathbf{x}^\top \mathbf{G} \mathbf{x}$ 。利用关系 $\mathbf{G}\mathbf{x}^* = \mathbf{b}$ 易验证 $\frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|_G^2 = q(\mathbf{x}) - q^*$ 。这样，该范数度量了当前目标值和最优值之差。将式(5.1.2)代入 $\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_G^2$ ，并利用 $\mathbf{g}^{(k)} = \mathbf{G}(\mathbf{x}^{(k)} - \mathbf{x}^*)$ 得

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_G^2 = \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_G^2 - \frac{(\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)})^2}{\mathbf{g}^{(k)^\top} \mathbf{G} \mathbf{g}^{(k)}}$$

又因为 $\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_G^2 = \mathbf{g}^{(k)^\top} \mathbf{G}^{-1} \mathbf{g}^{(k)}$ ，所以

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_G^2 = \left\{ 1 - \frac{(\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)})^2}{(\mathbf{g}^{(k)^\top} \mathbf{G} \mathbf{g}^{(k)}) (\mathbf{g}^{(k)^\top} \mathbf{G}^{-1} \mathbf{g}^{(k)})} \right\} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_G^2$$

该表达式描述了 $q(\mathbf{x})$ 在每一步的精确减小量，然而括号里面的表达式的意义不明朗。设 λ_1 和 λ_n 分别是 \mathbf{G} 的最大和最小特征值，借助引理 5.1.1 中的不等式，可以用一个不依赖于当前迭代 k 的上界来刻画上述表达式。

引理 5.1.1 (Kantorovich 不等式) 对于 n 阶对称正定矩阵 \mathbf{G} ，有

$$\frac{(\mathbf{x}^\top \mathbf{x})^2}{(\mathbf{x}^\top \mathbf{G} \mathbf{x})(\mathbf{x}^\top \mathbf{G}^{-1} \mathbf{x})} \geq \frac{4\lambda_n \lambda_1}{(\lambda_n + \lambda_1)^2}, \quad \forall \mathbf{x} \in \mathbb{R}^n$$

定理 5.1.2 (线性收敛) 使用精确线搜索的最速下降法极小化二次函数(5.1.1)时，对所有 k 满足

$$q^{(k+1)} - q^* \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^2 (q^{(k)} - q^*) \quad (5.1.3)$$

不等式(5.1.3)说明序列 $\{q^{(k)}\}$ 线性收敛到极小值 q^* 。作为该结论的一种特殊情况，如果所有的特征值都相等，则仅需一步迭代。在这种情况下， \mathbf{G} 是单位矩阵的倍数，因此函数(5.1.1)的等值线是圆，最速下降方向总是指向最优解。一般地，若条件数 $\kappa(\mathbf{G}) = \lambda_1/\lambda_n$ 增加，则二次

函数的等值线会变得更加狭长,图 5.1.1 中的锯齿形会变得更显著. 尽管式(5.1.3)是最坏情况的界,但它精确指出了 $n \geq 2$ 时算法的最坏行为. 表 5.1.2 列出了条件数变化时目标函数的误差减小一个数量级所需的迭代次数 l .

表 5.1.2 最速下降法的收敛常数对 Hessian 阵条件数的灵敏度

$\frac{\lambda_1}{\lambda_n}$	$\left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \right)^2$	l
1.1	0.0023	1
3.0	0.25	2
10.0	0.67	6
100.0	0.96	58
200.0	0.98	116
400.0	0.99	231

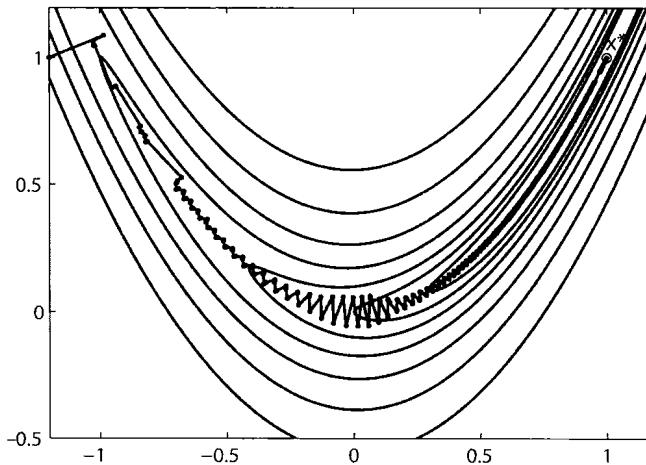


图 5.1.1 最速下降法的典型行为(步长由回溯 Armijo 线搜索确定)

对一般非线性目标函数,最速下降法的上述收敛行为本质上是相同的. 下面介绍具体结论.

定理 5.1.3 (线性收敛) 假设 $f(\mathbf{x}) \in C^2$, 设利用精确步长的最速下降法产生的迭代序列收敛到 \mathbf{x}^* , 且在 \mathbf{x}^* 处 Hessian 阵 \mathbf{G}^* 是正定的. 设 $r \in \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}, 1 \right)$, 其中 λ_1 和 λ_n 分别是 \mathbf{G}^* 的最大和最小特征值. 则对所有充分大的 k 有 $f^{(k+1)} - f^* \leq r^2 (f^{(k)} - f^*)$.

通常,如果使用非精确线搜索,将不期望收敛速度会提高. 因此,定理 5.1.3 说明最速下降法的速率常数可能会非常小,即使 Hessian 阵的条件数相当好,比如 $\kappa(\mathbf{G}^*) = 800$, $f^{(0)} = 1$, 且 $f^* = 0$. 该定理表明用最速下降法迭代 500 多次后,函数值将大约是 0.08.

像上面提到的,该定理表明最速下降法实际上是理想的大范围收敛的方法,所以在实践中,当许多其他的方法出现问题时,也求助于最速下降法. 但糟糕的是,该方法从理论上可能(实际上总是这样)会非常慢,有时会出现数值上的不收敛,仿佛迭代停滞不动了一般(如图 5.1.1 所示). 实践中,纯粹使用最速下降法在大多数情况下表现不佳. 图 5.1.1 展示了最速下降法非常典型的行为,这里迭代从目标函数“山谷”的一边振荡到另一边. 所有这些现象

可能要归咎于在构造搜索方向时没有利用问题的曲率.

5.1.2 牛顿法

在确定搜索方向的二次模型(4.2.9)中选取 $\mathbf{B}^{(k)} = \mathbf{G}^{(k)}$, 即 $\mathbf{x}^{(k)}$ 处的 Hessian 阵, 也即取二次模型为目标函数 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 的二阶 Taylor 展式

$$q^{(k)}(\mathbf{s}) := f^{(k)} + \mathbf{g}^{(k)\top} \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{G}^{(k)} \mathbf{s} \quad (5.1.4)$$

其中 $\mathbf{s} = \mathbf{x} - \mathbf{x}^{(k)}$. 如果 $\mathbf{G}^{(k)}$ 还是正定的, $q^{(k)}(\mathbf{s})$ 有唯一极小点 $\mathbf{s}^{(k)}$. 若不进行一维搜索, 即直接取步长为 1, 则得新的迭代点 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$, 称其为基本牛顿法(local Newton method). 这时, 由条件(4.1.1)和二次函数的梯度公式知, 牛顿法第 k 次迭代可以写为

- (a) 解方程组 $\mathbf{G}^{(k)} \mathbf{s} = -\mathbf{g}^{(k)}$ 得 $\mathbf{s}^{(k)}$;
- (b) 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.

步骤(a)要解 $n \times n$ 阶线性方程组. 最方便的方法是进行矩阵分解 $\mathbf{G}^{(k)} = \mathbf{L}\mathbf{D}\mathbf{L}^\top$ (见 4.1 节), 在解方程组的同时还能检验矩阵的正定性. 每次迭代都需要 $n^3/6 + O(n^2)$ 次乘法运算. 称 $\mathbf{s}_N^{(k)} = -\mathbf{G}^{(k)^{-1}} \mathbf{g}^{(k)}$ 是牛顿方向/牛顿步.

例 5.1.2 (牛顿法) 设 $f(x) = 7x - \ln x$, 则 $\mathbf{g}(x) = f'(x) = 7 - 1/x$, $\mathbf{G}(x) = f''(x) = 1/x^2$. 不难验证 $x^* = 1/7 \approx 0.142857143$ 是唯一的全局极小点. 在 $\mathbf{x}^{(k)}$ 处的牛顿步

$$\mathbf{s}^{(k)} = -f'(x^{(k)})/f''(x^{(k)}) = x^{(k)} - 7x^{(k)^2}$$

牛顿法产生的迭代 $\mathbf{x}^{(k+1)} = 2x^{(k)} - 7x^{(k)^2}$. 表 5.1.3 给出初始点不同时牛顿法所产生的迭代序列.

表 5.1.3 牛顿法对初始点的依赖性

k	$\mathbf{x}^{(k)}$	$\mathbf{x}^{(k)}$	$\mathbf{x}^{(k)}$	$\mathbf{x}^{(k)}$
0	1.0	0	0.1	0.01
1	-5.0	0	0.13	0.0193
2	-185.0	0	0.1417	0.03599257
3	-239.945.0	0	0.14284777	0.062916884
4	-4.0302 $\times 10^{11}$	0	0.142857142	0.098124028
5	-1.1370 $\times 10^{21}$	0	0.142857143	0.128849782
6	-9.0486 $\times 10^{48}$	0		0.1414837
7	-5.7314 $\times 10^{98}$	0		0.142843938
8	-	0		0.142857142
9	-	0		0.142857143

例 5.1.3 (牛顿法) 设 $f(\mathbf{x}) = -\ln(1 - x_1 - x_2) - \ln x_1 - \ln x_2$, 则

$$g(\mathbf{x}) = \begin{bmatrix} \frac{1}{1 - x_1 - x_2} - \frac{1}{x_1} \\ \frac{1}{1 - x_1 - x_2} - \frac{1}{x_2} \end{bmatrix}, \quad \mathbf{G}(\mathbf{x}) = \begin{bmatrix} \frac{1}{(1 - x_1 - x_2)^2} + \frac{1}{x_1^2} & \frac{1}{(1 - x_1 - x_2)^2} \\ \frac{1}{(1 - x_1 - x_2)^2} & \frac{1}{(1 - x_1 - x_2)^2} + \frac{1}{x_2^2} \end{bmatrix}$$

不难验证 $x^* = (1/3, 1/3)^T$ 是唯一的全局极小点, 且 $f^* \approx 3.295\ 836\ 866$. 表 5.1.4 给出了牛顿法所产生的序列.

表 5.1.4 牛顿法的二阶收敛性

k	$x_1^{(k)}$	$x_2^{(k)}$	$\ x^{(k)} - x^*\ $
0	0.85	0.05	0.589 255 650 988 79
1	0.717 006 802 721 08	0.096 598 639 455 78	0.450 831 061 926 01
2	0.512 975 199 133 20	0.176 479 706 723 55	0.238 483 249 157 46
3	0.352 478 577 567 27	0.273 248 784 105 08	0.063 061 029 429 74
4	0.338 449 016 006 35	0.326 238 070 059 96	0.008 747 169 263 80
5	0.333 337 722 134 80	0.333 259 330 511 65	$7.413 284 828 371 95 \times 10^{-5}$
6	0.333 333 343 617 61	0.333 333 327 241 28	$1.195 322 118 554 43 \times 10^{-8}$
7	0.333 333 333 333 33	0.333 333 333 333 33	$1.570 092 458 683 78 \times 10^{-16}$

在上面的例子中, 当初始点 $x^{(0)}$ 接近局部极小点 x^* 时, 对所有 k , 矩阵 $G^{(k)}$ 皆正定且方法所显示的快速收敛是牛顿法所具有的典型行为(见图 5.1.2). 如下定理是该事实的正式表述.

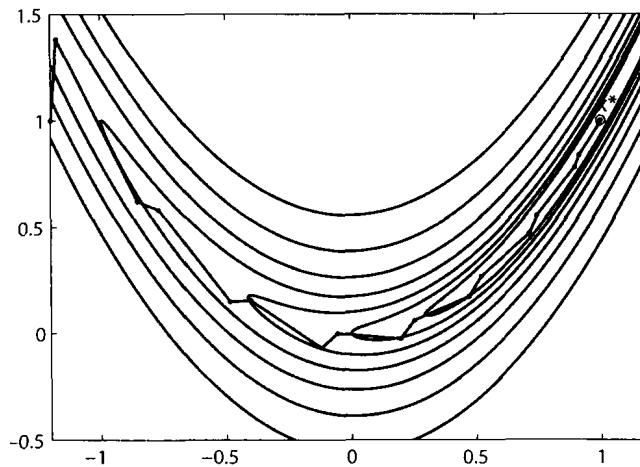


图 5.1.2 牛顿法的典型行为(步长由回溯 Armijo 线搜索确定)

定理 5.1.4 (局部二阶收敛) 假设 $f \in C^2$, $G(x)$ 是 Lipschitz 连续的. 设 x^* 是局部极小点, 且 G^* 正定. 如果初始点 $x^{(0)}$ 充分接近于 x^* , 则牛顿法有定义, 且产生的序列以二阶速率收敛于 x^* .

证明 由 $G(x)$ 的连续性及 G^* 正定, 存在 x^* 的半径为 δ' 的邻域 $N(x^*, \delta')$ 满足: 当 $x \in N(x^*, \delta')$ 时, $G(x)$ 正定, 且 $G(x)^{-1}$ 有界, 即存在 $M > 0$ 使得 $\|G(x)^{-1}\| \leq M$. 当 $x^{(0)}$ 位于该邻域内时, 牛顿法显然有定义, 且由迭代公式及 $g^* = \mathbf{0}$ 有

$$x^{(k+1)} - x^* = G^{(k)^{-1}} [G^{(k)}(x^{(k)} - x^*) - (g^{(k)} - g^*)]$$

依次由中值定理和 $G(x)$ 是 Lipschitz 连续的, 有

$$\|G^{(k)}(x^{(k)} - x^*) - (g^{(k)} - g^*)\| \leq L \|x^{(k)} - x^*\|^2$$

其中 L 是 Lipschitz 常数. 因此

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq LM \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2 \quad (5.1.5)$$

令 $\delta = \min\left(\delta', \frac{\alpha}{LM}\right)$, 其中 $\alpha \in (0, 1)$ 是常数. 若 $\mathbf{x}^{(0)} \in N(\mathbf{x}^*, \delta)$, 则由式(5.1.5)知 $\{\mathbf{x}^{(k)}\}$ 是压缩序列, 故收敛, 且是二阶收敛的. ■

必须强调的是, 上述二阶收敛的前提是初始点离解充分近. 当 $\mathbf{x}^{(k)}$ 远离最优解时, $\mathbf{G}^{(k)}$ 不一定正定; 此外, 即使 $\mathbf{G}^{(k)}$ 正定, 方法也不一定收敛(见例 5.1.2), 甚至 $f^{(k)}$ 不减小. 对于后一种可能性, 加上线搜索即可解决. 只要 $\mathbf{G}^{(k)}$ 正定, 牛顿方向就是下降方向, 从而沿牛顿方向

$$\mathbf{p}_N^{(k)} = -\mathbf{G}^{(k)^{-1}} \mathbf{g}^{(k)} \quad (5.1.6)$$

进行一维搜索. 因此实用牛顿法都结合某种线搜索确定步长 α_k , 得到新的迭代点 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}_N^{(k)}$, 此时第 4 章中关于一般下降法的收敛性结论对牛顿法也成立. 尽管可用任一种一维搜索算法确定步长, 但在实践中, 通常由算法 4.3.1 确定牛顿法中的步长. 在恰当的条件下, 可以证明对充分大的 $k, \alpha=1$ 满足 Armijo 条件. 这样, 带线搜索的牛顿法最后还原成基本牛顿法, 从而秉承了牛顿法二阶收敛的优点, 同时保证了方法的大范围收敛性.

对牛顿法进行修正时, 遇到的主要问题是 $\mathbf{G}^{(k)}$ 非正定, 即奇异或者至少有一个负特征值. 尽管大多数情况下, 由式(5.1.6)可以计算出 $\mathbf{p}^{(k)}$, 然后在 $\mathbf{p}^{(k)}$ 与 $-\mathbf{p}^{(k)}$ 中选择一个下降方向进行一维搜索. 但这时, 二次近似函数的稳定点未必是极小点, 因此沿着这样的方向进行一维搜索是要打问号的. 事实上, 考察函数

$$f(\mathbf{x}) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

即使结合一维搜索, 牛顿法也不收敛. 如取 $\mathbf{x}^{(0)} = (0, 0)^T$, 易验证

$$\mathbf{g}^{(0)} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad \mathbf{G}^{(0)} = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{p}^{(0)} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

因为沿 $\pm \mathbf{p}^{(0)}$ 进行一维搜索只改变第一个分量, 由目标函数的表达式, 易见 $x_1 = 0$ 是一维搜索的极小点, 因此算法没有进展. 出现该现象的原因是 $\mathbf{p}^{(0)T} \mathbf{g}^{(0)} = 0$, 从而 $\pm \mathbf{p}^{(0)}$ 均不是下降方向, 然而易见沿最速下降方向进行一维搜索时目标值可以减小. 用 $\pm \mathbf{p}^{(0)}$ 作为搜索方向失效的另一个典型例子是 $\mathbf{x}^{(0)}$ 是 $f(\mathbf{x})$ 的鞍点, 此时由于 $\mathbf{g}^{(0)}$ 本身是零, 从而由式(5.1.6)得不到搜索方向, 但沿着 $\mathbf{G}^{(0)}$ 负特征值对应的特征向量所确定的方向进行一维搜索可减小 f , 故 $\mathbf{x}^{(0)}$ 不是极小点.

因此, 为了使牛顿法变成一种实用方法, 还需要进行一些修正. 一种可能的方法是 $\mathbf{G}^{(k)}$ 非正定时, 求助于最速下降方向, 即此时放弃牛顿方向, 而将搜索方向选为负梯度方向. 然而, 如果连续几次迭代都要进行这种修正, 则方法很可能像最速下降法那样慢, 即产生振荡现象(如图 5.1.1 所示), 究其原因是忽略了二次函数(5.1.4)所提供的信息.

另一种可供选择的方法是对式(5.1.6)进行修正, 使其偏向最速下降方向 $-\mathbf{g}^{(k)}$. 最简单的做法是给 $\mathbf{G}^{(k)}$ 加上单位矩阵的倍数 $\lambda \mathbf{I}$, 再解方程

$$(\mathbf{G}^{(k)} + \lambda \mathbf{I}) \mathbf{p} = -\mathbf{g}^{(k)} \quad (5.1.7)$$

得到搜索方向 $\mathbf{p}^{(k)}$. 若 $\mathbf{G}^{(k)}$ 近似正定, 取较小的 $\lambda > 0$ 即可产生好的搜索方向. 因此, 从某种意义上说, 即使在 $\mathbf{G}^{(k)}$ 非正定的情况下, 这类方法也在一定程度上利用了函数的二次信息, 因而要比上述直接使用负梯度的修正方式更有效. 这里需要指出的是, 尽管是 Goldfeld 等人 1966 年首先把上述思想应用于牛顿法, 但是利用单位矩阵的倍数进行修正的思想在多年以前就被

提出了(Levenberg, 1944 年; Marquardt, 1963 年). 这种重要的思想促进了信赖域法的出现. 在式(5.1.7)中, λ 大致以反比例的方式决定着 $\mathbf{p}^{(k)}$ 的长度. 信赖域法则更直接地在步的长度受限制的条件下来极小化模型函数(5.1.4), 所以早期也称为限制步长法. 可以证明这类方法通常是大范围二阶收敛到满足二阶充分条件的点(定理 6.1.2). 这种方法的实现方式之一是按照 $\mathbf{G}^{(k)} + \lambda \mathbf{I}$ 的方式修正矩阵 $\mathbf{G}^{(k)}$, 详细讨论见第 6 章, 目前的应用已说明这种方法在实践中是有效的. 然而, 即使这样的方法也没有充分利用已有的二次信息, 特别是在鞍点附近(比如式(5.1.7)中 $\mathbf{g}^{(k)} = \mathbf{0}$ 时).

也有作者(Murray 和 Hebden)研究了不同的修正方式, 他们在式(5.1.7)中用正定矩阵 $\mathbf{G}^{(k)} + \mathbf{D}'$ (\mathbf{D}' 是对角矩阵) 来确定搜索方向; 修正是在矩阵分解过程中完成的. 具体地, 在分解过程中, 如果识别出 Cholesky 分解 $\mathbf{L}\mathbf{L}^T$ 不存在, 则进行启发式修正, 使得分解可以继续进行. 该方法的优点是 $\mathbf{G}^{(k)}$ 不定时计算分解因子所需的额外工作量可以忽略不计, 缺点同样是没有考虑 $\mathbf{x}^{(k)}$ 靠近鞍点时的情况.

5.2 共轭梯度法

共轭梯度法最初由 Hestenes 与 Stiefel 于 1952 年提出, 当时是作为求解大规模的系数矩阵对称正定的方程组

$$\mathbf{G}\mathbf{x} = \mathbf{b} \quad (5.2.1)$$

的方法. 在这种情况下, 求解方程组(5.2.1)与极小化二次函数(5.1.1)是等价的. 后来由 Fletcher、Reeves、Polak 与 Ribi  re 等将其推广, 用于非二次函数的极小化. 对于正定二次函数, 共轭梯度法的迭代次数不超过 Hessian 阵 \mathbf{G} 的互不相同特征值的个数. 因此, 当特征值高度聚集时, 其所需的迭代次数会更少. 基于该事实, 在应用共轭梯度法时, 通常会采用预条件技术来加速方法收敛.

5.2.1 扩展子空间定理

在引入共轭梯度法之前, 作为基础, 先介绍扩展子空间定理. 称向量 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 关于对称正定矩阵 \mathbf{G} 共轭(conjugacy), 如果

$$\mathbf{p}^{(i)^T} \mathbf{G} \mathbf{p}^{(j)} = 0, \quad \forall i \neq j \quad (5.2.2)$$

易验证相互共轭的方向必是线性无关的. 利用共轭概念构造搜索方向也可以得到二次终止性. 如果将方法应用于二次函数(5.1.1)时, 产生的方向是共轭的, 则称该方法是共轭方向法(conjugate direction method). 只要执行精确线搜索, 共轭方向法就具有二次终止性.

定理 5.2.1 (扩展子空间定理) 应用精确线搜索的共轭方向法极小化二次函数(5.1.1)时, 每个 $\mathbf{x}^{(k)}$ 是二次函数 $q(\mathbf{x})$ 在点集

$$F_k = \left\{ \mathbf{x} : \mathbf{x}^{(0)} + \sum_{i=0}^{k-1} \beta_i \mathbf{p}^{(i)}, \forall \beta_i \right\}$$

上的极小点. 该方法最多迭代 n 次后终止.

证明 为了证明结论成立, 仅需证明

$$\mathbf{g}^{(k)^T} \mathbf{p}^{(i)} = 0, \quad i = 0, 1, \dots, k-1 \quad (5.2.3)$$

即 $q(\mathbf{x})$ 沿着线性流形 F_k 中任一过 $\mathbf{x}^{(k)}$ 的直线的斜率是零，因为

$$\mathbf{g}^{(k)\top} \mathbf{p}^{(i)} = \mathbf{g}^{(i+1)\top} \mathbf{p}^{(i)} + \sum_{j=i+1}^{k-1} \mathbf{y}^{(j)\top} \mathbf{p}^{(i)}$$

其中 $\mathbf{y}^{(j)} = \mathbf{g}^{(j+1)} - \mathbf{g}^{(j)}$ 。因此，由精确线搜索的性质(4.2.10)，二次函数的性质(1.4.14)，以及共轭的定义(5.2.2)和 $\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)} = \alpha_i \mathbf{p}^{(i)}$ ，可以得到式(5.2.3)。

其次，由线搜索的迭代格式有 $\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \sum_{i=0}^{k-1} \alpha_i \mathbf{p}^{(i)}$ ，而且， $\forall \mathbf{x} \in F_k$ ，存在 β_i 使得 $\mathbf{x} = \mathbf{x}^{(0)} + \sum_{i=0}^{k-1} \beta_i \mathbf{p}^{(i)}$ 。再由可微凸函数的一阶充要条件(4.1.7)有

$$q(\mathbf{x}) \geq q(\mathbf{x}^{(k)}) + \mathbf{g}^{(k)\top} (\mathbf{x} - \mathbf{x}^{(k)})$$

将 \mathbf{x} 和 $\mathbf{x}^{(k)}$ 的特殊表达式代入该不等式，由式(5.2.3)知 $\mathbf{x}^{(k)}$ 是 $q(\mathbf{x})$ 在 F_k 上的解。假设迭代了 n 次，因为共轭方向是线性无关的，从而 $F_n = \mathbb{R}^n$ ，故 $\mathbf{x}^{(n)}$ 是函数 $q(\mathbf{x})$ 在 \mathbb{R}^n 上的极小点。 ■

证明虽易，但并不能帮助读者从本质上认清共轭性的基本结构。为此，把极小点 \mathbf{x}^* 表示成 $\mathbf{x}^* = \mathbf{x}^{(0)} + \sum_{i=0}^{n-1} \alpha_i^* \mathbf{p}^{(i)}$ ，任一点 \mathbf{x} 表示成 $\mathbf{x} = \mathbf{x}^{(0)} + \sum_{i=0}^{n-1} \beta_i \mathbf{p}^{(i)}$ ，则二次函数(5.1.1)(其中 $\mathbf{Gx}^* = \mathbf{b}$ ，并忽略常数 q^*)可用变量 $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_n)^\top$ 表示为

$$q(\boldsymbol{\beta}) = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^\top \mathbf{G} (\mathbf{x} - \mathbf{x}^*) = \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\alpha}^*)^\top \mathbf{P}^\top \mathbf{G} \mathbf{P} (\boldsymbol{\beta} - \boldsymbol{\alpha}^*)$$

其中 \mathbf{P} 是以 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 为列向量的矩阵。如果向量 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 是共轭的，则可化简为

$$q(\boldsymbol{\beta}) = \frac{1}{2} \sum_{i=0}^{n-1} (\beta_i - \alpha_i^*)^2 \mathbf{p}^{(i)\top} \mathbf{G} \mathbf{p}^{(i)}$$

易见选取 $\beta_0 = \alpha_0^*$, $\beta_1 = \alpha_1^*$, 依此类推即可极小化 $q(\boldsymbol{\beta})$ ，此即等价于在 \mathbf{x} 空间沿 $\mathbf{p}^{(i)}$ 进行精确线搜索。这样，共轭性蕴含着把 \mathbf{G} 对角化成 $\mathbf{P}^\top \mathbf{G} \mathbf{P}$ (即对原变量进行合同变换)，在这个新的坐标系($\boldsymbol{\beta}$)下，各个变量是相互分离的。共轭方向法即相当于在这个新坐标系中沿 $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ 进行线搜索。一言以蔽之，共轭方向法是将一个 n 维二次优化问题转化为 n 个一维优化问题。

所有的共轭方向法都以该理论为基础，不同方法的区别在于产生共轭方向的方式不同。最著名的共轭方向法是使用导数的共轭梯度法和不使用导数的 Powell 共轭方向法，其中共轭梯度法是共轭思想和最速下降法相结合的产物。

5.2.2 基本的共轭梯度法

共轭梯度法(conjugate gradient method)首先是非常重要的求解大规模线性方程组的方法，在非线性偏微分方程的数值解中有非常重要的应用；另外，它也是优化技术的重要组成部分。一些重要优化方法子问题的求解经常要用到共轭梯度法，如确定牛顿方向、确定信赖域子问题的近似解和求解内点法中的法方程等。尽管与 5.3 节的拟牛顿法相比，共轭梯度法的有效性和可靠性都稍逊一筹，但共轭梯度法的独特优点是确定搜索方向时不需要矩阵运算，通常仅需存储 3~4 个 n 维向量，因此共轭梯度法是为数不多的可以求解大规模问题的方法之一。

对于二次函数(5.1.1)，该方法即是取精确步长的线搜索法，且

$$\mathbf{p}^{(0)} = -\mathbf{g}^{(0)} \quad (5.2.4)$$

对 $k \geq 0$, 由式(5.2.3)有 $\mathbf{g}^{(k+1)^\top} \mathbf{p}^{(i)} = 0, i = 0, 1, \dots, k$, 故 $\mathbf{g}^{(k+1)}$ 与 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k)}$ 线性无关. 这样, 可以采用 Gram-Schmidt 正交化过程来构造搜索方向 $\mathbf{p}^{(k+1)}$, 即令 $\mathbf{p}^{(k+1)}$ 是 $-\mathbf{g}^{(k+1)}$ 中与 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k)}$ 共轭的部分. 利用待定系数法, 即令

$$\mathbf{p}^{(k+1)} = -\mathbf{g}^{(k+1)} + \sum_{i=0}^k \beta_i^{(k+1)} \mathbf{p}^{(i)}$$

为了保证共轭性, 即 $\mathbf{p}^{(k+1)^\top} \mathbf{G} \mathbf{p}^{(i)} = 0, i = 0, 1, \dots, k$, 必有 $\beta_i^{(k+1)} = \frac{\mathbf{g}^{(k+1)^\top} \mathbf{G} \mathbf{p}^{(i)}}{\mathbf{p}^{(i)^\top} \mathbf{G} \mathbf{p}^{(i)}}$. 因为 $\mathbf{g}^{(k+1)^\top} \mathbf{p}^{(i)} = 0$

($i = 0, 1, \dots, k$) 蕴含着 $\mathbf{g}^{(k+1)^\top} \mathbf{g}^{(i)} = 0, i = 0, 1, \dots, k$. 再由 $\mathbf{g}^{(i+1)} - \mathbf{g}^{(i)} = \alpha_i \mathbf{G} \mathbf{p}^{(i)}$ 得

$$\beta_i^{(k+1)} = \frac{\mathbf{g}^{(k+1)^\top} \mathbf{G} \mathbf{p}^{(i)}}{\mathbf{p}^{(i)^\top} \mathbf{G} \mathbf{p}^{(i)}} = \frac{\mathbf{g}^{(k+1)^\top} (\mathbf{g}^{(i+1)} - \mathbf{g}^{(i)})}{\alpha_i \mathbf{p}^{(i)^\top} \mathbf{G} \mathbf{p}^{(i)}} = 0, \quad i < k$$

和

$$\beta_k^{(k+1)} = \frac{\mathbf{g}^{(k+1)^\top} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}{\mathbf{p}^{(k)^\top} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})} = \frac{\mathbf{g}^{(k+1)^\top} \mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)}} \quad (5.2.5)$$

将 $\beta_k^{(k+1)}$ 简记为 β_{k+1} . 这样, 对 $k \geq 1$, 共轭梯度法的搜索方向为

$$\mathbf{p}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_{k+1} \mathbf{p}^{(k)} \quad (5.2.6)$$

其中 β_{k+1} 由式(5.2.5)确定. 这时, 线搜索的精确步长公式(4.2.11)也可进一步简化为

$$\alpha_k = \frac{\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)}}{\mathbf{p}^{(k)^\top} \mathbf{G} \mathbf{p}^{(k)}} \quad (5.2.7)$$

例 5.2.1 (共轭梯度法) 用共轭梯度法解方程组 $\mathbf{G}\mathbf{x} = \mathbf{b}$, 其中

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

初始点 $\mathbf{x}^{(0)} = (0, 0, 0)^\top$. 方法的迭代数据见表 5.2.1.

表 5.2.1 共轭梯度法解例 5.2.1 的迭代数据

k	$\mathbf{x}^{(k)}$	$\mathbf{g}^{(k)}$	β_k	$\mathbf{p}^{(k)}$	α_k
0	$(0, 0, 0)^\top$	$(-1, -1, -1)^\top$		$(1, 1, 1)^\top$	$\frac{1}{2}$
1	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^\top$	$(-\frac{1}{2}, 0, \frac{1}{2})^\top$	$\frac{1}{6}$	$(\frac{2}{3}, \frac{1}{6}, -\frac{1}{3})^\top$	$\frac{3}{5}$
2	$(\frac{9}{10}, \frac{3}{5}, \frac{3}{10})^\top$	$(-\frac{1}{10}, \frac{1}{5}, -\frac{1}{10})^\top$	$\frac{3}{25}$	$(\frac{9}{50}, -\frac{9}{50}, \frac{3}{50})^\top$	$\frac{5}{9}$
3	$(1, \frac{1}{2}, \frac{1}{3})^\top$	$(0, 0, 0)^\top$			

极小化二次函数(5.1.1)的共轭梯度法的伪码如算法 5.2.1 所示. 该方法最大的优点是每步的计算量小. 由算法 5.2.1 可以看出, 它仅需要计算 1 个矩阵与向量的乘积 $\mathbf{G} \mathbf{p}^{(k)}$, 2 个内积 $\mathbf{p}^{(k)^\top} \mathbf{d}$ 和 $\mathbf{g}^{(k)^\top} \mathbf{g}^{(k)}$, 以及 3 个向量求和(即 $\mathbf{x}^{(k)}, \mathbf{g}^{(k)}$ 和 $\mathbf{p}^{(k)}$ 的更新); 同时, 需要存储 4 个 n 维向量. 此外, 这里最主要的计算量是 $\mathbf{G} \mathbf{p}^{(k)}$, 且只需要知道矩阵与向量的乘积的计算结果, 不用显式存储 \mathbf{G} , 利用这一特点可以求解大规模问题. 下面的例子说明了该算法的特点.

Algorithm 5.2.1 Conjugate gradient method for problem(5.1.1)

```

1: Given  $x^{(0)}$ , and compute  $\mathbf{g}^{(0)} = \mathbf{Gx}^{(0)} - \mathbf{b}$ ;
2: set  $\mathbf{p}^{(0)} = -\mathbf{g}^{(0)}$ ,  $k = 0$ ;
3: while  $\|\mathbf{g}^{(k)}\| > \epsilon$  do
4:   set  $\mathbf{d} = \mathbf{Gp}^{(k)}$ ;
5:   set  $\alpha_k = \frac{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}}{\mathbf{p}^{(k)T} \mathbf{d}}$ ;
6:   set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ ;
7:   set  $\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \alpha_k \mathbf{d}$ ;
8:   set  $\beta_{k+1} = \frac{\mathbf{g}^{(k+1)T} \mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}}$ ;
9:   set  $\mathbf{p}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_{k+1} \mathbf{p}^{(k)}$ ;
10:  set  $k = k + 1$ ;
11: end while
12: return  $\mathbf{x}^{(k)}$  as  $\mathbf{x}^*$ .

```

例 5.2.2 (稀疏矩阵与向量的乘积) 考虑稀疏矩阵

$$\mathbf{G} = \begin{bmatrix} 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

对于该矩阵,可以通过如下算法计算乘积 $\mathbf{y} = \mathbf{Gv}$.

```

Set  $n = 6$ ;
for  $i = 1, 2, \dots, n$ 
   $y_i = 4v_i$ ;
  if  $i > 1$  then  $y_i \leftarrow y_i + v_{i-1}$ 
  if  $i < n$  then  $y_i \leftarrow y_i + v_{i+1}$ 
end for

```

该算法需要大约 $3n$ 次算术运算,比传统的矩阵与向量的乘积所需的运算量 $2n^2$ 少得多,并且不用存储矩阵 \mathbf{G} .

下面将上述算法推导过程中所用到的有关事实罗列如下,借助于数学归纳法不难一一验证,留给读者自行推导.

定理 5.2.2 (性质) 设 m 是算法 5.2.1 中使 $\mathbf{g}^{(m)} \neq \mathbf{0}$ 的最大整数,则 $m \leq n$,且对所有 $k = 1, 2, \dots, m$,下列事实成立:

$$\mathbf{p}^{(k)T} \mathbf{Gp}^{(i)} = 0, \quad i = 0, 1, \dots, k-1 \quad (5.2.8a)$$

$$\mathbf{g}^{(k)T} \mathbf{g}^{(i)} = 0, \quad i = 0, 1, \dots, k-1 \quad (5.2.8b)$$

$$\mathbf{p}^{(k)T} \mathbf{g}^{(k)} = -\mathbf{g}^{(k)T} \mathbf{g}^{(k)} \quad (5.2.8c)$$

$$\text{span}\{\mathbf{g}^{(0)}, \mathbf{g}^{(1)}, \dots, \mathbf{g}^{(k)}\} = \text{span}\{\mathbf{g}^{(0)}, \mathbf{Gg}^{(0)}, \dots, \mathbf{G}^k \mathbf{g}^{(0)}\} \quad (5.2.8d)$$

$$\text{span}\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k)}\} = \text{span}\{\mathbf{g}^{(0)}, \mathbf{Gg}^{(0)}, \dots, \mathbf{G}^k \mathbf{g}^{(0)}\} \quad (5.2.8e)$$

上述性质中,式(5.2.8a)和式(5.2.8b)分别说明共轭梯度法所产生的搜索方向是共轭的,梯度是正交的;式(5.2.8c)说明每一步产生的搜索方向是下降的;式(5.2.8d)和式(5.2.8e)说明每个搜索方向和梯度皆包含于 $\mathbf{g}^{(0)}$ 的 k 阶 Krylov 子空间,即

$$K(\mathbf{g}^{(0)}; k) := \text{span}\{\mathbf{g}^{(0)}, \mathbf{Gg}^{(0)}, \dots, \mathbf{G}^k \mathbf{g}^{(0)}\}$$

这个概念在分析共轭梯度法的收敛速度方面有非常重要的应用. 共轭梯度法也是解系数矩阵对称正定的方程组的最基本的 Krylov 子空间法.

前面给出的算法 5.2.1 是极小化二次函数的. 很自然地有人会问,可否修改该方法使其适合极小化一般的凸函数,或者甚至更一般的非线性函数. Fletcher 和 Reeves 于 1964 年首次将共轭梯度法推广来极小化一般函数,具体即在算法 5.2.1 中作两个简单的改变. 首先,需要执行一维搜索确定出非线性函数 f 沿着 $\mathbf{p}^{(k)}$ 的近似极小点;其次,算法 5.2.1 中的梯度必须按非线性函数 f 的梯度来计算. 这些变化促成了极小化非线性函数的算法 5.2.2.

Algorithm 5.2.2 Fletcher-Reeves(FR) conjugate gradient method

- 1: Given $\mathbf{x}^{(0)}$; evaluate $\mathbf{g}^{(0)} = \nabla f(\mathbf{x}^{(0)})$;
- 2: set $\mathbf{p}^{(0)} = -\mathbf{g}^{(0)}$, $k=0$;
- 3: **while** $\|\mathbf{g}^{(k)}\| > \epsilon$ **do**
- 4: compute α_k and set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$;
- 5: evaluate $\mathbf{g}^{(k+1)}$;
- 6: set $\beta_{k+1}^{\text{FR}} = \frac{\mathbf{g}^{(k+1)T} \mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}}$;
- 7: set $\mathbf{p}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_{k+1}^{\text{FR}} \mathbf{p}^{(k)}$;
- 8: set $k=k+1$;
- 9: **end while**

如果 f 是二次函数,且 α_k 是精确极小点,该算法即退化为算法 5.2.1. 因为每一次迭代仅要求计算目标函数及其梯度的值,仅要求存储几个向量,不用执行矩阵运算,所以算法 5.2.2 非常适合大规模的非线性优化问题. 需要指出的是,在步骤 7,除非 α_k 满足某些条件,否则搜索方向 $\mathbf{p}^{(k)}$ 有可能不是下降方向. 为了完全确定算法 5.2.2,需要对一维搜索进行更确切的说明. 用梯度向量 $\mathbf{g}^{(k)}$ 与步骤 7 中的方向(用 k 代替 $k+1$)进行内积运算,得到

$$\mathbf{g}^{(k)T} \mathbf{p}^{(k)} = -\|\mathbf{g}^{(k)}\|^2 + \beta_k^{\text{FR}} \mathbf{g}^{(k)T} \mathbf{p}^{(k-1)} \quad (5.2.9)$$

如果线搜索是精确的,即 α_{k-1} 是 f 沿着方向 $\mathbf{p}^{(k-1)}$ 的局部极小点,则有

$$\mathbf{g}^{(k)T} \mathbf{p}^{(k-1)} = 0$$

在这种情况下,由式(5.2.9)有 $\mathbf{g}^{(k)T} \mathbf{p}^{(k)} < 0$,因此 $\mathbf{p}^{(k)}$ 的确是下降方向. 但是,如果线搜索不是精确的,在式(5.2.9)中,第二项可能控制第一项,从而有 $\mathbf{g}^{(k)T} \mathbf{p}^{(k)} > 0$,此时 $\mathbf{p}^{(k)}$ 实际上是上升方向. 幸运的是,要求步长 α_k 满足强 Wolfe 准则,且其中的参数满足 $0 < \rho < \sigma < 1/2$,可以避免这种情形. 需要注意的是,这里要求 $\sigma < 1/2$,而不是先前的 $\sigma < 1$. 可以证明式(4.3.6)蕴含着式(5.2.9)小于零,因此可以断定任何一个满足强 Wolfe 条件的一维搜索算法都能保证 FR 法产生的方向是下降方向.

FR 法存在许多变形,它们之间的区别主要在于参数 β_{k+1} 的选择. 一种最重要的变形是由 Polak、Ribi  re 和 Polyak 提出的,其中参数

$$\beta_{k+1}^{\text{PRP}} = \frac{\mathbf{g}^{(k+1)\top}(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}{\|\mathbf{g}^{(k)}\|^2} \quad (5.2.10)$$

用式(5.2.10)代替步骤6中的参数,得到PRP法.当 f 是强凸二次函数且线搜索是精确的时候,因为梯度是相互正交的,因此 $\beta_k^{\text{FR}} = \beta_k^{\text{PRP}}$,即二者是相等的.然而,当运用到一般的非线性函数,且用非精确线搜索时,两个算法的行为有显著的差别.数值实验表明,二者之中PRP法更稳健.需要指出的是,对于PRP法,强Wolfe准则(式(4.3.1)和式(4.3.6))不能保证 $\mathbf{p}^{(k)}$ 总是下降方向.为此将参数 β 修正为

$$\beta_{k+1}^+ = \max(\beta_{k+1}^{\text{PRP}}, 0) \quad (5.2.11)$$

称对应的算法是PRP⁺法,此时对强Wolfe准则进行简单修改,即可确保下降性质成立. β_{k+1} 还有许多其他选择. Hestenes-Stiefel公式

$$\beta_{k+1}^{\text{HS}} = \frac{\mathbf{g}^{(k+1)\top}(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}{\mathbf{p}^{(k)\top}(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}$$

产生一个收敛性质和实用性能两方面都类似于PRP法的算法.其他的还有Dixon公式

$$\beta_{k+1}^D = \frac{\mathbf{g}^{(k+1)\top}\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)\top}\mathbf{p}^{(k)}}$$

以及Dai-Yuan公式

$$\beta_{k+1}^{\text{DY}} = \frac{\mathbf{g}^{(k+1)\top}\mathbf{g}^{(k+1)}}{\mathbf{p}^{(k)\top}(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}$$

在目标是二次的且用精确线搜索的情况下,所有这些公式和FR公式 β_{k+1}^{FR} 是一致的.对一般非线性函数,通常PRP公式(5.2.10)更有效.

例5.2.3(共轭梯度法) 考虑

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \frac{1}{10}(\mathbf{x} - \mathbf{1})^\top \mathbf{D}(\mathbf{x} - \mathbf{1}) + \left(\mathbf{x}^\top \mathbf{x} - \frac{1}{4}\right)^2$$

其中 \mathbf{D} 是对角矩阵,对角线元素为 $1, 2, \dots, n$.梯度和Hessian阵分别为: $\mathbf{g}(\mathbf{x}) = \mathbf{D}(\mathbf{x} - \mathbf{1})/5 + 4(\mathbf{x}^\top \mathbf{x} - 1/4)\mathbf{x}$, $\mathbf{G}(\mathbf{x}) = \mathbf{D}/5 + 4(\mathbf{x}^\top \mathbf{x} - 1/4)\mathbf{I} + 8\mathbf{x}\mathbf{x}^\top$.可以利用公式

$$\mathbf{G}(\mathbf{x})\mathbf{v} = \mathbf{D}\mathbf{v}/5 + 4(\mathbf{x}^\top \mathbf{x} - 1/4)\mathbf{v} + 8(\mathbf{x}^\top \mathbf{v})\mathbf{x}$$

来计算共轭梯度法中的矩阵与向量的乘积,这仅需要 $O(n)$ 次运算,而传统的矩阵与向量的乘积运算需要 $O(n^2)$.这里由回溯Armijo线搜索确定步长,且在算法4.3.1中置 $\rho = 0.1$,缩减因子 $\gamma = 1/2$. $n=4$ 的完整结果见表5.2.2,其中 k 是迭代次数,ls记录的是计算梯度的次数.

共轭梯度法的实现与算法5.2.1有关.通常,在线搜索过程中要沿着搜索方向 $\mathbf{p}^{(k)}$ 进行二次或三次插值.这样做的优点是,当 f 是二次函数时,步长 α_k 即为精确一维极小点,因此共轭梯度法即退化成算法5.2.1.此外,共轭梯度法中经常利用的技术是在每 n 步迭代后,通过置 $\beta_k = 0$ (即取最速下降方向)重新开始(restart)迭代,其目的是擦除那些由于误差积累带来的不利信息.可以证明关于重新开始方法的理论结果,它是 n 步二阶收敛,即

$$\|\mathbf{x}^{(k+n)} - \mathbf{x}^*\| = O(\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2) \quad (5.2.12)$$

该结论并不难理解.首先观察一个特例,考虑在解的某邻域内是二次但在其他地方不是二次的函数.假定算法收敛到这个解,迭代最终将进入二次区域.在解附近某一点,算法将重新开始.从该点以后,它的行为将大致同算法5.2.1一样,特别地,重新开始后的 n 步之内必然有限终止.在算法5.2.1中,仅当它的初始搜索方向 $\mathbf{p}^{(0)}$ 是负梯度时,有限终止等性质才成立,所

表 5.2.2 FR 法求解例 5.2.3 的详细数据

k	ls	$\ g^{(k)}\ _\infty$	k	ls	$\ g^{(k)}\ _\infty$
0	1	2×10^1	13	33	6×10^{-5}
1	6	8×10^{-1}	14	35	5×10^{-5}
2	8	2×10^{-1}	15	37	2×10^{-5}
3	11	2×10^{-1}	16	38	2×10^{-5}
4	12	1×10^{-1}	17	41	8×10^{-6}
5	16	6×10^{-2}	18	43	7×10^{-6}
6	18	3×10^{-2}	19	46	2×10^{-6}
7	20	1×10^{-2}	20	48	9×10^{-7}
8	23	1×10^{-3}	21	50	5×10^{-7}
9	24	1×10^{-3}	22	52	1×10^{-7}
10	27	7×10^{-4}	23	54	1×10^{-7}
11	29	5×10^{-4}	24	57	7×10^{-8}
12	32	2×10^{-4}			

以重新开始策略很重要. 一般地, Taylor 展式蕴含着光滑的函数在解的附近近似于一个二次函数, 虽然不指望 n 步二次终止, 仍然可以期望较快地收敛.

尽管结论(5.2.12)理论上很有意义, 但与实际应用没有太大的关系, 因为非线性共轭梯度法仅被推荐来求解 n 较大的问题. 在这样的问题中, 经常在比 n 小得多的步即可找到一个近似解, 所以甚至可能不会出现重新开始. 因此, 实用的非线性共轭梯度法的重新开始策略并不是机械地直接累计迭代次数. 最流行的重新开始策略利用式(5.2.8b), 即当 f 是二次函数时, 梯度是相互正交的. 只要两个相邻的梯度远离正交, 就执行重新开始. 远离正交性可由测试

$$\frac{|\mathbf{g}^{(k)T} \mathbf{g}^{(k+1)}|}{\|\mathbf{g}^{(k)}\|^2} \geq \nu \quad (5.2.13)$$

来度量, 其中参数 ν 的典型值是 0.1. 此外, PRP⁺ 公式(5.2.11)自动包含重新开始策略, 因为只要 β_{k+1}^{PRP} 非正, $\mathbf{p}^{(k+1)}$ 自然将取最速下降方向. 然而, β_{k+1}^{PRP} 大多数时间是正的, 因此与式(5.2.13)相比, 这种重新开始策略很少执行重新开始.

表 5.2.3 列出了不带重新开始策略的 FR 法、PRP 法和 PRP⁺ 法计算不同算例时的数值表现. 对于这些测试, 强 Wolfe 准则(4.3.1)和(4.3.6)中的参数选为 $\rho = 10^{-4}$, $\sigma = 0.1$. 当满足

$$\|\mathbf{g}^{(k)}\|_\infty < 10^{-5} (1 + |f^{(k)}|)$$

时, 或者在迭代 10 000 次(最大迭代次数)后, 终止迭代(在表中后一种终止记为“*”). 表头为 mod 的最后一列表明 PRP⁺ 算法中需要调整式(5.2.11)以确保 $\beta_{k+1}^{\text{PRP}} \geq 0$ 的迭代次数. FR 法在问题 GENROS 上的结果表明, 当迭代远离解时, 选取的步长非常小, 这样目标函数的改进很小. it/f-g 表示需要的迭代次数/计算函数和梯度的次数.

虽然 PRP 法, 或者它的变形 PRP⁺, 并非总是比 FR 法更有效, 而且它的一个微小的缺点

表 5.2.3 部分共轭梯度法在测试问题集上的表现

问题	n	FR it/f-g	PRP it/f-g	PRP ⁺ it/f-g	mod
CALCVAR3	200	2 808/5 617	2 631/5 263	2 631/5 263	0
GENROS	500	*	1 068/2 151	1 067/2 149	1
XPOWSING	1 000	533/1 102	212/473	97/229	3
TRIDIAI	1 000	264/531	262/527	262/527	0
MSQRT1	1 000	422/849	113/231	113/231	0
XPOWELL	1 000	568/1 175	212/473	97/229	3
TRIGON	1 000	231/467	40/92	40/92	0

是还要求多存储一个向量,然而一般情况下,还是建议选择 PRP 法或 PRP⁺ 法.

5.2.3 收敛速度与预条件

在精确算术运算下,共轭梯度法具有二次终止性. 假定迭代了 $m (\leq n)$ 次,理论分析表明 m 是 \mathbf{G} 的不同特征值的个数. 此外,也可得到共轭梯度法的收敛速度依赖于 \mathbf{G} 的条件数,或者更一般地依赖于 \mathbf{G} 的特征值分布.

图 5.2.1 展示了共轭梯度法在这类问题上的表现. 图 5.2.1(a)中,一个问题有 5 个大的特征值,其余特征值均聚集在 0.95 和 1.05 之间,如图 5.2.2 所示;另一个问题的特征值是均匀分布的. 可以看到,对于特征值聚集的问题,在第 7 次迭代后精度就很高了;相反地,对于特征值随机分布的问题,收敛要慢一些,且更均匀些. 这里可以近似地认为矩阵 \mathbf{G} 仅有 6 个不同的特征值,即 5 个大的值和 1. 则期望在第 6 次迭代后误差是零. 因为 1 附近的特征值是稍微铺开的,因此误差不会变得很小,直到下一次迭代(即第 7 次迭代)可保证误差很小. 图 5.2.1(b)显示了共轭梯度法在维数 $n=14$ 的问题上的性能,该矩阵的特征值有 4 个群:即 140 和 120 的单个特征值、10 附近的特征值群,其余特征值聚集在 0.95 和 1.05 之间. 在 4 次迭代后,误差范数相当小. 在 6 次迭代后,解的精度就相当高了.

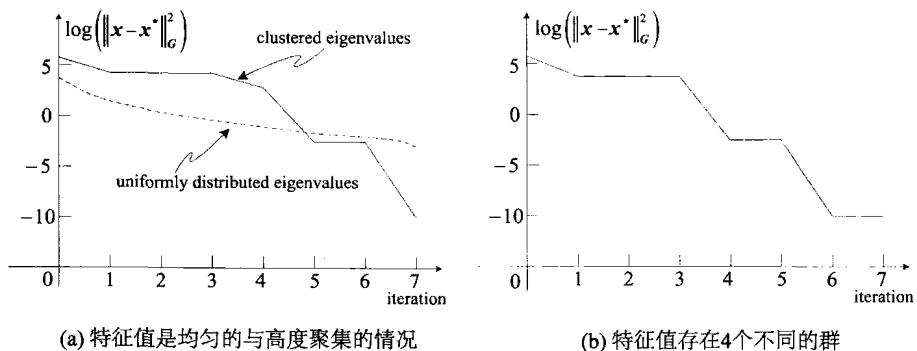


图 5.2.1 共轭梯度法的性能

为了加速算法,可以对变量进行变换,即通过非奇异矩阵 \mathbf{C} 将变量从 \mathbf{x} 变成 $\hat{\mathbf{x}}$,即

$$\hat{\mathbf{x}} = \mathbf{C}\mathbf{x} \quad (5.2.14)$$



图 5.2.2 特征值分布

这时,待极小化的二次函数(5.1.1)相应地变成

$$\hat{q}(\hat{x}) = \frac{1}{2} \hat{x}^T (\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}) \hat{x} - (\mathbf{C}^{-T} \mathbf{b})^T \hat{x} \quad (5.2.15)$$

选取线性变换矩阵使得 $\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}$ 的特征值分布较好,从而达到加速共轭梯度法的目的. 预条件 (precondition) 技术意味着用共轭梯度法极小化二次函数(5.2.15),或者等价地求解线性方程组

$$(\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}) \hat{x} = \mathbf{C}^{-T} \mathbf{b}$$

易见收敛速度依赖于矩阵 $\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}$ 的特征值,而不是矩阵 \mathbf{G} 的特征值. 因此,其目的是选择 \mathbf{C} 使得 $\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}$ 的条件数比原始矩阵 \mathbf{G} 的条件数小得多;也可以尝试选择 \mathbf{C} 使得 $\mathbf{C}^{-T} \mathbf{G} \mathbf{C}^{-1}$ 的特征值高度聚集,从而用少量的迭代即可找到质量较高的近似解.

整个过程中不必显式地执行变换(5.2.14). 先将算法 5.2.1 应用到变量为 \hat{x} 的函数(5.2.15)中,然后利用逆变换表示关于 x 的所有等式. 由此得到预条件共轭梯度法,即算法 5.2.3. 碰巧算法 5.2.3 没有显式利用 \mathbf{C} ,而是利用了对称正定矩阵 $\mathbf{M} := \mathbf{C}^T \mathbf{C}$,称 \mathbf{M} 为预条件子 (preconditioner).

Algorithm 5.2.3 Preconditioned Conjugate Gradient (PCG) method

```

1: Given  $x^{(0)}$ , preconditioner  $\mathbf{M}$  ;
2: set  $\mathbf{g}^{(0)} = \mathbf{G}x^{(0)} - \mathbf{b}$ ;
3: solve  $\mathbf{M}\mathbf{y}^{(0)} = \mathbf{g}^{(0)}$  for  $\mathbf{y}^{(0)}$  ;
4: set  $\mathbf{p}^{(0)} = -\mathbf{y}^{(0)}$ ,  $k=0$  ;
5: while  $\mathbf{g}^{(k)} \neq \mathbf{0}$  do
6:   set  $\mathbf{d} = \mathbf{G}\mathbf{p}^{(k)}$  ;
7:   set  $\alpha_k = \frac{\mathbf{g}^{(k)T} \mathbf{y}^{(k)}}{\mathbf{p}^{(k)T} \mathbf{d}}$  ;
8:   set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$  ;
9:   set  $\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \alpha_k \mathbf{d}$  ;
10:  solve  $\mathbf{M}\mathbf{y} = \mathbf{g}^{(k+1)}$  for  $\mathbf{y}^{(k+1)}$  ;
11:  set  $\beta_{k+1} = \frac{\mathbf{g}^{(k+1)T} \mathbf{y}^{(k+1)}}{\mathbf{g}^{(k)T} \mathbf{y}^{(k)}}$  ;
12:  set  $\mathbf{p}^{(k+1)} = -\mathbf{y}^{(k+1)} + \beta_{k+1} \mathbf{p}^{(k)}$  ;
13:  set  $k = k + 1$  ;
14: end while

```

实际上,预条件技术把解方程 $\mathbf{G}x = \mathbf{b}$ 转化成解方程 $\mathbf{M}^{-1} \mathbf{G}x = \mathbf{M}^{-1} \mathbf{b}$. 如果在算法 5.2.3 中置 $\mathbf{M} = \mathbf{I}$,即恢复为算法 5.2.1. 算法 5.2.1 的性质同样可以推广到算法 5.2.3. 特别地,梯度的正交性变成

$$\mathbf{g}^{(i)T} \mathbf{M}^{-1} \mathbf{g}^{(j)} = 0, \quad \forall i \neq j \quad (5.2.16)$$

对计算成本来说,主要区别是预条件共轭梯度法需要求解形如 $\mathbf{M}\mathbf{y} = \mathbf{g}$ 的线性方程组.

例 5.2.4 (预条件) 考虑线性方程组

$$\mathbf{G}\mathbf{x} = \begin{bmatrix} 2000 & 1000 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

系数矩阵的条件数 $\kappa(\mathbf{G}) \approx 1700$. 如果利用预条件子

$$\mathbf{M} = \begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix}$$

则变换后的方程组

$$\mathbf{M}^{-1}\mathbf{G}\mathbf{x} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.001 \\ 1 \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

变换后矩阵的条件数 $\kappa(\mathbf{M}^{-1}\mathbf{G}) = 3$, 与原始矩阵相比有相当大的提高. 易于验证原始方程组和变换后的方程组是同解的, 且 $\mathbf{x} \approx (-0.3227, 0.6663)^T$.

例 5.2.5 (预条件共轭梯度法) 考虑线性方程组 $\mathbf{G}\mathbf{x} = \mathbf{b}$, 其中 \mathbf{G} 是 15×15 的对角矩阵, 元素 $g_{ii} = i$, $\mathbf{b} = (1, \dots, 1)^T$. 利用对角线元素依次为 $1, 2, 3, 4, 5, 6, 7, 1, \dots, 1$ 的对角预条件矩阵 \mathbf{M} , 则 $\mathbf{M}^{-1}\mathbf{G}$ 有 9 个不同的特征值 $1, 8, 9, \dots, 15$. 从表 5.2.4 的前两列可以看到, 预条件共轭梯度法迭代 9 次就收敛了, 这与理论分析是一致的. 如果不用预条件, 则需要 15 次迭代, 这与所预期的相同. 详细结果见表 5.2.4 的后两列.

表 5.2.4 预条件共轭梯度法求解例 5.2.5 的详细数据

k	Preconditioned		Non-preconditioned	
	$\ \mathbf{g}^{(k)}\ $	$\ \mathbf{x}^{(k)} - \mathbf{x}^*\ $	$\ \mathbf{g}^{(k)}\ $	$\ \mathbf{x}^{(k)} - \mathbf{x}^*\ $
0	2×10^0	4×10^{-1}	2×10^0	4×10^{-1}
1	1×10^0	4×10^{-1}	9×10^{-1}	3×10^{-1}
2	1×10^0	2×10^{-1}	4×10^{-1}	2×10^{-1}
3	4×10^{-1}	4×10^{-2}	2×10^{-1}	1×10^{-1}
4	3×10^{-2}	3×10^{-3}	2×10^{-1}	9×10^{-2}
5	4×10^{-3}	3×10^{-4}	1×10^{-1}	5×10^{-2}
6	3×10^{-4}	3×10^{-5}	9×10^{-2}	2×10^{-2}
7	1×10^{-5}	2×10^{-6}	3×10^{-2}	7×10^{-3}
8	1×10^{-6}	8×10^{-8}	2×10^{-2}	3×10^{-3}
9	2×10^{-16}	8×10^{-17}	5×10^{-3}	1×10^{-3}
10			1×10^0	6×10^{-4}
11			9×10^{-4}	2×10^{-4}
12			4×10^{-4}	5×10^{-5}
13			1×10^{-4}	1×10^{-5}
14			6×10^{-6}	7×10^{-7}
15			7×10^{-18}	8×10^{-17}

通常不存在对所有类型的矩阵都最好的预条件策略, 需要具体问题具体分析. 总之, 选取预条件子 \mathbf{M} 的标准是有效、所需存储量小、求解 $\mathbf{M}\mathbf{y} = \mathbf{g}$ 容易, 并尽可能在这些目标中进行折衷. 目前已经有一些可供选择的设计有效的预条件子的方法, 具体细节本书不再涉及, 请读者参考应用数值线性代数中的相关内容^[6].

5.3 拟牛顿法

经过修正的确可以保证牛顿法大范围收敛,然而牛顿法要求用户必须给出二阶导数的计算公式,这通常是该方法的主要弊端. 尽管目前关于自动微分技术的研究部分避免了手工计算二阶导数的冗长乏味和在计算中引入误差的风险,但是当梯度不易得到(诸如函数值是某一计算的结果,或是由隐式计算得到的)时,可以利用与牛顿法密切相关的办法,即有限差分牛顿法(finite difference Newton method)和拟牛顿法(quasi-Newton method).

有限差分牛顿法即利用有限差分估计得到 $\mathbf{G}^{(k)}$. 具体地,确定沿坐标方向 \mathbf{e}_i 的增量 $h > 0$,用梯度向量的差分估计 $\mathbf{G}^{(k)}$,即首先得到

$$\mathbf{G}^{(k)} \mathbf{e}_i \approx \frac{\mathbf{g}(\mathbf{x}^{(k)} + h\mathbf{e}_i) - \mathbf{g}^{(k)}}{h} := \bar{\mathbf{G}}^{(k)} \mathbf{e}_i$$

再进行对称化处理,令 $\mathbf{B}^{(k)} = (\bar{\mathbf{G}}^{(k)} + \bar{\mathbf{G}}^{(k)\top})/2$,最后用 $\mathbf{B}^{(k)}$ 代替牛顿法中的 $\mathbf{G}^{(k)}$. 该方法有下列不足:矩阵 $\mathbf{B}^{(k)}$ 未必正定(需要修正技术),为了估计 $\mathbf{G}^{(k)}$ 需要计算 n 个梯度向量,每次迭代要解一个线性方程组,以及不易确定 h 的值(值若太大,所给的近似不够精确,而若太小,可能导致数值困难). 尽管如此,该方法还是有它的使用价值,特别是在大型稀疏问题中,计算差分的工作量可以进一步减小. 此外,有时也可利用上一次 $\mathbf{B}^{(k)}$ 的分解因子.

拟牛顿法可以避免上述不足. 拟牛顿法与执行线搜索的牛顿法格式一样,不同之处是用对称正定矩阵 $\mathbf{H}^{(k)}(\mathbf{B}^{(k)})$ 来近似 $\mathbf{G}^{(k)^{-1}}(\mathbf{G}^{(k)})$,且每次迭代仅需修正 $\mathbf{H}^{(k)}(\mathbf{B}^{(k)})$. 第 k 次迭代的基本结构如下:

- 置 $\mathbf{p}^{(k)} = -\mathbf{H}^{(k)} \mathbf{g}^{(k)}$ (解方程组 $\mathbf{B}^{(k)} \mathbf{p}^{(k)} = -\mathbf{g}^{(k)}$) 得到 $\mathbf{p}^{(k)}$;
- 沿 $\mathbf{p}^{(k)}$ 进行线搜索得到 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$;
- 校正 $\mathbf{H}^{(k)}(\mathbf{B}^{(k)})$ 得到 $\mathbf{H}^{(k+1)}(\mathbf{B}^{(k+1)})$.

初始矩阵 $\mathbf{H}^{(0)}(\mathbf{B}^{(0)})$ 可以是任意的正定矩阵,在没有更好估计的情况下,通常取 $\mathbf{H}^{(0)}(\mathbf{B}^{(0)})$ 为单位矩阵. 与牛顿法相比,拟牛顿法有如下优点:只需要一阶导数(牛顿法需要二阶导数); $\mathbf{H}^{(k)}$ 的正定性蕴含着下降性($\mathbf{G}^{(k)}$ 可能不定);每次迭代都需要 $O(n^2)$ 次乘法运算(牛顿法是 $O(n^3)$).

当用 $\mathbf{B}^{(k)}$ 近似 $\mathbf{G}^{(k)}$ 时,(c) 中更新的是 $\mathbf{B}^{(k)}$ 的 Cholesky 分解,这样(a)中解方程组的计算复杂度仍然是 $O(n^2)$. 与共轭梯度法相比,拟牛顿法不需要重新开始策略. 这里需要指出的是,有些拟牛顿法(如 SR1)并不能保证 $\mathbf{H}^{(k)}$ 是正定的,这可以用信赖域法来实现,详见第 6 章.

5.3.1 拟牛顿条件

假设已经通过迭代

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)} \quad (5.3.1)$$

产生了新的迭代点 $\mathbf{x}^{(k+1)}$,并希望构造新的二次模型

$$q^{(k+1)}(\mathbf{s}) = f^{(k+1)} + \mathbf{g}^{(k+1)\top} \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{B}^{(k+1)} \mathbf{s}$$

基于本次及前一次迭代所获取的信息, $\mathbf{B}^{(k+1)}$ 应该满足哪些条件呢? 一个合理的要求是: $q^{(k+1)}(\mathbf{s})$

的梯度应该与目标函数 f 在最近两次迭代 $\mathbf{x}^{(k)}$ 和 $\mathbf{x}^{(k+1)}$ 处的梯度相匹配。注意 $\nabla q^{(k+1)}(\mathbf{0})$ 正好是 $\mathbf{g}^{(k+1)}$ ，根据二次函数的性质(1.4.14)，即要求

$$\mathbf{B}^{(k+1)}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} \quad (5.3.2)$$

为了简化记号，定义向量

$$\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \alpha_k \mathbf{p}^{(k)}, \quad \mathbf{y}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$$

式(5.3.2)变成

$$\mathbf{B}^{(k+1)} \mathbf{s}^{(k)} = \mathbf{y}^{(k)} \quad (5.3.3)$$

有时称为拟牛顿条件 (quasi-Newton condition)，也称为拟牛顿方程或者割线方程。若令 $\mathbf{H}^{(k+1)} = \mathbf{B}^{(k+1)^{-1}}$ ，则得到互补形式的拟牛顿条件

$$\mathbf{H}^{(k+1)} \mathbf{y}^{(k)} = \mathbf{s}^{(k)} \quad (5.3.4)$$

给定偏移量 $\mathbf{s}^{(k)}$ 和梯度的变化量 $\mathbf{y}^{(k)}$ ，拟牛顿条件要求对称正定矩阵 $\mathbf{B}^{(k+1)}$ 将 $\mathbf{s}^{(k)}$ 映射到 $\mathbf{y}^{(k)}$ 。将式(5.3.3)两边同时左乘 $\mathbf{s}^{(k)^\top}$ ，易于看到仅当 $\mathbf{s}^{(k)}$ 和 $\mathbf{y}^{(k)}$ 满足曲率条件 (curvature condition)

$$\mathbf{s}^{(k)^\top} \mathbf{y}^{(k)} > 0 \quad (5.3.5)$$

时，这才是可能的。当 f 是强凸或者可微严格凸函数时，对任两点 $\mathbf{x}^{(k)}$ 和 $\mathbf{x}^{(k+1)}$ ，不等式(5.3.5)成立。然而，该条件对非凸函数并不总是成立(见习题 5.22)。故在一般情况下，要对确定的步长施加某种限制以强迫式(5.3.5)成立。事实上，如果要求步长满足 Wolfe 准则或者强 Wolfe 准则，就可保证式(5.3.5)成立。为了证实该断言，由 $\mathbf{s}^{(k)}$ 的定义和 Wolfe 准则中的式(4.3.5)，有 $\mathbf{g}^{(k+1)^\top} \mathbf{s}^{(k)} \geq \sigma \mathbf{g}^{(k)^\top} \mathbf{s}^{(k)}$ ，因此

$$\mathbf{y}^{(k)^\top} \mathbf{s}^{(k)} \geq (\sigma - 1) \mathbf{g}^{(k)^\top} \mathbf{s}^{(k)} = (\sigma - 1) \alpha_k \mathbf{g}^{(k)^\top} \mathbf{p}^{(k)}$$

因为 $\sigma < 1$ 且 $\mathbf{p}^{(k)}$ 是下降方向，所以不等式的右边是正的，从而曲率条件(5.3.5)成立。

5.3.2 DFP 法和 BFGS 法

当曲率条件满足时，拟牛顿方程(5.3.3)或(5.3.4)肯定有解。事实上，因为对称矩阵有 $n(n+1)/2$ 个自由度，拟牛顿条件仅代表 n 个条件，所以会有无穷多个矩阵满足拟牛顿条件。正定性要求施加了 n 个额外的条件——所有顺序主子式必须是正的，但是在满足这些条件的前提下，仍有很大的自由度。

先考虑确定 $\mathbf{H}^{(k+1)}$ 。在满足拟牛顿条件且是对称矩阵的基础上再施加额外的条件，比如要求 $\mathbf{H}^{(k+1)}$ 在某种意义上要靠近当前矩阵 $\mathbf{H}^{(k)}$ 。利用待定系数法，设

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + a \mathbf{u} \mathbf{u}^\top + b \mathbf{v} \mathbf{v}^\top \quad (5.3.6)$$

这里 a, b 为标量， \mathbf{u}, \mathbf{v} 为 n 维向量，这样的公式称为秩二校正。代入拟牛顿方程(5.3.4)有

$$\mathbf{H}^{(k+1)} \mathbf{y}^{(k)} = \mathbf{H}^{(k)} \mathbf{y}^{(k)} + a[\mathbf{u}^\top \mathbf{y}^{(k)}] \mathbf{u} + b[\mathbf{v}^\top \mathbf{y}^{(k)}] \mathbf{v} = \mathbf{s}^{(k)} \quad (5.3.7)$$

这里括号内的项是标量，而 $\mathbf{u}, \mathbf{v}, a, b$ 选择不唯一，比如可选

$$\mathbf{u} = \mathbf{s}^{(k)}, \quad \mathbf{v} = \mathbf{H}^{(k)} \mathbf{y}^{(k)}, \quad a[\mathbf{u}^\top \mathbf{y}^{(k)}] = 1, \quad b[\mathbf{v}^\top \mathbf{y}^{(k)}] = -1$$

代入式(5.3.7)，可得

$$a = \frac{1}{\mathbf{s}^{(k)^\top} \mathbf{y}^{(k)}}, \quad b = -\frac{1}{\mathbf{y}^{(k)^\top} \mathbf{H}^{(k)} \mathbf{y}^{(k)}}$$

将这些确定的参数代入式(5.3.6)，得 DFP 校正公式

$$\mathbf{H}_{\text{DFP}}^{(k+1)} = \mathbf{H}^{(k)} + \frac{\mathbf{s}^{(k)} \mathbf{s}^{(k)^\top}}{\mathbf{s}^{(k)^\top} \mathbf{y}^{(k)}} - \frac{\mathbf{H}^{(k)} \mathbf{y}^{(k)} \mathbf{y}^{(k)^\top} \mathbf{H}^{(k)}}{\mathbf{y}^{(k)^\top} \mathbf{H}^{(k)} \mathbf{y}^{(k)}} \quad (5.3.8)$$

该式本身体现了拟牛顿校正的基本思想：每次迭代不是从头开始计算迭代矩阵，而是进行简单的校正，这样将最新观测到的关于目标函数的信息和已有的嵌在当前 Hessian 阵近似中的信息结合起来。

对式(5.3.8)两边求逆，右边利用 Sherman-Morrison 公式（见习题 5.25），并注意 $\mathbf{H}^{(k)} = \mathbf{B}^{(k-1)}^{-1}$ ，读者可以推导出近似 Hessian 阵 $\mathbf{B}^{(k)}$ 的更新公式

$$\mathbf{B}_{\text{DFP}}^{(k+1)} = (\mathbf{I} - \gamma_k \mathbf{y}^{(k)} \mathbf{s}^{(k)} \mathbf{s}^{(k)} \mathbf{y}^{(k)} \mathbf{T}) \mathbf{B}^{(k)} (\mathbf{I} - \gamma_k \mathbf{s}^{(k)} \mathbf{y}^{(k)} \mathbf{y}^{(k)} \mathbf{T}) + \gamma_k \mathbf{y}^{(k)} \mathbf{y}^{(k)} \mathbf{T} \quad (5.3.9)$$

其中 $\gamma_k = \frac{1}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}}$ 。类似地，基于拟牛顿条件(5.3.3)进行如上的推导，可以得到 BFGS 校正公式

$$\mathbf{B}_{\text{BFGS}}^{(k+1)} = \mathbf{B}^{(k)} - \frac{\mathbf{B}^{(k)} \mathbf{s}^{(k)} \mathbf{s}^{(k)} \mathbf{T} \mathbf{B}^{(k)}}{\mathbf{s}^{(k)} \mathbf{T} \mathbf{B}^{(k)} \mathbf{s}^{(k)}} + \frac{\mathbf{y}^{(k)} \mathbf{y}^{(k)} \mathbf{T}}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}} \quad (5.3.10)$$

再由 Sherman-Morrison 公式，得到关于 $\mathbf{H}^{(k+1)}$ 的 BFGS 校正公式

$$\begin{aligned} \mathbf{H}_{\text{BFGS}}^{(k+1)} &= \mathbf{H}^{(k)} + \left(1 + \frac{\mathbf{y}^{(k)} \mathbf{T} \mathbf{H}^{(k)} \mathbf{y}^{(k)}}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}}\right) \frac{\mathbf{s}^{(k)} \mathbf{s}^{(k)} \mathbf{T}}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}} - \frac{\mathbf{H}^{(k)} \mathbf{y}^{(k)} \mathbf{s}^{(k)} \mathbf{T} + \mathbf{s}^{(k)} \mathbf{y}^{(k)} \mathbf{T} \mathbf{H}^{(k)}}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}} \\ &= (\mathbf{I} - \rho_k \mathbf{s}^{(k)} \mathbf{y}^{(k)} \mathbf{T}) \mathbf{H}^{(k)} (\mathbf{I} - \rho_k \mathbf{y}^{(k)} \mathbf{s}^{(k)} \mathbf{T}) + \rho_k \mathbf{s}^{(k)} \mathbf{s}^{(k)} \mathbf{T} \end{aligned} \quad (5.3.11)$$

其中 $\rho_k = \frac{1}{\mathbf{y}^{(k)} \mathbf{T} \mathbf{s}^{(k)}}$ 。容易看出，DFP 和 BFGS 校正公式呈互补关系，只要对式(5.3.8)进行简单对换，即 $\mathbf{H} \leftrightarrow \mathbf{B}$ 和 $\mathbf{s} \leftrightarrow \mathbf{y}$ ，便得到式(5.3.10)。

在定义一个完整的 BFGS 算法之前，还有一个尚未解决的问题是：如何选取初始近似 $\mathbf{H}^{(0)}$ ？不幸的是，不存在所有情况下都表现好的神奇公式。可以利用关于问题的特定信息，比如通过置它为由有限差商得到的 $\mathbf{x}^{(0)}$ 处近似 Hessian 阵的逆；也可以简单地置它为单位矩阵，或者一定程度上反映变量的尺度的数量矩阵。表 5.3.1 和表 5.3.2 给出了这两种方法求解例 5.2.1 的迭代数据。

表 5.3.1 DFP 法求解例 5.2.1 的迭代数据

k	$\mathbf{x}^{(k)}$	$\mathbf{g}^{(k)}$	$\mathbf{s}^{(k-1)}$	$\mathbf{y}^{(k-1)}$	$\mathbf{H}^{(k)}$	$\mathbf{p}^{(k)}$	α_k
0	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$\frac{1}{2}$
1	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{3}{2} \end{bmatrix}$	$\frac{1}{42} \begin{bmatrix} 46 & 1 & -2 \\ 1 & 37 & -11 \\ -2 & -11 & 22 \end{bmatrix}$	$\begin{bmatrix} \frac{4}{7} \\ \frac{1}{7} \\ -\frac{2}{7} \end{bmatrix}$	$\frac{7}{10}$
2	$\begin{bmatrix} \frac{9}{10} \\ \frac{3}{5} \\ \frac{3}{10} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{10} \\ \frac{1}{5} \\ -\frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{10} \\ -\frac{1}{5} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{5} \\ -\frac{3}{5} \end{bmatrix}$	$\frac{1}{190} \begin{bmatrix} 223 & 33 & 11 \\ -33 & 128 & -11 \\ 11 & -11 & 67 \end{bmatrix}$	$\begin{bmatrix} \frac{3}{19} \\ -\frac{3}{19} \\ \frac{1}{19} \end{bmatrix}$	$\frac{19}{30}$

续表 5.3.1

k	$\mathbf{x}^{(k)}$	$\mathbf{g}^{(k)}$	$\mathbf{s}^{(k-1)}$	$\mathbf{y}^{(k-1)}$	$\mathbf{H}^{(k)}$	$\mathbf{p}^{(k)}$	α_k
3	$\begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{10} \\ \frac{1}{30} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{5} \\ \frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$		

表 5.3.2 BFGS 法求解例 5.2.1 的迭代数据

k	$\mathbf{x}^{(k)}$	$\mathbf{g}^{(k)}$	$\mathbf{s}^{(k-1)}$	$\mathbf{y}^{(k-1)}$	$\mathbf{H}^{(k)}$	$\mathbf{p}^{(k)}$	α_k
0	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$\frac{1}{2}$
1	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{3}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{11}{9} & \frac{1}{18} & -\frac{1}{9} \\ \frac{1}{18} & \frac{8}{9} & -\frac{5}{18} \\ -\frac{1}{9} & -\frac{5}{18} & \frac{5}{9} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{3} \\ \frac{1}{6} \\ -\frac{1}{3} \end{bmatrix}$	$\frac{3}{5}$
2	$\begin{bmatrix} \frac{9}{10} \\ \frac{3}{5} \\ \frac{3}{10} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{10} \\ \frac{1}{5} \\ -\frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{10} \\ -\frac{1}{5} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{5} \\ -\frac{3}{5} \end{bmatrix}$	$\begin{bmatrix} \frac{31}{25} & -\frac{6}{25} & \frac{2}{25} \\ -\frac{6}{25} & \frac{37}{50} & -\frac{2}{25} \\ \frac{2}{25} & -\frac{2}{25} & \frac{9}{25} \end{bmatrix}$	$\begin{bmatrix} \frac{9}{50} \\ -\frac{9}{50} \\ \frac{3}{50} \end{bmatrix}$	$\frac{5}{9}$
3	$\begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{10} \\ \frac{1}{30} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{5} \\ \frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$		

BFGS 方法的伪码描述见算法 5.3.1.

Algorithm 5.3.1 BFGS method

- 1: Given starting point $\mathbf{x}^{(0)}$, convergence tolerance $\epsilon > 0$, positive definite inverse Hessian approximation $\mathbf{H}^{(0)}$;
 - 2: set $k = 0$;
 - 3: **while** $\|\mathbf{g}^{(k)}\| > \epsilon$ **do**
 - 4: compute search direction $\mathbf{p}^{(k)} = -\mathbf{H}^{(k)} \mathbf{g}^{(k)}$;
 - 5: set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ where α_k is computed from a line search procedure to satisfy the Wolfe test formula(4.3.1) and formula(4.3.4) or strong Wolfe test formula(4.3.1) and formula(4.3.6);
 - 6: set $\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$;
 - 7: compute $\mathbf{H}^{(k+1)}$ with formula(5.3.11);
 - 8: set $k = k + 1$;
 - 9: **end while**
-

算法执行一次迭代的费用是 $O(n^2)$ 次算术运算(加上计算函数和梯度的费用). 该算法是稳健的, 并且它的收敛速度是超线性的, 对于大多数实际问题, 超线性已经足够快了. 虽然牛顿法收敛得更快, 但它要求解线性方程组, 从而每一次迭代的费用要高一些. BFGS 法更重要的优点是它不用计算二阶导数.

也可以推导对矩阵 $\mathbf{B}^{(k)}$ 的操作, 而不是关于矩阵 $\mathbf{H}^{(k)}$ 的 BFGS 法. 表面上, 这种变形的自然实现对于无约束极小化是不划算的, 因为它要求求解系统 $\mathbf{B}^{(k)} \mathbf{p} = -\mathbf{g}^{(k)}$ 以得到搜索方向, 因此每步的计算费用增加到 $O(n^3)$. 但是注意, 解方程组的主要复杂度在 Cholesky 分解上, 因而可以每步不显式记录 $\mathbf{B}^{(k)}$, 而只记录它的 Cholesky 分解矩阵, 即每步不显式校正 $\mathbf{B}^{(k+1)}$, 而是直接校正 Cholesky 分解矩阵, 相关公式推导留给读者. 这样每步解方程组的复杂度也降到 $O(n^2)$. 选择这种方式解方程组的好处之一是, 可以避免 $\mathbf{H}^{(k+1)}$ 中特别大的数字所导致的数值误差.

5.3.3 DFP 法和 BFGS 法的性质

接下来讨论 DFP 和 BFGS 校正的性质. 首先在所有满足拟牛顿条件的矩阵中, $\mathbf{B}_{\text{DFP}}^{(k+1)}$ 在某种范数的意义下是最靠近 $\mathbf{B}^{(k)}$ 的, 具体而言, $\mathbf{B}_{\text{DFP}}^{(k+1)}$ 是

$$\begin{aligned} & \underset{\mathbf{B}}{\text{minimize}} \quad \|\mathbf{B} - \mathbf{B}^{(k)}\|_W \\ & \text{subject to } \mathbf{B} = \mathbf{B}^T, \quad \mathbf{B}\mathbf{s}^{(k)} = \mathbf{y}^{(k)} \end{aligned} \quad (5.3.12)$$

的唯一解, 其中 $\mathbf{s}^{(k)}$ 和 $\mathbf{y}^{(k)}$ 满足式(5.3.5), $\mathbf{B}^{(k)}$ 是对称正定的. 目标函数中的范数是加权 Frobenius 范数, 即

$$\|\mathbf{A}\|_W = \|\mathbf{W}^{1/2} \mathbf{A} \mathbf{W}^{1/2}\|_F \quad (5.3.13)$$

权重矩阵 \mathbf{W} 可以选为任一满足关系 $\mathbf{W}\mathbf{y}^{(k)} = \mathbf{s}^{(k)}$ 的矩阵, 比如 $\mathbf{W} = (\bar{\mathbf{G}}^{(k)})^{-1}$, 这里 $\bar{\mathbf{G}}^{(k)}$ 定义为

$$\bar{\mathbf{G}}^{(k)} = \int_0^1 \mathbf{G}(\mathbf{x}^{(k)} + \tau \mathbf{s}^{(k)}) d\tau \quad (5.3.14)$$

即平均 Hessian 阵. 由 Taylor 定理可以得到性质

$$\mathbf{y}^{(k)} = \bar{\mathbf{G}}^{(k)} \mathbf{s}^{(k)}$$

同样, BFGS 校正 $\mathbf{H}_{\text{BFGS}}^{(k+1)}$ 是

$$\begin{aligned} & \underset{\mathbf{H}}{\text{minimize}} \quad \|\mathbf{H} - \mathbf{H}^{(k)}\|_W \\ & \text{subject to } \mathbf{H} = \mathbf{H}^T, \quad \mathbf{H}\mathbf{y}^{(k)} = \mathbf{s}^{(k)} \end{aligned} \quad (5.3.15)$$

的唯一解. 这里范数也是加权 Frobenius 范数, 其中的权重矩阵 \mathbf{W} 是满足 $\mathbf{W}\mathbf{s}^{(k)} = \mathbf{y}^{(k)}$ 的任一矩阵, 比如可以假定 \mathbf{W} 是式(5.3.14)中的平均 Hessian 阵. 此外, BFGS 校正还是

$$\begin{aligned} & \underset{\mathbf{B}}{\text{minimize}} \quad \text{trace}(\mathbf{B}^{(k)^{-1/2}} \mathbf{B} \mathbf{B}^{(k)^{-1/2}}) - \det(\mathbf{B}^{(k)^{-1/2}} \mathbf{B} \mathbf{B}^{(k)^{-1/2}}) \\ & \text{subject to } \mathbf{B} = \mathbf{B}^T, \quad \mathbf{B}\mathbf{s}^{(k)} = \mathbf{y}^{(k)} \end{aligned}$$

的唯一解. 这些性质统称为极小化改变性质. 证明并不困难, 读者在学完约束优化的理论之后可以尝试给出证明.

注意, 前面推导出的 BFGS 更新公式并没有明确要求更新后的 Hessian 阵是正定的, 然而, 下面的结论表明只要 $\mathbf{H}^{(k)}$ 是正定的, 则由 DFP 和 BFGS 更新公式产生的 $\mathbf{H}^{(k+1)}$ 也是正定的.

定理 5.3.1 (正定性) 假设曲率条件 $y^{(k)^\top} s^{(k)} > 0$ 满足. 只要 $H^{(k)}$ 是正定的, 则由 DFP 更新公式(5.3.8)和 BFGS 更新公式(5.3.11)产生的 $H^{(k+1)}$ 也是正定的.

证明 首先, 因为 $y^{(k)^\top} s^{(k)}$ 是正的, 因此 BFGS 更新公式(5.3.11)和式(5.3.8)有定义. 对任一非零向量 z , 由更新公式(5.3.11)有

$$z^\top H_{\text{BFGS}}^{(k+1)} z = w^\top H^{(k)} w + \rho_k (z^\top s^{(k)})^2 \geq 0$$

其中 $w = z - \rho_k (z^\top s^{(k)}) y^{(k)}$. 仅当 $z^\top s^{(k)} = 0$ 时, 右边才可能为零, 但此时 $w = z \neq 0$, 这蕴含着第一项大于零. 因此, $H_{\text{BFGS}}^{(k+1)}$ 是正定的. 类似地, 由 DFP 的更新公式(5.3.9)知道 $B_{\text{DFP}}^{(k+1)}$ 是正定的, 从而 $H_{\text{DFP}}^{(k+1)}$ 是正定的. ■

由前面知道, DFP 和 BFGS 都是秩二校正公式, 进一步, 它们的加权组合定义出一族公式

$$B_{\phi_k}^{(k+1)} = \phi_k B_{\text{DFP}}^{(k+1)} + (1 - \phi_k) B_{\text{BFGS}}^{(k+1)}$$

其中 ϕ_k 为实数. 这类校正称为 **Broyden 族**. 定理 5.3.1 可以推广到所有 $\phi_k \geq 0$ 的 Broyden 族.

将具有精确线搜索的 BFGS 和 DFP 法应用于二次函数时, 有几个著名的性质. 下面的定理陈述这些性质中的一部分, 略去证明.

定理 5.3.2 (共轭性与遗传性) 假设将 BFGS 和 DFP 法应用于二次函数(5.1.1). 设 $x^{(0)}$ 是初始点, $H^{(0)}$ 是任一 n 阶对称正定矩阵. α_k 是精确步长. 则

(i) 拟牛顿条件对所有以前的搜索方向都成立, 即

$$H^{(k)} y^{(i)} = s^{(i)}, \quad i = k-1, \dots, 1, 0$$

(ii) 如果初始矩阵 $H^{(0)} = I$, 则搜索方向是共轭的, 即

$$p^{(i)^\top} G p^{(j)} = 0, \quad i \neq j$$

(iii) 至多迭代 n 步即收敛到解. 如果执行了 n 次迭代, 有 $H^{(n)} = G^{-1}$.

为了分析一般情况下的大范围收敛性和局部收敛性, 给出下列性质.

假定 5.3.1 (a) $f \in C^2$;

(b) 在 x^* 的邻域内, f 的 Hessian 阵 $G(x)$ 是 Lipschitz 连续的;

(c) 水平集 $L = \{x : f(x) \leq f(x_0)\}$ 是凸集, 且存在正常数 m 和 M 使得

$$m \|z\|^2 \leq z^\top G(x) z \leq M \|z\|^2, \quad \forall z \in \mathbb{R}^n, \quad x \in L$$

定理 5.3.3 (大范围收敛) 若假定 5.3.1 (a) 和 (c) 成立, 采用 Wolfe 线搜索的 BFGS 算法 5.3.1 产生的序列 $\{x^{(k)}\}$ 收敛到 f 的极小点 x^* .

进一步, 该大范围收敛性定理可以推广到所有 $\phi_k \in [0, 1)$ 的 Broyden 族, 但不包括 DFP 校正. DFP 校正法大范围收敛性的证明至今仍然是非线性规划领域的一个公开问题. 最后不加证明地给出 BFGS 法的局部超线性收敛定理.

定理 5.3.4 (超线性收敛) 若假定 5.3.1 成立, 且采用 Wolfe 线搜索的 BFGS 算法 5.3.1 产生的序列 $\{x^{(k)}\}$ 收敛到 f 的极小点 x^* , 再假定 $\sum_{k=0}^{\infty} \|x^{(k)} - x^*\| < \infty$, 那么序列 $\{x^{(k)}\}$ 超线性收敛到 x^* .

对于实际问题, 易于观察到 BFGS 法的超线性收敛速度. 下面比较最速下降法、BFGS 法和非精确牛顿法对 Rosenbrock 函数的最后几次迭代. 表 5.3.3 给出了 $\|x^{(k)} - x^*\|$ 的值. 在这 3 种方法中, 均要求步长满足 Wolfe 准则. 初始点 $x^{(0)} = (-1.2, 1)^\top$, 终止条件是梯度

表 5.3.3 3 种算法的数值比较

最速下降法	BFGS 法	牛顿法
1.827E-04	1.70E-03	3.48E-02
1.826E-04	1.17E-03	1.44E-02
1.824E-04	1.34E-04	1.82E-04
1.823E-04	1.01E-06	1.17E-08

范数小于 10^{-5} , 此时最速下降法要用 5 264 次迭代, 而 BFGS 法和牛顿法仅分别用 34 和 21 次迭代.

5.3.4 SR1 法

在 DFP 和 BFGS 校正公式中, 校正矩阵 $\mathbf{B}^{(k+1)} (\mathbf{H}^{(k+1)})$ 和 $\mathbf{B}^{(k)} (\mathbf{H}^{(k)})$ 仅相差一个秩二矩阵. 本小节介绍一个更简单的秩一校正. 对称秩一校正的一般形式可设为

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \sigma \mathbf{v} \mathbf{v}^T$$

其中 σ 是 1 或者 -1 , 且选取的 σ 和 \mathbf{v} 要使 $\mathbf{B}^{(k+1)}$ 满足拟牛顿条件(5.3.3). 将这种校正形式代入拟牛顿方程, 有

$$\mathbf{y}^{(k)} = \mathbf{B}^{(k)} \mathbf{s}^{(k)} + (\sigma \mathbf{v}^T \mathbf{s}^{(k)}) \mathbf{v} \quad (5.3.16)$$

因为括号中的项是标量, 可以推断 \mathbf{v} 必须是 $\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}$ 的倍数, 即存在标量 δ 使得 $\mathbf{v} = \delta(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})$. 将 \mathbf{v} 的这种表示代入式(5.3.16)得到

$$\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)} = \sigma \delta^2 [\mathbf{s}^{(k)}^T (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})] (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) \quad (5.3.17)$$

很显然, 当(且仅当)将参数 δ 和 σ 选为

$$\sigma = \text{sign}(\mathbf{s}^{(k)}^T (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})), \quad \delta = \pm |\mathbf{s}^{(k)}^T (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})|^{-1/2}$$

时, 方程(5.3.17)成立. 至此, 得到唯一的满足拟牛顿条件的对称秩一校正公式为

$$\mathbf{B}_{\text{SR1}}^{(k+1)} = \mathbf{B}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})^T}{(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})^T \mathbf{s}^{(k)}} \quad (5.3.18)$$

利用 Sherman-Morrison 公式(见习题 5.25), 得到对应的 Hessian 阵逆的近似的校正公式

$$\mathbf{H}_{\text{SR1}}^{(k+1)} = \mathbf{H}^{(k)} + \frac{(\mathbf{s}^{(k)} - \mathbf{H}^{(k)} \mathbf{y}^{(k)}) (\mathbf{s}^{(k)} - \mathbf{H}^{(k)} \mathbf{y}^{(k)})^T}{(\mathbf{s}^{(k)} - \mathbf{H}^{(k)} \mathbf{y}^{(k)})^T \mathbf{y}^{(k)}} \quad (5.3.19)$$

前面知道, 通过互相交换 \mathbf{B}, \mathbf{H} 以及 \mathbf{s}, \mathbf{y} , DFP 和 BFGS 呈互补关系, 从这个角度看, 这里的对称秩一校正公式是自互补的. 表 5.3.4 给出了利用 SR1 法求解例 5.2.1 的迭代数据.

表 5.3.4 SR1 法求解例 5.2.1 的迭代数据

k	$\mathbf{x}^{(k)}$	$\mathbf{g}^{(k)}$	$\mathbf{s}^{(k-1)}$	$\mathbf{y}^{(k-1)}$	$\mathbf{H}^{(k)}$	$\mathbf{p}^{(k)}$	α_k
0	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$\frac{1}{2}$
1	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{3}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{7}{8} & -\frac{1}{4} \\ 0 & -\frac{1}{4} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{8} \\ -\frac{1}{4} \end{bmatrix}$	$\frac{4}{5}$
	$\begin{bmatrix} \frac{9}{10} \\ \frac{3}{5} \\ \frac{3}{10} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{10} \\ \frac{1}{5} \\ -\frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{10} \\ -\frac{1}{5} \end{bmatrix}$	$\begin{bmatrix} \frac{2}{5} \\ \frac{1}{5} \\ -\frac{3}{5} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{10} \\ \frac{1}{30} \end{bmatrix}$	1
	$\begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{10} \\ \frac{1}{30} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} \\ -\frac{1}{5} \\ \frac{1}{10} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$		

从定义上看, SR1 校正的一个显著的缺点是式(5.3.18)或者式(5.3.19)中的分母可能为零. 事实上, 即使对二次函数, 也可能存在迭代步使得满足拟牛顿条件的秩一校正矩阵不存在. 这就有必要重新检查上面的推导. 就 $\mathbf{B}^{(k)}$ 而言(对 $\mathbf{H}^{(k)}$ 可进行类似讨论), 存在 3 种情况:

(a) 如果 $(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})^\top \mathbf{s}^{(k)} \neq 0$, 则上面的讨论说明, 存在一个唯一的秩一校正矩阵满足拟牛顿条件(5.3.3), 具体地由式(5.3.18)给出.

(b) 如果 $\mathbf{y}^{(k)} = \mathbf{B}^{(k)} \mathbf{s}^{(k)}$, 则唯一满足拟牛顿条件的矩阵是 $\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)}$.

(c) 如果 $\mathbf{y}^{(k)} \neq \mathbf{B}^{(k)} \mathbf{s}^{(k)}$ 且 $(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})^\top \mathbf{s}^{(k)} = 0$, 则式(5.3.17)说明不存在满足拟牛顿条件的对称秩一校正矩阵.

情况(c)直接导致数值不稳定和数值崩溃, 这表明秩一校正矩阵不能提供充分的自由度以开发出具有所有秩二校正矩阵所要求的理想性质的矩阵.

但是也不必悲观, 简单的保护措施足以阻止 SR1 校正的崩溃和数值不稳定, 比如一旦观察到 SR1 校正公式分母接近零, 就跳过校正. 具体地, 仅当

$$|\mathbf{s}^{(k)^\top} (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})| \geq r \|\mathbf{s}^{(k)}\| \|\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}\| \quad (5.3.20)$$

时, 应用校正式(5.3.18)(其中 $r \in (0, 1)$ 是很小的数, 比如 $r = 10^{-8}$). 如果式(5.3.20)不成立, 置 $\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)}$. SR1 法的大多数实现利用这种跳过原则, 效果往往较好.

读者可能会问: 为什么在这里倡导跳过原则, 而在 5.3.3 小节的 BFGS 情况中, 不鼓励这种策略呢? 两种情况截然不同. 条件 $\mathbf{s}^{(k)^\top} (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) \approx 0$ 只是偶尔发生, 且当它发生时, 跳过校正蕴含着 $\mathbf{s}^{(k)^\top} \bar{\mathbf{G}} \mathbf{s}^{(k)} \approx \mathbf{s}^{(k)^\top} \mathbf{B}^{(k)} \mathbf{s}^{(k)}$ (其中 $\bar{\mathbf{G}}$ 是上一步的平均 Hessian 阵), 这意味着 $\mathbf{B}^{(k)}$ 沿着 $\mathbf{s}^{(k)}$ 的曲率已经是正确的. 而对 BFGS 校正来说, 如果线搜索不施加 Wolfe 准则, 则其所要求的曲率条件 $\mathbf{s}^{(k)^\top} \mathbf{y}^{(k)} \geq 0$ 很容易失败, 因此可能会经常出现跳过 BFGS 校正的情况, 这会使 Hessian 阵近似的质量下降.

即使 SR1 校正公式分母远离 0, 若 $\mathbf{B}^{(k)}$ 是正定的, $\mathbf{B}^{(k+1)}$ 也不一定正定; 同样的事实对 $\mathbf{H}^{(k)}$ 也是成立的. 因此, 在非线性规划发展的较早年代, 这样的缺点使得线搜索一统天下, SR1 校正几乎无用武之地. 然而, 随着信赖域法的出现和发展, SR1 校正公式被重新起用并委以重任, 原因是 SR1 校正公式产生的矩阵是 Hessian 阵的一个非常好的近似——经常比 BFGS 近似更好, 它更能反映真正 Hessian 阵中的不定性. 在第 6 章中将看到, 信赖域法的二次模型并不需要 Hessian 阵是正定的.

SR1 校正的主要优点是它产生的 Hessian 阵近似的效果特别好(见表 5.3.4). 首先对二次函数来描述该性质. 二次函数的独特性在于 $\mathbf{y}^{(k)}$ 和 $\mathbf{s}^{(k)}$ 呈齐次线性关系, $\mathbf{s}^{(k)}$ 的长度, 即线搜索的步长选择, 不影响 SR1 校正公式(5.3.18)和公式(5.3.19), 故不妨假定步长均为 1, 即

$$\mathbf{p}^{(k)} = -\mathbf{H}^{(k)} \mathbf{g}^{(k)}, \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{p}^{(k)} \quad (5.3.21)$$

定理 5.3.5 考虑二次函数(5.1.1). 则对任一初始点 $\mathbf{x}^{(0)}$ 和任一对称的初始矩阵 $\mathbf{H}^{(0)}$, 倘若对所有 k 都有 $\mathbf{s}^{(k)} - \mathbf{H}^{(k)} \mathbf{y}^{(k)} \neq 0$, 则由 SR1 法(式(5.3.19)和式(5.3.21))产生的迭代 $\{\mathbf{x}^{(k)}\}$ 至多在 n 步收敛到极小点. 此外, 如果执行了 n 步, 且搜索方向 $\mathbf{p}^{(i)}$ 线性无关, 则 $\mathbf{H}^{(n)} = \mathbf{G}^{-1}$.

上述定理说明极小化二次函数时, 不管执行何种线搜索, 以前所有搜索方向都满足拟牛顿条件. 对于 BFGS 而言, 仅在精确线搜索的限制性假定下才能建立类似的结论. 对于一般的非线性函数, SR1 校正在某些条件下仍然产生好的 Hessian 阵的近似. 粗略地讲, 定理 5.3.6 中的术语“一致线性无关”意味着迭代步不会陷入维数比 n 小的子空间内. 在实践中该假设通常(但不总是)能得到满足.

定理 5.3.6 假设 $f \in C^2$, 且它的 Hessian 阵在点 x^* 的某个邻域内有界且 Lipschitz 连续. 设 $\{x^{(k)}\}$ 是任一使得 $x^{(k)} \rightarrow x^*$ 的序列. 此外假设不等式(5.3.20)对所有的 k 及某 $r \in (0, 1)$ 成立, 且步 $s^{(k)}$ 是一致线性无关的. 则由 SR1 校正公式产生的矩阵 $B^{(k)}$ 满足 $\lim_{k \rightarrow \infty} \|B^{(k)} - G^*\| = 0$.

5.4 最小二乘

例 1.1.3 中介绍过最小二乘问题. 在工程设计、金融分析、预测预报等领域存在大量以最小二乘形式出现的最优化问题. 这是一类特殊的无约束优化问题, 其基本形式为

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} f(x) = \frac{1}{2} \sum_{i=1}^m [r_i(x)]^2 = \frac{1}{2} \|r(x)\|_2^2 = \frac{1}{2} r(x)^T r(x) \quad (5.4.1)$$

称 $r_i(x) (i=1, 2, \dots, m)$ 为余量或者残量(residuals). 本质上在求解方程组

$$r_i(x) = 0, \quad i = 1, 2, \dots, m \quad (5.4.2)$$

特别地, 当上述方程组无解时, 最小二乘解是使残差平方和最小的解. 若 $r(x) = Ax - b$ 为线性函数, 则

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} f(x) = \frac{1}{2} \|Ax - b\|_2^2 = \frac{1}{2} x^T A^T A x - b^T A x + \frac{1}{2} b^T b \quad (5.4.3)$$

其中 A 为 $m \times n$ 阶矩阵, b 为 m 维向量. 不失一般性, 总可以假定 A 是列满秩的; 否则对变量 x 进行合理降维, 可等价成一个新的列满秩最小二乘问题. 称问题(5.4.3)为线性最小二乘(linear least-square), 其他的称为非线性最小二乘(nonlinear least-square).

5.4.1 线性最小二乘

线性最小二乘(5.4.3)是二次函数的极小化问题. 因为 Hessian 阵 $A^T A$ 是半正定的, 根据可微凸函数的最优化条件, x^* 是极小点当且仅当

$$\nabla f^* = A^T A x^* - A^T b = 0 \quad (5.4.4)$$

称该方程组为法方程或者正规方程(normal equations). A 列满秩表明 $A^T A$ 非奇异, 从而正定, 也即方程组(5.4.4)有唯一解

$$x^* = (A^T A)^{-1} A^T b \quad (5.4.5)$$

称这里的 $(A^T A)^{-1} A^T$ 是 A 的广义逆(general inverse), 记作 A^+ .

法方程(5.4.4)提供了求解线性最小二乘的方法. 但必须意识到, 如果直接分解 $A^T A$, 在数值计算中会损失很多精度. 设 $A^T A$ 存在误差 Δ , 导致 x^* 的误差为 δ , 不难直接验证

$$\frac{\|\delta\|}{\|x^* + \delta\|} \leq \kappa(A^T A) \frac{\|\Delta\|}{\|A^T A\|}$$

其中条件数 $\kappa(A^T A) = \sigma_1^2 / \sigma_n^2$, σ_1 和 σ_n 是 A 的最大和最小奇异值. 实践中这个界经常可以取到, 特别是当误差 Δ 是由舍入引起的随机误差, 且 $A^T A$ 中误差的量级是 ϵ 时, 计算的相对精度会被放大 $\kappa(A^T A)$ 倍.

现在介绍基于 QR 分解的间接法. 对列满秩矩阵 A 进行 QR 分解(见附录 A.7), 即

$$A = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

其中 \mathbf{Q} 为 $m \times m$ 阶正交矩阵, $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$, $\mathbf{Q}_2 \in \mathbb{R}^{m \times (m-n)}$; \mathbf{R} 为 $n \times n$ 阶上三角矩阵且非奇异。注意, 2-范数具有正交不变性, 即

$$\|\mathbf{Ax} - \mathbf{b}\|^2 = \|\mathbf{Q}^T(\mathbf{Ax} - \mathbf{b})\|^2 = \|\mathbf{Rx} - \mathbf{Q}_1^T \mathbf{b}\|^2 + \|\mathbf{Q}_2^T \mathbf{b}\|^2$$

于是, 最小二乘解迫使上式第一项为 0, 第二项即为最优值, 且解

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{Q}_1^T \mathbf{b}$$

因为 \mathbf{R} 和 \mathbf{A} 的奇异值一一对应, 从而 $\kappa(\mathbf{R}) = \kappa(\mathbf{A})$ 。相比第一种方法, 误差的影响要小得多。最坏情况下, $\kappa(\mathbf{A}^T \mathbf{A})$ 是 $\kappa(\mathbf{A})$ 的平方量级, 看 Gill 和 Murry 于 1991 年给出的一个简单例子:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 10^{-4} \end{bmatrix}, \quad \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 2 & 2 \\ 2 & 2 + 10^{-8} \end{bmatrix}$$

条件数分别为 $\kappa(\mathbf{A}) = 2.8 \times 10^4$ 和 $\kappa(\mathbf{A}^T \mathbf{A}) = 8.0 \times 10^8$ 。

正是由于可能出现坏条件数的问题(这是第一种方法的致命缺点), 实践中一般采用基于 QR 分解的间接法。当然直接法也并非一无是处, 由于系数矩阵 $\mathbf{A}^T \mathbf{A}$ 仅与 n 有关, Cholesky 分解与 m 无关, 因而直接法特别适用于 $m \gg n$ 的情形。

5.4.2 非线性最小二乘

回到问题(5.4.1), 首先易得 $f(\mathbf{x})$ 的梯度(见习题 1.4)

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \quad (5.4.6)$$

及 Hessian 阵

$$\mathbf{G}(\mathbf{x}) = \mathbf{A}(\mathbf{x})^T \mathbf{A}(\mathbf{x}) + \sum_{i=1}^m r_i(\mathbf{x}) \nabla^2 r_i(\mathbf{x}) \quad (5.4.7)$$

其中

$$\mathbf{A}(\mathbf{x})^T = [\nabla r_1(\mathbf{x}), \nabla r_2(\mathbf{x}), \dots, \nabla r_m(\mathbf{x})] = \nabla \mathbf{r}^T \quad (5.4.8)$$

是 $n \times m$ 阶矩阵, 它的第 i 列为 $\nabla r_i(\mathbf{x})$, 即 $a_{ij} = \frac{\partial r_i}{\partial x_j}$ 。基于式(5.4.7), 牛顿法需要所有函数 $r_i(\mathbf{x}) (i=1, 2, \dots, m)$ 的 Hessian 阵的表达式, 这在实践中通常具有局限性。

一个非常有趣的现象是: 当各分量 $r_i(\mathbf{x})$ 都非常小时(小残量问题, 即 $r_i(\mathbf{x}) \approx 0$), 或者 $r_i(\mathbf{x})$ 非线性程度较小时(即 $\nabla^2 r_i(\mathbf{x}) \approx 0$), 均可以忽略式(5.4.7)中最后一项, 于是得到 $\mathbf{G}(\mathbf{x})$ 的一个好近似, 即

$$\mathbf{G}(\mathbf{x}) \approx \mathbf{A}(\mathbf{x})^T \mathbf{A}(\mathbf{x}) \quad (5.4.9)$$

用这种方式考虑问题(5.4.1)的结构的重要特色是仅须确定一阶导数向量 $\mathbf{g}(\mathbf{x})$ 所需的信息(\mathbf{r} 和 \mathbf{A}), 即可得到 Hessian 阵 $\mathbf{G}(\mathbf{x})$ 的近似。

从统计角度可以解释式(5.4.9)。给定 m 个数据 d_1, d_2, \dots, d_m , 即关于某独立变量 t 在 t_1, t_2, \dots, t_m 时刻的抽样值。一个理想做法是用有 n 个可调参数 \mathbf{x} 的函数 $\Phi(t; \mathbf{x})$ 来拟合数据, 并选取最优参数使函数与数据实现最佳吻合, 这里残量

$$r_i(\mathbf{x}) = \Phi(t_i; \mathbf{x}) - d_i, \quad i = 1, 2, \dots, m \quad (5.4.10)$$

在数据拟合中, 如果经验模型是线性的, 则 $\Phi(t; \mathbf{x})$ 可以表示为

$$\Phi(t; \mathbf{x}) = \sum_{j=1}^n x_j \psi_j(t)$$

通常假设观测值 d_i 满足

$$\mathbf{d} = \mathbf{Ax}^* + \mathbf{e}$$

其中 $a_{ij} = \frac{\partial r_i}{\partial x_j} = \psi_j(t_i)$, \mathbf{x}^* 是真实解, \mathbf{e} 是误差向量. 通常假定分量 e_i 是满足均值为 0、方差为常数 σ^2 的独立正态分布, 则式(5.4.2)和式(5.4.10)的最小二乘解 $\hat{\mathbf{x}}$ 是一极大似然解, 且期望值 $E(\hat{\mathbf{x}}) = \mathbf{x}^*$. 式(5.4.9)的逆是协方差矩阵 \mathbf{V} 的倍数, 即可以证明

$$\mathbf{V} = E[(\hat{\mathbf{x}} - \mathbf{x}^*)(\hat{\mathbf{x}} - \mathbf{x}^*)^T] = (\mathbf{A}(\mathbf{x}^*)^T \mathbf{A}(\mathbf{x}^*))^{-1} \sigma^2$$

\mathbf{V} 的对角线元素是最大似然解的分量 \hat{x}_i 的方差, 非对角线元素是 \hat{x}_i 和 \hat{x}_j 的协方差. 可以证明 $E(\hat{f}) = (m-n)\sigma^2$, 因此得到 σ^2 的一个估计 $\hat{f}/(m-n)$, 其中 \hat{f} 是通过求解问题(5.4.1)得到的最大似然估计的平方和. 利用该事实及式(5.4.9), 便可以确定 \mathbf{V} , 从而给出关于最小二乘解的分布的非常重要的统计信息.

下面讨论如何利用 Hessian 阵的估计式(5.4.9)来求解问题(5.4.1). 当用式(5.4.9)来近似 $\mathbf{G}^{(k)}$ 时, 基本牛顿法也就变成高斯-牛顿法(Gauss-Newton method, GN 法). 利用导数表达式(5.4.6)和式(5.4.9), GN 法的第 k 次迭代是

(a) 求解

$$\mathbf{A}^{(k)T} \mathbf{A}^{(k)} \mathbf{s} = -\mathbf{A}^{(k)T} \mathbf{r}^{(k)} \quad (5.4.11)$$

得到 $\mathbf{s}^{(k)}$;

(b) 置 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.

方程(5.4.11)本质上正是线性最小二乘的法方程. 事实上, 利用一阶 Taylor 近似, 有 $\mathbf{r}(\mathbf{x}) \approx \mathbf{r}^{(k)} + \mathbf{A}^{(k)}(\mathbf{x} - \mathbf{x}^{(k)})$, 这样, GN 法第 k 次迭代产生的新点 $\mathbf{x}^{(k+1)}$ 恰为线性最小二乘问题

$$\min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{r}^{(k)} + \mathbf{A}^{(k)}(\mathbf{x} - \mathbf{x}^{(k)})\|^2$$

的解. 下面的例子说明, 如果 $\mathbf{r}(\mathbf{x})$ 的非线性程度较低, 则该法的确很好.

例 5.4.1 (GN 法, Powell) 考虑

$$\begin{aligned} r_1(x) &= x + 1 \\ r_2(x) &= \lambda x^2 + x - 1 \end{aligned}$$

此时 $m=2, n=1, x^* = 0$. 取 $\lambda = 0.1$, 迭代过程见表 5.4.1.

表 5.4.1 GN 法的简单例子

k	0	1	2	3	4	5
$x^{(k)}$	1	0.131 148	0.013 635	0.001 369	0.000 137	0.000 014

记 $\mathbf{h}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$. 该例表明 GN 法的收敛速度是线性的, 其中误差满足 $h^{(k+1)} \approx 0.1h^{(k)}$. 事实上可以更精确地刻画误差的行为. 如果 $\mathbf{B}^{(k)} = \mathbf{A}^{(k)T} \mathbf{A}^{(k)}$ 是 $\mathbf{G}^{(k)}$ 的近似, 则类似于定理 5.1.5, 可以证明

$$\|\mathbf{h}^{(k+1)}\|_2 \leq 2 \left\| \mathbf{B}^{*-1} \left(\sum_{i=1}^m r_i^* \nabla^2 r_i^* \right) \right\|_2 \cdot \|\mathbf{h}^{(k)}\|_2 + O(\|\mathbf{h}^{(k)}\|_2^2) \quad (5.4.12)$$

如果序列 $\{\mathbf{h}^{(k)}\}$ 收敛, 则式(5.4.12)表明其收敛速度不会比线性差. 当条件

$$\sum_{i=1}^m r_i^* \nabla^2 r_i^* = \mathbf{0}$$

成立时,式(5.4.12)表明收敛速度是二阶的;且如同定理5.1.5,如果某次迭代充分接近 \mathbf{x}^* ,则有 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$.对于数据拟合问题,更普遍的是矩阵 $\sum_i r_i^* \nabla^2 r_i^*$ 非零,此时最坏情形下收敛速度不会比线性的好很多.如果矩阵 $\sum_i r_i^* \nabla^2 r_i^*$ 充分大(使得 $\mathbf{B}^{*-1} \sum_i r_i^* \nabla^2 r_i^*$ 有绝对值大于1的特征值),无论初始点 $\mathbf{x}^{(0)}$ 离 \mathbf{x}^* 多近,迭代都有可能不收敛.用例5.4.1来说明该事实.首先可以证明

$$\mathbf{x}^{(k+1)} = \lambda \mathbf{x}^{(k)} + O((\mathbf{x}^{(k)})^2)$$

故当 $|\lambda| > 1$ 时,基本高斯-牛顿法不能收敛到 \mathbf{x}^* .这样仅当 $\mathbf{x}^{(0)}$ 接近 \mathbf{x}^* 且矩阵 $\sum_i r_i^* \nabla^2 r_i^*$ 较小(r_i^* 小或者 $r_i(\mathbf{x})$ 的非线性程度低),式(5.4.11)才有价值.

鉴于该事实,结合线搜索的GN法是特别理想的.此时,通过解方程组(5.4.11)得到搜索方向 $\mathbf{p}^{(k)}$,并利用4.3节中的线搜索策略确定步长 α_k .在实践中,带线搜索的GN法大多数情况下是有效和实用的.其一个明显好处是矩阵 $\mathbf{A}(\mathbf{x})^T \mathbf{A}(\mathbf{x})$ 是半正定的,且通常是正定的,因此不会出现不定的 $\mathbf{G}^{(k)}$.事实上如果条件数 $\kappa(\mathbf{A}(\mathbf{x})^T \mathbf{A}(\mathbf{x}))$ 距离零一致有界,则由定理4.3.4可以证明收敛性.因为 $\mathbf{A}(\mathbf{x})$ 通常是上方有界的,上述假定本质上要求 $\mathbf{A}(\mathbf{x})$ 在极限时不能秩亏.

不幸的是,有可能出现极限时秩亏的情况.比如 $\mathbf{A}(\mathbf{x}^*)$ 秩亏,就可能导致 $\mathbf{r}^* \neq \mathbf{0}$,但是 $\mathbf{g}^* = \mathbf{0}$.结果是在距解 \mathbf{x}^* 还有一段距离时, $\mathbf{p}^{(k)}$ 就开始在数值上与 $\mathbf{g}^{(k)}$ 正交,从而一维搜索得不到进展,这样会得到一个精度较差的解.因此,带一维搜索的GN法还需要进一步修正.

Levenberg和Marquardt分别于1944年和1963年提出了著名的修正.因为

$$\mathbf{A}(\mathbf{x})^T \mathbf{A}(\mathbf{x}) + \lambda \mathbf{I}$$

对 $\lambda > 0$ 一定正定,以此来修正 $\mathbf{s}^{(k)}$,即通过求解线性方程组

$$(\mathbf{A}^{(k)T} \mathbf{A}^{(k)} + \lambda_k \mathbf{I}) \mathbf{s} = -\mathbf{A}^{(k)T} \mathbf{r}^{(k)}, \quad \lambda_k \geq 0 \quad (5.4.13)$$

得到修正量 $\mathbf{s}^{(k)}$.这里 λ_k 为控制参数,它间接确定 $\mathbf{s}^{(k)}$ 的长度.另外,关于 λ_k 的更新设计上有一些启发式的经典技巧.称这种方法为Levenberg-Marquardt法(LM法),伪码见算法5.4.1.

用于非线性最小二乘问题的LM法是稳健的,实际表现也很好,然而如式(5.4.13)表明,搜索方向将会偏向最速下降方向,这可能会影响收敛速度,即缺乏二阶收敛速度.这样,尽管LM法通常被认为是求解非线性最小二乘问题的最好方法,但也并不是完全满意的.这激发了一些后继性研究.牛顿方向 $\mathbf{s} = -\mathbf{G}^{-1} \mathbf{g}$ 某种意义上总是最出色的,各种校正均是提高矩阵 $\mathbf{A}^T \mathbf{A}$ 逼近 \mathbf{G} 的程度.当 \mathbf{A} 接近秩亏时,LM法给 $\mathbf{A}^T \mathbf{A}$ 的所有特征值增加了 λ ,故不可能任意接近 \mathbf{G} 的特征值.Gill和Murray于1976年尝试在相应子空间中进行有限差分估计来提高由 $\mathbf{A}^T \mathbf{A}$ 给出的曲率估计.Brown和Dennis于1971年用拟牛顿更新来近似 $\nabla^2 r_i$,Dennis、Gay和Welsch描述了针对 $\sum_i r_i \nabla^2 r_i$ 的一种更新机制.

此外,LM法也可以推广来求解一般的极小化问题.具体地,即在LM法中将 $\mathbf{A}^{(k)T} \mathbf{A}^{(k)}$ 换成 $\mathbf{G}^{(k)}$,并在算法5.4.1的步骤4中添加如下操作:只要 $\mathbf{G}^{(k)} + \lambda_k \mathbf{I}$ 不正定,置 $\lambda_k = 4\lambda_k$ 并重复.值得一提的是,利用 $\mathbf{H}^{(0)} = (\mathbf{A}^{(0)T} \mathbf{A}^{(0)})^{-1}$ 的BFGS法可以求解任一类LM法不能求解的问题.

Algorithm 5.4.1 Levenberg-Marquardt method

```

1: Given  $\mathbf{x}^{(0)}$  and  $\lambda_0$  ;
2: repeat
3:   calculate  $\mathbf{r}^{(k)}$  and  $\mathbf{A}^{(k)}$  ;
4:   factorize  $\mathbf{A}^{(k)T} \mathbf{A}^{(k)} + \lambda_k \mathbf{I}$  ;
5:   solve the system(5.4.13) for  $\mathbf{s}^{(k)}$  ;
6:   evaluate  $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$  and hence  $\rho_k = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{f(\mathbf{x}^{(k)}) - \frac{1}{2} \|\mathbf{r}^{(k)} + \mathbf{A}^{(k)} \mathbf{s}^{(k)}\|_2^2}$  ;
7:   if  $\rho_k < 0.25$  then
8:     set  $\lambda_{k+1} = 2\lambda_k$  ;
9:   end if
10:  if  $\rho_k > 0.75$  then
11:    set  $\lambda_{k+1} = \lambda_k / 2$  ;
12:  else
13:    set  $\lambda_{k+1} = \lambda_k$  ;
14:  end if
15:  if  $\rho_k \leq 0$  then
16:    set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$  ;
17:  else
18:    set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$  ;
19:  end if
20: until  $\|\mathbf{r}^{(k)}\|$  is sufficiently small.

```

5.5 评注与参考

由定理 5.1.3, 我们知道即使对于二次函数, 基于精确线搜索的最速下降法也仅是线性收敛. 而且当 Hessian 阵的条件数很大时, 算法的收敛速度相当慢. Barzilai 和 Borwein 于 1988 年提出一个两点步长梯度法^[27], 其基本思想是利用迭代的当前点和前一点的信息来确定步长因子. 他们把迭代公式 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{g}^{(k)}$ 看成是 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{D}^{(k)} \mathbf{g}^{(k)}$, 其中 $\mathbf{D}^{(k)} = \alpha_k \mathbf{I}$. 为了使矩阵 $\mathbf{D}^{(k)}$ 具有“拟牛顿”性质, 选取 α_k 求解问题

$$\min_a \|\mathbf{s}_{k-1} - \alpha \mathbf{y}_{k-1}\|_2 \quad (5.5.1)$$

或者

$$\min_a \|\mathbf{s}_{k-1} / \alpha - \mathbf{y}_{k-1}\|_2 \quad (5.5.2)$$

分别求解问题(5.5.1)和(5.5.2), 得

$$\alpha_k = \mathbf{s}_{k-1}^T \mathbf{y}_{k-1} / \|\mathbf{y}_{k-1}\|_2^2 \quad (5.5.3)$$

和

$$\alpha_k = \|\mathbf{s}_{k-1}\|_2^2 / \mathbf{s}_{k-1}^T \mathbf{y}_{k-1} \quad (5.5.4)$$

对于两个变量的二次函数,若由精确线搜索确定初始步长,以后利用这两个中的任一个,可以证明算法是超线性收敛的。需要注意的是,该方法不是下降法,所以对非二次函数要加以适当修正才可应用^[28]。这里需要指出的是,该方法的提出掀起了研究梯度法的热潮。

Hestenes 和 Stiefel 于 1952 年提出了共轭梯度法,当时是作为求系数矩阵对称正定方程组的迭代法。直到若干年后,该法才逐渐被认为是“用比 n 少得多的迭代即可给出系统较好近似解”的迭代法。下面讨论共轭梯度法在牛顿法中的应用。在牛顿法中,当 $\mathbf{G}^{(k)}$ 正定时,为了确定搜索方向 $\mathbf{p}^{(k)} = -\mathbf{G}^{(k)^{-1}} \mathbf{g}^{(k)}$, 等价于极小化二次函数

$$q^{(k)}(\mathbf{s}) = f^{(k)} + \mathbf{s}^T \mathbf{g}^{(k)} + \frac{1}{2} \mathbf{s}^T \mathbf{G}^{(k)} \mathbf{s} \quad (5.5.5)$$

对于大规模问题,通常找到二次函数式(5.5.5)的某个近似极小点即可。这样,通常从 $\mathbf{s} = \mathbf{0}$ 开始,用共轭梯度法极小化该函数。在实践中,可以将上面的共轭梯度法看成是提供了一种在最速下降方向(当 $i=1$ 时停止)和牛顿方向(当 $i=n$ 时停止)之间的折衷。因此,称在线搜索(或者信赖域)框架下使用这样一个缩短的(cretailed)共轭梯度步的方法为截断(truncated)牛顿法。它通常将 $\mathbf{r}^{(i)} = \mathbf{G}^{(k)} \mathbf{s}^{(i)} + \mathbf{g}^{(k)}$ (这里 $\mathbf{s}^{(i)}$ 是采用共轭梯度法极小化函数(5.5.5)时,第 i 次迭代产生的点)相对于 $\mathbf{g}^{(k)}$ 的尺寸作为迭代终止准则,一种特别流行的准则是当

$$\|\mathbf{r}^{(i)}\| \leq \min(\|\mathbf{g}^{(k)}\|^\omega, \eta) \|\mathbf{g}^{(k)}\| \quad (5.5.6)$$

时,终止共轭梯度法,其中 η 和 $\omega \in (0, 1)$ 是参数。参数 η 的典型取值是 0.5。选取不同的参数 ω 即可得到不同的渐近收敛速度,从而得到一种在每步的计算量和渐近收敛速度之间的一种显式折衷。例如选 $\omega=0, 1/2$ 和 1 分别可以得到线性收敛、超线性收敛和二次收敛。完成习题 5.14 后,读者对截断牛顿法的理解会更深刻。可以证明上述截断牛顿法的一次迭代中,共轭梯度法产生的任一点 $\mathbf{s}^{(i)}$ 均是原目标函数 f 在 $\mathbf{x}^{(k)}$ 处的下降方向,即对 $i=1, 2, \dots, n$, 有

$$\mathbf{g}^{(k)^T} \mathbf{s}^{(i)} \leq \mathbf{g}^{(k)^T} \mathbf{s}^{(i-1)} < 0$$

第一个拟牛顿法由美国 Argonne 国家实验室物理学家 Davidon 于 1959 年提出,这是非线性规划发展史上一次伟大的革新,而后由 Fletcher 和 Powell 加以提炼和分析,论证了其卓越的有效性,这就是 DFP 法。Davidon 的原创性文章一直未发表,直到 1991 年在 *SIAM Journal on Optimization* 上第一卷作为首篇文章发表。目前最成功的拟牛顿法是 BFGS 法,这是由来自英美两国的 4 位科学家 Broyden、Fletcher、Goldfarb 和 Shanno 各自独立提出来的。它与 DFP 法呈互补关系。

Gauss 被认为是最小二乘分析的奠基人,时年 18 岁,但是直到 1809 年才发表相关成果。而法国数学家 Legendre 1805 年即被认为是正式发表“最小二乘法”的第一人。二人曾为谁最早创立最小二乘法原理发生争执。美国科学家 Adrain 于 1808 年也独立提出了该方法。最小二乘法因早期在天文学中的成功应用而名声大震。1801 年 2 月 1 日,意大利天文学家 Piazzi 发现了谷神星,经过 40 天的跟踪观测后,由于谷神星运行至太阳背后,使得 Piazzi 失去了谷神星的位置。随后全世界的科学家利用 Piazzi 的观测数据开始寻找谷神星,但是根据大多数人计算的结果来寻找谷神星都没有结果。时年 24 岁的高斯也加入了谷神星的轨道计算,后来匈牙利天文学家 Franz Xaver von Zach 根据高斯计算出来的轨道重新发现了谷神星。

直接搜索法指不显式使用导数的优化技术。不少实际问题导数不可得或者非常难计算,直接搜索法就成了求解这类问题的必经之路。直接搜索法仅需要计算函数值,从而具有易于使用、结构简单、所需内存小等优点;不足之处是该类方法大多依赖直观技巧,很难得到深入的

数学理论. 单纯形法是典型的直接法. 参考文献[30]是直接搜索法一个全面而深入的综述, 有需要或者感兴趣的读者可以参考.

习题 5

说明 该练习中计算部分的目的是使读者迅速获得有关本课程所学方法的一些第一手的经验. 我们建议将算法的每一段编制成独立的子程序, 然后以适当的组合调用它们. 比如, 有一个回溯一维搜索子程序(一个独立文件)将会非常有用, 可以在最速下降法的每次迭代中调用它. 当然, 需要考虑子程序的输入参数和输出参数. 再比如, 在最速下降法的每一次迭代中, 调用回溯一维搜索得到不同的单变量函数的近似极小点, 即满足某种条件的可接收步长.

此外, 计算练习中的一些具体细节留给读者自己决定, 比如在使用一维搜索时, 自行选择一维搜索的误差容限和参数. 还有, 自行确定自己的停止准则及程序的输出结果, 并对结果进行恰当的展示和分析.

- 5.1 考虑函数 $q(\mathbf{x}) = x_1^2 + 4x_2^2 - 4x_1 - 8x_2$.
 - (a) 画出函数的等值线; 求出该函数的极小点 \mathbf{x}^* .
 - (b) 对此函数, 考虑初始点 $\mathbf{x}^{(0)} = (0, 0)^T$ 的最速下降法; 说明方法在有限步内不能收敛到极小点 \mathbf{x}^* ; 是否存在其他初始点 $\mathbf{x}^{(0)}$ 使得最速下降法在有限步内收敛.
- 5.2 设将最速下降法应用于函数 $q(\mathbf{x}) = 2x_1^2 - 2x_1x_2 + x_2^2 + 2x_1 - 2x_2$ 得到序列 $\{\mathbf{x}^{(k)}\}$. 如果 $\mathbf{x}^{(2k+1)} = (0, 1 - 1/5^k)^T$, 证明 $\mathbf{x}^{(2k+3)} = (0, 1 - 1/5^{k+1})^T$. 在 $x_1 - x_2$ 平面上画出初始点 $\mathbf{x}^{(0)} = \mathbf{0}$ 的最速下降法产生的迭代序列, 并推断 $f(\mathbf{x})$ 的极小点.
- 5.3 考虑将最速下降法应用于 Hessian 阵为 \mathbf{G} 的正定二次函数. 设初始点 $\mathbf{x}^{(0)} \neq \mathbf{x}^*$, 且可以表示为

$$\mathbf{x}^{(0)} = \mathbf{x}^* + \mu \mathbf{s}$$

其中 \mathbf{s} 是 \mathbf{G} 的相应于特征值 λ 的特征向量.

- (a) 证明 $\mathbf{g}^{(0)} = \mu \lambda \mathbf{s}$, 且若沿最速下降方向进行精确一维搜索, 则方法进行一次迭代后终止.
- (b) 如果 \mathbf{G} 为单位阵的倍数, 证明对任意的初始点 $\mathbf{x}^{(0)}$, 方法经一次迭代后终止.
- (c) 若 $\mathbf{x}^{(0)}$ 不能表示成如上形式, 则 $\mathbf{x}^{(0)}$ 可表示为

$$\mathbf{x}^{(0)} = \mathbf{x}^* + \sum_{i=1}^m \mu_i \mathbf{s}_i$$

其中 $m > 1$, 且对所有的 i 有 $\mu_i \neq 0$, \mathbf{s}_i 是 \mathbf{G} 的对应于不同特征值 λ_i 的特征向量. 证明此时方法经一次迭代后不能终止.

- 5.4 考虑使用精确线搜索的最速下降法, 证明对所有的 k , 搜索方向 $\mathbf{p}^{(k+1)}$ 正交于 $\mathbf{p}^{(k)}$. 将该方法应用于函数 $q(\mathbf{x}) = 10x_1^2 + x_2^2$, 选取初始点 $\mathbf{x}^{(0)} = (1/10, 1)^T$. 从数值上验证本例可以达到最速下降法的最坏收敛速度, 即方法产生的 $q^{(k)} - q^*$ 以速率常数 $(\lambda_1 - \lambda_n)^2 / (\lambda_1 + \lambda_n)^2$ 线性收敛, 其中 λ_1, λ_n 分别为 \mathbf{G} 的最大和最小特征值. 注意, 如果 \mathbf{G} 相当病态, 则该因子可以任意接近于 1.

5.5 假设我们需要极小化 $q(\mathbf{x}) = 5x_1^2 + 5x_2^2 - x_1x_2 - 11x_1 + 11x_2 + 11$.

- (a) 找到一个满足一阶必要条件的解.
- (b) 说明该点是全局极小点.
- (c) 针对该问题, 最速下降法的收敛因子最大不会超过多少?
- (d) 从 $\mathbf{x}^{(0)} = (1, 1)^T$ 出发, 最多需要多少步可将目标函数值减小到 10^{-11} ?

5.6 考虑极小化函数

$$q(\mathbf{x}) = (10x_1^2 - 18x_1x_2 + 10x_2^2)/2 + 4x_1 - 15x_2 + 13$$

该问题的最优解是 $\mathbf{x}^* = (5, 6)^T$, 最优值 $f^* = -22$. 实现关于该问题的最速下降法, 其中线搜索的步长使用精确步长, 并选取下列初始点:

$$\mathbf{x}^{(0)} = (0, 0)^T$$

$$\mathbf{x}^{(0)} = (-0.4, 0)^T$$

$$\mathbf{x}^{(0)} = (10, 0)^T$$

$$\mathbf{x}^{(0)} = (11, 0)^T$$

针对上面的每个初始点, 确定线性收敛因子, 即序列 $(f^{(k+1)} - f^*)/(f^{(k)} - f^*)$ 的收敛子列中极限值的最大者(上极限).

5.7 假设我们想在定义域上利用牛顿法极小化函数 $f(x) = 9x - 4\ln(x-7)$.

- (a) 针对给定的 $x^{(0)}$ 值, 给出牛顿迭代的确切公式.
- (b) 计算从下面每一个点出发的牛顿法的 5 步迭代:

$$x^{(0)} = 7.40$$

$$x^{(0)} = 7.20$$

$$x^{(0)} = 7.01$$

$$x^{(0)} = 7.80$$

$$x^{(0)} = 7.88$$

(c) 依据经验验证算法对 $(7, 7.8888)$ 内的所有初始点收敛. 该区间之外, 牛顿法呈现出怎样的行为?

5.8 假设我们想在定义域上利用牛顿法极小化函数

$$f(\mathbf{x}) = -9x_1 - 10x_2 - \mu[\ln(100 - x_1 - x_2) + \ln x_1 + \ln x_2 + \ln(50 - x_1 + x_2)]$$

其中 μ 是参数, 定义域 $X = \{(x_1, x_2)^T : x_1 > 0, x_2 > 0, x_1 + x_2 < 100, x_1 - x_2 < 50\}$. 该练习要求实现关于该问题的牛顿法, 首先不用线搜索, 然后用线搜索. 针对 $\mu = 1$ 和 $\mu = 0.1$ 运行你的算法, 使用下面的初始点:

$$\mathbf{x}^{(0)} = (8, 90)^T$$

$$\mathbf{x}^{(0)} = (1, 40)^T$$

$$\mathbf{x}^{(0)} = (15, 68.69)^T$$

$$\mathbf{x}^{(0)} = (10, 20)^T$$

- (a) 当你运行针对该问题的没有线搜索的牛顿法时, 观察到怎样的行为?
- (b) 当你运行针对该问题的具有线搜索的牛顿法时, 观察到怎样的行为?

5.9 编写使用回溯线搜索(算法 4.3.1, 置初始步长 $\bar{\alpha} = 1$) 的最速下降法和牛顿法的程序. 利用它们极小化 Rosenbrock 函数(1.4.2), 并记录每次迭代的步长. 首先试用初始点 $\mathbf{x}^{(0)} = (1.2, 1.2)^T$, 然后试用更难一些的点 $\mathbf{x}^{(0)} = (-1.2, 1)^T$. 在平面上画出该函数的

等值线和两种算法的迭代轨迹.

- 5.10 验证点 $\mathbf{x}' = \mathbf{0}$ 与 $\mathbf{x}'' = (-\sqrt{7/12}, \sqrt{7/12})^\top$ 是函数

$$f(\mathbf{x}) = (x_1 + x_2)^2 + [2(x_1^2 + x_2^2 - 1) - 1/3]^2$$

的稳定点, 并确定每个稳定点的类型(即极小点、极大点或者鞍点). 说明初始点 $\mathbf{x}^{(0)} = (\sqrt{7/6}, 0)^\top$ 的牛顿法有定义, 并验证得到的 $\mathbf{x}^{(1)}$ 在连接点 \mathbf{x}' 和 \mathbf{x}'' 的直线上(不进行线搜索, 即牛顿法中步长取 1).

- 5.11 考虑函数

$$f(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2^2) e^{x_1^2 - x_2^2}$$

确定该函数的所有一阶和二阶偏导数, 并验证 $\mathbf{x}' = \mathbf{0}$ 是该函数的局部极小点. 计算函数在点 $\mathbf{x}' = (1, 1)^\top$ 处的梯度和 Hessian 阵, 并说明 Hessian 阵 \mathbf{G}' 非正定; 说明 $\lambda = 3$ 是使 $\mathbf{G}' + \lambda \mathbf{I}$ 为正定的最小正整数; 使用所得矩阵从 \mathbf{x}' 开始进行一次牛顿迭代(用基本牛顿法, 即不进行一维搜索).

- 5.12 应用牛顿法于函数

$$f(\mathbf{x}) = \frac{11}{546}x^6 - \frac{38}{364}x^4 + \frac{1}{2}x^2$$

其中 $\mathbf{x}^{(0)} = 1.01$ 为初始点. 验证 $\mathbf{G}^{(k)}$ 总是正定的, 且 $f^{(k)}$ 是单调递减的. 说明方法产生的序列 $\{\mathbf{x}^k\}$ 的聚点 $\mathbf{x}^\infty = \pm 1$, 且 $\mathbf{g}^\infty \neq \mathbf{0}$. 验证对任意固定的正数 ρ , 不管其多么小, Armijo 条件(4.3.1)对充分大的 k 总不成立(舍入误差起支配作用的情况除外).

- 5.13 应用牛顿法于函数

$$f(\mathbf{x}) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

说明初始点 $\mathbf{x}^{(0)} = \mathbf{0}$ 为什么不能很满意地应用该方法? 若在 $\mathbf{x}^{(0)}$ 处取搜索方向 $\mathbf{p} = -\mathbf{G}^{(0)^{-1}} \mathbf{g}^{(0)}$, 说明 $\pm \mathbf{p}$ 均不是下降方向, 从而不能使 $f(\mathbf{x})$ 减小. 如果在修正牛顿法中取 $\mathbf{G}^{(0)} + \lambda \mathbf{I}$, 那么 λ 取何值可使得 $\mathbf{G}^{(0)} + \lambda \mathbf{I}$ 正定? 当 $\lambda = 1$ 时, 得到的新点是多少? (不执行一维搜索, 即步长取 1.)

- 5.14 编写一个没有线搜索的基本牛顿法, 其中利用共轭梯度法求解方程组确定的搜索方向, 即截断牛顿法(详见 5.5 节), 选取停机准则使得收敛速度分别是线性、超线性和二次的. 针对下面的凸 4 次函数

$$f(\mathbf{x}) = 0.5\mathbf{x}^\top \mathbf{x} + 0.25\sigma(\mathbf{x}^\top \mathbf{G}\mathbf{x})^2$$

来试用你的程序, 其中 σ 是一个参数

$$\mathbf{G} = \begin{bmatrix} 5 & 1 & 0 & 0.5 \\ 1 & 4 & 0.5 & 0 \\ 0 & 0.5 & 3 & 0 \\ 0.5 & 0 & 0 & 2 \end{bmatrix}$$

这是一个严格凸函数, 且我们可以设置参数 σ 来控制它与二次函数的偏差. 初始点

$$\mathbf{x}^{(0)} = (\cos 70^\circ, \sin 70^\circ, \cos 70^\circ, \sin 70^\circ)^\top$$

对 $\sigma = 1$ 或更大的值求解所给问题, 并观察迭代的收敛速度. 提示: 该函数的梯度和 Hessian 矩阵分别是 $\nabla f(\mathbf{x}) = \mathbf{x} + \sigma(\mathbf{x}^\top \mathbf{G}\mathbf{x}) \mathbf{G}\mathbf{x}$ 和 $\nabla^2 f(\mathbf{x}) = \mathbf{I} + \sigma[(\mathbf{x}^\top \mathbf{G}\mathbf{x}) \mathbf{G} + 2\mathbf{G}\mathbf{x}\mathbf{x}^\top \mathbf{G}]$, 其中 \mathbf{I} 表示单位矩阵.

- 5.15 已知 $q(\mathbf{x})$ 是二次函数, Hessian 阵是对称正定矩阵 \mathbf{G} . 如果 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k-1)}$ 关于 \mathbf{G} 共轭, 证明函数 $h(\boldsymbol{\sigma}) = q(\mathbf{x}^{(0)} + \sigma_0 \mathbf{p}^{(0)} + \dots + \sigma_{k-1} \mathbf{p}^{(k-1)})$ 关于 $\boldsymbol{\sigma} = (\sigma_0, \dots, \sigma_{k-1})^T$ 是严格凸的.
- 5.16 设矩阵 \mathbf{S} 的列向量 $\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \dots, \mathbf{s}^{(n)}$ 关于正定矩阵 \mathbf{G} 共轭, 且 $\mathbf{S}^T \mathbf{G} \mathbf{S} = \mathbf{I}$. 则当且仅当 $\bar{\mathbf{S}} = \mathbf{S} \mathbf{Q}$ 时有 $\bar{\mathbf{S}}^T \mathbf{G} \bar{\mathbf{S}} = \mathbf{I}$, 其中 \mathbf{Q} 是正交矩阵.
- 5.17 定义向量组 $\mathbf{s}^{(i)} (i=1, 2, \dots, n)$ 为

$$s_j^{(i)} = \begin{cases} j, & j \leq i \\ 0, & j > i \end{cases}$$

- 证明该向量组关于三对角矩阵 \mathbf{G} 共轭, 其中 $g_{ii} = 2, g_{i,i+1} = g_{i+1,i} = -1, i=1, 2, \dots, n$.
- 5.18 利用共轭梯度法的迭代形式直接证明共轭梯度法的性质(式(5.2.8a)~式(5.2.8e))对 $k=1$ 成立.
- 5.19 实现算法 5.2.1, 并且用它求解线性方程组(5.2.1), 其中 \mathbf{G} 是 Hilbert 矩阵, 元素 $g_{ij} = 1/(i+j-1)$, 置右端向量 $\mathbf{b} = (1, 1, \dots, 1)^T$, 初始点 $\mathbf{x}^{(0)} = \mathbf{0}$. 试对维数 $n=5, 8, 12, 20$ 分别求解并报告迭代次数, 要求残量 $\mathbf{r}^{(k)} = \mathbf{G} \mathbf{x}^{(k)} - \mathbf{b}$ (即 $\mathbf{g}^{(k)}$) 的 2-范数减小到 10^{-6} 之下.
- 5.20 画出函数 $f(\mathbf{x}) = x_1^2 + 4x_2^2 - 4x_1 - 8x_2$ 的等值线, 由此确定该函数的极小点 \mathbf{x}^* .
- (a) 证明: 对此函数, 如果最速下降法中 $\mathbf{x}^{(0)} = \mathbf{0}$, 则迭代在有限步内不能收敛到极小点 \mathbf{x}^* ; 是否存在任何其他的 $\mathbf{x}^{(0)}$, 使得最速下降法在有限步内收敛?
- (b) 用 Fletcher-Reeves 法极小化该函数, 其中 $\mathbf{x}^{(0)} = \mathbf{0}$. 验证共轭梯度法的性质(式(5.2.8a)~式(5.2.8e))成立.
- (c) 确定初始点 $\mathbf{x}^{(0)}$ 使得 Fletcher-Reeves 法经一次迭代后收敛(因此退化为最速下降法). 验证序列 $\mathbf{g}^{(0)}, \mathbf{Gg}^{(0)}, \dots$ 中仅有一个独立向量.
- (d) 把 BFGS 法应用于该函数, 其中取 $\mathbf{x}^{(0)} = \mathbf{0}, \mathbf{H}^{(0)} = \mathbf{I}$. 证明: 直线 $\mathbf{x} = \mathbf{x}^{(1)} + \alpha \mathbf{p}^{(1)}$ 经过点 \mathbf{x}^* , 从而保证经两次迭代后方法的终止性. 验证该方法等价于 Fletcher-Reeves 法.
- 5.21 考虑四元二次函数 $0.5 \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{b}^T \mathbf{x}$, 其中 \mathbf{G} 为习题 5.17 中的三对角矩阵, $\mathbf{b} = (-1, 0, 2, \sqrt{5})^T$. 取初始点 $\mathbf{x}^{(0)} = \mathbf{0}$, 应用共轭梯度法极小化该函数, 并证明经两次迭代后终止, 且序列 $\mathbf{g}^{(0)}, \mathbf{Gg}^{(0)}, \dots$ 中只有两个独立向量.
- 5.22 (a) 如果 f 是连续可微的, 则 f 严格凸当且仅当
- $$f(\mathbf{y}) > f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}), \quad \forall \mathbf{x} \neq \mathbf{y}$$
- 利用该事实证明: 对于连续可微的严格凸函数, 曲率条件
- $$\mathbf{s}^{(k)}^T \mathbf{y}^{(k)} > 0$$
- 对任一向量 $\mathbf{x}^{(k)} \neq \mathbf{x}^{(k+1)}$ 成立, 其中 $\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \mathbf{y}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$.
- (b) 给出一个单变量函数 f 满足 $f(0) = -1$ 和 $f(1) = -1/4$, 并说明在这种情况下曲率条件不成立.
- 5.23 把精确步长的 DFP 法应用于习题 5.4, 其中 $\mathbf{H}^{(0)} = \mathbf{I}$.
- (a) 验证: 经过 n 次一维搜索后的二次终止性, 且 $\mathbf{H}^{(n)} = \mathbf{G}^{-1}$.
- (b) 验证该方法与共轭梯度法的等价性.
- 5.24 设矩阵 \mathbf{H} 对称正定.

(a) 证明 $\mathbf{H} = \mathbf{H} \mathbf{y} \mathbf{y}^T \mathbf{H} / \mathbf{y}^T \mathbf{H} \mathbf{y}$ 为奇异的半正定阵(可利用分解式 $\mathbf{H} = \mathbf{L} \mathbf{L}^T$ 及柯西不等式 $\mathbf{z}^T \mathbf{z} \mathbf{y}^T \mathbf{y} \geq (\mathbf{y}^T \mathbf{z})^2$).

(b) 证明当且仅当 $\mathbf{s}^{(k)}^T \mathbf{y}^{(k)} > 0$ 时 $\mathbf{H}_{\text{DFP}}^{(k+1)}$ 正定.

(c) 又设 \mathbf{H} 为对称的半正定阵且奇异, 即存在 $\mathbf{u} \neq \mathbf{0}$ 使得 $\mathbf{H} \mathbf{u} = \mathbf{0}$. 证明 $\mathbf{H}_{\text{DFP}}^{(k+1)}$ 奇异.

(d) 若 $\mathbf{H}^{(0)} \mathbf{u} = \mathbf{0}$, 证明当 $\mathbf{x}^* - \mathbf{x}^{(0)}$ 与 \mathbf{u} 不正交时, DFP 法不能确定出 \mathbf{x}^* .

- 5.25 一个 n 阶方阵 \mathbf{A} 的秩 $m (\leq n)$ 修正通常可以写成 $\mathbf{A}' = \mathbf{A} + \mathbf{R} \mathbf{S}^T$, 其中 \mathbf{R} 与 \mathbf{T} 为 $n \times m$ 阶矩阵, \mathbf{S} 为 $m \times m$ 阶矩阵. 验证

$$(\mathbf{A}')^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{R} \mathbf{U}^{-1} \mathbf{T}^T \mathbf{A}^{-1}$$

其中 $\mathbf{U} = \mathbf{S}^{-1} + \mathbf{T}^T \mathbf{A}^{-1} \mathbf{R}$ (Sherman-Morrison 公式)^[26]. 利用这个公式给出由 $\mathbf{H}^{(k)} (= \mathbf{B}^{(k-1)})$, $\mathbf{s}^{(k)}$ 及 $\mathbf{y}^{(k)}$ 表示的 $\mathbf{H}_{\text{DFP}}^{(k+1)}$ 的修正式, 并利用该事事实验证它的对偶公式 $\mathbf{B}_{\text{BFGS}}^{(k+1)}$, 取 $m = 2$ 及 $\mathbf{R} = \mathbf{T}$.

- 5.26 考虑在点 $t_i = -1, 0, 1, 2$ 处拟合函数 $\phi(t, \mathbf{x}) = x_1 e^{-x_2 t}$ 到数据值 $d_i = 2.7, 1, 0.4, 0.1$ 的问题, 这里 d_i 为精确值舍入至一位小数的结果. 以 $\mathbf{x}^{(0)} = (1, 1)^T$ 为初始点, 按 GN 法迭代一次, 计算所得近似解的误差. 近似地计算矩阵 $\nabla^2 r_i$ 及矩阵 $2\mathbf{B}^{-1} \sum r_i \nabla^2 r_i$, 并由此验证线性收敛速度与式(5.4.12)一致.

- 5.27 确定下列数据拟合问题的解, 其中

$$\phi(t, \mathbf{x}) = (1 - x_1 t / x_2)^{1/(x_1 c - 1)}, \quad c = 96.05$$

数据为

t_i	2 000	5 000	10 000	20 000	30 000	50 000
d_i	0.942 7	0.861 6	0.738 4	0.536 2	0.373 9	0.309 6

使用带线搜索的 GN 法求解问题(5.4.1), 并给出数据 d_i 与变量 x_1 和 x_2 的标准差.

- 5.28 考虑习题 5.26 中的数据拟合问题, 证明如果 x_2 固定, 则对 x_1 的线性最小二乘解为 $x_1 = \sum d_i e^{-x_2 t_i} / \sum e^{-2x_2 t_i}$, 而最优的平方和 $f = \sum d_i^2 - x_1 \sum d_i e^{-x_2 t_i}$, 于是可以把 x_1 与 f 看作 x_2 的函数, 并对 x_2 进行一次一维搜索以确定数据拟合问题的解. 验证所得解与习题 5.26 中的解相同.
- 5.29 已知函数在给定点 t_1, t_2, \dots, t_m 的取值为 y_1, y_2, \dots, y_m , 参数向量 $\mathbf{x} \in \mathbb{R}^n$ 未知. 试拟合函数 $y(t; \mathbf{x}) (\mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R})$ 使残量在最小二乘意义下最小. 在这种情况下 $y(t; \mathbf{x})$ 不是直接已知的, 但对所有的 t 和固定的 \mathbf{x} , 可通过求解常微分方程

$$y' = f(y, t, \mathbf{x}), \quad y(0) = h(\mathbf{x})$$

确定它, 其中函数 f 和 h 是已知的, 且 f 使得微分方程必须能够用数值方法求解. 由 GN 法求解该数据拟合问题还需要什么额外的条件? 说明求解如下一阶常微分方程初值问题(由向量 $\mathbf{z}(t, \mathbf{x}) (\mathbf{z} \in \mathbb{R}^{n+1})$ 定义的)可得到所需要的条件

$$z_1 = y, \quad z_{1+i} = \partial y / \partial x_i, \quad i = 1, 2, \dots, n$$

作为一个例子考虑常微分方程

$$y' = -x_1 y / (x_2 + y), \quad y(0) = x_3$$

及如下数据

y_i	24.44	19.44	15.56	10.56	9.07	6.85	4.07	1.67
t_i	0	23.60	49.10	74.50	80.00	100.00	125.50	147.30

利用上面的方法编制程序,给出最小二乘意义下的曲线拟合;给出平方和(精确到小数点后5位);对未知参数的初值估计 $x=(0.22, 3.27, 24.44)^T$;对使用该初值所得结果的精度和GN法成功求解该问题时所能达到的精度进行评价;使用更小的步长重复上面的计算,进而估计初值问题的截断误差对方法的影响.

第 6 章 无约束优化: 信赖域法

当牛顿法中 Hessian 阵不是正定的时, f 在 $x^{(k)}$ 的二阶 Taylor 展式(5.1.4)可能没有极小点, 或者极小点不唯一, 从而基本牛顿法没有定义. 对此另一种理解是: 使 Taylor 展式有效的区域相对较小, 以至于将 $q^{(k)}(s)$ 的极小点排除在外. 因此, 相应的方式是定义 $x^{(k)}$ 的某个邻域 Ω_k , 使得在这个邻域里 $q^{(k)}(s)$ 与 $f(x^{(k)} + s)$ 在某种意义上吻合得很好. 在 Ω_k 内选取一个极小化 $q^{(k)}(s)$ 的 $s^{(k)}$ 来直接得到下一个迭代点 $x^{(k+1)} = x^{(k)} + s^{(k)}$. 称 Ω_k 是信赖域(trust region), 即使得 Taylor 展式有效的区域. 正是由于这类方法将迭代步限制在信赖域内, 称为信赖域法(trust region method), 早期也称作限制步长法. 其发明起源于 Powell 的工作, 既能保持牛顿法的快速收敛, 又是大范围收敛的, 还兼具可靠性和通用性. 信赖域法的出现打破了线搜索法“垄断”优化算法的局面, 并发展成能与之平分优化算法天下的一类重要算法格式.

6.1 原型算法

下面讨论信赖域法的一些细节. 假设信赖域 $\Omega_k = \{x: \|x - x^{(k)}\| \leq \Delta_k\}$, 其中 Δ_k 称为信赖域半径(radius), 相应的子问题为

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad q^{(k)}(s) \\ & \text{subject to} \quad \|s\| \leq \Delta_k \end{aligned} \tag{6.1.1}$$

6.2 节将会看到, 对于 2-范数, 必定可以求到子问题(6.1.1)的全局解, 而其他的范数通常需要假定 $q^{(k)}$ 是凸函数才能保证求得全局解. 在迭代过程中, 需要及时地动态调整信赖域半径, 调整的指导原则是: 在 $f^{(k)} - q^{(k)}(s^{(k)})$ 和 $f^{(k)} - f(x^{(k)} + s^{(k)})$ 吻合得相当好时, 应该选取尽可能大的 Δ_k . 为此, 具体的量化方式是: 计算第 k 步 $f(x)$ 的真实下降量(actual reduction)

$$\delta f^{(k)} := f^{(k)} - f(x^{(k)} + s^{(k)})$$

和由模型得到的预计下降量(predicted reduction)

$$\delta q^{(k)} := f^{(k)} - q^{(k)}(s^{(k)})$$

并计算比值

$$\rho_k = \frac{\delta f^{(k)}}{\delta q^{(k)}} \tag{6.1.2}$$

该比值在一定意义下度量了 $q^{(k)}(s^{(k)})$ 逼近 $f(x^{(k)} + s^{(k)})$ 的程度: ρ_k 越接近 1, 说明吻合得越好. 由此可得信赖域法的一个原型算法 6.1.1. 下面先借助一个简单的例子来直观地理解该方法.

例 6.1.1 (信赖域法) 考虑

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \quad f(x_1, x_2) = (x_1^4 + 2x_1^3 + 24x_1^2) + (x_2^4 + 12x_2^2)$$

初始猜测 $x^{(0)} = (2, 1)^T$, 初始信赖域半径 $\Delta_0 = 1$. 在 $x^{(0)}$ 处, $f^{(0)} = 141$, $\mathbf{g}^{(0)} = \begin{bmatrix} 152 \\ 28 \end{bmatrix}$, $\mathbf{G}^{(0)} =$

Algorithm 6.1.1 A prototype algorithm

```

1: Given  $\mathbf{x}^{(0)}$  and  $\Delta_0$  ;
2: repeat
3:   calculate  $\mathbf{g}^{(k)}$  and  $\mathbf{G}^{(k)}$  ;
4:   solve formula(6.1.1) for  $\mathbf{s}^{(k)}$  ;
5:   evaluate  $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$  and hence  $\rho_k$  with formula (6.1.2) ;
6:   if  $\rho_k < 0.25$  then
7:      $\Delta_{k+1} = \|\mathbf{s}^{(k)}\|/4$  ;
8:   end if
9:   if  $\rho_k > 0.75$  and  $\|\mathbf{s}^{(k)}\| = \Delta_k$  then
10:    set  $\Delta_{k+1} = 2\Delta_k$  ;
11:   else
12:     set  $\Delta_{k+1} = \Delta_k$  ;
13:   end if
14:   if  $\rho_k \leq 0$  then
15:     set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$  ;
16:   else
17:     set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$  ;
18:   end if
19: until  $\|\mathbf{g}^{(k)}\|$  is sufficiently small

```

$\begin{bmatrix} 120 & 0 \\ 0 & 36 \end{bmatrix}$. 二阶 Taylor 近似为

$$q^{(0)}(\mathbf{s}) = 141 + 152s_1 + 28s_2 + \frac{1}{2}(120s_1^2 + 36s_2^2)$$

信赖域子问题是

$$\begin{aligned} & \underset{\mathbf{s} \in \mathbb{R}^2}{\text{minimize}} \quad q^{(0)}(\mathbf{s}) \\ & \text{subject to} \quad \sqrt{s_1^2 + s_2^2} \leq 1 \end{aligned}$$

因为子问题的 Hessian 阵正定, 故等值线是同心椭圆族, 中心

$$\mathbf{s}_N^{(0)} = -\mathbf{G}^{(0)-1}\mathbf{g}^{(0)} = \begin{bmatrix} -152/120 \\ -28/36 \end{bmatrix} \approx \begin{bmatrix} -1.266 & 7 \\ -0.777 & 8 \end{bmatrix}$$

是牛顿步. 由于 $\|\mathbf{s}_N^{(0)}\| = 1.4864 > \Delta_0 = 1$, 所以信赖域子问题的解在边界上得到, 此时即等价于解

$$\begin{aligned} & \underset{\mathbf{s} \in \mathbb{R}^2}{\text{minimize}} \quad q^{(0)}(\mathbf{s}) \\ & \text{subject to} \quad (s_1^2 + s_2^2)/2 = 1/2 \end{aligned}$$

由 Lagrange 乘子法, 需要解方程组

$$(120 + \lambda)s_1 = -152$$

$$(36 + \lambda)s_2 = -28$$

$$s_1^2 + s_2^2 = 1$$

由前两个方程解出 s_1 和 s_2 , 并代入第 3 个方程, 得

$$\left[\frac{152}{120 + \lambda} \right]^2 + \left[\frac{28}{36 + \lambda} \right]^2 = 1$$

用数值方法解方程得 $\lambda \approx 42.655$. 因此, 信赖域步 $s^{(0)} = (-0.9345, -0.3560)^T$ 满足 $\|s^{(0)}\| = 1$. 对于该信赖域步, 由模型函数得到的预计值 $q^{(0)}(s^{(0)}) = 43.6680$. 目标函数在 $x^{(0)}$ 处的值 $f(x^{(0)} + s^{(0)}) = 36.1668$. 于是, 真实-预计下降量的比率 $\rho_0 = (141 - 36.1668) / (141 - 43.6680) = 1.0771$. 因为 $\rho_0 > 3/4$, 该步成功, 所以 $x^{(1)} = x^{(0)} + s^{(0)} = (1.0665, 0.6440)^T$; 又因为 $\|s^{(0)}\| = 1$, 所以 $\Delta_1 = 2\Delta_0 = 2$.

线搜索法仅在一维搜索时通过测验来阻止那些难以驾驭的行为. 该方法的弊端: 首先, 选择下降方向与进行一维搜索明显不匹配, 前者是在 \mathbb{R}^n 的半空间 $\{p : p^T g^{(k)} < 0\}$ 中选取搜索方向, 后者则是在一个一维的射线上搜索. 其次, 二者有先后顺序, 如果下降方向选取得不合适, 那么线搜索只能将错就错. 信赖域法在选择搜索方向时则进行更多的控制, 希望这样能增大选取满步(步长为 1)的可能性, 而满步能使目标函数有效地减小. 这种自然的“保守”做法是信赖域法的基础. 上面的讨论换个说法, 即线搜索法从差步恢复的方式是沿着参数曲线(总是线性)回退的, 而信赖域法的恢复方式则是重新考虑整个求步的过程(见图 6.1.1). 信赖域法的优点之一是, 它具有非常强的大范围收敛性, 且对需要求解的问题没有太强的限制. 以下给出原型算法的收敛性结论, 借此可进一步理解信赖域法.

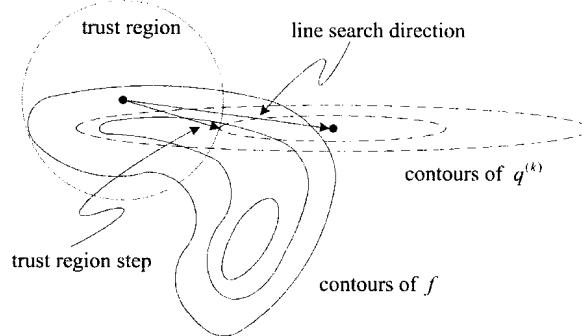


图 6.1.1 线搜索法与信赖域法

定理 6.1.1 (大范围收敛) 若算法 6.1.1 产生的序列 $\{x^{(k)}\}$ 有界, 且 $f(x)$ 二次连续可微, 则序列 $\{x^{(k)}\}$ 必有聚点 x^* 满足一阶和二阶最优化条件, 即 $g^* = 0$ 且 G^* 半正定.

证明 下面分 $\inf_k \Delta_k = 0$ 与 $\inf_k \Delta_k > 0$ 两种情况来考虑, 即存在收敛子序列 $x^{(k)} \rightarrow x^\infty, k \in \mathcal{K}$, 或 (i) $\rho_k < 0.25, \Delta_{k+1} \rightarrow 0$, 从而 $\|s^{(k)}\| \rightarrow 0$, 或 (ii) $\rho_k \geq 0.25$ 且 $\inf \Delta_k > 0$.

对上述任一情形, 可证明二阶必要条件仍然成立. 在情形(i)中, 设点 x^∞ 处存在下降方向 p ($\|p\| = 1$), 于是

$$p^T g^\infty = -d, \quad d > 0 \quad (6.1.3)$$

$f(x)$ 在点 $x^{(k)}$ 处的 Taylor 展式为

$$f(x^{(k)} + s^{(k)}) = q^{(k)}(s^{(k)}) + o(\|s^{(k)}\|^2)$$

由此

$$\delta f^{(k)} = \delta q^{(k)} + o(\|s^{(k)}\|^2) \quad (6.1.4)$$

对 $k \in \mathcal{K}$ 考虑沿着方向 \mathbf{p} 的长为 $\epsilon_k = \|\mathbf{s}^{(k)}\|$ 的步。根据 $\mathbf{s}^{(k)}$ 在子问题(6.1.1)中的最优性和连续性,由式(6.1.3)可得

$$\delta q^{(k)} \geq q^{(k)}(\mathbf{0}) - q^{(k)}(\epsilon_k \mathbf{p}) = -\epsilon_k \mathbf{p}^T \mathbf{g}^{(k)} + o(\epsilon_k) = \epsilon_k d + o(\epsilon_k)$$

再由 $\epsilon_k \rightarrow 0$ 和式(6.1.4)可知 $\rho_k = 1 + o(1)$, 这与 $\rho_k < 0.25$ 矛盾。因此,式(6.1.3)不成立,从而 $\mathbf{g}^{(k)} = \mathbf{0}$ 。

再设点 \mathbf{x}^* 处存在二阶下降方向 $\mathbf{p}(\|\mathbf{p}\|=1)$, 于是

$$\mathbf{p}^T \mathbf{G}^* \mathbf{p} = -d, \quad d > 0 \quad (6.1.5)$$

对 $k \in \mathcal{K}$, 考虑沿着方向 $\sigma \mathbf{p}$ 的长为 ϵ_k 的步, 其中 $\sigma = \pm 1$ 且选取 σ 使得 $\sigma \mathbf{p}^T \mathbf{g}^{(k)} \leq 0$ 。再次根据 $\mathbf{s}^{(k)}$ 的最优性和连续性得

$$\delta q^{(k)} \geq q^{(k)}(\mathbf{0}) - q^{(k)}(\epsilon_k \sigma \mathbf{p}) \geq -\frac{1}{2} \epsilon_k^2 \mathbf{p}^T \mathbf{G}^{(k)} \mathbf{p} = \frac{1}{2} \epsilon_k^2 d + o(\epsilon_k^2)$$

再由式(6.1.4)可知 $\rho_k = 1 + o(1)$, 与 $\rho_k < 0.25$ 矛盾。因此式(6.1.5)不成立,这表明 \mathbf{G}^* 半正定。由此当 $\inf_k \Delta_k = 0$ 时,可证明一阶和二阶必要条件仍然成立。

对情形(ii)中的子序列 \mathcal{K} , $f^{(0)} - f^* \geq \sum_{k \in \mathcal{K}} \delta f^{(k)}$, 再根据 $\rho_k \geq 0.25$ 可知 $\delta q^{(k)} \rightarrow 0$ 。定义 $q^*(s) = f^* + s^T \mathbf{g}^* + \frac{1}{2} s^T \mathbf{G}^* s$ 。设 $\bar{\Delta}$ 满足 $0 < \Delta < \inf_k \Delta_k$, 且 \bar{s} 在 $\|s\| \leq \bar{\Delta}$ 上最小化 $q^*(s)$ 。定义 $\bar{x} = \mathbf{x}^* + \bar{s}$, 可发现对充分大的 k , 有

$$\|\bar{x} - \mathbf{x}^{(k)}\| \leq \|\bar{s}\| + \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = \|\bar{s}\| + o(1) \leq \bar{\Delta} + o(1) \leq \Delta_k$$

从而 $\bar{x} - \mathbf{x}^{(k)}$ 是子问题的可行解。因此

$$q^{(k)}(\bar{x} - \mathbf{x}^{(k)}) \geq q^{(k)}(\bar{s}) = f^* - \delta q^{(k)}$$

取极限 $f^{(k)} \rightarrow f^*$, $\mathbf{g}^{(k)} \rightarrow \mathbf{g}^*$, $\mathbf{G}^{(k)} \rightarrow \mathbf{G}^*$, $\delta q^{(k)} \rightarrow 0$ 且 $\bar{x} - \mathbf{x}^{(k)} \rightarrow \bar{s}$ 。由此可知 $q^*(\bar{s}) \geq f^* = q^*(\mathbf{0})$, 所以 $\mathbf{s} = \mathbf{0}$ 是 $q^*(s)$ 在 $\|s\| \leq \bar{\Delta}$ 上的最小点, 又因为后一个约束是非积极的, 从而意味着一阶和二阶必要条件 $\mathbf{g}^* = \mathbf{0}$ 与 \mathbf{G}^* 半正定成立。 ■

在稍强一些的假设下,还可得到下列结论。

定理 6.1.2 (二阶收敛) 若定理 6.1.1 中的聚点 \mathbf{x}^* 满足二阶充分条件,即 \mathbf{G}^* 正定,则

(a) $\rho_k \rightarrow 1$, $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$, $\inf\{\Delta_k : k=1,2,\dots\} > 0$;

(b) 对充分大的 k , 信赖域约束 $\|s\| \leq \Delta_k$ 是非积极的,且收敛速度是二阶的。

证明 考虑 $k \in \mathcal{K}$, 其中 \mathcal{K} 是存在于定理 6.1.1 中的子序列。因为 \mathbf{G}^* 正定, 所以对充分大的 k , 向量 $\mathbf{p}^{(k)} = -\mathbf{G}^{(k)^{-1}} \mathbf{g}^{(k)}$ 是有定义的。选取适当的 α 使迭代步 $\alpha \mathbf{p}^{(k)}$ 在 $\|\alpha \mathbf{p}^{(k)}\| \leq \Delta_k$ 上最小化 $q^{(k)}(\alpha \mathbf{p}^{(k)})$ 。如果 $\|\mathbf{p}^{(k)}\| \leq \Delta_k$, 那么 $\alpha = 1$, 且 $\mathbf{s}^{(k)} = \mathbf{p}^{(k)}$ 是问题(6.1.1)的解。此时,

$$\delta q^{(k)} = \frac{1}{2} \mathbf{p}^{(k)^T} \mathbf{G}^{(k)} \mathbf{p}^{(k)} \geq \frac{1}{2} \mu_k \|\mathbf{s}^{(k)}\|^2$$

其中 $\mu_k > 0$ 是 $\mathbf{G}^{(k)}$ 的最小特征值。若 $\|\mathbf{p}^{(k)}\| > \Delta_k$, 则 $\alpha = \Delta_k / \|\mathbf{p}^{(k)}\| < 1$ 。对任意二次型, 当 $\alpha q'(\alpha) \leq 0$ 时可得

$$q(\alpha) = q(0) + \frac{1}{2} \alpha [q'(0) + q'(\alpha)] \leq q(0) + \frac{1}{2} \alpha q'(0)$$

因此

$$\delta q^{(k)} \geq \frac{1}{2} \alpha^2 \mathbf{p}^{(k)^T} \mathbf{G}^{(k)} \mathbf{p}^{(k)} \geq \frac{1}{2} \mu_k \Delta_k^2 \|\mathbf{p}^{(k)}\|^2 / \|\mathbf{p}^{(k)}\|^2 = \frac{1}{2} \mu_k \Delta_k^2$$

无论 $p^{(k)}$ 的长度是多少, 结论仍然成立. 根据式(6.1.4)可知 $\rho_k \rightarrow 1 (k \in \mathcal{K})$, 所以 x^∞ 只能在情形(ii)中出现, 而无法在情形(i)中得到. 若 k 充分大, 则 $\|x^{(k)} - x^\infty\| \leq \frac{1}{2} \inf_k \Delta_k$, 并且 $x^{(k)}$ 在点 x^∞ 的使得定理 5.1.5 仍然成立的邻域内, 从而 $x^{(k+1)} = x^{(k)} - G^{(k)}^{-1} g^{(k)}$ 满足 $\|x^{(k+1)} - x^\infty\| < \|x^{(k)} - x^\infty\|$, 且是问题(6.1.1)的可行解, 所以序列 $x^{(k)} \rightarrow x^\infty$; 同时, 对于这个子序列有 $\rho_k \rightarrow 1$ (同上) 与 $\inf_k \Delta_k > 0$. 由定理 5.1.5 可推出它是二阶收敛的. ■

定理 6.1.2 中, $\{x^{(k)}\}$ 有界的要求可用水平集 $\{x \mid f(x) \leq f(x^{(0)})\}$ 有界来保证. 这样, 除了水平集无界的病态情况外, 定理 6.1.2 是很强的结论. 它说明算法产生的序列中有子序列收敛于满足一阶和二阶必要条件的点, 这已经与充分条件很接近了; 当这个子序列的极限点还满足二阶充分条件时, 整个序列 $\{x^{(k)}\}$ 收敛. 信赖域法具有想要的特性, 即: 在 $x^{(k)}$ 接近局部解之前, 信赖域法用约束 $\|s\| \leq \Delta_k$ 来限制探测步 $s^{(k)}$, 使 $f(x)$ 获取充分下降; 而在 $x^{(k)}$ 接近局部解时, 该限制无效, 从而迭代恢复为快速收敛的基本牛顿法.

6.2 信赖域子问题

当信赖域由 2-范数定义时, 信赖域子问题(6.1.1)是在 2-范数定义的球内极小化二次函数, 即

$$\begin{aligned} \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad & q(s) := f + g^T s + \frac{1}{2} s^T B s \\ \text{subject to} \quad & s^T s / 2 \leq \Delta^2 / 2 \end{aligned} \quad (6.2.1)$$

其中 $\Delta > 0, g \neq 0$. 这里为了表述简洁和具有一般性, 将二阶 Taylor 展式中的 Hessian 阵替换成 $B^{(k)}$, 并去掉式(6.1.1)中的迭代指标 k . 首先研究信赖域子问题全局解的充分必要条件, 然后讨论基于此得到精确解的方法, 最后介绍得到非精确解的一些重要方法.

6.2.1 解的刻画

定理 6.2.1 (2-范数信赖域子问题的全局解) s^* 是问题(6.2.1)的全局解当且仅当存在 $\lambda^* \geq 0$ 使得

$$(B + \lambda^* I)s^* = -g \quad (6.2.2)$$

和

$$\lambda^*(\Delta^2 - s^{*T} s^*) = 0 \quad (6.2.3)$$

成立, 且 $B + \lambda^* I$ 半正定. 如果 $B + \lambda^* I$ 正定, 则 s^* 是问题(6.2.1)的唯一解.

证明 充分性. 设 s^* 和 λ^* 满足式(6.2.2)和式(6.2.3), 且 $B + \lambda^* I$ 半正定, 于是 s^* 极小化二次函数

$$\hat{q}(s) = g^T s + \frac{1}{2} s^T (B + \lambda^* I) s = q(s) + \frac{1}{2} \lambda^* s^T s \quad (6.2.4)$$

这样, 对任一 $s \in \mathbb{R}^n$ 有 $\hat{q}(s) \geq \hat{q}(s^*)$, 即

$$q(s) \geq q(s^*) + \frac{1}{2} \lambda^* (s^{*T} s^* - s^T s)$$

如果 $s^T s \leq \Delta^2$, 由式(6.2.3)和 $\lambda^* \geq 0$ 有 $q(s) \geq q(s^*)$, 因此 s^* 是问题(6.2.1)的全局解. 如果

$\mathbf{B} + \lambda^* \mathbf{I}$ 正定, 且 $s \neq s^*$, 则有 $\hat{q}(s) > \hat{q}(s^*)$, 可得 s^* 是(6.2.1)的唯一全局解.

反之, 设 s^* 求解问题(6.2.1). 若 $s^{*T} s^* < \Delta^2$, 则 s^* 是无约束极小点, 由无约束优化的二阶最优化条件知 $\lambda^* = 0$ 满足条件; 否则, 约束是积极的. 约束在 s^* 的法向量 $s^* \neq \mathbf{0}$ (由 $\Delta > 0$ 可得) 是线性无关的. 这蕴含着引理 7.3.3(ii) 成立, 从而定理 7.2.1 和定理 7.4.1 成立. 这样, 存在乘子 $\lambda^* \geq 0$ 使得 $\nabla_s \mathcal{L}(s^*, \lambda^*) = \mathbf{0}$, 其中

$$\mathcal{L}(s, \lambda^*) = q(s) + \frac{1}{2} \lambda^* (s^T s - \Delta^2) \quad (6.2.5)$$

是子问题(6.2.1)的 Lagrange 函数, 此即条件(6.2.2). 定理 7.2.1 的互补条件即式(6.2.3). 由定理 7.4.1 的二阶必要条件, 有

$$v^T (\mathbf{B} + \lambda^* \mathbf{I}) v \geq 0, \quad \forall v: v^T s^* = 0 \quad (6.2.6)$$

进一步, 考虑任意的向量 $w: w^T s^* \neq 0$, 如图 6.2.1 所示, 可以构造位于约束边界的可行点 $s' = s^* + \theta w, \theta \neq 0$. 将 s' 代入二次函数(6.2.5)在 s^* 处的二阶 Taylor 展式, 由 $\nabla_s \mathcal{L}(s^*, \lambda^*) = \mathbf{0}$ 和 $\|s^*\| = \Delta$ 有

$$\mathcal{L}(s', \lambda^*) = q(s^*) + \frac{1}{2} (s' - s^*)^T (\mathbf{B} + \lambda^* \mathbf{I}) (s' - s^*)$$

而 $\|s'\| = \Delta$ 和 s^* 的最优性蕴含着 $\mathcal{L}(s', \lambda^*) = q(s') \geq q(s^*)$. 将此代入上式, 即得 $w^T (\mathbf{B} + \lambda^* \mathbf{I}) w \geq 0$. 这和式(6.2.6)说明 $\mathbf{B} + \lambda^* \mathbf{I}$ 是半正定的. ■

在第 7 章中将介绍约束规划的最优化条件. 对凸规划而言, 充分条件等同于必要条件; 对一般的非凸规划, 二者之间存有间隙, 通常很难给出非凸规划(即可能有许多局部解)的全局解的充分必要条件. 这里信赖域子问题是一个特例, 定理 6.2.1 给出的条件是全局解的充分和必要条件. 究其原因, 信赖域子问题是一个隐(hidden)凸规划问题, 即本质上等价于一个凸规划. 具体地, 因为 \mathbf{B} 是对称的, 所以存在正交矩阵 $U = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_n]$ 和对角矩阵 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ 满足 $\mathbf{B} = U \Lambda U^T$, 其中 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ 是 \mathbf{B} 的特征值(即对 \mathbf{B} 进行特征值分解, 也称谱分解). 令 $s = U^T x$, 则子问题(6.2.1)等价于

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \mathbf{g}^T U^T x + \frac{1}{2} x^T \Lambda x \\ & \text{subject to} && x^T x \leq \Delta^2 \end{aligned}$$

再引进变量 $y_i = x_i^2, i = 1, 2, \cdots, n$. 读者不难验证上面的问题等价于

$$\begin{aligned} & \underset{y \in \mathbb{R}^n}{\text{minimize}} && \sum_{i=1}^n -|(\mathbf{Ug})_i| \sqrt{y_i} + \frac{1}{2} \sum_{i=1}^n \lambda_i y_i \\ & \text{subject to} && \sum_{i=1}^n y_i \leq \Delta^2, \quad y \geq \mathbf{0} \end{aligned}$$

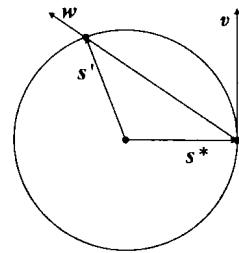


图 6.2.1 向量 s' 的构造

该问题的目标函数是凸函数, 约束是线性的, 故是凸规划问题. 至此, 读者不必再惊讶于为何能给出信赖域子问题这样一个非凸优化问题的全局解了.

6.2.2 求解子问题的牛顿法

早期求解 2-范数信赖域子问题的算法类似于算法 5.4.1, 即将 λ 作为迭代控制参数, 由 λ 确定步 $s(\lambda)$. 在现代信赖域法中, 如算法 6.1.1, 将信赖域半径 Δ 作为迭代控制参数, 由此确定非负 λ 使得 $\|s(\lambda)\| \leq \Delta$. 对于现代信赖域法, 一种典型的方法是沿着 LM 轨道 (LM trajectory)

$$\{s : (\mathbf{B} + \lambda \mathbf{I})s = -\mathbf{g}, \lambda \geq 0\}$$

进行搜索的迭代法. 在设计具体的算法之前, 先借助 \mathbf{B} 的特征值分解对 LM 轨道进行分析.

利用上述的特征值分解 $\mathbf{B} = \mathbf{U} \Lambda \mathbf{U}^T$, 并根据式(6.2.2), 可以用 λ 显式表示 s , 即

$$s(\lambda) = -(\mathbf{B} + \lambda \mathbf{I})^{-1} \mathbf{g} = -\sum_{j=1}^n \frac{\gamma_j}{\lambda_j + \lambda} \mathbf{u}_j, \quad (6.2.7)$$

其中 $\gamma_j = \mathbf{u}_j^T \mathbf{g}$. 假定存在 $i \in \mathcal{I} := \{i : \lambda_i = \lambda_n\}$, 使得 $\gamma_i \neq 0$ (即存在与 λ_n 对应的特征向量与 \mathbf{g} 不正交), 则函数

$$\psi(\lambda) := \|s(\lambda)\|^2 = \|\mathbf{U}^T (\Lambda + \lambda \mathbf{I})^{-1} \mathbf{U} \mathbf{g}\|^2 = \sum_{j=1}^n \frac{\gamma_j^2}{(\lambda_j + \lambda)^2} \quad (6.2.8)$$

在区间 $(-\lambda_n, \infty)$ 上是 λ 的连续、单调减函数, 且有 $\lim_{\lambda \rightarrow -\infty} \psi(\lambda) = 0$ 和 $\lim_{\lambda \rightarrow \lambda_n} \psi(\lambda) = \infty$. 因此, 可以在这个区间中找到唯一的 λ 满足

$$\psi(\lambda) = \Delta^2 \quad (6.2.9)$$

如果这个解 $\lambda \geq 0$, 则找到了信赖域子问题的解. 如果这个解 $\lambda < 0$ 或者 $\forall i \in \mathcal{I}, \gamma_i = 0$, 则会出现一些其他情况. 具体可以分成下面的 3 种情况:

(i) $\lambda_n > 0$, 此时 \mathbf{B} 正定, 牛顿步 $s(0) = -\mathbf{B}^{-1} \mathbf{g}$ 有定义. 若 $\|s(0)\| \leq \Delta$, 则牛顿步 $s(0)$ 是信赖域子问题的解, 这时信赖域约束是非积极的 ($\lambda^* = 0$); 否则, 方程(6.2.9)在 $(0, \infty)$ 内有唯一正根, 这时信赖域约束是积极的.

(ii) $\lambda_n \leq 0$ 且存在 $i \in \mathcal{I}$ 使得 $\gamma_i \neq 0$. 此时方程(6.2.9)在 $(-\lambda_n, \infty)$ 内有唯一正根, 这时信赖域约束是积极的.

(iii) $\lambda_n \leq 0$ 且 $\forall i \in \mathcal{I}, \gamma_i = 0$. 令

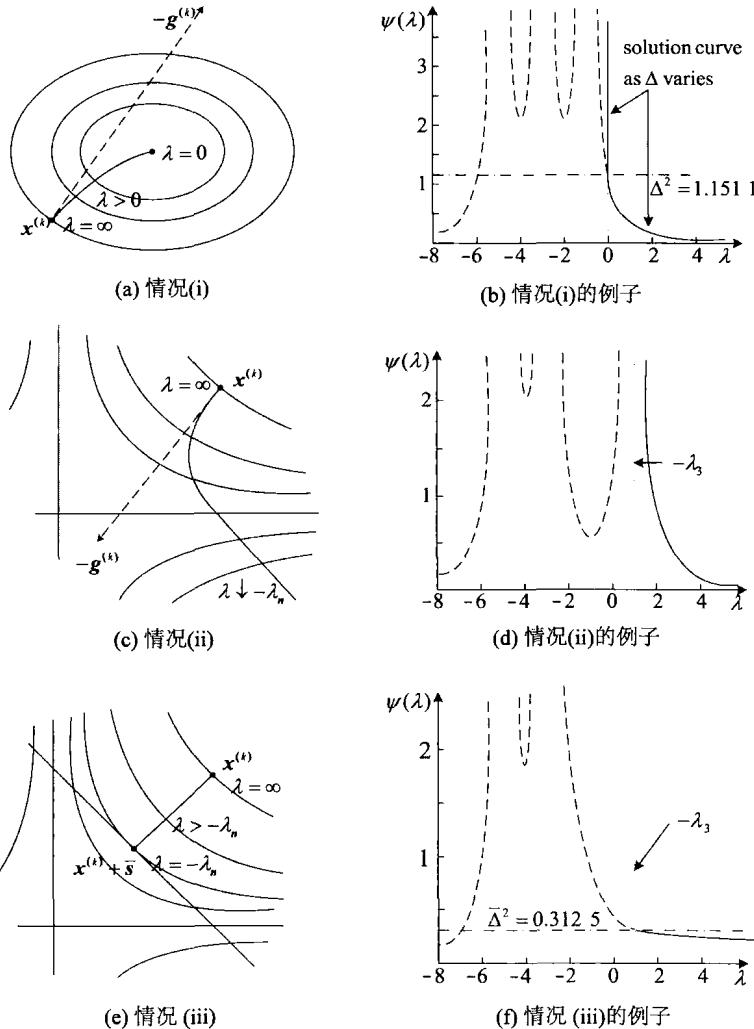
$$\bar{s} = \sum_{j : \lambda_j \neq \lambda_n} \frac{\gamma_j}{\lambda_j + \lambda_n} \mathbf{u}_j, \quad \bar{\Delta} = \|s\|$$

则随着 λ 从 $-\lambda_n$ 增加到 ∞ , $\psi(\lambda)$ 从 $\bar{\Delta}$ 减小到 0. 若 $\bar{\Delta} > \Delta$, 则方程(6.2.9)在 $(-\lambda_n, \infty)$ 内有唯一正根; 否则, 方程(6.2.9)在 $(-\lambda_n, \infty)$ 内无根. 但定理 6.2.1 保证在区间 $[-\lambda_n, \infty)$ 内有根, 故必有 $\lambda^* = -\lambda_n$, 称其是复杂情况 (hard case). 此时 $\mathbf{B} + \lambda^* \mathbf{I}$ 是奇异的, 易验证形如 $s = \bar{s} + \tau \mathbf{u}_n$ 的向量 s 满足定理 6.2.1 所需要的条件, 其中 τ 使得 $\|s\| = \Delta$, \mathbf{u}_n 是与 λ_n 对应的特征向量.

图 6.2.2 中的(a)、(c)和(e)是上述 3 种情况的图示. 下面再用例子来说明这 3 种情况.

例 6.2.1 考虑两个问题, 二者的梯度皆为 $\mathbf{g} = (1, 1, 1)^T$, 但 Hessian 阵分别为

$$\mathbf{B}_a = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_b = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

图 6.2.2 LM 轨道和函数 $\psi(\lambda)$

Hessian 阵 \mathbf{B}_a 是正定的, 子问题(6.2.1)是凸规划, 这里的 $\psi(\lambda)$ 如图 6.2.2(b)所示. 当 $\Delta^2 \geq \|\mathbf{B}^{-1}\mathbf{g}\|^2 = 1.151 1$ 时, 问题的解位于信赖域的内部, 且 $s^* = -\mathbf{B}^{-1}\mathbf{g}$; 否则, 解在信赖域的边界上. 这时先求方程(6.2.9)的最右端的根, 之后将它代入(6.2.7)得到 s^* . 然而 Hessian 阵 \mathbf{B}_b 是不定的, 对应子问题(6.2.1)是非凸的, 这里的 $\psi(\lambda)$ 如图 6.2.2(d)所示. 此时 $\lambda \geq -\lambda_n > 0$, 故解必须在信赖域的边界上, 且再一次求方程(6.2.9)在 $-\lambda_n$ 右边的根得到要找的 λ , 再由式(6.2.7)得到 s^* .

例 6.2.2 (复杂情况) 现在考虑 $\mathbf{B}_c = \mathbf{B}_b$ 且 $\mathbf{g} = (1, 1, 0)^\top$ 的问题. 这时子问题(6.2.1)也是非凸的, 这里的 $\psi(\lambda)$ 如图 6.2.2(f)所示. 式(6.2.7)中的系数 $\gamma_n = 0$, 即 \mathbf{g} 正交于 \mathbf{B} 的(与特征值 $\lambda_3 = -1$ 所对应)特征向量 \mathbf{u}_3 . 令

$$\bar{s} = \frac{1}{5-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{3-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ 0 \end{bmatrix}$$

则 $\bar{\Delta}^2 = \|\bar{s}\|^2 = 0.3125$. 注意, 由定理 6.2.1 有 $\lambda \geq 1 = -\lambda_n$. 但是图 6.2.2(f) 显示, 如果 Δ 大于 $\bar{\Delta}$, 则方程(6.2.9)没有大于 1 的根. 这就是上面情况(iii)中的复杂情况, 即 $\lambda_n \leq 0$, 且 λ_n 的所有特征向量都与 \mathbf{g} 正交, 且 $\bar{\Delta} \leq \Delta$. 当 Δ 较大时, 经常会出现这种情况. 此时 $\lambda^* = 1$ 且 $s^* = \bar{s} + \tau(0, 0, 1)^T$, 其中 $\tau = \sqrt{\Delta^2 - 0.3125}$.

由上面分析可知, 除了情况(i)中 $\lambda = 0$ (此时牛顿步是信赖域子问题的解) 和情况(iii)中 $\bar{\Delta} \leq \Delta$ (此时取 $\lambda = -\lambda_n$) 外, 都需要在特定区间内求单变量非线性方程(6.2.9)的根. 注意, 一旦得到 λ , 就得到信赖域子问题的解(6.2.7). 下面讨论如何更好地解方程(6.2.9). 图 6.2.2 中的(b)、(d) 和 (f) 表明函数 $\phi(\lambda)$ 的图形有许多趋向无穷大的渐近线, 直接求解方程(6.2.9)并不是最好的做法. 一种更好的做法是求解等价方程

$$\phi(\lambda) := \frac{1}{\|s(\lambda)\|_2} - \frac{1}{\Delta} = 0 \quad (6.2.10)$$

如图 6.2.3 所示, 这里的函数 $\phi(\lambda)$ 性态良好, 适合于用牛顿法求解. λ 处的牛顿校正是 $-\phi'(\lambda)/\phi''(\lambda)$. 因为

$$\phi(\lambda) = \frac{1}{\|s(\lambda)\|_2} - \frac{1}{\Delta} = \frac{1}{(s(\lambda)^T s(\lambda))^{\frac{1}{2}}} - \frac{1}{\Delta}$$

对 $\phi(\lambda)$ 微分, 得

$$\phi'(\lambda) = -\frac{s(\lambda)^T \nabla s(\lambda)}{\|s(\lambda)^T s(\lambda)\|^{\frac{3}{2}}} = \frac{-s(\lambda)^T \nabla s(\lambda)}{\|s(\lambda)\|_2^3}$$

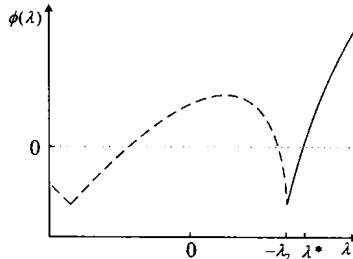


图 6.2.3 问题“ $\min -\frac{1}{4}s_1^2 + \frac{1}{4}s_2^2 + \frac{1}{2}s_1 + s_2$ subject to $\|s\| \leq 4$ ”的 $\phi(\lambda)$

此外, 再对定义 $s(\lambda)$ 的方程 $(\mathbf{B} + \lambda \mathbf{I})s(\lambda) = -\mathbf{g}$ 进行微分, 必有 $(\mathbf{B} + \lambda \mathbf{I})\nabla s(\lambda) + s(\lambda) = \mathbf{0}$. 这里不用求出 $\nabla s(\lambda)$ 的值, 只需替换 $\phi'(\lambda)$ 的表达式中的分子

$$-s(\lambda)^T \nabla s(\lambda) = s(\lambda)^T (\mathbf{B} + \lambda \mathbf{I})^{-1} s(\lambda)$$

当得到分解 $\mathbf{B} + \lambda \mathbf{I} = \mathbf{L}(\lambda) \mathbf{L}^T(\lambda)$ 后, 可以得到以下简单的关系:

$$s(\lambda)^T (\mathbf{B} + \lambda \mathbf{I})^{-1} s(\lambda) = s(\lambda)^T \mathbf{L}^{-T}(\lambda) \mathbf{L}^{-1}(\lambda) s(\lambda) = \|w(\lambda)\|_2^2$$

其中 $\mathbf{L}(\lambda)w(\lambda) = s(\lambda)$. 算法 6.2.1 是 Hebden 于 1973 年提出的求解标量方程(6.2.10)的牛顿法的伪码.

Algorithm 6.2.1 Hebbden iteration for equation(6.2.10)

```

1: Given  $\lambda > -\lambda_n, \Delta > 0$ ;
2: repeat
3:   factorize  $\mathbf{B} + \lambda \mathbf{I} = \mathbf{L} \mathbf{L}^\top$ ;
4:   solve  $\mathbf{L} \mathbf{L}^\top \mathbf{s} = -\mathbf{g}$ ;
5:   solve  $\mathbf{L} \mathbf{w} = \mathbf{s}$ ;
6:   set  $\lambda = \lambda + \left( \frac{\|\mathbf{s}\|_2 - \Delta}{\Delta} \right) \left( \frac{\|\mathbf{s}\|_2}{\|\mathbf{w}\|_2} \right)^2$ ;
7: until convergence

```

除了复杂情况外,若从区间 $(-\lambda_n, \lambda^*)$ 里的点开始迭代,则该算法是大范围收敛的,且最终是二阶收敛的。需要采取一些保护措施,使得算法在遇到复杂情况和解在信赖域内部时也是稳健的。虽然该算法通常只需要迭代3~4次,但每次迭代的主要计算花费是 $\mathbf{B} + \lambda \mathbf{I}$ 的Cholesky分解。对小规模问题而言还可以接受,而当变量的个数很多时,它将非常昂贵,以致不能接受。

6.3 求解子问题的近似方法

由6.2节的分析可知,精确求解信赖域子问题的计算量偏大,除了一些非常特殊的情况外,使用精确方法求解大规模信赖域子问题是不可能的。然而也不需要悲观,在线搜索时面临同样的问题——精确线搜索代价很高,以至于需要用非精确线搜索来取而代之。作为借鉴,这里同样可以考虑非精确解法。类似各种非精确线搜索条件,也需要给出一个近似解的“近似”标准。

6.3.1 柯西点

前面已经介绍过最速下降法有非常强的(理论)收敛性质。该事实在信赖域框架下同样是成立的。基于该事实,可以将柯西点(Cauchy point),即在模型(6.2.1)中将 \mathbf{s} 限制到信赖域内沿负梯度方向所得到的极小点,作为近似标准,其示意图如图6.3.1所示。定理6.4.1表明这是保证实用信赖域法(算法6.4.1)收敛到一阶临界点的“最小条件”。为了描述方便,记柯西点为 \mathbf{s}_C ,即 $\mathbf{s}_C = -\alpha_C \mathbf{g}$,其中

$$\alpha_C = \arg \min \{ \phi(\alpha) : \|\alpha \mathbf{g}\| \leq \Delta \} = \arg \min \left\{ \phi(\alpha) : 0 \leq \alpha \leq \frac{\Delta}{\|\mathbf{g}\|} \right\}$$

这里 $\phi(\alpha) = -\|\mathbf{g}\|^2 \alpha + \frac{1}{2} \mathbf{g}^\top \mathbf{B} \mathbf{g} \alpha^2$ 。要求所选的步 \mathbf{s}' 满足

$$q(\mathbf{s}') \leq q(\mathbf{s}_C) \quad \text{且} \quad \|\mathbf{s}'\| \leq \Delta \quad (6.3.1)$$

柯西点极易求,因为求它仅需在区间上极小化二次函数 $\phi(\alpha)$ 。在实践中,希望(并且能够)比这做得更好,但在收敛性结论中仅需式(6.3.1)即可。注意,柯西点保证模型函数 $q(\mathbf{s})$ 有一个合理的下降,而且下降量依赖于当前梯度的范数大小,同时也受信赖域半径和Hessian阵的影响。

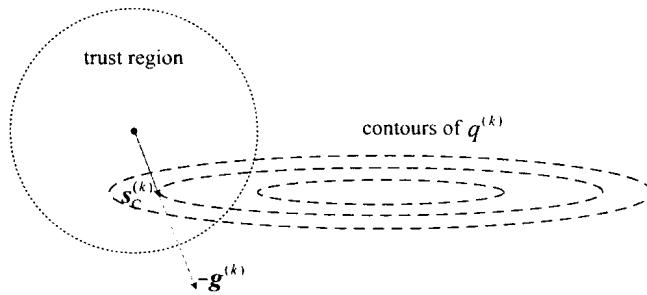


图 6.3.1 柯西点

定理 6.3.1 (模型在柯西点处所获取的下降量) 设 s_C 是柯西点, $f = f^{(k)}$. 则

$$f - q(s_C) \geq \frac{1}{2} \|g\| \min \left(\frac{\|g\|}{\|B\|}, \Delta \right)$$

证明 易验证 $f - q(s_C) = -\phi(\alpha_C)$, 从而得到 α_C 即可解决问题. $\phi(\alpha)$ 在区间 $[0, \Delta/\|g\|]$ 上的极小点与二次项系数的符号有关, 即要讨论 $\phi(\alpha)$ 的图形是开口向上、还是向下的抛物线.

为表示简单, 记 $a = g^T B g$, $d = \Delta/\|g\|$. 如果 $a \leq 0$, 则 $\phi(\alpha)$ 在区间 $(0, \infty)$ 上单调不增, 此时柯西点落在信赖域边界上, 即 $\alpha_C = d$. 这样,

$$-\phi(\alpha_C) \geq \alpha_C \|g\|^2 \geq \frac{1}{2} \Delta \|g\|^2$$

如果 $a > 0$, 设 α^* 为 $\phi(\alpha)$ 的唯一极小点, 则

$$\alpha^* = \frac{\|g\|^2}{a} \quad (6.3.2)$$

若无约束极小点超出信赖域边界, 即 $\alpha^* > d$, 则 $\alpha_C = d$. 由式(6.3.2)有 $\alpha_C a \leq \alpha^* a = \|g\|^2$, 从而 $-\phi(\alpha_C) = \alpha_C \|g\|^2 - (\alpha_C)^2 a / 2 \geq \alpha_C \|g\|^2 / 2 = \Delta \|g\|^2 / 2$. 若无约束极小点落在信赖域内或位于信赖域边界上, 则 $\alpha_C = \alpha^*$. 因此, 由 Cauchy-Schwarz 不等式和范数的相容性有 $a = |g^T B g| \leq \|g\|^2 \|B\|$, 从而

$$-\phi(\alpha_C) = -\phi(\alpha^*) = \frac{1}{2} \frac{\|g\|^4}{g^T B g} \geq \frac{1}{2} \frac{\|g\|^2}{\|B\|}$$

因为设计的近似算法要求步至少和柯西点一样好, 所以有下面的推论. 它将模型函数的减少量和度量最优性的距离(即梯度的范数)结合起来, 是信赖域法中的经典结论.

推论 如果 $s^{(k)}$ 满足条件(6.3.1), 则

$$f^{(k)} - q^{(k)}(s^{(k)}) \geq \frac{1}{2} \|g^{(k)}\| \min \left(\frac{\|g^{(k)}\|}{\|B^{(k)}\|}, \Delta_k \right)$$

证明 由定理 6.3.1 和 s' 满足条件(6.3.1)立即可得. ■

6.3.2 Dog-leg 法

像前面已经提到的, 目的是找到 s' 满足条件(6.3.1), 而与基本要求(即至少和柯西点一样好)相一致的最简单近似是柯西点本身. 当然, 如果这样做, 那么得到的即是简单的最速下降法, 一般不可能成为实用方法. 一种稍微复杂些, 且仅适用于模型函数是严格凸二次函数的方法是折线法(dog-leg method). 由 6.2.2 小节的分析可知, 在情形(i)中 $s(0)$ 是牛顿步, 并且当 $\lambda \rightarrow \infty$ 时, $s(\lambda) \rightarrow -g^{(k)}/\lambda$, 即增量最速下降步(见图 6.2.2). 对于一般情形, $s(\lambda)$ 可理解成这些

极端情形之间的内部插值. 因此, 可以利用一条折线, 即 dog-leg 轨道(Powell 于 1970 提出)来逼近 $s(\lambda)$, 如图 6.3.2 所示. 若 $\mathbf{B}^{(k)}$ 是 $\mathbf{G}^{(k)}$ 的近似, 则轨道由两条线段组成: 一段连接 $s=0$ 和修正柯西点 \hat{s}_c (即沿着最速下降方向预测的最优修正), 而另一段连接 \hat{s}_c 和类牛顿校正 s_N . 这里

$$\hat{s}_c = -\mathbf{g}^{(k)} \|\mathbf{g}^{(k)}\|_2^2 / \mathbf{g}^{(k)T} \mathbf{B}^{(k)} \mathbf{g}^{(k)}, \quad s_N = -\mathbf{B}^{(k)T} \mathbf{g}^{(k)}$$

在算法 6.1.1 中利用所得轨道, 可以得到类似的算法. 具体地, 仅需要将算法 6.1.1 中根据式(6.2.2)计算 $s(\lambda)$, 替换成在 dog-leg 轨道中确定长度为 Δ_k 的点 $s^{(k)}$. 若 $\|s_N\|_2 \leq \Delta_k$, 则可直接取为 s_N (如情形(i)), 也可以证明利用这种轨道的方法收敛到稳定点.

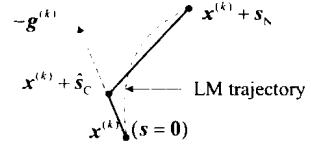


图 6.3.2 Powell 的 dog-leg 轨道

6.3.3 Steihaug 共轭梯度法

一种显而易见的方法是利用迭代法来逼近子问题的解, 比如推广共轭梯度法来求解子问题. 共轭梯度法中, 若取初始点为零, 则第一步是最速下降方向, 且后继的迭代进一步使目标值下降. 该方法产生的任一步作为信赖域子问题的近似解, 在这里的理论框架下均是容许的, 然而必须先处理一些具体的问题, 比如共轭梯度法与信赖域法之间如何进行交互, 当 \mathbf{B} 不定时如何处理等. 考虑用共轭梯度法近似求解问题(6.2.1). 记迭代过程中产生的点为 $\mathbf{x}^{(i)}$, 搜索方向为 $\mathbf{p}^{(i)}$, 步长为 α_i , 梯度为 $\mathbf{r}^{(i)} = \mathbf{B}\mathbf{x}^{(i)} + \mathbf{g}$, 则方法的伪码见算法 6.3.1.

Algorithm 6.3.1 A framework with conjugate gradient method for problem(6.2.1)

1: Given $\mathbf{x}^{(0)} = \mathbf{0}$, set $\mathbf{r}^{(0)} = \mathbf{g}$, $\mathbf{p}^{(0)} = -\mathbf{g}$ and $i = 0$;

2: **repeat**

3: set $\mathbf{d} = \mathbf{B}\mathbf{p}^{(i)}$;

4: set $\alpha_i = \frac{\mathbf{r}^{(i)T} \mathbf{r}^{(i)}}{\mathbf{p}^{(i)T} \mathbf{d}}$;

5: set $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)}$;

6: set $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} + \alpha_i \mathbf{d}$;

7: set $\beta_{i+1} = \frac{\mathbf{r}^{(i+1)T} \mathbf{r}^{(i+1)}}{\mathbf{r}^{(i)T} \mathbf{r}^{(i)}}$;

8: set $\mathbf{p}^{(i+1)} = -\mathbf{r}^{(i+1)} + \beta_{i+1} \mathbf{p}^{(i)}$;

9: set $i = i + 1$;

10: **until** “break down” or $\mathbf{r}^{(i)}$ “small”.

需要指出的是, 当算法崩溃(break down)时即终止. 如此处理是因为若 \mathbf{B} 不正定, 则会出现当 $\mathbf{p}^{(i)T} \mathbf{B} \mathbf{p}^{(i)} = 0$ (或者实践中很接近零) 时迭代无定义的致命情况和当 $\mathbf{p}^{(i)T} \mathbf{B} \mathbf{p}^{(i)} < 0$ (此时 $q(s)$ 沿着负曲率方向 $\mathbf{p}^{(i)}$ 无下界) 时的非致命情况. 关于信赖域约束有如下非常重要的结论.

定理 6.3.2 假设应用初始点 $\mathbf{x}^{(0)} = \mathbf{0}$ 的共轭梯度法求解问题(6.2.1), 且对所有 $0 \leq i \leq k$ 有 $\mathbf{p}^{(i)T} \mathbf{B} \mathbf{p}^{(i)} > 0$, 则迭代满足

$$\|\mathbf{x}^{(i)}\|_2 < \|\mathbf{x}^{(i+1)}\|_2, \quad 0 \leq i \leq k-1$$

证明 因为 $\|\mathbf{x}^{(i+1)}\|_2^2 = \|\mathbf{x}^{(i)}\|_2^2 + 2\alpha_i \mathbf{x}^{(i)T} \mathbf{p}^{(i)} + \alpha_i^2 \|\mathbf{p}^{(i)}\|_2^2$. 由条件 $\mathbf{p}^{(i)T} \mathbf{B} \mathbf{p}^{(i)} > 0$ 知 $\alpha_i > 0$, 故仅

需证明 $\mathbf{x}^{(i)}^T \mathbf{p}^{(i)} \geq 0$ 即可. 采用数学归纳法. 当 $i=0$ 时, $\mathbf{x}^{(0)}^T \mathbf{p}^{(0)}=0$, 结论成立. 假设结论对 i 成立, 则

$$\begin{aligned}\mathbf{x}^{(i+1)}^T \mathbf{p}^{(i+1)} &= \mathbf{x}^{(i+1)}^T (-\mathbf{r}^{(i+1)} + \beta_{i+1} \mathbf{p}^{(i)}) \\ &= -\mathbf{x}^{(i+1)}^T \mathbf{r}^{(i+1)} + \beta_{i+1} \mathbf{x}^{(i+1)}^T \mathbf{p}^{(i)} \\ &= -(\alpha_0 \mathbf{p}^{(0)} + \alpha_1 \mathbf{p}^{(1)} + \cdots + \alpha_i \mathbf{p}^{(i)})^T \mathbf{r}^{(i+1)} + \beta_i (\mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)})^T \mathbf{p}^{(i)} \\ &= \beta_i \mathbf{x}^{(i)}^T \mathbf{p}^{(i)} + \alpha_i \beta_i \|\mathbf{p}^{(i)}\|_2^2 \\ &\geq 0\end{aligned}$$

其中第 4 个等式利用了共轭梯度法的重要性质, 即梯度与以前的搜索方向是正交的; 最后一个不等式利用了归纳假设. 由归纳法知结论成立. ■

共轭梯度法每次产生的近似解的范数在增大, 因此如果迭代满足 $\|\mathbf{x}^{(i+1)}\|_2 > \Delta$, 则信赖域子问题的解必位于信赖域的边界上, 即 $\|s^*\|^2 = \Delta$. 这表明应用算法 6.3.1 时, 如果 i 满足 $\mathbf{p}^{(i)}^T \mathbf{B} \mathbf{p}^{(i)} \leq 0$ (蕴含着 $q(s)$ 沿着 $\mathbf{p}^{(i)}$ 是无界的), 或者 $\|\mathbf{x}^{(i)} + \alpha_i \mathbf{p}^{(i)}\|_2 > \Delta$ (蕴含着解必在信赖域的边界上), 则算法应该在第 i 次迭代后终止. 在这两种情况下, 最简单的终止策略是停止在边界上的点 $s' = \mathbf{x}^{(i)} + \alpha_B \mathbf{p}^{(i)}$, 其中 α_B 是二次方程

$$\|\mathbf{x}^{(i)} + \alpha \mathbf{p}^{(i)}\|_2^2 = \Delta^2$$

的根. 重要的是这个 s' 满足条件(6.3.1). 这样, 由 6.4 节中将介绍的定理 6.4.1 知整个信赖域法收敛. 综上所述, 可得 Steihaug 共轭梯度法的伪码, 即算法 6.3.2.

Algorithm 6.3.2 Steihaug's conjugate gradient method for problem(6.2.1)

```

1: Given  $\epsilon > 0$ ; set  $\mathbf{x}^{(0)} = \mathbf{0}, \mathbf{r}^{(0)} = \mathbf{g}, \mathbf{p}^{(0)} = -\mathbf{g}$ ;
2: if  $\|\mathbf{r}^{(0)}\|_2 < \epsilon$  then
3:   return  $s' = \mathbf{x}^{(0)}$ .
4: end if
5: for  $j=0, 1, 2, \dots$  do
6:   if  $\mathbf{p}^{(j)}^T \mathbf{B} \mathbf{p}^{(j)} \leq 0$  then
7:     find  $\tau$  such that  $s' = \mathbf{x}^{(j)} + \tau \mathbf{p}^{(j)}$  minimizes  $q(s)$  in problem(6.2.1) and satisfies  $\|s'\|_2 = \Delta$ ;
8:     return  $s'$ .
9:   end if
10:  set  $\alpha_j = -\frac{\mathbf{r}^{(j)}^T \mathbf{r}^{(j)}}{\mathbf{p}^{(j)}^T \mathbf{B} \mathbf{p}^{(j)}}$ ;
11:  set  $\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \alpha_j \mathbf{p}^{(j)}$ ;
12:  if  $\|\mathbf{x}^{(j+1)}\|_2 \geq \Delta$  then
13:    find  $\tau \geq 0$  such that  $s' = \mathbf{x}^{(j)} + \tau \mathbf{p}^{(j)}$  satisfies  $\|s'\| = \Delta$ ;
14:    return  $s'$ .
15:  end if
16:  set  $\mathbf{r}^{(j+1)} = \mathbf{r}^{(j)} + \alpha_j \mathbf{B} \mathbf{p}^{(j)}$ ;
17:  if  $\|\mathbf{r}^{(j+1)}\| < \epsilon \|\mathbf{r}^{(0)}\|$  then
18:    return  $s' = \mathbf{x}^{(j+1)}$ .
19:  end if
20:  set  $\beta_{j+1} = \frac{\mathbf{r}^{(j+1)}^T \mathbf{r}^{(j+1)}}{\mathbf{r}^{(j)}^T \mathbf{r}^{(j)}}$ ;
21:  set  $\mathbf{p}^{(j+1)} = -\mathbf{r}^{(j+1)} + \beta_{j+1} \mathbf{p}^{(j)}$ ;
22: end for

```

这种截断共轭梯度法到底有多好呢? 在凸的情况下,由袁亚湘证明^[29]的定理 6.3.3 表明该方法非常好,即由截断共轭梯度法所得到的目标函数的下降量至少为最优下降量的一半. 然而在非凸($\mathbf{B}^{(k)}$ 不定)情况下,该方法有可能相当差. 例如,若 $\mathbf{g}=\mathbf{0}$ 且 \mathbf{B} 是不定的,则截断共轭梯度法将在 $s=\mathbf{0}$ 处终止,而真正的解位于信赖域的边界. 鉴于非凸情况时的复杂性,这里不再进一步讨论.

定理 6.3.3 假设应用算法 6.3.2 求解问题(6.2.1),且 $\mathbf{B}^{(k)}$ 是正定的. 记问题的精确解为 s^* , 算法所得解为 $s^{(k)}$, 则它们满足

$$\frac{1}{2}(f^{(k)} - q^{(k)}(s^*)) \leq f^{(k)} - q^{(k)}(s^{(k)}) \leq f^{(k)} - q^{(k)}(s^*)$$

6.4 实用信赖域法

尽管信赖域法的实际效果很好,但还有一些问题需要进一步探讨. 一是当所确定的范数没有反映出问题本身的度量时,该方法可能会很慢,从而实用算法需要对变量进行调整;二是实用算法通常仅能得到一个近似解;三是当得不到目标函数的二阶导数 $\mathbf{G}^{(k)}$ 时,也只能使用更一般化的模型函数 $q^{(k)}(s) = f^{(k)} + \mathbf{g}^{(k)T} s + \frac{1}{2} s^T \mathbf{B}^{(k)} s$. 综上所述,得实用信赖域法的框架,即算法 6.4.1.

Algorithm 6.4.1 A framework of practical trust region method

```

1: Choose  $0 \leq \eta_s < \eta_v < 1$ ,  $\gamma_i \geq 1$ ,  $0 \leq \gamma_d \leq 1$ ,  $\Delta_0 > 0$  and  $\mathbf{x}^{(0)}$ ;
2: set  $k = 0$ ;
3: repeat
4:   build the second-order model  $q^{(k)}(s)$  of  $f(\mathbf{x}^{(k)} + s)$ ;
5:   solve the trust-region subproblem problem(6.2.1) to find  $s^{(k)}$  satisfying condition(6.3.1);
6:   calculate  $\rho_k$  with formula(6.1.2);
7:   if  $\rho_k \geq \eta_v$  then
8:     set  $\Delta_{k+1} = \gamma_i \Delta_k$ ;
9:   else
10:    if  $\rho_k \geq \eta_s$  then
11:      set  $\Delta_{k+1} = \Delta_k$ ;
12:    else
13:      set  $\Delta_{k+1} = \gamma_d \Delta_k$ ;
14:    end if
15:  end if
16:  if  $\rho_k \leq 0$  then
17:    set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ ;
18:  else
19:    set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + s^{(k)}$ ;
20:  end if
21:  set  $k = k + 1$ ;
22: until convergence

```

算法 6.4.1 中参数的一些典型取值是: $\gamma_c = 0.9$ 或者 0.75 , $\gamma_i = 0.1$ 或者 0.25 , $\gamma_r = 2$, $\gamma_d = 0.5$. 在实践中,甚至可以让这些参数在不同的迭代之间变化,当然要在合理的限制内. 特别地,在一次非常成功的迭代后,除非有 $\|s^{(k)}\| \approx \Delta_k$ 才会增加信赖域半径,否则并无此必要,比如步 $\|s^{(k)}\|$ 比信赖域半径的一半还小,就不用再增加信赖域半径. 初始信赖域半径的确定有一些经验可以利用. 实践中,如果问题变量的数量级基本相同,则取 $\Delta_0 = O(1)$ 会比较合理. 设信赖域半径的上界为 $\hat{\Delta}$,表 6.4.1 给出了信赖域法中信赖域半径及迭代点基于 ρ_k 的更新方式.

表 6.4.1 信赖域法中信赖域半径及迭代点基于 ρ_k 的更新方式

ρ_k 的情况	Δ_{k+1}	$x^{(k+1)}$
$\rho_k < \eta_c$	$\gamma_d \Delta_k$	$x^{(k)}$
$\rho_k \in [\eta_c, \eta_r)$	Δ_k	$x^{(k)} + s^{(k)}$
$\rho_k > \eta_r$, $\ s^{(k)}\ \ll \Delta_k$	Δ_k	$x^{(k)} + s^{(k)}$
$\rho_k > \eta_r$, $\ s^{(k)}\ \approx \Delta_k$	$\min(\gamma_i \Delta_k, \hat{\Delta})$	$x^{(k)} + s^{(k)}$

信赖域法的优点之一是,它具有非常强的大范围收敛性结论,且对需要求解的问题没有太强的限制. 我们不加证明地给出实用信赖域法的收敛性,需要强调的是,定理 6.3.1 的推论在实用信赖域法的收敛性分析中起着最基本的作用.

定理 6.4.1 (大范围收敛) 假设 $f \in C^2$,且 $\mathbf{B}^{(k)}$ 和 $\mathbf{G}(x)$ 一致有界,即存在非负常数 h, b 满足

$$\|\mathbf{B}^{(k)}\| \leq h, \quad \forall k; \quad \|\mathbf{G}(x)\| \leq b, \quad \forall x$$

则算法 6.4.1 产生的序列 $\{x^{(k)}\}$ 满足对某 l 有 $\mathbf{g}^{(l)} = \mathbf{0}$,或者 $\lim_{k \rightarrow \infty} f^{(k)} = -\infty$,或者 $\lim_{k \rightarrow \infty} \|\mathbf{g}^{(k)}\| = 0$.

这样,由算法 6.4.1 所产生的梯度序列 $\{\mathbf{g}^{(k)}\}$ 收敛到零或者最后全是零,此结果已经非常令人满意,但这并不意味着序列 $\{x^{(k)}\}$ 是收敛的,如果它收敛,则极限是一阶临界(critical)点. 关于上述算法的更强的结论是收敛到二阶临界点. 总而言之,信赖域法有非常丰富的收敛性结论.

6.5 评注与参考

信赖域法最早可以追溯到 LM 法(见算法 5.4.1),LM 法本意是通过引入参数 λ 来避免 Jacobi 矩阵病态的情况,即防止确定的步太大. 具体地,LM 法对 Hessian(近似)阵 $\mathbf{G}^{(k)}$ 进行单参数变形 $\mathbf{G}^{(k)} + \lambda \mathbf{I}$,其中 λ 的大小间接决定 $s^{(k)}$ 的长短. 与之相比,信赖域算法更直接地对 s 的长度施加精确限制—— $\|s\| \leq \Delta$. Powell 于 1970 年第一次提出了信赖域法,并给出算法的一个详细的求解框架,其中给出的信赖域半径的更新策略一直沿用至今. Powell 证明算法具有大范围收敛性,并且具有很好的稳定性和收敛速度.

习题 6

6.1 设 $f(\mathbf{x}) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$,并假定信赖子问题的二次模型函数中取 $\mathbf{B}^{(k)} = \mathbf{G}^{(k)}$.

- (a) 完整写出信赖域法在点 $\mathbf{x}^{(k)} = (0, -1)^T$ 的二次子问题, 并画出子问题目标函数的等值线.
- (b) 针对该子问题, 画出信赖域半径从 $\Delta = 0$ 变到 $\Delta = 2$ 时, 信赖域子问题解族的示意图.
- (c) 对 $\Delta = 1$, 求出该信赖域子问题的柯西点.
- (d) 对于点 $\mathbf{x}^{(k)} = (0, 0.5)^T$, 重复上面的工作.
- 6.2 假设在信赖域子问题(6.2.1)中取 $\mathbf{B} = \nu \mathbf{I}$, 其中 ν 为参数. 记问题(6.2.1)的解为 \mathbf{s}^* .
- (a) 写出 $\nu = 0$ 时 \mathbf{s}^* 的表达式.
- (b) 写出 $\nu < 0$ 时信赖域子问题的柯西点 \mathbf{s}_c 的表达式.
- (c) 画出 $\nu > 0$ 时 $\frac{q(\mathbf{s}_c)}{q(\mathbf{s}^*)}$ 随信赖域半径 Δ 变化的图像.
- 6.3 编写程序实现 dog-leg 法. 选取 $\mathbf{B}^{(k)}$ 为精确 Hessian 阵. 运用其极小化 Rosenbrock 函数. 对信赖域法的更新规则进行实验, 即改变算法中的参数, 或者设计你自己的规则.
- 6.4 编写基于 Steihaug 共轭梯度法的信赖域法的程序. 选取 $\mathbf{B}^{(k)}$ 为精确 Hessian 阵, 利用其极小化函数
- $$f(\mathbf{x}) = \sum_{i=1}^n [(1 - x_{2i-1})^2 + 10(x_{2i} - x_{2i-1}^2)^2]$$
- 取 $n=10$, 并对初始点和共轭梯度法迭代的停止准则进行实验. 用 $n=50$ 重复计算. 你的程序应该显示出: 在每一次迭代, Steihaug 共轭梯度法的终止情况, 即遇到非正曲率, 或者达到信赖域的边界, 或者满足停止测试.
- 6.5 证明如果用信赖域原型算法求解正定二次函数, 则对所有的 k 有 $\rho_k = 1, \Delta_{k+1} = 2\Delta_k$, 且在有限步内能得到问题的最优解.
- 6.6 称问题(6.1.1)中取 l_∞ 范数时的解集 $\{\mathbf{s}(\Delta) : \Delta > 0\}$ 为超立方体轨道. 考虑当一组固定的上下界约束 $s_i \leq \Delta$ 或 $s_i \geq -\Delta$ 是积极约束时可能出现的情况, 指出该轨道由一组直线段组成. 若对所有的 i 有 $g_i^{(k)} \neq 0$, 证明对于小的 Δ 值, 该轨道为直线段 $\mathbf{x}^{(k)} + \alpha \mathbf{s}$, 其中 $s_i = -\text{sign}(g_i^{(k)})$.
- 6.7 确定函数 $f(\mathbf{x}) = (x_2^2 - x_1^2)/2$ 在点 $\mathbf{x}^{(0)} = (1/2, -1)^T$ 处的超立方体轨道. 证明该轨道由两部分组成: 当 $\Delta \leq 1$ 时连接 $\mathbf{x}^{(0)}$ 与 $(3/2, 0)^T$ 的直线段和当 $\Delta \geq 1$ 时的射线 $(3/2, 0)^T + (\Delta - 1)\mathbf{e}_1$. 再证明当 $\Delta > 1/2$ 时, 存在另一条局部极小点的轨道也是一条折线.
- 6.8 确定函数 $f(\mathbf{x}) = x_1 x_2$ 在点 $\mathbf{x}^{(0)} = (1/2, 1)^T$ 处的超立方体轨道. 证明其全局最优解的轨道在 $\Delta = 1/2$ 处不连续.
- 6.9 考虑二元函数 $f(\mathbf{x}) = x_1^4 + x_1 x_2 + (1 + x_2)^2$, 取初始点 $\mathbf{x}^{(0)} = \mathbf{0}$.
- (a) 确定使 $\mathbf{G}^{(0)} + \lambda \mathbf{I}$ 正定的 λ 的取值范围.
- (b) 计算 $\lambda = 1$ 时由式(6.2.2)所确定的修正项, 并验证 $f(\mathbf{x})$ 的值是减小的.
- (c) 验证使 $f(\mathbf{x})$ 的值减小的 λ 的范围为 $\lambda \geq 0.9$, 且在 $\lambda = 1.2$ (近似的) 处下降量最大.
- (d) 确定式(6.1.2)中 ρ_0 的值.
- 6.10 对习题 5.2 中的二次函数, 确定其在点 $\mathbf{x}^{(0)} = \mathbf{0}$ 处的 LM 轨道. 说明 6.2.2 小节中所说的情况(i)存在; 且对任意的 $\lambda \in (0, +\infty)$, 该轨道可表示为
- $$\mathbf{s}(\lambda) = (-\lambda, 2 + \lambda)^T / (\lambda^2/2 + 3\lambda + 2) \quad (a)$$

绘出轨道的图像,验证初始方向为下降方向且 $\|s(\lambda)\|_2$ 关于 λ 单调增加. Δ 取何值时约束 $\|s(\lambda)\|_2 \leq \Delta$ 为非积极约束? 分 $\lambda < -3 - \sqrt{5}$, $\lambda \in (-3 - \sqrt{5}, \sqrt{5} - 3)$ 和 $\lambda \in (\sqrt{5} - 3, 0)$ 三种情况分别给出式(a)的解释.

- 6.11 确定函数 $f(\mathbf{x}) = x_1 x_2$ 在点 $\mathbf{x}^{(0)} = (1, 1/2)^T$ 处的 LM 轨道. 说明 6.2.2 小节中所说的情况(ii)存在;且对任意的 $\lambda \in (1, \infty)$, 该轨道可表示为

$$s(\lambda) = 0.5(2 - \lambda, 1 - 2\lambda)^T / (\lambda^2 - 1) \quad (b)$$

确定 λ 使得 $\Delta = \|s(\lambda)\| = 1/2$ 的值,说明除了由式(b)给出的全局解的轨道外,在区间 $(0.350\ 667, 1)$ 内还存在另一局部解的轨道,其中 $0.350\ 667$ 是方程 $5\lambda^3 - 12\lambda^2 + 15\lambda - 4 = 0$ 的唯一根. 对 $\Delta = 1$ 找出一个这样的局部解. 使局部解不存在的最大的 Δ 值是多少? 对于 $\lambda \in (0, 0.350\ 667)$, $\lambda \in (-1, 0)$, $\lambda \in (-\infty, -1)$ 的情形分别对式(b)进行解释.

- 6.12 确定函数 $f(\mathbf{x}) = x_1 x_2$ 在点 $\mathbf{x}^{(0)} = (1, 1)^T$ 处的 LM 轨道,说明 6.2.2 小节中所说的情况(iii)存在,其中 $\bar{\Delta} = \sqrt{2}/2$. 对 $\Delta \leq \sqrt{2}/2$,给出 s 的关于 λ 的表达式. 说明 $\Delta \geq \sqrt{2}/2$ 时轨道可以表示为 $s = \bar{s} \pm (2\Delta^2 - 1)(1/2, -1/2)^T$.

- 6.13 Chebyquad 问题($n=2$)是以 $\mathbf{x}^{(0)} = (1/3, 2/3)^T$ 为初始点来极小化函数 $f(\mathbf{x}) = (x_1 + x_2 - 1)^2 + [(2x_1 - 1)^2 + (2x_2 - 1)^2 - 2/3]^2$. 说明由函数的二阶 Taylor 近似得到的 LM 轨道会出现 6.2.2 小节中所说的情况(iii),其中临界值 $\lambda = -\lambda_n = 28/9$, $\bar{s} = (8, -8)^T/21$.

- 6.14 应用 Hebden 迭代算法 6.2.1 求解习题 6.10,其中 $\Delta = 1/2$, $\lambda^{(0)} = 3$. 说明迭代依次为 $\lambda^{(1)} = 1.890\ 8$, $\lambda^{(2)} = 1.996\ 7$,且对 $k=1, 2, 3$ 分别有 $\|s^{(k)}\| = 0.318\ 7, 0.540\ 5, 0.501\ 0$. 由此验证该方法收敛很快.

第 7 章 约束优化：理论

本书的后半部分考虑约束优化问题,它的一般表述为

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} \\ & \quad c_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{I} \end{aligned} \tag{7.0.1}$$

其中 f 和 c_i 都是光滑函数, \mathcal{E} 和 \mathcal{I} 是两个有限指标集. 称 f 是目标函数, $c_i (i \in \mathcal{E})$ 是等式(equality)约束, $c_i (i \in \mathcal{I})$ 是不等式(inequality)约束. 定义可行域(feasible region)是所有满足约束的点 \mathbf{x} 所组成的集合,即

$$\Omega := \{\mathbf{x} \in \mathbb{R}^n : c_i(\mathbf{x}) = 0, i \in \mathcal{E}; c_i(\mathbf{x}) \leq 0, i \in \mathcal{I}\}$$

这样,问题(7.0.1)可以更紧凑地表示为 $\min_{\mathbf{x} \in \Omega} f(\mathbf{x})$.

7.1 概述

大多数实用方法都需要假设目标函数和约束函数是光滑的,比如一阶或者二阶导数存在且连续($f, c_i \in C^1$ 或者 C^2). 同第1章一样,用记号 $\nabla f (= \mathbf{g})$ 和 $\nabla^2 f (= \mathbf{G})$ 表示 f 的梯度向量和Hessian阵. 用记号 ∇c_i 和 $\nabla^2 c_i$ 表示约束 c_i 相应的一阶和二阶导数. 也将向量 ∇c_i 记为 \mathbf{a}_i , 称其为约束 c_i 的法向量(normal vector). 有时也以这些向量为列形成 Jacobi 矩阵(注意这与线性规划中的表示相反,那里法向量是 \mathbf{A} 的行向量). 向量 \mathbf{a}'_i (即 $\mathbf{a}_i(\mathbf{x})$ 在 $\mathbf{x} = \mathbf{x}'$ 处的值)是 $c_i(\mathbf{x})$ 在 \mathbf{x}' 增加最快的方向. 对 $i \in \mathcal{I}$, 如果 $c'_i = 0$, 则该向量指向约束不可行的一边,且与零值等值线成直角,见图 7.1.1. 后面的大部分内容需要利用这些导数,从而假设它们是存在的. 大多数情况下需要一阶导数的公式,有些情况下也需要二阶导数的公式. 如果没有导数可用,则可以用有限差分来估计这些导数,当然,这样做可能会牺牲算法的稳健性和有效性.

另一个很重要的概念是积极(active)约束或紧(binding)约束. 指标集

$$\mathcal{A}' = \mathcal{A}(\mathbf{x}') = \{i : c_i(\mathbf{x}') = 0\} \tag{7.1.1}$$

定义了点 \mathbf{x}' 处的积极约束. 如果 \mathbf{x}' 在可行域的边界上,则确定对应边界的约束是积极的. 如果 \mathbf{x}' 可行,则显然有 $\mathcal{E} \subset \mathcal{A}'$. 式(7.0.1)的解 \mathbf{x}' 处的积极约束特别重要. 如果事先知道这个集合,忽略其余的非积极约束,而将积极集 \mathcal{A}' 中的作为等式约束,则 \mathbf{x}' 为这个新问题的局部解. 注意,对 $i \notin \mathcal{A}'$ 的约束进行微小的扰动并不影响 \mathbf{x}' 的最优性,但这个事实对积极约束通常是不成立的. 请读者完成习题 7.20,这将会有对积极约束有更深刻的理解.

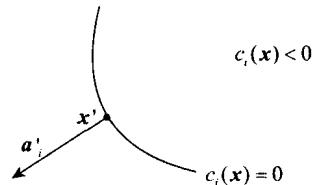


图 7.1.1 法向量

例 7.1.1 (积极约束) 考虑

$$\begin{aligned}
 & \text{minimize} \quad f(\mathbf{x}) = -x_1 - x_2 \\
 & \text{subject to} \quad c_1(\mathbf{x}) = x_1^2 - x_2 \leq 0 \\
 & \quad c_2(\mathbf{x}) = x_1^2 + x_2^2 - 1 \leq 0
 \end{aligned} \tag{7.1.2}$$

由图 7.1.2 知, 解 $\mathbf{x}^* = (1/\sqrt{2}, 1/\sqrt{2})^\top$. 这样积极集 \mathcal{A}^* $= \{2\}$, 圆约束 $c_2(\mathbf{x})$ 是积极的. 抛物线约束 $c_1(\mathbf{x})$ 是非积极的, 对其进行扰动或者去掉它均不会改变解 \mathbf{x}^* . 积极约束的定义可进一步精细化为强积极和弱积极约束, 详见图 7.2.2.

求解问题(7.0.1)的方法通常是迭代法, 即从给定点 $\mathbf{x}^{(0)}$ 开始, 产生序列 $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$, 希望其收敛到 \mathbf{x}^* . 如果 \mathbf{x}^* 是序列的一员, 则称该方法具有有限终止性.

依据约束函数的特点可将约束优化分成特色各异的两个部分, 即线性约束 (linearly constrained) 规划和非线性约束 (nonlinearly constrained) 规划. 在线性约束规划中, 每个约束是线性函数 $c_i(\mathbf{x}) = \mathbf{a}_i^\top \mathbf{x} - b_i$ (或称仿射函数). 这时可行域的边界是超平面, 法向量 ∇c_i 是常量, 即向量 \mathbf{a}_i . 结合消元法和积极集法可以处理线性约束, 且迭代 $\mathbf{x}^{(k)}$ 总是可行的. 最简单的情况是目标函数是线性或者二次的 (分别对应第 2 章的线性规划和第 8 章的二次规划), 这两种情况都存在具有有限终止性的算法. 第 8 章将这种方法推广到一般的目标函数, 可以借鉴无约束优化的很多内容, 比如拟牛顿法、高斯-牛顿法、信赖域法和使用有限差分近似的无导数方法等. 其他的如一维搜索和测验终止条件等也是相同的.

光滑约束极小化问题中最难的是非线性约束规划 (第 9 章), 其问题中有某些非线性的约束函数, 尚未有完全满意的方法. 首先想到的是消元法. 如果可以直接消元, 则重新整理后可以由非线性约束给出因变量的显式表达式, 代入目标函数得无约束的既约问题. 但是符合这种情况的问题极少. 也可以利用数值的方法解方程组来进行间接消元, 实验表明这种做法的效率通常很低, 且伴随着其他的困难. 与消元法的思想密切相关的是可行方向法 (feasible direction method). 另一种与之不同的方法是, 尝试使用罚函数将问题转化成无约束极小化 (见 9.1~9.3 节). 通常也需要借助于某些特殊的罚函数来得到大范围收敛性. 此外, 对原问题进行恰当的建模也能设计有效的方法, 特别是约束函数的线性化, 即

$$c_i(\mathbf{x}^{(k)} + \mathbf{s}) \approx l_i^{(k)}(\mathbf{s}) = c_i^{(k)} + \mathbf{a}_i^{(k)^\top} \mathbf{s} \tag{7.1.3}$$

其中线性化的函数 $l_i^{(k)}$ 是由 $\mathbf{x}^{(k)}$ 的校正 \mathbf{s} 定义的, 是当前迭代 $\mathbf{x}^{(k)}$ 处的一阶 Taylor 多项式. 这种近似使得每次迭代需要求解的子问题都是线性约束的. 同无约束极小化一样, 也可以考虑目标函数的二次模型. 进一步将约束曲率考虑进来, 并用一种恰当的方式来修正二次模型中的二次项能得到更好的效果 (见 9.4 节).

这些算法, 特别是非线性约束规划的算法依赖于问题(7.0.1)的最优化条件的研究. 总体来说, 约束极小化比无约束问题要复杂得多, 读者最好能较深刻地理解这些结论, 特别是 Lagrange 乘子和一阶条件. 为此, 先在 7.2 节中对这些内容进行简单的介绍, 目的是让大家看到 Lagrange 乘子是如何产生的, 以及如何对它们进行解释, 并用例子说明一阶条件. 紧接着

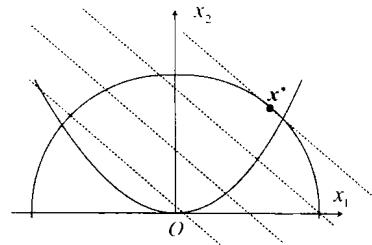


图 7.1.2 积极约束和非积极约束

在 7.3 节中给出严谨的叙述和论证, 7.4 节用同样的处理方式讨论二阶最优化条件. 7.5 节简单地讨论凸规划, 7.6 节进一步讨论凸规划和 Lagrange 乘子. 7.7 节讨论对偶. 7.8 节介绍 20 世纪 90 年代发展起来的一类重要的凸规划——半定规划, 以及半定规划的若干应用.

7.2 Lagrange 乘子

有关约束最优化的理论已有不少的著作, 在本章只叙述一些最重要的结论, 叙述的方式力求实际并避免过分的一般性. 本节需要引入一个重要的概念, 即 Lagrange 乘子. 为了论述清晰, 这一节先采用不十分严密的叙述方式, 下节再对有关内容进行严密的论证.

第 4 章已经介绍过稳定点(即使得 $\mathbf{g}(\mathbf{x})=\mathbf{0}$ 的点)的概念是无约束优化的基础, 并给出了局部解的必要条件. 当为约束优化问题(7.0.1)的最优解确定类似的必要条件时, 会用到 Lagrange 乘子.

对于第 4 章的无约束优化问题, 必要条件是目标函数在点 \mathbf{x}^* 处沿任何方向的斜率为零, 曲率非负, 即在点 \mathbf{x}^* 处无任何下降方向. 对约束最优化问题, 可行域的限制使问题变得复杂. 因此一个局部最优解首先必须是一个可行点, 此外还要求点 \mathbf{x}^* 处无任何可行的下降方向. 然而, 当可行域的边界是弯曲的时候, 会出现一些困难. 为了简单起见, 首先讨论只有等式约束(即 $\mathcal{I}=\emptyset$)的情况.

设 \mathbf{s} 是极小点 \mathbf{x}^* 处的可行增量, 利用 Taylor 展式, 得

$$c_i(\mathbf{x}^* + \mathbf{s}) = c_i^* + \mathbf{s}^\top \mathbf{a}_i^* + o(\|\mathbf{s}\|)$$

其中 $\mathbf{a}_i^* = \nabla c_i(\mathbf{x}^*)$, 而 $o(\cdot)$ 表示在极限情况下相对于 \mathbf{s} 可忽略不计的项. 由可行性有 $c_i(\mathbf{x}^* + \mathbf{s}) = c_i^* = 0$, 因此任何可行增量均会确定一个满足

$$\mathbf{p}^\top \mathbf{a}_i^* = 0, \quad \forall i \in \mathcal{E} \quad (7.2.1)$$

的可行方向 \mathbf{p} . 在正则情况下(比如对任意 $i \in \mathcal{E}$, 向量 \mathbf{a}_i^* 线性无关), 任意给一个这样的方向 \mathbf{p} , 均能构造可行增量 \mathbf{s} (见 7.3 节). 如果 $f(\mathbf{x})$ 沿着 \mathbf{p} 的斜率是负的, 即

$$\mathbf{p}^\top \mathbf{g}^* < 0 \quad (7.2.2)$$

那么方向 \mathbf{p} 是下降方向, 从而沿 \mathbf{p} 的可行增量 \mathbf{s} 会使 $f(\mathbf{x})$ 减小. 然而, \mathbf{x}^* 是局部极小点, 不可能出现这种情况, 因此不存在既满足式(7.2.1)、又满足式(7.2.2)的方向 \mathbf{p} . 保证该事实成立的一个充分条件是 $-\mathbf{g}^*$ 是向量 \mathbf{a}_i^* ($i \in \mathcal{E}$) 的线性组合, 即

$$-\mathbf{g}^* = \sum_{i \in \mathcal{E}} \mathbf{a}_i^* \lambda_i^* = \mathbf{A}^* \boldsymbol{\lambda}^* \quad (7.2.3)$$

正如下面要指出的, 该条件也是必要的, 所以式(7.2.3)是局部极小点的必要条件. 线性组合中的乘子 λ_i^* 称作 Lagrange 乘子(Lagrangian multipliers), 而上标“ $*$ ”表明它们与最优点 \mathbf{x}^* 对应. \mathbf{A}^* 表示以 \mathbf{a}_i^* ($i \in \mathcal{E}$) 为列的矩阵. 需要指出的是, 每个约束均有一个乘子与之对应, 如果 \mathbf{A}^* 满秩, 则 $\boldsymbol{\lambda}^*$ 可(由式(7.2.3))唯一地定义为

$$\boldsymbol{\lambda}^* = -\mathbf{A}^{*+} \mathbf{g}^*$$

其中 $\mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ 表示 \mathbf{A} 的广义逆(见习题 7.8). 当没有约束时, 式(7.2.3)可简化为通常的稳定点条件 $\mathbf{g}^* = \mathbf{0}$.

通常用反证法证明式(7.2.3)是必要条件. 如果式(7.2.3)不成立, 则 $-\mathbf{g}^*$ 可表示为

$$-\mathbf{g}^* = \mathbf{A}^* \boldsymbol{\lambda}^* + \boldsymbol{\mu} \quad (7.2.4)$$

其中 $\boldsymbol{\mu} \neq \mathbf{0}$ 是 $-\mathbf{g}^*$ 中与所有 \mathbf{a}^* 正交的部分, 因而有 $\mathbf{A}^{*\top} \boldsymbol{\mu} = \mathbf{0}$. 于是 $\mathbf{p} = \boldsymbol{\mu}$ 既满足式(7.2.1), 又满足式(7.2.2). 再由正则性条件, 沿 \mathbf{p} 存在可行增量 s 使得 $f(\mathbf{x})$ 减小. 这与 \mathbf{x}^* 是局部极小点矛盾, 因而条件(7.2.3)是必要的. 图 7.2.1 是该必要条件的图示. 在非局部极小点 \mathbf{x}' 处, $-\mathbf{g}' \neq \lambda \mathbf{a}'$, 因而存在正交于 \mathbf{a}' 的非零向量 $\boldsymbol{\mu}$, 沿着可行下降方向 $\mathbf{p} = \boldsymbol{\mu}$ 的增量 s 可使 $f(\mathbf{x})$ 减小. 在点 \mathbf{x}^* 处有 $-\mathbf{g}^* = \lambda^* \mathbf{a}^*$, 不存在任何可行的下降方向.

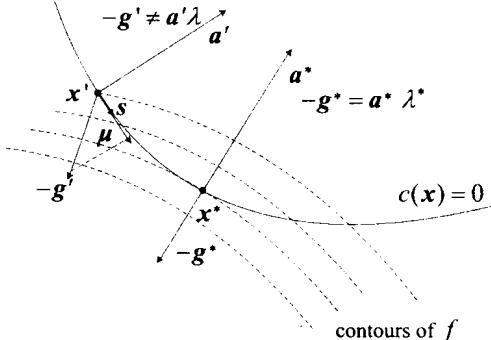


图 7.2.1 Lagrange 乘子的存在性

例 7.2.1 (一阶必要要件) 考虑

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) = x_1 + x_2 \\ \text{subject to} \quad & c(\mathbf{x}) = x_1^2 - x_2 = 0 \end{aligned}$$

解 $\mathbf{x}^* = (-1/2, 1/4)^\top$, 于是 $\mathbf{g}^* = (1, 1)^\top$, $\mathbf{a}^* = (-1, -1)^\top$. 取 $\lambda^* = 1$ 可使式(7.2.3)满足. 因为 \mathbf{a}^* 非零, 线性无关的正则性假设显然成立.

7.3 节将详细讨论正则性条件. 除此之外, 通过引入序列可行方向的概念, 可将上面采用的可行增量的表述严谨化, 当然也可借助可微弧的概念 (Kuhn 与 Tucker 于 1951 年提出). 然而, 本节的宗旨在于尽可能地避免这些技巧.

上述条件是求解等式约束极值问题的经典 Lagrange 乘子法, 即通过解方程组

$$\left. \begin{aligned} \mathbf{g}(\mathbf{x}) + \sum_{i \in \mathcal{E}} \mathbf{a}_i(\mathbf{x}) \lambda_i &= \mathbf{0} \\ c_i(\mathbf{x}) &= 0, \quad i \in \mathcal{E} \end{aligned} \right\} \quad (7.2.5)$$

得到解向量 \mathbf{x}^* 与 Lagrange 乘子 $\boldsymbol{\lambda}^*$, 即式(7.2.3)与可行性条件. 如果有 m 个等式约束, 则有 $m+n$ 个方程和 $m+n$ 个未知数 (\mathbf{x} 与 $\boldsymbol{\lambda}$), 因而方程组是适定的. 然而该方程组关于 \mathbf{x} 是非线性的, 除了某些特例外 (见第 8 章), 通常不容易求解. 另一个不太令人满意的地方是, 它没有考虑二阶导数的信息, 因而约束条件下的鞍点与极大值点也满足条件(7.2.5). 以例 7.2.1 为例来说明该方法. 此时式(7.2.5)中有 3 个方程, 即

$$\begin{aligned} 1 + 2\lambda x_1 &= 0 \\ 1 - \lambda &= 0 \\ x_1^2 - x_2 &= 0 \end{aligned}$$

其解为 $\lambda^* = 1$, $x_1^* = -1/2$, $x_2^* = 1/4$. 易见该法不同于直接消元法. 在直接消元法中, 将 $x_2 = x_1^2$ 代入 $f(\mathbf{x})$ 消去 x_2 , 得 $h(x_1) = x_1 + x_1^2$. 易见 $h(x_1)$ 的极小点 $x_1^* = -1/2$, 再回代得 $x_2^* =$

1/4.

陈述式(7.2.5)的另一个比较方便的做法是引入 Lagrange 函数

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_i \lambda_i c_i(\mathbf{x}) \quad (7.2.6)$$

这时式(7.2.5)可以简洁地表示为

$$\nabla \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0} \quad (7.2.7)$$

其中, $\nabla = \begin{bmatrix} \nabla_{\mathbf{x}} \\ \nabla_{\boldsymbol{\lambda}} \end{bmatrix}$ 是 $m+n$ 维变量空间中的一阶导算子. 于是局部极小点的一阶必要条件是,

$(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 是 Lagrange 函数的稳定点.

下面给出 Lagrange 乘子的一种工程解释. 为此, 考察约束右端有扰动的情况, 设

$$c_i(\mathbf{x}) = \epsilon_i, \quad i \in \mathcal{E} \quad (7.2.8)$$

扰动后问题的 Lagrange 函数为

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) = f(\mathbf{x}) + \sum_{i \in \mathcal{E}} \lambda_i (c_i(\mathbf{x}) - \epsilon_i)$$

这里, $\mathbf{x}(\boldsymbol{\epsilon})$, $\boldsymbol{\lambda}(\boldsymbol{\epsilon})$ 表示相应的解与乘子. 由式(7.2.8)有 $f(\mathbf{x}(\boldsymbol{\epsilon})) = \mathcal{L}(\mathbf{x}(\boldsymbol{\epsilon}), \boldsymbol{\lambda}(\boldsymbol{\epsilon}), \boldsymbol{\epsilon})$, 于是利用求导数的链式法则及式(7.2.7), 有

$$\frac{\partial f}{\partial \epsilon_i} \Big|_{\epsilon=0} = \frac{\partial \mathcal{L}}{\partial \epsilon_i} = \sum_{j=1}^n \frac{\partial \mathcal{L}}{\partial x_j} \frac{dx_j}{d \epsilon_i} + \sum_{k \in \mathcal{E}} \frac{\partial \mathcal{L}}{\partial \lambda_k} \frac{d \lambda_k}{d \epsilon_i} + \frac{\partial \mathcal{L}}{\partial \epsilon_i} = -\lambda_i^* \quad (7.2.9)$$

任一约束的 Lagrange 乘子的相反数均反映约束函数发生变化时所引起的最优值的变化率. 这一信息十分有用, 它指出了最优值对约束变化的敏感程度(见习题 7.4).

现在, 再讨论由不等式约束所引起的问题. 首先必须认识到的一点是: 只有点 \mathbf{x}^* 处的积极约束 \mathcal{A}^* 对最优性有影响, 用 $\mathcal{I}^* = \mathcal{I}(\mathbf{x}^*)$ 表示点 \mathbf{x}^* 处的不等式积极约束. 设 s 是可行增量, 对 $i \in \mathcal{I}^*$ 有 $c_i^* = 0$ 及 $c_i(\mathbf{x}^* + s) \leq 0$, 由可行增量 s 确定的可行方向 \mathbf{p} 除了满足式(7.2.1)外, 还应满足

$$\mathbf{p}^T \mathbf{a}_i^* \leq 0, \quad \forall i \in \mathcal{I}^* \quad (7.2.10)$$

这时, 不存在方向 \mathbf{p} 同时满足式(7.2.1)、式(7.2.2)与式(7.2.10)的充分条件是

$$\mathbf{g}^* + \sum_{i \in \mathcal{A}^*} \lambda_i^* \mathbf{a}_i^* = \mathbf{0} \quad (7.2.11)$$

与

$$\lambda_i^* \geq 0, \quad i \in \mathcal{I}^* \quad (7.2.12)$$

同时成立. 与式(7.2.3)相比, 这里唯一多出来的要求是, 不等式积极约束的乘子必须非负. 事实上, 正如下面所说明的, 仅要求这两个条件就足够了, 因此它们是局部极小点的必要条件. 在正则性假设(即假定法向量 \mathbf{a}_i^* , $i \in \mathcal{A}^*$ 线性无关)下, 用反证法同样也可证明这些条件成立. 设方程(7.2.11)成立, 但条件(7.2.12)不成立, 即存在某一乘子 $\lambda_p^* < 0$, 这时总能找到方向 \mathbf{p} 满足 $\mathbf{p}^T \mathbf{a}_i^* = 0, i \in \mathcal{A}^*, i \neq p$ 且 $\mathbf{p}^T \mathbf{a}_p^* = -1$ (例如 $\mathbf{p} = -\mathbf{A}^{++T} \mathbf{e}_p$, 其中 $\mathbf{A}^+ = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 表示 \mathbf{A} 的广义逆). 因此 \mathbf{p} 满足式(7.2.1)与式(7.2.10), 是一个可行方向. 同时, 由式(7.2.11)可得

$$\mathbf{p}^T \mathbf{g}^* = -\mathbf{p}^T \mathbf{a}_p^* \lambda_p^* = \lambda_p^* < 0 \quad (7.2.13)$$

因而 \mathbf{p} 也是下降方向. 再由正则性假设, 沿 \mathbf{p} 存在可行增量 s 使得 $f(\mathbf{x})$ 减小, 这与 \mathbf{x}^* 是局部极小点相矛盾, 因此条件(7.2.12)是必要的. 注意, 这里的证明利用向量 \mathbf{a}_i^* ($i \in \mathcal{A}^*$) 线性无关的性质构造广义逆. 更为一般的证明利用引理 7.3.4(Farkas 引理)及其推论, 并不需要这样

的假设.

还可以简单地从式(7.2.8)与式(7.2.9)推出条件(7.2.12). 若积极约束 $i \in \mathcal{I}^+$ 从 $c_i(x) \leq 0$ 扰动到 $c_i(x) \leq \epsilon_i$ (其中 $\epsilon_i > 0$), 即在 $x(\epsilon)$ 上引入一可行的改变量, 则 $f(x(\epsilon))$ 的值不应该增加. 这就意味着在局部最优解处有 $\partial f^*/\partial \epsilon_i \leq 0$, 即有 $\lambda_i^* \geq 0$, 因此式(7.2.12)给出的必要性有一个明显的解释. 为说明这些条件, 考虑问题(7.1.2). 如图 7.1.2 所示, 最优解 $x^* = (1/\sqrt{2}, 1/\sqrt{2})^T$, c_1 是非积极约束, $\mathcal{A}^* = \{2\}$. 于是 $g^* = (-1, -1)^T$, $a_2^* = (\sqrt{2}, \sqrt{2})^T$, 因而当 $\lambda_2^* = 1/\sqrt{2} \geq 0$ 时式(7.2.11)与式(7.2.12)满足.

需要强调的是, 必须将一般的不等式约束重新表述成 $c_i(x) \leq 0$ 的形式, 才能采用这里的 $\lambda_i \geq 0$ 的条件.

实际上, 上述在 $\lambda_p^* < 0$ 时可以构造出下降方向的过程还表明了与不等式约束相联系的 Lagrange 乘子的另一个重要性质. 条件 $p^T a_i^* = 0, i \neq p$ 与 $p^T a_p^* = -1$ 表明, 由此产生的可行增量 s 满足对 $i \neq p$ 有 $c_i(x^* + s) = 0$, 而 $c_p(x^* + s) < 0$. 这说明沿离开第 p 个约束边界的方向移动能减小 $f(x)$ 的值. 这个结果也可从式(7.2.9)得出, 而且在各种处理不等式约束的积极集法(见 8.2 节)中起重要作用, 该方法中如果条件(7.2.12)不满足, 则将 $\lambda_p^* < 0$ 对应的约束指标 p 从积极集中删去. 这一结论也可以用例 7.1.1 来说明. 考虑精确到 3 位小数的可行点 $x' = (0.786, 0.618)^T$, 在该点处两个约束均是积极的, 由 $g' = (-1, -1)^T$, $a_1' = (1.572, -1)^T$, $a_2' = (1.572, 1.236)^T$, 可知 $\lambda' = (-0.096, 0.732)^T$ 时式(7.2.11)成立, 但条件(7.2.12)并不满足, 因而 x' 不是局部极小点. 由于 $\lambda_1' < 0$, 沿离开第一个约束边界的方向移动能使目标函数下降, 即沿满足 $p^T a_1' = -1, p^T a_2' = 0$ 的方向移动, 比如 $p = (-0.352, 0.447)^T$, 它事实上是圆在点 x' 处的切线方向. 按此方向沿圆弧移动即得最优解 x^* , 在该点只有第二个约束是积极的.

可以用所有的约束而不仅仅是积极约束来表述条件(7.2.11)与条件(7.2.12). 为此, 将任何非积极约束的 Lagrange 乘子取为零, 再将式(7.2.11)、式(7.2.12)以及可行性条件结合起来, 得到下述定理.

定理 7.2.1 (一阶必要条件) 若 x^* 是问题(7.0.1)的局部极小点, 且在 x^* 处正则性假设(7.3.4)成立, 则存在 Lagrange 乘子 λ^* 满足

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = \mathbf{0} \quad (7.2.14a)$$

$$c_i(x^*) = 0, \quad i \in \mathcal{E} \quad (7.2.14b)$$

$$c_i(x^*) \leq 0, \quad i \in \mathcal{I} \quad (7.2.14c)$$

$$\lambda_i^* \geq 0, \quad i \in \mathcal{I} \quad (7.2.14d)$$

$$\lambda_i^* c_i(x^*) = 0, \quad i \in \mathcal{I} \quad (7.2.14e)$$

早些时候称这些条件是 Kuhn-Tucker (KT) 条件, 称满足这些条件的点 x^* 是 KT 点. 事实上 Karush 于 1939 年在未出版的硕士论文中就独立地推导出这些条件, 但真正被大家所了解的是 Kuhn 和 Tucker 描述的该条件^[15]. 现在大家已经倾向于称这些条件是 **KKT 条件**, 称满足这些条件的点是 **KKT 点**.

正则性假设(7.3.4)是向量 a_i^* ($i \in \mathcal{A}^*$) 线性无关条件的进一步放松, 在 7.3 节将对此进行详细讨论, 详见引理 7.3.3. 称条件(7.2.14e)为**互补条件**(complementarity condition), 它指出 λ_i^* 与 c_i^* 不可能同时非零, 或等价地说非积极约束的 Lagrange 乘子为 0. 如果不存在任何

i 使得 $\lambda_i^* = c_i^* = 0$, 则称严格(strict)互补性成立. 当 $\lambda_i^* = c_i^* = 0$ 时, 约束处于强积极与非积极之间. 图 7.2.2 给出各种情况的图示.

定理 7.2.1 的证明对理解约束优化问题很重要, 但是相当复杂. 因此在给出证明之前, 先用几个例子来说明 KKT 条件.

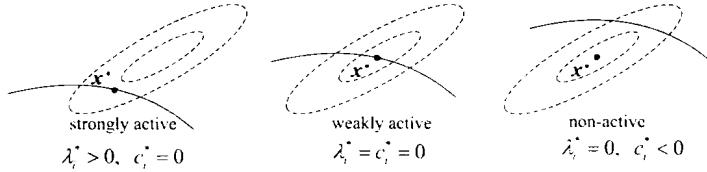


图 7.2.2 互补性

例 7.2.2 (单个等式约束) 考虑问题

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_1^2 + x_2^2 - 2 = 0 \end{aligned} \quad (7.2.15)$$

其有两个变量和一个等式约束, 见图 7.2.3(a). 由式(7.0.1)的描述, 有 $f(\mathbf{x}) = x_1 + x_2$, $\mathcal{I} = \emptyset$, $\mathcal{E} = \{1\}$, 且 $c_1(\mathbf{x}) = x_1^2 + x_2^2 - 2$. 易见该问题的可行域是中心在原点、半径为 $\sqrt{2}$ 的圆周. 解 $\mathbf{x}^* = (-1, -1)^T$. 从该圆上的任一其他点, 易找到一条前进轨线, 它在使 f 减小的同时保持可行, 即保持在圆上. 例如, 从点 $\mathbf{x} = (\sqrt{2}, 0)^T$ 围绕这个圆周沿顺时针方向移动, 既能保持可行, 又能使目标值减小. 从图 7.2.3(a)中也可以看到, 解 \mathbf{x}^* 处约束的法向量 \mathbf{a}_1^* 与目标函数的梯度向量 \mathbf{g}^* 是平行的, 即存在标量 $\lambda_1^* = 1/2$ 使得

$$-\mathbf{g}^* = \lambda_1^* \mathbf{a}_1^*, \quad \mathbf{a}_1^* = 2 \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \mathbf{g}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (7.2.16)$$

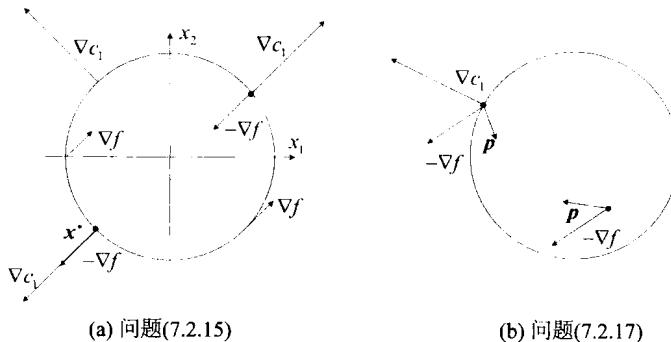


图 7.2.3 不同可行点处的约束函数和目标函数的梯度

例 7.2.3 (单个不等式约束) 稍微修正例 7.2.2, 即将其中的等式约束用不等式约束代替, 得

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_1^2 + x_2^2 - 2 \leq 0 \end{aligned} \quad (7.2.17)$$

可行域由问题(7.2.15)的圆周和它的内部组成(见图 7.2.3 (b)). 注意对圆周上的每个点, 约束法向量 \mathbf{a}_1 指向可行域的外部. 易见解仍是 $(-1, -1)^T$, 且式(7.2.16)对 $\lambda_1^* = 1/2$ 仍然成立. 然而, 这个不等式约束问题与例 7.2.2 中的问题(7.2.15)是不同的, 这里需要 Lagrange 乘子是非负的.

例 7.2.4 (两个不等式约束) 给问题(7.2.17)再加一个约束得到

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_1^2 + x_2^2 \leq 2, \quad x_2 \geq 0 \end{aligned} \quad (7.2.18)$$

可行域如图 7.2.4 中所示的半圆盘, 易见解 $x^* = (-\sqrt{2}, 0)^\top$, 该点处两个约束都是积极的. 在解 x^* 处, 有

$$\mathbf{g}^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{a}_1^* = \begin{bmatrix} -2\sqrt{2} \\ 0 \end{bmatrix}, \quad \mathbf{a}_2^* = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

因此, 当选取 $\lambda^* = \left(\frac{1}{2\sqrt{2}}, 1\right)^\top$ 时, 易验证 $\nabla_x \mathcal{L}(x^*, \lambda^*) = \mathbf{0}$. 需要注意的是, 这里 λ^* 的两个分量都是正的.

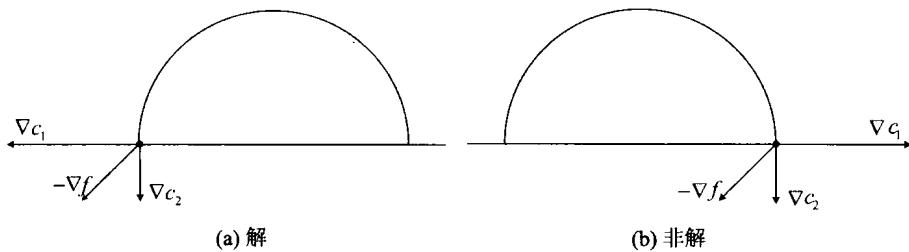


图 7.2.4 问题(7.2.18)在不同可行点处的约束函数和目标函数的梯度

现在考虑问题(7.2.18)的一些非解的可行点. 对于点 $x = (\sqrt{2}, 0)^\top$, 两个约束同样均是积极的. 在该点, 易见目标函数的梯度 \mathbf{g} 不再位于不等式组 $\mathbf{a}_i^\top \mathbf{p} \geq 0 (i=1,2)$ 的解集所对应的区域内, 见图 7.2.4(b).

最后来考虑点 $x = (1, 0)^\top$, 此时第二个约束是积极的. $c_1(x) < 0$, 首先由互补条件式(7.2.14e)有 $\lambda_1 = 0$. 因此, 在尝试满足 $\nabla_x \mathcal{L}(x, \lambda) = \mathbf{0}$ 时, 要寻找 λ_2 使得 $\mathbf{g} + \lambda_2 \mathbf{a}_2 = \mathbf{0}$. 由于不存在这样的 λ_2 , 故该点不满足 KKT 条件.

至此只考虑了一阶(由一阶导数表示的)条件. 还需要叙述二阶条件, 它给出了目标函数与约束函数在局部极小点处曲率的信息. 这方面的内容放在 7.4 节讨论. 当问题是凸规划时, 还能给出更强的结果, 见 7.5 节.

7.3 一阶条件

本节对 7.2 节的结论进行详细讨论. 首先给出可行增量的严格定义. 设 x' 是可行点, $\{x^{(k)}\}$ 是可行序列, 满足 $x^{(k)} \rightarrow x'$, 且对所有 k 有 $x^{(k)} \neq x'$, 则 $x^{(k)} - x'$ 是 x' 处的可行增量, 且可以表示为

$$x^{(k)} - x' = \delta_k \mathbf{p}^{(k)} \quad (7.3.1)$$

其中 $\delta_k > 0$ 是标量且 $\delta_k \rightarrow 0$, $\mathbf{p}^{(k)}$ 是长度固定的向量. 称 $\mathbf{p}^{(k)}$ 的任一聚点 \mathbf{p} 是序列(sequential)可行方向, 用 \mathcal{F}' 表示 x' 处所有序列可行方向的全体, 称为序列可行方向集. 根据定义, 一个序

列可行方向能确定一个可行序列；反之，给定序列 $\{x^{(k)}\}$, $x' \neq x^{(k)} \in \Omega$ 且 $x^{(k)} \rightarrow x'$, 因为 $\{(x^{(k)} - x') / \|x^{(k)} - x'\|\}$ 至少有一个聚点，所以至少可以确定一个序列可行方向。

在开始讨论局部解的必要条件之前，不妨记 $f(x)$ 在 x' 的下降方向集 $D' = \{p \in \mathbb{R}^n \mid p^T g' < 0\}$. 下列引理给出了最基本的必要条件。

引理 7.3.1 若 x^* 是局部极小点，则 $\mathcal{F}^* \cap D^* = \emptyset$, 即在 x^* 处不存在序列可行的下降方向。

证明 设 $p \in \mathcal{F}^*$, 则存在可行序列 $\{x^{(k)}\}$ 满足式(7.3.1). 由一阶 Taylor 展式有

$$f(x^* + \delta_k p^{(k)}) - f(x^*) = \delta_k g^* \cdot p^{(k)} + o(\delta_k)$$

对充分大的 k , 由 x^* 的最优化及 $x^{(k)} \rightarrow x^*$ 知等式的左边非负. 两边同时除以 δ_k , 并令 $k \rightarrow \infty$, 得 $p^T g^* \geq 0$. 这样, $p \notin D^*$. ■

然而, 集合 \mathcal{F}^* 通常不易计算, 为此, 考虑另一个易于计算的可行方向集. 由约束函数在 x' 的一阶 Taylor 近似

$$c_i(x' + s) \approx c_i(x') + \nabla c_i(x')^T s$$

定义点 x' 处的线性化 (linearized) 可行方向是满足

$$p^T a'_i = 0, \quad i \in \mathcal{E}; \quad p^T a'_i \leq 0, \quad i \in \mathcal{I}' \quad (7.3.2)$$

的非零向量 p . 记所有线性化可行方向形成的集合为 F' . 如果两个可行方向集相同, 将会很方便. 为此, 下面讨论二者的关系.

引理 7.3.2 $\mathcal{F}' \subseteq F'$.

证明 令 $p \in \mathcal{F}'$, 则存在可行序列 $x^{(k)} = x' + \delta_k p^{(k)}$, $\delta_k \rightarrow 0$ 和 $p^{(k)} \rightarrow p$. 把约束在 x' 处展开, 得到

$$c_i(x^{(k)}) = c_i(x') + \delta_k p^{(k)T} a'_i + o(\delta_k)$$

对 $i \in \mathcal{E}$, 有 $c_i(x^{(k)}) = c_i(x') = 0$; 对 $i \in \mathcal{I}'$, 有 $c_i(x^{(k)}) \leq c_i(x') = 0$. 因此, 将上式两边同时除以 δ_k , 并令 $k \rightarrow \infty$, 可得 $p \in F'$. ■

令人遗憾的是, 相反的包含关系 $\mathcal{F}' \supseteq F'$ 不一定成立, 而定理 7.2.1 的证明中恰好需要这一事实. 为了避开这个难题, Kuhn 与 Tucker 于 1951 年假设 $\mathcal{F}' = F'$, 并称该事实是点 x' 处的约束规范 (constraint qualification). 根据该约束规范, 对每个 $p \in F'$, 均存在可行序列, 与其对应的方向序列收敛于 p . 他们还给出了该假设不成立的例子.

例 7.3.1 定义

$$\Omega = \{x \in \mathbb{R}^2 : x_2 \leq x_1^3, x_2 \geq 0\}$$

该集合如图 7.3.1 中阴影所示. 考虑点 $x' = (0, 0)^T$ 和方向 $p = (-1, 0)^T \in F'$, 易见不存在可行序列满足与其对应的方向序列收敛于 p . 因而, $(-1, 0)^T \notin \mathcal{F}'$.

然而需要指出的是, 约束规范失效的情况很少出现, 从而假定 $\mathcal{F}' = F'$ 通常是切实可行的. 事实上, 正如下面的结论所显示的, 只要线性约束规范 (Linear Constraint Quality, LCQ) 或者线性无关约束规范 (Linear Independence Constraint Quality, LICQ) 成立, 该假设就成立.

引理 7.3.3 (约束规范条件) 在可行点 x' 处, 如果条件

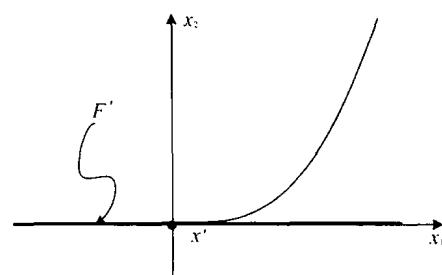


图 7.3.1 约束规范不成立示例

(i) (LCQ) $c_i(\mathbf{x}), i \in \mathcal{A}'$, 是线性约束, 或者

(ii) (LICQ) $\mathbf{a}'_i, i \in \mathcal{A}'$, 线性无关

成立, 则有 $\mathcal{F}' = F'$.

证明 根据 F' 的定义, 当条件(i)成立时, 易验证 $\mathbf{x}^{(k)} = \mathbf{x}' + \frac{1}{k} \mathbf{p}^{(k)}$, 其中 $\mathbf{p}^{(k)} = \mathbf{p}$ 满足式(7.3.1), 从而结论成立. 现在讨论条件(ii). 任取 $\mathbf{p} \in F'$, 需要构造可行序列, 其对应的方向序列收敛到 \mathbf{p} . 为此, 假定 $\mathcal{A}' = \{1, 2, \dots, m\}$ 并考虑非线性方程组

$$\mathbf{r}(\mathbf{x}, \theta) = \mathbf{0} \quad (7.3.3)$$

它的各分量定义为

$$r_i(\mathbf{x}, \theta) = c_i(\mathbf{x}) - \theta \mathbf{p}^\top \mathbf{a}'_i, \quad i = 1, 2, \dots, m$$

$$r_i(\mathbf{x}, \theta) = (\mathbf{x} - \mathbf{x}')^\top \mathbf{b}_i - \theta \mathbf{p}^\top \mathbf{b}_i, \quad i = m+1, m+2, \dots, n$$

记 $\mathbf{A}' = [\mathbf{a}'_1 \cdots \mathbf{a}'_m]$, $\mathbf{B} = [\mathbf{b}_{m+1} \cdots \mathbf{b}_n]$. 当 $\theta = 0$ 时, 方程组有解 \mathbf{x}' ; 此外, 当条件(ii)成立时, 因为矩阵 \mathbf{A}' 满秩, 故可以选择矩阵 \mathbf{B} 使得 $\mathbf{r}(\mathbf{x}, \theta)$ 的 Jacobi 矩阵

$$\mathbf{J}' = \mathbf{J}'(\mathbf{x}', 0) = \nabla_{\mathbf{x}} \mathbf{r}^\top(\mathbf{x}, 0) \mid_{\mathbf{x}=\mathbf{x}'} = [\mathbf{A}' \mid \mathbf{B}]$$

非奇异. 根据隐函数定理, 存在 \mathbf{x}' 的开邻域 $N_{\mathbf{x}'}$ 和 $\theta = 0$ 的开邻域 N_0 , 使得对任意 $\theta \in N_0$, 方程(7.3.3)有唯一解 $\mathbf{x}(\theta) \in N_{\mathbf{x}'}$, 且 $\mathbf{x}(\theta)$ 是 θ 的连续可微函数. 由式(7.3.3)及链式法则, 有

$$0 = \frac{d r_i}{d \theta} = \sum_j \frac{\partial r_i}{\partial x_j} \cdot \frac{d x_j}{d \theta} + \frac{\partial r_i}{\partial \theta}$$

即

$$\mathbf{0} = \mathbf{J}'^\top \frac{d \mathbf{x}}{d \theta} - \mathbf{J}'^\top \mathbf{p}$$

于是有 $\frac{d \mathbf{x}}{d \theta} \Big|_{\theta=0} = \mathbf{p}$. 因此, 如果 $\{\delta_k\}$ 是任一满足 $\delta_k \downarrow 0$ 的序列且 $\delta_k \in N_0$, 则 $\{\mathbf{x}^{(k)} = \mathbf{x}(\delta_k)\}$ 是以 \mathbf{p} 为序列可行方向的可行序列(令 $\mathbf{p}^{(k)} = \|\mathbf{p}\| \cdot (\mathbf{x}^{(k)} - \mathbf{x}') / \|\mathbf{x}^{(k)} - \mathbf{x}'\|$, 则 $\mathbf{x}^{(k)} - \mathbf{x}' = \mathbf{p}^{(k)} \cdot \|\mathbf{x}^{(k)} - \mathbf{x}'\| / \|\mathbf{p}\|$, 按定义即可验证). ■

为了得到易于验证的最优性条件, 需要如下的正则性假定 (regularity assumption)

$$F^* \cap D^* = \mathcal{F}^* \cap D^* \quad (7.3.4)$$

易见 Kuhn-Tucker 约束规范 $F^* = \mathcal{F}^*$ 蕴含着该假定. 然而, 当 $F^* \neq \mathcal{F}^*$ 时, 式(7.3.4)也可能成立.

例 7.3.2 以例 7.3.1 定义的 Ω 为可行域, 考虑下面两个问题

$$\begin{array}{ll} \underset{\mathbf{x} \in \Omega}{\text{minimize}} & x_2 \\ \underset{\mathbf{x} \in \Omega}{\text{minimize}} & x_1 \end{array}$$

二者的解都是 $\mathbf{x}^* = \mathbf{0}$. 由例 7.3.1 知 $\mathcal{F}^* \neq F^*$. 易验证, 对于前者正则性假设(7.3.4)成立, 从而解 \mathbf{x}^* 是 KKT 点, 而对于后者有 $(-1, 0)^\top \in F^* \cap D^*$, 但 $\mathcal{F}^* \cap D^*$ 是空集. 此问题中尽管 \mathbf{x}^* 是极小点, 但不是 KKT 点, 这个例子说明 KKT 条件(7.2.14)的确需要正则性假设.

根据假定(7.3.4), 引理 7.3.1 中的必要条件变成 $F^* \cap D^* = \emptyset$, 即在 \mathbf{x}^* 处不存在线性化可行下降方向. 至此, 可以建立该条件和式(7.2.11)与式(7.2.12)中 Lagrange 乘子的存在性之间的联系. 事实上, 当 $\mathbf{a}'_i, i \in \mathcal{A}'$, 线性无关时, 利用式(7.2.12)后的构造性证明即可说明这一点. 下面的引理给出了更一般的结论.

引理 7.3.4 (Farkas 引理) 设 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ 与 \mathbf{g} 是任意给定的向量, 则集合

$$S = \{ \mathbf{p} : \mathbf{p}^\top \mathbf{g} < 0, \mathbf{p}^\top \mathbf{a}_i \leq 0, i = 1, 2, \dots, m \}$$

是空集当且仅当存在乘子 $\lambda_i \geq 0$ 使得

$$-\mathbf{g} = \sum_{i=1}^m \mathbf{a}_i \lambda_i. \quad (7.3.5)$$

证明 “充分性”易于验证. 如果 \mathbf{p} 满足 $\mathbf{p}^\top \mathbf{a}_i \leq 0$, 因为 $\lambda_i \geq 0$, 式(7.3.5)蕴含着 $\mathbf{p}^\top \mathbf{g} = -\sum_{i=1}^m \lambda_i \mathbf{p}^\top \mathbf{a}_i \geq 0$, 则 S 是空集.

用反证法证明“必要性”. 假设方程组(7.3.5)没有非负解, 下面证明存在向量 $\mathbf{p} \in S$. 该结论从几何上看很直观. 考虑向量集

$$C = \{ \mathbf{v} : \mathbf{v} = \sum_{i=1}^m \mathbf{a}_i \lambda_i, \lambda_i \geq 0 \} \quad (7.3.6)$$

该向量集是一个多面锥 (polyhedral cone), 并且是闭凸集. 由图 7.3.2 可以看到, 如果 $-\mathbf{g} \notin C$, 则存在以 \mathbf{p} 为法向量的超平面分离 (separate) C 和 $-\mathbf{g}$ (参见引理 7.3.5), 即满足 $\mathbf{p}^\top \mathbf{a}_i \leq 0 (i=1, 2, \dots, m), \mathbf{p}^\top \mathbf{g} < 0$, 从而 $\mathbf{p} \in S$. ■

在正则性假定下, 将引理 7.3.1 中的必要条件与 Lagrange 乘子的存在性建立联系时, 确定线性化可行下降方向集 $F^* \cap D^*$ 的条件中含有等式. 这需要将 Farkas 引理推广到含等式的情况.

推论 集合

$$S = \{ \mathbf{p} : \mathbf{p}^\top \mathbf{g}^* < 0, \mathbf{p}^\top \mathbf{a}_i^* = 0, i \in \mathcal{E}, \mathbf{p}^\top \mathbf{a}_i^* \leq 0, i \in \mathcal{I}^* \}$$

是空集当且仅当存在乘子 λ_i 使得式(7.2.11)与式(7.2.12)成立.

证明 忽略上标“*”, 则可将 $\mathbf{p}^\top \mathbf{a}_i = 0 (i \in \mathcal{E})$ 表示成 $-\mathbf{p}^\top \mathbf{a}_i \leq 0$ 和 $\mathbf{p}^\top \mathbf{a}_i \leq 0, i \in \mathcal{E}$. 由 Farkas 引理知, 存在非负乘子 $\lambda_i, i \in \mathcal{I}^*, \lambda_i^+, \lambda_i^- \in \mathcal{E}$ 使得

$$-\mathbf{g} = \sum_{i \in \mathcal{I}^*} \lambda_i \mathbf{a}_i + \sum_{i \in \mathcal{E}} \lambda_i^+ \mathbf{a}_i - \sum_{i \in \mathcal{E}} \lambda_i^- \mathbf{a}_i$$

定义 $\lambda_i = \lambda_i^+ - \lambda_i^-, i \in \mathcal{E}$, 即得式(7.2.11)与式(7.2.12). ■

引理 7.3.5 (点与闭凸锥的分离定理) 考虑式(7.3.6)定义的闭凸锥 C . 若给定的向量 $\mathbf{a} \notin C$, 则存在超平面 $\mathbf{p}^\top \mathbf{x} = 0$ 分离 C 和向量 \mathbf{a} .

证明 利用构造性方法证明结论. 考虑在 C 上极小化 $\|\mathbf{x} - \mathbf{a}\|^2$. 设 $\mathbf{x}_1 \in C$. 因为解满足 $0 \leq \|\mathbf{x} - \mathbf{a}\| \leq \|\mathbf{x}_1 - \mathbf{a}\|$, 故 $\|\mathbf{x} - \mathbf{a}\|$ 在 C 上有界; 再由 2-范数的连续性知极小点存在, 记为 $\hat{\mathbf{a}}$. 因为对所有 $\lambda \geq 0$ 有 $\lambda \hat{\mathbf{a}} \in C$, 故 $\varphi(\lambda) := \|\lambda \hat{\mathbf{a}} - \mathbf{a}\|^2$ 在 $\lambda = 1$ 处取最小值, 从而 $\varphi'(1) = 0$, 即

$$\hat{\mathbf{a}}^\top (\mathbf{a} - \hat{\mathbf{a}}) = 0 \quad (7.3.7)$$

设 $\mathbf{x} \in C$, 由凸性知, 对所有 $\theta \in (0, 1)$ 有 $\hat{\mathbf{a}} + \theta(\mathbf{x} - \hat{\mathbf{a}}) \in C$, 因此有

$$\|\theta(\mathbf{x} - \hat{\mathbf{a}}) + \hat{\mathbf{a}} - \mathbf{a}\|^2 \geq \|\hat{\mathbf{a}} - \mathbf{a}\|^2$$

把上式展开, 将式(7.3.7)代入, 并令 $\theta \downarrow 0$ 得到

$$(\mathbf{x} - \hat{\mathbf{a}})^\top (\mathbf{a} - \hat{\mathbf{a}}) = \mathbf{x}^\top (\mathbf{a} - \hat{\mathbf{a}}) \leq 0$$

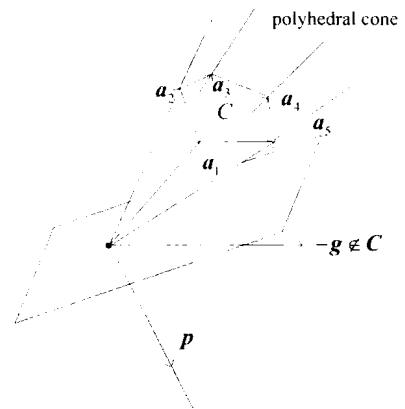


图 7.3.2 分离超平面的存在性

令向量 $p = a - \hat{a}$, 则对所有 $x \in C$ 有 $p^T x \leq 0$. 再由式(7.3.7)有 $a^T p = p^T p$, 且因为 $a \notin C$, 故有 $p \neq 0$. 因此 $a^T p > 0$, 超平面 $p^T x = 0$ 分离 C 和 a . ■

从几何观点看, 引理的证明中构造的向量 p 沿着 $-g$ 到 C 的最近边界的垂线; 在图 7.3.2 中, 相应的超平面沿着 a_5 与锥相切. 此外, 这个结论可以推广到一般的闭凸集 C , 并把 \hat{a} 称为 a 在 C 上的投影(projection).

现在, 可以把本节的内容组织起来证明一阶条件(式(7.2.11)与式(7.2.12))(或者等价地, 定理 7.2.1 中的条件(7.2.14))是局部极小点的必要条件. 考虑点 x^* , 由引理 7.3.1 知 x^* 处没有序列可行且下降的方向, 即 $\mathcal{F}^* \cap D^* = \emptyset$; 再由正则性假设(7.3.4), 在 x^* 处没有线性化可行下降方向. 于是根据 Farkas 引理的推论知式(7.2.11)与式(7.2.12)成立. 至此, 已经相当严密地证明了这些结论.

7.4 二阶条件

7.2 节内容的自然延续就是在局部极小点的某个邻域内检验二阶曲率项. 对于无约束优化问题(第 4 章), G^* 正定这个充分条件在设计满意的算法时具有重要的内涵. 对于约束优化问题, 情况有些特殊, 但此时约束的曲率也起相当重要的作用, 而不能只孤立地考虑 $f(x)$ 的曲率. 例 7.4.1 中目标函数的稳定点 x^* 处的 G^* 是正定的, 然而当 $\beta > 1/2$ 时它不是局部极小点. 采用与一阶条件相同的处理办法, 先用相当直接的方法叙述基本结论, 紧接着给出更一般的严格处理. 假定 $f(x)$ 和所有 $c_i(x)$ 都是 C^2 函数.

先考虑只有等式约束的情况, 设最优解 x^* 存在, 且在点 x^* 处向量 $a_i^* (i \in \mathcal{E})$ 线性无关. 在这些条件下, 对于序列可行方向 $p \in \mathcal{F}^*$, 存在可行序列 $x^{(k)}$ 及对应的方向序列 $p^{(k)} \rightarrow p$. 由可行性有 $f(x^{(k)}) = f(x^* + \delta_k p^{(k)}) = \mathcal{L}(x^* + \delta_k p^{(k)}, \lambda^*)$. 因为 x^* 是 \mathcal{L} 的稳定点, 因此由 $\mathcal{L}(x, \lambda^*)$ 在 x^* 的 Taylor 展式可将二阶项孤立出来, 即

$$\begin{aligned} f(x^* + \delta_k p^{(k)}) &= \mathcal{L}(x^* + \delta_k p^{(k)}, \lambda^*) \\ &= f^* + \frac{1}{2} \delta_k^2 p^{(k)T} W^* p^{(k)} + o(\delta_k^2) \end{aligned} \quad (7.4.1)$$

其中 $W^* = \nabla_x^2 \mathcal{L}(x^*, \lambda^*) = \nabla^2 f(x^*) + \sum_i \lambda_i^* \nabla^2 c_i(x^*)$ 表示 Lagrange 函数关于 x 的 Hessian 阵. 在式(7.4.1)中, 将两边同时除以 δ_k^2 , 并令 $k \rightarrow \infty$, 再由 f^* 是局部极小值得到

$$p^T W^* p \geq 0 \quad (7.4.2)$$

像 7.3 节那样, 序列可行方向满足 $a_i^{*T} p = 0, i \in \mathcal{E}$, 写成矩阵形式即

$$A^{*T} p = 0 \quad (7.4.3)$$

这样, 局部极小点的二阶必要条件是式(7.4.2)必须对任一满足式(7.4.3)的向量 p 成立, 即 Lagrange 函数 $\mathcal{L}(x, \lambda^*)$ 在 x^* 处沿所有序列可行方向的曲率必须非负. 当然, 如果没有约束条件, 式(7.4.2)即退化成通常的目标函数的 Hessian 阵 G^* 是半正定的.

与第 4 章中无约束优化问题相同, 也可以给出非常类似的充分条件. 如果对某可行点 x^* 有式(7.2.3)成立, 且对所有满足式(7.4.3)的 $p (\neq 0)$ 有

$$p^T W^* p > 0 \quad (7.4.4)$$

则 \mathbf{x}^* 是严格局部极小点. 用反证法可以说明该事实. 假设 \mathbf{x}^* 不是严格局部极小点, 则存在可行序列 $\mathbf{x}^{(k)} \neq \mathbf{x}^* (k=1, 2, \dots)$ 使得 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$, 且 $f(\mathbf{x}^{(k)}) \leq f^*$. 设 \mathbf{p} 是由该可行序列确定的序列可行方向, 则其满足式(7.4.3). 但由 $f(\mathbf{x}^{(k)}) \leq f^*$ 和 $\mathcal{L}(\mathbf{x}, \lambda^*)$ 在 \mathbf{x}^* 的二阶 Taylor 展式, 类似于式(7.4.1)可得 $\mathbf{p}^\top \mathbf{W}^* \mathbf{p} \leq 0$, 这与所给条件矛盾. 需要注意的是, 这里的充分性证明不需要正则性假定.

例 7.4.1 (二阶条件) 考虑

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) = \frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}x_2^2 \\ \text{subject to} \quad & c(\mathbf{x}) = x_1 - \beta x_2 = 0 \end{aligned} \quad (7.4.5)$$

其中 β 是参数. 讨论 β 取何值时, $\mathbf{x}^* = \mathbf{0}$ 是局部极小点.

图 7.4.1 给出了 $\beta = 1/4$ (\mathbf{x}^* 是局部极小点) 和 $\beta = 1$ (\mathbf{x}^* 不是局部极小点) 时问题的图示. 这里 $\mathbf{g}^* = (-1, 0)^\top$, $\mathbf{a}^* = (1, 0)^\top$, 因此一阶条件(7.2.3)满足, 其中 $\lambda^* = 1$, 且 \mathbf{x}^* 是可行的. 此时式(7.4.3)中的可行方向集是 $\{\mathbf{p} = (0, p_2)^\top : p_2 \neq 0\}$. 由 $\mathbf{W}^* = \begin{bmatrix} 1 & 0 \\ 0 & 1-2\beta \end{bmatrix}$, 得 $\mathbf{p}^\top \mathbf{W}^* \mathbf{p} = (1-2\beta)p_2^2$. 这样, 当 $\beta > 1/2$ 时, 必要条件不成立, 因此断定 \mathbf{x}^* 不是局部极小点. 当 $\beta < 1/2$ 时, 二阶充分条件满足, 因此断定 \mathbf{x}^* 是局部极小点. 当 $\beta = 1/2$ 时, \mathcal{L} 沿着可行方向的曲率为零, 故由二阶条件得不出结论. 此时需要更高阶导数或者别的办法(比如变量消元法)来验证, 我们留给读者思考.

一个重要的推广就是把这些结论推广到有不等式约束的优化问题. 到目前为止, 仅考虑沿可行稳定方向($\mathbf{p}^\top \mathbf{g}^* = 0$)的二阶条件, 而不用考虑沿上升方向的条件. 如果出现不等式约束 $c_i(\mathbf{x}) \leq 0$, 且它的乘子 $\lambda_i^* > 0 (i \in \mathcal{I}^*)$, 则使得 $\mathbf{p}^\top \mathbf{a}_i^* < 0 (i \in \mathcal{I}^*)$ 的方向是目标函数的上升方向, 因而稳定方向满足

$$\mathbf{p}^\top \mathbf{a}_i^* = 0, \quad i \in \mathcal{A}^* \quad (7.4.6)$$

这种情况下, 二阶必要条件是对所有满足式(7.4.6)的 \mathbf{p} 皆有式(7.4.2)成立. 理解该结论的另一种方式是, 如果 \mathbf{x}^* 是问题(7.0.1)的局部解, 则它也必为问题

$$\begin{aligned} \text{minimize}_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{A}^* \end{aligned} \quad (7.4.7)$$

的局部解, 因此由式(7.4.2)和式(7.4.3)可以得到这些条件.

至于充分条件, 假设 \mathbf{x}^* 可行, 式(7.2.11)与式(7.2.12)成立, 且如果 $\lambda_i^* > 0 (i \in \mathcal{I}^*)$ 成立(即 \mathbf{x}^* 是满足严格互补性的 KKT 点), 则对所有式(7.4.6)中的 $\mathbf{p} \neq \mathbf{0}$, 式(7.4.4)是充分条件. 可以在一个更大的子空间(在该空间上 $\lambda_i^* = 0$ 的条件可以去掉)上假定有正曲率. 将式(7.4.5)中的等式约束改为 $c(\mathbf{x}) \leq 0$ 来说明这些条件. 这时可行方向 $\mathbf{p} = (p_1, p_2)^\top, p_1 \geq 0, p_2 \in \mathbb{R}$, 且 $\mathbf{p} \neq \mathbf{0}$. 由于 $\lambda^* = 1 > 0$, 故可行方向中除 $p_1 = 0$ 外均为上升方向. 于是, 稳定的可行方向与等式约束问题的相同, 形如 $\mathbf{p} = (0, p_2)^\top$. 由此对 β 可以得出相同的结论.

下面对结论进一步推广, 使其包含 $\lambda_i^* = 0 (i \in \mathcal{I}^*)$ 这种情况. 此时, 可能存在满足 $\mathbf{p}^\top \mathbf{a}_i^* < 0$

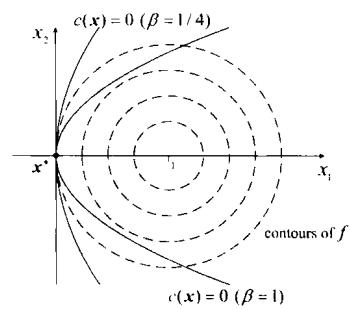


图 7.4.1 二阶条件

的稳定方向,且必要条件中的 λ^* 可能不唯一. 下面给出包含这些特征的二阶条件的严格推导. 任给固定向量 λ^* , 定义严格(strictly)积极约束或强(strongly)积极约束

$$\mathcal{A}^+ = \{i : i \in \mathcal{E} \text{ 或者 } \lambda_i^* > 0, i \in \mathcal{I}^*\} \quad (7.4.8)$$

从 \mathcal{A}^* 中删除 $\lambda_i^* = 0 (i \in \mathcal{I}^*)$ 的元素, 即可得到该集合. 考虑由满足

$$c_i(\mathbf{x}^{(k)}) = 0, \quad \forall i \in \mathcal{A}^+ \quad (7.4.9)$$

的所有可行序列(其中 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$)定义的序列可行方向, 记这些方向所形成的集合是 \mathcal{G}^+ . 如同 7.3 节, 考虑可行方向集

$$G^* = \{\mathbf{p} \in \mathbb{R}^n : \mathbf{p} \neq \mathbf{0}, \mathbf{a}_i^T \mathbf{p} = 0, i \in \mathcal{A}^+, \mathbf{a}_i^T \mathbf{p} \leq 0, i \in \mathcal{I}^+, \lambda_i^* = 0\} \quad (7.4.10)$$

即将 \mathcal{A}^* 中的约束(包括式(7.4.8))线性化后的可行方向集. 用与引理 7.3.2 类似的讨论可得 $\mathcal{G}^+ \subseteq G^*$. 然而, 为了陈述二阶必要条件, 还需要一个反包含关系. 为此, 进行另一个正则性假设, 即

$$\mathcal{G}^+ = G^* \quad (7.4.11)$$

在引理 7.3.3 所给的条件下, 用类似的方法可以证明正则性假设(7.4.11)成立, 所以进行这样的假设是合理的. 现在可以用更为一般的形式表述本节的主要结论.

定理 7.4.1 (二阶必要条件) 若 \mathbf{x}^* 是问题(7.0.1)的局部极小点, 且正则性假设(7.3.4)成立, 则存在 Lagrange 乘子 λ^* 使得 KKT 条件(7.2.14)成立. 对任一这样的 λ^* , 如果式(7.4.11)成立, 则

$$\mathbf{p}^T \mathbf{W}^* \mathbf{p} \geq 0, \quad \forall \mathbf{p} \in G^* \quad (7.4.12)$$

证明 设 $\mathbf{p} \in G^*$, 则 $\mathbf{p} \in \mathcal{G}^+$. 由 \mathcal{G}^+ 的定义知, 存在可行序列 $\{\mathbf{x}^{(k)}\}$ 及方向序列 $\{\mathbf{p}^{(k)}\}$ 使得式(7.4.9)成立. 当 $i \in \mathcal{A}^+$ 时, 有 $c_i^{(k)} = 0$; 否则, $\lambda_i^* = 0$. 因此, $f^{(k)} = \mathcal{L}(\mathbf{x}^{(k)}, \lambda^*)$. 由式(7.3.1)、KKT 条件(7.2.14)以及 $\mathcal{L}(\mathbf{x}, \lambda^*)$ 关于 \mathbf{x}^* 的二阶 Taylor 展式, 有

$$\mathcal{L}(\mathbf{x}^{(k)}, \lambda^*) = f^* + \frac{1}{2} \delta_k^2 \mathbf{p}^{(k)T} \mathbf{W}^* \mathbf{p}^{(k)} + o(\delta_k^2) \quad (7.4.13)$$

因为 \mathbf{x}^* 是局部极小点, 对充分大的 k 有 $f^{(k)} \geq f^*$, 因此由(7.4.13)式得到

$$\frac{1}{2} \mathbf{p}^{(k)T} \mathbf{W}^* \mathbf{p}^{(k)} + o(1) \geq 0$$

取极限即可得到所需结论(7.4.12). ■

定理 7.4.2 (二阶充分条件) 如果在 \mathbf{x}^* 处存在乘子 λ^* 使得条件(7.2.14)成立, 且

$$\mathbf{p}^T \mathbf{W}^* \mathbf{p} > 0, \quad \forall \mathbf{p} \in G^* \quad (7.4.14)$$

则 \mathbf{x}^* 是问题(7.0.1)的严格局部极小点.

证明 假定 \mathbf{x}^* 不是严格局部极小点, 则存在可行序列 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$ 使得 $\mathbf{x}^{(k)} \neq \mathbf{x}^*$, 且 $f^{(k)} \leq f^*$. 在式(7.3.1)中固定 $\|\mathbf{p}^{(k)}\| = 1$, 该事实蕴含着存在子序列 $\mathbf{p}^{(k)} \rightarrow \mathbf{p}$. 由引理 7.3.2 知 $\mathbf{p} \in F^*$, 类似于引理 7.3.1 中的讨论, 有 $\mathbf{p}^T \mathbf{g}^* \leq 0$. 因此必有 $\mathbf{p} \in G^*$. 否则, 存在 $i : \lambda_i^* > 0$ 且 $\mathbf{a}_i^T \mathbf{p} \leq 0$, 由此得 $-\mathbf{p}^T \mathbf{g}^* = \sum \lambda_i^* \mathbf{p}^T \mathbf{a}_i^* < 0$, 产生矛盾.

此时, 由 $\mathbf{x}^{(k)}$ 的可行性有 $\mathcal{L}(\mathbf{x}^{(k)}, \lambda^*) \leq f^{(k)}$. 再由式(7.4.13)有

$$0 \geq f^{(k)} - f^* \geq \frac{1}{2} \delta_k^2 \mathbf{p}^{(k)T} \mathbf{W}^* \mathbf{p}^{(k)} + o(\delta_k^2)$$

不等式两边除以 δ_k^2 , 并取极限, 所得结果与式(7.4.14)矛盾. ■

保证式(7.4.14)成立的一个充分条件是 $\forall \mathbf{p} \neq \mathbf{0}$ 使得 $\mathbf{p}^T \mathbf{a}_i^* = 0 (i \in \mathcal{A}^+)$, 有 $\mathbf{p}^T \mathbf{W}^* \mathbf{p} > 0$.

在实践中,该条件更易于验证.

7.5 凸规划

称在凸集 $\Omega \subseteq \mathbb{R}^n$ 上极小化凸函数的问题是凸规划(convex programming),用式(7.0.1)可表示为

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad x \in \Omega := \{x \mid c_i(x) \leq 0, i = 1, 2, \dots, m\} \end{aligned} \quad (7.5.1)$$

其中 $f(x), c_i(x) (i=1, 2, \dots, m)$ 为 \mathbb{R}^n 上的凸函数. 式(7.5.1)的可行域为凸集的结论可由下列引理与命题 2.1.1(iii) 推出.

引理 7.5.1 若 $f(x)$ 是凸集 $C \subseteq \mathbb{R}^n$ 上的凸函数, 则水平集(level set) $L = \{x \in C : f(x) \leq \gamma\}$ 是凸集.

证明 对 $x_0, x_1 \in L$, 若 x_θ 由式(4.1.6)给定, 则由集合 C 凸知 $x_\theta \in C$. 由 $f(x)$ 是凸的及 L 的定义可得

$$f(x_\theta) \leq (1-\theta)f_0 + \theta f_1 \leq (1-\theta)\gamma + \theta\gamma = \gamma$$

因而 $x_\theta \in L$, 即 L 是凸集. ■

注意问题(7.5.1)不能包括一般的等式约束, 但可以包括线性等式 $c_i(x) = \mathbf{a}_i^\top x - b_i = 0$, 把它们作为 $c_i(x) \leq 0$ 与 $-c_i(x) \leq 0$ 的交集. 线性规划是凸规划. 至于二次规划(8.0.1), 当其目标函数的 Hessian 阵半正定时, 是一个凸规划问题.

凸性除了保证局部极小点是全局极小点外, 它的第二个有用之处在于它搭建了一个架构. 在该架构下, 一阶必要条件(KKT)是局部极小点的充分条件, 见如下定理. 与其他的充分条件(定理 7.4.1)一样, 该定理也不需要正则性假设.

定理 7.5.1 若凸规划问题(7.5.1)中的函数 $f(x)$ 与 $c_i(x) (i=1, 2, \dots, m)$ 是 \mathbb{R}^n 上的 C^1 函数, 且在 x^* 处条件(7.2.14)满足, 则 x^* 是问题(7.5.1)的全局最优解.

证明 令 $x' \neq x^*$ 是任意的可行解, 则由于 $\lambda_i^* \geq 0, c_i' \leq 0$. 利用式(4.1.7), 并由 f 与 c_i 的凸性可得

$$f' \geq f' + \sum_{i=1}^m \lambda_i^* c_i' \geq f^* + (x' - x^*)^\top \mathbf{g}^* + \sum_{i=1}^m \lambda_i^* (c_i^* + (x' - x^*)^\top \mathbf{a}_i^*)$$

由式(7.2.14a)和式(7.2.14e), 有 $\lambda_i^* c_i^* = 0$ 及 $\mathbf{g}^* + \sum_{i=1}^m \mathbf{a}_i^* \lambda_i^* = 0$, 从而 $f' \geq f^*$. 因而 x^* 是全局最优解. ■

例 7.5.1 (凸规划) 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^2}{\text{minimize}} \quad \left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^4 \\ & \text{subject to} \quad \begin{aligned} x_1 + x_2 - 1 &\leq 0 \\ x_1 - x_2 - 1 &\leq 0 \\ -x_1 + x_2 - 1 &\leq 0 \\ -x_1 - x_2 - 1 &\leq 0 \end{aligned} \end{aligned} \quad (7.5.2)$$

如图 7.5.1 所示, 该问题的解 $\mathbf{x}^* = (1, 0)^T$. 在该点, 第一个和第二个约束是积极的. 记它们为 c_1 和 c_2 , 有

$$\mathbf{g}^* = \begin{bmatrix} -1 \\ -\frac{1}{2} \end{bmatrix}, \quad \mathbf{a}_1^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{a}_2^* = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

因此, 置 $\lambda^* = (3/4, 1/4, 0, 0)^T$ 可验证 KKT 条件(7.2.14)满足. 因为该问题是凸规划, 故这个 KKT 点也是全局解.

例 7.1.1 也是凸规划. 其目标函数是线性的, 因而是凸的; Hessian 阵

$$\nabla^2 c_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{与} \quad \nabla^2 c_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

是半正定的, 因而约束函数是凸的. 从图 7.1.2 中可以看出可行域是凸集. 由于在点 $\mathbf{x}^* = (1/\sqrt{2}, 1/\sqrt{2})^T$ 处一阶条件成立, 因此由定理 7.5.1 知它是问题的全局解.

概括起来, 要使 \mathbf{x}^* 是凸规划问题(7.5.1)的解, 则条件(7.2.14)与正则性假设(7.3.4)是必要的, 而条件(7.2.14)本身又是充分的. 当然, 正则性假设(7.3.4)可由约束规范 $\mathcal{F}^* = F^*$ 推出, 而这又可由引理 7.3.3 的假设推出. 需要强调的一点是, 不能把正则性假设(7.3.4)省掉. 考虑不等式 $x_2 \geq x_1^2$ 和 $x_2 \leq 0$ 确定的可行域在 $\mathbf{x}^* = 0$ 的情况, 易见集合是凸的, 但 $\mathcal{F}^* = \emptyset \neq F^*$.

从上面诸定理可以看到凸性假设的本质是曲率或二阶假设, 它要求方向导数沿任何直线都是非降的. 因此, 虽然凸性对某些特殊类型的问题给出了非常有用的结论, 但这样的假设在一般情况下并不成立. 定理 7.4.1 的二阶条件所需的假设要弱得多, 即仅要求

$$\mathbf{W}^* = \nabla^2 f^* + \sum_{i \in \mathcal{A}^*} \lambda_i^* \nabla^2 c_i^*$$

在一个限定的子空间内正定. 在局部最优解处不满足这一条件的情况一般比较少. 因此, 凸性并没有为一般的非线性规划问题提供一个有用的可以导出具体算法的模式, 而二阶条件却能做到这一点.

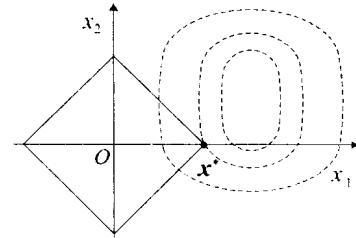


图 7.5.1 凸规划问题(7.5.2)

7.6 凸规划和 Lagrange 乘子

本节给出凸规划最优解的刻画. 与前面不同的是, 这里用凸性取代可微性要求, 利用它能得到全局解的充分条件, 而非 7.2 节中局部解的必要条件. 为了表述简单, 假定考虑的问题形如

$$\begin{aligned} & \underset{\mathbf{x} \in X}{\text{minimize}} && f(\mathbf{x}) \\ & \text{subject to} && c_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{I} \end{aligned} \tag{7.6.1}$$

其中, $f, c_i, i \in \mathcal{I}$, 为凸函数, X 为凸集. 集合约束 $\mathbf{x} \in X \subseteq \mathbb{R}^n$ 表示一些需要处理的特殊约束. 它们可能是额外的显式函数确定的约束, 比如关于变量的非负性约束等. 当 $X = \mathbb{R}^n$ 时, 不存在“特殊要求”.

为了方便,令 $|\mathcal{I}|=m$, $\mathbf{c}(\mathbf{x})=(c_i(\mathbf{x})) \in \mathbb{R}^m$. 称式(7.6.1)为原始(primal)问题. 定义与式(7.6.1)相联系的扰动问题(perturbation problem)为对 $\forall \mathbf{z} \in \mathbb{R}^m$,有

$$\begin{aligned} & \underset{\mathbf{x} \in X}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad \mathbf{c}(\mathbf{x}) \leq \mathbf{z} \end{aligned} \quad (7.6.2)$$

再定义 \mathbb{R}^m 中的集合

$$\Gamma = \{\mathbf{z} \in \mathbb{R}^m : \exists \mathbf{x} \in X \text{ s. t. } \mathbf{c}(\mathbf{x}) \leq \mathbf{z}\} \quad (7.6.3)$$

可证明 Γ 为 \mathbb{R}^m 中的凸集. 取 $\mathbf{z}_0, \mathbf{z}_1 \in \Gamma$ 及 $\theta \in (0, 1)$, 则 $\exists \mathbf{x}_i \in X$ 满足 $\mathbf{c}(\mathbf{x}_i) \leq \mathbf{z}_i$. 由 X 凸知 $(1-\theta)\mathbf{x}_0 + \theta\mathbf{x}_1 \in X$. 此外, 由 $c_i(\mathbf{x})$ 凸有

$$\mathbf{c}((1-\theta)\mathbf{x}_0 + \theta\mathbf{x}_1) \leq (1-\theta)\mathbf{c}(\mathbf{x}_0) + \theta\mathbf{c}(\mathbf{x}_1) \leq (1-\theta)\mathbf{z}_0 + \theta\mathbf{z}_1$$

故 $(1-\theta)\mathbf{z}_0 + \theta\mathbf{z}_1 \in \Gamma$. 在集合 Γ 上定义原始函数(primal function)

$$\omega(\mathbf{z}) = \inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{c}(\mathbf{x}) \leq \mathbf{z}\} \quad (7.6.4)$$

其取值不必有限. 扰动问题即给定 $\mathbf{z} \in \Gamma$, 求 $\omega(\mathbf{z})$; 原始问题即求 $\omega(\mathbf{0})$. 首先可以得到 $\omega(\mathbf{z})$ 的如下性质.

命题 7.6.1 原始函数 ω 是凸的.

证明 任取 $\mathbf{z}_0, \mathbf{z}_1 \in \Gamma, \theta \in (0, 1)$, 则

$$\begin{aligned} \omega((1-\theta)\mathbf{z}_0 + \theta\mathbf{z}_1) &= \inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{c}(\mathbf{x}) \leq (1-\theta)\mathbf{z}_0 + \theta\mathbf{z}_1\} \\ &\leq \inf\{f(\mathbf{x}) : \mathbf{x} = (1-\theta)\mathbf{x}_0 + \theta\mathbf{x}_1, \mathbf{x}_i \in X, \mathbf{c}(\mathbf{x}_i) \leq \mathbf{z}_i, i = 0, 1\} \\ &\leq (1-\theta)\inf\{f(\mathbf{x}_0) : \mathbf{x}_0 \in X, \mathbf{c}(\mathbf{x}_0) \leq \mathbf{z}_0\} + \\ &\quad \theta \inf\{f(\mathbf{x}_1) : \mathbf{x}_1 \in X, \mathbf{c}(\mathbf{x}_1) \leq \mathbf{z}_1\} \\ &= (1-\theta)\omega(\mathbf{z}_0) + \theta\omega(\mathbf{z}_1) \end{aligned}$$

第一个不等式是因为 $\{(1-\theta)\mathbf{x}_0 + \theta\mathbf{x}_1 : \mathbf{x}_i \in X, \mathbf{c}(\mathbf{x}_i) \leq \mathbf{z}_i, i = 0, 1\} \subseteq \{\mathbf{x} \in X : \mathbf{c}(\mathbf{x}) \leq (1-\theta)\mathbf{z}_0 + \theta\mathbf{z}_1\}$, 而这个集合包含关系可由 X 是凸集和 $c_i(\mathbf{x})$ 是凸函数得到. 第二个不等式源于 f 是凸函数的事实. ■

命题 7.6.2 函数 ω 是单调不增的, 即如果 $\mathbf{z}_1 \geq \mathbf{z}_2$, 则 $\omega(\mathbf{z}_1) \leq \omega(\mathbf{z}_2)$.

证明 如果 $\mathbf{z}_1 \geq \mathbf{z}_2$, 则 $\{\mathbf{x} \in X : \mathbf{c}(\mathbf{x}) \leq \mathbf{z}_1\} \supseteq \{\mathbf{x} \in X : \mathbf{c}(\mathbf{x}) \leq \mathbf{z}_2\}$, 直接可得结论. ■

当 $m=1$ 时,一个典型的 $\omega(\mathbf{z})$ 如图 7.6.1 所示. 因为 ω 是凸的,从而在 $(\mathbf{0}, \omega(\mathbf{0}))$ 处存在支撑超平面. 如果尝试倾斜 ω 使得这个相切的超平面变成新的水平面,那么 ω 就会在 $\mathbf{z}=\mathbf{0}$ 处取到最小值. 换句话说,即给 $\omega(\mathbf{z})$ 添加一个恰当的函数 $\mathbf{z}^\top \boldsymbol{\lambda}^*$,所得到的组合 $\omega(\mathbf{z}) + \mathbf{z}^\top \boldsymbol{\lambda}^*$ 在 $\mathbf{z}=\mathbf{0}$ 处取到最小值. 这个线性函数的梯度向量 $\boldsymbol{\lambda}^*$ 就是这个问题的 Lagrange 乘子. 图 7.6.1 中显示的切平面的法向量为 $(\boldsymbol{\lambda}^*, 1)$. 这些讨论可由如下定理来精确描述.

定理 7.6.1 (Lagrange 乘子的存在性) 假设问题(7.6.1)是凸规划,即其中的 $f, c_i (i \in \mathcal{I})$ 为凸函数, X 为凸集;且 Slater 约束规范条件成立,即

$$\exists \mathbf{x}' \in X \text{ s. t. } c_i(\mathbf{x}') < 0, \quad i \in \mathcal{I} \quad (7.6.5)$$

如果

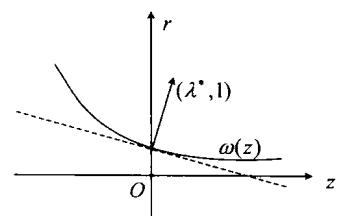


图 7.6.1 $\omega(\mathbf{z})$ 及其在 $\mathbf{z}=\mathbf{0}$ 处的支撑超平面

$$p^* = \inf\{f(\mathbf{x}) : \mathbf{x} \in X, \mathbf{c}(\mathbf{x}) \leq \mathbf{0}\}$$

是有限值, 则存在 $\lambda_i^* \geq 0 (i \in \mathcal{I})$ 使得

$$p^* = \inf_{\mathbf{x} \in X} \left[f(\mathbf{x}) + \sum_{i=1}^m \lambda_i^* c_i(\mathbf{x}) \right] \quad (7.6.6)$$

进一步, 如果原问题中的下确界可以在 \mathbf{x}^* 处取到, 则式(7.6.6)中的下确界也可以在 \mathbf{x}^* 处取到, 且满足

$$\lambda_i^* c_i(\mathbf{x}^*) = 0, \quad i \in \mathcal{I} \quad (7.6.7)$$

证明 定义集合

$$A = \{(\mathbf{z}, r) \in \mathbb{R}^m \times \mathbb{R} : \exists \mathbf{x} \in X \text{ s.t. } c_i(\mathbf{x}) \leq z_i, f(\mathbf{x}) \leq r\}$$

和

$$B = \{(\mathbf{z}, r) \in \mathbb{R}^m \times \mathbb{R} : \mathbf{z} \leq \mathbf{0}, r \leq p^*\}$$

凸性假设蕴含着 A, B 均为凸集. p^* 的定义蕴含着 $A \cap \overset{\circ}{B} = \emptyset$, 其中 $\overset{\circ}{B}$ 表示集合 B 的内部.

$p^* > -\infty$ 蕴含着 $\overset{\circ}{B} \neq \emptyset$. 由凸集分离定理, 存在 $(\lambda^*, \nu^*) \neq \mathbf{0}$ 使得

$$\mathbf{z}_1^T \lambda^* + r_1 \nu^* \geq \mathbf{z}_2^T \lambda^* + r_2 \nu^*, \quad \forall (\mathbf{z}_1, r_1) \in A, (\mathbf{z}_2, r_2) \in B$$

由 B 的定义和上述分离事实, 立即可以得到 $\nu^* \geq 0$ 和 $\lambda^* \geq \mathbf{0}$.

现在证明 $\nu^* > 0$. 一方面, 因为点 $(\mathbf{0}, p^*) \in B$, 由分离事实有

$$\mathbf{z}^T \lambda^* + \nu^* r \geq \nu^* p^*, \quad \forall (\mathbf{z}, r) \in A \quad (7.6.8)$$

如果 $\nu^* = 0$, 则 $\lambda^* \neq \mathbf{0}$. 进而由 $(c(\mathbf{x}'), f(\mathbf{x}')) \in A$ 得

$$c(\mathbf{x}')^T \lambda^* \geq 0$$

另一方面, 又因为 $c(\mathbf{x}') < \mathbf{0}$, 且 $\lambda^* \geq \mathbf{0}$ 但 $\lambda^* \neq \mathbf{0}$, 故有 $c(\mathbf{x}')^T \lambda^* < 0$, 从而产生矛盾. 因此 $\nu^* > 0$, 不失一般性, 可以设 $\nu^* = 1$. 这样, 由 A 的定义和式(7.6.8)有(其中 $\nu^* = 1$)

$$\begin{aligned} p^* &\leq \inf\{\mathbf{z}^T \lambda^* + r : (\mathbf{z}, r) \in A\} \\ &\leq \inf\{f(\mathbf{x}) + c(\mathbf{x})^T \lambda^* : \mathbf{x} \in X\} \\ &\leq \inf\{f(\mathbf{x}) + c(\mathbf{x})^T \lambda^* : \mathbf{x} \in X, c(\mathbf{x}) \leq \mathbf{0}\} \\ &\leq \inf\{f(\mathbf{x}) : \mathbf{x} \in X, c(\mathbf{x}) \leq \mathbf{0}\} \\ &= p^* \end{aligned} \quad (7.6.9)$$

从而式(7.6.6)成立.

如果原始问题的下确界 p^* 可以在 \mathbf{x}^* 处取到, 即存在 $\mathbf{x}^* \in X, c(\mathbf{x}^*) \leq \mathbf{0}$ 且 $f(\mathbf{x}^*) = p^*$, 则由式(7.6.9)有

$$p^* \leq f(\mathbf{x}^*) + c(\mathbf{x}^*)^T \lambda^* \leq f(\mathbf{x}^*) = p^*$$

故 $c(\mathbf{x}^*)^T \lambda^* = 0$. 进一步, $\forall \mathbf{x} \in X$, 由式(7.6.9)有

$$f(\mathbf{x}) + c(\mathbf{x})^T \lambda^* \geq p^* = f(\mathbf{x}^*) = f(\mathbf{x}^*) + c(\mathbf{x}^*)^T \lambda^*$$

故式(7.6.6)中的下确界也可以在 \mathbf{x}^* 处取到. ■

定义问题(7.6.1)的 Lagrange 函数 $\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i \in \mathcal{I}} \lambda_i c_i(\mathbf{x})$. 定理 7.6.3 是凸问题的几何版本的 Lagrange 乘子定理, 它的等价代数表述是如下的鞍点命题.

推论 设定理 7.6.1 中的假设成立, 并且假设问题(7.6.1)在 \mathbf{x}^* 处取到最小值, 则存在 $\lambda^* \geq \mathbf{0}$ 使得 $(\mathbf{x}^*, \lambda^*)$ 是 Lagrange 函数 $\mathcal{L}(\mathbf{x}, \lambda)$ 的鞍点, 即

$$\mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*), \quad \forall x \in X, \quad \forall \lambda \geq 0$$

证明 设 λ^* 是定理 7.6.1 所得到的, 由定理 7.6.1 的结论知后半部分不等式成立. 此外, 对任意的 $\lambda \geq 0$, 由 $c(x^*)^\top \lambda^* = 0$ 有

$$\mathcal{L}(x^*, \lambda) - \mathcal{L}(x^*, \lambda^*) = c(x^*)^\top \lambda \leq 0$$

故前半部分不等式成立. ■

需要说明的是, 定理 7.6.1 中的凸性和 Slater 约束规范条件是保证存在非竖直支撑超平面的充分条件. 然而, 只要存在合适的超平面, 即使这些条件不成立, 仍可应用 Lagrange 技术给出最优解的描述, 详见 Luenberger 的专著^[16].

定理 7.6.1 没有完全揭示问题在 \mathbb{R}^{m+1} 中所表述的几何性质. 基于函数 $\omega(z)$ 的图像, 可以得到两个重要的性质, 即灵敏度和对偶性. 它们对于理论和应用都很重要.

定理 7.6.2 设定理 7.6.1 中的假设成立, 并且 x^* 是原始问题(7.6.1)的解, λ^* 是相应的 Lagrange 乘子, 则有

$$\omega(z) \geq \omega(0) + (-\lambda^*)^\top z, \quad \forall z \in \Gamma \quad (7.6.10)$$

证明 由定理 7.6.1, 有

$$x^* = \arg \min_{x \in X} [f(x) + \lambda^{* \top} c(x)]$$

和 $\lambda^{* \top} c(x^*) = 0$. 对任意的 $z \in \Gamma$, 存在 $x \in X$ 且 $c(x) \leq z$, 从而

$$f(x) + \lambda^{* \top} z \geq f(x) + \lambda^{* \top} c(x) \geq f(x^*) + \lambda^{* \top} c(x^*) = \omega(0)$$

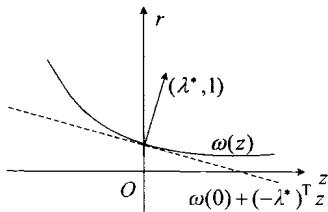


图 7.6.2 灵敏度

该不等式左边关于 x 取下确界, 由 $\omega(z)$ 的定义可以得到式(7.6.10). ■

定理的结论表明 $-\lambda^* \in \partial \omega(0)$, 其中 $\partial \omega(\cdot)$ 表示泛函的次微分. 进一步, 若 ω 是 Fréchet 可微的, 则 $\omega'(0) = -\lambda^*$ (见式(7.2.9)和习题 7.11), 结论的几何直观如图 7.6.2 所示.

7.7 对偶

本节主要考虑对偶问题, 可视作线性规划对偶理论(2.3节)的推广. 对偶的概念经常出现在数学规划的文献中, 其目的是为数学规划问题提供另一种更易计算或具有某些理论意义的表述. 关于该问题的一般性的对偶原理基于如图 7.7.1 所示的简单几何性质, 即函数 $\omega(z)$ 的上方图 $[\omega, \Gamma]$ (定义见习题 7.10) 的支撑超平面在纵轴上截距的最大值等于 $p^* = \omega(0)$, 且定理 7.6.1 中 Lagrange 乘子确定的超平面可以获得这个最大截距.

下面揭示这个重要的几何事实. $\forall \lambda \geq 0$, 定义对偶函数(dual function)

$$\varphi(\lambda) = \inf_{x \in X} \mathcal{L}(x, \lambda) = \inf_{x \in X} [f(x) + \lambda^\top c(x)] \quad (7.7.1)$$

通常, 对有些 $\lambda \geq 0$, $\varphi(\lambda)$ 有可能是无限的, 但它必是凹函数.

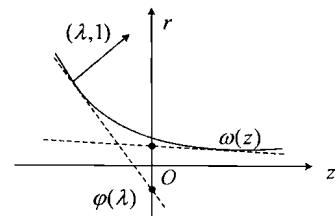


图 7.7.1 对偶性

命题 7.7.1 $\varphi(\lambda)$ 是凹的, 且可以表示为

$$\varphi(\lambda) = \inf_{(z, r) \in [\omega, \Gamma]} [r + \lambda^T z] = \inf_{z \in \Gamma} [\omega(z) + \lambda^T z] \quad (7.7.2)$$

其中 Γ 由式(7.6.3)定义.

证明 $\forall \lambda_0, \lambda_1 \geq 0$ 及 $\theta \in (0, 1)$ 有

$$\begin{aligned} \varphi((1-\theta)\lambda_0 + \theta\lambda_1) &= \inf_{x \in X} \{(1-\theta)[f(x) + \lambda_0^T c(x)] + \theta[f(x) + \lambda_1^T c(x)]\} \\ &\geq (1-\theta) \inf_{x \in X} [f(x) + \lambda_0^T c(x)] + \theta \inf_{x \in X} [f(x) + \lambda_1^T c(x)] \\ &= (1-\theta)\varphi(\lambda_0) + \theta\varphi(\lambda_1) \end{aligned}$$

其中的不等式是因为函数和的下确界大于等于各自下确界的和. 这样, $\varphi(\lambda)$ 是凹的.

对任意的 $\lambda \geq 0, z \in \Gamma$, 有

$$\begin{aligned} \varphi(\lambda) &= \inf_{x \in X} [f(x) + \lambda^T c(x)] \\ &\leq \inf\{f(x) + \lambda^T z : x \in X, c(x) \leq z\} \\ &= \omega(z) + \lambda^T z \end{aligned}$$

另一方面, $\forall x' \in X$, 令 $z' = c(x')$, 则 $z' \in \Gamma$, 且有

$$\begin{aligned} f(x') + \lambda^T c(x') &\geq \inf\{f(x) + \lambda^T z' : x \in X, c(x) \leq z'\} \\ &= \omega(z') + \lambda^T z' \geq \inf_{z \in \Gamma} [\omega(z) + \lambda^T z] \end{aligned}$$

上式两边关于 $x' \in X$ 取下确界, 得 $\varphi(\lambda) \geq \inf_{z \in \Gamma} [\omega(z) + \lambda^T z]$. 因此, 式(7.7.2)成立. ■

以向量 $(\lambda, 1) \in \mathbb{R}^{m+1}$ 作为法向量, 可以确定 \mathbb{R}^{m+1} 中的一簇超平面 $\{(z, r) : \lambda^T z + r = \gamma\}$, 其中 γ 是常数. 式(7.7.2)表明与 $\gamma = \varphi(\lambda)$ 对应的超平面 $\{(z, r) : \lambda^T z + r = \varphi(\lambda)\}$ 支撑集合 $[\omega, \Gamma]$, 即 ω 图形上方的区域; 进一步, 将 $z = 0$ 代入, 有 $r = \varphi(\lambda)$. 因此, $\varphi(\lambda)$ 等于这个超平面在纵轴的截距(如图 7.7.1 所示).

综上所述, 借助对偶函数可以精确表示对偶原理. 为此, 定义问题(7.6.1)的对偶问题为

$$\max_{\lambda \geq 0} \varphi(\lambda) \quad (7.7.3)$$

需要说明的是, 在前面的整个讨论中已经假定在原始问题(7.6.1)中, 所涉及的约束都是不等式. 然而, 如果有一些约束是等式, 那么在概念上将相应的结论推广没有任何困难. 如果要求某 $c_i(x) = 0$, 则对偶函数的定义(7.7.1)中与之对应的乘子 λ_i 没有符号限制. 在给出一般性的对偶原理之前, 先看两个例子.

例 7.7.1 (对偶函数有解析形式) 考虑

$$\begin{aligned} &\text{minimize} && x^2 \\ &\text{subject to} && x \geq 1 \end{aligned}$$

该问题的解 $x^* = 1$, Lagrange 函数 $\mathcal{L}(x, \lambda) = x^2 + \lambda(1-x)$. 对 $\lambda \geq 0$, 对偶函数 $\varphi(\lambda) = \min_x [x^2 + \lambda(1-x)]$. 易见极小点 $x(\lambda) = \lambda/2$. 这样, 对 $\lambda \geq 0$, 对偶函数 $\varphi(\lambda) = \lambda - \lambda^2/4$. 因此, 对偶问题是 $\max_{\lambda \geq 0} \lambda - \lambda^2/4$. 易于看到解 $\lambda^* = 2$, 它的确是与 x^* 对应的 Lagrange 乘子. 对偶问题的最优值 $\varphi(\lambda^*)$ 和 $f(x^*)$ 相等.

例 7.7.2 (对偶函数没有解析形式) 考虑

$$\begin{aligned} &\text{minimize} && e^x \\ &\text{subject to} && x^2 \leq 1 \end{aligned}$$

对 $\lambda \geq 0$, 对偶函数 $\varphi(\lambda) = \min_x [e^x + \lambda(x^2 - 1)]$. 函数 $e^x + \lambda(x^2 - 1)$ 关于 x 是凸的, 它的任一稳定点是全局极小点, 即对任给的 λ 值, 极小点是非线性方程 $e^x + 2x\lambda = 0$ 的解. 尽管可以用数值的方法得到这个解, 但是该解不能用 λ 显式表示. 这样, 对偶问题是 $\max_{\lambda \geq 0} e^x + \lambda(x^2 - 1)$, 其中 x 满足 $e^x + 2x\lambda = 0$. 等价地, 可以将该问题写成

$$\begin{aligned} & \underset{x, \lambda}{\text{maximize}} && e^x + \lambda(x^2 - 1) \\ & \text{subject to} && e^x + 2x\lambda = 0 \\ & && \lambda \geq 0 \end{aligned}$$

对偶函数不能显式表示出来的问题是普遍的, 通常对偶问题的表述可能包含原始变量和对偶变量. 如果将原始问题表示成

$$\begin{aligned} & \underset{x}{\text{minimize}} && e^x \\ & \text{subject to} && -1 \leq x \leq 1 \end{aligned}$$

则可以写出对偶函数的显式形式. 因此, 对偶问题的表述依赖于原始问题的特定表述方式, 原始问题的不同的等价表述对应可能完全不同, 甚至彼此不等价的对偶问题.

定理 7.7.1 (弱对偶性) 设 \hat{x} 是原始问题的可行解(即 $\hat{x} \in X, c(\hat{x}) \leq 0$), 对偶变量 $\hat{\lambda} \geq 0$, 则有 $f(\hat{x}) \geq \varphi(\hat{\lambda})$.

推论 1 $p^* \geq \max_{\lambda \geq 0} \varphi(\lambda)$, 即 $\min_{x \in X} \max_{\lambda \geq 0} \mathcal{L}(x, \lambda) \geq \max_{\lambda \geq 0} \min_{x \in X} \mathcal{L}(x, \lambda)$.

推论 2 如果原始问题是无界的, 则对所有的 $\lambda \geq 0$ 有 $\varphi(\lambda) = -\infty$ (因为对偶问题要极大化, 这种情况可以理解成对偶问题不可行); 如果对偶问题是无界的, 则原始问题是不可行的.

推论 1 表明原始问题的最优值大于等于对偶问题的最优值. 在例 7.7.1 中两个最优值是相等的. 就像在线性规划中那样, 这可能会使大家误认为“只要问题的最优解存在, 则原始问题、对偶问题的最优值是相等的”. 然而事实并不是这样的. 存在一些问题会具有对偶间隙 (dual gap), 即原始问题的最优目标值严格大于对偶问题的最优目标值, 这些问题通常是非凸规划.

例 7.7.3 (对偶间隙) 考虑

$$\begin{aligned} & \underset{x \in [0, 2]}{\text{minimize}} && -x^2 \\ & \text{subject to} && x = 1 \end{aligned}$$

显然解 $x^* = 1$, 最优值是 -1 . 记约束 $x = 1$ 的 Lagrange 乘子为 λ . 因为是等式约束, λ 没有非负限制. 对偶函数 $\varphi(\lambda) = \min_{0 \leq x \leq 2} -x^2 + \lambda(x - 1)$. 函数 $-x^2 + \lambda(x - 1)$ 在 \mathbb{R} 上没有局部极小点, 因此它将在 $x = 0$ 或者 $x = 2$ 处达到它在区间 $[0, 2]$ 上的极小值. 比较函数值有

$$\varphi(\lambda) = \begin{cases} -4 + \lambda, & \lambda \leq 2 \\ -\lambda, & \lambda \geq 2 \end{cases}$$

易见对偶函数在 $\lambda^* = 2$ 处取得最大值. 在该点处, 对偶问题的最优值是 -2 , 原始问题的最优值是 -1 , 两个最优值不相等. 对偶间隙 $f(x^*) - \varphi(\lambda^*) = 1$.

例 7.7.4 (对偶间隙无穷大) 考虑

$$\begin{aligned} & \underset{x}{\text{minimize}} && -x^2 \\ & \text{subject to} && 0 \leq x \leq 1 \end{aligned}$$

解 $x^* = 1$, 上下界约束的 Lagrange 乘子分别是 2 和 0. Lagrange 函数 $\mathcal{L}(x, \lambda) = -x^2 - \lambda_1 x +$

$\lambda_2(x-1)$, 且对所有的 $\lambda_1 \geq 0, \lambda_2 \geq 0$, 对偶函数 $\varphi(\lambda) = \min_x [-x^2 - \lambda_1 x + \lambda_2(x-1)] = -\infty$. 因此, 对偶问题的最优值是 $-\infty$.

定理 7.7.2 (强对偶性) 假设定理 7.6.1 中的条件成立, 即问题(7.6.1)是凸规划, 且 Slater 约束规范条件(7.6.5)成立. 如果 p^* 有限, 则有

$$p^* = \max_{\lambda \geq 0} \varphi(\lambda) \quad (7.7.4)$$

且最大值在某 λ^* 处可达. 若 x^* 是问题(7.6.1)的解, 则 $c(x^*)^T \lambda^* = 0$, 且 $x^* = \arg \min_{x \in X} \mathcal{L}(x, \lambda^*)$.

证明 由定理 7.7.1 的推论 1, 有 $p^* \geq \max_{\lambda \geq 0} \varphi(\lambda)$. 定理 7.6.1 表明存在 $\lambda^* \geq 0$ 使得 $\varphi(\lambda^*) = p^*$, 即等式(7.7.4). 其余结论由定理 7.6.1 给出. ■

7.8 半定规划

设 $\mathbf{X} \in S^n$, 这里 S^n 表示 $n \times n$ 阶对称矩阵集合. 可以将 \mathbf{X} 看作一个矩阵, 或者等价地看作一个具有 n^2 个分量的形如 $(x_{11}, x_{12}, \dots, x_{nn})^T$ 的向量, 也可以仅把 \mathbf{X} 看作空间 S^n 中的一个向量. 这 3 种对 \mathbf{X} 的不同理解都很有用. 记 \mathbf{X} 的线性函数 $\mathbf{C}(\mathbf{X}) = \mathbf{C} \cdot \mathbf{X} = \text{trace}(\mathbf{C}^T \mathbf{X}) = \sum_{i,j=1}^n c_{ij} x_{ij}$.

不失一般性, 也可以假设矩阵 \mathbf{C} 是对称的; 否则, 令 $\mathbf{C}' = (\mathbf{C} + \mathbf{C}^T)/2$, 则 \mathbf{C}' 为对称矩阵, 且 $\mathbf{C}'(\mathbf{X}) = \mathbf{C}(\mathbf{X})$. 半定规划 (Semi-Definite Programming, SDP) 是形如

$$\begin{aligned} & \underset{\mathbf{X} \in S^n}{\text{minimize}} \quad \mathbf{C} \cdot \mathbf{X} \\ & \text{subject to} \quad \mathbf{A}_i \cdot \mathbf{X} = b_i, \quad i = 1, 2, \dots, m \\ & \quad \mathbf{X} \geq \mathbf{0} \end{aligned} \quad (7.8.1)$$

的优化问题. 在问题(7.8.1)中, 变量是矩阵 \mathbf{X} , 目标函数是线性函数 $\mathbf{C} \cdot \mathbf{X}$, 且有 m 个 \mathbf{X} 必须满足的线性方程, 即 $\mathbf{A}_i \cdot \mathbf{X} = b_i, i = 1, 2, \dots, m$; $\mathbf{X} \geq \mathbf{0}$ 表示变量 \mathbf{X} 是半正定的, 即 \mathbf{X} 必须属于半正定对称矩阵(闭凸)锥 S_+^n . 注意, 确定问题(7.8.1)的数据有 $m+1$ 个对称矩阵 $\mathbf{C}, \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m$ 和 m 维向量 \mathbf{b} .

例 7.8.1 (半定规划) 对 $n=3, m=2$, 令

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 11 \\ 19 \end{bmatrix}$$

则变量 \mathbf{X} 将是 3×3 的对称矩阵

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix}$$

因此

$$\begin{aligned} \mathbf{C} \cdot \mathbf{X} &= x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33} \\ \mathbf{A}_1 \cdot \mathbf{X} &= x_{11} + 0x_{12} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} \\ \mathbf{A}_2 \cdot \mathbf{X} &= 0x_{11} + 4x_{12} + 16x_{13} + 6x_{22} + 0x_{23} + 4x_{33} \end{aligned}$$

这样,可以将该 SDP 问题写为

$$\begin{aligned} \text{minimize} \quad & x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 7x_{33} \\ \text{subject to} \quad & x_{11} + 2x_{13} + 3x_{22} + 14x_{23} + 5x_{33} = 11 \\ & 4x_{12} + 16x_{13} + 6x_{22} + 4x_{33} = 19 \\ & \mathbf{X} \geq \mathbf{0} \end{aligned}$$

注意,SDP 看起来和线性规划非常类似. 然而,这里用变量 \mathbf{X} 必须位于半正定锥代替了线性规划标准形中的 x 必须位于非负卦限. 正像 $x \geq \mathbf{0}$ 表明 n 个分量都必须是非负的那样, 将约束 $\mathbf{X} \geq \mathbf{0}$ 看作 \mathbf{X} 的 n 个特征值都必须是非负的, 有助于我们理解半定规划. 此外, 利用 \mathbf{X} 半正定当且仅当它的所有主子式皆大于等于零的事实, SDP 可以等价处理一类非凸非线性规划问题. 需要注意的是, 所有顺序主子式大于或者等于零是不能保证 \mathbf{X} 是半正定的. 比如

$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$$

就是一个反例.

下面说明线性规划是 SDP 的特例. 设确定线性规划问题的数据为 n 维向量 $\mathbf{c}, \mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^m$ 和 m 维向量 \mathbf{b} . 定义 $\mathbf{A}_i = \text{diag}(\mathbf{a}^i), i = 1, 2, \dots, m$; $\mathbf{C} = \text{diag}(\mathbf{c})$, 其中 $\text{diag}(\mathbf{c})$ 表示以向量 \mathbf{c} 为对角线元素的对角矩阵; \mathbf{E}_{ij} 表示 $e_{ij} = e_{ji} = 1$, 其余元素为零的 $n \times n$ 矩阵. 线性规划标准形可以表示为

$$\begin{aligned} \text{minimize} \quad & \mathbf{C} \cdot \mathbf{X} \\ \text{subject to} \quad & \mathbf{A}_i \cdot \mathbf{X} = b_i, \quad i = 1, 2, \dots, m \\ & \mathbf{E}_{ij} \cdot \mathbf{X} = 0, \quad i = 1, 2, \dots, n-1, \quad j = i+1, i+2, \dots, n \\ & \mathbf{X} \geq \mathbf{0} \end{aligned}$$

最后的解是 \mathbf{X} 的对角线元素. 当然, 在实际中, 人们从来不会把线性规划问题转化成 SDP 问题. 这里的转化仅仅是为了说明线性规划是 SDP 的一个特例.

7.8.1 半定规划的对偶理论

定义 SDP(7.8.1) 的对偶问题为(也可以由 Lagrange 对偶推导出)

$$\begin{aligned} \underset{\mathbf{y} \in \mathbb{R}^m}{\text{maximize}} \quad & \sum_{i=1}^m y_i b_i \\ \text{subject to} \quad & \sum_{i=1}^m y_i \mathbf{A}_i + \mathbf{S} = \mathbf{C} \\ & \mathbf{S} \geq \mathbf{0} \end{aligned} \tag{7.8.2}$$

也可以将问题(7.8.2)理解成: 给定 m 个乘子 y_1, y_2, \dots, y_m , 目标是极大化线性函数 $\sum_{i=1}^m y_i b_i$, 问题(7.8.2)的约束表明矩阵 $\mathbf{S} = \mathbf{C} - \sum_{i=1}^m y_i \mathbf{A}_i$ 必须是半正定的, 即 $\mathbf{C} - \sum_{i=1}^m y_i \mathbf{A}_i \geq \mathbf{0}$. 用例 7.8.1 来说明这种结构. 在该例中, 对偶问题是

$$\begin{aligned}
 & \text{maximize} \quad 11y_1 + 19y_2 \\
 & \text{subject to} \quad y_1 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix} + y_2 \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix} + \mathbf{S} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix} \\
 & \quad \mathbf{S} \geq \mathbf{0}
 \end{aligned}$$

可以将其写成

$$\begin{aligned}
 & \text{maximize} \quad 11y_1 + 19y_2 \\
 & \text{subject to} \quad \begin{bmatrix} 1 - 1y_1 - 0y_2 & 2 - 0y_1 - 2y_2 & 3 - 1y_1 - 8y_2 \\ 2 - 0y_1 - 2y_2 & 9 - 3y_1 - 6y_2 & 0 - 7y_1 - 0y_2 \\ 3 - 1y_1 - 8y_2 & 0 - 7y_1 - 0y_2 & 7 - 5y_1 - 4y_2 \end{bmatrix} \geq \mathbf{0}
 \end{aligned}$$

该问题的可行域和解如图 7.8.1 所示. 由图解法, 可得最优值 13.902 2 和解 $\mathbf{y}^* = (0.4847, 0.4511)^T$.

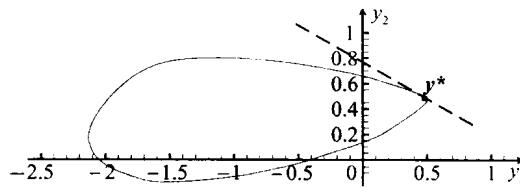


图 7.8.1 SDP 对偶问题的可行域和解

当半定规划以对偶形式(7.8.2)表示时, 因为变量是 m 个乘子, 所以经常更易于理解和处理. 下面的命题表明 SDP 的原始问题和对偶问题也满足弱对偶性.

命题 7.8.1 (弱对偶性) 给定问题(7.8.1)的可行解 \mathbf{X} 和问题(7.8.2)的可行解 (\mathbf{y}, \mathbf{S}) , 则

$$\mathbf{C} \cdot \mathbf{X} \geq \sum_{i=1}^m y_i b_i \quad (7.8.3)$$

如果 $\mathbf{S} \cdot \mathbf{X} = 0$, 则 \mathbf{X} 和 (\mathbf{y}, \mathbf{S}) 分别为问题(7.8.1)和问题(7.8.2)的最优解, 且 $\mathbf{S} \mathbf{X} = \mathbf{0}$.

证明 先证明: 如果 $\mathbf{S} \geq \mathbf{0}$ 且 $\mathbf{X} \geq \mathbf{0}$, 则 $\mathbf{S} \cdot \mathbf{X} \geq 0$. 设 $\mathbf{S} = \mathbf{U} \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \mathbf{U}^T$, 其中 \mathbf{U} 是正交矩阵, $\lambda_i \geq 0, i=1, 2, \dots, n$, 则有

$$\begin{aligned}
 \mathbf{S} \cdot \mathbf{X} &= \text{trace}(\mathbf{S} \mathbf{X}) = \text{trace}(\mathbf{U} \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \mathbf{U}^T \mathbf{X}) \\
 &= \text{trace}(\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \mathbf{U}^T \mathbf{X} \mathbf{U}) \\
 &= \sum_{i=1}^n \lambda_i (\mathbf{U}^T \mathbf{X} \mathbf{U})_{ii} \\
 &\geq 0
 \end{aligned} \quad (7.8.4)$$

其中最后一个不等式是因为 $\lambda_i \geq 0$, 且 $\mathbf{U}^T \mathbf{X} \mathbf{U}$ 是半正定矩阵, 从而对角线元素非负. 这里的证明利用了关于迹的两个基本等式, 详见习题 7.16. 而由可行性易验证 $\mathbf{C} \cdot \mathbf{X} = \sum_{i=1}^m y_i b_i = \mathbf{S} \cdot \mathbf{X}$, 所以不等式(7.8.3)成立.

如果 $\mathbf{S} \cdot \mathbf{X} = 0$, 则由上面的弱对偶性知 \mathbf{X} 和 (\mathbf{y}, \mathbf{S}) 是各自的最优解. 再由式(7.8.4)有 $\sum_{i=1}^n \lambda_i (\mathbf{U}^T \mathbf{X} \mathbf{U})_{ii} = 0$. 这蕴含着 $\lambda_i (\mathbf{U}^T \mathbf{X} \mathbf{U})_{ii} = 0 (i=1, 2, \dots, n)$, 即 $\lambda_i = 0$ 或者 $(\mathbf{U}^T \mathbf{X} \mathbf{U})_{ii} = 0$. 而

后一种情况蕴含着 $\mathbf{U}^\top \mathbf{X} \mathbf{U}$ 的第 i 行和第 i 列都为零. 设 $\mathbf{U}^\top \mathbf{X} \mathbf{U}$ 的行向量依次为 $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$, 易见

$$\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \mathbf{U}^\top \mathbf{X} \mathbf{U} = \begin{bmatrix} \lambda_1 \mathbf{d}_1 \\ \lambda_2 \mathbf{d}_2 \\ \vdots \\ \lambda_n \mathbf{d}_n \end{bmatrix}$$

则有 $\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \mathbf{U}^\top \mathbf{X} \mathbf{U} = \mathbf{0}$, 将该等式两边同时左乘 \mathbf{U} 、右乘 \mathbf{U}^\top , 得 $\mathbf{S} \mathbf{X} = \mathbf{0}$. ■

对于线性规划问题, 可以断言只要原始问题或对偶问题之一有最优解, 则另一个也必有最优解, 且最优值相等. 对于半定规划及其对偶问题, 需要假定某种正则性条件满足, 才可保证二者都可以取到最优值, 且无对偶间隙. 一个确保强对偶性成立的常用正则性条件是类似于式(7.6.5)的 Slater 约束规范条件. 这里仅给出强对偶性定理, 略去证明.

定理 7.8.1 (强对偶性) 设 p^* 和 d^* 分别表示问题(7.8.1)和问题(7.8.2)的最优值. 假设二者分别存在可行解 \mathbf{X}' 和 $(\mathbf{y}', \mathbf{S}')$ 使得 $\mathbf{X}' > \mathbf{0}, \mathbf{S}' > 0$. 则两个问题都将达到它们各自的最优值, 且 $p^* = d^*$.

7.8.2 最大割问题的 0.878 近似算法

对于一个给定的图 $\mathcal{G} = (\mathcal{N}, \mathcal{E}; \mathbf{w})$, 边 (i, j) 上的权值为 w_{ij} , 则最大割问题可以表述为

$$\begin{aligned} & \text{maximize} \quad \sum_{i=1}^n \sum_{j=1}^n w_{ij} \frac{1 - x_i x_j}{2} \\ & \text{subject to} \quad x_i^2 = 1, \quad i = 1, 2, \dots, n \end{aligned} \quad (7.8.5)$$

为了方便, 把它写成矩阵形式

$$\begin{aligned} & \text{maximize} \quad \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} \\ & \text{subject to} \quad x_i^2 = 1, \quad i = 1, 2, \dots, n \end{aligned}$$

其中 $q_{ij} = -w_{ij}$ ($i \neq j$) 且 $q_{ii} = \sum_{j=1}^n w_{ij}$, 也称为图 \mathcal{G} 的 Laplace 矩阵. 进一步, 如果 $w_{ij} \geq 0$, 则对任何 $i \neq j, q_{ij} \leq 0$. 因此, 由 $\mathbf{x}^\top \mathbf{Q} \mathbf{x} = \frac{1}{2} \sum_{i,j} w_{ij} (x_i - x_j)^2$ 有 $\mathbf{Q} \geq \mathbf{0}$.

最大割问题也可重新写为

$$\begin{aligned} & \text{maximize} \quad \frac{1}{2} \mathbf{Q} \cdot \mathbf{X} \\ & \text{subject to} \quad \mathbf{X} = \mathbf{x} \mathbf{x}^\top, \quad x_i = 1, \quad i = 1, 2, \dots, n \end{aligned} \quad (7.8.6)$$

或等价于

$$\begin{aligned} & \text{maximize} \quad \frac{1}{2} \mathbf{Q} \cdot \mathbf{X} \\ & \text{subject to} \quad \mathbf{X} \in \mathbf{S}_+^n, \quad \text{rank}(\mathbf{X}) = 1, \quad x_i = 1, \quad i = 1, 2, \dots, n \end{aligned}$$

若略去 rank-1 约束, 得到最大割问题(7.8.5)的半定规划松弛问题

$$\begin{aligned} & \text{maximize} \quad \frac{1}{2} \mathbf{Q} \cdot \mathbf{X} \\ & \text{subject to} \quad \mathbf{X} \in \mathbf{S}_+^n, \quad x_i = 1, \quad i = 1, 2, \dots, n \end{aligned} \quad (7.8.7)$$

但是如何才能得到原问题的解呢？此处随机化思想大有用武之地。假设 \mathbf{X}^* 是问题(7.8.7)的最优解。根据可行性要求， \mathbf{X}^* 的对角线元素都为 1。因为该解在松弛样本空间是最优的，所以 \mathbf{X}^* 的目标值肯定不小于原始问题的最优值。现在，生成一个期望向量为 $\mathbf{0}$ ，协方差矩阵为 \mathbf{X}^* 的正态分布 $\mathcal{N}(\mathbf{0}, \mathbf{X}^*)$ 。对矩阵 \mathbf{X}^* 进行 Cholesky 分解 $\mathbf{X}^* = \mathbf{L}\mathbf{L}^T$ ，则 $\mathbf{L}^T\mathcal{N}(\mathbf{0}, \mathbf{I})$ 恰好是 $\mathcal{N}(\mathbf{0}, \mathbf{X}^*)$ 。然后，从正态分布 $\mathcal{N}(\mathbf{0}, \mathbf{X}^*)$ 中随机抽取一个样本 $\boldsymbol{\xi}$ ，令 $x_i(\boldsymbol{\xi}) := \text{sign}(\xi_i)$ ，即对 $i=1, 2, \dots, n$ ，有

$$x_i(\boldsymbol{\xi}) = \begin{cases} +1, & \xi_i \geq 0 \\ -1, & \xi_i < 0 \end{cases}$$

这样，利用 \mathbf{X}^* 得到了一个抽样方案，即算法 7.8.1。

Algorithm 7.8.1 Randomized algorithm for SDP problem(7.8.6)

1: **SDP Relaxation.** Solve the SDP relaxation problem(7.8.7)，and let \mathbf{X}^* be an optimal solution. Let $\mathbf{X}^* = \mathbf{L}\mathbf{L}^T$ ；

2: **Randomized Sampling.** Take a random sample $\boldsymbol{\eta} \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ and let $\boldsymbol{\xi} = \mathbf{L}^T \boldsymbol{\eta}$ 。

3: **Rounding.** For $i=1, \dots, n$, let

$$x_i(\boldsymbol{\xi}) = \begin{cases} +1, & \xi_i \geq 0 \\ -1, & \xi_i < 0 \end{cases} \quad i = 1, 2, \dots, n$$

4: **Exit?** Check if the best sample so far is good enough or not. If not, then go to the Randomized Sampling step and continue with the sampling.

这种方法的问题是：怎样得到一个好的抽样方案？下面的分析着重于随机解的平均情况。实际上需要分析 $E[x_i(\boldsymbol{\xi})x_j(\boldsymbol{\xi})]$ 和 $E[\xi_i\xi_j]$ 之间的关系。如果 $i=j$ ，则 $E[x_i(\boldsymbol{\xi})^2] = E[1] = 1$ 。剩下需要分析的是 $i \neq j$ 的情况。此时

$$E[x_i(\boldsymbol{\xi})x_j(\boldsymbol{\xi})] = \Pr\{\xi_i \geq 0, \xi_j \geq 0\} + \Pr\{\xi_i < 0, \xi_j < 0\} - \Pr\{\xi_i \geq 0, \xi_j < 0\} - \Pr\{\xi_i < 0, \xi_j \geq 0\} \quad (7.8.8)$$

n 维(实)正态分布 $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 的密度函数为

$$f(\mathbf{z}) = \frac{1}{(2\pi)^{n/2} \sqrt{\det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right)$$

现在 $\begin{bmatrix} \xi_i \\ \xi_j \end{bmatrix}$ 服从正态分布 $\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & x_{ij}^* \\ x_{ij}^* & 1 \end{bmatrix}\right)$ ，因此

$$\Pr\{\xi_i \geq 0, \xi_j \geq 0\} = \int_{x \geq 0, y \geq 0} \frac{1}{2\pi \sqrt{1 - (x_{ij}^*)^2}} \exp\left(\frac{1}{2}(x, y)^T \begin{bmatrix} 1 & x_{ij}^* \\ x_{ij}^* & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}\right) dx dy$$

利用极坐标变换，经计算得

$$\begin{aligned} \Pr\{\xi_i \geq 0, \xi_j \geq 0\} &= \frac{1}{2\pi \sqrt{1 - (x_{ij}^*)^2}} \int_0^{\frac{\pi}{2}} \int_0^{\infty} \rho \exp\left(-\frac{1}{2(1 - (x_{ij}^*)^2)}(\rho^2 - 2\rho^2 x_{ij}^* \sin \theta \cos \theta)\right) d\rho d\theta \\ &= \frac{\sqrt{1 - (x_{ij}^*)^2}}{4\pi} \int_0^{\frac{\pi}{2}} \frac{1}{1 - x_{ij}^* \sin(2\theta)} d\theta \end{aligned}$$

$$= \frac{1}{4} + \frac{\arcsin x_{ij}^*}{2\pi}$$

由函数 $f(z)$ 的对称性, 有

$$\Pr\{\xi_i < 0, \xi_j < 0\} = \Pr\{\xi_i \geq 0, \xi_j \geq 0\} = \frac{1}{4} + \frac{\arcsin x_{ij}^*}{2\pi}$$

类似地

$$\begin{aligned} \Pr\{\xi_i \geq 0, \xi_j < 0\} &= \Pr\{\xi_i < 0, \xi_j \geq 0\} \\ &= \int_{x \geq 0, y < 0} \frac{1}{2\pi \sqrt{1 - (x_{ij}^*)^2}} \exp\left(-\frac{1}{2}(x, y)^T \begin{bmatrix} 1 & x_{ij}^* \\ x_{ij}^* & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}\right) dx dy \\ &= \frac{1}{4} - \frac{\arcsin x_{ij}^*}{2\pi} \end{aligned}$$

将上述计算结果代入式(7.8.8), 可得

$$E[x_i(\xi)x_j(\xi)] = \left(\frac{1}{2} + \frac{\arcsin x_{ij}^*}{\pi}\right) - \left(\frac{1}{2} - \frac{\arcsin x_{ij}^*}{\pi}\right) = \frac{2}{\pi} \arcsin x_{ij}^*$$

因此, 如果 $x_{ij}^* = 1$, 则 $E[x_i(\xi)x_j(\xi)] = 1$; 如果 $x_{ij}^* = -1$, 则 $E[x_i(\xi)x_j(\xi)] = -1$. 在其他所有情况下, 需利用如下的 Goemans 和 Williamson 发现的不等式来描述^[22].

引理 7.8.1 对所有 $x \in [-1, 1]$, 有

$$\frac{2}{\pi} \arcsin x \leq 1 - \alpha + \alpha x$$

其中 $\alpha = 0.87856\cdots$.

在最大割问题中, 因为所有的权值都是非负的, 所以可得

$$\begin{aligned} E\left[\sum_{i \neq j} w_{ij} \frac{1 - x_i(\xi_i)x_j(\xi_j)}{2}\right] &= \sum_{i \neq j} w_{ij} \frac{1 - \frac{2}{\pi} \arcsin x_{ij}^*}{2} \\ &\geq \sum_{i \neq j} w_{ij} \frac{1 - (1 - \alpha) - \alpha x_{ij}^*}{2} \\ &= \alpha \sum_{i \neq j} w_{ij} \frac{1 - x_{ij}^*}{2} \\ &\geq \alpha \times \text{最大割的目标值} \end{aligned}$$

最后一个不等式之所以成立, 是因为半定规划问题(7.8.7)是原始最大割问题(7.8.6)的松弛形式. 这就是著名的最大割问题的近似比为 0.878 的近似算法^[22].

7.8.3 半定规划的其他应用

SDP 在凸优化中也有非常广泛的应用. 可以用 SDP 进行建模的约束类型包括线性不等式、凸二次不等式、矩阵范数的下界、对称半正定矩阵行列式的下界、非负向量几何平均值的下界等. 利用这些或者其他构造, 可以将许多问题表述成半定规划, 比如线性规划、极小化凸二次函数受约束于凸二次不等式、极小化能覆盖给定点的椭球的容积、极大化包含在给定多面体中的椭球的容积, 以及极大化特征值和极小化特征值的变形问题等. 下面以两个典型问题为例, 说明如何用 SDP 来描述凸优化问题.

二次约束二次规划问题形如

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} + \mathbf{q}_0^T \mathbf{x} + c_0 \\ & \text{subject to} \quad \mathbf{x}^T \mathbf{Q}_i \mathbf{x} + \mathbf{q}_i^T \mathbf{x} + c_i \leq 0, \quad i = 1, 2, \dots, m \end{aligned}$$

该问题等价于

$$\begin{aligned} & \underset{\mathbf{x}, \theta}{\text{minimize}} \quad \theta \\ & \text{subject to} \quad \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} + \mathbf{q}_0^T \mathbf{x} + c_0 - \theta \leq 0 \\ & \quad \mathbf{x}^T \mathbf{Q}_i \mathbf{x} + \mathbf{q}_i^T \mathbf{x} + c_i \leq 0, \quad i = 1, 2, \dots, m \end{aligned}$$

对于凸二次约束二次规划问题, 其中 $\mathbf{Q}_i \geq \mathbf{0}, i = 0, 1, \dots, m$. 由 Cholesky 分解或特征值分解有 $\mathbf{Q}_i = \mathbf{M}_i^T \mathbf{M}_i$. 再由 Schur 补定理(见习题 7.18), 有

$$\begin{bmatrix} \mathbf{I} & \mathbf{M}_0 \mathbf{x} \\ \mathbf{x}^T \mathbf{M}_0^T & -c_0 - \mathbf{q}_0^T \mathbf{x} \end{bmatrix} \geq \mathbf{0} \Leftrightarrow \mathbf{x}^T \mathbf{Q}_0 \mathbf{x} + \mathbf{q}_0^T \mathbf{x} + c_0 \leq 0$$

于是可将凸二次约束二次规划问题表述为

$$\begin{aligned} & \underset{\mathbf{x}, \theta}{\text{minimize}} \quad \theta \\ & \text{subject to} \quad \begin{bmatrix} \mathbf{I} & \mathbf{M}_0 \mathbf{x} \\ \mathbf{x}^T \mathbf{M}_0^T & -c_0 - \mathbf{q}_0^T \mathbf{x} + \theta \end{bmatrix} \geq \mathbf{0} \\ & \quad \begin{bmatrix} \mathbf{I} & \mathbf{M}_i \mathbf{x} \\ \mathbf{x}^T \mathbf{M}_i^T & -c_i - \mathbf{q}_i^T \mathbf{x} \end{bmatrix} \geq \mathbf{0}, \quad i = 1, 2, \dots, m \end{aligned}$$

注意, 在这个 SDP 表述中, 变量是 \mathbf{x} 和 θ , 且所有的矩阵元素是 \mathbf{x} 和 θ 的线性函数.

一个典型的特征值优化问题是: 给定对称矩阵 $\mathbf{B}, \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k$, 确定权重 w_1, w_2, \dots, w_k 使得构建的矩阵 $\mathbf{S} = \mathbf{B} - \sum_{i=1}^k w_i \mathbf{A}_i$ 的最大特征值和最小特征值之差尽可能小. 可以将此问题表述为

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{S}}{\text{minimize}} \quad \lambda_1(\mathbf{S}) - \lambda_n(\mathbf{S}) \\ & \text{subject to} \quad \mathbf{S} = \mathbf{B} - \sum_{i=1}^k w_i \mathbf{A}_i \end{aligned} \tag{7.8.9}$$

其中 $\lambda_1(\mathbf{S})$ 和 $\lambda_n(\mathbf{S})$ 分别表示 \mathbf{S} 的最大特征值和最小特征值. 现在说明如何将此问题转化成一个 SDP 问题. 条件 $\lambda \mathbf{I} \leq \mathbf{S} \leq \mu \mathbf{I}$ 表示 $\mathbf{S} - \lambda \mathbf{I} \geq \mathbf{0}, \mu \mathbf{I} - \mathbf{S} \geq \mathbf{0}$. 设 \mathbf{S} 的特征值为 λ_i , 则 $\mathbf{S} - \lambda \mathbf{I}$ 和 $\mu \mathbf{I} - \mathbf{S}$ 的特征值分别为 $\lambda_i - \lambda$ 和 $\mu - \lambda_i$. 故 $\mathbf{S} - \lambda \mathbf{I} \geq \mathbf{0}$ 等价于 $\lambda_n(\mathbf{S}) \geq \lambda$, $\mu \mathbf{I} - \mathbf{S} \geq \mathbf{0}$ 等价于 $\lambda_1(\mathbf{S}) \leq \mu$. 从而 $\lambda \mathbf{I} \leq \mathbf{S} \leq \mu \mathbf{I}$ 等价于 $\lambda \leq \lambda_n(\mathbf{S}) \leq \lambda_1(\mathbf{S}) \leq \mu$. 因此, 特征值优化问题(7.8.9)可以写成

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{S}, \mu, \lambda}{\text{minimize}} \quad \mu - \lambda \\ & \text{subject to} \quad \mathbf{S} = \mathbf{B} - \sum_{i=1}^k w_i \mathbf{A}_i \\ & \quad \lambda \mathbf{I} \leq \mathbf{S} \leq \mu \mathbf{I} \end{aligned}$$

这是一个半定规划问题. 利用如上的构造方法, 也可以将许多其他类型的特征值优化问题表述成半定规划问题.

7.9 评注与参考

约束优化在经济学中扮演重要角色。比如消费者的选择问题可以表述为在预算约束下极大化效用函数，此时的 Lagrange 乘子从经济学上可以解释为预算约束的影子价格，即收入的边际效用。在最优控制理论中，Pontryagin 极小值原理将 Lagrange 函数重新表述为 Hamilton 泛函，将 Lagrange 乘子解释为 **协态变量** (costate variable)。此时，解是 Hamilton 泛函的局部极小点。

凡是涉及优化的著作都会讨论最优化条件，这里采用的是文献[14]的第9章；凸规划最优化条件的描述参考了文献[16]的第8章；半定规划的介绍取材于麻省理工学院(MIT)“Non-linear Programming”的课程资料。

为了方便大家应用对偶理论，这里不加证明地给出不等式约束中有线性函数时的强对偶性。

定理 7.9.1 (强对偶性) 考虑问题

$$\begin{aligned} & \underset{\mathbf{x} \in X}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad c_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, k \\ & \quad c_i(\mathbf{x}) = 0, \quad i = k+1, k+2, \dots, m \end{aligned} \quad (7.9.1)$$

其中 $X \subseteq \mathbb{R}^n$ 是凸集， f 是凸函数， $c_i, i=1, 2, \dots, r (\leq k)$ 是凸的非线性函数， $c_{r+1}, c_{r+2}, \dots, c_m$ 是线性函数。令

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i c_i(\mathbf{x}), \quad \varphi(\boldsymbol{\lambda}) = \min_{\mathbf{x} \in X} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$$

假设弱 **Slater 约束规范** 条件成立，即存在 $\mathbf{x}' \in X$ 使得

$$\left. \begin{array}{l} c_i(\mathbf{x}') < 0, \quad i = 1, 2, \dots, r \\ c_i(\mathbf{x}') \leq 0, \quad i = r+1, r+2, \dots, k \\ c_i(\mathbf{x}') = 0, \quad i = k+1, k+2, \dots, m \end{array} \right\} \quad (7.9.2)$$

如果问题(7.9.1)的最优值 p^* 有限，则有

$$p^* = \max_{\lambda_i \geq 0, i=1, 2, \dots, k} \varphi(\boldsymbol{\lambda})$$

且上式的最大值在某 $\boldsymbol{\lambda}^*$ 处可达。若问题(7.9.1)在 \mathbf{x}^* 取到最小值，则 $\lambda_i^* c_i(\mathbf{x}^*) = 0, i=1, 2, \dots, k$ ，且 $\mathbf{x}^* = \arg \min_{\mathbf{x} \in X} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*)$ 。

需要指出的是，如果原始问题(7.6.1)不是凸规划，那么尽管强对偶性不再成立，但命题7.7.1和弱对偶性依然成立，即对偶问题仍然为凸规划，且下界性质依然成立。比如对于整数规划问题，式(7.6.1)中的 X 可能是 \mathbb{R}^n 中具有整数分量的点集(这种选择权在设计求解整数规划的方法中非常有用)。此外，对偶问题仅有界约束，在很多时候容易求解，如例3.3.1。

除了这里介绍的 Lagrange 对偶外，还有 Fenchel 对偶和 Wolfe 对偶等；此外，本节介绍的最优化条件和对偶理论对于一般向量空间中的泛函优化问题也是成立的，详见参考文献[16]。

源自电路设计、网络优化和管理科学等许多领域的实际问题构造出的数学问题，很多都是 NP-完全问题。除了 $NP=P$ 之外，这些问题都不存在多项式时间算法，所以研究近似算法有

着实际的和潜在的应用价值. Goemans 和 Williamson 利用半定规划松弛对最大割问题给出了 0.878 的近似算法^[22], 突破了维持 30 多年的近似比为 0.5 的结论. 而不用半定规划工具, 直到最近, 才有学者获得近似比为 0.531 的算法, 并由 Soto 改进到 0.614.

非线性规划和组合优化两个领域的研究工具是如此的不同, 而半定规划像个超级明星, 把二者给联系起来了! 尽管半定规划在 20 世纪 70 年代就出现了, 但当时无法求解, 直到内点法发展起来^[21], 在 20 世纪 90 年代才成为研究热点.

鲁棒控制中的线性矩阵不等式(Linear Matrix Inequality, LMI)是以 Boyd 和 Gahinet^[23]等的成果为标志而发展形成的. 由于它在理论分析、矩阵条件转换以及设计满足多项要求的控制器等方面有着独特的优点, 一经形成, 便立即得到广泛的重视. 尤其值得一提的是, 由美国 Mathworks 公司开发的计算软件 Matlab 中专门推出了 LMI 工具箱^[25], 从而为解决实际的工程设计问题带来了很大的方便. 近年来, LMI 方法几乎被应用到控制理论的各个分支, 特别是鲁棒 H_∞ 控制, 更是吸引了众多学者的注意, 并取得了丰硕的成果. 求解 LMI 相关问题需要借助于求解半定规划的内点法. 目前, 求解半定规划的流行软件有 SDPT3 和 Sedumi 等. CVX 和 YALMIP 是两个有名的外包接口软件, 利用它们也可以在 Matlab 环境求解半定规划问题.

习题 7

7.1 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^2}{\text{minimize}} && (x_1 - 4)^2 + (x_2 - 6)^2 \\ & \text{subject to} && -x_1^2 + x_2 \geq 0 \\ & && x_2 \leq 4 \end{aligned}$$

写出最优解的必要条件, 并验证点 $(2, 4)^\top$ 是否满足. 它是最优解吗? 给出理由.

7.2 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^3}{\text{minimize}} && 3x_1 - x_2 + x_3^2 \\ & \text{subject to} && x_1 + x_2 + x_3 \leq 0 \\ & && -x_1 + 2x_2 + x_3^2 = 0 \end{aligned}$$

(a) 写出 KKT 条件;

(b) 说明该问题无界.

7.3 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^2}{\text{minimize}} && \left(x_1 - \frac{9}{4}\right)^2 + (x_2 - 2)^2 \\ & \text{subject to} && -x_1^2 + x_2 \geq 0 \\ & && x_1 + x_2 \leq 6 \\ & && x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

(a) 写出 KKT 条件, 并验证点 $x^* = (3/2, 9/4)^\top$ 满足这些条件.

(b) 给出 \mathbf{x}^* 处 KKT 条件的几何解释.

(c) 说明 \mathbf{x}^* 是该问题的最优解.

7.4 计划修建一个长 x_1 、高 x_2 、宽 x_3 (单位为 m), 容积 1500 m^3 的仓库. 每平方米的修建费用是: 墙 4 元, 屋顶 6 元, 地板加地面处理共 12 元. 由于美学原因, 长应该是高的两倍. 为了寻找花费最小的设计方案, 完成以下工作.

(a) 将该问题表述成优化问题, 写出 KKT 条件, 并由 KKT 条件确定解 \mathbf{x}^* 和 Lagrange 乘子 $\boldsymbol{\lambda}^*$.

(b) 从所得优化问题中消去 x_1 和 x_3 , 说明距所得解最近的整数 $x_2 = 10$ 使得费用最小, 然后回代算出 x_1 和 x_3 .

(c) 设容积约束为 $c_1(\mathbf{x}) = 0$. 在问题中将容积约束变成 $c_1(\mathbf{x}) = \epsilon$ 时, 用(a)中的方法求出 $f(\mathbf{x})$ 在所得解处的改变量 $h(\epsilon)$ 并验证 $h'(0) = -\lambda_1^*$; 计算目标值的改变量 $h(-150)$, 即将所需容积缩减 10% 时成本的改变量; 比较 $h(-150)$ 与由 Lagrange 乘子得到的估计值 $-\lambda_1^* \epsilon$.

7.5 考虑问题

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && \sum_{j=1}^n \frac{c_j}{x_j} \\ & \text{subject to} && \sum_{j=1}^n a_j x_j = b \\ & && x_j \geq 0, \quad j = 1, 2, \dots, n \end{aligned}$$

假设其中的 $c_j, a_j (j = 1, 2, \dots, n)$ 和 b 全为正常数. 写出该问题的 KKT 条件, 给出该问题解的解析表达式.

7.6 考虑抛物线上哪个点离原点最近(在 Euclidean 范数意义下)的问题. 可以将该问题表述为

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(x, y) = x^2 + y^2 \\ & \text{subject to} && (x - 1)^2 = y^2 \end{aligned}$$

(a) 用图解法求解该问题. 消去 y 求解问题, 所得到的函数有极小点吗? 给出结果不一致的可能原因. 消去 x 求解问题, 得到怎样的结果?

(b) 对于该问题, 找到所有的 KKT 点. LICQ 成立吗? 这些点中的哪一些是解?

7.7 验证 $\mathbf{x}^* = (0, 1)^T$ 是问题

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} && f(\mathbf{x}) = -2x_1 + x_2 \\ & \text{subject to} && -(1 - x_1)^3 + x_2 \leq 0 \\ & && 1 - x_2 - 0.25x_1^2 \leq 0 \end{aligned}$$

的解. LICQ 在该点成立吗? KKT 条件满足吗?

7.8 证明: 如果式(7.2.3)中的矩阵 \mathbf{A}^* 是列满秩的, 则唯一的 Lagrange 乘子可以由 $\boldsymbol{\lambda}^* = -\mathbf{A}^{*+} \mathbf{g}^*$ 确定, 或者求解子方程组 $\mathbf{A}_1^* \boldsymbol{\lambda}^* = -\mathbf{g}_1$, 其中 \mathbf{A}_1^* 是 \mathbf{A}^* 的非奇异子矩阵. 如果知道矩阵分解 $\mathbf{A}^* = \mathbf{QR}$ (\mathbf{Q} 是正交矩阵, \mathbf{R} 是上三角矩阵), 则计算前者非常稳定, 且利用 $\mathbf{R}\boldsymbol{\lambda}^* = -\mathbf{Q}^T \mathbf{g}^*$ 进行回代即可求得 $\boldsymbol{\lambda}^*$.

7.9 利用一阶和二阶最优性条件找到函数 $f(\mathbf{x}) = x_1 x_2$ 在单位圆 $x_1^2 + x_2^2 = 1$ 上的极小点. 从

几何上求解该问题.

7.10 设 f 是定义在凸集 $C \subseteq \mathbb{R}^n$ 上的函数, 定义集合

$$[f, C] = \{(x, r) \in \mathbb{R}^n \times \mathbb{R} : x \in C, f(x) \leq r\}$$

称该集合为 f 在 C 上的上方图(epigraph). 证明 $f(x)$ 是凸函数当且仅当 $[f, C]$ 是凸集.

7.11 设 f 是定义在 \mathbb{R}^n 上的函数. 如果向量 $p \in \mathbb{R}^n$ 满足

$$f(y) \geq f(x) + p^T(y - x), \quad \forall y \in \mathbb{R}^n$$

称 p 是 f 在 x 处的次梯度(subgradient); f 在 x 处所有次梯度的集合记为 $\partial f(x)$, 称为 f 在 x 处的次微分(subdifferential).

(a) 证明如果 f 在 x 处可微, 则 $\partial f(x) = \{\nabla f(x)\}$.

(b) 设 $f(x) = |x|$, 求 $\partial f(x)$.

7.12 考虑问题

$$\begin{aligned} & \underset{x \in X = [0, \infty)}{\text{minimize}} && f(x) = -\sqrt{x} \\ & \text{subject to} && x \leq 0 \end{aligned}$$

写出该问题的 Lagrange 对偶问题.

7.13 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && f(x) = c^T x \\ & \text{subject to} && Ax \geq b \\ & && x \geq 0 \end{aligned}$$

分别基于集合约束 $x \in \{x \in \mathbb{R}^n \mid x \geq 0\}$ 和 $x \in \mathbb{R}^n$ 写出该问题的对偶问题.

7.14 考虑问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && f(x) = \frac{1}{2} x^T G x + d^T x \\ & \text{subject to} && Ax \leq b \end{aligned}$$

其中 G 是 $n \times n$ 的对称正定矩阵.

(a) 写出该问题的对偶问题;

(b) 说明该问题的对偶问题的对偶是原始问题.

7.15 写出问题

$$\begin{aligned} & \underset{x_1, x_2}{\text{minimize}} && \frac{1}{2} \sigma x_1^2 + \frac{1}{2} x_2^2 + x_1 \\ & \text{subject to} && x_1 \geq 0 \end{aligned}$$

的 Lagrange 对偶问题. 分 $\sigma = 1$ 和 $\sigma = -1$ 两种情况讨论对偶问题的解是否是原问题最优解处的 Lagrange 乘子, 并解释两种情况下的结果.

7.16 对两个矩阵 $A, B \in \mathbb{R}^{k \times l}$ 定义

$$A \cdot B := \sum_{i=1}^k \sum_{j=1}^l a_{ij} b_{ij}$$

其中 a_{ij} 和 b_{ij} 分别为 A 和 B 的第 (i, j) 个元素. 证明

(a) $A \cdot B = \text{trace}(A^T B)$;

$$(b) \operatorname{trace}(\mathbf{A}^T \mathbf{B}) = \operatorname{trace}(\mathbf{B}^T \mathbf{A}).$$

7.17 设 λ_n 表示对称矩阵 Q 的最小特征值. 说明下面的 3 个优化问题的最优值均是 λ_n .

$$(a) \underset{\mathbf{d}}{\operatorname{minimize}} \quad \mathbf{d}^T \mathbf{Q} \mathbf{d}$$

$$\text{subject to} \quad \mathbf{d}^T \mathbf{d} = 1$$

$$(b) \underset{\lambda}{\operatorname{maximize}} \quad \lambda$$

$$\text{subject to} \quad \mathbf{Q} \geq \lambda \mathbf{I}$$

$$(c) \underset{\mathbf{X}}{\operatorname{minimize}} \quad \mathbf{Q} \cdot \mathbf{X}$$

$$\text{subject to} \quad \mathbf{I} \cdot \mathbf{X} = 1$$

$$\mathbf{X} \geq \mathbf{0}$$

7.18 考虑矩阵 $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}$, 其中 \mathbf{A}, \mathbf{C} 是对称矩阵, 且 \mathbf{A} 是正定的. 证明 $\mathbf{M} \geq \mathbf{0}$ 当且仅当 Schur 补 $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \geq \mathbf{0}$.

7.19 设 $\mathbf{r} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是光滑的向量值函数. 考虑 $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$, 其中

$$(a) f(\mathbf{x}) = \|\mathbf{r}(\mathbf{x})\|_\infty;$$

$$(b) f(\mathbf{x}) = \max\{r_i(\mathbf{x}), i=1, 2, \dots, m\};$$

$$(c) f(\mathbf{x}) = \|\mathbf{r}(\mathbf{x})\|_1.$$

请将这些(通常是非光滑的)问题重新表述成光滑的优化问题.

7.20 求出 $\min_{x \geq 0} \frac{1}{2} \left(\frac{1}{2} x^2 - 14 \right)^2 - x^2 + 3x$ 的解 x^* 和 \mathcal{A}^* . 去掉 x^* 处的非积极约束后再次求解问题. 给出你发现的结论.

第 8 章 约束优化: 线性约束规划

除线性规划外, 另一类能在有限步内求解的问题是二次规划(Quadratic Programming QP)问题. 其目标函数是二次的, 约束函数是线性的, 用问题(7.0.1)的形式来表述即

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad q(\mathbf{x}) := \frac{1}{2} \mathbf{x}^\top \mathbf{G} \mathbf{x} + \mathbf{d}^\top \mathbf{x} \\ & \text{subject to} \quad \mathbf{a}_i^\top \mathbf{x} = b_i, \quad i \in \mathcal{E} \\ & \quad \mathbf{a}_i^\top \mathbf{x} \leq b_i, \quad i \in \mathcal{I} \end{aligned} \quad (8.0.1)$$

其中 \mathbf{G} 是 $n \times n$ 阶对称矩阵, $\mathbf{d}, \mathbf{a}_i \in \mathbb{R}^n$. 与线性规划相同, 问题有可能是不可行的或者无界的, 这些情况皆可在算法中识别出来, 因此通常假设解是存在的. 如果 Hessian 阵 \mathbf{G} 是半正定的, 则问题(8.0.1)是凸规划, 从而 KKT 点是全局解; 进一步, 如果 \mathbf{G} 是正定的, 则 $q(\mathbf{x})$ 是严格凸的, 从而解是唯一的. 当 Hessian 阵 \mathbf{G} 不定时, 局部解有可能不是全局解. 本书着力介绍计算局部解的方法. 8.1 节说明怎样求解等式约束问题, 8.2 节将这些方法进行推广, 求解含不等式约束的凸二次规划问题. 最后需要指出的是, 鉴于二次规划问题的特殊结构, 除了自身很重要外, 它也经常以子问题的形式出现在一般约束优化的方法中, 诸如 9.4 节的逐步二次规划法.

接下来讨论约束函数仍然是线性的, 但目标函数任意的线性约束(linear constrained)优化问题, 即

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad \mathbf{a}_i^\top \mathbf{x} = b_i, \quad i \in \mathcal{E} \\ & \quad \mathbf{a}_i^\top \mathbf{x} \leq b_i, \quad i \in \mathcal{I} \end{aligned} \quad (8.0.2)$$

处理线性约束的方法大部分与二次规划中使用的一样, 但非二次的目标函数导致了另外的问题, 例如其求解方法通常不再具有有限步终止性质, 因此通常由某迭代序列 $\{\mathbf{x}^{(k)}\}$ 的极限来获得解 \mathbf{x}^* . 常用广义消元法求解等式约束问题 ($\mathcal{I} = \emptyset$), 从而其主要性质必然与第 4~6 章的无约束优化相似(见 8.3 节). 非等式约束可以利用积极集法处理(见 8.4 节), 它是二次规划中积极集法的推广. 这里新出现的问题是: 如何选定每次迭代中等式约束子问题的求解精度? 倘若精度选择不当, 则有可能出现锯齿现象, 这会使收敛速度显著降低. 8.5 节将介绍解决这个问题的方法. 另外, 在 8.4 节的最后, 还将介绍处理不等式约束的信赖域型方法.

8.1 等式约束二次规划

首先讨论等式约束二次规划, 问题形如

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad q(\mathbf{x}) := \frac{1}{2} \mathbf{x}^\top \mathbf{G} \mathbf{x} + \mathbf{d}^\top \mathbf{x} \\ & \text{subject to} \quad \mathbf{A}^\top \mathbf{x} = \mathbf{b} \end{aligned} \quad (8.1.1)$$

其中 $b \in \mathbb{R}^m$, A 是 $n \times m$ 阶矩阵, 其列向量是式(8.0.1)中的 $a_i, i \in \mathcal{E}$ 同时, 假定 A 的秩是 m (如果约束有冗余, 则去掉线性相关的约束). 进行该假定的好处是保证了与 x^* 相关的 Lagrange 乘子 λ^* 是唯一的. 后面的算法中同样要计算这些乘子(进行灵敏度分析和积极集法中都会用到).

求解问题(8.1.1)的一个最直接的方法是由等式约束消去部分变量. 定义剖分

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \quad G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

其中 $x_1 \in \mathbb{R}^m$, $x_2 \in \mathbb{R}^{n-m}$, 其余类似可得, 且 A_1 可逆. 于是式(8.1.1)中的方程组可以表示为 $A_1^T x_1 + A_2^T x_2 = b$, 利用高斯消元法, 即用 x_2 表示 x_1 , 得

$$x_1 = A_1^{-T} (b - A_2^T x_2) \quad (8.1.2)$$

代入 $q(x)$ 得关于 x_2 的二次函数

$$\begin{aligned} \psi(x_2) := & \frac{1}{2} x_2^T (G_{22} - G_{21} A_1^{-T} A_2^T - A_2 A_1^{-1} G_{12} + A_2 A_1^{-1} G_{11} A_1^{-T} A_2^T) x_2 \\ & + x_2^T (G_{21} - A_2 A_1^{-1} G_{11}) A_1^{-T} b + \frac{1}{2} b^T A_1^{-1} G_{11} A_1^{-T} b \\ & + x_2^T (d_2 - A_2 A_1^{-1} d_1) + d_1^T A_1^{-T} b \end{aligned}$$

这样, 就将问题(8.1.1)转化成函数 $\psi(x_2)$ 的无约束极小化. 如果 $\nabla^2 \psi$ 是正定的, 则解方程组 $\nabla \psi(x_2) = \mathbf{0}$ 得到唯一解 x_2^* , 将它代入式(8.1.2)得到 x_1^* . Lagrange 乘子向量由 $A \lambda^* = -g^*$ 确定, 其中 $g^* = \nabla q(x^*)$. 具体计算时求解剖分中的第一块 $A_1 \lambda^* = -g_1^*$ 即可. 因为 $g^* = Gx^* + d$, 因此可得 λ^* 的显式表达式是 $\lambda^* = -A_1^{-1}(d_1 + G_{11}x_1^* + G_{12}x_2^*)$. 下面用例子来说明该方法.

例 8.1.1 (直接消元) 考虑

$$\begin{aligned} \text{minimize} \quad & x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} \quad & x_1 + 2x_2 - x_3 = 4 \\ & x_1 - x_2 + x_3 = -2 \end{aligned}$$

为了消去 x_3 , 将约束方程写为

$$\begin{aligned} x_1 + 2x_2 &= 4 + x_3 \\ x_1 - x_2 &= -2 - x_3 \end{aligned}$$

用高斯消元法得到 $x_1 = -x_3/3$, $x_2 = 2 + 2x_3/3$. 这里使用的剖分为

$$\begin{aligned} x_1 &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, x_2 = x_3 \\ A_1 &= \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, A_2 = \begin{bmatrix} -1 & 1 \end{bmatrix} \\ G_{11} &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, G_{12} = G_{21}^T = \mathbf{0}, G_{22} = \mathbf{2}, d = \mathbf{0} \end{aligned}$$

将 x_1 带入 $q(x)$ 后, 得到 $\psi(x_3) = \frac{14}{9}x_3^2 + \frac{8}{3}x_3 + 4$, 易见 Hessian 阵 $\nabla^2 \psi(x_3) = \psi''(x_3) = \frac{28}{9} > 0$, 故有唯一解. 由 $\psi'(x_3) = 0$ 得 $x_3^* = -6/7$. 回代, 得到 $x_1^* = 2/7$, $x_2^* = 10/7$. 方程组 $A \lambda^* = -g^*$ 为

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^* \\ \lambda_2^* \end{bmatrix} = \frac{2}{7} \begin{bmatrix} -2 \\ -10 \\ 6 \end{bmatrix}$$

由前两个方程解得 $\lambda_1^* = -8/7, \lambda_2^* = 4/7$, 其自动满足第 3 个方程.

消元法并不是求解问题(8.1.1)的唯一方法,当然也不是最好的方法.另一种是广义消元(generalized elimination)法,它的本质是先对变量进行线性变换.令 \mathbf{Y} 和 \mathbf{Z} 分别是 $n \times m$ 阶与 $n \times (n-m)$ 阶矩阵,满足 $[\mathbf{Y} \quad \mathbf{Z}]$ 非奇异,且 $\mathbf{A}^T \mathbf{Y} = \mathbf{I}, \mathbf{A}^T \mathbf{Z} = \mathbf{0}$.可将矩阵 \mathbf{Y}^T 看作 \mathbf{A} 的左广义逆,因而 $\mathbf{x} = \mathbf{Y}\mathbf{b}$ 是方程组 $\mathbf{A}^T \mathbf{x} = \mathbf{b}$ 的一个特解.其他的可行解可以表示为 $\mathbf{x} = \mathbf{Y}\mathbf{b} + \mathbf{s}$,其中 \mathbf{s} 是齐次线性方程组 $\mathbf{A}^T \mathbf{s} = \mathbf{0}$ 的解,或者说属于 \mathbf{A} 的列零空间(null column space of \mathbf{A}),即

$$\mathbf{s} \in \{\mathbf{s} : \mathbf{A}^T \mathbf{s} = \mathbf{0}\} \quad (8.1.3)$$

矩阵 \mathbf{Z} 的列 $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{n-m}$ 即相当于该空间的一个基(也称既约坐标方向).也就是说,在任一可行点 \mathbf{x} 处的可行增量 \mathbf{s} 可以表示为

$$\mathbf{s} = \mathbf{Z}\mathbf{y} = \sum_{i=1}^{n-m} y_i \mathbf{z}_i \quad (8.1.4)$$

其中 y_1, y_2, \dots, y_{n-m} 是每个既约坐标方向的分量(或者既约变量),见图 8.1.1.于是任一可行点 \mathbf{x} 均可表示成

$$\mathbf{x} = \mathbf{Y}\mathbf{b} + \mathbf{Z}\mathbf{y} \quad (8.1.5)$$

如图 8.1.2 所示,这可以解释为:先从原点到可行点 $\mathbf{Y}\mathbf{b}$,之后进行可行校正 $\mathbf{Z}\mathbf{y}$ 后到达点 \mathbf{x} .这样,式(8.1.5)表示以 \mathbf{y} 为既约变量可将约束 $\mathbf{A}^T \mathbf{x} = \mathbf{b}$ 消去,因此是式(8.1.2)的推广.将式(8.1.5)代入 $q(\mathbf{x})$,得既约(reduced)二次函数

$$\psi(\mathbf{y}) := \frac{1}{2} \mathbf{y}^T \mathbf{Z}^T \mathbf{G} \mathbf{Z} \mathbf{y} + (\mathbf{d} + \mathbf{G} \mathbf{Y} \mathbf{b})^T \mathbf{Z} \mathbf{y} + \frac{1}{2} (\mathbf{2d} + \mathbf{G} \mathbf{Y} \mathbf{b})^T \mathbf{Y} \mathbf{b}$$

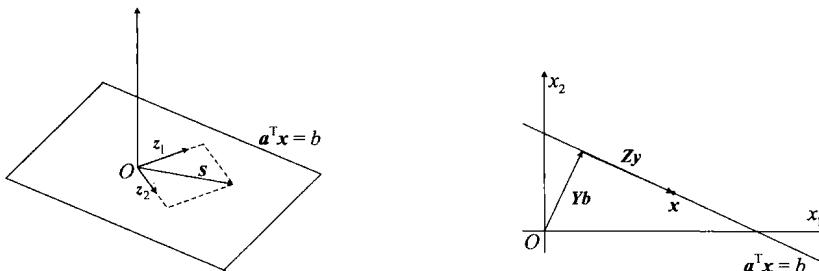


图 8.1.1 可行域的既约坐标

图 8.1.2 基于特殊形式(8.1.9)的广义消元法

如果 $\mathbf{Z}^T \mathbf{G} \mathbf{Z}$ 正定,则存在唯一的极小点 \mathbf{y}^* .令 $\nabla \psi(\mathbf{y}) = \mathbf{0}$,解线性方程

$$\mathbf{Z}^T \mathbf{G} \mathbf{Z} \mathbf{y} = -\mathbf{Z}^T (\mathbf{d} + \mathbf{G} \mathbf{Y} \mathbf{b}) \quad (8.1.6)$$

可确定 \mathbf{y}^* .为此,可计算 $\mathbf{Z}^T \mathbf{G} \mathbf{Z}$ 的 LL^T 分解或者 LDL^T 分解,这样做的好处是可以检验矩阵是否正定.最后得解 $\mathbf{x}^* = \mathbf{Y}\mathbf{b} + \mathbf{Z}\mathbf{y}^*$.通常称 $\mathbf{Z}^T \mathbf{G} \mathbf{Z}$ 是既约 Hessian 阵,称向量 $\mathbf{Z}^T (\mathbf{d} + \mathbf{G} \mathbf{Y} \mathbf{b})$ 是既约梯度. $q(\mathbf{x})$ 在 $\mathbf{x} = \mathbf{Y}\mathbf{b}$ 处的梯度 $\nabla q(\mathbf{Y}\mathbf{b}) = \mathbf{d} + \mathbf{G} \mathbf{Y} \mathbf{b}$,因此给它左乘 \mathbf{Z}^T 即得既约梯度向量.此外,将 $-\mathbf{g}^* = \mathbf{A}\lambda^*$ 左乘 \mathbf{Y}^T 可得 Lagrange 乘子

$$\lambda^* = -\mathbf{Y}^T \mathbf{g}^* \quad (8.1.7)$$

用这种方式可以解释许多方法,不同之处在于 \mathbf{Y} 和 \mathbf{Z} 的选取.一种特别重要的选取方式是用矩阵 \mathbf{A} 的任一 QR(比如基于 Householder 变换)分解,其可以表述为

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1 \mathbf{R} \quad (8.1.8)$$

其中 \mathbf{Q} 是 $n \times n$ 阶正交阵, \mathbf{R} 是 $m \times m$ 阶的可逆上三角阵. \mathbf{Q}_1 和 \mathbf{Q}_2 分别是 $n \times m$ 阶和 $n \times (n-m)$ 阶的. 然后选

$$\mathbf{Y} = \mathbf{A}^{+T} = \mathbf{Q}_1 \mathbf{R}^{-T}, \quad \mathbf{Z} = \mathbf{Q}_2 \quad (8.1.9)$$

即满足要求. 此外, 式(8.1.5)中的向量 \mathbf{Yb} 正交于约束流形 $\{x: \mathbf{A}^T x = \mathbf{b}\}$, 且既约坐标方向也是相互正交的, 见图 8.1.2. 建立并求解方程组(8.1.6)可以得到 \mathbf{y}^* , 进而得到解 \mathbf{x}^* . 对 $\mathbf{R}^T \mathbf{v} = \mathbf{b}$ 执行前向回代, 得到 $\mathbf{Yb} = \mathbf{Q}_1 \mathbf{v}$. 对

$$\mathbf{R} \boldsymbol{\lambda}^* = -\mathbf{Q}_1^T \mathbf{g}^* \quad (8.1.10)$$

执行后向回代可以计算出式(8.1.7)中的乘子 $\boldsymbol{\lambda}^*$. 这种格式是 Gill 和 Murray 于 1974 年提出的, 称为正交分解法(orthogonal factorization method), 其对舍入误差的传播比较稳定. 建议对稠密问题使用这种方法.

例 8.1.2 (正交分解法) 求解例 8.1.1, 先得到 \mathbf{A} 的 QR 分解, 即

$$\mathbf{A} = \begin{bmatrix} 1/\sqrt{6} & 4/\sqrt{21} & 1/\sqrt{14} \\ 2/\sqrt{6} & -1/\sqrt{21} & -2/\sqrt{14} \\ -1/\sqrt{6} & 2/\sqrt{21} & -3/\sqrt{14} \end{bmatrix} \begin{bmatrix} \sqrt{6} & -\sqrt{6}/3 \\ 0 & \sqrt{21}/3 \\ 0 & 0 \end{bmatrix}$$

取

$$\mathbf{Y} = \frac{1}{14} \begin{bmatrix} 5 & 8 \\ 4 & -2 \\ -1 & 4 \end{bmatrix}, \quad \mathbf{Z} = \frac{1}{\sqrt{14}} \begin{bmatrix} 1 \\ -2 \\ -3 \end{bmatrix}$$

此时 $\mathbf{Z}^T \mathbf{GZ} = 2$, 问题有唯一解. 向量 $\mathbf{Yb} = (2, 10, -6)^T / 7$. 因为 $\mathbf{d} = \mathbf{0}$ 及 $\mathbf{G} \mathbf{Yb} = 2\mathbf{Yb}$, 所以 $\mathbf{Z}^T(\mathbf{d} + \mathbf{G} \mathbf{Yb}) = \mathbf{0}$. 因此 $\mathbf{y}^* = \mathbf{0}$ 且 $\mathbf{x}^* = \mathbf{Yb} = (2, 10, -6)^T / 7$. 此外, $\mathbf{g}^* = \mathbf{d} + \mathbf{G} \mathbf{Yb} = 2(2, 10, -6)^T / 7$, 因此 $\boldsymbol{\lambda}^* = (-8/7, 4/7)^T$. 这些结果与前面用直接消元法得到的完全相同.

矩阵 \mathbf{Y} 和 \mathbf{Z} 的通用计算格式如下. 任选一个 $n \times (n-m)$ 的矩阵 \mathbf{V} 使得 $[\mathbf{A} \quad \mathbf{V}]$ 是非奇异的, 且逆可以分块表示为

$$[\mathbf{A} \quad \mathbf{V}]^{-1} = \begin{bmatrix} \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix} \quad (8.1.11)$$

其中 \mathbf{Y} 和 \mathbf{Z} 分别是 $n \times m$ 和 $n \times (n-m)$ 阶矩阵. 易验证 $\mathbf{Y}^T \mathbf{A} = \mathbf{I}$ 和 $\mathbf{Z}^T \mathbf{A} = \mathbf{0}$, 因此这些矩阵适用于广义消元法. 也可以利用这种格式解释前面的方法. 如果选取

$$\mathbf{V} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \quad (8.1.12)$$

则由恒等式

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_1^{-1} & \mathbf{0} \\ -\mathbf{A}_2 \mathbf{A}_1^{-1} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix} \quad (8.1.13)$$

可以给出 \mathbf{Y} 和 \mathbf{Z} 的表达式, 易见此时方法即还原成直接消元法. 另外, 如果选取

$$\mathbf{V} = \mathbf{Q}_2 \quad (8.1.14)$$

其中 \mathbf{Q}_2 由式(8.1.8)定义, 则由恒等式

$$[\mathbf{A} \quad \mathbf{V}]^{-1} = [\mathbf{Q}_1 \mathbf{R} \quad \mathbf{Q}_2]^{-1} = \begin{bmatrix} \mathbf{R}^{-1} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix}$$

得上面的正交分解法. 其实从理论的角度讲, 并不一定非要直接由式(8.1.11)显式形成 \mathbf{Y} 和 \mathbf{Z} . 有时利用 $[\mathbf{A} \quad \mathbf{V}]$ 方便且稳定的分解间接地产生 \mathbf{Y} 和 \mathbf{Z} 更可取些. 比如对于大规模问题或者更大规模的稀疏问题, 简单的公式(8.1.12)和公式(8.1.13)更优越些. 这种框架可以描述许多方法, 详见参考文献[14]的第 10 章. 尽管 \mathbf{Y} 和 \mathbf{Z} 有各种各样的选取方式, 然而正交分解法通常是首选的. 这是因为其一, 计算 \mathbf{Z} 时涉及正交矩阵的操作, 而正交矩阵具有良好的数值稳定性; 其二, 选择 $\mathbf{Z} = \mathbf{Q}_2$ 给出了条件数 $\kappa(\mathbf{Z}^T \mathbf{GZ})$ 的最好上界, 即

$$\kappa(\mathbf{Z}^T \mathbf{GZ}) \leq \kappa(\mathbf{G})$$

利用 Lagrange 乘子法(7.2.5), 可以给出另一种得到解 \mathbf{x}^* 和对应乘子 $\boldsymbol{\lambda}^*$ 的方式. 问题(8.1.1)的 Lagrange 函数(7.2.6)为

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{x}^T \mathbf{Gx} + \mathbf{d}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{A}^T \mathbf{x} - \mathbf{b})$$

稳定点条件(7.2.7)即方程

$$\nabla_{\mathbf{x}} \mathcal{L} = \mathbf{0} \Leftrightarrow \mathbf{Gx} + \mathbf{d} + \mathbf{A}\boldsymbol{\lambda} = \mathbf{0}$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L} = \mathbf{0} \Leftrightarrow \mathbf{A}^T \mathbf{x} - \mathbf{b} = \mathbf{0}$$

重新整理得线性方程组

$$\begin{bmatrix} \mathbf{G} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{d} \\ \mathbf{b} \end{bmatrix} \quad (8.1.15)$$

称系数矩阵为 Lagrange 矩阵, 也称 KKT 矩阵, 它是对称不定的. 如果逆矩阵存在, 可将其表示为

$$\begin{bmatrix} \mathbf{G} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{H} & \mathbf{T} \\ \mathbf{T}^T & \mathbf{U} \end{bmatrix} \quad (8.1.16)$$

则方程组(8.1.15)的解可以写为

$$\left. \begin{aligned} \mathbf{x}^* &= -\mathbf{Hd} + \mathbf{Tb} \\ \boldsymbol{\lambda}^* &= -\mathbf{T}^T \mathbf{d} + \mathbf{Ub} \end{aligned} \right\} \quad (8.1.17)$$

Fletcher 于 1971 年利用这些关系求解积极集法中出现的等式约束问题(8.2.1). 在这种情况下, 因为 $\mathbf{b} = \mathbf{0}$, 仅需要存储矩阵 \mathbf{H} 和 \mathbf{T} . 当 \mathbf{G}^{-1} 存在时, 利用分块矩阵的逆可以得 \mathbf{H} , \mathbf{T} 和 \mathbf{U} 的显式表达式为

$$\left. \begin{aligned} \mathbf{H} &= \mathbf{G}^{-1} - \mathbf{G}^{-1} \mathbf{A} (\mathbf{A}^T \mathbf{G}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{G}^{-1} \\ \mathbf{T} &= \mathbf{G}^{-1} \mathbf{A} (\mathbf{A}^T \mathbf{G}^{-1} \mathbf{A})^{-1} \\ \mathbf{U} &= -(\mathbf{A}^T \mathbf{G}^{-1} \mathbf{A})^{-1} \end{aligned} \right\} \quad (8.1.18)$$

注意, \mathbf{G}^{-1} 存在并不是 Lagrange 矩阵非奇异的必要条件. 此外, 这些公式是直接用逆表示的, 因此存在潜在的稳定性问题, 实际应用中几乎不会直接使用这些公式. 事实上, 利用式(8.1.11)定义的 \mathbf{Y} 和 \mathbf{Z} , Lagrange 矩阵的逆矩阵可以表示为

$$\left. \begin{aligned} \mathbf{H} &= \mathbf{Z} (\mathbf{Z}^T \mathbf{GZ})^{-1} \mathbf{Z}^T \\ \mathbf{T} &= \mathbf{Y} - \mathbf{Z} (\mathbf{Z}^T \mathbf{GZ})^{-1} \mathbf{Z}^T \mathbf{G} \mathbf{Y} \\ \mathbf{U} &= \mathbf{Y}^T \mathbf{GZ} (\mathbf{Z}^T \mathbf{GZ})^{-1} \mathbf{Z}^T \mathbf{G} \mathbf{Y} - \mathbf{Y}^T \mathbf{G} \mathbf{Y} \end{aligned} \right\} \quad (8.1.19)$$

由式(8.1.11)可以验证这些关系(见习题 8.5). 这样, 在 Lagrange 乘子法中也可以利用消元法中的 \mathbf{Y} , \mathbf{Z} 和 $\mathbf{Z}^T \mathbf{GZ}$ 的 Cholesky 分解 \mathbf{LL}^T 等计算方法来分解 Lagrange 矩阵.

总而言之, 基于 Lagrange 矩阵的逆的这些表示, 有两种平行的计算式(8.1.17)的方法. 最通用的是零空间法(null space method), 假定 \mathbf{Y}, \mathbf{Z} 和 Cholesky 分解 $\mathbf{Z}^T \mathbf{GZ} = \mathbf{L}\mathbf{L}^T$ 存在. 这种方法基于表述式(8.1.19)来计算式(8.1.17). 对应于不同的 \mathbf{Y} 和 \mathbf{Z} 的选取, 有不同的零空间法, 如上面介绍的广义消元法.

另一种方法是值空间法(range space method), 它要求 Hessian 阵是正定的, 并要利用 Hessian 阵的 Cholesky 分解, 比如 $\mathbf{G} = \mathbf{L}\mathbf{L}^T$. 值空间法与表述方式(8.1.18)相关. 有各种有效的数值方法可以实现它, 详见参考文献[14]的 10.2 节.

8.2 积极集法

大多数二次规划问题还包含不等式约束, 可以表示为问题(8.0.1). 本节描述如何利用积极集法(active set method)将等式约束问题的求解方法推广以求解不等式约束问题. 最常用的是原始(primal)积极集法. 为了叙述简洁, 假设 Hessian 阵正定, 这时有唯一的全局极小点, 并且避免了一些可能出现的困难.

在原始积极集法中, 将积极集(active set) \mathcal{A} 确定的约束看作等式, 而暂时忽略掉其余的, 并通过某种方式调整这个集合, 直到识别出问题(8.0.1)的解处的正确的积极约束. 因为目标函数是二次的, 从而积极约束的个数 m 满足 $0 \leq m \leq n$. 与此形成对比的是, 线性规划中 $m = n$, 即线性规划必可在顶点处取到极值, 顶点共有 n 个积极约束. 在第 k 次迭代, 可行点 $\mathbf{x}^{(k)}$ 满足 $\mathbf{a}_i^T \mathbf{x}^{(k)} = b_i, i \in \mathcal{A}$, 而除了退化情况外, $\mathbf{a}_i^T \mathbf{x}^{(k)} < b_i, i \notin \mathcal{A}$. 因此当前积极集等价于式(7.1.1)定义的积极约束集合 $\mathcal{A}^{(k)} = \mathcal{A}(\mathbf{x}^{(k)})$.

每次迭代求解一个仅包括积极约束的等式问题(Equality Problem, EP)的解. 最方便的做法是将原点平移到 $\mathbf{x}^{(k)}$, 找到问题

$$\begin{aligned} & \underset{\mathbf{s} \in \mathbb{R}^n}{\text{minimize}} && \frac{1}{2} \mathbf{s}^T \mathbf{G} \mathbf{s} + \mathbf{g}^{(k)T} \mathbf{s} \\ & \text{subject to} && \mathbf{a}_i^T \mathbf{s} = 0, \quad i \in \mathcal{A} \end{aligned} \quad (8.2.1)$$

的解 $\mathbf{s}^{(k)}$ 作为校正, 其中 $\mathbf{g}^{(k)} = \mathbf{G}\mathbf{x}^{(k)} + \mathbf{d}$ 是由式(8.0.1)定义的函数 $q(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 处的梯度, 即 $\nabla q(\mathbf{x}^{(k)})$. 这是一个等式约束二次规划问题, 可用 8.1 节中的任一方法求解. 检验试探点 $\mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ 是否满足那些不在 \mathcal{A} 中的约束条件. 如果满足, 则下一个迭代 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$; 否则, 试探点不可行, 即存在 $i \notin \mathcal{A}$ 使得 $c_i(\mathbf{x}^{(k)}) = \mathbf{a}_i^T \mathbf{x}^{(k)} - b_i + \mathbf{a}_i^T \mathbf{s}^{(k)} > 0$, 此时需要将 $\mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ 投影到可行域. 一种方法是将问题(8.2.1)的解 $\mathbf{s}^{(k)}$ 作为搜索方向 $\mathbf{p}^{(k)}$, 并选一个小于 1、但尽可能大的步长, 即选

$$\bar{\alpha}_k = \max \{ \alpha: \alpha > 0, \mathbf{a}_i^T \mathbf{x}^{(k)} - b_i + \alpha \mathbf{a}_i^T \mathbf{p}^{(k)} \leq 0, \mathbf{a}_i^T \mathbf{p}^{(k)} > 0, \forall i \notin \mathcal{A} \}$$

请读者思考为什么可以删去 $\mathbf{a}_i^T \mathbf{p}^{(k)} \leq 0$ 且不属于 \mathcal{A} 中的指标 i . 设在第 j 个约束取到上面的最大值, 即

$$\bar{\alpha}_k = \frac{b_j - \mathbf{a}_j^T \mathbf{x}^{(k)}}{\mathbf{a}_j^T \mathbf{p}^{(k)}} = \min_{\substack{i: i \notin \mathcal{A} \\ \mathbf{a}_i^T \mathbf{p}^{(k)} > 0}} \frac{b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}}{\mathbf{a}_i^T \mathbf{p}^{(k)}} \quad (8.2.2)$$

此时, 取下一次迭代 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \bar{\alpha}_k \mathbf{p}^{(k)}$. 注意, 此时非积极约束 j 变成积极的, 因此要将指标 j 添加到积极集 \mathcal{A} 中. 这样, 可将上面的两种情形统一表述为: 以问题(8.2.1)的解 $\mathbf{s}^{(k)}$ 为搜索方向 $\mathbf{p}^{(k)}$, 以

$$\alpha_k = \min(1, \bar{\alpha}_k) \quad (8.2.3)$$

为步长, 得新迭代 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$. 若 $\alpha_k < 1$, 则约束 j 变成积极的, 将其添到积极集 \mathcal{A} 中.

如果 $\mathbf{x}^{(k)}$ (即 $\mathbf{s} = \mathbf{0}$) 是当前等式问题(8.2.1)的解, 则可以按 8.1 节中的方法计算积极约束的乘子, 记为 $\lambda^{(k)}$, 即

$$\mathbf{g}^{(k)} + \sum_{i \in \mathcal{A}} \lambda_i^{(k)} \mathbf{a}_i = \mathbf{0}$$

又因为 $\mathbf{x}^{(k)}$ 是可行的, 且 $\mathbf{a}_i^T \mathbf{x}^{(k)} - b_i = 0, i \in \mathcal{A}$, 从而除了对偶可行性条件 $\lambda_i \geq 0, i \in \mathcal{I}$ 外, 向量 $\mathbf{x}^{(k)}$ 和 $\lambda^{(k)}$ 满足一阶条件(7.2.14)中的其他条件. 因此, 需要判定 $\lambda_i^{(k)} \geq 0, \forall i \in \mathcal{A} \cap \mathcal{I}$, 是否成立. 如果成立, 则一阶条件满足; 又因为 $q(\mathbf{x})$ 是凸的, 从而 $\mathbf{x}^{(k)}$ 是全局解. 否则, 存在指标, 设为 $q \in \mathcal{A} \cap \mathcal{I}$, 使得 $\lambda_q^{(k)} < 0$. 这时, 由 7.2 节关于灵敏度分析的讨论知, 将约束 q 变成非积极的可以使 $q(\mathbf{x})$ 的值减小. 于是从 \mathcal{A} 中去掉 q . 算法像前一次一样求解所得到的等式问题(8.2.1). 如果使得 $\lambda_q^{(k)} < 0$ 的指标有多个, 通常选

$$q = \arg \min_{i \in \mathcal{A} \cap \mathcal{I}} \lambda_i^{(k)} \quad (8.2.4)$$

这种选择既方便, 效果也好, 因而应用比较普遍. 给定可行点 $\mathbf{x}^{(0)}$, 并令 $\mathcal{A} = \mathcal{A}^{(0)}$ (这里 $\mathcal{A}^{(0)}$ 是 $\mathbf{x}^{(0)}$ 处的积极约束), 原始积极集法可以用伪码表示为算法 8.2.1. 下面用简单例子来说明该方法, 其几何直观见图 8.2.1.

Algorithm 8.2.1 Active-set method for solving quadratic programming problem(8.0.1)

```

1: Given  $\mathbf{x}^{(0)}$  and  $\mathcal{A}$ , set  $k=0$ ;
2: while 1 do
3:   if  $\mathbf{s} = \mathbf{0}$  solves the equality problem(8.2.1) then
4:     compute Lagrange multiplier  $\lambda^{(k)}$ ;
5:     find  $q$  to solve  $\min\{\lambda_i^{(k)} : i \in \mathcal{I} \cap \mathcal{A}\}$ ;
6:     if  $\lambda_q^{(k)} \geq 0$  then
7:       terminate with  $\mathbf{x}^* = \mathbf{x}^{(k)}$ .
8:     else
9:       remove  $q$  from  $\mathcal{A}$ ;
10:    end if
11:   end if
12:   solve the equality problem(8.2.1) for  $\mathbf{p}^{(k)}$ ;
13:   find  $\alpha_k$  with the formula(8.2.3) and set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ ;
14:   if  $\alpha_k < 1$  then
15:     add  $j$  to  $\mathcal{A}$ ;
16:   end if
17:   set  $k=k+1$ ;
18: end while

```

例 8.2.1 (积极集法) 考虑

$$\begin{aligned}
 & \underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad q(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 2.5)^2 \\
 & \text{subject to} \quad -x_1 + 2x_2 - 2 \leq 0 \\
 & \quad x_1 + 2x_2 - 6 \leq 0 \\
 & \quad x_1 - 2x_2 - 2 \leq 0 \\
 & \quad x_1 \geq 0 \\
 & \quad x_2 \geq 0
 \end{aligned}$$

其可行域如图 8.2.1 所示. 假设选取 $\mathbf{x}^{(0)} = (2, 0)^T$, 用 1 到 5 依次作为约束的指标. 在 $\mathbf{x}^{(0)}$ 处, 约束 3 和 5 是积极的, 置 $\mathcal{A} = \{3, 5\}$.

在 $\mathbf{x}^{(0)}$ 处, 有两个积极约束, 因此 $\mathbf{x}^{(0)}$ (即 $\mathbf{s} = 0$) 求解当前 EP(8.2.1). 解方程组

$$\begin{bmatrix} 1 \\ -2 \end{bmatrix} \lambda_3^{(0)} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \lambda_5^{(0)} = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$$

得 $(\lambda_3^{(0)}, \lambda_5^{(0)}) = (-2, -1)$. 因为约束 3 的乘子最小, 且是负的, 从而变成非积极的, 即从 \mathcal{A} 中去掉 3. 再次求解所得 EP(8.2.1), 得 $\mathbf{s}^{(0)} = (-1, 0)^T$, 因为 $\mathbf{x}^{(0)} + \mathbf{s}^{(0)} = (1, 0)^T$ 是可行的, 因此得下一次迭代 $\mathbf{x}^{(1)}$. 此时 $\mathbf{x}^{(1)}$ 求解当前的 EP(8.2.1), 计算得 $\lambda_5^{(1)} = -5$. 因此, 约束 5 变成非积极的, 这时 \mathcal{A} 变成空集. 再次求解当前的 EP(8.2.1), 得 $\mathbf{s}^{(1)} = (0, 2.5)^T$. 因为 $\mathbf{x}' = \mathbf{x}^{(1)} + \mathbf{s}^{(1)} = (1, 2.5)^T$ 不可行, 因此令搜索方向 $\mathbf{p}^{(1)} = (0, 2.5)^T$, 由式(8.2.2)计算步长, 得 $\alpha_1 = 0.6$, 从而得到最好的可行点 $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + 0.6\mathbf{s}^{(1)} = (1, 1.5)^T$, 且约束 1 变成积极的, 此时 $\mathcal{A} = \{1\}$. 再次求解当前 EP

(8.2.1), 得 $\mathbf{s}^{(2)} = (0.4, 0.2)^T$, 且 $\mathbf{x}^{(2)} + \mathbf{s}^{(2)} = (1.4, 1.7)^T$ 可行, 得新的迭代 $\mathbf{x}^{(3)}$. $\mathbf{x}^{(3)}$ 求解当前 EP(8.2.1), 计算乘子得 $\lambda_1^{(3)} = 0.8 > 0$. 因此终止算法, 得到解 $\mathbf{x}^* = (1.4, 1.7)^T$.

如果初始积极集 \mathcal{A} 中约束的梯度是线性无关的, 则对积极集的修正策略可以确保该性质对后面所有的积极集成立. 具体地, 若一个非积极约束变成积极的, 则它的法向量不能表示成当前积极集中法向量 \mathbf{a}_i 的线性组合, 即由式(8.2.2)确定出的 j 使得向量 $\mathbf{a}_j \in \mathcal{A}$, \mathbf{a}_j 线性无关 (见习题 8.9). 另一方面, 从积极集中删除指标一定不会导致线性相关性. 因此, EP(8.2.1) 总是适当的. 这样, 如果每步的 $\alpha_k \neq 0$, 此时 $q(\mathbf{x})$ 在每次迭代都会减小 (见习题 8.10), 由此通常可以证明算法的有限终止性. 终止证明依赖于: 必存在迭代子序列 $\{\mathbf{x}^{(k)}\}$, 其求解当前 EP. 仅当式(8.2.3)中的 $\alpha_k < 1$ 时, $\mathbf{x}^{(k+1)}$ 不是 EP 的解, 此时将指标 j 添加到 \mathcal{A} 中. 这种情况最多出现 n 次, 直到 $\mathbf{x}^{(k)}$ 是一个顶点, 其必求解当前的 EP. 然而可能的 EP 数目是有限的, 子序列中的 $\mathbf{x}^{(k)}$ 是每个 EP 的唯一的全局解, 且 $q(\mathbf{x}^{(k)})$ 单调递减, 由这些可知算法会迭代有限步终止. 如果某步长为零, 则 $q(\mathbf{x}^{(k)})$ 不会减小, 算法有可能返回序列中先前的某个积极集, 从而引起循环. 这是由约束集的退化引起的, 类似于线性规划的单纯形法中定义的退化. 另一种比较麻烦的情况是 $\alpha_k = 1$ 时, 恰好式(8.2.2)中有一个非积极约束变成积极的. 可以利用有关扰动的理论来打破这些平局, 从而在理论上避免出现循环, 然而实际上却存在一定的困难. 因此基本的算法通

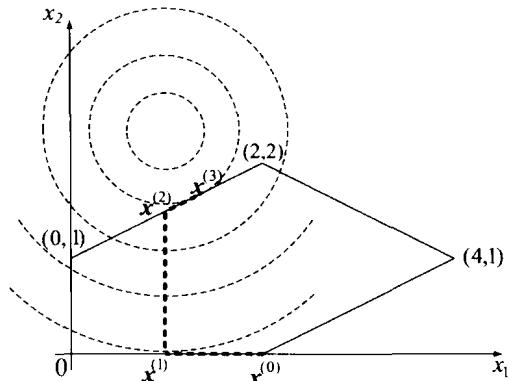


图 8.2.1 积极集法的迭代过程图示

常不考虑退化现象,即假设不会出现退化情况.

原始积极集法需要一个初始可行点 $\mathbf{x}^{(0)}$. 可以利用 2.2.5 小节介绍的技术,求解辅助问题得 $\mathbf{x}^{(0)}$. 另一种避免需要可行点的做法是将二次函数的常数倍加上两阶段法中第一阶段的目标函数,得到 ($\mathcal{E} = \emptyset$)

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \psi(\mathbf{x}) := \nu q(\mathbf{x}) + \sum_{i \in \mathcal{I}} \max(\mathbf{a}_i^T \mathbf{x} - b_i, 0) \quad (8.2.5)$$

对充分小的 ν ,求解该问题即可求解原始问题. 这是 ℓ_1 -QP 问题,是 ℓ_1 精确罚函数的一个具体应用例子,详细描述见 9.3 节. 对算法 8.2.1 稍作改变,可得求解问题(8.2.5)的积极集法. 首先将式(8.2.3)的线搜索变成找问题(8.2.5)的极小点,如果有一个约束 j 变成积极的,就像前面一样,将它加到 \mathcal{A} 中. 其次,问题(8.2.5)的最优性条件是存在乘子 λ 满足 $0 \leq \lambda_i \leq 1$ (见式(9.3.8)和 9.3 节). 这样需要将式(8.2.4)变成选取最违反这些条件的乘子 λ_i . 同样地,根据 $l_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - b_i < 0$ (或 > 0),非积极约束的乘子是 0 (或 1),且需要求解等式问题

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad \nu q(\mathbf{x}) + \sum_{i \in \mathcal{I}} \lambda_i l_i(\mathbf{x}) \\ & \text{subject to} \quad l_i(\mathbf{x}) = 0, \quad i \in \mathcal{A} \end{aligned}$$

这是一个等式约束二次规划问题,可以利用刚刚讲过的相关技术求解,包括合适的更新方法.

任何积极集法的重要特征都是:积极集改变后,要有效求解 EP(8.2.1). 在每次迭代中,不是重新分解 Lagrange 矩阵 ($O(n^3)$ 运算),而是根据 \mathcal{A} 的变化更新(update)算法所需要的因子 ($O(n^2)$ 运算). 不同的方法采用不同的矩阵分解,以保证这种更新是可行且有效的. 参考文献 [32] 和 [34] 是分别针对稠密 QP 和稀疏 QP 的相关技术的综述. 基于 8.1 节中的正交分解法的实现细节见参考文献[33].

8.3 线性等式约束规划

本节考虑线性等式约束问题

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad \mathbf{A}^T \mathbf{x} = \mathbf{b} \end{aligned} \quad (8.3.1)$$

其中 \mathbf{A} 是 $n \times m$ 矩阵, $\mathbf{b} \in \mathbb{R}^m$. 与 8.1 节相同,假设 $\text{rank}(\mathbf{A}) = m$. 利用 8.1 节中的广义消元法可将问题(8.3.1)既约为无约束问题. 为此,引入矩阵 \mathbf{Y}, \mathbf{Z} ,其中 $\mathbf{Y}^T \mathbf{A} = \mathbf{I}, \mathbf{Z}^T \mathbf{A} = \mathbf{0}$,且 $[\mathbf{Y} \quad \mathbf{Z}]$ 是非奇异的. 如果当前迭代点 $\mathbf{x}^{(k)}$ 是可行的,则可行点表示为

$$\mathbf{x} = \mathbf{x}^{(k)} + \mathbf{s} \quad (8.3.2)$$

其中 $\mathbf{s} = \mathbf{Zy}$ 是 \mathbf{A} 的列零空间中的一个可行校正(见式(8.1.3)和式(8.1.4)). 因此,问题(8.3.1)的一个等价形式是求解既约的(reduced)无约束问题

$$\underset{\mathbf{y} \in \mathbb{R}^{n-m}}{\text{minimize}} \quad \psi(\mathbf{y}) := f(\mathbf{x}^{(k)} + \mathbf{Zy}) \quad (8.3.3)$$

通常,为了计算方便及稳定性,最好在每次迭代中定义一个新的既约函数(8.3.3). 利用链式法则对 $\psi(\mathbf{y})$ 求导,得既约梯度向量

$$\nabla_{\mathbf{y}} \psi(\mathbf{y}) = \mathbf{Z}^T \mathbf{g}(\mathbf{x}) \quad (8.3.4)$$

和既约 Hessian 阵

$$\nabla_y^2 \psi(\mathbf{y}) = \mathbf{Z}^T \mathbf{G}(\mathbf{x}) \mathbf{Z} \quad (8.3.5)$$

可用第 4~6 章中介绍的任一合适的无约束优化技术求解既约问题(8.3.3). 由式(8.3.4)和式(8.3.5)知, \mathbf{x}^* 为局部最优解的充分条件是 $\mathbf{Z}^T \mathbf{g}^* = \mathbf{0}$ 和 $\mathbf{Z}^T \mathbf{G}^* \mathbf{Z}$ 是正定的, 而必要条件是 $\mathbf{Z}^T \mathbf{g}^* = \mathbf{0}$ 或 $\mathbf{Z}^T \mathbf{G}^* \mathbf{Z}$ 是半正定的. 易于证明这些条件与 7.2 节和 7.4 节所给出的是等价的(见习题 8.12). 在正交分解中, 可以取初始可行点 $\mathbf{x}^{(0)} = \mathbf{Yb}$, 它是离原点最近的可行点; 或者更一般地, 对任意给定的点 \mathbf{x}' , 取 $\mathbf{x}^{(0)} = \mathbf{x}' + \mathbf{Y}(\mathbf{b} - \mathbf{A}^T \mathbf{x}')$, 它是与 \mathbf{x}' 最近的可行点. 如同 5.1.2 小节那样, 先考虑基于二次模型的牛顿法, 这里的二次模型是 $\psi(\mathbf{y})$ 在 $\mathbf{y} = \mathbf{0}$ 处的二阶 Taylor 展式, 即

$$\psi(\mathbf{y}) \approx q^{(k)}(\mathbf{y}) := f^{(k)} + \mathbf{y}^T \mathbf{Z}^T \mathbf{g}^{(k)} + \frac{1}{2} \mathbf{y}^T \mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z} \mathbf{y} \quad (8.3.6)$$

由于 \mathbf{y} 空间的原点对应于 \mathbf{x} 空间的 $\mathbf{x}^{(k)}$ (见式(8.3.2)), 因此 $f^{(k)}$, $\mathbf{g}^{(k)}$, $\mathbf{G}^{(k)}$ 指相关量在点 $\mathbf{x}^{(k)}$ 处的值. $q^{(k)}(\mathbf{y})$ 表示第 k 次迭代得到的模型二次函数, 从而用基本牛顿法确定 $\mathbf{y}^{(k)}$ 来极小化 $q^{(k)}(\mathbf{y})$. 极小点唯一的充要条件是 $\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z}$ 正定, 令 $\nabla q^{(k)} = \mathbf{0}$, 即求解

$$(\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z}) \mathbf{y} = -\mathbf{Z}^T \mathbf{g}^{(k)} \quad (8.3.7)$$

得 $\mathbf{y}^{(k)}$. 此时, 再由式(8.3.2)知下次迭代是 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{Z} \mathbf{y}^{(k)}$. 若 $\mathbf{x}^{(k)}$ 离 \mathbf{x}^* 充分近, 则方法收敛, 且是二阶收敛的. 然而这种方法也可能不收敛, 此外在 $\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z}$ 不正定时无定义. 此时可采用线搜索, 并借助修正既约 Hessian 阵使其正定等方法来弥补这些不足. 后者包括信赖域法和 LM 法的思想, 详见 5.1.2 小节和第 6 章. 一个重要的事实是, 可以利用 6.2 节中对某 $\nu \geq 0$ 得到的修正 $\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z} + \nu \mathbf{I}$ 来处理信赖域约束 $\|\mathbf{y}\|_2 \leq \Delta_k$, 若是利用正交分解法来得到既约问题的解, 则该信赖域约束也对应于关于 \mathbf{x} 的信赖域约束 $\|\mathbf{x} - \mathbf{x}^{(k)}\|_2 \leq \Delta_k$. 注意, 其他广义消元法并不能保持这种对应关系.

通常情况下, 不可能或不方便提供二阶导数的计算公式, 因而考虑仅需要一阶导数信息, 甚至无需导数的方法尤为重要. 当一阶导数信息可用时, 一种做法是应用有限差分牛顿法(见 5.3 节). 此时, 最好在 \mathbf{y} 空间进行差分, 从而对足够小的 h , 既约 Hessian 阵的第 i 列定义为

$$\mathbf{Z}^T (\mathbf{g}(\mathbf{x}^{(k)} + z_i h) - \mathbf{g}^{(k)}) / h \quad (8.3.8)$$

然后对称化该矩阵, 再用所得矩阵代替式(8.3.7)中的 $\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z}$. 这样每次迭代仅需计算另外 $n-m$ 个梯度的值, 该方法适用于 $n-m$ 较小的情形. 特别地, 用这种方法来估计初始既约 Hessian 阵的近似也非常有效.

然而由于 5.3 节所述的诸多情况, 通常更愿意选取拟牛顿法求解既约问题(8.3.3). 在诸多建议中, Gill 和 Murray 的方法是最好的, 即用正定矩阵 $\mathbf{M}^{(k)}$ 来近似既约 Hessian 阵, 用分解形式可以表示为

$$\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z} \approx \mathbf{M}^{(k)} = \mathbf{L}^{(k)} \mathbf{D}^{(k)} \mathbf{L}^{(k)T} \quad (8.3.9)$$

用方程组

$$\mathbf{M}^{(k)} \mathbf{p} = -\mathbf{Z}^T \mathbf{g}^{(k)} \quad (8.3.10)$$

的解 $\mathbf{p} = \mathbf{p}^{(k)}$ 作为 \mathbf{y} 空间中的搜索方向, 这类似于式(8.3.7); 随之, 在直线 $\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)}$ ($\mathbf{d}^{(k)} = \mathbf{Z} \mathbf{p}^{(k)}$) 上进行搜索. 线搜索终止条件见 4.3 节和 4.4 节. 此外, 也可以像 5.3 节那样, 用近似的既约 Hessian 阵的逆矩阵 $\mathbf{H}^{(k)}$ 来表示 $\mathbf{M}^{(k)}$, 即

$$\mathbf{H}^{(k)} = \mathbf{M}^{(k)^{-1}} \approx (\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z})^{-1} \quad (8.3.11)$$

由此得搜索方向

$$\mathbf{d}^{(k)} = -\mathbf{Z}\mathbf{H}^{(k)}\mathbf{Z}^T\mathbf{g}^{(k)} \quad (8.3.12)$$

需要注意的是,式(8.3.11)的稳定性不比式(8.3.9)差,且使用更方便.每次迭代后,需要修正 $\mathbf{H}^{(k)}$ (或 $\mathbf{M}^{(k)}$) 的分解因子,以便纳入线搜索中获取的额外曲率信息.如 5.3 节所述,目前 BGFS 公式更流行些.根据式(5.3.11),需要计算的 $\mathbf{y}^{(k)}$ 和 $\mathbf{s}^{(k)}$ 在这里分别变为既约梯度之间的差和既约变量之间的差,即 $\mathbf{y}^{(k)} = \mathbf{Z}^T(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})$ 和 $\mathbf{s}^{(k)} = \mathbf{y}^{(k)} - \mathbf{0} = \mathbf{y}^{(k)}$. 可任意取正定阵作为初始矩阵 $\mathbf{H}^{(0)}$. 若没有任何其他信息,通常选取 $\mathbf{H}^{(0)} = \mathbf{I}$. 在 5.3 节中叙述的关于矩阵 $\mathbf{H}^{(k)}$ 的性质以及二次终止性(这里为 $n-m$ 步)等性质在这里也同样成立.

另一种可供选择的做法是应用共轭梯度法(如算法 5.2.2)求解既约问题,仅适用于大规模问题中矩阵 $\mathbf{H}^{(k)}$ 不便存储的情况.此时,存储矩阵 \mathbf{Y} 和 \mathbf{Z} 的方式也十分重要.最好的办法是通过分解矩阵 $[\mathbf{A} \quad \mathbf{V}]$ 间接地确定它们,以便尽可能地保持稀疏性.同样,选取的 \mathbf{V} 也要保持稀疏性,如式(8.1.12)中给出的用于消元法的例子.当问题的导数不可得时,5.3 节的经验表明利用差商近似导数的拟牛顿法是很可取的.最好在既约变量空间计算这些差商,从而对某个小区间 h ,用向前差商

$$[f(\mathbf{x}^{(k)} + \mathbf{z}, h) - f^{(k)}]/h \quad (8.3.13)$$

近似 $\mathbf{Z}^T\mathbf{g}^{(k)}$ 的第 i 个分量.这里仅需计算额外的 $n-m$ 个函数值.邻近最优解时,利用中心差商格式

$$\frac{1}{2} [f(\mathbf{x}^{(k)} + \mathbf{z}_i h) - f(\mathbf{x}^{(k)} - \mathbf{z}_i h)]/h \quad (8.3.14)$$

更好.

也易于处理带线性等式约束的最小二乘问题($f(\mathbf{x}) := \mathbf{r}^T(\mathbf{x})\mathbf{r}(\mathbf{x})$).既约坐标系内的 Jacobi 矩阵为 $\nabla_y(\mathbf{r}^{(k)T}) = \mathbf{Z}^T\mathbf{A}^{(k)}$, 其中 $\mathbf{A} = \nabla_x\mathbf{r}^T$. 式(8.3.7)利用估计 $\mathbf{Z}^T\mathbf{G}^{(k)}\mathbf{Z} = 2\mathbf{Z}^T\mathbf{A}^{(k)}\mathbf{A}^{(k)T}\mathbf{Z}$ 可以得到高斯-牛顿法等的类似算法.由此产生的性质与 5.4.2 小节中所述的相同.

历史上求解问题(8.3.1)的最早的方法是最速下降法,典型算法是 Rosen 的梯度投影法(gradient projection method)^[35].这种做法等价于在既约坐标系内选取搜索方向 $\mathbf{p}^{(k)} = -\nabla_y\psi^{(k)} = -\mathbf{Z}^T\mathbf{g}^{(k)}$ 作为最速下降向量,因而 $\mathbf{d}^{(k)} = -\mathbf{Z}\mathbf{Z}^T\mathbf{g}^{(k)}$ 是 \mathbf{x} 空间的搜索方向.当用正交分解法确定 \mathbf{Z} 时,可得到 $\mathbf{d}^{(k)} = -\mathbf{P}\mathbf{g}^{(k)}$, 其中 $\mathbf{P} = \mathbf{Z}\mathbf{Z}^T = \mathbf{I} - \mathbf{A}\mathbf{A}^+$.在这种特殊情况下 $(\mathbf{Z}\mathbf{Z}^T)^2 = \mathbf{Z}\mathbf{Z}^T$, 故 \mathbf{P} 是投影矩阵.利用隐式定义 $\mathbf{P} = \mathbf{Z}\mathbf{Z}^T$ 会更好些,而 Rosen 提出的是由 $\mathbf{P} = \mathbf{I} - \mathbf{A}\mathbf{A}^+$ 直接计算此矩阵.采用积极集法的思想也源于 Rosen^[35].

最后一个有意义的问题是,如何计算问题(8.3.1)的解处的 Lagrange 乘子 λ^* .当等式约束问题是积极集法的子问题时,就需要这些信息;而且也可以利用它进行 7.2 节中描述的灵敏度分析. λ^* 的本质定义为 $\mathbf{A}^*\lambda^* = -\mathbf{g}^*$,且通常由式(8.1.7)或者式(8.1.10)得到 λ^* .然而,由于求解问题(8.3.1)的方法的非有限终止特性,无法从计算上获得 \mathbf{x}^* 和 \mathbf{g}^* 的精确值,所以需要在 \mathbf{x}^* 的近似点 $\mathbf{x}^{(k)}$ 处,考虑如何近似计算乘子 $\lambda^{(k)}$.尤其在积极集法中,因为无法避免有关的误差,所以 $\mathbf{x}^{(k)}$ 作为 \mathbf{x}^* 的近似往往很差.根据式(8.1.7),显然可以用公式

$$\lambda^{(k)} = -\mathbf{Y}^T\mathbf{g}^{(k)} \quad (8.3.15)$$

来计算 $\lambda^{(k)}$.因为 $\mathbf{g}^{(k)} - \mathbf{g}^* = O(\|\mathbf{x}^{(k)} - \mathbf{x}^*\|)$,可将这里的 $\lambda^{(k)}$ 看作 λ^* 的一阶估计.与式(8.1.7)不同的是,因为方程 $-\mathbf{g}^{(k)} = \mathbf{A}\lambda^{(k)}$ 不再是相容的,因而所得的 $\lambda^{(k)}$ 依赖于 \mathbf{Y} 的选取.正交分解法中取 $\mathbf{Y} = \mathbf{A}^+$ 恰好可以给出这些方程的最小二乘解.

当问题的导数不可得时,若是利用式(8.3.13)或式(8.3.14)中的差商来估计既约导数,则计算 Lagrange 乘子时没有可用信息.此时,需要额外计算 m 个函数的值,即利用

$$\lambda_i^{(k)} \approx [f(\mathbf{x}^{(k)} + h\mathbf{y}_i) - f^{(k)}]/h \quad (8.3.16)$$

估计 Lagrange 乘子,其中 y_i 是矩阵 Y 的对应列.

8.4 线性不等式约束规划

大多数实际问题包含不等式约束,并可以表示成问题(8.0.2)的形式.本节主要讨论如何将8.2节的原始积极集法推广来处理这种问题.本节最后讨论一种信赖域法.在原始积极集法中,将积极集中的约束当作等式约束,其余的如同8.2节那样,暂时忽略掉.每次迭代 $x^{(k)}$ 是可行点,且每次迭代中需要求解等式约束问题(EP)

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad f(x^{(k)} + s) \\ & \text{subject to} \quad a_i^T s = 0, \quad i \in \mathcal{A} \end{aligned} \quad (8.4.1)$$

这里已经将原点平移到当前点 $x^{(k)}$.可利用2.2.5小节或者8.2节中的方法获取初始点 $x^{(0)}$.根据所利用的方法,由问题(8.4.1)的解产生搜索方向 $p^{(k)}$,且将直线 $x^{(k)} + \alpha p^{(k)}$ 上的最佳可行点(理想情况下)作为 $x^{(k+1)}$.一个重要的事实是,在二次规划中要么 $x^{(k+1)}$ 是 EP 的解,要么上一次的某一个非积极约束在线搜索中变成积极的.该事实对于更一般的问题(8.0.2)不再成立,而只能通过同一积极集 \mathcal{A} 产生的迭代序列的极限得到子问题的极小点.因此,每次迭代都需要判断 $x^{(k)}$ (即 $s=0$)是否是子问题的可接受解,如果不是,则需要针对同一积极集进行多次迭代.为此必须认真考虑可接受解的定义,8.5节将会讨论更多的细节.如果 $x^{(k)}$ 作为 EP(8.4.1)的解是可接受的,那么通过检查 Lagrange 乘子 $\lambda^{(k)}$ 来判断 $x^{(k)}$ 是否为 KKT 点.设 $\lambda_q^{(k)}$ 是最小的 Lagrange 乘子,若 $\lambda_q^{(k)} \geq 0$,则终止迭代;否则,同8.2节一样,从 \mathcal{A} 中去掉相应的不等式约束.这样,即可得到求解(8.0.2)的原始积极集法,伪码见算法8.4.1.

Algorithm 8.4.1 Active-set method for solving quadratic programming problem(8.0.2)

```

1: Given  $x^{(0)}$  and  $\mathcal{A} = \mathcal{A}(x^{(0)})$ , set  $k=0$ ;
2: while 1 do
3:   if  $s=0$  is an acceptable solution of EP(8.4.1) then
4:     let  $\lambda_q^{(k)}$  solve  $\min \{\lambda_i^{(k)} : i \in \mathcal{I} \cap \mathcal{A}\}$ ;
5:     if  $\lambda_q^{(k)} \geq 0$  then
6:       terminate with  $x^* = x^{(k)}$ ;
7:     else
8:       remove  $q$  from  $\mathcal{A}$ ;
9:     end if
10:   end if
11:   solve EP(8.4.1) for  $p^{(k)}$ ;
12:   choose  $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$  as a near best feasible point along the line  $x^{(k)} + \alpha p^{(k)}$ ;
13:   if  $\alpha_k = \bar{\alpha}_k$  then
14:     add  $j$  to  $\mathcal{A}$ ;
15:   end if
16:   set  $k=k+1$ ;
17: end while

```

算法 8.4.1 的步骤 13 中, $\bar{\alpha}_k$ 由式(8.2.2)定义. 对于一般情形, 线搜索也不能在有限步内找到最佳可行点, 因而会比式(8.2.3)更复杂. 如同 4.3 节那样, 需要选取一些条件来定义 α 值的可接受区间, 并将插值和分割结合起来保证迭代终止于一个满足条件的值. 确切地说, 除了 α 新增了由式(8.2.2)确定的上界 $\bar{\alpha}_k$ 外, 其余的完全相同. 因而, 无论是线搜索中选定的 α 的初始值, 还是由外推法确定的任何 α 值, 一旦越过这个上界, 就必须缩小至 $\bar{\alpha}_k$. 一种可能出现的情况是: $\bar{\alpha}_k$ 是区间的端点, 且不能保证这个区间包含 4.4 节所描述的可接受 α 值, 即 $[0, \bar{\alpha}_k]$ 不是定义 4.4.1 中的恰当覆盖. 此时选取 $\alpha_k = \bar{\alpha}_k$, 从而算法 8.4.1 的步骤 12 中的一维搜索需要用这种方法确定 α_k 的值.

需要注意的是, 在步骤 8 中将 q 从积极集 \mathcal{A} 中去掉后, 假设步骤 11 计算所得到的搜索方向 $p^{(k)}$ 是下降的 ($p^{(k)T} g^{(k)} < 0$), 且关于消去的约束是严格可行的 ($p^{(k)T} a_q < 0$, 大多数方法均可保证该事实是成立的). 对于步骤 11 中求解 EP(8.4.1) 的方法的选取问题, 依据有无导数信息, 一般可选用 8.3 节中叙述的任一方法. 同 8.2 节一样, 也假定不退化. 此时, 算法的收敛性依赖于步骤 3 中可接受解的选取, 细节见 8.5 节的讨论.

积极集法的重要特征是根据积极集的变化, 设计求解问题(8.4.1)的有效方法. 具体地, \mathbf{A} 会增加或者去掉一列. 此时, 不用花费 $O(n^3)$ 次运算重新计算矩阵 \mathbf{Y}, \mathbf{Z} , 而是对这些矩阵进行修正, 其仅需要 $O(n^2)$ 次运算; 也可以修正其他的矩阵. 为此, 考察当积极集 \mathcal{A} 发生变化时, 既约 Hessian 阵 $\mathbf{Z}^T \mathbf{G}^{(k)} \mathbf{Z}$ 和矩阵 \mathbf{Z} 是如何改变的. 如果利用基本牛顿法, 则通常 $\mathbf{G}^{(k)}$ 也会改变, 所以没有可用于修正的信息. 如果利用拟牛顿法, 则情况会有所不同. 首先考虑算法 8.4.1 的步骤 14, 即 $\alpha_k = \bar{\alpha}_k$, 此时将有一个约束变成积极的. 这里有可能出现曲率估计 $\mathbf{y}^T \mathbf{s} \leq 0$ 的情况(见 5.3 节), 不必更新矩阵 $\mathbf{H}^{(k)}$. 新的矩阵 \mathbf{Z} 比原来的少了一列, 通过线性变换进行列重排, 使从 \mathbf{Z} 中去掉的列是 \mathbf{z}_{n-m} . 因而, 移去(变换后的)原矩阵 $\mathbf{M}^{(k)}$ 的第 $n-m$ 行和列即可得到式(8.3.9)中的新矩阵 $\mathbf{M}^{(k)}$. 这些运算与二次规划中的相同, 细节见参考文献[33]. 在算法 8.4.1 的步骤 8 中, 当从积极集 \mathcal{A} 中移去一个约束指标时, 需要给 \mathbf{Z} 增加一列 \mathbf{z}_{n-m} . 此时自由变量的空间增加一维, 且没有可用的曲率信息来确定新方向. 因此, 可任意扩充 $\mathbf{M}^{(k)}$, 最简便的方法是令

$$\mathbf{M}^{(k)} := \begin{bmatrix} \mathbf{M}^{(k)} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (8.4.2)$$

这与选取初始的 $\mathbf{M}^{(0)} = \mathbf{I}$ 是一致的. 可类似地修正 $\mathbf{H}^{(k)}$.

最后, 给出一种牛顿法的变形, 不是使用积极集法, 而是改用信赖域法求解问题(8.0.2). 具体地, 在迭代 $\mathbf{x}^{(k)}$ 处, 求解直接从问题(8.0.2)得到的信赖域子问题

$$\underset{\mathbf{s} \in \mathbb{R}^n}{\text{minimize}} \quad q^{(k)}(\mathbf{s}) \quad (8.4.3a)$$

$$\text{subject to} \quad \mathbf{a}_i^T \mathbf{s} = b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}, \quad i \in \mathcal{E} \quad (8.4.3b)$$

$$\mathbf{a}_i^T \mathbf{s} \leq b_i - \mathbf{a}_i^T \mathbf{x}^{(k)}, \quad i \in \mathcal{I} \quad (8.4.3c)$$

$$\|\mathbf{s}\|_+ \leq \Delta_k \quad (8.4.3d)$$

式(8.4.3a)中的目标函数是原始目标函数的二阶 Taylor 级数近似, 即

$$f(\mathbf{x}^{(k)} + \mathbf{s}) \approx q^{(k)}(\mathbf{s}) = f^{(k)} + \mathbf{s}^T \mathbf{g}^{(k)} + \frac{1}{2} \mathbf{s}^T \mathbf{G}^{(k)} \mathbf{s} \quad (8.4.4)$$

此外, 为确保大范围收敛, 添加了信赖域约束(8.4.3d). 通常选取 ℓ_∞ 范数, 这样需要求解一系列带有不等式约束的二次规划子问题. 这是 9.4 节的 SQP 法的特殊形式, 通常无法用拟牛顿法的修正技巧把每步的计算量减至 $O(n^2)$, 所以每次迭代的计算费用相对较高. 但是求解

子问题(8.4.3)能迅速地确定正确的积极集, 并且避免出现锯齿现象(见8.5节的定理8.5.3). 所以当离解较远时其收敛更快. 特别是当函数值及其导数计算费用过高时, 这种方法是更可取的. 此外, 这种算法也有拟牛顿形式, 且通常与更新公式(9.4.11)一起使用.

8.5 锯齿现象

当求解有不等式约束的问题时, 有可能出现锯齿现象, 这会影响任何类型的算法的收敛速度. 尽管局部最优解处的积极集 \mathcal{A}^* 很容易确定, 然而一些方法中, 迭代点 $\mathbf{x}^{(k)}$ 处的积极集 $\mathcal{A}^{(k)}$ (如式(7.1.1)所定义)并不能固定下来(使得对所有 $k \geq K$, $\mathcal{A}^{(k)} = \mathcal{A}^*$, 其中 K 足够大), 而是在问题的不同约束子集之间振荡. 对于线性约束, 这相当于在不同的线性流形之间的锯齿现象, 其中的这些线性流形与问题(8.4.1)中不同积极集 \mathcal{A} 的可行域相对应(见图8.5.1). 如果积极集固定(对 $\forall k \geq K$, $\mathcal{A}^{(k)} = \mathcal{A}^*$), 那么其收敛速度的阶次将变得与等式约束问题的方法(见8.3节)的一样, 对大多数方法而言通常是超线性的. 若出现锯齿现象, 则收敛阶可能退化成(速率常数较大的)线性收敛, 而在一些情况下, 方法不能收敛到最优解.

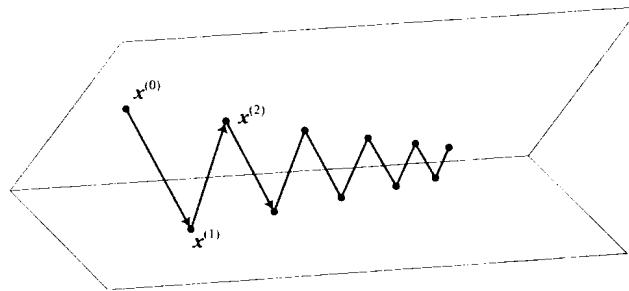


图 8.5.1 锯齿现象

对于算法8.4.1给出的积极集法, 锯齿现象出现的可能性与步骤3中可接受解的测验有关. 如果在有限步可以精确求解任何EP(8.4.1), 且在步骤4中按精确解的要求(如二次规划中那样, 即算法8.2.1的步骤5)进行检验, 则算法也会终止, 从而不可能出现锯齿现象(除了退化情形). 其原因与8.2节中描述的相同: 一旦在步骤8中从积极集 $\mathcal{A}^{(k)}$ 去掉一个约束, 根据 $\mathbf{x}^{(k)}$ 的最优性以及 $f^{(k)}$ 是单调递减的事实, 算法就不会再次返回至该积极集. 遗憾的是当 $f(\mathbf{x})$ 为一般函数时, 无法精确求解等式约束子问题(8.4.1); 且因为EP的解不一定与问题(8.0.2)相匹配, 因此找高精度解既浪费时间, 也没有必要. 另一种与之相对应的可能性是, 在每次迭代中执行步骤8. 当乘子的估计较差时, 这种策略最容易引起锯齿现象. 习题8.14是一个因锯齿现象以致无法收敛到解的例子, 那里的目标函数不是二次连续可微的, 并在算法8.4.1的步骤11中采用最速下降法. 尽管这个例子显得有些病态, 然而在实际中发现, 对拟牛顿法和光滑函数, 该策略也可能导致很慢的线性收敛. 因此, 在步骤4中需要对两种极端情况进行折衷, 从而排除锯齿现象出现的可能性, 并使积极集 $\mathcal{A}^{(k)}$ 不同于 \mathcal{A}^* 时也能够发生改变.

避免锯齿现象的方法通常基于如下估计量

$$\delta_k = \frac{1}{2} \mathbf{g}^{(k)T} \mathbf{Z} \mathbf{H}^{(k)} \mathbf{Z}^T \mathbf{g}^{(k)} \quad (8.5.1)$$

根据式(8.3.6)和式(8.3.7), 这是 $f(\mathbf{x})$ 关于同一个积极集在点 $\mathbf{x}^{(k)}$ 处下降量的二阶估计. 这

里 $H^{(k)}$ 是正定矩阵, 它是既约 Hessian 阵的逆或者既约 Hessian 阵的逆矩阵的近似(如式(8.3.11)所定义).

通常希望所用方法简单, 能证明它收敛于问题(8.0.2)的解, 且可以避免锯齿现象. 下面给出一种在很大程度上能满足这些要求的测验. 令 $l_k (< k)$ 是 k 次迭代中最后一次从积极集 \mathcal{A} 中去掉一个约束的迭代指标; 当这样的指标不存在, 就令 $l_k = 0$. 如果

$$\delta_k \leq f^{(l_k)} - f^{(k)} \quad (8.5.2)$$

则步骤 3 中的迭代点 $x^{(k)}$ 是可接受的. 此时, 函数关于当前 \mathcal{A} 的预测减少量小于最后一次从 \mathcal{A} 中移去一个约束至当前迭代所产生的总的下降量. 这种测验的动机是: 一旦出现锯齿现象, 式(8.5.2)的右端会趋于零, 由此可以保证 δ_k 趋于零, 这通常也确保与同一积极集 \mathcal{A} 对应的子序列会收敛. 如果 $x^{(k)}$ 是 EP 的可接受解, 那么根据算法 8.4.1 的步骤 4 决定去掉哪一个约束. 这里必须加上终止检验, 为了方便, 可以选为 $\delta_k \leq \epsilon$, 其中 $\epsilon > 0$ 是关于 f^* 的误差容限. 将算法 8.4.1 的步骤 3 和步骤 4 进行上述修正, 得到算法 8.5.1. 需要注意的是, 当 $x^{(k)}$ 是顶点时, $\delta_k = 0$, 式(8.5.2)为真, 从而不必特意去检测.

Algorithm 8.5.1 Modified active-set method for solving quadratic programming problem(8.0.2)

```

1: Given  $x^{(0)}$  and  $\mathcal{A} = \mathcal{A}(x^{(0)})$ , set  $k = 0$ ;
2: while 1 do
3:   if  $\delta_k \leq f^{(l_k)} - f^{(k)}$  then
4:     calculate  $\lambda^{(k)}$  from formula(8.3.15);
5:     let  $\lambda_q^{(k)}$  solve  $\min \{\lambda_i^{(k)} : i \in \mathcal{I} \cap \mathcal{A}\}$ ;
6:     if  $\lambda_q^{(k)} \geq 0$  then
7:       if  $\delta_k \leq \epsilon$  then
8:         terminate with  $x^* = x^{(k)}$ .
9:       end if
10:      else
11:        remove  $q$  from  $\mathcal{A}$ 
12:      end if
13:    end if
14:    solve EP(8.4.1) for  $p^{(k)}$ ;
15:    choose  $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$  as a near best feasible point along the line  $x^{(k)} + \alpha p^{(k)}$ ;
16:    if  $\alpha_k = \bar{\alpha}_k$  then
17:      add  $j$  to  $\mathcal{A}$ ;
18:    end if
19:    set  $k = k + 1$ ;
20: end while

```

下面的结论说明算法 8.5.1 在一些假设下是收敛的, 且不会出现锯齿现象. 该结论在某种程度上依赖于确定 EP 的解和乘子的方法的性质. 为了叙述简洁, 假定可行域 Ω 是非空的, 函数 $f(x)$ 在 Ω 上是 C^2 的, 且对任意的 $x \in \Omega$ 而言, Hessian 阵 $G(x)$ 的最小特征值 μ_1 满足 $\mu_1 \geq a > 0$, 该假设蕴含着 f 在 Ω 上是严格凸的. 采用基于线搜索的牛顿法, 可在 4.3 节中任意选择一种终止条件, 像定理 4.3.4 那样的收敛性结论在这里也成立. 也要假设对任意 $x \in \Omega$, 向量 $a_i, i \in \mathcal{A}(x)$, 是线性无关的. 此时, 可用式(8.3.15)计算乘子 $\lambda^{(k)}$. 其实这里需要的事实是: 对某固定的 \mathcal{A} , 如果 $x^{(k)} \rightarrow x^*$, 则 $\lambda^{(k)} \rightarrow \lambda^*$.

定理 8.5.1 在上述假设下, 算法 8.5.1 可从任意初始可行点 $x^{(0)}$ 收敛到问题(8.0.2)的解. 此外, 若严格互补条件在 x^* 处成立, 则不会出现锯齿现象, 且对所有充分大的 k 有 $\mathcal{A}^{(k)} = \mathcal{A}^*$.

需要指出的是, 这里利用综合的独立性假设, 使得证明避免了由退化、循环和线性相关等情况引起的困难. 也有更一般的结论, 但一般而言, 这种算法在退化情况下都会遇到困难.

最后考虑由于问题(8.4.3)确定的算法, 即信赖域型算法 6.1.1 的推广. 这种算法的优点是: 不需要假设极限点 x^∞ 处的向量 $a_i, i \in \mathcal{A}$, 是线性无关的, 即可得到大范围收敛性的结论. 由此可见, 算法不会收敛到满足 $a_i, i \in \mathcal{A}$, 线性相关的非稳定点, 从而很少失败.

定理 8.5.2 考虑基于子问题(8.4.3)的信赖域算法, 如果 $B \subset \mathbb{R}^n$ 有界, $x^{(k)} \in B, \forall k$, 且 $f \in C^2$, 那么必存在聚点 x^∞ , 其满足一阶必要条件(KKT)和弱形式的二阶必要条件

$$s^T G^\infty s \geq 0 \quad \forall s: a_i^T s = 0, \quad i \in \mathcal{A}^\infty \quad (8.5.3)$$

定理 6.1.1 下面的注记在此处也同样成立. 在点 x^∞ 处稍作额外的假设, 可证明对充分大的 k , 不会出现锯齿现象, 信赖域边界是非积极的, 且算法等价于求解既约优化问题的牛顿法.

定理 8.5.3 在定理 8.5.2 中, 令聚点 x^∞ 处的约束梯度向量 $a_i (i \in \mathcal{A}^\infty)$, 是线性无关的, 从而 Lagrange 乘子 λ^∞ 是存在且唯一的. 假设它们满足严格互补条件 $\lambda_i^\infty > 0, i \in \mathcal{I} \cap \mathcal{A}^\infty$, 并假设对所有 $p \neq 0, p^T a_i = 0 (i \in \mathcal{A}^\infty)$ 的 p 有 $p^T G^\infty p > 0$ (强形式的二阶充分条件). 则对于主序列中的 k , 有 $x^{(k)} \rightarrow x^\infty, \rho_k \rightarrow 1, \inf \Delta_k > 0$; 且当 k 充分大时, $\mathcal{A}^{(k)} = \mathcal{A}^\infty$. 此时算法等价于求解既约(消去约束 $i \in \mathcal{A}^\infty$ 后所得到的)优化问题的牛顿法.

推论 在上面的定理中, 如果 x^∞ 是一个顶点 ($|\mathcal{A}^\infty| = n$), 那么算法会有限终止.

证明 因为存在 x^∞ 的邻域, 使得 x^∞ 是唯一一个满足 $\mathcal{A}(x) = \mathcal{A}^\infty$ 的点. 所以 $\mathcal{A}^{(k)} = \mathcal{A}^\infty$ 只可能有限次. ■

总而言之, 可证明算法的表现十分令人满意, 且定理 6.1.2 后的评注在这里也是适用的. 利用矩阵 $B^{(k)}$ (比如拟牛顿技术) 来近似 Hessian 阵 $G^{(k)}$ 时, 也可得到类似的结论.

8.6 评注与参考

基于 8.2 节中求解二次规划的积极集法, 可以给出更一般的二次规划算法, 它可以求解 Hessian 阵不定的问题, 并有效求解算法中出现的序列等式约束二次规划问题. 当变量只有上下界约束, 或者是从最小二乘问题产生的二次规划问题时, 这里的算法根据问题的特殊结构可以变得更有效^[44].

在原始积极集法中, 每个迭代点是原始问题的可行点. 除此之外, 还有对偶(dual)积极集法. 该方法产生的关于 x, λ 的向量序列满足除了原始可行性外的 KKT 条件. 将 QP 转化成线性互补问题, 也可以得到一类原始-对偶(primal-dual)法, 其序列中的点既不满足原始可行性, 也不满足对偶可行性.

最后, 积极集法中, 每次迭代仅能使一个约束发生变化, 即从积极的变成非积极的, 或者反之. 而梯度投影法(gradient projection method)尝试通过快速改变积极集来加速求解过程, 并且对界约束问题(仅有关于变量的上下界的限制条件)非常有效. 内点法也是求解大规模凸二次规划的有效算法之一^[48].

习题 8

8.1 下面的问题有解吗? 给出解释.

(a) minimize $x_1 + x_2$
 subject to $x_1^2 + x_2^2 = 2, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$

(b) minimize $x_1 + x_2$
 subject to $x_1^2 + x_2^2 \leq 1, x_1 + x_2 = 3$

(c) minimize $x_1 x_2$
 subject to $x_1^2 + x_2^2 = 2$

8.2 考虑点 $\mathbf{x}^{(0)}$ 到多面集 $\{\mathbf{x} \mid \mathbf{A}^T \mathbf{x} = \mathbf{b}\}$ (其中 \mathbf{A} 是列满秩的) 的最短距离问题. 它可以表述为二次规划

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(0)})^T (\mathbf{x} - \mathbf{x}^{(0)}) \\ \text{subject to} \quad & \mathbf{A}^T \mathbf{x} = \mathbf{b} \end{aligned}$$

证明最优乘子 $\lambda^* = (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{b} - \mathbf{A}^T \mathbf{x}^{(0)})$, 最优解 $\mathbf{x}^* = \mathbf{x}^{(0)} + \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{b} - \mathbf{A}^T \mathbf{x}^{(0)})$. 说明当 $\mathbf{A} = \mathbf{a}$ 是一个列向量时, 从 $\mathbf{x}^{(0)}$ 到超平面 $\mathbf{a}^T \mathbf{x} = \mathbf{b}$ 的最短距离是 $\frac{|\mathbf{b} - \mathbf{a}^T \mathbf{x}^{(0)}|}{\|\mathbf{a}\|}$.

8.3 考虑等式约束 QP 问题(8.1.1), 并假定 \mathbf{A} 是列满秩的, 且 \mathbf{Z} 的列向量生成 \mathbf{A}^T 的零空间. 证明下面 3 个命题:

(a) QP 问题(8.1.1)有唯一解 \mathbf{x}^* 当且仅当 $\mathbf{Z}^T \mathbf{GZ}$ 是正定的.

(b) 假设方程 $\mathbf{A}^T \mathbf{x} = \mathbf{b}, \mathbf{Gx} + \mathbf{A}\lambda = -\mathbf{d}$ 有解. 说明如果既约 Hessian 阵 $\mathbf{Z}^T \mathbf{GZ}$ 是半正定的且奇异, 则问题(8.1.1)有无限多个解.

(c) 如果 $\mathbf{Z}^T \mathbf{GZ}$ 不定, 或者方程 $\mathbf{Gx}^* + \mathbf{A}\lambda^* = -\mathbf{d}$ 无解, 则问题无解.

8.4 假定 $\mathbf{A} \neq \mathbf{0}$. 说明等式约束 QP 问题(8.1.1)的 Lagrange 矩阵

$$\begin{bmatrix} \mathbf{G} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}$$

是不定的.

8.5 设 $\mathbf{G}, \mathbf{A}, \mathbf{Y}, \mathbf{Z}$ 分别为 $n \times n, n \times m, n \times m$ 和 $n \times (n-m)$ 阶矩阵 ($m \leq n$). 令 $\mathbf{Y}^T \mathbf{A} = \mathbf{I}, \mathbf{Z}^T \mathbf{A} = \mathbf{0}$, 且 $[\mathbf{Y} \ \mathbf{Z}]$ 非奇异. 证明存在唯一的 $n \times (n-m)$ 阶矩阵 \mathbf{V} 使得 $[\mathbf{A} \ \mathbf{V}]^{-1} = [\mathbf{Y} \ \mathbf{Z}]^T$ 以及 $\mathbf{Y}^T \mathbf{V} = \mathbf{0}, \mathbf{Z}^T \mathbf{V} = \mathbf{I}, \mathbf{A} \mathbf{Y}^T + \mathbf{V} \mathbf{Z}^T = \mathbf{I}$. 由此证明式(8.1.16)成立, 其中 $\mathbf{H}, \mathbf{T}, \mathbf{U}$ 由式(8.1.19)给出. 说明利用 Lagrange 乘子法如何求解问题

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \mathbf{x}^T \mathbf{Gx} + \mathbf{d}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}^T \mathbf{x} = \mathbf{b} \end{aligned}$$

以及如何利用式(8.1.16)和式(8.1.19)确定解 \mathbf{x}^* 及对应的 Lagrange 乘子向量 λ^* .

8.6 定义多面体

$$\mathbf{P} = \{\mathbf{x} : \mathbf{x} = \mathbf{A}\mathbf{y}, \mathbf{e}^T \mathbf{y} = 1, \mathbf{y} \geq \mathbf{0}\}$$

这是 $n \times m$ 阶矩阵 \mathbf{A} 的列向量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ 的凸包. 可将找多面体上距原点最近的点 \mathbf{y}^* 的问题建模为 QP

$$\begin{aligned} & \underset{\mathbf{y} \in \mathbb{R}^m}{\text{minimize}} \quad \|\mathbf{A}\mathbf{y}\|^2 \\ & \text{subject to} \quad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{e}^T \mathbf{y} = 1 \end{aligned}$$

求解问题

$$\begin{aligned} & \underset{\mathbf{y} \in \mathbb{R}^m}{\text{minimize}} \quad \|\mathbf{A}\mathbf{y}\|^2 + (\mathbf{e}^T \mathbf{y} - 1)^2 \\ & \text{subject to} \quad \mathbf{y} \geq \mathbf{0} \end{aligned}$$

得到向量 \mathbf{y}' , 规范化得到 $\mathbf{y}^* = \mathbf{y}' / \sum y'_i$. 求这个问题的对偶问题可以推导出最短距离问题.

- 8.7 利用矩阵增加或者减少一行时更新 QR 分解的方法编写算法 8.2.1 的程序, 并利用它求解

$$\begin{aligned} & \underset{\mathbf{x}_1, \mathbf{x}_2}{\text{minimize}} \quad x_1^2 + 2x_2^2 - 2x_1 - 6x_2 - 2x_1x_2 \\ & \text{subject to} \quad \frac{1}{2}x_1 + \frac{1}{2}x_2 \geq 1 \\ & \quad -x_1 + 2x_2 \leq 2 \\ & \quad x_1, x_2 \geq 0 \end{aligned}$$

选取 3 个初始点:一个在可行域的内部;一个在顶点;另一个在可行域的边界上,但不是顶点. 提示:可以利用 Matlab 中的 qr.m, qrinsert.m 和 qrdelete.m 函数.

- 8.8 考虑二次规划

$$\begin{aligned} & \underset{\mathbf{x}_1, \mathbf{x}_2}{\text{maximize}} \quad -x_1^2 - x_2^2 + 6x_1 + 4x_2 - 13 \\ & \text{subject to} \quad x_1 + x_2 \geq 3 \\ & \quad x_1, x_2 \geq 0 \end{aligned}$$

先用图解法求解, 然后利用习题 8.7 编写的积极集法的程序求解.

- 8.9 在积极集法中, 即算法 8.2.1, 假定初始积极集 \mathcal{A} 满足 $\mathbf{a}_i, i \in \mathcal{A}$ 线性无关. 当使用式 (8.2.3) 来确定要加入积极集的约束时, 证明任一选中的向量 \mathbf{a}_i 均与该集合中其余向量线性无关. 因此由归纳法可知 \mathcal{A} 中约束的梯度向量是线性无关的.
- 8.10 假设算法 8.2.1 的步骤 9 中, 从 \mathcal{A} 中删除指标 q . 证明紧接着的搜索方向 $\mathbf{p}^{(k)}$ 是下降的 ($(\mathbf{p}^{(k)T} \mathbf{g}^{(k)}) < 0$), 且是严格可行的 ($(\mathbf{p}^{(k)T} \mathbf{a}_q) < 0$).
- 8.11 设 \mathbf{W} 是 $n \times n$ 阶对称矩阵. 假设 \mathbf{Z} 是 $n \times m$ 矩阵, 使得 $\mathbf{Z}^T \mathbf{W} \mathbf{Z}$ 正定, 且 $\mathbf{Z} = [\tilde{\mathbf{Z}} \quad \mathbf{z}]$. 假设去掉 \mathbf{Z} 中的一列 \mathbf{z} , 说明 $\tilde{\mathbf{Z}}^T \mathbf{W} \tilde{\mathbf{Z}}$ 是正定的.
- 8.12 在问题(8.3.1)中, 假设 $\text{rank}(\mathbf{A}) = m$. 考虑由式(8.3.4)和式(8.3.5)得到的条件 ($\mathbf{Z}^T \mathbf{g}^* = \mathbf{0}$ 和 $\mathbf{Z}^T \mathbf{G}^* \mathbf{Z}$ 是正定的)与约束优化问题的一阶和二阶条件之间的联系, 即
- 证明 $\mathbf{Z}^T \mathbf{g}^* = \mathbf{0}$ 等价于 $\mathbf{g}^* + \mathbf{A}\lambda^* = \mathbf{0}$.
 - 说明 \mathbf{Z} 的列为线性空间 $N = \{\mathbf{p} : \mathbf{A}^T \mathbf{p} = \mathbf{0}\}$ 的一个基, 因此 \mathbf{W}^* 在集合 N 上正定等价于 $\mathbf{Z}^T \mathbf{G}^* \mathbf{Z}$ 是正定的.
- 8.13 给定 \mathbf{x}' , 考虑给问题(8.3.1)增加一个线性等式约束. 假定 \mathbf{x}' 对两个问题都是可行的, 则说明添加约束后, 既约 Hessian 阵在点 \mathbf{x}' 处的条件数不会增加. 由此人们或许会认为: 在一般情况下, 增加线性约束不会使最优化问题恶化. 但下述例子表明情况并非如此. 考虑采用指数函数 $a e^{\alpha t} + b e^{\beta t}$ 的最小二乘数据拟合. 需要选取未知参数 a, b, α, β 来最好地拟合某给定数据. 这类问题通常有适定解(尽管有些病态). 考虑加上线性约束 $\alpha = \beta$, 则参数 a 与 b 变成欠定的, 且既约 Hessian 阵在任何可行点都是奇异的. 解释这种矛盾现象.

8.14 考虑线性约束问题

$$\begin{aligned} & \text{minimize} \quad \frac{4}{3}(x_1^2 - x_1 x_2 + x_2^2)^{3/4} - x_3 \\ & \text{subject to} \quad x_3 \leq 2, \quad x \geq 0 \end{aligned}$$

说明目标函数是凸的,但非 C^2 的,且问题的解是 $x^* = (0, 0, 2)^T$. 以 $x^{(0)} = (0, a, 0)^T$ 为初始点,用算法 8.4.1 解这个问题,其中 $0 < a \leq \sqrt{2}/4$. 在步骤 4 中允许任何点为可接受解,在步骤 11 中用最速下降法. 说明 $x^{(1)} = (a, 0, \sqrt{a})^T/2$,且对 $k \geq 1$ 有

$$x^{(k)} = \begin{cases} (0, \alpha, \beta)^T, & k \text{ 是偶数} \\ (\alpha, 0, \beta)^T, & k \text{ 是奇数} \end{cases}$$

其中 $\alpha = \left(\frac{1}{2}\right)^{k-1} a, \beta = \frac{1}{2} \sum_{j=0}^{k-2} (a/2^j)^{\frac{1}{2}}$. 由此证明 $x^{(k)} \rightarrow (0, 0, (1 + \sqrt{2}/2)\sqrt{2}a)^T$,其既非最优点,也非 KKT 点.

第 9 章 约束优化: 非线性约束规划

非线性约束规划是式(7.0.1)的一般情况, 其至少有一个非线性的约束函数, 这是光滑优化问题中最困难的. 事实上, 什么样的方法最有效目前尚未达成共识, 还有许多工作有待进一步研究. 历史上最早提出的方法是基于 9.1 节和 9.2 节介绍的罚函数和障碍函数的序列极小化技术. 这些方法具有一些数值计算上的不足, 因而效率不高. 尽管如此, 特别是当问题的导数不可得且缺少可用的合适软件时, 9.2 节的方法还是值得推荐的. 鉴于 9.1 节的方法很简单 (尤其后面看到的捷径法), 通常会受到计算经验不多的使用者的青睐, 因而罚函数技术也还是有一定作用的, 并值得学习和掌握.

另一个很直观的想法是定义精确(exact)罚函数, 使得罚函数的极小点和非线性规划问题的解是一致的. 由此克服序列技术效率低的缺点. 最著名的精确罚函数是 ℓ_1 精确罚函数 (见 9.3 节), 但它是非光滑的, 不能直接应用第 4~6 章的用于光滑函数的技术来极小化它. 如何更好地利用 ℓ_1 精确罚函数有待进一步的研究. 也可以直接应用光滑精确罚函数, 但往往伴随着一些缺点.

罚函数和障碍函数均属于非线性约束规划问题的大范围方法. 也可以考虑在其解的某个邻域内表现很好的局部方法 (见 9.4 节). 应用牛顿法解由 Lagrange 乘子法 (见 7.2 节) 得到的一阶条件是非常重要的思想. 推广这种方法得到逐步二次规划法 (见 9.4 节), 此方法是局部二阶收敛的, 在非线性规划中的地位类似于无约束极小化领域的牛顿法, 是非常重要的. 在二阶导数不可得时, 人们也提出了 SQP 法的拟牛顿变形. 将 SQP 方法与精确罚函数结合起来可以提高 SQP 法的大范围收敛性质, 研究者已经提出了很多具体的结合方式. 当利用非光滑的罚函数时, 导数的不连续有可能降低收敛速度, 如 Maratos 效应. 人们已经研究出避免这些困难的方法. 一旦确定出哪一种方法是最好的之后, 无导数法 (也可能是利用有限差商来近似导数) 也会迎头赶上. Karmarkar 提出了有效的求解线性规划的多项式时间算法^[12], 鉴于该方法的效率及其实质上等价于障碍函数法, 所以人们开始重新审视罚函数法. 许多软件已经实现了基于该方法的原始-对偶内点法, 且数值表现良好. 基于此, 作为一个范例, 将在 9.5 节详细介绍线性规划的路径跟踪算法, 这种方法也可以推广到一般的非线性约束规划, 特别是凸规划问题.

可行方向法是另一个引起广泛兴趣的方法, 它的目的在于避免使用罚函数, 其包括求解线性约束规划的积极集法 (见第 8 章), 它的本质等价于变量的非线性广义消元法. 已经有相关的可用软件, 然而作为一个完全可靠的方法还是有内在困难. 还有一些其他的方法, 尽管这些方法各有特色, 但在目前尚不属于主流方法.

9.1 惩罚和障碍函数

为简化表示, 讨论下面的等式约束优化问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad c(x) = 0 \end{aligned} \quad (9.1.1)$$

其中 $c(x)$ 为 $\mathbb{R}^m \rightarrow \mathbb{R}^m$, 或者不等式约束问题

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad c(x) \leq 0 \end{aligned} \quad (9.1.2)$$

通常可以直接推广这些方法来求解混合问题(7.0.1).

9.1.1 Courant 罚函数

在求解约束不容易消去的非线性规划问题时,为了保证大范围收敛(即从任意的初始近似解都能收敛于一个局部解),必须在“减小目标值”与“保留在可行域内或接近可行域”这两个目标之间进行折衷,这不可避免地引出罚函数(penalty function)的思想. 罚函数是 f 与 c 的某种组合,它通过惩罚项控制违反约束(或者几乎违反约束)来极小化 f . 为了利用光滑无约束优化的有效技术,早期考虑的罚函数均是光滑的. 针对等式问题(9.1.1),最早的罚函数^[36](Courant 于 1943 年提出)为

$$\phi(x, \sigma) = f(x) + \frac{1}{2} \sigma \sum (c_i(x))^2 = f(x) + \frac{1}{2} \sigma c(x)^T c(x) \quad (9.1.3)$$

惩罚项由约束违反量的平方和组成,参数 σ 确定惩罚的力度. 也称 Courant 罚函数为二次罚函数,即采用约束违反量的平方和作为罚项.

例 9.1.1 (Courant 罚函数) 问题

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad x \\ & \text{subject to} \quad 1 - x = 0 \end{aligned} \quad (9.1.4)$$

的罚函数 $\phi(x, \sigma) = x + \sigma(1 - x)^2/2$ 如图 9.1.1 所示,易见 $\sigma \rightarrow \infty$ 时, $\phi(x, \sigma)$ 的极小点趋于问题的最优解 $x^* = 1$.

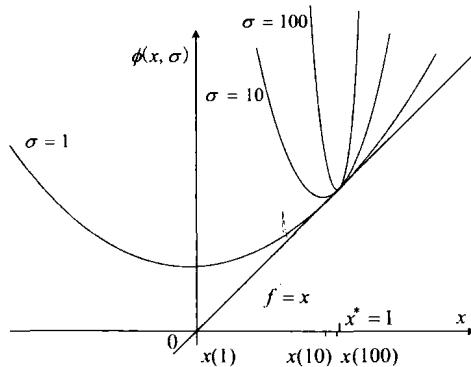


图 9.1.1 Courant 罚函数的收敛性

由此,人们提出了求解一系列极小化问题的技术,即序列极小化技术(sequential minimization technique),它的伪码描述见算法 9.1.1.

Algorithm 9.1.1 Framework for sequential minimization technique

- 1: Choose a fixed sequence $\{\sigma_k\}$ with $\sigma_k \rightarrow \infty$, typically $\{1, 10, 10^2, \dots\}$;
- 2: **repeat**
- 3: for each σ_k find a local minimizer, say $\mathbf{x}(\sigma_k)$, to $\min_{\mathbf{x}} \phi(\mathbf{x}, \sigma_k)$;
- 4: **until** $c(\mathbf{x}(\sigma_k))$ is sufficiently small.

例 9.1.2 (Courant 罚函数) 问题

$$\begin{aligned} & \text{minimize} && -x_1 - x_2 \\ & \text{subject to} && x_1^2 + x_2^2 - 1 = 0 \end{aligned} \quad (9.1.5)$$

的最优解和 Lagrange 乘子满足 $x_1^* = x_2^* = \lambda^* = 1/\sqrt{2}$, $\phi(\mathbf{x}, \sigma) = -x_1 - x_2 + \sigma(x_1^2 + x_2^2 - 1)^2/2$. 表 9.1.1 给出了方法应用于该问题的结果. 从表中可以看出: $\mathbf{x}(\sigma_k) \rightarrow \mathbf{x}^*$, 收敛速度是线性的, 且每次迭代可增加一位精确小数. 事实上, 对所有的问题均可以证明该事实, 详见定理 9.1.2 的式(9.1.10b).

表 9.1.1 应用 Courant 罚函数的结果

k	σ_k	$x_1^{(k)} = x_2^{(k)}$	$c^{(k)}$	$\lambda^{(k)}$	$\phi^{(k)}$	$\phi^{(k)} + \frac{1}{2} \lambda^{(k)T} c^{(k)}$
1	1	0.884 646 2	0.565 197 8	0.565 197 8	-1.609 568	-1.449 844
2	10	0.730 893 1	0.068 409 4	0.684 094	-1.438 387	-1.414 988
3	100	0.709 593 6	0.007 046 2	0.704 620	-1.416 705	-1.414 222
4	1 000	0.707 356 6	0.000 706 7	0.706 700	-1.414 463	-1.414 213
5	10 000	0.707 131 8	0.000 070 8	0.708 000	-1.414 239	-1.414 214
6	100 000	0.707 109 3	0.000 007 1	0.710 000	-1.414 216	-1.414 213

需要强调的是, 算法 9.1.1 的步骤 3 在实际应用中要用数值方法, 即用某一无约束最优化方法求解. 方法的选择依赖于能否求导以及问题的规模(见第 4~6 章). 通常将 $\mathbf{x}(\sigma_k)$ 作为 $\phi(\mathbf{x}, \sigma_{k+1})$ 的极小点的初始近似, 比如 Hessian 阵的逆矩阵的近似等信息也可以从前一次迭代转入下一次迭代. 事实上, 步骤 3 中不可能在有限次迭代得到精确解, 这样算法 9.1.1 仅是一个理想化的算法. 尽管适当放松精度(见式(9.1.16))也可以保证收敛, 但是仍需假定得到的 $\mathbf{x}(\sigma_k)$ 尽可能地精确. 该方法还需要假定局部极小点 $\mathbf{x}(\sigma_k)$ 存在. 然而实际应用中, 无论是非线性规划问题无界的情况, 还是存在局部解的情况, 该假定均有可能不成立. 针对后一种情况, 增大初始值 σ_0 后再重复.

可以给出与这个罚函数序列的收敛性相关的各种结论. 为此用 $\mathbf{x}^{(k)}, f^{(k)}$ 等表示由 σ_k 得到的各相关量 $\mathbf{x}(\sigma_k), f(\mathbf{x}(\sigma_k))$ 等. 第一个定理还需要假定 $f(\mathbf{x})$ 在非空可行域上有下界, 即

$$f^* = \inf_{\mathbf{x}: c(\mathbf{x}) = 0} f(\mathbf{x}) \quad (9.1.6)$$

是有限值. 如果算法 9.1.1 的步骤 3 能得到全局解, 则有下面的结论.

定理 9.1.1 若 $\sigma_k \uparrow \infty$, 则

- (i) $\{\phi(\mathbf{x}^{(k)}, \sigma_k)\}$ 单调非减,
- (ii) $\{c^{(k)T} c^{(k)}\}$ 单调非增,
- (iii) $\{f^{(k)}\}$ 单调非减,

同时 $c^{(k)} \rightarrow 0$, 且 $\{\mathbf{x}^{(k)}\}$ 的任何聚点 \mathbf{x}^* 是问题(9.1.1)的解.

证明 设 $\sigma_k < \sigma_l$, 则由 $\mathbf{x}^{(k)}$ 的定义及式(9.1.3)有

$$\phi(\mathbf{x}^{(k)}, \sigma_k) \leq \phi(\mathbf{x}^{(l)}, \sigma_k) \leq \phi(\mathbf{x}^{(l)}, \sigma_l) \leq \phi(\mathbf{x}^{(k)}, \sigma_l)$$

由前两个不等式知结论(i)成立, 且有 $\phi(\mathbf{x}^{(k)}, \sigma_l) - \phi(\mathbf{x}^{(k)}, \sigma_k) \geq \phi(\mathbf{x}^{(l)}, \sigma_l) - \phi(\mathbf{x}^{(l)}, \sigma_k)$, 即得

$$(\sigma_l - \sigma_k)(\mathbf{c}^{(k)\top} \mathbf{c}^{(k)} - \mathbf{c}^{(l)\top} \mathbf{c}^{(l)}) \geq 0$$

因此, 结论(ii)成立. 将结论(ii)代入第一个不等式即得结论(iii). 由 $\mathbf{x}^{(k)}$ 的定义和式(9.1.6)有

$$\phi(\mathbf{x}^{(k)}, \sigma_k) \leq \inf_{\mathbf{x}: \mathbf{c}(\mathbf{x})=0} \phi(\mathbf{x}, \sigma_k) = f^* \quad (9.1.7)$$

由式(9.1.3)和上述结论(iii)知, $\sigma_k \uparrow \infty$ 蕴含着 $\mathbf{c}^{(k)} \rightarrow \mathbf{0}$. 如果 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$, 有 $\mathbf{c}(\mathbf{x}^*) = \mathbf{0}$, 则根据式(9.1.6)的定义有 $f(\mathbf{x}^*) \geq f^*$. 由式(9.1.3)和式(9.1.7)又有 $f^{(k)} \leq f^*$, 因此有 $f(\mathbf{x}^*) = f^*$, 即 \mathbf{x}^* 为问题(9.1.1)的解. ■

有趣的是, 这一结果并不要求可微性或 Kuhn-Tucker 的正则性假设. 当计算得到的是局部极小点时(需对问题进行不同的假设), 也能证明类似的结论, 同时得到收敛速度的渐近估计. 为此定义向量

$$\boldsymbol{\lambda}^{(k)} = \sigma_k \mathbf{c}^{(k)} \quad (9.1.8)$$

根据下面的式(9.1.9a), 可认为这是一个 Lagrange 乘子的估计, 并可像第 7 章那样, 利用符号 $\mathbf{h}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$, $\mathbf{a}_i = \nabla c_i$ 与 $\mathbf{g} = \nabla f$ 等.

定理 9.1.2 若 $\sigma_k \rightarrow \infty$, $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$ 为任意聚点, 且 \mathbf{A}^* 的秩为 m , 则 \mathbf{x}^* 为 KKT 点, 并且有

$$\boldsymbol{\lambda}^{(k)} = \boldsymbol{\lambda}^* + o(1) \quad (9.1.9a)$$

$$\mathbf{c}^{(k)} = \boldsymbol{\lambda}^* / \sigma_k + o(1/\sigma) \quad (9.1.9b)$$

$$\sigma_k \mathbf{c}^{(k)\top} \mathbf{c}^{(k)} = \boldsymbol{\lambda}^{*\top} \boldsymbol{\lambda}^* / \sigma_k + o(1/\sigma) \quad (9.1.9c)$$

进一步, 若在 $\mathbf{x}^*, \boldsymbol{\lambda}^*$ 处二阶充分条件(7.4.4)成立, 则

$$f^* = \phi^* = \phi^{(k)} + \frac{1}{2} \sigma_k \mathbf{c}^{(k)\top} \mathbf{c}^{(k)} + o(1/\sigma) \quad (9.1.10a)$$

$$\mathbf{h}^{(k)} = \mathbf{T}^* \boldsymbol{\lambda}^* / \sigma_k + o(1/\sigma) \quad (9.1.10b)$$

其中 \mathbf{T}^* 使得

$$\begin{bmatrix} \mathbf{W}^* & \mathbf{A}^* \\ \mathbf{A}^{*\top} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{H}^* & \mathbf{T}^* \\ \mathbf{T}^{*\top} & \mathbf{U}^* \end{bmatrix} \quad (9.1.11)$$

证明 算法 9.1.1 的步骤 3 中 $\mathbf{x}^{(k)}$ 是极小点的事实蕴含着

$$\nabla \phi(\mathbf{x}^{(k)}, \sigma_k) = \mathbf{g}^{(k)} + \sigma_k \mathbf{A}^{(k)} \mathbf{c}^{(k)} = \mathbf{0} \quad (9.1.12)$$

由此从式(9.1.8)可推出

$$\mathbf{g}^{(k)} + \mathbf{A}^{(k)} \boldsymbol{\lambda}^{(k)} = \mathbf{0} \quad (9.1.13)$$

由于 \mathbf{A}^* 的秩是 m , 故对充分大的 k , $\mathbf{A}^{(k)\top}$ 存在且有界, 因此得

$$\boldsymbol{\lambda}^{(k)} = -\mathbf{A}^{(k)\top} \mathbf{g}^{(k)} = \boldsymbol{\lambda}^* + o(1) \quad (9.1.14)$$

其中 $\boldsymbol{\lambda}^*$ 定义为 $\boldsymbol{\lambda}^* = -\mathbf{A}^{*\top} \mathbf{g}^*$. 利用连续性, 从式(9.1.13)可得 $\mathbf{g}^* + \mathbf{A}^* \boldsymbol{\lambda}^* = \mathbf{0}$, 从式(9.1.14)与式(9.1.8)可得到 $\mathbf{c}^{(k)} = \boldsymbol{\lambda}^* / \sigma_k + o(1/\sigma)$. 因此, 当 $\sigma_k \rightarrow \infty$ 时, 有 $\mathbf{c}^* = \mathbf{0}$. 于是 \mathbf{x}^* 满足 KKT 条件(见 7.2 节), 并得到式(9.1.9a)与式(9.1.9b). 可直接推出方程(9.1.9c), 并证明 $\lim_{k \rightarrow \infty} \phi^{(k)} = \phi^* = f^*$. 利用 $f(\mathbf{x})$ 关于点 $\mathbf{x}^{(k)}$ 的 Taylor 展式和式(9.1.13)也可证明

$$\begin{aligned} f^* &= f^{(k)} - \mathbf{h}^{(k)\top} \mathbf{g}^{(k)} + o(\|\mathbf{h}^{(k)}\|) \\ &= f^{(k)} + \mathbf{h}^{(k)\top} \mathbf{A}^{(k)} \boldsymbol{\lambda}^{(k)} + o(\|\mathbf{h}^{(k)}\|) \\ &= f^{(k)} + \mathbf{c}^{(k)\top} \boldsymbol{\lambda}^{(k)} + o(\|\mathbf{h}^{(k)}\|) \end{aligned}$$

最后一个等式是根据 $c(x)$ 的类似的 Taylor 展式得到的. 再由算法 9.1.1 和式(9.1.8)有

$$\phi^* = \phi^{(k)} + \frac{1}{2} \sigma_k c^{(k)^\top} c^{(k)} + o(\|h^{(k)}\|) \quad (9.1.15)$$

二阶充分条件(7.4.4)和 $\text{rank}(A^*) = m$ 蕴含着 x^*, λ^* 处的 Lagrange 矩阵非奇异, 因而式(9.1.11)中的逆矩阵存在(见习题 9.9). 利用 $\nabla \mathcal{L}(x, \lambda)$ 在点 (x^*, λ^*) 的 Taylor 展式, 式(7.2.7)及式(9.1.13)得

$$\begin{bmatrix} \mathbf{0} \\ c^{(k)} \end{bmatrix} = \begin{bmatrix} W^* & A^* \\ A^{*\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} h^{(k)} \\ \lambda^{(k)} - \lambda^* \end{bmatrix} + o(\max(\|h^{(k)}\|, \|\lambda^{(k)} - \lambda^*\|))$$

于是从式(9.1.9b)以及式(9.1.11)得 $h^{(k)} = O(1/\sigma)$, 将此代入式(9.1.15)得式(9.1.10a). 用式(9.1.11)左乘上式并利用式(9.1.8)即给出式(9.1.10b). ■

从表 9.1.1 容易看出, 式(9.1.9a)与式(9.1.9b)的收敛性; 式(9.1.9c)蕴含着 $\phi^{(k)} \rightarrow f^*$. 还可以将定理 9.1.1 的其他结论用在更复杂的算法中. 式(9.1.10a)给出 f^* 的一个 $O(1/\sigma)$ 的估计, 这要优于由 $\phi^{(k)}$ 本身给出的 $O(1/\sigma)$ 的估计. 利用式(9.1.10b)中 $h^{(k)}$ 的渐近形式可以估计 x^* . 可以利用这些估计来终止罚函数的迭代, 也可以为极小化 $\phi(x, \sigma_k)$ 提供一个更好的初始近似解. 此外还应该注意, 关于 A^* 的秩的假定是必需的. 例如, 当问题(9.1.1)无可行点时, 必出现 $c^* \neq \mathbf{0}$, 因此随着 $\sigma \rightarrow \infty$, 再由式(9.1.12)必有 $A^{(k)} c^{(k)} \rightarrow \mathbf{0}$, 即 A^* 的列线性相关.

这些好的理论结果表明除了效率差些外, 序列极小化方法似乎是能让人放心使用的可靠方法. 然而事实并非完全如此, 该方法在实际使用时会产生严重的数值困难. 其原因在于随着 $\sigma_k \rightarrow \infty$, 算法 9.1.1 的步骤 3 中求解极小化问题会变得越来越困难. 用图 9.1.2 来说明这种行为, 这里针对递增的 σ 值画出了问题(9.1.5)的罚函数 $\phi(x, \sigma) = -x_1 - x_2 + \sigma(x_1^2 + x_2^2 - 1)^2/2$ 的等值线. 在 $\sigma = 100$ 时可以看出, 最优解 $x(100)$ 在径向方向是适当的, 但该方向不是很正切于约束边界, 因而从数值上很难确定 $x(100)$ 的精确位置. 基于 Hessian 阵 $\nabla^2 \phi(x^{(k)}, \sigma_k)$ 的条件数(因为 $0 < m < n$)随着 $\sigma_k \rightarrow \infty$ 逐渐恶化的事实, 也可以从数学上说明这一点.

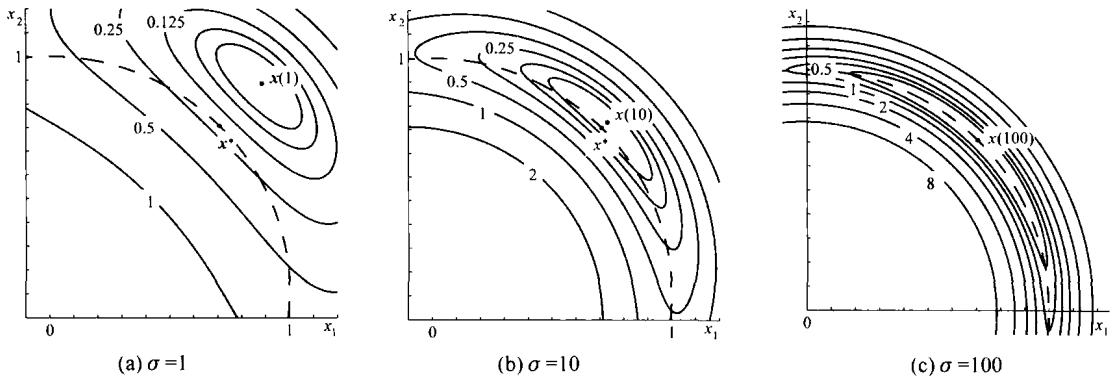


图 9.1.2 罚函数的病态性($\phi - \phi_{\min}$ 的取值为 2 的幂次方的等值线)

具体地, 因为

$$\nabla^2 \phi(x^{(k)}, \sigma_k) = W^{(k)} + \sigma_k A^{(k)} A^{(k)\top}$$

其中 $W^{(k)} = \nabla_x^2 \mathcal{L}(x^{(k)}, \lambda^{(k)})$, $\lambda^{(k)}$ 见式(9.1.8). 上式中 $\sigma_k A A^\top$ 的秩为 m , 因此随着 $\sigma_k \rightarrow \infty$, $\nabla^2 \phi$ 有 m 个特征值趋于 ∞ . 又根据 Courant-Fisher 定理, $\nabla^2 \phi$ 的其余特征值保持有界, 因此 $\nabla^2 \phi$ 的条

件数趋于 ∞ . 在实践中, 该结论表明当得到的 $\nabla^2\phi$ 的值较大时, 极小化程序已经很难再使 ϕ 降低了.

应用上述讨论来考虑如何选取序列 $\{\sigma_k\}$. 如果选取非常大的 σ_0 , 或者 σ 序列的增速过于剧烈, 则势必很难精确求解极小化问题. 相反地, 选取小的 σ_0 , 或使 σ 的递增较慢以使得 $\mathbf{x}^{(k)}$ 比较接近 $\phi(\mathbf{x}, \sigma_{k+1})$ 的极小点, 将易于得到精确解, 但这样做的效率却很低. 能在这两种情况之间取得某种折衷的典型序列见算法 9.1.1 的步骤 1. 应合理选取 σ_0 以使罚函数(9.1.3)中的 f 与 $\frac{1}{2}\mathbf{c}^T\mathbf{c}$ 之间取得某种折衷, 或极小化 $\nabla\phi$ 的幅值. 该讨论也突出了这样一个事实: 由于 $\mathbf{x}^{(0)}$ 远离 \mathbf{x}^* , 算法 9.1.1 也不可能充分利用 \mathbf{x}^* 的精确估计. 事实上许多用户并不利用序列技术, 而是采用所谓的捷径法(short cut method), 即对某个较大的 σ 值极小化罚函数(9.1.3). 如果采取这种做法, 就必须接受一阶条件内的误差. 建议这些用户注意观察约束的误差, 并利用式(9.1.10a)来估计目标函数的误差以及 KTT 条件 $\mathbf{g}^{(k)} + \mathbf{A}^{(k)}\boldsymbol{\lambda}^{(k)} = \mathbf{0}$ 的误差, 从而确定这样的误差是否是可接受的. 如果不可接受, 则可以很容易地继续利用 9.2 节的乘子罚函数法. 但如果导数及有关的软件许可, 那么从长远来看, 用 9.4 节的 Lagrange 法要有效得多.

正如前面提过的, 算法 9.1.1 与定理 9.1.1 都是理想化的, 即它们假定能得到罚函数的精确解. 事实上, 当得到近似极小点时, 也可直接给出与定理 9.1.2 中相同的结论. 为此, 在极小化 $\phi(\mathbf{x}, \sigma_k)$ 的程序中, 设以 $\mathbf{x}^{(k)}$ 作为近似极小点的终止准则为

$$\|\nabla\phi(\mathbf{x}^{(k)}, \sigma_k)\| \leq \nu \|\mathbf{c}^{(k)}\| \quad (9.1.16)$$

其中 $\nu > 0$ 为预先指定的常数, $\mathbf{c}^{(k)} = \mathbf{c}(\mathbf{x}^{(k)})$. 在与定理 9.1.2 相同的假设下, 利用存在常数 $\alpha > 0$ 使得对充分大的 k 有

$$\|\mathbf{A}^{(k)}\mathbf{c}^{(k)}\| \geq \alpha \|\mathbf{c}^{(k)}\| \quad (9.1.17)$$

成立的事实(见习题 9.2), 从式(9.1.16)与式(9.1.12)可推出

$$\nu \|\mathbf{c}^{(k)}\| \geq \|\mathbf{g}^{(k)} + \sigma_k \mathbf{A}^{(k)}\mathbf{c}^{(k)}\| \geq \sigma_k \alpha \|\mathbf{c}^{(k)}\| - \|\mathbf{g}^{(k)}\|$$

这表明 $\mathbf{c}^{(k)} \rightarrow \mathbf{0}$, 于是从式(9.1.16)有

$$-\mathbf{g}^{(k)} = \mathbf{A}^{(k)}\boldsymbol{\lambda}^{(k)} + o(1)$$

同前面一样, 可以得到定理 9.1.2 中的其他结论. 式(9.1.16)明确地限制了所能接受的 $\phi(\mathbf{x}^{(k)}, \sigma_k)$ 的近似极小点的梯度值的范围.

不等式约束问题(9.1.2)可以等价转化成等式约束问题, 这是因为

$$c_i(\mathbf{x}) \leq 0 \Leftrightarrow \max(c_i(\mathbf{x}), 0) = 0$$

从而不等式约束问题(9.1.2)的罚函数为

$$\phi(\mathbf{x}, \sigma) = f(\mathbf{x}) + \frac{1}{2}\sigma \sum_{i \in \mathcal{I}} [\max(c_i(\mathbf{x}), 0)]^2 \quad (9.1.18)$$

例 9.1.3 (Courant 罚函数) 将简单问题(9.1.4)中的等式约束变成不等式, 即

$$\begin{aligned} & \text{minimize} && \mathbf{x} \\ & \text{subject to} && 1 - \mathbf{x} \leq 0 \end{aligned} \quad (9.1.19)$$

得罚函数 $\phi(\mathbf{x}, \sigma) = \mathbf{x} + \sigma(\max(1 - \mathbf{x}, 0))^2 / 2$, 见图 9.1.1. 当 $\mathbf{x} \geq 1$ 时 ϕ 与 f 相等, 当 $\mathbf{x} < 1$ 时 ϕ 的图像与图 9.1.1 中的完全相同.

例 9.1.4 (Courant 罚函数) 将简单问题(9.1.5)中的约束换成不等式, 即

$$\begin{aligned} & \text{minimize} && -\mathbf{x}_1 - \mathbf{x}_2 \\ & \text{subject to} && \mathbf{x}_1^2 + \mathbf{x}_2^2 - 1 \leq 0 \end{aligned} \quad (9.1.20)$$

罚函数 $\phi = -\mathbf{x}_1 - \mathbf{x}_2 + \sigma(\max(\mathbf{x}_1^2 + \mathbf{x}_2^2 - 1, 0))^2 / 2$ 的等值线在单位圆外的如图 9.1.2 所示, 在

单位圆内的是 $-x_1 - x_2$ 的等值线.

由上述例子的图形可以看出函数(9.1.18)的二阶导数在 $c=0$ 有跳跃, 是不连续的(例如在 x^*). 此外, $x^{(k)}$ 是从不等式约束不可行的一侧接近 x^* , 因此也把(9.1.18)称为外罚函数(exterior penalty function). 只要用 $\max(c_i^{(k)}, 0)$ 代替 $c_i^{(k)}$, 就可得出与定理9.1.1和9.1.2完全相同的结论.

9.1.2 障碍函数

障碍函数法(barrier function methods)是另一类可用于求解不等式约束问题(9.1.2)的序列极小化方法. 这一类方法的特点是障碍项在约束边界上取值为无穷大, 从而在任何时候都能严格满足约束. 该方法非常适合于可行域外目标函数无定义的问题, 因为得到的极小点序列也是可行的, 有时也称相应的方法为内点法(interior point methods). 两个最重要的障碍函数分别是Frisch提出的对数障碍函数^[37]

$$\phi(x, \mu) = f(x) - \mu \sum_i \log(-c_i(x)) \quad (9.1.21)$$

和Carroll提出的倒数障碍函数^[38]

$$\phi(x, \mu) = f(x) - \mu \sum_i [c_i(x)]^{-1} \quad (9.1.22)$$

如同式(9.1.3)中的 σ , 参数 μ 用于控制障碍函数的迭代. 所不同的是这里选取的序列 $\{\mu_k\}$ 满足 $\mu_k \rightarrow 0$ 以确保障碍项越来越小, 以至于可忽略不计(接近边界除外). 同样, 定义 $x(\mu_k)$ 为 $\phi(x, \mu_k)$ 的极小点, 其他的过程与算法9.1.1完全一样.

例9.1.5(对数障碍函数) 问题(9.1.19)的对数障碍函数 $\phi(x, \mu) = x - \mu \log(x-1)$, 图9.1.3对一组 μ_k 的值给出了函数的图像, 由此可以看到随着 $\mu_k \rightarrow 0$, $x(\mu_k) \rightarrow x^*$.

用完全类似于定理9.1.1的方法可以严密地证明该结论. 还可以给出其他特征, 如Lagrange乘子的估计、 $h^{(k)}$ 的渐近特征等, 可以用后者来确定适当的序列 $\{\mu_k\}$ (见习题9.3). 一般来说, 倒数罚函数比二次罚函数和对数障碍函数要差.

遗憾的是, 障碍函数法在极限情形也会产生和罚函数法相同的数值困难, 尤其是 $x(\mu_k)$ 正切于约束曲面这一十分坏的特性, 以及由于条件数恶化和大梯度以致很难找到极小点. 除此之外, 还伴随着其他的困难. 首先对于不可行点而言, 障碍函数无定义, 而简单地把不可行点处的罚函数置为无穷大的做法将会降低一维搜索的效率. 一维

搜索过程中常规的二次或三次插值也因为奇异性而不像往常那样有效. 为此需要采用特殊的一维搜索技巧, 因而失去了一维搜索简单的特点. 其次, 该方法需要有一个初始的严格可行内点, 这需要求解一个严格不等式组, 其本身就不是一个简单的问题. 基于这些困难以及序列技术效率通常较低的特点, 很长一段时间人们极少采用障碍函数. 但Karmarkar提出的能有效求解线性规划的多项式时间算法^[12], 本质上等价于一个对数障碍函数法, 所以人们开始重新审视对数障碍函数法, 并提出了实用的求解各种凸优化问题的内点法.

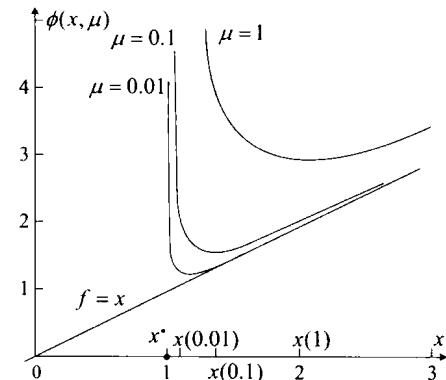


图9.1.3 障碍函数中递增的病态性

具体而言,从理论的角度出发,路径(path) $\{x(\mu) : \mu \in (0, \mu_0)\}$ 的性质对算法的有效性影响很大,尤其想知道序列 $\{x(\mu_k)\}$ 是否收敛于约束问题(9.1.2)的局部解.从实用的角度出发,为了实现障碍函数法,首先,需要找到一个严格可行解 $x^{(0)}$ 来开始迭代;其次,需要一个子程序解决 $\phi(x, \mu_k)$ 的最小化问题;最后,需要给出参数 μ_k 的更新方法.9.5节针对线性规划问题讨论了这些问题.作为拓展,推荐读者进一步阅读参考文献[47]、[48]、[50]、[51].

9.2 乘子罚函数

求解问题(9.1.1)的罚函数(9.1.3)可以设想为:在 $\sigma_k \rightarrow \infty$ 的极限情况下得到一个局部极小点 x^* (见图 9.1.1 与图 9.1.2).更确切地说,由式(9.1.9b),罚函数(9.1.3)的极小点 $x^{(k)}$ 不再精确地满足条件 $c_i(x) = 0, i \in \mathcal{E}$,而是被扰动成

$$c_i^{(k)} \approx \lambda_i^* / \sigma_k, \quad i \in \mathcal{E} \quad (9.2.1)$$

由于要求 $c_i^{(k)} \rightarrow 0$,且一般情形下至少有一个乘子 λ_i^* 非零,故必然导致 $\sigma_k \rightarrow \infty$.

一种想法是对约束 $c_i(x)$ 进行平移,即在罚函数(9.1.3)中用 $c_i(x) - \theta_i$ 代替 $c_i(x)$,使得对有限的 σ, ϕ 可以在 x^* 取到极小值,其中参数 θ_i 是相对于原点的偏移.另外,这种做法还可以保持 $\nabla c_i(x)$ 的方向.基于这种思想得到函数^[40]

$$\phi(x, \theta, \sigma) = f(x) + \frac{1}{2} \sigma \sum_{i \in \mathcal{E}} [c_i(x) - \theta_i]^2 \quad (9.2.2)$$

例 9.2.1 以简单问题(9.1.4)为例,此时

$$\phi(x, \theta, \sigma) = x + \frac{1}{2} \sigma (1 - x - \theta)^2$$

其几何直观如图 9.2.1 所示.从图中可以看到,如果选取合适的偏移 θ (与 σ 有关),则 x^* 就是 $\phi(x, \theta, \sigma)$ 的极小点.如此所得算法,在保持 σ 有限以避免极限 $\sigma \rightarrow \infty$ 的病态问题的同时,尝试找出最优的偏移向量 θ .

例 9.2.2(乘子罚函数) 以问题(9.1.5)为例,若取 $\sigma = 1$,则最优偏移 $\theta = -1/\sqrt{2}$,得

$$\phi(x, -1/\sqrt{2}, 1) = -x_1 - x_2 + \frac{1}{2} (x_1^2 + x_2^2 - 1 + 1/\sqrt{2})^2$$

其等值线如图 9.2.2 所示,对比图 9.1.2 中 $\sigma_k \rightarrow \infty$ 的情形,可以看出二者的差异.

另一种做法是想办法改变函数(9.1.3)以避免这种系统的扰动,即对适中的 σ_k 值,近似极小点更好地满足等式约束 $c_i(x) = 0$.在罚函数(9.1.3)中引入 Lagrange 乘子 λ 的显式估计可以达到这种目标,即考虑函数^[41]

$$\phi(x, \lambda, \sigma) = f(x) + \lambda^T c(x) + \frac{1}{2} \sigma c(x)^T c(x) \quad (9.2.3)$$

与式(9.1.3)相比,这里添加了乘子项 $\lambda^T c$,因而称式(9.2.3)为乘子罚函数(multiplier penalty function).此外,为以 f 为目标的 Lagrange 函数扩充罚项 $\sigma c^T c / 2$ 也可得到函数(9.2.3),因而也称为增广 Lagrange 函数(augmented Lagrangian function).

如果令

$$\lambda_i = -\theta_i \sigma, \quad i = 1, 2, \dots, m \quad (9.2.4)$$

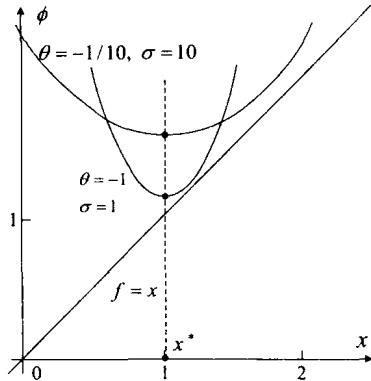
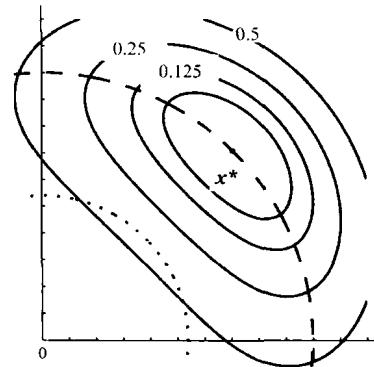


图 9.2.1 乘子罚函数

图 9.2.2 乘子罚函数 ($\sigma = 1, \theta = -1/\sqrt{2}$, $\phi - \phi_{\min}$ 的取值为 2 的幂次) 的等值线

则知式(9.2.2)和式(9.2.3)只相差与 x 无关的项 $\sigma \sum \theta_i^2 / 2$. 所以有时也称函数(9.2.3)为 **Powell-Hestenes 函数**. 下列的结论表明(9.2.3)中最优的控制参数向量是问题(9.1.1)的解 x^* 处的 Lagrange 乘子 λ^* .

定理 9.2.1 (精确性) 如果点 x^*, λ^* 处二阶充分条件成立, 则存在 $\sigma' \geq 0$ 使得对任意 $\sigma > \sigma'$, x^* 为 $\phi(x, \lambda^*, \sigma)$ 的严格局部极小点, 即 $x^* = x(\lambda^*)$.

证明 对函数(9.2.3)关于 x 求导, 有

$$\nabla \phi(x, \lambda^*, \sigma) = g + A\lambda^* + \sigma A c \quad (9.2.5)$$

二阶条件要求 x^*, λ^* 为 KKT 点, 即 $g^* + A^* \lambda^* = 0, c^* = 0$, 由此得 $\nabla \phi(x^*, \lambda^*, \sigma) = 0$. 对式(9.2.5)求导得

$$\nabla^2 \phi(x, \lambda^*, \sigma) = W + \sigma A A^T := W_\sigma \quad (9.2.6)$$

其中 $W = \nabla^2 f + \sum_i (\lambda_i^* + \sigma c_i) \nabla^2 c_i$. 令

$$W_\sigma^* := \nabla^2 \phi(x^*, \lambda^*, \sigma) = W^* + \sigma A^* A^{*T} \quad (9.2.7)$$

设 $\text{rank}(A^*) = r \leq m$, $B \in \mathbb{R}^{n \times r}$ 是 A^* 的标准正交基矩阵, 即 $B^T B = I$ 且 $A^* = BC$, 其中 $C = B^T A^*$ 的秩为 r . 考虑任意的非零向量 u , 并设 $u = v + Bw$, 其中 $B^T v = 0 = A^{*T} v$, 则

$$u^T W_\sigma^* u = v^T W^* v + 2 v^T W^* B w + w^T B^T W^* B w + \sigma w^T C C^T w$$

由式(7.4.3)和式(7.4.4), 存在常数 $a > 0$ 使得 $v^T W^* v \geq a \|v\|_2^2$. 令 b 为矩阵 $W^* B$ 的最大奇异值, $d = \|B^T W^* B\|_2$, 又令 $\mu > 0$ 为 CC^T 的最小特征值, 则

$$u^T W_\sigma^* u \geq a \|v\|_2^2 - 2b \|v\|_2 \|w\|_2 + (\sigma \mu - d) \|w\|_2^2$$

令 $\sigma' = (d + b^2/a)/\mu$, 由于 $\|v\| = \|w\| = 0$ 不可能成立, 因而只要 $\sigma > \sigma'$ 就有 $u^T W_\sigma^* u > 0$. 于是 $\nabla \phi(x^*, \lambda^*, \sigma) = 0$ 且 $\nabla^2 \phi(x^*, \lambda^*, \sigma)$ 是正定的. 因此对充分大的 σ , x^* 为 $\phi(x, \lambda^*, \sigma)$ 的严格局部极小点. ■

在实践中, 并不知道 λ^* 的精确值, 然而该结论和它的证明过程表明: 倘若 λ 是 λ^* 的一个足够好的估计, 即使 σ 不是特别大, 通过极小化 $\phi(x, \lambda, \sigma)$ 也可以得到 x^* 的一个很好的估计. 下面考察关于二阶条件的假定.

例 9.2.3 问题

$$\begin{aligned} & \text{minimize} && x_1^2 + x_1 x_2 \\ & \text{subject to} && x_2 = 0 \end{aligned}$$

的解 $x^* = \mathbf{0}$, 相应的 Lagrange 乘子是唯一的, 且 $\lambda^* = 0$, 可是上述二阶条件并不满足. 事实上, 对任何 σ , $x^* = \mathbf{0}$ 都不是 $\phi(x, 0, \sigma) = x_1^2 + x_1 x_2 + \sigma x_2^2 / 2$ 的极小点. 该例说明这里关于二阶条件的假定起重要的作用, 且不能再放宽. 为此, 今后都假定二阶充分条件成立, 且 σ 充分大.

基于以上的讨论, 选取充分大的控制参数 σ 后, 用 λ 作为序列极小化算法中的控制参数, 即得算法 9.2.1.

Algorithm 9.2.1 Framework for multiplier penalty function

- 1: Determine a sequence $\{\lambda^{(k)}\}$ with $\lambda^{(k)} \rightarrow \lambda^*$;
 - 2: **repeat**
 - 3: for each $\lambda^{(k)}$ find a local minimizer, say $x(\lambda^{(k)})$, to $\min \phi(x, \lambda^{(k)}, \sigma)$;
 - 4: **until** $c(x(\lambda^{(k)}))$ is sufficiently small.
-

算法 9.2.1 与算法 9.1.1 的主要区别在于这里不可能预先知道 λ^* , 因而在步骤 1 不可能预先确定一个序列. 下面将会说明如何用构造的方式产生这样的序列.

可以把 $\phi(x, \lambda, \sigma)$ 的极小点 $x(\lambda)$ 看作非线性方程

$$\nabla \phi(x, \lambda, \sigma) = \mathbf{0} \quad (9.2.8)$$

的解. 这个方程的 Jacobi 矩阵 $\nabla^2 \phi(x^*, \lambda^*, \sigma)$ 正定, 因此由隐函数定理知: 存在 λ^* 的开邻域 $N_{\lambda^*} \subset \mathbb{R}^m$, x^* 的开邻域 $N_{x^*} \subset \mathbb{R}^n$, 及 C^1 函数 $x(\lambda)$ ($N_{\lambda^*} \rightarrow N_{x^*}$), 使得 $\nabla^2 \phi(x(\lambda), \lambda, \sigma) = \mathbf{0}$. 此外对所有 $x \in N_{x^*}$ 与 $\lambda \in N_{\lambda^*}$, $\nabla^2 \phi(x, \lambda, \sigma)$ 正定, 因而 $x(\lambda)$ 为 $\phi(x, \lambda, \sigma)$ 的极小点. 这里假定极小化程序得到的 $x(\lambda)$ 总属于邻域 N_{x^*} .

这样, 在确定 $x(\lambda)$ 之后, 得到关于 λ 的函数

$$\psi(\lambda) := \phi(x(\lambda), \lambda, \sigma) \quad (9.2.9)$$

利用 $x(\lambda)$ 在 N_{x^*} 上的最优性和 $c^* = \mathbf{0}$, 对 N_{λ^*} 中的任意 λ 有

$$\psi(\lambda) = \phi(x(\lambda), \lambda, \sigma) \leq \phi(x^*, \lambda, \sigma) = \phi(x^*, \lambda^*, \sigma) = \psi(\lambda^*) \quad (9.2.10)$$

因而 λ^* 是 $\psi(\lambda)$ 的一个局部无约束极大点. 事实上, 如果 $x(\lambda)$ 还是 $\phi(x, \lambda)$ 的全局极小点, 那么 λ^* 也是 $\psi(\lambda)$ 的全局极大点. 因此, 应用无约束极小化方法来极小化函数 $\psi(\lambda)$, 即可得到使序列 $\lambda^{(k)} \rightarrow \lambda^*$ 的方法. 为此需要 $\psi(\lambda)$ 关于 λ 的一阶与二阶导数 $\nabla \psi$ 与 $\nabla^2 \psi$ 的表达式. 用矩阵记号 $[\partial x / \partial \lambda]_{ij}$ 表示 $\partial x_i / \partial \lambda_j$, 由链式法则有

$$[\partial \psi / \partial \lambda] = [\partial \phi / \partial x] [\partial x / \partial \lambda] + [\partial \phi / \partial \lambda]$$

这里用全导数表明是 ψ 关于 $x(\lambda)$ 和 λ 的变化率. 由式(9.2.8)有 $[\partial \phi / \partial x] = \mathbf{0}$, 而由式(9.2.3)有 $\partial \phi / \partial \lambda_i = c_i$, 因此得

$$\nabla \psi(\lambda) = c(x(\lambda)) \quad (9.2.11)$$

再利用链式法则有

$$[\partial c / \partial \lambda] = [\partial c / \partial x] [\partial x / \partial \lambda] = \mathbf{A}^T [\partial x / \partial \lambda]$$

对式(9.2.8)作用 $[\partial / \partial \lambda]$, 得

$$[\partial \nabla \phi(x(\lambda), \lambda) / \partial \lambda] = [\partial \nabla \phi / \partial x] [\partial x / \partial \lambda] + [\partial \nabla \phi / \partial \lambda] = \mathbf{0}$$

但因 $[\partial \nabla \phi / \partial x] = \nabla^2 \phi(x(\lambda), \lambda, \sigma) = \mathbf{W}_\sigma$ 与 $[\partial \nabla \phi / \partial \lambda] = \mathbf{A}$, 因此有

$$\nabla^2 \psi(\lambda) = [\mathbf{d}c/d\lambda] = -\mathbf{A}^T \mathbf{W}_\sigma^{-1} \mathbf{A} \Big|_{x(\lambda)} \quad (9.2.12)$$

由于 $c(x(\lambda^*)) = c(x^*) = \mathbf{0}$, 且 \mathbf{W}_σ 正定, 因而有 $\nabla \psi(\lambda^*) = \mathbf{0}$ 且(当 \mathbf{A}^* 的秩为 m 时) $\nabla^2 \psi(\lambda^*)$ 是负定的, 这再一次证实了式(9.2.10)中有关极大化的结论.

关于算法 9.2.1 的步骤 1, 一种很直观的做法是按牛顿迭代选取序列 $\{\lambda^{(k)}\}$, 即确定初始估计 $\lambda^{(0)}$, 再利用

$$\lambda^{(k+1)} = \lambda^{(k)} + (\mathbf{A}^T \mathbf{W}_\sigma^{-1} \mathbf{A})^{-1} \mathbf{c} \Big|_{x(\lambda^{(k)})}, \quad (9.2.13)$$

这种方式要用到 \mathbf{W}_σ , 缺陷是需要二阶导数的显式表达式. 当只有一阶导数可用, 且利用拟牛顿法求 $x(\lambda^{(k)})$ 时, 由 5.3 节产生的矩阵 \mathbf{H} 可以很好地近似 \mathbf{W}_σ^{-1} . 在仅用一阶导数的情况下, 在式(9.2.13)中使用这样的近似矩阵可以保留 Newton 法的优点.

对于充分大的 σ 有(见习题 9.5)

$$(\mathbf{A}^T \mathbf{W}_\sigma^{-1} \mathbf{A})^{-1} \approx \sigma \mathbf{I} \quad (9.2.14)$$

Powell 和 Hestenes 于 1969 年基于该事实, 在式(9.2.13)中利用该近似, 得到迭代

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} + \sigma c_i^{(k)}, \quad i \in \mathcal{E} \quad (9.2.15)$$

该式不需要任何导数, 因此它特别适合不需要计算或估计导数的极小化 $\phi(x, \lambda, \sigma)$ 的程序. 而且, 通过取充分大的 σ , 可以让 $\lambda^{(k)}$ 以任意快的速度线性收敛到 λ^* (见习题 9.6). 通常的乘子法指按式(9.2.15)产生乘子序列的方法, 也称为增广 Lagrange 函数法或 Powell-Hestenes 法. 算法 9.2.2 是一个具体的乘子法的框架, 其中根据需要来修正罚参数的值.

例 9.2.4 (乘子的更新) 应用基于式(9.2.13)与式(9.2.15)的方法求解问题(9.1.5), 结果见表 9.2.1, 其中初始的 $\lambda^{(0)} = 0$. 从表中可以看出, 对各种情况都有 $\lambda^{(k)} \rightarrow \lambda^*$ 与 $c^{(k)} \rightarrow 0$. 乘子法显示了线性收敛的性质, 其速率常数在 $\sigma = 1$ 时为 0.26, 在 $\sigma = 10$ 时为 0.034. 这揭示了把 σ 扩大 10 倍可使 c_i 的速率常数渐近地缩小为 1/10 的事实. 牛顿法的收敛速度看起来接近于二次, 稍微优于乘子法的 $\sigma = 10$ 的结果.

表 9.2.1 不同的 $\lambda^{(k)}$ 的修正公式

k	Newton 法 (式(9.2.13))		Powell-Hestenes 法 (式(9.2.15))			
	$\sigma = 1$		$\sigma = 1$		$\sigma = 10$	
	$\lambda^{(k)}$	$c^{(k)}$	$\lambda^{(k)}$	$c^{(k)}$	$\lambda^{(k)}$	$c^{(k)}$
1	0	0.565 197 7	0	0.565 197 7	0	0.068 409 5
2	0.667 245 0	0.029 617 4	0.565 197 7	0.106 898 1	0.684 094 6	0.002 222 8
3	0.706 885 3	0.000 163 7	0.672 095 8	0.025 995 6	0.706 322 2	0.000 075 8
4	0.707 106 8	0.149×10^{-7}	0.698 091 4	0.006 669 2	0.707 080 1	0.000 002 6
5			0.704 760 6	0.001 733 9	0.707 105 8	0.894×10^{-7}
6			0.706 494 5	0.000 452 4		
7			0.706 946 9	0.000 118 1		
8			0.707 065 0	0.000 030 8		

虽然上述提及的算法具有较好的局部收敛性质, 但它们必须要嵌在一个更一般的算法中执行以确保大范围收敛性, 这可按 Powell 于 1969 年提出的算法 9.2.2 执行, 其中根据情况, 需要时才增加罚参数.

算法 9.2.2 能获取速率常数至少是 0.25 的线性收敛. 如果 $\phi(x, \lambda^{(k)}, \sigma)$ 的极小点 $x(\lambda, \sigma)$ 所得到的 c 不能获取这个速率, 则将罚参数 σ 增大 10 倍以期更快的收敛速度(见习题 9.5). 可按如下思路得到 $c^{(k)} \rightarrow \mathbf{0}$ 的结论.

Algorithm 9.2.2 A global algorithm for multiplier penalty function method

```

1: Initially set  $\lambda = \lambda^{(1)}$ ,  $\sigma = \sigma^{(1)}$ ,  $k = 1$ ; choose  $\mathbf{x}^{(0)}$  and compute  $\| \mathbf{c}^{(0)} \|_\infty$ ;
2: repeat
3:   find the minimizer  $\mathbf{x}'$  of  $\phi(\mathbf{x}, \lambda^{(k)}, \sigma)$  with initial point  $\mathbf{x}^{(k-1)}$ , and denote  $\mathbf{c}' = \mathbf{c}(\mathbf{x}')$ ;
4:   if  $\| \mathbf{c}' \|_\infty > \frac{1}{4} \| \mathbf{c}^{(k-1)} \|_\infty$  then
5:     set  $\sigma = 10\sigma$ ;
6:     set  $\mathbf{x}^{(k-1)} = \mathbf{x}'$ ;
7:   else
8:     set  $\mathbf{x}^{(k)} = \mathbf{x}'$ ,  $\mathbf{c}^{(k)} = \mathbf{c}'$ ;
9:   set  $\lambda^{k+1} = \lambda^{(k)} + \sigma \mathbf{c}^{(k)}$ ;
10:  set  $k = k + 1$ ;
11: end if
12: until  $\mathbf{c}^{(k)}$  is sufficiently small.

```

首先内层循环,即步骤 6→步骤 3,肯定会在有限次迭代后终止.在这个内层迭代中, λ 的取值是固定的,若对某个 i , $|c_i| > \frac{1}{4} \| \mathbf{c}^{(k)} \|_\infty$ 出现无限次,则有 $\sigma \rightarrow \infty$. 利用定理 9.1.1, 又可得到 $c_i \rightarrow 0$. 这与上述不等式出现无限次相矛盾. 由此知内层迭代必有限终止. 算法 9.2.2 的步骤 9 不管选用什么样的公式,该结论都是成立的. 如同定理 9.1.2, 可得出任意极限点为问题(9.1.1)的 KKT 点,且 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$, $\lambda^{(k)} + \sigma_k \mathbf{c}^{(k)} \rightarrow \lambda^*$. 对于式(9.2.13)和(9.2.15),当 σ 充分大时,算法将转入基本迭代,即 σ 保持不变而只有参数 λ 发生变化,由此即可得到所需要的收敛速度.

实际上这一结果并不如想像得那样好. 如同 9.1.1 小节指出的那样,增加 σ 可能会导致由于病态问题所引起的困难,这时会使解的精度受损. 另一个令人不满意的情况是:当没有可行解时,算法并不能检查出这种情况,导致无限制地增大 σ .

对于不等式约束问题(9.1.2),可将函数(9.2.2)修正为

$$\phi(\mathbf{x}, \theta, \sigma) = f(\mathbf{x}) + \frac{1}{2} \sigma \sum_{i \in I} [\max(c_i(\mathbf{x}) - \theta_i, 0)]^2 \quad (9.2.16)$$

例 9.2.5 (乘子罚函数) 考虑问题(9.1.19),得到的乘子罚函数

$$\phi(x, \theta, \sigma) = x + \frac{1}{2} \sigma [\max(1 - x - \theta, 0)]^2$$

见图 9.2.1. 除 $c(x) > \theta$ (即 $x \geq 1 - \theta$ 时 ϕ 的图形与 f 的图形重合) 外,其余的同图 9.2.1 中对应的曲线.

尽管 $\phi(x, \theta, \sigma)$ 在使得 $c_i(x) = \theta_i$ 的点处二阶导数有跳跃,是间断的,但这些点通常离最优解较远,因此在实际计算时不会对无约束极小化程序产生不良影响. 另一个例子是问题(9.1.20),此时乘子罚函数

$$\phi(x, \theta, \sigma) = -x_1 - x_2 + \frac{1}{2} \sigma [\max(x_1^2 + x_2^2 - 1 - \theta, 0)]^2$$

参照图 9.2.2 可得到 $\theta = -1/\sqrt{2}$ 时该函数的等值线. 图 9.2.2 中的点线表示圆 $c(\mathbf{x}) = \theta = -1/\sqrt{2}$, 即 $x_1^2 + x_2^2 = 1 - 1/\sqrt{2}$. 仅在圆内,该函数的等值线与图 9.2.2 中的不同,是未受惩罚的 $f(\mathbf{x})$ 的等值线,是线性的. 由此可见曲面在圆周上出现跳跃,是不连续的,但此时曲面远离 \mathbf{x}^* .

如同式(9.2.2)利用变换(9.2.4)并略去与 x 无关的项那样, 可把函数(9.2.16)重新表述为下列的乘子罚函数

$$\phi(x, \lambda, \sigma) = f(x) + \sum_i \begin{cases} \lambda_i c_i + \frac{1}{2} \sigma c_i^2, & c_i \geq -\lambda_i / \sigma \\ -\frac{1}{2} \lambda_i^2 / \sigma, & c_i \leq -\lambda_i / \sigma \end{cases} \quad (9.2.17)$$

其中 $c_i = c_i(x)$. 如同式(9.2.3)一样, 函数(9.2.17)便于理论推导.

关于等式约束的大多数理论结果可直接推广至不等式约束的情况. 例如, 如果严格互补条件成立, 则立即可以得到定理 9.2.1 的推广. 对偶函数 $\psi(\lambda)$ 的定义仍是式(9.2.9). 类似于式(9.2.10)的全局性结论是

$$\begin{aligned} \psi(\lambda) &= \phi(x(\lambda), \lambda, \sigma) \leq \phi(x^*, \lambda, \sigma) \\ &= f^* + \sum_i \begin{cases} \lambda_i c_i^* + \frac{1}{2} \sigma c_i^{*2}, & c_i^* \geq -\lambda_i / \sigma \\ -\frac{1}{2} \lambda_i^2 / \sigma, & c_i^* \leq -\lambda_i / \sigma \end{cases} \\ &\leq f^* + \sum_i \begin{cases} -\frac{1}{2} \sigma c_i^{*2} \\ -\frac{1}{2} \lambda_i^2 / \sigma \end{cases} \leq f^* = \phi(x^*, \lambda^*, \sigma) = \psi(\lambda^*) \end{aligned} \quad (9.2.18)$$

当严格互补条件成立时, 这一结论也是局部成立的, 也可以推广至严格互补条件不成立的情况^[20].

也可以利用结论(9.2.14)推广式(9.2.15), 得

$$\lambda_i^{(k+1)} = \max(\lambda_i^{(k)} + \sigma c_i^{(k)}, 0), \quad i = 1, 2, \dots, m \quad (9.2.19)$$

可以将这些公式与大范围收敛的算法 9.2.2 相结合, 但必须用 $\|\nabla \psi\|_\infty$ 代替 $\|c\|_\infty$ 来控制收敛速度.

数值试验结果表明: 虽然 $\lambda^{(k)}$ 的牛顿型更新公式与 Powell-Hestenes 更新公式都很有效, 但牛顿型公式要更有效一些. 其局部收敛较快, 一般经 4 到 6 次极小化即可获得较高精度的解. 此时, 若 σ 的取值也大小适中, 则不会出现由病态问题所引起的困难以及精度的损失. 此外, 也可以把 Hessian 阵传入下次迭代, 且在增加 σ 时对其进行修正, 因而序列极小化过程中所需要计算的开销随迭代进行而迅速减少. $\phi(x, \lambda, \sigma)$ 总是适定的, 因而也很容易处理不可行点, 而且很容易利用现有的拟牛顿法子程序来编写方法的程序. 该方法的主要缺点是它的序列性质. 与 9.4 节的直接法相比, 这种方法的效率较低, 而且方法的大范围收敛性依赖于逐渐增大 σ 的事实. 虽然方法的理论结果较强, 但在实际应用中已出现过病态问题并损失精度的情况.

最后, 再指出两点. 第一, 如果问题的约束既有线性的, 也有非线性的, 建议保留线性约束, 对非线性约束构造相应的乘子罚函数. 这样, 在算法 9.1.1 的步骤 3 中, 需要在线性约束条件下极小化 $\phi(x, \lambda, \sigma)$. 形如 $l_i \leq x_i \leq u_i$ 的界约束对无约束极小化程序所引起的变动很小, 所以这种做法特别适合变量有上(下)界约束的情况. 第二, 通常会考虑求罚函数的近似极小点, 见式(9.1.16).

9.3 ℓ_1 精确罚函数

精确罚函数是非线性规划中一个很重要的概念,它作为 $f(\mathbf{x}), \mathbf{c}(\mathbf{x})$ (也可能含这些函数的导数)的函数,也在问题(7.0.1)的解 \mathbf{x}^* 处取到局部极小值.与早期罚函数相比,精确罚函数的优点是,其中的罚参数不用趋于无穷即可得到原始问题的解(也是精确一词的由来).本节介绍最简单的精确罚函数,即违背约束条件就添加 ℓ_1 惩罚项.遗憾的是, ℓ_1 罚函数是非光滑或不可微的,从而无法完全利用许多有效的光滑最小化问题的方法求解.一个更实际的做法是把精确罚函数作为一个价值函数,结合其他的非线性规划迭代法一起使用,详见 9.4.3 小节和 9.4.4 小节.

需要指出的是,本节考虑的 ℓ_1 罚函数是学者研究最多,也是应用最广泛的,常见于非线性规划的应用以及最小偏差曲线拟合问题(也称最小一乘问题)等.对于含有等式和不等式约束的非线性规划问题(7.0.1),相应的 ℓ_1 罚函数是

$$\phi(\mathbf{x}) = \nu f(\mathbf{x}) + \sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \sum_{i \in \mathcal{I}} (c_i(\mathbf{x}))^+ \quad (9.3.1)$$

其中 $a^+ := \max(a, 0)$.参数 $\nu (\nu > 0)$ 等价于把参数 $\sigma = \nu^{-1}$ 乘到罚项上.

例 9.3.1 (ℓ_1 精确罚函数) 问题(9.1.20)的 ℓ_1 罚函数为

$$\phi(\mathbf{x}) = \nu(-x_1 - x_2) + \max(x_1^2 + x_2^2 - 1, 0) \quad (9.3.2)$$

其中 ν 满足 $0 < \nu < 1/\lambda^* = \sqrt{2}$. 罚函数 $\phi(\mathbf{x})$ 在 $\nu = 1$ 处的等值线如图 9.3.1 所示,而且显然 \mathbf{x}^* 是 $\phi(\mathbf{x})$ 的极小点. 很容易看到罚函数 $\phi(\mathbf{x})$ 的非光滑特性,即在单位圆(虚线)上具有弯曲的“沟槽”.

对于充分小的 ν ,罚函数(9.3.1)是精确的,即非线性规划问题(7.0.1)的局部解很大程度上等价于函数(9.3.1)的局部极小点.而大多数情况下,非线性规划问题和罚函数问题的解存在差异,比如前者有可行点,但是后者的极小点不满足非线性规划问题的约束条件.在这些情况下,它的优点在于避免了寻找初始可行点的麻烦,且当非线性规划问题无可行点时能确定最好(best)解.

本节的主旨是推导出 ℓ_1 罚函数的一阶和二阶最优化条件;另外,还考虑了非线性规划问题的局部解和精确罚函数的局部极小点的等价性,并讨论了 ℓ_1 罚函数(9.3.1)在二次规划中的应用.首先, ϕ 在 \mathbf{x}' 处的方向导数

$$\phi_p(\mathbf{x}') := \lim_{\theta \downarrow 0} \frac{\phi(\mathbf{x}(\theta)) - \phi(\mathbf{x}')}{\theta}$$

其中 $\mathbf{x}(\theta) = \mathbf{x}' + \theta \mathbf{p}$ 是一条以 \mathbf{x}' 为起点的射线,或者更一般地,它可以是一条弧或一个方向序列.由方向导数的定义可以直接验证下面的事实.

引理 9.3.1 ℓ_1 精确罚函数(9.3.1)的方向导数

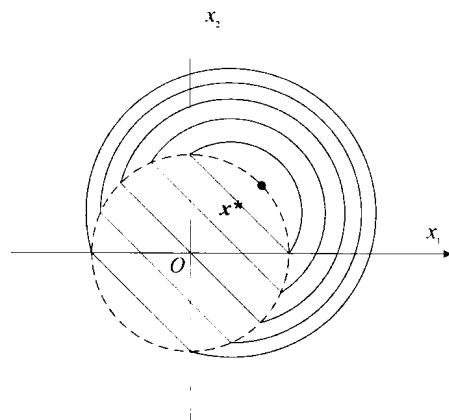


图 9.3.1 精确罚函数(9.3.2)的等值线

$$\phi_p(x) = \nu \mathbf{p}^\top \mathbf{g} + \sum_{i \in \mathcal{I}, c_i > 0} \mathbf{p}^\top \mathbf{a}_i + \sum_{i \in \mathcal{I}, c_i = 0} (\mathbf{p}^\top \mathbf{a}_i)^+ + \sum_{i \in \mathcal{E}, c_i \neq 0} \text{sign}(c_i) \mathbf{p}^\top \mathbf{a}_i + \sum_{i \in \mathcal{E}, c_i = 0} |\mathbf{p}^\top \mathbf{a}_i| \quad (9.3.3)$$

为了表述方便, 假设式(9.3.1)中的集合 \mathcal{E} 是空集, 从而变为不等式问题(9.1.2)的精确罚函数. 这些结论对一般问题(7.0.1)也成立. 令 $\mathcal{I}^* := \{i \in \mathcal{I} : c_i(x^*) = 0\}$ 表示在点 x^* 的不等式约束取到等式的指标集合. 假设向量 $\mathbf{a}_i^*, i \in \mathcal{I}^*$, 是线性无关的, 则易得 x^* 极小化 $\phi(x)$ 的一阶必要条件. 为此, 先考虑下面的命题.

命题 9.3.1 方向导数

$$\phi_p^* := \phi_p(x^*) \geq 0 \quad (9.3.4)$$

当且仅当存在乘子 $\lambda_i^* (i \in \mathcal{I}^*)$ 满足条件

$$\nu \mathbf{g}^* + \sum_{i \in \mathcal{I}, c_i^* > 0} \mathbf{a}_i^* + \sum_{i \in \mathcal{I}^*} \lambda_i^* \mathbf{a}_i^* = \mathbf{0} \quad (9.3.5a)$$

$$0 \leq \lambda_i^* \leq 1, \quad i \in \mathcal{I}^* \quad (9.3.5b)$$

证明 把式(9.3.5)代入式(9.3.3), 得到

$$\phi_p^* = \sum_{i \in \mathcal{I}^*} [(\mathbf{p}^\top \mathbf{a}_i^*)^+ - \lambda_i^* \mathbf{p}^\top \mathbf{a}_i^*] \geq 0 \quad (9.3.6)$$

下面证明必要性. 假设不存在满足条件(9.3.5)的乘子 λ_i^* , 则可构造方向 \mathbf{p} 使 $\phi_p^* < 0$, 从而与式(9.3.4)矛盾. 用 \mathbf{A}^* 表示以 $\mathbf{a}_i^* (i \in \mathcal{I}^*)$ 为列的矩阵. 如果条件(9.3.5a)不满足, 则由式(7.2.4)的构造可推出, 存在向量 $\boldsymbol{\lambda}$ 和 $\boldsymbol{\mu} \neq \mathbf{0}$ 使 $\mathbf{A}^{*\top} \boldsymbol{\mu} = \mathbf{0}, \bar{\mathbf{g}}^* + \mathbf{A}^* \boldsymbol{\lambda} + \boldsymbol{\mu} = \mathbf{0}$ 均成立, 其中

$$\bar{\mathbf{g}}^* = \nu \mathbf{g}^* + \sum_{i \in \mathcal{I}, c_i^* > 0} \mathbf{a}_i^* \quad (9.3.7)$$

令 $\mathbf{p} = \boldsymbol{\mu}$, 可得

$$\phi_p^* = \mathbf{p}^\top \bar{\mathbf{g}}^* = -\boldsymbol{\mu}^\top \boldsymbol{\mu} < 0$$

因此, \mathbf{p} 是所需的下降方向. 另一种情况是式(9.3.5a)成立, 但是对于某个 $p \in \mathcal{I}^*$, 式(9.3.5b)不成立. 由独立性假设知 $\mathbf{A}^{*\top} = (\mathbf{A}^{*\top} \mathbf{A}^*)^{-1} \mathbf{A}^{*\top}$ 必存在, 这时考虑向量 $\mathbf{p} = \mathbf{A}^{*\top} \mathbf{e}_p$, 可推知 $\mathbf{p}^\top \mathbf{a}_p^* = 1$, $\mathbf{p}^\top \mathbf{a}_i^* = 0, i \in \mathcal{I}^*, i \neq p$. 因而

$$\phi_p^* = \sum_{i \in \mathcal{I}^*} [(\mathbf{p}^\top \mathbf{a}_i^*)^+ - \lambda_i^* \mathbf{p}^\top \mathbf{a}_i^*] = 1 - \lambda_p^*$$

若式(9.3.5b)是因为 $\lambda_p^* > 1$ 不成立, 则由此构造了一个下降方向. 最后, 若式(9.3.5b)是因为 $\lambda_p^* < 0$ 不成立, 则方向 $\mathbf{p} = -\mathbf{A}^{*\top} \mathbf{e}_p$ 满足 $\phi_p^* = \lambda_p^*$, 从而 \mathbf{p} 仍为下降方向. ■

总而言之, x^* 最小化 $\phi(x)$ 的一阶必要条件是 $\phi_p^* \geq 0$. 上面讨论还显示该条件等价于式(9.3.5). 对 $i \in \mathcal{I}$, 若 $c_i^* > 0$, 令 $\lambda_i^* = 1$; 若 $c_i^* < 0$, 令 $\lambda_i^* = 0$, 则这些条件可写为

$$\nu \mathbf{g}^* + \sum_{i \in \mathcal{I} \cup \mathcal{E}} \lambda_i^* \mathbf{a}_i^* = \mathbf{0} \quad (9.3.8a)$$

$$\left. \begin{array}{l} 0 \leq \lambda_i^* \leq 1 \\ c_i^* > 0 \Rightarrow \lambda_i^* = 1 \\ c_i^* < 0 \Rightarrow \lambda_i^* = 0 \end{array} \right\} \quad i \in \mathcal{I} \quad (9.3.8b)$$

$$\left. \begin{array}{l} -1 \leq \lambda_i^* \leq 1 \\ c_i^* \neq 0 \Rightarrow \lambda_i^* = \text{sign}(c_i^*) \end{array} \right\} \quad i \in \mathcal{E} \quad (9.3.8c)$$

实际上, 式(9.3.8)是一般形式的一阶条件, 这里用完全类似的推导方法得到关于 $i \in \mathcal{E}$ 的一阶条件. 非线性规划问题(7.0.1)的 KKT 条件(7.2.14)与该系统的差异不是很大. 式(9.3.8)含有参数 ν , 而式(7.2.14)中没有. 然而, 如果对问题(7.0.1)进行比例放缩, 以 $\nu f(x) (\nu > 0)$ 代替

$f(\mathbf{x})$ (这不会改变(7.0.1)的局部解),于是 KKT 条件中同样包含等式 $\nu g^* + \sum \lambda_i^* \mathbf{a}_i^* = \mathbf{0}$. 虽然式(9.3.8)中可能包含(7.0.1)的不可行点,但是如果 \mathbf{x}^* 不满足约束 $c_i(\mathbf{x}) = 0$,则相应乘子的值必为 $\lambda_i^* = \text{sign}(c_i^*)$. 因此,可以将函数(9.3.1)在点 \mathbf{x}^* 是局部光滑的项写成

$$\bar{f}(\mathbf{x}) = \nu f(\mathbf{x}) + \sum_{i \in \mathcal{I}^*} \lambda_i^* c_i(\mathbf{x}) \quad (9.3.9)$$

而且式(9.3.7)中的 \bar{g}^* 实际上是这些项的梯度向量. 这样, \mathbf{x}^* 等价地求解问题

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \bar{f}(\mathbf{x}) \\ & \text{subject to} && c_i(\mathbf{x}) = 0, \quad i \in \mathcal{I}^* \end{aligned} \quad (9.3.10)$$

并且乘子 $\lambda_i^*, i \in \mathcal{I}^*$, 也是该问题的 Lagrange 乘子.

式(7.2.14)和式(9.3.8)之间的另一个重要区别是对于 $i \in \mathcal{I}$, 后者有边界条件 $\lambda_i^* \leq 1$. 可以简单地解释为: 假设问题(9.3.10)对某约束 $p \in \mathcal{I}^*$ 有 $\lambda_p^* > 1$, 类似于上面的构造方法, 方向 $\mathbf{p} = \mathbf{A}^{*+T} \mathbf{e}_p$ 对于约束 p 是不可行的, 而且其方向导数 $\phi_p^* = 1 - \lambda_p^*$, 其中 $1 - \lambda_p^*$ 是光滑部分(9.3.9)的斜率, 1 是罚项 c_p^+ 的斜率. 由于 $\lambda_p^* > 1$, 罚项不足以占优光滑部分, 因而 $\phi(\mathbf{x})$ 在 \mathbf{x}^* 处取不到局部极小值. 图 9.3.2 是一个不等式约束时对应的各种情况. 图 9.3.2(b)和 9.3.2(c)显示的是 λ^* 不在 $[0,1]$ 内的情况, 这时可以看到罚函数 $\phi = f + c^+$ 在 $c^* = 0$ 处没有最小点; 而图 9.3.2(d)显示了 λ^* 在 $(0,1)$ 内的情况, 此时加入惩罚项就可得到一个局部极小点. 为了在图 9.3.2(c)中的 $\lambda^* > 1$ 的情况下也得到一个极小点, 很明显需要增加 c^+ 项的权值, 或者等价地缩小函数 f , 用 $\nu f (0 < \nu < 1/\lambda^*)$ 取代之. 这也阐明了当 \mathbf{x}^* 最小化 $\phi(\mathbf{x})$ (其中 \mathbf{x}^*, λ^* 是问题(7.0.1)的 KKT 点)时参数 ν 的取值的上限, 即如果

$$\nu \leq 1 / \|\lambda^*\| \quad (9.3.11)$$

那么一阶条件(9.3.8)和比例变换后问题的 KKT 条件都成立, 而且 \mathbf{x}^* 满足两个问题的一阶必要条件. 但若 $\nu > 1 / \|\lambda^*\|$, 且存在某个 p 对于比例变换问题而言 $\lambda_p^* > 1$, 则沿着上面构造的方向离开 \mathbf{x}^* , 会使比例变换问题和罚函数的值减小. 需要注意的是, ν 的上限值只依赖于一阶信息, 而不像 9.1.1 小节和 9.2 节那样需要二阶信息.

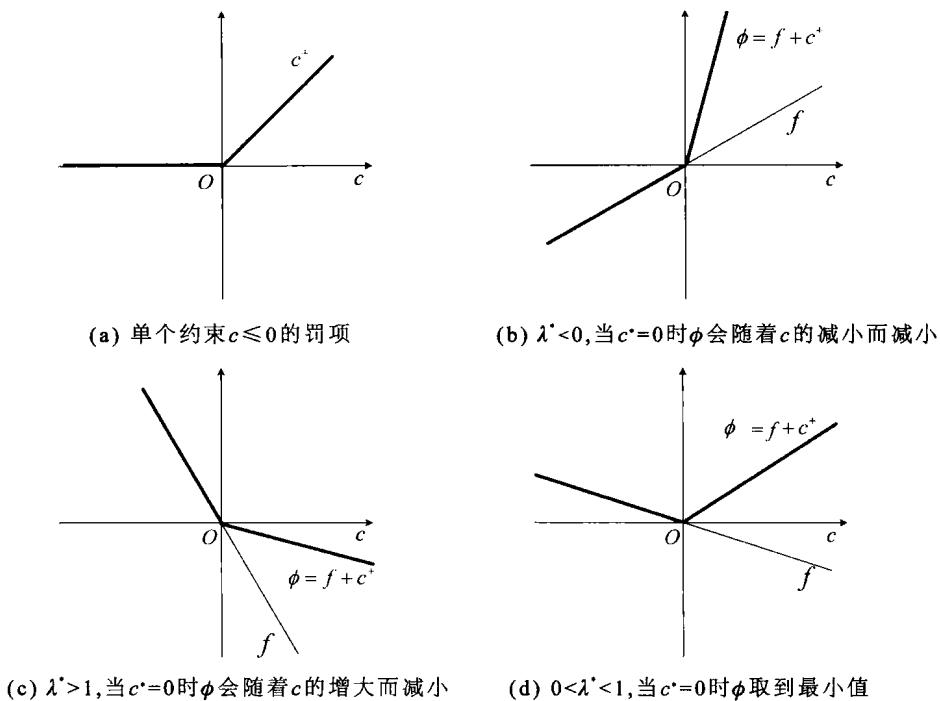
类似地, 可以得到二阶条件. 设 \mathbf{x}^* 满足一阶条件(9.3.5)(\mathcal{E} 为空集), 此外, 对所有 $i \in \mathcal{I}^*$ 设 $0 < \lambda_i^* < 1$ (本质上是 7.2 节中严格互补性的推广). 如果对某个 $i \in \mathcal{I}^*$, $\mathbf{p}^T \mathbf{a}_i^* \neq 0$, 那么不等式(9.3.6)严格成立, 因此这些方向都是严格上升方向($\phi_i^* > 0$), 此时二阶项的作用无关紧要. 可是, 若 \mathbf{p} 对所有 $i \in \mathcal{I}^*$ 满足 $\mathbf{p}^T \mathbf{a}_i^* = 0$, 则 $\phi_i^* = 0$, 从而二阶项的作用至关重要. 对任意满足这些条件的向量 \mathbf{p} , 如引理 7.3.3(ii), 根据独立性假设可构造一条弧 $\mathbf{x}(\theta), \theta \in [0, \bar{\theta}]$. 它满足 $\mathbf{x}(0) = \mathbf{x}^*, \dot{\mathbf{x}}(0) = \mathbf{p}$, 且对所有 $i \in \mathcal{I}^*$, $c_i(\mathbf{x}(\theta)) = 0$. 利用 Lagrange 函数, 对于充分小的 θ , 由式(9.3.1)和式(9.3.8b), 并利用 Taylor 展式有

$$\phi(\mathbf{x}) = \mathcal{L}(\mathbf{x}(\theta), \lambda^*) = \phi^* + \frac{1}{2} \mathbf{s}^T \mathbf{W}^* \mathbf{s} + o(\mathbf{s}^T \mathbf{s}) \quad (9.3.12)$$

其中 $\mathbf{W}^* = \nabla^2 \mathcal{L}(\mathbf{x}^*, \lambda^*)$, $\mathbf{s} = \mathbf{x}(\theta) - \mathbf{x}^*$. 由 ϕ^* 的最优性, 并令 $\theta \downarrow 0$, 可得

$$\mathbf{s}^T \mathbf{W}^* \mathbf{s} \geq 0, \quad \forall \mathbf{s} \in \{\mathbf{s}: \mathbf{s}^T \mathbf{a}_i^* = 0, i \in \mathcal{I}^*\} \quad (9.3.13)$$

这是 \mathbf{x}^* 最小化 $\phi(\mathbf{x})$ 的二阶必要条件. 如果不等式(9.3.13)严格成立(及式(9.3.5)和严格互补性成立), 那么它也是充分的. 同一阶条件一样, 非线性规划(7.4.7)的二阶条件与 ℓ 罚函数的二阶条件密切相关. 实际上若 \mathbf{x}^* 可行, 并假设 $\lambda^* < 1$, 则它们是等价的. $\lambda^* \leq 1$ 保证与式(9.3.5b)的等价性, 而 $\lambda^* < 1$ 保证严格互补性, 若此条件不满足, 则结论不成立(见例 9.3.2).

图 9.3.2 ℓ_1 精确罚函数的最优性条件 ($\nu = 1$)

也可以用上面关于最优性条件的等价性讨论来说明精确罚函数与非线性规划问题的解的等价性. 假设 x^* 是后者的可行解, 此时上面的结论可总结为: 如果 $\nu \leq 1/\|\lambda^*\|_\infty$, 则满足一阶条件的点(即 KKT 点)是等价的; 若满足某种正则性条件(如向量 a_i^* , $i \in \mathcal{A}^*$, 线性无关)且 $\nu < 1/\|\lambda^*\|_\infty$, 则满足二阶条件的点是等价的. 另外, 若 x^* 还满足相互等价的二阶充分条件, 则它是这两个问题的严格局部极小解. 易给出一些简单的例子说明不等价的情况.

例 9.3.2 对于问题

$$\begin{aligned} & \text{minimize} && 0 \\ & \text{subject to} && x^3 + 3x^2 + 3 = 0 \end{aligned}$$

ℓ_1 罚函数的局部极小点 $x^* = 0$ 不是该问题的可行点.

下面的几个问题由于不同的原因导致非线性规划问题的解 $x^* = 0$ 也不是参数 $\nu = 1$ 的 ℓ_1 罚函数的极小点. 对于问题

$$\begin{aligned} & \text{minimize} && x \\ & \text{subject to} && x^2 \leq 0 \end{aligned}$$

x^* 不是 KKT 点, 且线性无关条件不成立. 对于问题

$$\begin{aligned} & \text{minimize} && x^3 \\ & \text{subject to} && x^5 \geq 0 \end{aligned}$$

曲率条件不是严格成立的(x^* 满足二阶条件). 对于问题

$$\begin{aligned} & \text{minimize} && x - \frac{1}{2}x^2 \\ & \text{subject to} && 0 \leq x \leq 1 \end{aligned}$$

$\lambda^* = 1$, 因此条件 $\nu < 1/\|\lambda^*\|_\infty$ 不是严格成立的(x^* 是一个 KKT 点, 即一阶极小点). 最后, 对

于问题

$$\begin{aligned} & \text{minimize} && 2x - x^2 \\ & \text{subject to} && 0 \leq x \leq 1 \end{aligned}$$

$\lambda^* = 2$, 从而条件 $\nu \leq 1/\|\lambda^*\|$ 不满足, x^* 甚至不是一阶极小点. 上述例外只会在极端情况下出现, 且 x^* 是可行的, 可以选取有效的参数 ν . 因此, 从实用的目的讲, 可以假设非线性规划问题和极小化 ℓ_1 罚函数是等价的.

当 $\phi(x)$ 具有简单约束

$$l \leq x \leq u \quad (9.3.14)$$

时, 推广上面的讨论可产生另一组有用的一阶条件. 不失一般性, 可以假定 $l < u$. 若为式(9.3.14)中的界约束配置乘子, 则必要条件是 x^* 可行, 且存在 λ^* 和 π^* 满足式(9.3.8b)、式(9.3.8c)及

$$\begin{aligned} \nu g^* + \sum_{i \in I \cup E} \lambda_i^* a_i^* - \pi^* &= 0 \\ \pi_j^* \geq 0, \quad x_j^* = l_j \\ \pi_j^* \leq 0, \quad x_j^* = u_j \\ \pi_j^* = 0, \quad \text{其他} \end{aligned} \quad j = 1, 2, \dots, n \quad (9.3.15)$$

也易于得到与式(9.3.13)类似的二阶条件.

一种特殊的 ℓ_1 罚函数是 ℓ_1 二次规划(ℓ_1 QP)问题, 比如将二次规划算法的第 I 阶段与第 II 阶段结合起来时所碰到的问题(类似于线性规划的大 M 法), 以及求解非线性规划的 $S\ell_1$ QP 法中的子问题(见 9.4.4 小节)等. 如果对不同类型的约束配置不同的相对权重, 则 ℓ_1 QP 问题可以表述为

$$\begin{aligned} & \text{minimize} && \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{p}^T (\mathbf{A}^T \mathbf{x} - \mathbf{b})^+ + \boldsymbol{\delta}^T (\mathbf{A}^T \mathbf{x} - \mathbf{b}) \\ & \text{subject to} && l \leq x \leq u \end{aligned} \quad (9.3.16)$$

其中 $\mathbf{A} \in \mathbb{R}^{n \times m}$, $a^- = \max(-a, 0) = (a)^+$. 比如对于单位权重, 可以按如下方式设置:

$$\text{对 } \mathbf{a}_i^T \mathbf{x} \geq b_i, \quad \rho_i = 0, \quad \delta_i = 1$$

$$\text{对 } \mathbf{a}_i^T \mathbf{x} \leq b_i, \quad \rho_i = 1, \quad \delta_i = 0$$

$$\text{对 } \mathbf{a}_i^T \mathbf{x} = b_i, \quad \rho_i = 1, \quad \delta_i = 1$$

尽管可以引入额外的变量把这个问题转化成普通的 QP 问题, 但设计积极集法时直接考虑它将更方便. 考虑由当前积极的 ℓ_1 罚项和界约束组成的积极集. 先由 8.1 节描述的方法得到 ℓ_1 罚函数的光滑部分在当前积极集的极小点, 之后在当前点 $\mathbf{x}^{(k)}$ 执行 8.2 节叙述的朝向该极小点的一维搜索. 若存在结点(knot), 即导数不连续的点, 则选取步为离无约束极小点最近的结点, 否则取无约束极小点. 如果两者都不能得到一个可行点, 则取至边界最近的步. 当一维搜索到达一个结点或边界时, 就给积极集添加一个约束. 若通过一维搜索得到 ℓ_1 罚函数光滑部分的极小点, 则再次利用 8.1 节中的方法和梯度中的光滑部分 $\bar{\mathbf{g}} = \mathbf{d} + \mathbf{G} \mathbf{x} + \sum_{i \in A} \lambda_i \mathbf{a}_i$ 为积极约束确定乘子. 此时的乘子若可行, 则当前点是最优的; 否则, 将最不可行乘子对应的约束松弛, 从而可进行下一步的一维搜索. Fletcher 于 1985 年简要讨论了使用这种方法时如何处理其中必需的矩阵分解问题.

最后, 讨论如何在实际应用中选取 ℓ_1 罚函数中的参数 ν , 尤其是一般情况下的非线性规划. 一般来说, ν 越小, 函数 f 相对于罚项在罚函数 $\phi(\mathbf{x})$ 中的作用越低, 进而位于积极约束切平面上的解的准确性也随之越低(参见图 9.1.2(c)). 而且, 当在 $c_i(\mathbf{x}) = 0$ 的表面上沿着一个弯

曲的沟槽进行一系列一维搜索时, 若 ν 减小, 则使得 ϕ 降低的校正长度也会减小, 随之一维搜索的迭代次数会增加. 因此, 保持 ν 足够大十分有利, 但须小于上限值 $1/\|\lambda^*\|_{\infty}$. 当积极约束的乘子都很小时, 表明 ν 太小了, 需要增大参数 ν . 此外, 也许会出现序列 $\phi^{(k)} \rightarrow -\infty$, 这需要减小 ν 或选取其他初始点. 如果积极乘子的量级相差很大, 则意味着应该对约束进行比例变换. 有些算法可自动完成这种比例变换, 但总体来说, 在这种情况下, 为每个约束设置自己的权重, 并可自动调整各自权重量级的罚函数会更好.

9.4 逐步二次规划法

罚函数法是以稍微间接一点的方式来尝试求解非线性约束问题. 一种更直接且有效的方法是基于问题中的函数 $f(\mathbf{x})$ 和 $c_i(\mathbf{x})$ 的某种近似的迭代法, 特别是利用约束函数 $c_i(\mathbf{x})$ 的线性近似. 基于这种思想的一个典型算法可以解释为利用 Newton 法找(仅有等式约束)问题的 Lagrange 函数的稳定点, 因此被称为 Lagrange-Newton 法. 在 Lagrange-Newton 法的基础上发展了逐步二次规划法. 该方法已经成为当前求解一般非线性约束规划问题的一类最重要的方法, 其特点是同时产生解向量 \mathbf{x}^* 和最优 Lagrange 乘子 λ^* 的近似序列 $\{(\mathbf{x}^{(k)}, \lambda^{(k)})\}$. 本节先介绍 Lagrange-Newton 法和基本逐步二次规划法, 然后详细介绍实用逐步二次规划法.

9.4.1 Lagrange-Newton 法

利用 7.2 节的 Lagrange 乘子法求解等式约束问题(9.1.1), 则稳定点条件(7.2.5)为

$$\mathbf{g}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\lambda = \mathbf{0}, \quad \mathbf{c}(\mathbf{x}) = \mathbf{0} \quad (9.4.1)$$

其中 $\mathbf{A}(\mathbf{x}) = [\nabla c_1(\mathbf{x}), \nabla c_2(\mathbf{x}), \dots, \nabla c_m(\mathbf{x})]$. 式(9.4.1)是关于原始变量 \mathbf{x} 和 Lagrange 乘子 λ 的非线性方程. 请注意该系统关于 λ 实际上是线性的. 只要知道 \mathbf{x} , 就可以方便地解出 λ .

现在假设 $(\mathbf{x}^{(k)}, \lambda^{(k)})$ 是式(9.4.1)的近似解. 像以往那样, 可以直接应用牛顿法来尝试提高这个近似解. 为此构造牛顿校正 $(\mathbf{s}^{(k)}, \mathbf{w}^{(k)})$ 满足

$$\begin{bmatrix} \mathbf{W}^{(k)} & \mathbf{A}^{(k)} \\ \mathbf{A}^{(k)T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(k)} \\ \mathbf{w}^{(k)} \end{bmatrix} = - \begin{bmatrix} \mathbf{g}^{(k)} + \mathbf{A}^{(k)}\lambda^{(k)} \\ \mathbf{c}^{(k)} \end{bmatrix} \quad (9.4.2)$$

这里 $\mathbf{A}^{(k)} = \mathbf{A}(\mathbf{x}^{(k)})$, $\mathbf{W}^{(k)} = \nabla^2 f(\mathbf{x}^{(k)}) + \sum \lambda_i^{(k)} \nabla^2 c_i(\mathbf{x}^{(k)})$. 令 $\lambda^{(k+1)} = \lambda^{(k)} + \mathbf{w}^{(k)}$, 则方程(9.4.2)变成

$$\begin{bmatrix} \mathbf{W}^{(k)} & \mathbf{A}^{(k)} \\ \mathbf{A}^{(k)T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(k)} \\ \lambda^{(k+1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{g}^{(k)} \\ \mathbf{c}^{(k)} \end{bmatrix} \quad (9.4.3)$$

Lagrange-Newton 法在当前迭代点构造并解方程组(9.4.3)得到 $(\mathbf{s}^{(k)}, \lambda^{(k+1)})$. 然后令

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)} \quad (9.4.4)$$

9.4.2 基本逐步二次规划法

直接推广 Lagrange-Newton 法来求解一般的约束优化问题(7.0.1)的困难之处在于, Lagrange-Newton 法(9.4.3)和式(9.4.4)不适用于不等式约束. 为此, 利用另外一种方式来重新表述这种方法, 即考虑二次规划子问题

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} s^T \mathbf{W}^{(k)} s + s^T \mathbf{g}^{(k)} + f^{(k)} \\ & \text{subject to} \quad \mathbf{A}^{(k)^T} s + \mathbf{c}^{(k)} = \mathbf{0} \end{aligned} \quad (9.4.5)$$

该问题的一阶条件正是式(9.4.3). 所以方程组(9.4.3)的解 $s^{(k)}$ 是式(9.4.5)的 KKT 点. 如果既约 Hessian 阵 $\mathbf{Z}^{(k)^T} \mathbf{W}^{(k)} \mathbf{Z}^{(k)}$ 是正定的(等价于 $s^{(k)}$ 满足问题(9.4.5)的二阶充分条件), 则 $s^{(k)}$ 是问题(9.4.5)的全局解. 当 $(\mathbf{x}^{(k)}, \lambda^{(k)})$ 距 $(\mathbf{x}^*, \lambda^*)$ 充分近时, 由连续性知, 这些条件的确是满足的(见习题 9.13). 基于这些讨论, 可以得到基本 SQP 法, 即算法 9.4.1.

Algorithm 9.4.1 Local SQP method

- 1: Given $\mathbf{x}^{(0)}, \lambda^{(0)}$, set $k=0$;
 - 2: **while** convergence is not arise **do**
 - 3: compute $\mathbf{W}^{(k)}, \mathbf{A}^{(k)}, \mathbf{g}^{(k)}$ with $\mathbf{x}^{(k)}$ and $\lambda^{(k)}$;
 - 4: find the solution $s^{(k)}$ of problem(9.4.5) (or problem(9.4.6)) along with associated Lagrange multiplier $\lambda^{(k+1)}$;
 - 5: set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + s^{(k)}$ and increase k by 1;
 - 6: **end while**
 - 7: **return** $\mathbf{x}^{(k)}$ and $\lambda^{(k)}$ as \mathbf{x}^* and λ^* .
-

如果对每个 k , 子问题(9.4.5)有唯一极小点, 则算法 9.4.1 确定的迭代序列和 Lagrange-Netwon 法(式(9.4.3)和式(9.4.4))确定的序列是完全相同的, 但后者有可能收敛到问题(9.1.1)的一个非极小点的 KKT 点, 所以事实上算法 9.4.1 要优于 Lagrange-Netwon 法. 这种情况类似于无约束优化的牛顿法, 它的最好的解释是逐步极小化目标函数的二阶 Taylor 近似, 参见 5.1.2 小节.

子问题(9.4.5)清楚地显示了与原始问题(9.1.1)的联系. 用约束 $c_i(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 的一阶 Taylor 展式近似代替原始约束. 类似地, 用

$$\frac{1}{2} s^T \mathbf{W}^{(k)} s + s^T \mathbf{g}^{(k)} + f^{(k)}$$

代替问题(9.1.1)的目标函数, 这不是目标函数在 $\mathbf{x}^{(k)}$ 的二阶 Taylor 展式, 因为在 Hessian 阵中增加了约束的曲率项. 在子问题中包含二阶约束项是很重要的, 否则不能获取关于非线性约束的二阶收敛性. 用问题(9.1.5)可以说明这一点. 这个问题的目标函数是线性的, 正因为增加了约束的曲率才保证问题(9.4.5)有解. 此时, 因为包含了约束的曲率, 由算法 9.4.1 确定的序列是适当的.

很容易推广上面的解释来求解一般的约束优化问题(7.0.1). 将约束用线性 Taylor 展式代替, 将目标函数用对应的二次函数代替后得到子问题

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} s^T \mathbf{W}^{(k)} s + \mathbf{g}^{(k)^T} s + f^{(k)} \quad (9.4.6a)$$

$$\text{subject to} \quad \mathbf{a}_i^{(k)^T} s + c_i^{(k)} = 0, \quad i \in \mathcal{E} \quad (9.4.6b)$$

$$\mathbf{a}_i^{(k)^T} s + c_i^{(k)} \leq 0, \quad i \in \mathcal{I} \quad (9.4.6c)$$

因为子问题(9.4.5)和(9.4.6)都是二次规划, 这样可以得到上面的逐步二次规划(Sequential Quadratic Programming, SQP)法, 是 Wilson 于 1963 年首次提出此类方法的.

需要注意的是, 对于这样直接构造的二次规划子问题, 线性化约束可能是不相容的, 即子问题(9.4.6)没有可行解(见习题 9.10). 因此, 设计实用算法时, 在求解 QP 子问题之前需要

采用某种技术先解决它的相容性问题. 为了方便起见, 以下假设 QP 子问题总是相容的.

例 9.4.1 (基本 SQP 法) 考虑用基本 SQP 法求解例 7.1.1 中的问题, 初始估计 $\mathbf{x}^{(0)} = (1/2, 1)^T, \boldsymbol{\lambda}^{(0)} = \mathbf{0}$. 因为 $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, 故 $\mathbf{W}^{(0)}$ 中没有约束的曲率项; 又因为 $f(\mathbf{x})$ 是线性的, 所以 $\mathbf{W}^{(0)}$ 是零矩阵. 这样第一个子问题是线性规划, 且 $\mathbf{x}^{(1)}$ 是约束在 $\mathbf{x}^{(0)}$ 处线性化后所得多面集的顶点. 事实上, 尽管 $c_1(\mathbf{x}) \leq 0$ 在解处是非积极的, 但是对它线性化后所得约束条件使得第一个子问题是可解的. 此时线性规划子问题的解存在, 且解处的乘子向量 $\boldsymbol{\lambda}^{(1)} = (1/3, 2/3)^T$, 这表明两个线性化约束都是积极的. 这样, 第二次迭代时

$$\mathbf{W}^{(1)} = \mathbf{0} + \frac{1}{3} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \frac{2}{3} \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & \frac{4}{3} \end{bmatrix}$$

是正定的. 求解子问题, 发现第一个线性化约束变成非积极的, 因此 $\lambda_1^{(2)} = 0$. 这样找到了正确的积极集. 完整的计算结果见表 9.4.1, 在后几次迭代中, 观察到了类似于牛顿法那样的快速收敛. 如果基本 SQP 法可以求解原始问题, 则需要计算函数值和导数值的次数较少.

表 9.4.1 用基本 SQP 法求解例 7.1.1

k	$\mathbf{x}_1^{(k)}$	$\mathbf{x}_2^{(k)}$	$\lambda_1^{(k)}$	$\lambda_2^{(k)}$	$c_1^{(k)}$	$c_2^{(k)}$
0	$\frac{1}{2}$	1	0	0	$-\frac{3}{4}$	$\frac{1}{4}$
1	$\frac{11}{12}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0.173 611	0.284 722
2	0.747 120	0.686 252	0	0.730 415	-0.128 064	0.029 130
3	0.708 762	0.706 789	0	0.706 737	-0.204 445	0.001 893
4	0.707 107	0.707 108	0	0.707 105	-0.207 108	0.28×10^{-5}

表 9.4.1 说明了该方法的一个重要特征, 即方法最终是二阶收敛的. 这里针对等式约束问题(9.1.1)详细陈述该事实. 如果问题(9.1.1)的二阶充分条件在 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 处成立, 且 $\text{rank}(\mathbf{A}^*) = m$, 则 Lagrange 矩阵

$$\mathbf{K}^* = \begin{bmatrix} \mathbf{W}^* & \mathbf{A}^* \\ \mathbf{A}^{*\top} & \mathbf{0} \end{bmatrix} \quad (9.4.7)$$

是非奇异的(见习题 9.9). 该方法相当于应用牛顿法解方程组(9.4.1), 因此当 $\mathbf{x}^{(k)}$ 和 $\boldsymbol{\lambda}^{(k)}$ 分别充分接近 \mathbf{x}^* 和 $\boldsymbol{\lambda}^*$ 时, 可以得到二阶收敛性. 事实上, 因为这里的 Lagrange 乘子仅出现在涉及 $\mathbf{W}^{(k)}$ 的二阶项中, 从而它扮演的是一个相对次要的角色. 挖掘该事实可以得到下面更强的结论.

定理 9.4.1 (局部二阶收敛) 假设 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 满足问题(9.1.1)的二阶充分条件, 且 $\text{rank}(\mathbf{A}^*) = m$.

如果 $\mathbf{x}^{(0)}$ 充分接近 \mathbf{x}^* , 且 $\boldsymbol{\lambda}^{(0)}$ 使得 Lagrange 矩阵 $\begin{bmatrix} \mathbf{W}^{(0)} & \mathbf{A}^{(0)} \\ \mathbf{A}^{(0)\top} & \mathbf{0} \end{bmatrix}$ 是非奇异的, Lagrange-Newton

迭代(式(9.4.3)和式(9.4.4))收敛, 且是二次收敛的; 如果 $\boldsymbol{\lambda}^{(0)}$ 使得 QP 子问题(9.4.5)有唯一解 $\mathbf{s}^{(0)}$, 则同样的事实对 SQP 法(算法 9.4.1)也是成立的.

证明 定义误差 $\mathbf{h}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ 和 $\boldsymbol{\delta}_k = \boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^*$, 并假设 f, c_i 是二阶连续可微的, 且它们的 Hessian 阵的元素是 Lipschitz 连续的, 则在 $\mathbf{x}^{(k)}$ 处相关的 Taylor 级数为

$$\mathbf{c}^* = \mathbf{c}^{(k)} - \mathbf{A}^{(k)\top} \mathbf{h}^{(k)} + O(\|\mathbf{h}^{(k)}\|^2)$$

$$\mathbf{g}^* = \mathbf{g}^{(k)} - \nabla^2 f^{(k)} \mathbf{h}^{(k)} + O(\|\mathbf{h}^{(k)}\|^2)$$

$$\mathbf{a}_i^* = \mathbf{a}_i^{(k)} - \nabla^2 c_i^{(k)} \mathbf{h}^{(k)} + O(\|\mathbf{h}^{(k)}\|^2), \quad i = 1, 2, \dots, m$$

由这些等式以及 $\mathbf{g}^* + \mathbf{A}\lambda^* = \mathbf{0}$ 和式(9.4.3)可以推出 $\mathbf{h}^{(k+1)}$ ($= \mathbf{h}^{(k)} + \mathbf{s}^{(k)}$), $\boldsymbol{\delta}^{(k+1)}$ 满足方程

$$\begin{bmatrix} \mathbf{W}^{(k)} & \mathbf{A}^{(k)} \\ \mathbf{A}^{(k)\top} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{h}^{(k+1)} \\ \boldsymbol{\delta}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \sum_i \delta_i^{(k)} \nabla^2 c_i^{(k)} \mathbf{h}^{(k)} + O(\|\mathbf{h}^{(k)}\|^2) \\ O(\|\mathbf{h}^{(k)}\|^2) \end{bmatrix} = \begin{bmatrix} O(\|\mathbf{h}^{(k)}\|^2) + O(\|\mathbf{h}^{(k)}\| \|\boldsymbol{\delta}^{(k)}\|) \\ O(\|\mathbf{h}^{(k)}\|^2) \end{bmatrix} \quad (9.4.8)$$

因为 $(\mathbf{x}^*, \lambda^*)$ 处的 Lagrange 矩阵是非奇异的(见习题 9.9), 因此对于 $(\mathbf{x}^*, \lambda^*)$ 某邻域内的 $(\mathbf{x}^{(k)}, \lambda^{(k)})$ 有

$$\begin{bmatrix} \mathbf{h}^{(k+1)} \\ \boldsymbol{\delta}^{(k+1)} \end{bmatrix} = O(\|\mathbf{h}^{(k)}\|^2) + O(\|\mathbf{h}^{(k)}\| \|\boldsymbol{\delta}^{(k)}\|)$$

且存在常数 $c > 0$ 使得

$$\max(\|\mathbf{h}^{(k+1)}\|, \|\boldsymbol{\delta}^{(k+1)}\|) \leq c \|\mathbf{h}^{(k)}\| \max(\|\mathbf{h}^{(k)}\|, \|\boldsymbol{\delta}^{(k)}\|) \quad (9.4.9)$$

这样, 在一个更小的邻域内, 比如 $c \max(\|\mathbf{h}^{(k)}\|, \|\boldsymbol{\delta}^{(k)}\|) \leq \alpha < 1$, 有

$$\max(\|\mathbf{h}^{(k+1)}\|, \|\boldsymbol{\delta}^{(k+1)}\|) \leq \alpha \|\mathbf{h}^{(k)}\| \leq \alpha \max(\|\mathbf{h}^{(k)}\|, \|\boldsymbol{\delta}^{(k)}\|)$$

因此迭代收敛, 由式(9.4.9)知是二阶的. 令 $\mathbf{x}^{(0)}$ 在 \mathbf{x}^* 的某邻域内使得 $\mathbf{A}^{(0)}$ 是满秩的, 且 $\lambda^{(0)}$ 使得 Lagrange 矩阵 $\mathbf{K}^{(0)}$ 是非奇异的, 则 $\|\boldsymbol{\delta}^{(0)}\| \geq \|\mathbf{h}^{(0)}\|$. 因此, 同上存在常数 d 使得

$$\max(\|\mathbf{h}^{(1)}\|, \|\boldsymbol{\delta}^{(1)}\|) \leq d \|\mathbf{h}^{(0)}\| \|\boldsymbol{\delta}^{(0)}\|$$

如果 $\mathbf{x}^{(0)}$ 充分接近于 \mathbf{x}^* 使得 $\|\mathbf{h}^{(0)}\| < 1/(cd \|\boldsymbol{\delta}^{(0)}\|)$, 则 $\max(\|\mathbf{h}^{(1)}\|, \|\boldsymbol{\delta}^{(1)}\|) < 1/c$. 因此 $(\mathbf{x}^{(1)}, \lambda^{(1)})$ 在收敛区域内.

当 $\lambda^{(k)}$ 接近 λ^* 时, 二阶充分条件的连续性(见习题 9.13)可以保证: 对于所有的 k , Lagrange - Newton 法中的 $\mathbf{s}^{(k)}$ 是问题(9.4.5)的唯一解. 而由上面的讨论, 当 $k \geq 1$ 时, $\lambda^{(k)}$ 的确接近 λ^* , 从而恰当选取 $\lambda^{(0)}$ 的先验假设对于定理推导的后半部分而言是必不可少的. ■

式(9.4.8)的右边仅出现 $\boldsymbol{\delta}^{(k)}$ 的线性项表明 $\lambda^{(k)}$ 扮演一个相对次要的角色, 且定理中利用了该事实. 比如, 如果 $\mathbf{x}^{(0)} = \mathbf{x}^*$, 则不管 $\lambda^{(0)}$ 的误差有多大, 必有 $\mathbf{x}^{(1)} = \mathbf{x}^*$ 和 $\lambda^{(1)} = \lambda^*$. 这表明利用该方法时, 使得 $\mathbf{x}^{(0)}$ 接近 \mathbf{x}^* 比 $\lambda^{(0)}$ 接近 λ^* 更重要. 该事实与 9.2 节的乘子罚函数法刚好相反, 那里对于固定的 σ , 非精确的 $\lambda^{(0)}$ 肯定会限制 $\mathbf{x}(\lambda^{(0)})$ 接近于 \mathbf{x}^* 的程度. 也可以将该定理推广到有不等式约束的优化问题.

SQP 法一个潜在的缺点是需要计算二阶导数才能得到 $\mathbf{W}^{(k)}$. 已经提出了各种类似于拟牛顿法中那样的更新公式, 即利用修正矩阵 $\mathbf{B}^{(k)}$ 近似 $\mathbf{W}^{(k)}$. Han 于 1967 年建议利用 DFP 公式, 然而其中的 $\mathbf{y}^{(k)}$ 定义为

$$\mathbf{y}^{(k)} = \nabla_x \mathcal{L}(\mathbf{x}^{(k+1)}, \lambda^{(k+1)}) - \nabla_x \mathcal{L}(\mathbf{x}^{(k)}, \lambda^{(k+1)}) \quad (9.4.10)$$

并证明所得到的算法是超线性收敛的. 考虑到 BFGS 公式在无约束优化问题求解中的成功, Powell 于 1978 年建议使用 BFGS 公式来近似 $\mathbf{W}^{(k)}$, 并尽可能保持矩阵正定以使子问题总是适定的. 为此, 置

$$\mathbf{r}^{(k)} = \theta \mathbf{y}^{(k)} + (1 - \theta) \mathbf{B}^{(k)} \mathbf{s}^{(k)}, \quad 0 \leq \theta \leq 1 \quad (9.4.11)$$

并选取参数 θ 使得 $\mathbf{r}^{(k)}$ 在条件 $\mathbf{s}^{(k)\top} \mathbf{r}^{(k)} \geq 0.2 \mathbf{s}^{(k)\top} \mathbf{B}^{(k)} \mathbf{s}^{(k)}$ 下最接近于 $\mathbf{y}^{(k)}$, 然后在 BFGS 更新公式中用 $\mathbf{r}^{(k)}$ 代替 $\mathbf{y}^{(k)}$. 因为 \mathbf{W}^* 不必是正定的, 因此 $\mathbf{B}^{(k)}$ 永远也不可能接近 \mathbf{W}^* . 读者可能会觉得这种做法中人为的痕迹过于明显, 然而就所关心的子问题的解而言, $\mathbf{B}^{(k)}$ 和 \mathbf{W}^* 在积极约束的切平

面的投影有可能很接近, 这是 $\mathbf{B}^{(k)}$ 中重要的部分. Powell 于 1978 年运用该事实证明: 即使 \mathbf{W}^* 是不定的, 也可以得到超线性收敛性. 因此, 人们也将 SQP 法称为 Wilson-Han-Powell (WHP) 法.

实践证明, SQP 法的拟牛顿法版本能够有效求解中小规模的问题. 但这种方法的困难是不能充分利用矩阵 $\mathbf{W}^{(k)}$ 中的稀疏性. 该缺陷限制了方法所能求解问题的规模.

上述结论均表明 SQP 法的局部性质已经非常满意了, 但是当初始点离解较远时, 方法可能不收敛, 子问题(9.4.6)甚至无解(可能是无界或者不可行). 因此, 这里的主要困难是克服方法对初始点的依赖性, 即当初始估计离 $(\mathbf{x}^*, \lambda^*)$ 很远时, 也保证方法收敛. 为了使得方法大范围收敛, 需要利用与无约束优化类似的线搜索或者信赖域技术, 同时还需要恰当地对所得到的 $(\mathbf{x}^{(k)}, \lambda^{(k)})$ 的优良性进行评价.

如果 $\mathbf{W}^{(k)}$ 在积极约束的切空间是正定的, 则二次规划子问题(9.4.6)有唯一解. 当 $\mathbf{W}^{(k)}$ 不正定时, 线搜索法用正定矩阵 $\mathbf{B}^{(k)}$ 代替它, 或者在矩阵分解过程中直接修正 $\mathbf{W}^{(k)}$. 在所有这些情况下, 子问题(9.4.6)将变成适定的, 但是这些修正可能会在模型中引入一些不必要的扭曲.

信赖域 SQP 法给子问题添加一个约束, 将步限制在模型(9.4.6)是一个相当可靠的近似问题的区域内. 这种方法能够处理不定的 Hessian 阵 $\mathbf{W}^{(k)}$. 然而引入信赖域有可能使子问题变得不可行, 而处理这种情况的程序将会使算法复杂化, 从而增加它们的计算开销. 一般而言, 不能说线搜索 SQP 或信赖域 SQP 谁更优越, 只能具体问题具体分析.

用来接受步或者拒绝步的策略也会影响 SQP 法的性能. 对于无约束优化, 用目标函数 f 来进行简单的评价. 对于约束优化问题, 通常利用价值函数或者滤子技术等策略. 与无约束优化中 f 作为价值函数在整个优化过程中保持不变的事实相比, 在约束优化中必须更新这些技术中的参数或者条目, 才能与 SQP 法产生的步相匹配.

9.4.3 ℓ_1 价值函数

SQP 法通常需要用价值函数来判断试探点的好坏. 在线搜索法中, 价值函数控制步长的大小; 在信赖域法中, 它决定是拒绝试探步还是接受试探步, 并且还决定是否需要调整信赖域半径. 在 SQP 法中已经使用了许多价值函数, 比如非光滑的罚函数和光滑的增广 Lagrange 函数. 这里仅限于讨论典型的精确非光滑的 ℓ_1 价值函数.

定义 $\bar{c}_i(\mathbf{x}) = c_i(\mathbf{x}), i \in \mathcal{E}; \bar{c}_i(\mathbf{x}) = (c_i(\mathbf{x}))^+, i \in \mathcal{I}$. 显然 \mathbf{x} 是问题(7.0.1)的可行解当且仅当 $\bar{c}(\mathbf{x}) = \mathbf{0}$, 所以称 $\bar{c}(\mathbf{x})$ 为约束违反度 (constraint violation). 问题(7.0.1)的 ℓ_1 价值函数即精确罚函数(9.3.1), 用约束违反度可以表示为

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \sigma \|\bar{c}(\mathbf{x})\|_1 \quad (9.4.12)$$

在线搜索法中, 假设当前点为 \mathbf{x}' , 搜索方向为 \mathbf{p}' , 如果充分减少条件

$$\phi(\mathbf{x}' + \alpha \mathbf{p}') \leq \phi(\mathbf{x}') + \rho \phi_{\mathbf{p}'}(\mathbf{x}') \alpha \quad (9.4.13)$$

成立, 将接受试探步 $\alpha \mathbf{p}'$. 这里 $\phi_{\mathbf{p}'}(\mathbf{x}')$ 表示 ϕ 沿方向 \mathbf{p}' 的方向导数, $\rho \in (0, 1)$ 是参数. 该要求类似于无约束优化中的 Armijo 条件(4.3.1), 那里要求 \mathbf{p}' 是下降方向, 即 $f_{\mathbf{p}'}(\mathbf{x}') = \mathbf{g}'^\top \mathbf{p}' < 0$. 在约束优化中, 当罚参数 σ 选取得充分大时, 下面的结论表明这个下降条件也成立.

引理 9.4.1 在 QP 子问题(9.4.6)中, 记 $\mathbf{x}' = \mathbf{x}^{(k)}$. 设 \mathbf{p}' 和 λ' 分别为 QP 子问题(9.4.6)的解和对应的 Lagrange 乘子, 则 $\phi(\mathbf{x}' + \alpha \mathbf{p}')$ 在 $\alpha = 0$ 处可微, 且

$$\phi_{\mathbf{p}'}(\mathbf{x}') \leq \mathbf{g}'^\top \mathbf{p}' - \sigma \|\bar{c}'\|_1 \quad (9.4.14)$$

其中 $\|\mathbf{c}'\|_1 = \sum_{i \in \mathcal{E}} |c_i(\mathbf{x}')| + \sum_{i \in \mathcal{I}} (c_i(\mathbf{x}'))^+$. 如果没有不等式约束 ($\mathcal{I} = \emptyset$), 则上述的不等式可以取到等号. 进一步, 如果 $\sigma > \|\lambda'\|_1$ 且 $\mathbf{p}'^\top \mathbf{B}' \mathbf{p}' > 0$, 则 \mathbf{p}' 是罚函数(9.4.12)在 \mathbf{x}' 的下降方向.

证明 当 $\mathcal{I} = \emptyset$ 时, 由引理 9.3.1 和 \mathbf{p}' 满足式(9.4.6b)有

$$\phi_{\mathbf{p}'}(\mathbf{x}') = \mathbf{g}'^\top \mathbf{p}' - \sigma \sum_{i \in \mathcal{E}} |c'_i|$$

当 $\mathcal{I} \neq \emptyset$ 时, 由 \mathbf{p}' 还满足式(9.4.6c)得

$$\phi_{\mathbf{p}'}(\mathbf{x}') \leq \mathbf{g}'^\top \mathbf{p}' - \sigma \left[\sum_{i \in \mathcal{E}} |c'_i| + \sum_{i \in \mathcal{I}} (c'_i)^+ \right]$$

此即不等式(9.4.14).

进一步, QP 子问题(9.4.6)的 KKT 条件要求

$$\mathbf{W}' \mathbf{p}' + \mathbf{g}' + \sum_{i \in \mathcal{E}} \lambda'_i \mathbf{a}'_i + \sum_{i \in \mathcal{I}} \lambda'_i \mathbf{a}'_i = \mathbf{0}$$

和

$$\lambda'_i \geq 0, \quad \lambda'_i (\mathbf{a}'_i^\top \mathbf{p}' + c'_i) = 0, \quad i \in \mathcal{I}$$

这些条件和可行性条件(9.4.6b)蕴含着

$$\mathbf{g}'^\top \mathbf{p}' = -\mathbf{p}'^\top \mathbf{W}' \mathbf{p}' + \sum_{i \in \mathcal{E}} \lambda'_i c'_i + \sum_{i \in \mathcal{I}} \lambda'_i c'_i$$

将此式代入不等式(9.4.14), 得

$$\phi_{\mathbf{p}'}(\mathbf{x}') \leq -\mathbf{p}'^\top \mathbf{W}' \mathbf{p}' + \sum_{i \in \mathcal{E}} [\lambda'_i c'_i - \sigma |c'_i|] + \sum_{i \in \mathcal{I}} [\lambda'_i c'_i - \sigma (c'_i)^+]$$

将 $\sigma > \|\lambda'\|_1$, $\mathbf{p}'^\top \mathbf{W}' \mathbf{p}' > 0$ 和 $\lambda'_i \geq 0 (i \in \mathcal{I})$ 代入上述不等式, 经演算得到方向导数 $\phi_{\mathbf{p}'}(\mathbf{x}') < 0$. ■

基于式(9.4.14), 一种选取 σ 的方法是要求方向导数充分负, 即

$$\phi_{\mathbf{p}^{(k)}}(\mathbf{x}^{(k)}) = \mathbf{g}^{(k)\top} \mathbf{p}^{(k)} - \sigma \|\bar{\mathbf{c}}^{(k)}\|_1 \leq -\eta \sigma \|\mathbf{c}^{(k)}\|_1 \quad (9.4.15)$$

对某 $\eta \in (0, 1)$ 成立. 如果

$$\sigma \geq \frac{\mathbf{g}^{(k)\top} \mathbf{p}^{(k)}}{(1 - \eta) \|\bar{\mathbf{c}}^{(k)}\|_1} \quad (9.4.16)$$

则不等式(9.4.15)是成立的. 这种选择不依赖于 Lagrange 乘子, 并且在实践中执行的效果很好.

另一种在线搜索和信赖域中都很有效的选择 σ 的策略是, 考虑试探步对价值函数的模型产生的效果. 定义 ϕ 的逐段二次模型为

$$\psi^{(k)}(\mathbf{p}) = f^{(k)} + \mathbf{g}^{(k)\top} \mathbf{p} + \frac{\tau}{2} \mathbf{p}^\top \mathbf{W}^{(k)} \mathbf{p} + \sigma m_1(\mathbf{p}) \quad (9.4.17)$$

其中 $m_1(\mathbf{p}) = \sum_{i \in \mathcal{E}} |c_i^{(k)} + \mathbf{a}_i^{(k)\top} \mathbf{p}| + \sum_{i \in \mathcal{I}} (c_i^{(k)} + \mathbf{a}_i^{(k)\top} \mathbf{p})^+$, $\tau \in (0, 1)$ 是下面将要给出的参数. 在得到步 $\mathbf{p}^{(k)}$ 后, 选择罚参数 σ 充分大以使得

$$\psi^{(k)}(\mathbf{0}) - \psi^{(k)}(\mathbf{p}^{(k)}) \geq \eta \sigma [m_1(\mathbf{0}) - m_1(\mathbf{p}^{(k)})] \quad (9.4.18)$$

成立, 其中参数 $\eta \in (0, 1)$. 由式(9.4.17)、式(9.4.6b)和式(9.4.6c)知, 不等式(9.4.18)对

$$\sigma \geq \frac{\mathbf{g}^{(k)\top} \mathbf{p}^{(k)} + \frac{\tau}{2} \mathbf{p}^{(k)\top} \mathbf{W}^{(k)} \mathbf{p}^{(k)}}{(1 - \eta) \|\bar{\mathbf{c}}^{(k)}\|_1} \quad (9.4.19)$$

成立. 如果 SQP 法的前一次迭代中的 σ 满足式(9.4.19), 则保持罚参数不变; 否则, 增大 σ 使得它以某种余量满足该不等式. 这里利用常数 τ 来处理 Hessian 阵 $\mathbf{W}^{(k)}$ 不正定的情况. 将 τ 定义为

$$\tau = \begin{cases} 1, & \mathbf{p}^{(k)T} \mathbf{W}^{(k)} \mathbf{p}^{(k)} > 0 \\ 0, & \text{其他} \end{cases} \quad (9.4.20)$$

如果 σ 满足式(9.4.19), 易于验证如此选择的 τ 使得 $\mathbf{p}^{(k)}$ 是价值函数 ϕ 的下降方向. 如果 $\tau=1$ 且 $\mathbf{p}^{(k)T} \mathbf{W}^{(k)} \mathbf{p}^{(k)} < 0$, 则该结论并不总是成立. 对比式(9.4.16)和式(9.4.19), 当 $\tau > 0$ 时, 基于式(9.4.19)的策略将选择一个较大的罚参数, 这样加大了约束违反度的权重. 如果步 $\mathbf{p}^{(k)}$ 使得约束违反度减小, 但是目标函数值增大, 则步被价值函数接受的机会将增大. 对于这种情况, 该性质是很有利的.

ℓ_1 精确罚函数的缺陷之一是 ℓ_1 项带来的非光滑性(SQP 和 $S\ell_1$ QP). 导数不连续将会在罚函数的表面产生沟槽, 并且这些沟槽是弯曲和单边陡峭的, 从而算法很难跟踪它们. 考虑由 Rosenbrock 函数(1.4.2)得到的方程

$$\begin{cases} c_1(\mathbf{x}) = \sigma(x_2 - x_1^2) \\ c_2(\mathbf{x}) = 1 - x_1 \end{cases} \quad (9.4.21)$$

ℓ_1 精确罚函数 $\phi(\mathbf{x}) = \|\mathbf{c}(\mathbf{x})\|_1$ 沿着抛物面 $x_2 = x_1^2$ 有一条沟槽, 同时随着 σ 增大, 导数沿沟槽的跳跃度也增大. 表 9.4.2 给出了 $S\ell_1$ QP 法的性能, 其中用的是标准初始点 $\mathbf{x}^{(0)} = (-1.2, 1)^T$. 可以看到性能会随着陡峭因子 σ 增大而恶化. 出现这种现象的原因是伴随着不连续性的线性化, 能使 $\phi(\mathbf{x})$ 降低的步的长度剧烈地缩小. 从表 9.4.2 中也能看到这一点.

表 9.4.2 由导数不连续性引起的慢收敛

σ	10	100	1 000
迭代次数	12	193	2 000
典型的信赖域半径	0.25	0.008	0.0007

导数不连续的另一个影响是 **Maratos 效应**(Maratos effect). 也就是说, ℓ_1 罚函数有可能破坏 SQP 法的超线性收敛, 即 $(\mathbf{x}^{(k)}, \lambda^{(k)})$ 可以任意接近 $(\mathbf{x}^*, \lambda^*)$, 但是由 SQP 法得到的单位步却不能使 ℓ_1 精确罚函数减小.

例 9.4.2 (Maratos 效应) 问题

$$\begin{aligned} & \text{minimize} && 3x_2^2 - 2x_1 \\ & \text{subject to} && x_1 - x_2^2 = 0 \end{aligned}$$

有唯一极小点 $\mathbf{x}^* = (0, 0)^T$, $\lambda^* = 2$, 且它满足二阶充分条件. 考虑任何接近 \mathbf{x}^* 的点 $\mathbf{x}(\epsilon) = (\epsilon^2, \epsilon)^T$. 取 $\mathbf{B}(\mathbf{x}(\epsilon)) = \mathbf{W}(\mathbf{x}^*, \lambda^*)$, 则 SQP 中的二次规划子问题为

$$\begin{aligned} & \text{minimize} && -2s_1 + 6\epsilon s_2 + s_2^2 \\ & \text{subject to} && s_1 - 2\epsilon s_2 = 0 \end{aligned}$$

解 $\mathbf{s}(\epsilon) = (-2\epsilon^2, -\epsilon)^T$. 于是有

$$\|\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon) - \mathbf{x}^*\| = O(\|\mathbf{x}(\epsilon) - \mathbf{x}^*\|^2)$$

因此, $\mathbf{s}(\epsilon)$ 是一超线性收敛步. 图 9.4.1 画出了 $\epsilon = 1/4$ 时问题的图示, 其中虚线表示目标函数的等值线, 实线表示约束曲线, \circ 表示解 \mathbf{x}^* .

直接计算还表明 $f(\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon)) = 2\epsilon^2$ 和 $c(\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon)) = -\epsilon^2$. 由于 $f(\mathbf{x}(\epsilon)) = \epsilon^2$ 和 $c(\mathbf{x}(\epsilon)) = 0$, 所以

$$f(\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon)) > f(\mathbf{x}(\epsilon)), \quad |c(\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon))| > |c(\mathbf{x}(\epsilon))|$$

也就是说, 尽管 $\mathbf{s}(\epsilon)$ 是一超线性收敛步, 即 $\mathbf{x}(\epsilon) + \mathbf{s}(\epsilon)$ 比 $\mathbf{x}(\epsilon)$ 远远近于 \mathbf{x}^* , 但无论是从目标

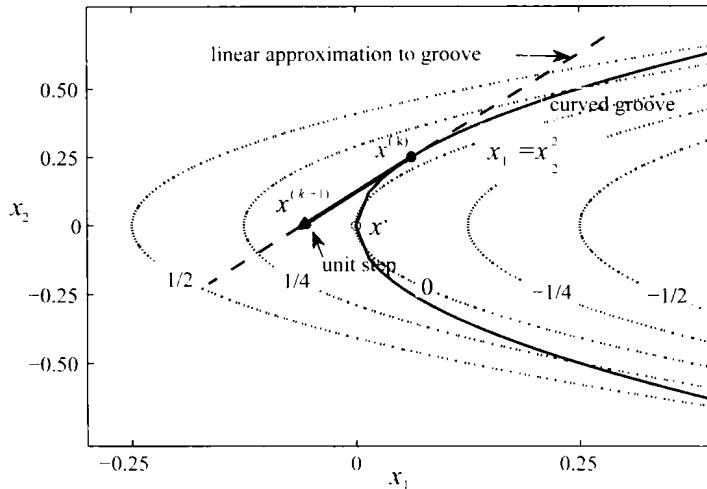


图 9.4.1 Maratos 效应

函数值来看,还是从约束的违反度来看, $x(\epsilon) + s(\epsilon)$ 都比 $x(\epsilon)$ “差”, 不会被 ℓ_1 价值函数接受. 事实上, 对于大部分常用罚函数 $\phi(x)$ 都有 $\phi(x(\epsilon) + s(\epsilon)) > \phi(x(\epsilon))$. Maratos 效应揭示了对于许多罚函数, 超线性收敛步并不一定能被接受, 从而有可能破坏算法的收敛性.

克服 Maratos 效应的方法主要有 3 种. 第 1 种是放松试探步的条件. 粗略地说, 既然试探步 $s^{(k)}$ 是一超线性收敛步, 应当在保证收敛性的前提下, 尽可能地接受 $\alpha_k = 1$, 即接受满步. 第 2 种是引进二阶校正步(second order correction step) $\hat{s}^{(k)}$ 的技巧, 其中 $\hat{s}^{(k)}$ 满足 $\|\hat{s}^{(k)}\| = O(\|s^{(k)}\|^2)$ 和 $\phi(x^{(k)} + s^{(k)} + \hat{s}^{(k)}) < \phi(x^{(k)})$. 这样, $s^{(k)} + \hat{s}^{(k)}$ 仍是一超线性步, 且它可被价值函数接受. 第 3 种是在算法中用光滑罚函数作为价值函数. 如果函数 $\phi(x)$ 是光滑的, 则只要 $s^{(k)}$ 是超线性步, 就有 $\phi(x^{(k)} + s^{(k)}) < \phi(x^{(k)})$.

9.4.4 实用逐步二次规划法

本节给出 3 个典型的实用逐步二次规划算法, 其中算法 9.4.2 基于线搜索技术, 算法 9.4.3 和 $S\ell_1$ QP 是基于信赖域技术的.

在算法 9.4.2 中, 为了表述的简洁性, 没有引入确保子问题是可行的和二阶校正步的机制, 而是简单地求解子问题(9.4.6)得到搜索方向; 同时, 假定二次规划(9.4.6)是凸的, 因此可以利用二次规划的积极集法, 即算法 8.2.1 求解它.

利用 warm-start 技术可以显著提高二次子问题的求解效率, 比如可以将每个 QP 子问题的积极集初始化为前一次 SQP 迭代的最终积极集. 这里没有指定用某种特定的拟牛顿近似, 比如对于大规模问题, 有限内存 BFGS 是适合的.

如果可以利用精确 Hessian 阵 $\mathbf{W}^{(k)}$, 则可以在必要时对它进行修正, 使得修正后的矩阵在等式约束的零空间上是正定的. 如果不利用价值函数, 则也可以在内部的 while 循环中利用一个滤子来确定步长 α_k . 如果由回溯线搜索产生的步长小于设定的阈值, 则还需要激发一个可行性恢复阶段. 最后, 无论利用的是价值函数还是滤子, 都应该耦合上一种类似于二阶校正的机制来克服 Maratos 效应.

Algorithm 9.4.2 Line search SQP algorithm

```

1: Choose parameters  $\rho \in (0, 0.5)$ ,  $\gamma \in (0, 1)$ , and an initial pair  $(\mathbf{x}^{(0)}, \boldsymbol{\lambda}^{(0)})$ ;
2: evaluate  $f^{(0)}, \nabla f^{(0)}, \mathbf{c}^{(0)}, \mathbf{A}^{(0)}$ ;
3: if a quasi-Newton approximation is used, choose an initial  $n \times n$  symmetric positive definite Hessian
   approximation  $\mathbf{B}^{(0)}$ , otherwise compute  $\mathbf{W}^{(0)}$ ;
4: repeat
5:   compute  $\mathbf{p}^{(k)}$  by solving problem(9.4.6) with the corresponding multiplier  $\hat{\boldsymbol{\lambda}}$ ;
6:   set  $\mathbf{p}_\lambda = \hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^{(k)}$ ;
7:   choose  $\sigma_k$  to satisfy inequality(9.4.19) with  $\tau = 1$ ;
8:   set  $\alpha_k = 1$ ;
9:   set  $a = \phi_p(\mathbf{x}^{(k)}, \sigma_k)$  be the directional derivative of  $\phi$  with parameter  $\sigma_k$  along the direction  $\mathbf{p}^{(k)}$ ;
10:  while  $\phi(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}, \sigma_k) > \phi(\mathbf{x}^{(k)}, \sigma_k) + \rho a \alpha_k$  do
11:    reset  $\alpha_k = \gamma_a \alpha_k$  for some  $\gamma_a \in (0, \gamma]$ ;
12:  end while
13:  set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ ;
14:  set  $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \alpha_k \mathbf{p}_\lambda$ ;
15:  evaluate  $f^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{c}^{(k+1)}, \mathbf{A}^{(k+1)}$  and possibly  $\mathbf{W}^{(k+1)}$ ;
16:  if a quasi-Newton approximation is used then
17:    set  $\mathbf{s}^{(k)} = \alpha_k \mathbf{p}^{(k)}$ ;
18:    set  $\mathbf{y}^{(k)} = \nabla_x \mathcal{L}(\mathbf{x}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) - \nabla_x \mathcal{L}(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k+1)})$ ;
19:    obtain  $\mathbf{B}^{(k+1)}$  by updating  $\mathbf{B}^{(k)}$  using a quasi-Newton formula;
20:  end if
21: until a convergence test is satisfied;
22: return  $\mathbf{x}^{(k)}$  and  $\boldsymbol{\lambda}^{(k)}$  as  $\mathbf{x}^*$  and  $\boldsymbol{\lambda}^*$ 

```

信赖域 SQP 法有一些迷人的性质, 比如它不要求 QP 子问题(9.4.6)中的 Hessian 阵是正定的, 且当 Hessian 阵和 Jacobi 矩阵非奇异时, 这种方法仍然可以控制步的质量, 并提供一种强迫收敛的机制. 信赖域 SQP 法最简单的表述是在 QP 子问题(9.4.6)中增加信赖域约束, 即

$$\underset{\mathbf{s}}{\text{minimize}} \quad \frac{1}{2} \mathbf{s}^\top \mathbf{W}^{(k)} \mathbf{s} + \mathbf{s}^\top \mathbf{g}^{(k)} + f^{(k)} \quad (9.4.22a)$$

$$\text{subject to} \quad \mathbf{a}_i^{(k)\top} \mathbf{s} + c_i^{(k)} = 0, \quad i \in \mathcal{E} \quad (9.4.22b)$$

$$\mathbf{a}_i^{(k)\top} \mathbf{s} + c_i^{(k)} \leq 0, \quad i \in \mathcal{I} \quad (9.4.22c)$$

$$\|\mathbf{s}\| \leq \Delta_k \quad (9.4.22d)$$

这种方法的主要障碍是线性化约束式(9.4.22b)和式(9.4.22c)在信赖域内可能无解, 图 9.4.2 是一个例子. 这里只有一个等式约束, 圆盘表示信赖域约束, 直线为线性化约束, 而满足线性化约束的步 \mathbf{s} 在信赖域之外. 该例说明, 如果限定了解的范数, 则原本相容的等式和不等式系统有可能无解.

为了解决该问题, 一种做法是简单地增大信赖域半径 Δ_k , 直到这些约束是相容的. 但是这种做法会违反信赖域法的初衷, 即定义一个信赖域, 并在这个区域内认为模型(9.4.22a)~(9.4.22c)可以准确地反映目标和约束函数的行为. 从分析上讲, 这种做法会损害算法的收敛行为.

一个更恰当的观点是:这种在每一步要求线性化约束精确满足是没有理由的. 这里的目标是在每一步提高这些约束的可行性,并在信赖域约束容许的情况下让它们精确地得到满足. 基于这种观点处理这个问题的方法主要有3种,即松弛法、惩罚法和滤子法. 本节给出的算法9.4.3是基于松弛法的.

下面以等式约束优化问题(9.1.1)为背景来描述这种方法,也可以推广该方法求解一般的优化问题. 在第 k 次迭代,首先选取松弛向量 $r^{(k)}$,然后求解子问题

$$\underset{s}{\text{minimize}} \quad \frac{1}{2} s^T \mathbf{W}^{(k)} s + s^T \mathbf{g}^{(k)} + f^{(k)} \quad (9.4.23a)$$

$$\text{subject to} \quad \mathbf{a}_i^{(k)T} s + c_i^{(k)} = r_i^{(k)}, \quad i \in \mathcal{E} \quad (9.4.23b)$$

$$\|s\|_2 \leq \Delta_k \quad (9.4.23c)$$

得到 SQP 步. $r^{(k)}$ 的选取会影响方法的有效性,具体地,首先缩小信赖域半径求解子问题

$$\underset{v}{\text{minimize}} \quad \|\mathbf{A}^{(k)T} v + \mathbf{c}^{(k)}\|_2^2 \quad (9.4.24a)$$

$$\text{subject to} \quad \|v\|_2 \leq 0.8\Delta_k \quad (9.4.24b)$$

记该问题的解为 $v^{(k)}$,定义 $r^{(k)} = \mathbf{A}^{(k)T} v^{(k)} + \mathbf{c}^{(k)}$. 这样,因为 $s = v^{(k)}$ 是问题(9.4.23)的可行解,因此子问题(9.4.23)的约束是相容的. 再求解(9.4.23)得到步 $s^{(k)}$. 下面考虑 Lagrange 乘子的估计.

设 $p^{(k)}$ 是 QP 子问题(9.4.5)的解, $\lambda^{(k+1)}$ 是与之对应的 Lagrange 乘子,则有

$$(\mathbf{A}^{(k)T} \mathbf{Y}^{(k)})^T \lambda^{(k+1)} = -\mathbf{Y}^{(k)T} (\mathbf{g}^{(k)} + \mathbf{W}^{(k)} p^{(k)}) \quad (9.4.25)$$

其中 $[\mathbf{Y}^{(k)} \quad \mathbf{Z}^{(k)}]$ 非奇异, $\mathbf{A}^{(k)T} \mathbf{Y}^{(k)}$ 非奇异,且 $\mathbf{A}^{(k)T} \mathbf{Z}^{(k)} = \mathbf{0}$. 可以删去上式右边关于 $p^{(k)}$ 的项,从而 $p^{(k)}$ 和 $\lambda^{(k+1)}$ 可以分开计算. 这种处理的解释是随着逐渐逼近解, $p^{(k)}$ 会收敛到零,而 $\mathbf{g}^{(k)}$ 通常不会收敛到零. 因此,在解附近,用这种方式得到的乘子的估计应该是 QP 乘子的一个很好的近似. 特别地,如果选 $\mathbf{Y}^{(k)} = \mathbf{A}^{(k)}$ (若 $\mathbf{A}^{(k)}$ 是列满秩的,则这种选取是有效的),则可得到

$$\hat{\lambda}^{(k+1)} = -(\mathbf{A}^{(k)T} \mathbf{A}^{(k)})^{-1} \mathbf{A}^{(k)T} \mathbf{g}^{(k)} \quad (9.4.26)$$

其恰是问题

$$\min_{\lambda} \|\nabla_x \mathcal{L}(\mathbf{x}^{(k)}, \lambda)\|_2^2 = \|\mathbf{g}^{(k)} + \mathbf{A}^{(k)} \lambda\|_2^2$$

的解,所以也称式(9.4.26)中的 $\hat{\lambda}^{(k+1)}$ 为最小二乘乘子(least-squares multipliers).

下面讨论其中的计算问题. 因为辅助问题(9.4.24)是带信赖域型约束的线性最小二乘问题,可以使用6.3节的 dog-leg 法. 该方法需要修正 Cauchy 步 \hat{s}_C 和牛顿步 s_N ,前者是目标函数(9.4.24a)沿着负梯度方向 $-\mathbf{A}^{(k)} \mathbf{c}^{(k)}$ 的极小点,后者是目标函数(9.4.24a)的无约束极小点. 因为式(9.4.24a)中的 Hessian 阵是奇异的,所以有无限多个 s_N 满足 $\mathbf{A}^{(k)T} s_N + \mathbf{c}^{(k)} = \mathbf{0}$. 令

$$s_N = -\mathbf{A}^{(k)} (\mathbf{A}^{(k)T} \mathbf{A}^{(k)})^{-1} \mathbf{c}^{(k)}$$

即所有牛顿步中约束违反度的欧氏范数最小者. 现在将目标函数(9.4.24a)在由 \hat{s}_C 和 s_N 定义的 dog-leg 轨道上的极小点选作 $v^{(k)}$. 计算问题(9.4.23)的近似解的首选方法是投影共轭梯度法,即对等式约束二次规划(式(9.4.23a)和式(9.4.23b))应用该方法,监控信赖域约束(9.4.23c)

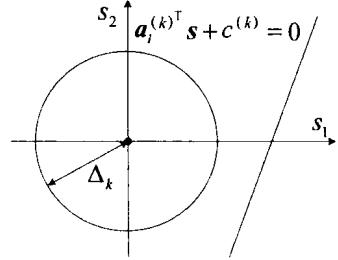


图 9.4.2 信赖域模型中的不相容约束

的满意度. 如果到达这个区域的边界或者检测到负曲率, 终止算法. 该算法需要一个可行的初始点, 可以取 $\mathbf{v}^{(k)}$.

与该方法相适应的一个价值函数是非光滑的 ℓ_2 罚函数 $\phi(\mathbf{x}, \sigma) = f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})\|_2$ (注意这里范数不用平方). 将该函数建模为

$$q^{(k)}(\mathbf{s}) = f^{(k)} + \mathbf{g}^{(k)T} \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{W}^{(k)} \mathbf{s} + \sigma m_2(\mathbf{s}) \quad (9.4.27)$$

其中 $m_2(\mathbf{s}) = \|\mathbf{c}^{(k)} + \mathbf{A}^{(k)T} \mathbf{s}\|_2$ (参见式(9.4.17)). 在计算了步 $\mathbf{s}^{(k)}$ 后, 选择罚参数 σ 充分大以使得不等式

$$q^{(k)}(\mathbf{0}) - q^{(k)}(\mathbf{s}^{(k)}) \geq \eta \sigma [m_2(\mathbf{0}) - m_2(\mathbf{s}^{(k)})] \quad (9.4.28)$$

对某参数 $\eta \in (0, 1)$ 成立.

Fletcher 建议以另一种方式利用信赖域步来避免 SQP 法中可能出现的许多难题. 这种方法不是用 Taylor 展式近似式(9.4.6b)和式(9.4.6c)来代替非线性规划问题(7.0.1)中的原始约束, 而是将它们直接代入 ℓ_1 精确罚函数(9.4.12), 得到逐段二次近似函数 $\psi^{(k)}(\mathbf{s})$ (9.4.17) 和相应的子问题

$$\begin{aligned} & \text{minimize}_{\mathbf{s} \in \mathbb{R}^n} \quad \psi^{(k)}(\mathbf{s}) \\ & \text{subject to} \quad \|\mathbf{s}\|_\infty \leq \Delta_k \end{aligned} \quad (9.4.29)$$

这个子问题和(9.4.6)的复杂度类似, 事实上, 这是 9.3 节描述的 ℓ_1 QP 问题的特例. 与子问题(9.4.6)相比, 子问题(9.4.29)没有出现约束的线性近似得到的显式约束, 从而不会出现子问题不可行的情形. 使用信赖域也保证了子问题是有界的: 应用时应该对变量重新进行比例变换, 使得应用 ℓ_∞ 范数是实际可行的. 可用无约束优化时类似的方式来调整信赖域半径. 称这种方法为逐步 ℓ_1 QP(或者 $S\ell_1$ QP 法).

$S\ell_1$ QP 法的大范围收敛性要比 SQP 法优越. 由式(9.3.8b)、式(9.3.8c)和式(9.3.15)知乘子 $\lambda^{(k+1)}$ 是有界的, 利用该性质可以证明方法大范围收敛到某个稳定点. 该方法可以求解 SQP 法不能求解的一些问题. 也可以像 SQP 法那样, 有拟牛顿法版本, 即更新 $\mathbf{W}^{(k)}$ 的近似矩阵 $\mathbf{B}^{(k)}$. 方法的渐近性能和 SQP 法是等价的, 且也具有相同的局部收敛性. 当离解很远时, 方法是有区别的, 且信赖域起着举足轻重的作用. 在问题(9.4.29)中, 因为约束函数的线性化仅出现在罚项中, 即使它们都是等式约束, 在子问题的解处, 它们也不必全为零. 可以将那些取到零的看作局部积极约束(locally active constraints). 在 SQP 法的一次迭代中, 这个集合通常是积极约束的子集. 该事实可以部分解释: 与 SQP 法相比, $S\ell_1$ QP 法不会出现约束梯度相关和乘子无界等诸多问题.

Algorithm 9.4.3 Byrd-Omojokun trust-region SQP method

- 1: Choose constants $\epsilon > 0$ and $\eta, \eta_v, \gamma \in (0, 1)$;
- 2: choose starting point $\mathbf{x}^{(0)}$, initial trust region radius $\Delta_0 > 0$;
- 3: **for** $k = 0, 1, 2, \dots$ **do**
- 4: compute $f^{(k)}, \mathbf{c}^{(k)}, \mathbf{g}^{(k)}, \mathbf{A}^{(k)}$;
- 5: compute multiplier estimates $\hat{\lambda}^{(k+1)}$ by formula(9.4.26);
- 6: **if** $\|\mathbf{g}^{(k)} - \mathbf{A}^{(k)} \hat{\lambda}^{(k+1)}\|_\infty < \epsilon$ **and** $\|\mathbf{c}^{(k)}\|_\infty < \epsilon$ **then**
- 7: stop with approximate solution $\mathbf{x}^{(k)}$.

```

8: end if
9: solve normal subproblem(9.4.24) for  $v^{(k)}$  ;
10: set  $r^{(k)} = A^{(k)^\top} v^{(k)} + c^{(k)}$  ;
11: compute  $W^{(k)}$  or a quasi-Newton approximation;
12: compute  $s^{(k)}$  by applying the projected CG method to problem(9.4.23);
13: choose  $\sigma_k$  to satisfy inequality(9.4.28);
14: compute  $\rho_k = \frac{\text{ared}_k}{\text{pred}_k} = \frac{\phi(x^{(k)}, \sigma_k) - \phi(x^{(k)} + s^{(k)}, \sigma_k)}{q^{(k)}(0) - q^{(k)}(s^{(k)})}$  ;
15: if  $\rho_k > \eta_v$  then
16:   set  $x^{(k+1)} = x^{(k)} + s^{(k)}$  ;
17:   choose  $\Delta_{k+1} \geq \Delta_k$  ;
18: else
19:   set  $x^{(k+1)} = x^{(k)}$  ;
20:   choose  $\Delta_{k+1}$  to satisfy  $\Delta_{k+1} \leq \gamma \|s^{(k)}\|$  ;
21: end if
22: end for
23: return  $x^{(k)}$  and  $\lambda^{(k+1)}$  as  $x^*$  and  $\lambda^*$  .

```

9.5 线性规划的路径跟踪算法

这一节利用障碍函数法来设计求解线性线性规划的有效算法. 考虑标准形问题

$$\begin{aligned}
 & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\
 & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b} \\
 & && \mathbf{x} \geq \mathbf{0}
 \end{aligned} \tag{9.5.1}$$

其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. 它的对偶问题为

$$\begin{aligned}
 & \text{maximize} && \mathbf{b}^\top \mathbf{y} \\
 & \text{subject to} && \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c} \\
 & && \mathbf{s} \geq \mathbf{0}
 \end{aligned} \tag{9.5.2}$$

定义原始-对偶可行域

$$\mathcal{F} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}\}$$

和严格的原始-对偶可行域

$$\mathcal{F}^\circ = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{x} > \mathbf{0}, \mathbf{s} > \mathbf{0}\}$$

在以下讨论中假定 $\text{rank}(\mathbf{A}) = m$ 且 $\mathcal{F}^\circ \neq \emptyset$.

9.5.1 障碍函数子问题和中心路径

原始问题(9.5.1)的对数障碍函数子问题为

$$\begin{aligned}
 & \text{minimize} && \phi(\mathbf{x}, \mu) := \mathbf{c}^\top \mathbf{x} - \mu \sum_{i=1}^n \log x_i \\
 & \text{subject to} && \mathbf{A}\mathbf{x} = \mathbf{b}
 \end{aligned} \tag{9.5.3}$$

下面的定理说明障碍函数子问题的解是唯一确定的.

定理 9.5.1 对每个 $\mu > 0$, 问题(9.5.3)有唯一最优解 $\mathbf{x}(\mu)$.

因为 $-\log x_i$ 是严格凸函数, 所以定理 9.5.1 中解的唯一性很容易证明, 但解的存在性证明较难, 具体细节见参考文献[44]. 子问题(9.5.3)的 KKT 条件为

$$\left. \begin{array}{l} \mathbf{A}^T \mathbf{y} + \mu \mathbf{X}^{-1} \mathbf{1} = \mathbf{c} \\ \mathbf{A} \mathbf{x} = \mathbf{b} \\ \mathbf{x} > \mathbf{0} \end{array} \right\} \quad (9.5.4)$$

其中 $\mathbf{X} = \text{diag}(x_1, x_2, \dots, x_n)$, $\mathbf{1}$ 表示分量全为 1 的向量. 注意问题(9.5.3)满足线性约束规范 (LCQ), 且目标函数是凸的, 因此式(9.5.4)是充要条件. 在式(9.5.4)中, 令 $s_i = \mu/x_i$. 由上面的讨论, 方程

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (9.5.5a)$$

$$\mathbf{A}^T \mathbf{y} + \mathbf{s} = \mathbf{c} \quad (9.5.5b)$$

$$x_i s_i = \mu, \quad i = 1, 2, \dots, n \quad (9.5.5c)$$

$$\mathbf{x} > \mathbf{0}, \mathbf{s} > \mathbf{0} \quad (9.5.5d)$$

存在唯一解. 条件(9.5.5)与原始问题(9.5.1)的 KKT 条件(即互补松弛条件)的唯一不同之处是式(9.5.5c)右边的 μ . 对所有的指标 i , 这里用逐对乘积 $x_i s_i$ 取相等的(正)值代替精确的互补条件 $x_i s_i = 0$. 中心路径在内点法中扮演重要的角色, 它是一条由严格可行点组成的弧. 记系统(9.5.5)的唯一解为 $(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{s}_\mu)$, 则中心路径(central path)可由标量 $\mu > 0$ 参数化地表示为

$$\begin{aligned} C &= \{(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{s}_\mu) : \mu > 0\} \\ &= \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^\circ : x_i s_i = \mu, i = 1, 2, \dots, n, \mu > 0\} \end{aligned} \quad (9.5.6)$$

下面的定理指出: 跟踪中心路径, 就会得到原始问题(9.5.1)的最优解. 这是求解线性规划问题的路径跟踪算法的基础. 如下定理的证明推荐读者阅读参考文献[50].

定理 9.5.2 考虑中心路径 C (即式(9.5.6)), 则

(a) 给定 $\mu_0 > 0$, 那么集合 $\{(\mathbf{x}_\mu, \mathbf{y}_\mu, \mathbf{s}_\mu) : 0 < \mu < \mu_0\}$ 是有界的.

(b) 假设 $\mathbf{b} \neq \mathbf{0}, \mathbf{c} \neq \mathbf{0}$. 令 $0 < \mu_1 < \mu_2$, 则有 $\mathbf{c}^T \mathbf{x}_{\mu_1} < \mathbf{c}^T \mathbf{x}_{\mu_2}$ 和 $\mathbf{b}^T \mathbf{y}_{\mu_1} > \mathbf{b}^T \mathbf{y}_{\mu_2}$.

(c) 当 $\mu \downarrow 0$, 序列 $\{\mathbf{x}_\mu\}$ 收敛于原始问题(9.5.1)的最优解, 并且序列 $\{(\mathbf{y}_\mu, \mathbf{s}_\mu)\}$ 收敛于对偶问题(9.5.2)的最优解.

9.5.2 用牛顿法求解障碍函数子问题

在以下内容中, 假设 $\mu > 0$ 给定, 且有严格可行点 $\bar{\mathbf{x}}$, 即满足 $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}, \bar{\mathbf{x}} > \mathbf{0}$. 应用牛顿法求解障碍函数子问题(9.5.3). 因为 $\nabla \phi(\bar{\mathbf{x}}, \mu) = \mathbf{c} - \mu \bar{\mathbf{X}}^{-1} \mathbf{1}$, $\nabla^2 \phi(\bar{\mathbf{x}}, \mu) = \mu \bar{\mathbf{X}}^{-2}$, 则牛顿法的下一个迭代点 $\mathbf{x}' = \bar{\mathbf{x}} + \mathbf{p}$, 其中 \mathbf{p} 满足 $\mu \bar{\mathbf{X}}^{-2} \mathbf{p} = -(\mathbf{c} - \mu \bar{\mathbf{X}}^{-1} \mathbf{1})$. 同时, 希望得到的下一个迭代点 \mathbf{x}' 也满足 $\mathbf{A}\mathbf{x}' = \mathbf{b}$, 为此加入约束 $\mathbf{A}\mathbf{p} = \mathbf{0}$. 这样, 求解

$$\text{minimize} \quad \frac{1}{2} \mu \mathbf{p}^T \bar{\mathbf{X}}^{-2} \mathbf{p} + (\mathbf{c} - \mu \bar{\mathbf{X}}^{-1} \mathbf{1})^T \mathbf{p}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{p} = \mathbf{0}$$

得到可行的牛顿校正 \mathbf{p} . 该问题的 KKT 条件为

$$\mu \bar{\mathbf{X}}^{-2} \mathbf{p} + \mathbf{c} - \mu \bar{\mathbf{X}}^{-1} \mathbf{1} = \mathbf{A}^T \mathbf{y} \quad (9.5.7a)$$

$$\mathbf{A}\mathbf{p} = \mathbf{0} \quad (9.5.7b)$$

对等式(9.5.7a)两边同时乘以 $\bar{\mathbf{X}}^2$, 并且注意到 $\bar{\mathbf{X}}\mathbf{1} = \bar{\mathbf{x}}$, 于是有

$$\mathbf{p} = \bar{\mathbf{x}} + \frac{1}{\mu} \bar{\mathbf{X}}^2 (\mathbf{A}^\top \mathbf{y} - \mathbf{c}) \quad (9.5.8)$$

而给等式(9.5.7a)两边同时左乘 $\mathbf{A}\bar{\mathbf{X}}^2$, 并且利用式(9.5.7b), 可以得到 $\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^\top \mathbf{y} = \mathbf{A}\bar{\mathbf{X}}^2\mathbf{c} - \mu\mathbf{A}\bar{\mathbf{X}}\mathbf{1}$. 因为 \mathbf{A} 的秩是 m , 并且 $\bar{\mathbf{X}}^2$ 是正定的, 所以 $\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^\top$ 是可逆的, 于是得到

$$\mathbf{y} = (\mathbf{A}\bar{\mathbf{X}}^2\mathbf{A}^\top)^{-1}\mathbf{A}\bar{\mathbf{X}}(\bar{\mathbf{X}}\mathbf{c} - \mu\mathbf{1}) \quad (9.5.9)$$

将此式代入式(9.5.8)即得牛顿校正 \mathbf{p} . 对于任意的 α , 线搜索牛顿迭代 $\mathbf{x}' = \bar{\mathbf{x}} + \alpha\mathbf{p}$ 满足 $\mathbf{A}\mathbf{x}' = \mathbf{b}$. 此外, 为了保证 $\mathbf{x}' > \mathbf{0}$, 令 $\alpha \in (0, \min_i \{-\bar{x}_i/p_i : p_i < 0\})$.

9.5.3 理论分析

下面来分析前面给出的一般障碍函数法的性能. 首要问题是量化给定的严格可行点 $\bar{\mathbf{x}}$ 与中心路径的接近程度. 由于这里的目的是找到 (\mathbf{y}, \mathbf{s}) 满足

$$\bar{\mathbf{X}}\mathbf{s} = \mu\mathbf{1}, \quad \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c}, \quad \mathbf{s} \geq \mathbf{0}$$

一个很自然的想法就是松弛条件 $\mathbf{s} \geq \mathbf{0}$, 通过求解如下问题得到向量 (\mathbf{y}, \mathbf{s}) :

$$\begin{aligned} & \underset{\mathbf{y}, \mathbf{s}}{\text{minimize}} \quad \frac{1}{2} \|\bar{\mathbf{X}}\mathbf{s} - \mu\mathbf{1}\|_2^2 \\ & \text{subject to} \quad \mathbf{A}^\top \mathbf{y} + \mathbf{s} = \mathbf{c} \end{aligned} \quad (9.5.10)$$

易验证由式(9.5.9)确定的 $\mathbf{y}(\bar{\mathbf{x}}, \mu)$ 和

$$\mathbf{s}(\bar{\mathbf{x}}, \mu) = \mathbf{c} - \mathbf{A}^\top \mathbf{y} \quad (9.5.11)$$

是问题(9.5.10)的解.

基于上述讨论, 定义贴近性度量(proximity measure)为

$$\delta(\mathbf{x}, \mu) = \left\| \frac{1}{\mu} \bar{\mathbf{X}}\mathbf{s}(\mathbf{x}, \mu) - \mathbf{1} \right\|_2$$

这里 $\mathbf{s}(\mathbf{x}, \mu)$ 是问题(9.5.10)的一个最优解. 如果 \mathbf{x} 在中心路径上, 那么 $\delta(\mathbf{x}, \mu) = 0$. 所以 δ 在某种程度上反映了 \mathbf{x} 与中心路径的距离. 下面将给出贴近性度量 $\delta(\mathbf{x}, \mu)$ 的一些特殊性质. 定义 $\mathbf{z} = \bar{\mathbf{X}}\mathbf{s}/\mu$, 则有

$$\delta(\mathbf{x}, \mu)^2 = \sum_{i=1}^n \left(\frac{1}{\mu} x_i s_i - 1 \right)^2 = \sum_{i=1}^n (z_i - 1)^2 \quad (9.5.12)$$

进一步, 将式(9.5.11)代入式(9.5.8), 得搜索方向 $\mathbf{p} = \bar{\mathbf{x}} - \bar{\mathbf{X}}^2\mathbf{s}(\mathbf{x}, \mu)/\mu$, 其中 $\bar{\mathbf{x}}$ 是当前迭代点. 由此得基本牛顿法的下一个迭代点

$$\mathbf{x}' = \bar{\mathbf{x}} + \mathbf{p} = 2\bar{\mathbf{x}} - \bar{\mathbf{X}}\mathbf{z} \quad (9.5.13)$$

其中 $\mathbf{z} = \frac{1}{\mu} \bar{\mathbf{X}}\mathbf{s}(\bar{\mathbf{x}}, \mu)$, 即

$$z'_i = 2\bar{x}_i - \bar{x}_i z_i = (2 - z_i)\bar{x}_i, \quad i = 1, 2, \dots, n \quad (9.5.14)$$

如果当前迭代充分接近中心路径, 则利用上面定义的贴近性度量可以证明牛顿法是二次收敛的.

定理 9.5.3 设严格可行点 $\bar{\mathbf{x}}$ 满足 $\delta(\bar{\mathbf{x}}, \mu) < 1$, 则牛顿迭代得到的新点 $\mathbf{x}' = \bar{\mathbf{x}} + \mathbf{p}$ 也是严格可行的, 并且 $\delta(\mathbf{x}', \mu) \leq \delta(\bar{\mathbf{x}}, \mu)^2$.

证明 因为 $\mathbf{A}\mathbf{p} = \mathbf{0}$, 所以有 $\mathbf{A}\mathbf{x}' = \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$. 又因为 $\delta(\bar{\mathbf{x}}, \mu) < 1$, 由式(9.5.12)可知 $|z_i - 1| < 1$, 这意味着 $0 < z_i < 2, i = 1, 2, \dots, n$. 所以, 由式(9.5.14)可以得到 $z'_i > 0, i = 1, 2, \dots, n$.

依据式(9.5.10), 贴近性度量

$$\delta(\mathbf{x}, \mu) = \min \left\{ \left\| \frac{1}{\mu} \mathbf{Xs} - \mathbf{1} \right\|_2 : \mathbf{A}^T \mathbf{y} + \mathbf{s} = \mathbf{c} \right\}$$

因此, 可以得到 $\delta(\mathbf{x}', \mu) \leq \|\mathbf{X}'\mathbf{s}(\bar{\mathbf{x}}, \mu)/\mu - \mathbf{1}\|_2$. 利用已知的 $\mathbf{s}(\bar{\mathbf{x}}, \mu) = \mu \bar{\mathbf{X}}^{-1} \bar{\mathbf{z}}$ 和式(9.5.13), 可以得到

$$\frac{1}{\mu} \mathbf{X}' \mathbf{s}(\bar{\mathbf{x}}, \mu) = \mathbf{X}' \bar{\mathbf{X}}^{-1} \bar{\mathbf{z}} = (2\bar{\mathbf{X}} - \bar{\mathbf{X}}\bar{\mathbf{Z}}) \bar{\mathbf{X}}^{-1} \bar{\mathbf{z}} = 2\bar{\mathbf{z}} - \bar{\mathbf{Z}}^2 \mathbf{1}$$

这意味着 $\delta(\mathbf{x}', \mu) \leq \|2\bar{\mathbf{z}} - \bar{\mathbf{Z}}^2 \mathbf{1} - \mathbf{1}\|_2$. 由此有

$$\delta(\mathbf{x}', \mu)^2 \leq \sum_{i=1}^n (2\bar{z}_i - \bar{z}_i^2 - 1)^2 = \sum_{i=1}^n (\bar{z}_i - 1)^4 \leq \delta(\bar{\mathbf{x}}, \mu)^4$$

从而条件 $\delta(\bar{\mathbf{x}}, \mu) < 1$ 保证基本牛顿迭代二次收敛于中心路径上的点 \mathbf{x}_μ . ■

下面的定理给出了依次递减的两个参数 μ 所对应的迭代点的贴近性度量间的关系.

定理 9.5.4 设 $\theta \in (0, 1)$, $\mu' = (1 - \theta)\mu$, 那么对任意的严格可行点 \mathbf{x} , 都有 $\delta(\mathbf{x}, \mu') \leq \frac{\delta(\mathbf{x}, \mu) + \theta\sqrt{n}}{1 - \theta}$.

证明 令 $\nu = (1 - \theta)^{-1} = \mu/\mu'$, 则 $\nu > 1$ 和 $\nu - 1 = \theta/(1 - \theta)$ 成立. 根据 $\delta(\mathbf{x}, \mu')$ 的定义, 有

$$\delta(\mathbf{x}, \mu') \leq \left\| \frac{1}{\mu} \mathbf{Xs}(\mathbf{x}, \mu) - \mathbf{1} \right\|_2 = \left\| \frac{\nu}{\mu} \mathbf{Xs}(\mathbf{x}, \mu) - \mathbf{1} \right\|_2 = \|\nu \mathbf{z}(\mathbf{x}, \mu) - \mathbf{1}\|_2$$

再由三角不等式得

$$\begin{aligned} \delta(\mathbf{x}, \mu') &\leq \|\nu \mathbf{z}(\mathbf{x}, \mu) - \mathbf{1}\|_2 + (\nu - 1)\|\mathbf{1}\|_2 \\ &\leq \nu \|\mathbf{z}(\mathbf{x}, \mu) - \mathbf{1}\|_2 + (\nu - 1)\|\mathbf{1}\|_2 \\ &= \frac{\delta(\mathbf{x}, \mu) + \theta\sqrt{n}}{1 - \theta} \end{aligned}$$

结合定理 9.5.3 和定理 9.5.4 可以得到贴近性度量的范围.

定理 9.5.5 假设对严格可行点 $\bar{\mathbf{x}}$ 有 $\delta(\bar{\mathbf{x}}, \mu) \leq 1/2$. 令 $\theta = 1/(6\sqrt{n})$, $\mu' = (1 - \theta)\mu$, 那么, 由式(9.5.13)给出的新的牛顿迭代 \mathbf{x}' 满足 $\delta(\mathbf{x}', \mu') \leq 1/2$.

证明 依次由定理 9.5.4、定理 9.5.3 和 $(1 - \theta)^{-1} \leq 6/5$ 有

$$\delta(\mathbf{x}', \mu') \leq \frac{\delta(\mathbf{x}', \mu) + \theta\sqrt{n}}{1 - \theta} \leq \frac{\delta(\bar{\mathbf{x}}, \mu)^2 + \theta\sqrt{n}}{1 - \theta} \leq \frac{5/12 + \theta\sqrt{n}}{1 - \theta} \leq \frac{1}{2}$$

因为松弛了 $\mathbf{s} \geq \mathbf{0}$, 故问题(9.5.10)的解不一定满足对偶可行性. 当贴近性度量小于 1 时, 可以证明问题(9.5.10)的解是对偶可行的; 此外, 还可以得到贴近性度量与对偶间隙之间的大小关系.

定理 9.5.6 假设 $\delta(\bar{\mathbf{x}}, \mu) < 1$, 并且向量 $(\bar{\mathbf{y}}, \bar{\mathbf{s}})$ 是问题(9.5.10)的解. 那么 $\bar{\mathbf{y}}$ 满足 $\mathbf{A}^T \bar{\mathbf{y}} \leq \mathbf{c}$, 并且有

$$\mu(n - \delta(\bar{\mathbf{x}}, \mu)\sqrt{n}) \leq \mathbf{c}^T \bar{\mathbf{x}} - \mathbf{b}^T \bar{\mathbf{y}} \leq \mu(n + \delta(\bar{\mathbf{x}}, \mu)\sqrt{n})$$

证明 因为 $\bar{\mathbf{x}} > \mathbf{0}$ 并且 $\delta(\bar{\mathbf{x}}, \mu) < 1$, 由式(9.5.12)可得 $\bar{x}_i \bar{s}_i \geq 0$, $i = 1, 2, \dots, n$. 由 $\bar{\mathbf{s}} \geq \mathbf{0}$ 可得 $\mathbf{A}^T \bar{\mathbf{y}} \leq \mathbf{c}$.

与原始-对偶可行解 $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ 对应的对偶间隙 $\mathbf{c}^T \bar{\mathbf{x}} - \mathbf{b}^T \bar{\mathbf{y}} = \bar{\mathbf{x}}^T \bar{\mathbf{s}}$. 根据 $\delta(\bar{\mathbf{x}}, \mu)$ 的定义和 Cauchy-Schwarz 不等式得

$$\delta(\bar{\mathbf{x}}, \mu)\sqrt{n} = \left\| \frac{1}{\mu} \bar{\mathbf{X}} \bar{\mathbf{s}} - \mathbf{1} \right\|_2 \cdot \|\mathbf{1}\|_2 \geq \left| \mathbf{1}^T \left(\frac{1}{\mu} \bar{\mathbf{X}} \bar{\mathbf{s}} - \mathbf{1} \right) \right| = \left| \frac{1}{\mu} \bar{\mathbf{x}}^T \bar{\mathbf{s}} - n \right|$$

进一步有

$$n - \delta(\bar{x}, \mu) \sqrt{n} \leq \frac{\bar{x}^T \bar{s}}{\mu} \leq n + \delta(\bar{x}, \mu) \sqrt{n}$$

两边同时乘以 μ 就会得到想要的结论. ■

需要注意的是,可以用定理 9.5.6 获得的界,来界定问题(9.5.1)在可行点 \bar{x} 处的目标值 $c^T \bar{x}$ 与最优值 $c^T x^*$ 的接近程度.此外,用 \bar{x} 和 \bar{y} 分别表示问题(9.5.1)和对偶问题的可行解,用 x^* 和 y^* 分别表示二者的最优解.假设 $c^T \bar{x} - b^T y = \beta$.由线性规划的强对偶定理可知 $c^T \bar{x}^* = b^T y^*$,因此

$$c^T \bar{x} - c^T x^* = \beta + b^T \bar{y} - b^T y^*$$

再根据 $c^T x^* \leq c^T \bar{x}$ 和 $b^T y^* \geq b^T \bar{y}$ 有

$$0 \leq c^T \bar{x} - c^T x^* \leq \beta$$

这样,定理 9.5.3~定理 9.5.6 自然地给出了求解线性规划(9.5.1)的路径跟踪算法 9.5.1.紧接着的定理给出算法在最坏情况下的迭代次数.

Algorithm 9.5.1 Path following algorithm for linear programming problem(9.5.1)

```

1: Find a strictly feasible primal solution  $x^{(0)}$ , an initial barrier parameter  $\mu_0 > 0$  with  $\delta(x^{(0)}, \mu_0) \leq 1/2$ , and an accuracy parameter  $t$ ;
2: set  $\theta = 1/(6\sqrt{n})$ ;
3: set  $k = 0$ ;
4: while  $n\mu_k > e^{-t}$  do
5:   set  $\mu_{k+1} = (1-\theta)\mu_k$ ;
6:   find an optimal solution  $s^{(k)} = s(x^{(k)}, \mu_k)$  to problem(9.5.10);
7:   set  $x^{(k+1)} = 2x^{(k)} - (X^{(k)})^2 s^{(k)}/\mu_k$ ;
8:   set  $k = k + 1$ ;
9: end while
10: return a solution triple  $(\bar{x}, \bar{y}, \bar{s}) \in \mathcal{F}^\circ$  satisfying  $c^T \bar{x} - b^T \bar{y} \leq 3e^{-t}/2$ .

```

定理 9.5.7 令 $t_0 = \lceil \ln(n\mu_0) \rceil$,那么算法 9.5.1 最多在 $6\sqrt{n}(t + t_0)$ 步终止,并且得到的解 $(\bar{x}, \bar{y}, \bar{s})$ 满足

$$c^T \bar{x} - b^T \bar{y} \leq \frac{3}{2} e^{-t}$$

证明 由定理 9.5.3 和定理 9.5.5 可知 $x^{(k)}$ 是严格可行的,并且 $\delta(x^{(k)}, \mu_k) \leq 1/2, k = 0, 1, \dots$ 根据终止条件和 t_0 的定义,当

$$n\mu_k = n(1-\theta)^k \mu_0 \leq (1-\theta)^k e^{t_0} \leq e^{-t}$$

即当 $-k \ln(1-\theta) > t + t_0$ 时,算法 9.5.1 终止.又因为对所有的 $\theta < 1$ 有 $-\ln(1-\theta) > \theta$ 成立,所以当 $k\theta > t + t_0$,即 $k > 6\sqrt{n}(t + t_0)$ 时算法终止.

下面令 $\bar{x} = x^{(k)}$ 为最终的迭代点,令 $\bar{y} = y(x^{(k)}, \mu_k)$.由定理 9.5.6 可知, \bar{y} 是问题(9.5.2)的可行解,且

$$c^T \bar{x} - b^T \bar{y} \leq \mu_k (n + \delta(x^{(k)}, \mu_k) \sqrt{n})$$

因为 $n\mu_k \leq e^{-t}$ 并且 $\delta(x^{(k)}, \mu_k) \leq 1/2$,所以

$$c^T \bar{x} - b^T \bar{y} \leq e^{-t} \left(1 + \frac{\delta(x^{(k)}, \mu_k)}{\sqrt{n}} \right) \leq \frac{3}{2} e^{-t}$$

9.6 评注与参考

早期罚函数的缺点是需要罚参数趋于无穷, 才能使极小化罚函数和求解原始问题等价. 乘子罚函数利用 Lagrange 乘子的显式估计, 从而不需要无穷大的罚参数. ℓ_1 精确罚函数与简单罚函数的不同之处在于: 将约束违反度的 2-范数的平方换成 1-范数. 它是精确罚函数, 即只要罚参数大于某个阈值时, 求解一个无约束优化问题即可得到原始问题的解. 但不幸的是, ℓ_1 罚函数是非光滑的, 而且一般情况下, 它在解处是不可微的, 因而一些收敛较快的利用导数的方法不能直接用来极小化 ℓ_1 罚函数.

有关内点法的研究可以追溯到 Fiacco 和 McCormick 的关于对数障碍函数的工作^[39], 他们给出了中心路径的存在性证明. McLinden 在非线性互补的背景下分析了中心路径^[42]. Dikin 提出了一种原始的仿射-尺度内点法^[43]. 直到 Karmarkar 的经典论文发表之后^[12], 研究者才开始重新审视这些工作. Karmarkar 的论文使用了投影几何中的一个灵活的思想, 但丝毫没有提及中心路径, 而后者正是内点法理论的基础. Megiddo 发现了 Karmarkar 算法与原始-对偶路径之间的联系^[44]. 这激发了关于原始-对偶方法的研究, 这些研究最终导致今天有效的软件包. Todd 给出了势函数减小法的一个很好的综述^[45], 他将上面提到的原始-对偶势函数减小法与原始势函数减小法联系起来, 包括原始的 Karmarkar 算法.

9.5 节的大部分内容都是从参考文献[49]中选取的. 从结构上看, 算法 9.5.1 跟踪问题(9.5.1)和问题(9.5.2)的中心路径, 使得我们能够得到问题(9.5.1)的解 \bar{x} , 并且 \bar{x} 的值能够任意接近最优解. 但是算法不能保证在有限次迭代给出精确的最优解. 该算法为原始内点法, 根据对偶变量的确定方式不同, 还有对偶内点法和原始-对偶内点法. 此外, 这里没有指明严格初始可行点 $\mathbf{x}^{(0)}$ 是怎样获得的, 也没有给出初始障碍参数 $\mu_0 > 0$ 的选取方法. 这些问题都是非常重要的, 读者可进一步阅读参考文献[47]、[48]和[51]. Andersen 讨论了许多与内点法的实现有关的实际问题^[52].

习题 9

9.1 考虑用罚函数(9.1.3)求解问题

$$\begin{aligned} \text{minimize} \quad & -x_1 x_2 x_3 \\ \text{subject to} \quad & 72 - x_1 - 2x_2 - 2x_3 = 0 \end{aligned}$$

验证由 $x_2 = x_3 = 24/(1 + \sqrt{(1 - 8/\sigma)})$, $x_1 = 2x_2$ 给出的 $\mathbf{x}(\sigma)$ 满足表达式 $\nabla \phi(\mathbf{x}(\sigma), \sigma) = \mathbf{0}$.

再验证当 $\sigma \rightarrow \infty$ 时有 $\mathbf{x}(\sigma) \rightarrow \mathbf{x}^*$. 确定 $\sigma = 9$ 时的 $\mathbf{x}(\sigma)$, 并验证 $\nabla^2 \phi(\mathbf{x}(9), 9)$ 是正定的.

9.2 令 $\mathbf{c}^{(k)} \rightarrow \mathbf{c}^*$, $\mathbf{A}^{(k)} \rightarrow \mathbf{A}^*$ ($\mathbf{A} \in \mathbb{R}^{n \times m}$, $n > m$), 且设 $\text{rank}(\mathbf{A}^*) = m$. 利用 Rayleigh 商的结论 ($\mathbf{u}^\top \mathbf{M} \mathbf{u} \geq \lambda_n \mathbf{u}^\top \mathbf{u}$, $\forall \mathbf{u}$) 及 \mathbf{A} 的奇异值 σ_i 的定义(矩阵 $\mathbf{A}^\top \mathbf{A}$ 的特征值 λ_i 的平方根)证明: 如果 $0 < \beta < \sigma_n$, 则对所有充分大的 k 有 $\|\mathbf{A}^{(k)} \mathbf{c}^{(k)}\|_2 \geq \beta \|\mathbf{c}^{(k)}\|_2$. 由此即有式(9.1.17). 这里用 λ_n 表示 \mathbf{M} 的最小特征值, σ_n 表示 \mathbf{A} 的最小奇异值.

9.3 针对对数障碍函数(9.1.21), 证明:

(a) 拉格朗日乘子的估计是 $\lambda_i^{(k)} = -\mu_k / c_i^{(k)}$;

- (b) x^*, λ^* 为 KKT 点；
 (c) $h^{(k)}$ 的渐近行为是 $O(\mu)$ ；
 (d) 可以得到 f^* 的 $o(\mu)$ 估计。
- 9.4 考虑用乘子罚函数 (9.2.3) 求解习题 9.1 中的问题。设所取的初始控制参数 $\lambda^{(0)} = 0$, $\sigma = 9$ 。这时极小点 $x(0, 9)$ 与习题 9.1 中已确定的极小点 $x(9)$ 相同, 用式(9.2.13)与式(9.2.15)计算出 $\lambda^{(1)}$ 。哪一个公式给出的结果作为 λ^* 的估计较好?
- 9.5 利用 Sherman-Morrison 公式(见习题 5.25)证明: 若 $\sigma > 0$, 则
- $$\begin{bmatrix} W + \sigma AA^T & A \\ A^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} W & A \\ A^T & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \sigma I \end{bmatrix}$$
- 由此, 利用式(8.1.18)可以证明
- $$(A^T W_{\sigma}^{-1} A)^{-1} = (A^T W^{-1} A)^{-1} + \sigma I$$
- 这里的记号与式(9.2.6)中的相同。设 σ_1 给定, 且 W_{σ_1} 是正定的, 证明 $(A^T W_{\sigma_1}^{-1} A)^{-1} = \sigma I + (A^T W_{\sigma_1}^{-1} A)^{-1} - \sigma_1 I = \sigma I + O(1)$, 即式(9.2.14)。
- 9.6 考虑不动点迭代
- $$\lambda^{(k+1)} = \phi(\lambda^{(k)}) := \lambda^{(k)} + M^{(k)} c^{(k)}$$
- 这类似于乘子罚函数中的式(9.2.13)与式(9.2.15)。证明
- $$\nabla \phi^T(\lambda^*) = I + (A^* W_{\sigma}^{-1} A^*) M^*^{-1}$$
- 利用习题 9.5 的结果证明 Powell-Hestenes 公式(9.2.15)是线性收敛的, 且取充分大的 σ 可以使收敛速度任意快。
- 9.7 考虑
- $$\begin{aligned} \text{minimize } \quad f(x) &= \frac{1}{2}(x_1^2 - x_2^2) - 3x_2 \\ \text{subject to } \quad x_2 &= 0 \end{aligned}$$
- (a) 计算最优解和 Lagrange 乘子。
 (b) 对 $k=0, 1, 2$ 和罚参数 $\sigma_k = 10^{k+1}$ 计算 Courant 罚函数和乘子罚函数(取 $\lambda^{(0)} = 0$)所得的点。
 (c) 针对该问题, 画出目标函数的等值线, 在其上标出这两种方法产生的迭代点, 从几何上理解这两种方法。
 (d) 假设在乘子罚函数中将罚参数取为常数 σ , 对于 σ 的哪一些值, 增广 Lagrange 函数将有一个极小点? 对于 σ 的哪一些值, 方法将是收敛的?
- 9.8 考虑极小化问题
- $$\begin{aligned} \text{minimize } \quad x \\ \text{subject to } \quad x^2 &\geq 0 \\ x + 1 &\geq 0 \end{aligned}$$
- 该问题的解 $x^* = -1$ 。写出该问题的对数障碍函数 $\phi(x, \mu)$, 并找到它的局部极小点。说明对数障碍函数的全局极小点收敛到 $x^* = -1$; 但是非全局极小点的局部极小点收敛到零, 它不是所给问题的解。
- 9.9 如果问题(9.1.1)的二阶充分条件在 x^*, λ^* 处成立, 则 Lagrange 矩阵(9.4.7)是非奇异的。提示: 令 $K^* \begin{bmatrix} s \\ t \end{bmatrix} = \mathbf{0}$ 。如果 $s = \mathbf{0}$, 则由 A^* 的秩为 m 可推出 $t = \mathbf{0}$ 。如果 $s \neq \mathbf{0}$, 则说明 s 满足 $A^T s = \mathbf{0}$ 与 $s^T W^* s = 0$ 。这与二阶充分条件矛盾。由此可以肯定 K^* 是非奇异的。

9.10 考虑约束条件 $c(\mathbf{x}) = x_1^2 + x_2^2 - 1 = 0$. 写出该约束在点 $(0,0)^T, (0,1)^T, (0.1, 0.02)^T, (-0.1, -0.02)^T$ 处的线性化约束.

9.11 用基本 SQP 法求解问题

$$\begin{aligned} & \text{minimize} && x_1 + x_2 \\ & \text{subject to} && x_2 \geq x_1^2 \end{aligned}$$

(a) 取 $\mathbf{x}^{(0)} = \mathbf{0}, \boldsymbol{\lambda}^{(0)} = 0$. 给出所得结果的一个可能的原因.

(b) 取 $\mathbf{x}^{(0)} = \mathbf{0}, \boldsymbol{\lambda}^{(0)} = 1$.

9.12 编写程序实现算法 9.4.1, 利用其求解

$$\begin{aligned} & \text{minimize} && e^{x_1 x_2 x_3 x_4 x_5} - \frac{1}{2} (x_1^3 + x_2^3 + 1)^2 \\ & \text{subject to} && x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ & && x_2 x_3 - 5x_4 x_5 = 0 \\ & && x_1^3 + x_2^3 + 1 = 0 \end{aligned}$$

取初始点 $\mathbf{x}^{(0)} = (-1.71, 1.59, 1.82, -0.763, -0.763)^T$, 这里解 $\mathbf{x}^* = (-1.8, 1.7, 1.9, -0.8, -0.8)^T$.

9.13 假定 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 满足问题 (9.1.1) 的二阶充分条件. 令 $(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) \rightarrow (\mathbf{x}^*, \boldsymbol{\lambda}^*)$, 且对所有 k , 设存在向量 $\mathbf{s}^{(k)}$ ($\|\mathbf{s}^{(k)}\|_2 = 1$) 使得 $\mathbf{A}^{(k)T} \mathbf{s}^{(k)} = \mathbf{0}$ 但是 $\mathbf{s}^{(k)T} \mathbf{W}^{(k)} \mathbf{s}^{(k)} \leq 0$. 利用连续性证明这与二阶充分条件矛盾. 因此对充分接近于 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 的 $(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})$, 子问题 (9.4.5) 是适当的.

也可以用如下方法证明之. 关于点 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 的二阶充分性条件等价于 $\mathbf{Z}^{*T} \mathbf{W}^* \mathbf{Z}^*$ 正定 (见习题 8.12). 由式 (8.1.13) (直接消元) 定义 \mathbf{Z}^* , 不失一般性, 假定 \mathbf{A}_1^* 是非奇异的, 那么在 \mathbf{x}^* 的某个邻域内 \mathbf{Z} 是关于 \mathbf{A} 的连续函数. 同样对于充分靠近 $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ 的 $(\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})$, 由特征值的连续性知既约矩阵 $\mathbf{Z}^{(k)T} \mathbf{W}^{(k)} \mathbf{Z}^{(k)}$ 也是正定的. 因此, 子问题 (9.4.5) 的二阶充分条件成立.

9.14 写出下面有等式和不等式约束的凸二次规划问题

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{d}^T \mathbf{x} \\ & \text{subject to} && \mathbf{A}^T \mathbf{x} \geq \mathbf{b}, \quad \bar{\mathbf{A}}^T \mathbf{x} = \bar{\mathbf{b}} \end{aligned}$$

的 KKT 条件, 其中 \mathbf{G} 是对称和半正定的. 利用这些条件来得到该问题的原始-对偶步.

附录 A 基础知识

A. 1 集合

集合可以用列举法表示,即在大括号内具体地列举出它的元素,比如 $S=\{1, 2, 3, 4\}$. 另外,也可以用描述法表示集合,即 $S=\{x; P(x)\}$,其中 $P(x)$ 是集合的每一个元素要满足的性质,例如上面的集合也可以用描述法表示为 $S=\{x; 1 \leq x \leq 4, x \text{ 是整数}\}$. 如果 x 是集合 S 的元素,记为 $x \in S$,如果 y 不是 S 的元素,记为 $y \notin S$. 由所有不在 S 中的元素组成的集合称为 S 的补集,记为 \bar{S} . 集合 S 和 T 的并(union)记为 $S \cup T$,它是由属于 S 的元素或属于 T 的元素组成的集合. 集合 S 和 T 的交(intersection)记为 $S \cap T$,它是由既属于 S 又属于 T 的元素组成的集合. 如果 S 是 T 的子集,即 S 的每一个元素也是 T 的元素,记为 $S \subseteq T$ 或 $T \supseteq S$.

一个函数在某集合上的极小化有两种表示方式,即用 $\min_{x \in S} f(x)$ 或者 $\min\{f(x); x \in S\}$ 来记 f 在集合 S 上的最小值.

设 a 和 b 为实数, $[a, b]$ 表示满足 $a \leq x \leq b$ 的实数组成的集合. 在上述表示中,如果用圆括号代替方括号,则在定义中表示严格不等式成立,比如 $(a, b]$ 表示所有满足 $a < x \leq b$ 的 x .

如果 S 是有上界的实数集,则存在一个最小的实数 y 使得对所有 $x \in S$ 有 $x \leq y$ 成立,称 y 为集合 S 的最小上界(least upper bound)或上确界(supremum),记为 $\sup_{x \in S} x$ 或者 $\sup\{x; x \in S\}$. 类似地,集合 S 的最大下界(greatest lower bound)或下确界(infimum),记为 $\inf_{x \in S} x$ 或者 $\inf\{x; x \in S\}$.

A. 2 矩阵

矩阵是由称为元素(elements)的数排成的矩形阵列,一般用大写字母表示. 当不表示具体的数字时,用斜体的带双下标的小写字母表示矩阵的元素. 这样, m 行 n 列的矩阵可以写为

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

称这样的矩阵是 $m \times n$ 矩阵. 如果希望通过定义一般的元素来确定矩阵,使用符号 $\mathbf{A} = [a_{ij}]$.

如果 $m \times n$ 矩阵的所有元素全为零,则称其为零矩阵(zero matrix),记为 $\mathbf{0}$. 称元素满足 $i \neq j$ 时 $a_{ij} = 0$, $i = 1, 2, \dots, n$ 时 $a_{ii} = 1$ 的方阵($m = n$)为单位矩阵(identity matrix),记为 \mathbf{I} .

两个 $m \times n$ 矩阵 \mathbf{A} 和 \mathbf{B} 的和(sum)写为 $\mathbf{A} + \mathbf{B}$,它是一个 $m \times n$ 矩阵,其元素为 \mathbf{A} 和 \mathbf{B} 对

应元素的和. 矩阵 A 和标量 λ 的乘积 (product) 写为 λA 或 $A\lambda$, 它由 A 的每一个元素乘以 λ 得到. $m \times n$ 矩阵 A 和 $n \times p$ 矩阵 B 的乘积 (product) 记为 AB , 其是一个 $m \times p$ 矩阵 C , 元素 $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$.

$m \times n$ 矩阵 A 的转置 (transpose) 是 $n \times m$ 矩阵 A^T , 其元素 $a_{ij}^T = a_{ji}$. 如果 $A^T = A$, 则称 (方) 矩阵 A 是对称的 (symmetric). 如果存在一个称为 A 的逆 (inverse) 矩阵 A^{-1} 使得 $A^{-1}A = I = AA^{-1}$, 则方阵 A 是非奇异的 (nonsingular). 方阵 A 的行列式 (determinant) 记为 $\det(A)$. 一个矩阵的行列式非零当且仅当矩阵非奇异. $\det(AB) = \det(A)\det(B)$; $\det(A^{-1}) = 1/\det(A)$. $n \times n$ 矩阵的迹 (trace) 定义为 $\text{trace}(A) = \sum_{i=1}^n a_{ii}$. 如果存在非奇异矩阵 S 使得 $B = S^{-1}AS$, 则两个 $n \times n$ 的方阵 A 和 B 是相似的 (similar).

称具有一行的矩阵为行向量 (row vector); 称具有一列的矩阵为列向量 (column vector). 通常, 任一类型的向量都用小写字母表示. 为了进一步区别行向量与列向量, 如果 a 是有 n 个元素的列向量, 记 $a \in \mathbb{R}^n$. 如果 a 是有 n 个元素的列向量, b 是有 m 个元素的列向量, 记作 $(a, b) \in \mathbb{R}^n \times \mathbb{R}^m$. 本书中若无特别声明, 则统一作如上默认.

将一个矩阵剖分为子矩阵经常会带来很大的方便, 这可由矩阵的剖分线来剖分. 比如

$$A = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \end{array} \right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

剖分所得到的子矩阵如上所示, 一般记为 A_{ij} .

一个矩阵可被剖分成行向量或列向量, 在这种情况下, 用特殊的记号会很方便. 记 $m \times n$ 矩阵 A 的列为 a_j , $j = 1, 2, \dots, n$, 则 $A = [a_1, a_2, \dots, a_n]$. 类似地, 记 A 的行为 a^i , $i = 1, 2, \dots, m$, 则 $A^T = [a^{1T}, a^{2T}, \dots, a^{mT}]^T$. 基于同样的模式, 经常记矩阵 $A := [B|C]$ 的剖分为 $A = [B C]$.

A. 3 空 间

考虑由 n 维实向量作为元素组成的向量空间, 其是 n 维实欧氏空间, 记为 \mathbb{R}^n . 该空间中向量的“加”及与标量的“乘”运算均是通过对分量进行相应运算实现. $x \geq 0$ 表示 x 的每一个分量都是非负的.

连接两个向量 x 和 y 的线段 (line segment) 记为 $[x, y]$, 其由所有形如 $\alpha x + (1 - \alpha)y$, $0 \leq \alpha \leq 1$ 的向量组成.

两个向量 $x = (x_1, x_2, \dots, x_n)^T$ 和 $y = (y_1, y_2, \dots, y_n)^T$ 的标量积 (scalar product) 定义为 $x^T y = y^T x = \sum_{i=1}^n x_i y_i$. 一个向量 x 的 2-范数 (norm) 是 $\|x\|_2 = (x^T x)^{1/2}$, 通常省略脚标 2. 对 \mathbb{R}^n 中的任意两个向量 x 和 y , Cauchy-Schwarz 不等式 $|x^T y| \leq \|x\| \cdot \|y\|$ 成立. 基于该事实, 定义两个非零向量 x, y 的夹角为

$$\arccos \frac{x^T y}{\|x\|_2 \|y\|_2}$$

如果 $\mathbf{x}^T \mathbf{y} = 0$, 称向量 \mathbf{x} 和 \mathbf{y} 正交(orthogonal).

如果存在不全为零的标量 $\lambda_1, \lambda_2, \dots, \lambda_k$ 使得 $\sum_{i=1}^k \lambda_i \mathbf{a}_i = \mathbf{0}$, 则称向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ 是线性相关的(linearly dependent). 如果不存在满足条件的标量, 则称向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ 线性无关(linearly independent). 向量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ 的线性组合(linear combination)指形如 $\sum_{i=1}^k \lambda_i \mathbf{a}_i$ 的向量. 向量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ 的所有线性组合所成之集称为由其生成的集合(span). 生成 \mathbb{R}^n 的线性无关向量集称为 \mathbb{R}^n 的基(basis). \mathbb{R}^n 的每个基正好含有 n 个向量.

一个向量组的一个部分组称为一个极大线性无关组, 如果这个部分组本身是线性相关的, 并且从这个向量组中任意添一个向量(如果还有的话), 所得的部分向量组都线性相关. 向量组的极大线性无关组所含向量的个数称为这个向量组的秩.

矩阵 \mathbf{A} 的秩(rank)等于 \mathbf{A} 的行向量组的秩, 它也等于 \mathbf{A} 的列向量组的秩. 如果 $m \times n$ 矩阵 \mathbf{A} 的秩等于 m 与 n 中较小者, 则称 \mathbf{A} 满秩(full rank). 矩阵 \mathbf{A} 的范数定义为 $\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\|$. 此外性质

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\| \quad (\text{A. 3. 1})$$

对所有维数相容的矩阵都成立. 非奇异矩阵的条件数(condition number)定义为

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \quad (\text{A. 3. 2})$$

矩阵 \mathbf{A} 的 Frobenius 范数定义为

$$\|\mathbf{A}\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2} \quad (\text{A. 3. 3})$$

它在很多场合很有用.

对于向量 $\mathbf{x} \in \mathbb{R}^n$, 除前面定义的欧氏范数(Euclidean norm) $\|\cdot\|_2$ 之外, 还有下面的两种常用范数:

$$\|\mathbf{x}\|_1 := \sum_{i=1}^n |x_i|, \quad \|\mathbf{x}\|_\infty := \max_{i=1,2,\dots,n} |x_i|$$

所有这些范数在某种意义上度量了向量的长度, 且满足

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n} \|\mathbf{x}\|_\infty, \quad \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq n \|\mathbf{x}\|_\infty \quad (\text{A. 3. 4})$$

类似地, 从这些向量范数的定义可以得到相应的矩阵范数

$$\|\mathbf{A}\| := \sup_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\| \quad (\text{A. 3. 5})$$

这些范数的显式公式为

$$\|\mathbf{A}\|_1 := \max_{j \in \{1,2,\dots,n\}} \sum_{i=1}^m |a_{ij}|, \quad \|\mathbf{A}\|_\infty := \max_{i \in \{1,2,\dots,m\}} \sum_{j=1}^n |a_{ij}|$$

\mathbb{R}^n 的子空间(subspace) M 是其关于向量加运算和标量乘运算封闭的子集. 即如果 \mathbf{a} 和 \mathbf{b} 是 M 中的向量, 则对任意的标量 λ 和 μ , $\lambda\mathbf{a} + \mu\mathbf{b}$ 属于 M . 例如, 如果 M 是(i)整个空间 \mathbb{R}^2 , (ii)任一过原点的直线, (iii)仅有原点, (iv)空集, 则它是 \mathbb{R}^2 的子空间. 子空间 M 的维数等于 M 中极大线性无关组所含向量的个数. 如果 M 是 \mathbb{R}^n 的子空间, M 的正交补(orthogonal complement)记为 M^\perp , 则 M^\perp 由所有与 M 中每一向量都正交的向量组成. 易见 M 的正交补是一个子空间, 且 M 和 M^\perp 在如下意义下可以生成 \mathbb{R}^n : 每一向量 $\mathbf{x} \in \mathbb{R}^n$ 能被唯一地分解成 $\mathbf{x} = \mathbf{a} + \mathbf{b}$,

其中 $a \in M, b \in M^\perp$. 假如这样, 则 a 和 b 分别称为 x 在子空间 M 和 M^\perp 上的正交投影(orthogonal projections), 记为 $M \oplus M^\perp = \mathbb{R}^n$.

给定向量 $a_i \in \mathbb{R}^n, i=1, 2, \dots, m$, 集合

$$M = \{w \in \mathbb{R}^n \mid a_i^T w = 0, i = 1, 2, \dots, m\} \quad (\text{A. 3.6})$$

是子空间. 然而, 集合

$$\{w \in \mathbb{R}^n \mid a_i^T w \geq 0, i = 1, 2, \dots, m\} \quad (\text{A. 3.7})$$

一般不是子空间. 例如, 当 $n=2, m=1$ 和 $a_1 = (1, 0)^T$ 时, 该集合由所有具有 $w_1 \geq 0$ 的向量组成. 但是给定该集合中的两个向量 $x = (1, 0)^T$ 和 $y = (2, 3)^T$, 易于选择倍数 α 和 β 使得 $\alpha x + \beta y$ 的第一个分量为负, 因此位于该集合之外. 形如式(A. 3.6)和式(A. 3.7)的集合出现在约束优化的二阶最优化条件的讨论中, 详见 7.4 节.

如果 A 是任一 $m \times n$ 矩阵, 则零空间(null space)是子空间

$$\text{Null}(A) = \{w \in \mathbb{R}^n \mid Aw = 0\}$$

而值空间(range space)是

$$\text{Range}(A) = \{w \in \mathbb{R}^m \mid w = Av, v \in \mathbb{R}^n\}$$

线性代数基本定理(fundamental theorem of linear algebra)表明

$$\text{Null}(A) \oplus \text{Range}(A^T) = \mathbb{R}^n$$

将空间 X 中的每一个点与空间 Y 中的一点联系起来的对应法则 A 称为从 X 到 Y 的映射或者变换. 为方便起见, 用符号 $A: X \rightarrow Y$ 表示. 变换 A 可以是线性的或非线性的. 线性变换 A 的范数定义为 $\|A\| = \max_{\|x\|=1} \|Ax\|$. 对任何 x , 有 $\|Ax\| \leq \|A\| \|x\|$.

A. 4 特征值与二次型

对于 $n \times n$ 矩阵 A , 称满足等式 $Ax = \lambda x$ 的标量 λ 和非零向量 x 分别为 A 的特征值(eigenvalue)和特征向量(eigenvector). 显然, λ 为特征值的充分必要条件是 $A - \lambda I$ 奇异, 即 $\det(A - \lambda I) = 0$. 将最后一个等式展开即得到 n 阶多项式方程, 求解其可以得到 n 个复根(可能有重根) λ , 它们是 A 的特征值. 本节以下部分假设 A 是对称矩阵, 下面是关于特征值和特征向量的基本性质:

(a) A 的特征值都是实的.

(b) 与不同特征值所对应的特征向量是正交的.

(c) 存在 \mathbb{R}^n 的正交基, 它的每一个元素是 A 的特征向量.

在性质(c)中, 如果基 u_1, u_2, \dots, u_n 中的每个向量的长度都是 1, 定义矩阵 $Q = [u_1, u_2, \dots, u_n]$, 则有 $Q^T Q = I$, 因此 $Q^T = Q^{-1}$. 称具有此性质的矩阵为正交矩阵(orthogonal matrix). 易见还有下述等式成立:

$$Q^{-1} A Q = Q^T A Q = Q^T [A u_1, A u_2, \dots, A u_n] = Q^T [\lambda_1 u_1, \lambda_2 u_2, \dots, \lambda_n u_n]$$

这样

$$Q^{-1} A Q = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

因此, A 相似于对角矩阵.

如果对所有非零向量 x , 二次型(quadratic form) $x^T A x$ 是正(负)的, 则称矩阵 A 是正(负)定的(positive (negative) definite). 类似地, 如果对所有 x , 有 $x^T A x \geq 0 (\leq 0)$, 则称矩阵 A 是半正(负)定的(positive (negative) semidefinite).

对于对称矩阵, 容易得到正定(或半正定)性和 A 的特征值之间的联系. 对任一 x , 令 $y = Q^{-1}x$, 其中 Q 如上定义, 则 $x^T A x = y^T Q^T A Q y = \sum_{i=1}^n \lambda_i y_i^2$. 因为 y_i 是任意的(x 是任意的), 显然, A 正定(或半正定)当且仅当 A 的所有特征值是正的(或非负的).

通过对角化, 容易证明对称半正定矩阵 A 有半正定(对称的)的平方根 $A^{1/2}$ 满足 $A^{1/2} \cdot A^{1/2} = A$. 为此, 利用 Q 定义

$$A^{1/2} = Q \begin{bmatrix} \lambda_1^{1/2} & & & \\ & \lambda_2^{1/2} & & \\ & & \ddots & \\ & & & \lambda_n^{1/2} \end{bmatrix} Q^T$$

易验证该矩阵是 A 的平方根.

此外, 针对 $n \times n$ 矩阵 A 可以证明

$$\text{trace}(A) = \sum_{i=1}^n \lambda_i, \quad \det(A) = \prod_{i=1}^n \lambda_i$$

即矩阵的迹是它的特征值之和, 矩阵的行列式是它的特征值的乘积. 对于任一 $m \times n$ 矩阵 A 还有

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

其中 $\lambda_{\max}(\cdot)$ 表示最大特征值.

A.5 拓扑概念

向量序列 $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ 记为 $\{x^{(k)}\}_{k=0}^{\infty}$, 或者简记为 $\{x^{(k)}\}$. 当 $k \rightarrow \infty$ 时, 如果有 $\|x^{(k)} - x\| \rightarrow 0$ (即给定 $\epsilon > 0$, 存在 N 使得 $k \geq N$ 蕴含着 $\|x^{(k)} - x\| < \epsilon$), 则称其收敛(convergence)到极限 x . 如果 $\{x^{(k)}\}$ 收敛到 x , 记作 $x^{(k)} \rightarrow x$ 或 $\lim_{k \rightarrow \infty} x^{(k)} = x$.

如果存在 $\{x^{(k)}\}$ 的子序列收敛到 x , 则称点 x 是序列 $\{x^{(k)}\}$ 的极限/聚点(limit/accumulation point). 这样, 如果存在正整数的子集 \mathcal{H} 使得 $\{x^{(k)}\}_{k \in \mathcal{H}}$ 收敛到 x , 则 x 是 $\{x^{(k)}\}$ 的极限点.

以 x 为中心的球(sphere around x)指形如 $\{y: \|y - x\| < \delta\}$ 的集合, 其中 $\delta > 0$. 也称这样的球为 x 的半径为 δ 的邻域, 记为 $N(x, \delta)$. 当我们不强调邻域的半径时, 简写为 N_x .

设 S 是 \mathbb{R}^n 的子集. 若以点 $x \in S$ 为中心的某个球包含在 S 中, 则称 x 是 S 的内点. S 的内点组成 S 的内部(interior). 集合 $\{x: \|x\| \leq 1\}$ 的内部是球 $\{x: \|x\| < 1\}$. 如果对 S 中的每个点, 存在以其为中心的球包含在 S 中, 则称 S 是开的(open). 等价地, 如果给定 $x \in S$, 存在 $\delta > 0$ 使得 $\|y - x\| < \delta$ 蕴含着 $y \in S$, 则 S 是开的. 这样, 球 $\{x: \|x\| < 1\}$ 是开的. 集合的内部总是开的, 且它是包含在 S 中的最大开集.

如果与集合 P 可以任意接近的点都属于 P , 则称集合 P 是闭的(closed). 等价地, 如果 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ 且 $\mathbf{x}^{(k)} \in P$ 蕴含着 $\mathbf{x} \in P$, 则 P 是闭的. 这样, 球 $\{\mathbf{x}: \|\mathbf{x}\| \leq 1\}$ 是闭的. \mathbb{R}^n 的子集 P 的闭包(closure)是包含在 P 中的最小闭集. 集合的边界(boundary)是它的闭包中不属于内部的部分.

如果集合既是闭的, 又是有界的(即其是闭的, 且包含在某一半径有限的球内), 则称它是紧的(compact). 一个重要的结果(归功于 Weierstrass)是: 如果 S 是紧集且序列 $\{\mathbf{x}^{(k)}\}$ 的每一元素属于 S , 则 $\{\mathbf{x}^{(k)}\}$ 有一个属于 S 的聚点(即存在子序列收敛到 S 中的点).

与有界实数序列 $\{r_k\}_{k=0}^{\infty}$ 相对应, 如果我们令 $s_k = \sup\{r_i: i \geq k\}$, 则 $\{s_k\}$ 收敛到某一实数 s_0 . 该实数称为 $\{r_k\}$ 的上极限(limit superior), 记为 $\limsup_{k \rightarrow \infty} r_k$ 或 $\overline{\lim}_{k \rightarrow \infty} r_k$. 类似地, 用 $\liminf_{k \rightarrow \infty} r_k$ 或 $\underline{\lim}_{k \rightarrow \infty} r_k$ 表示 $\{r_k\}$ 的下极限(limit inferior).

A.6 函数

连续

设 f 是定义在 \mathbb{R}^n 的子集上的实值函数, 如果 $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ 蕴含着 $f(\mathbf{x}^{(k)}) \rightarrow f(\mathbf{x})$, 称 f 在 \mathbf{x} 处连续(continuous). 等价地, 如果给定 $\epsilon > 0$, 则存在 $\delta > 0$ 使得 $\|\mathbf{y} - \mathbf{x}\| < \delta$ 蕴含着 $|f(\mathbf{y}) - f(\mathbf{x})| < \epsilon$. 与连续函数相关的一个重要结果是 Weierstrass 定理(theorem of Weierstrass): 定义在紧集 S 上的函数有极小点, 即存在 $\mathbf{x}^* \in S$ 使得对所有 $\mathbf{x} \in S$ 有 $f(\mathbf{x}) \geq f(\mathbf{x}^*)$.

可以把 \mathbb{R}^n 上的实值函数集 f_1, f_2, \dots, f_m 看作一个向量值函数

$$\mathbf{f} = (f_1, f_2, \dots, f_m)^T$$

该函数将向量 $\mathbf{x} \in \mathbb{R}^n$ 映射到 \mathbb{R}^m 中的向量 $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T$. 如果每个分量函数在 \mathbb{R}^n 的某开集上是连续的, 则称 f 是连续的. 如果存在常数 $L > 0$ 使得在任意两点 \mathbf{x}, \mathbf{y} 满足

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$$

则称函数 $f(\mathbf{x})$ 是 Lipschitz 连续的. 进一步, 如果每个分量函数在该集合上有连续的一阶偏导数, 则记作 $f \in C^1$. 一般地, 如果分量函数有连续的 p 阶偏导数, 则记作 $f \in C^p$.

一阶导数和二阶导数

如果 $f \in C^1$ 是 \mathbb{R}^n 上的实值函数, 定义 f 在 \mathbf{x} 的梯度(gradient)为列向量

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right)^T$$

也用记号 $\mathbf{g}(\mathbf{x})$ 表示梯度. 如果 $f \in C^2$, 定义 f 在 \mathbf{x} 的 Hessian 阵为 $n \times n$ 矩阵 $\left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right]$, 记为 $\nabla^2 f(\mathbf{x})$ 或 $\mathbf{G}(\mathbf{x})$. 因为 $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$, 易见 Hessian 阵是对称的.

对于向量值函数 $\mathbf{f} = (f_1, f_2, \dots, f_m)^T$, 情形类似. 如果 $f \in C^1$, 一阶导数定义为 $m \times n$ 矩阵 $\nabla \mathbf{f}(\mathbf{x}) = \left[\frac{\partial f_i(\mathbf{x})}{\partial x_j} \right]$, 则称其为 f 的 Jacobi 矩阵. 如果 $f \in C^2$, 相应于 m 个分量函数可以定义 m 个 Hessian 阵 $\mathbf{G}_1(\mathbf{x}), \mathbf{G}_2(\mathbf{x}), \dots, \mathbf{G}_m(\mathbf{x})$. 对于向量值函数而言, 二阶导数自身是一个三阶的张量(tensor). 任给 $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T \in \mathbb{R}^m$, 实值函数 $\lambda^T \mathbf{f}$ 的梯度为 $\lambda^T \nabla \mathbf{f}(\mathbf{x})$, Hessian 阵

(记作 $\lambda^T \mathbf{G}(\mathbf{x})$) 为

$$\lambda^T \mathbf{G}(\mathbf{x}) = \sum_{i=1}^m \lambda_i \mathbf{G}_i(\mathbf{x})$$

方向导数

设 f 是连续可微的, 且 $\mathbf{0} \neq \mathbf{p} \in \mathbb{R}^n$, 则 f 沿方向 \mathbf{p} 的方向导数 (directional derivative) 为

$$f_{\mathbf{p}}(\mathbf{x}) := \lim_{\alpha \downarrow 0} \frac{f(\mathbf{x} + \alpha \mathbf{p}) - f(\mathbf{x})}{\alpha} = \nabla f(\mathbf{x})^T \mathbf{p} \quad (\text{A. 6. 1})$$

为了验证该公式, 定义函数 $\phi(\alpha) = f(\mathbf{x} + \alpha \mathbf{p}) = f(\mathbf{y}(\alpha))$, 其中 $\mathbf{y}(\alpha) = \mathbf{x} + \alpha \mathbf{p}$. 注意, $\lim_{\alpha \downarrow 0} \frac{f(\mathbf{x} + \alpha \mathbf{p}) - f(\mathbf{x})}{\alpha} = \lim_{\alpha \downarrow 0} \frac{\phi(\alpha) - \phi(0)}{\alpha} = \phi'(0)$. 通过对 $f(\mathbf{y}(\alpha))$ 运用链式法则得到

$$\begin{aligned} \phi'(\alpha) &= \sum_{i=1}^n \frac{\partial f(\mathbf{y}(\alpha))}{\partial y_i} \frac{dy_i}{d\alpha} \\ &= \sum_{i=1}^n \frac{\partial f(\mathbf{y}(\alpha))}{\partial y_i} \mathbf{p}_i \\ &= \nabla f(\mathbf{y}(\alpha))^T \mathbf{p} = \nabla f(\mathbf{x} + \alpha \mathbf{p})^T \mathbf{p} \end{aligned}$$

通过置 $\alpha = 0$, 并比较最后两个表达式可以得到式 (A. 6. 1).

即使函数 f 本身不可微, 有时也可以定义方向导数. 比如 $f(\mathbf{x}) = \|\mathbf{x}\|_1$, 易见只要 \mathbf{x} 有一个零分量, 它的一阶导数就不存在. 但由定义 (A. 6. 1) 有

$$f_{\mathbf{p}}(\mathbf{x}) = \lim_{\alpha \downarrow 0} \frac{\|\mathbf{x} + \alpha \mathbf{p}\|_1 - \|\mathbf{x}\|_1}{\alpha} = \lim_{\alpha \downarrow 0} \frac{\sum_{i=1}^n |x_i + \alpha p_i| - \sum_{i=1}^n |x_i|}{\alpha}$$

如果 $x_i > 0$, 则对所有充分小的 $\alpha > 0$, 有 $|x_i + \alpha p_i| = x_i + \alpha p_i$; 如果 $x_i < 0$, 则有 $|x_i + \alpha p_i| = -x_i - \alpha p_i$; 如果 $x_i = 0$, 则有 $|x_i + \alpha p_i| = \alpha |p_i|$. 因此, 有

$$f_{\mathbf{p}}(\mathbf{x}) = \sum_{i: x_i \neq 0} \text{sign}(x_i) p_i + \sum_{i: x_i = 0} |p_i|$$

因此, 该函数的方向导数对任意的 \mathbf{x} 和 \mathbf{p} 都存在.

中值定理

进行理论分析时经常用到的一组结论是 **Taylor 定理** 或 **中值定理** (mean value theorem): 如果在包含线段 $[\mathbf{x}, \mathbf{y}]$ 的区域上有 $f \in C^1$, 则存在 $\theta \in (0, 1)$ 使得

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x} + \theta(\mathbf{y} - \mathbf{x}))^T (\mathbf{y} - \mathbf{x}) \quad (\text{A. 6. 2})$$

进一步, 如果 $f \in C^2$, 则存在 $\theta \in (0, 1)$ 使得

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{1}{2} (\mathbf{y} - \mathbf{x})^T \mathbf{G}(\mathbf{x} + \theta(\mathbf{y} - \mathbf{x})) (\mathbf{y} - \mathbf{x}) \quad (\text{A. 6. 3})$$

这里 \mathbf{G} 表示 f 的 Hessian 阵.

例 A. 6. 1 (中值定理) 考虑 $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(\mathbf{x}) = x_1^3 + 3x_1 x_2^2$, 并设 $\mathbf{x} = (0, 0)^T$ 和 $\mathbf{y} = (1, 2)^T$. 易于验证 $f(\mathbf{x}) = 0$ 和 $f(\mathbf{y}) = 13$. 因为

$$\nabla f(\mathbf{x} + \theta(\mathbf{y} - \mathbf{x})) = \begin{bmatrix} 3(x_1 + \theta)^2 + 3(x_2 + 2\theta)^2 \\ 6(x_1 + \theta)(x_2 + 2\theta) \end{bmatrix} = \begin{bmatrix} 15\theta^2 \\ 12\theta^2 \end{bmatrix}$$

有 $\nabla f(\mathbf{x} + \theta(\mathbf{y} - \mathbf{x}))^T (\mathbf{y} - \mathbf{x}) = 39\theta^2$. 因此, 当置 $\theta = 1/\sqrt{3}$ 时, 式 (A. 6. 2) 成立, 且像所断言的那

样, θ 属于开区间 $(0, 1)$.

隐函数定理

考虑含有 n 个变量的 m 个方程

$$h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m$$

隐函数定理 (implicit function theorem) 希望解决的问题是: 如果固定 $n-m$ 个变量, 则方程组关于剩下的 m 个变量是否有解. 这样, 选 m 个变量, 比如说 x_1, x_2, \dots, x_m , 希望来确定它们是否可以由剩余变量以如下形式

$$x_i = \phi_i(x_{m+1}, x_{m+2}, \dots, x_n), \quad i = 1, 2, \dots, m$$

来表示. 如果存在函数 ϕ_i , 则称它是由所给方程组确定的隐函数 (implicit functions).

定理 A.6.1 (隐函数定理) 设 $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^\top$ 为 \mathbb{R}^n 中的点, 且满足

- (i) 在 $\mathbf{x}^{(0)}$ 的某个邻域内, 函数 $h_i \in C^1, i = 1, 2, \dots, m$;
- (ii) $h_i(\mathbf{x}^{(0)}) = 0, i = 1, 2, \dots, m$;
- (iii) $m \times n$ 阶 Jacobi 矩阵

$$J(\mathbf{x}_i) = \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_1(\mathbf{x})}{\partial x_m} \\ \vdots & \cdots & \vdots \\ \frac{\partial h_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial h_m(\mathbf{x})}{\partial x_m} \end{bmatrix}_{\mathbf{x} = \mathbf{x}^{(0)}}$$

是非奇异的.

则存在 $\hat{\mathbf{x}}^{(0)} = (x_{m+1}^{(0)}, x_{m+2}^{(0)}, \dots, x_n^{(0)})^\top \in \mathbb{R}^{n-m}$ 的邻域 $N_{\hat{\mathbf{x}}^{(0)}}$ 及定义在该邻域上的函数 $\phi_i(\hat{\mathbf{x}}), i = 1, 2, \dots, m$ 满足

- (a) $\phi_i \in C^1$;
- (b) $x_i^{(0)} = \phi_i(\hat{\mathbf{x}}^{(0)}), i = 1, 2, \dots, m$;
- (c) $h_i(\phi_1(\hat{\mathbf{x}}), \phi_2(\hat{\mathbf{x}}), \dots, \phi_m(\hat{\mathbf{x}}), \hat{\mathbf{x}}) = 0, i = 1, 2, \dots, m, \forall \hat{\mathbf{x}} \in N_{\hat{\mathbf{x}}^{(0)}}$.

其中 $\hat{\mathbf{x}} = (x_{m+1}, x_{m+2}, \dots, x_n)^\top$.

例 A.6.2 考虑方程 $x_1^2 + x_2 = 0$. $x_1 = 0, x_2 = 0$ 是方程的一个解. 然而, 在这个解的任何邻域上都不存在函数 ϕ 使得 $x_1 = \phi(x_2)$. 因为在该解处, $J(\mathbf{x}^{(0)}) = 2x_1 \Big|_{x_1=0} = 0$, 隐函数定理中的条件 (iii) 不满足. 然而在任何其他解处, 均存在这样的 ϕ .

例 A.6.3 设 \mathbf{A} 是 $m \times n$ 矩阵 ($m < n$), 考虑线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$. 如果 \mathbf{A} 被剖分为 $\mathbf{A} = [\mathbf{B} \ \mathbf{C}]$, 其中 \mathbf{B} 是 $m \times m$ 阶的, 则条件 (iii) 满足当且仅当 \mathbf{B} 是非奇异的. 当然, 该条件与线性方程组的理论是一致的. 鉴于该例, 可将隐函数定理看作线性消元理论的非线性推广.

此外, 该定理经常被运用到参数化的线性方程组中. 在这些方程组中, \mathbf{z} 可看作 $\mathbf{M}(t)\mathbf{z} = \mathbf{g}(t)$ 的解, 其中 $\mathbf{M}(\cdot) \in \mathbb{R}^{n \times n}$ 且 $\mathbf{M}(0)$ 非奇异, $\mathbf{g}(\cdot) \in \mathbb{R}^n$. 为了运用定理, 定义

$$\mathbf{h}(\mathbf{z}, t) = \mathbf{M}(t)\mathbf{z} - \mathbf{g}(t)$$

如果 $\mathbf{M}(\cdot)$ 和 $\mathbf{g}(\cdot)$ 在 0 的某邻域内可微, 定理蕴含着 $\mathbf{z}(t) = \mathbf{M}(t)^{-1}\mathbf{g}(t)$ 在 0 的某邻域内是关于 t 的连续可微函数. 7.3 节讨论最优性的一阶必要条件时用到了该定理.

\mathbf{o}, \mathbf{O} 记号

在分析算法的收敛速度时, 经常会关心所得点列的最终 (eventually) (即当沿着点列渐远

时)行为. 例如, 会问序列的元素是否是有界的, 或者是否在尺寸上相似于一个对应序列的元素, 或者是否是递减的, 如果是, 会有多快. 当要检查类似于这样的问题时, 使用阶的符号(order notation)可以避免引入许多常数, 从而使讨论和分析变得更简洁.

如果 g 是实变量的实值函数, 记号 $g(x)=O(x)$ 指 $g(x)$ 趋于 0 的速度至少与 x 趋于 0 的速度一样快. 更精确地说, 存在 $K \geq 0$, 使得当 x 充分小时, 有 $|g(x)/x| \leq K$. 记号 $g(x)=o(x)$ 指 $g(x)$ 趋于 0 比 x 趋于 0 快, 即 $\lim_{x \rightarrow 0} g(x)/x = 0$.

类似地, 给定两个非负标量序列 $\{\eta_k\}$ 和 $\{\nu_k\}$, 记号 $\eta_k=O(\nu_k)$ 指存在正常数 C 使得对充分大的 k 有 $|\eta_k| \leq C|\nu_k|$. 如果比值序列 $\{\eta_k/\nu_k\}$ 趋于零, 即 $\lim_{k \rightarrow \infty} \eta_k/\nu_k = 0$, 记为 $\eta_k=o(\nu_k)$.

作为上面定义的一个稍微变形, 用记号 $\eta_k=O(1)$ 来表示存在常数 C 使得对所有 k 有 $|\eta_k| \leq C$; 同时用记号 $\eta_k=o(1)$ 表明 $\lim_{k \rightarrow \infty} \eta_k = 0$. 有时以向量和矩阵作为自变量, 在这种情况下, 上面的定义中应用这些量的范数. 例如, 如果 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, 则用 $f(x)=O(\|x\|)$ 表示存在常数 $C > 0$, 使得对 f 的定义域内的所有 x 有 $\|f(x)\| \leq C\|x\|$. 一般仅对 f 的定义域内的某些子集感兴趣, 比如 0 的一个小邻域.

A.7 矩阵分解

矩阵分解在算法设计和分析中都很重要. 下面介绍本书涉及的 4 种矩阵分解: LU 分解、Cholesky 分解、QR 分解和奇异值分解.

下面将要描述的所有分解算法都利用到置换矩阵 (permutation matrices). 假设需要交换矩阵 A 的第 1 行和第 4 行. 通过给矩阵 A 左乘一个置换矩阵 P 可以执行该运算, 其中的置换矩阵即交换(和 A 有相同行数的)单位矩阵的第 1 行和第 4 行所得到的矩阵. 例如, 假设 A 是 5×5 的矩阵, 此时 P 为

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

利用类似的技术可以找到交换矩阵的列向量的置换矩阵 P .

A.7.1 高斯消元法与 LU 分解

下面先描述求解线性方程组的高斯消元法 (Gaussian elimination), 它不仅最流行, 而且快, 对舍入误差的积累也不敏感. 这里主要强调这种经典的消元技术本身, 以及它与非奇异矩阵的 LU 分解理论的关系.

首先说明求解三角形方程组很容易. 考虑方程组

$$\begin{aligned}
 a_{11}x_1 &= b_1 \\
 a_{21}x_1 + a_{22}x_2 &= b_2 \\
 \vdots &\vdots \\
 a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
 \end{aligned}$$

假设每个对角线元素 a_{ii} ($i=1, 2, \dots, n$) 非零 (如果系统非奇异, 则该事实必然成立), 按如下递归方式

$$\begin{aligned}
 x_1 &= b_1/a_{11} \\
 x_2 &= (b_2 - a_{21}x_1)/a_{22} \\
 \vdots &\vdots \\
 x_n &= (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn}
 \end{aligned}$$

可以得到解. 将任意一个方程组化归成三角形方程组后, 就可以方便地得到解.

如果对 $i < j$, 有 $l_{ij} = 0$, 则称方阵 $\mathbf{L} = [l_{ij}]$ 是下三角的 (lower triangular). 类似地, 如果对 $i > j$ 有 $u_{ij} = 0$, 则称 \mathbf{U} 是上三角的 (upper triangular).

在矩阵概念中, 高斯消元法的本质就是找到给定的 $n \times n$ 阶矩阵 \mathbf{A} 的形如 $\mathbf{A} = \mathbf{LU}$ 的分解, 其中 \mathbf{L} 是下三角矩阵, \mathbf{U} 是上三角矩阵. 从而通过求解两个三角形方程组

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{y} \quad (\text{A. 7. 1})$$

得到方程组

$$\mathbf{Ax} = \mathbf{b} \quad (\text{A. 7. 2})$$

的解. 具体地, 先得到分解 \mathbf{L}, \mathbf{U} ; 再利用向前消元法 (forward elimination) 求解 $\mathbf{Ly} = \mathbf{b}$, 最后利用回代 (back substitution) 求解 $\mathbf{Ux} = \mathbf{y}$.

在对非奇异方阵 \mathbf{A} 进行 LU 分解时, 如果必要可以交换 \mathbf{A} 的行——这实际是对方程组中的方程进行重新排序, 显然不影响问题的讨论. 然而为了记号简单, 假设不需要进行这样的交换.

考虑如何由消元法确定非奇异矩阵 \mathbf{A} 的 LU 分解因子 \mathbf{L} 和 \mathbf{U} . 给定系统, 试图通过初等行变换使得非零元素仅出现在主对角线以下. 假设 $a_{11} \neq 0$, 可以给其他方程减去第一个方程的倍数使得第一列中 a_{11} 之下的元素全变为 0. 如果定义 $m_{ki} = a_{ki}/a_{11}$, 并设

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & & 1 & & \\ \vdots & & & \ddots & \\ -m_{n1} & & & & 1 \end{bmatrix}$$

可以将新方程组表示为

$$\mathbf{A}^{(2)} \mathbf{x} = \mathbf{b}^{(2)}$$

其中

$$\mathbf{A}^{(2)} = \mathbf{M}^{(1)} \mathbf{A}, \quad \mathbf{b}^{(2)} = \mathbf{M}^{(1)} \mathbf{b}$$

对于矩阵 $\mathbf{A}^{(2)} = [a_{ij}^{(2)}]$ 有 $a_{ki}^{(2)} = 0, k > 1$.

接下来, 假设 $a_{22}^{(2)} \neq 0$. 在新方程组中, 从第 3 到第 n 个方程减去第 2 个方程的倍数, 以使得第二列中 $a_{22}^{(2)}$ 以下的元素为 0. 这等价于给 $\mathbf{A}^{(2)}$ 和 $\mathbf{b}^{(2)}$ 左乘

$$\mathbf{M}^{(2)} = \begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & \\ 0 & -m_{32} & 1 & & & \\ 0 & -m_{42} & & 1 & & \\ \vdots & \vdots & & & \ddots & \\ 0 & -m_{n2} & & & & 1 \end{bmatrix}$$

其中 $m_{k2} = a_{k2}^{(2)} / a_{22}^{(2)}$. 这产生 $\mathbf{A}^{(3)} = \mathbf{M}^{(2)} \mathbf{A}^{(2)}$ 和 $\mathbf{b}^{(3)} = \mathbf{M}^{(2)} \mathbf{b}^{(2)}$. 依此类推, 将有 $\mathbf{M} = \mathbf{M}^{(n-1)} \mathbf{M}^{(n-2)} \cdots \mathbf{M}^{(1)} \mathbf{A}$ 是一个下三角矩阵, 且因为 $\mathbf{M}\mathbf{A} = \mathbf{U}$, 有 $\mathbf{A} = \mathbf{M}^{-1} \mathbf{U}$. 从而, 矩阵 $\mathbf{L} = \mathbf{M}^{-1}$ 也是下三角矩阵, 并且成为 \mathbf{A} 的 LU 分解中所期望的 \mathbf{L} .

因为 $\mathbf{M}^{(k)-1}$ 与 $\mathbf{M}^{(k)}$ 相比, 非对角元素的符号相反, 其余均相同. 基于该事实, 可以给出 \mathbf{M} 更明确的表示. 进一步有

$$\mathbf{L} = \mathbf{M}^{-1} = \mathbf{M}^{(1)-1} \mathbf{M}^{(2)-1} \cdots \mathbf{M}^{(n-1)-1}$$

易验证

$$\mathbf{L} = \begin{bmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & m_{32} & 1 & & & \\ \vdots & \vdots & & \ddots & & \\ m_{n1} & m_{n2} & m_{n3} & \cdots & 1 \end{bmatrix}$$

因此, 可以由消元过程所需的计算直接得到 \mathbf{L} .

如果仅是针对某个向量 \mathbf{b} 来求解原始系统 (A. 7. 1), 一般同时计算出满足 $\mathbf{Ly} = \mathbf{b}$ 的向量 $\mathbf{y} = \mathbf{b}^{(n)} = \mathbf{Mb}$. 进行回代后, 从 $\mathbf{Ux} = \mathbf{y}$ 即可得到最终的解 \mathbf{x} . 一旦得到 \mathbf{A} 的 LU 分解, 则相应于任何右端向量, 都可以求解 (A. 7. 2) 中的两个系统得到待求方程组的解.

实践中, $\mathbf{A}^{(k)}$ 的对角元素 $a_{kk}^{(k)}$ 可能变为 0 或非常接近 0. 这时, 将第 k 行与它下面的某行交换就显得很重要. 为了考虑数值精度, 最好确保对所有的 i, j , 以 $|m_{ij}| \leq 1$ 这种方式连续地引入行交换. 如果这样做, 则高斯消元过程的稳定性会非常好.

综上所述, 矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 的 LU 分解定义为

$$\mathbf{PA} = \mathbf{LU} \tag{A. 7. 3}$$

其中 \mathbf{P} 是 $n \times n$ 的置换矩阵(通过重新安排 $n \times n$ 单位矩阵的行可得到该矩阵), \mathbf{L} 是单位下三角矩阵(对角线元素等于 1 的下三角矩阵), \mathbf{U} 是上三角矩阵. 利用式 (A. 7. 3) 可以有效求解线性方程组 (A. 7. 2), 具体步骤如下:

- ① 改变 \mathbf{b} 的元素次序形成 $\tilde{\mathbf{b}} = \mathbf{Pb}$;
- ② 执行向前消元法得到 $\mathbf{Lz} = \tilde{\mathbf{b}}$ 的解 \mathbf{z} ;
- ③ 执行回代得到 $\mathbf{Ux} = \mathbf{z}$ 的解 \mathbf{x} .

利用部分行交换的高斯消元可得到分解式 (A. 7. 3). 当 \mathbf{A} 稠密时, 该算法近似地需要 $2n^3/3$ 次浮点运算. 易于得到实现该算法的标准软件(尤其是 LAPACK). 算法 A. 7. 1 是该方法的伪码描述.

Algorithm A. 7.1 Gaussian elimination with row partial pivoting

```

1: Given  $A \in \mathbb{R}^{n \times n}$ ; Set  $P \leftarrow I, L \leftarrow 0$ ;
2: for  $i = 1, 2, \dots, n$  do
3:   find a index  $j$  such that  $|a_{ij}| = \max\{|a_{ik}| : k = i, i+1, \dots, n\}$ ;
4:   if  $a_{ij} = 0$  then
5:     stop. (* matrix  $A$  is singular *)
6:   end if
7:   if  $i \neq j$  then
8:     swap rows  $i$  and  $j$  of matrices  $A$  and  $L$ ;
9:   end if
10:   $l_{ii} \leftarrow 1$ ; (* elimination step *)
11:  for  $k = i+1, i+2, \dots, n$  do
12:     $l_{ki} \leftarrow a_{ki} / a_{ii}$ ;
13:    for  $l = i+1, i+2, \dots, n$  do
14:       $a_{kl} \leftarrow a_{kl} - l_{ki} a_{il}$ ;
15:    end for
16:  end for
17: end for
18:  $U \leftarrow$  upper triangular part of  $A$ .

```

基本算法的变形允许在分解过程中同时重排行和列,但是这些不能增加算法的实用稳定性. 然而,当矩阵 A 是稀疏的时,列转轴可以确保因子 L 和 U 也合理地稀疏,从而提升高斯消元法的性能.

A. 7.2 Cholesky 分解

当 $A \in \mathbb{R}^{n \times n}$ 是对称正定矩阵时,以大约一半的费用($n^3/3$ 次运算)可以得到类似的但更特殊的分解,即 Cholesky 分解,其产生一个下三角矩阵 L 使得

$$A = LL^\top \quad (\text{A. 7. 4})$$

如果要求 L 的对角线元素为正,则满足式(A. 7. 4)的 L 是唯一的. 方法的伪码描述见算法 A. 7. 2.

Algorithm A. 7.2 Cholesky factorization

```

1: Given  $A \in \mathbb{R}^{n \times n}$  symmetric positive definite;
2: for  $i = 1, 2, \dots, n$  do
3:    $l_{ii} \leftarrow \sqrt{a_{ii}}$ ;
4:   for  $j = i+1, i+2, \dots, n$  do
5:      $l_{ji} \leftarrow a_{ji} / l_{ii}$ ;
6:     for  $k = i+1, i+2, \dots, j$  do
7:        $a_{jk} \leftarrow a_{jk} - l_{ji} l_{ki}$ ;
8:     end for
9:   end for
10: end for

```

注意,该方法仅涉及 A 的下三角位置的元素.事实上,在任一种情况下,仅需存储这些元素即可(由对称性,可以将它们简单地复制在上三角的位置).

不同于高斯消元,Cholesky 分解不用交换任何行和列即可产生对称正定矩阵的有效分解.然而,可以利用对称置换(即以同样的方式对行和列重新排序)来提高因子 L 的稀疏性.在这种情况下,算法对某置换矩阵 P 产生形如

$$P^T AP = LL^T$$

的分解.

像高斯消元法产生的因子 L 和 U 的情形一样,可以利用 Cholesky 分解,通过执行分别由 L 和 L^T 确定的向前消元法和回代得到方程组(A. 7. 2)的解.

A. 7.3 QR 分解

考虑长方形矩阵 $A \in \mathbb{R}^{m \times n}$,其中 $m \geq n$ 且 A 是列满秩的. A 的另一种有用分解形如

$$AP = QR \quad (\text{A. 7. 5})$$

其中 P 是 $n \times n$ 置换矩阵, Q 是 $m \times m$ 正交矩阵, R 是 $m \times n$ 上三角矩阵.在方阵 $m = n$ 的情况,可以通过下面的过程利用该分解计算线性方程组(A. 7. 2)的解:

- ① $\tilde{b} = Q^T b$;
- ② 执行回代得到 $Rz = \tilde{b}$ 的解;
- ③ 重新安排 z 的元素,即置 $x = Pz$.

对于稠密矩阵 A ,计算 QR 分解需要大约 $4m^2n/3$ 次算术运算,这大约是通过高斯消元法计算 LU 分解所需计算量的两倍.与高斯消元法相比,一般不能修正 QR 分解以确保分解因子是稀疏矩阵,即不管如何选取列置换矩阵 P ,因子 Q 和 R 一般都是稠密的(即使进行列转轴和行转轴,情况仍然是这样).

执行 QR 分解的算法几乎与高斯消元法及 Cholesky 分解的算法一样简单.大多数广泛流行的算法是对 A 运用一系列的特殊正交矩阵,如 **Householder 变换** (Householder transformation)或者**吉文斯旋转**(Givens rotation).完整的算法描述请参看参考文献[24]的第 5 章.

当矩阵是长方形的时($m < n$),可以利用 A^T 的 QR 分解来找到以 A 的零空间的基为列形成的矩阵.更确切地记

$$A^T P = QR = [Q_1 \quad Q_2]R$$

其中 Q_1 由 Q 的前 m 列组成, Q_2 包含后 $n - m$ 列.易于说明矩阵 Q_2 的列生成 A 的零空间.这种方法产生零空间的基矩阵比高斯消元法产生的更令人满意,因为 Q_2 的列相互正交,且有单位长度.然而,它的计算开销大,当 A 稀疏时尤其明显.

当 A 列满秩时,对式(A. 7. 5)中的 R 因子和 Cholesky 分解因子进行比较.将式(A. 7. 5)乘以它的转置,得到

$$P^T A^T A P = R^T Q^T Q R = R^T R$$

与式(A. 7. 4)相比,看到 R^T 是对称正定矩阵 $P^T A^T A P$ 的 Cholesky 因子.当限制它的对角线元素是正的时, L 是唯一的.该观察蕴含着对于选定的置换矩阵 P ,倘若强迫 R 的对角线元素是正的,则 R 也是唯一的.也可以对式(A. 7. 5)稍加等价变形,得到 $A P R^{-1} = Q$.在这种情况下,可以断言 Q 也是唯一的.

由欧氏范数的性质知,式(A. 3. 1)和式(A. 7. 5)中的矩阵 P 和 Q 的欧氏范数都是 1,于是有

$$\|A\| = \|Q R P^T\| \leq \|Q\| \|R\| \|P^T\| = \|R\|$$

同时

$$\|\mathbf{R}\| = \|\mathbf{Q}^T \mathbf{A} \mathbf{P}\| \leq \|\mathbf{Q}^T\| \|\mathbf{A}\| \|\mathbf{P}\| = \|\mathbf{A}\|$$

由这两个不等式,可以断言 $\|\mathbf{A}\| = \|\mathbf{R}\|$. 当 \mathbf{A} 是方阵时,通过类似的讨论,有 $\|\mathbf{A}^{-1}\| = \|\mathbf{R}^{-1}\|$. 因此,在表达式(A.3.2)中用 \mathbf{R} 代替 \mathbf{A} 可以得到 \mathbf{A} 的欧氏范数的条件数. 有许多技术可以估计三角矩阵 \mathbf{R} 的条件数,因此该结果意义重大. 更多的讨论见参考文献[24]中的 128~130 页.

A.7.4 奇异值分解

所有矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 可被分解成

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (\text{A.7.6})$$

其中 \mathbf{U} 和 \mathbf{V} 分别是 $m \times m$ 和 $n \times n$ 的正交矩阵,即它们满足关系 $\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$ 和 $\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}$, \mathbf{S} 是 $m \times n$ 对角矩阵. 对角线元素为 $\sigma_i, i=1, 2, \dots, \min\{m, n\}$, 其满足

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{m, n\}} \geq 0$$

称这些对角线元素为 \mathbf{A} 的奇异值(singular value),称式(A.7.6)为矩阵 \mathbf{A} 的奇异值分解(Singular Value Decomposition, SVD).

若 \mathbf{A} 是对称正定矩阵,则它的奇异值和特征值是一致的,且对于欧氏范数有

$$\|\mathbf{A}\| = \sigma_1(\mathbf{A}) = \lambda_{\max}(\mathbf{A}), \quad \|\mathbf{A}^{-1}\| = \sigma_1(\mathbf{A}^{-1}) = \frac{1}{\lambda_{\min}(\mathbf{A})}$$

因此,对所有 $\mathbf{x} \in \mathbb{R}^n$ 有

$$\sigma_n(\mathbf{A}) \|\mathbf{x}\|^2 = \|\mathbf{x}\|^2 / \|\mathbf{A}^{-1}\| \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \leq \|\mathbf{A}\| \|\mathbf{x}\|^2 = \sigma_1(\mathbf{A}) \|\mathbf{x}\|^2$$

对于正交矩阵 \mathbf{Q} 及欧氏范数,有 $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$,且该矩阵的所有奇异值都等于 1.

A.8 其他

A.8.1 标量方程求根

若 $f(\mathbf{x})$ 可微,则 $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ 的必要条件是 $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. 这里简要讨论解标量方程 $g(x) = 0$ 的方法. 牛顿法的基本步

$$s^{(k)} = -g(x^{(k)})/g'(x^{(k)}), \quad x^{(k+1)} \leftarrow x^{(k)} + s^{(k)} \quad (\text{A.8.1})$$

从图形上看,该更新步利用了 g 的图形在对应点 $x^{(k)}$ 处的切线,并将该切线和 x 轴的交点作为下一个迭代点(见图 A.8.1(a)). 如果函数 g 是几乎线性的,则该切线将会是 g 本身一个很好的近似,因此牛顿迭代将会与 g 真正的根相当接近.

而标量方程的割线法可以看作极小化单变量函数 $f(x)$ 的 BFGS 法,其中 $f'(x) = g(x)$. 在这种特殊情况下,割线方程 $\mathbf{B}^{(k)} \mathbf{s}^{(k)} = \mathbf{y}^{(k)}$ 完全确定了 1×1 阶近似 Hessian 阵的值. 这样,不需要应用额外的条件就可以完全确定 $\mathbf{B}^{(k)}$. 由 BFGS 法,当 $n=1$ 时

$$\begin{aligned} \mathbf{B}^{(k)} &= [g(x^{(k)}) - g(x^{(k-1)})]/(x^{(k)} - x^{(k-1)}) \\ s^{(k)} &= -g(x^{(k)})/\mathbf{B}^{(k)}, \quad x^{(k+1)} = x^{(k)} + s^{(k)} \end{aligned} \quad \left. \right\} \quad (\text{A.8.2})$$

这就是割线法的公式. 考虑过点 $(x^{(k-1)}, g(x^{(k-1)}))$ 和 $(x^{(k)}, g(x^{(k)}))$ 的割线(见图 A.8.1(b)),用它的斜率 $\mathbf{B}^{(k)}$ 作为函数在 $x^{(k)}$ 的斜率($f(x)$ 的二阶导数)的近似,求割线和 x 轴的交点得到 $x^{(k+1)}$. 应用割线法时,必须给出两个初始点 $x^{(0)}$ 和 $x^{(1)}$.

例 A.8.1 应用割线法求解问题 $\min \sin x$. 此时, $g(x) = f'(x) = \cos x$. 此处令 $x^{(0)} = 0, x^{(1)} = -1$, 得

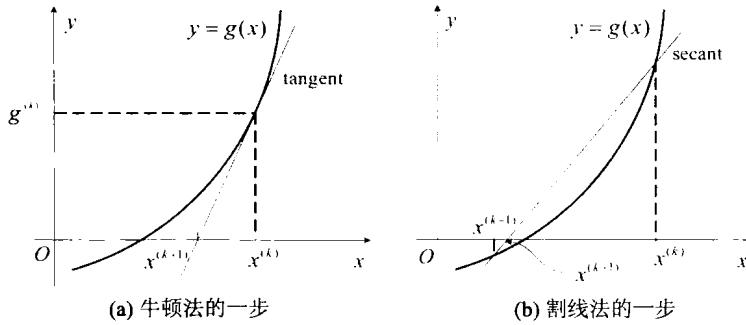


图 A.8.1 解标量方程的牛顿法(切线法)和割线法

$$\begin{aligned}
 x^{(2)} &= x^{(1)} - \frac{(x^{(1)} - x^{(0)})}{g(x^{(1)}) - g(x^{(0)})} g(x^{(1)}) \\
 &= -1 - \frac{(-1 - 0)}{\cos(-1) - \cos(0)} \cos(-1) \\
 &= -1 + \frac{0.5403}{0.5403 - 1} \\
 &= -2.1753
 \end{aligned}$$

后续的迭代依次为 $x^{(3)} = -1.5728, x^{(4)} = -1.5707, x^{(5)} = -1.5708 \approx -\pi/2$. 该序列收敛于所给问题的解.

A.8.2 误差分析和浮点计算

本书中大部分的算法和分析都是针对实数进行的. 然而, 现代数字计算机不能对一般的实数进行完全精确的存储和计算. 人们用浮点数(floating-point numbers)的子集代替实数进行计算. 存储在计算机中的任一量, 不管是直接从一个文件或者程序读出的, 还是作为计算的中间结果出现的, 都必须用一个浮点数来近似. 因而, 由实际计算产生的数一般与那些精确计算产生的数是不同的. 当然, 需要努力地设计恰当的算法使这些区别尽可能地小.

讨论误差时需要分清绝对误差 (absolute error) 和相对误差 (relative error). 如果 x 是某精确量(标量, 向量, 矩阵), \tilde{x} 是它的近似值, 则绝对误差是二者之差的范数, 即 $\|x - \tilde{x}\|$ (通常该定义中的范数可以是 1-范数, 2-范数或者 ∞ -范数中的任一种). 相对误差是绝对误差和精确量的幅值之比, 即

$$\frac{\|x - \tilde{x}\|}{\|x\|}$$

当该比值远远小于 1 时, 可以用近似量 $\|\tilde{x}\|$ 代替分母. 这样做, 对它的值没有太大影响.

通常用双精度计算执行优化算法所需的大多数计算. 双精度数是以 64 字节长度的字来存储的. 这些字节中的大部分(比如说 i)比特专用于存储分数部分, 而剩余的编码存储指数 e 和其他信息, 诸如数的符号, 或者指示是 0 还是“未定义”. 典型地, 分数部分形如

$$d_1 d_2 \cdots d_t$$

其中每个 $d_i (i=1, 2, \dots, t)$ 是 0 或者 1(在有些系统中, 隐含假定 d_1 是 1, 而不存储). 浮点数的值是

$$\sum_{i=1}^t d_i 2^{-i} \times 2^e$$

称 2^{-t} 是单位舍入(unit roundoff), 记为 u . 任一绝对值在区间 $[2^L, 2^U]$ (其中 L 和 U 是指数 e 的取值的下界和上界) 的实数, 都可以在相对精度 u 的范围内由一个浮点数近似, 即

$$\text{fl}(x) = x(1 + \epsilon), \quad |\epsilon| \leq u \quad (\text{A. 8.3})$$

其中 $\text{fl}(\cdot)$ 表示浮点近似. 双精度计算的 $\text{fl}(x)$ 的典型值大约是 10^{-15} . 换句话说, 如果实数 x 和它的浮点近似都可写成以 10 为底的数(通常的方式), 它们至少有 15 位数字是相同的. 关于浮点计算的更多信息参见参考文献[24]的 2.4 节.

当对一个或者两个浮点数执行算术运算时, 必须将结果存为浮点数. 该过程会引入小的舍入误差(roundoff error), 可以利用两个输入的尺寸来量化误差的大小. 如果 x 和 y 是两个浮点数, 则有

$$|\text{fl}(x * y) - x * y| \leq u |x * y| \quad (\text{A. 8.4})$$

其中“*”表示 $+, -, \times, \div$ 中的任一运算.

尽管一次浮点数运算的误差看起来是良性的, 但当 x 和 y 都是两个浮点数的近似, 或者接连执行了一系列的计算时, 可能会发生较大的误差. 例如, 假设 x 和 y 都是值非常相似的大实数. 当将它们存储在计算机中时, 用浮点数 $\text{fl}(x)$ 和 $\text{fl}(y)$ 来近似它们, 这两个浮点数满足

$$\text{fl}(x) = x + \epsilon_x, \quad \text{fl}(y) = y + \epsilon_y, \quad |\epsilon_x| \leq u |x|, \quad |\epsilon_y| \leq u |y|$$

如果取这两个存储的浮点数的差, 则得到最终的结果 $\text{fl}(\text{fl}(x) - \text{fl}(y))$ 满足

$$\text{fl}(\text{fl}(x) - \text{fl}(y)) = (\text{fl}(x) - \text{fl}(y))(1 + \epsilon_{xy}), \quad |\epsilon_{xy}| \leq u$$

将这两个表达式接合起来, 会发现该结果和真正值 $x - y$ 之间的差可能与

$$\epsilon_x + \epsilon_y + \epsilon_{xy} |x - y|$$

一样大, 而这个量由 $u(|x| + |y| + |x - y|)$ 界定. 当 x 和 y 很大且很接近时, 因为 $|x| \gg |x - y|$, 所以相对误差大约是 $2u|x|/|x - y|$, 这是相当大的.

称该现象为抵消(cancellation). 这种现象也可以作如下不太正式的解释: 如果 x 和 y 都精确到 k 位数字, 并且它们的前 \tilde{k} 位数字是一致的, 则前 \tilde{k} 个有效数字会相互抵消, 从而仅包含大约 $k - \tilde{k}$ 个有效数字. 这也是著名的数值计算格言“人们应该在所有可能的场合避免对两个相近数求差”产生的原因.

A. 8.3 条件数和稳定性

条件数(conditioning)和稳定性(stability)是两个与数值计算联系在一起的常用术语. 通常, 在关于线性代数、优化、微分方程等问题的数值解中, 需要考虑当确定问题的数据有小的扰动时, 对数值问题的解所带来的影响. 条件数度量了这种影响的最坏情况下的渐近性能, 代表着数值问题自身的适定性. 称低条件数的问题是良态的(well conditioned), 其表示对确定这些问题的数据进行小的扰动时, 对问题的解的影响不大. 称高条件数的问题是病态的(ill conditioned). 比如与线性方程组 $\mathbf{Ax} = \mathbf{b}$ 相联系的条件数是用矩阵 \mathbf{A} 的条件数来刻画的.

下面以 2×2 线性方程组为例进行说明. 首先考虑问题

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

计算系数矩阵的逆, 得到解

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

如果用 3.000 01 代替右端项的第一个元素, 结果变成

$$(x_1, x_2)^\top = (0.999 99, 1.000 01)^\top$$

这与原精确解 $(1, 1)^T$ 仅有微小差异. 如果对右端项的另一个元素或者系数矩阵的元素进行扰动, 解的变换不大, 则可断言该问题是良态的. 而问题

$$\begin{bmatrix} 1.000 & 01 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.000 & 01 \\ 2 \end{bmatrix}$$

是病态的, 它的精确解是 $x = (1, 1)$. 但是, 如果将右端项的第一个元素从 2.000 01 变成 2, 解将急剧地变成 $x = (0, 2)^T$.

对于一般的线性方程组 (A. 7. 2), 可以利用矩阵的条件数 (定义见式 (A. 3. 2)) 来量化这种性质. 具体地, 如果将 A 扰动成 \tilde{A} , 将 b 扰动成 \tilde{b} , 并记 \tilde{x} 为扰动系统 $\tilde{A}\tilde{x} = \tilde{b}$ 的解, 则可以证明

$$\frac{\|x - \tilde{x}\|}{\|x\|} \approx \kappa(A) \left(\frac{\|A - \tilde{A}\|}{\|A\|} + \frac{\|b - \tilde{b}\|}{\|b\|} \right)$$

(可参见参考文献 [24] 的 2.7 节). 因此, 若条件数 $\kappa(A)$ 很大, 则表明问题 (A. 7. 2) 是病态的, 类似地, 若条件数比较适中, 则表明问题是良态的.

注意, 条件数的概念与用来求解问题的特定算法没有任何关系, 仅与数值问题本身有关.

稳定性 (stability) 是算法的性质. 如果一个算法对该问题类中的所有良态问题均产生一个高度精确的解答 (即使是浮点计算), 则称该算法是稳定的.

作为一个范例, 再次考虑线性方程组 (A. 7. 2). 可以证明: 结合向前消元法时, 算法 A. 7. 1 所得解的相对误差

$$\frac{\|x - \tilde{x}\|}{\|x\|} \approx \kappa(A) \frac{\text{growth}(A)}{\|A\|} u \quad (\text{A. 8. 5})$$

其中 $\text{growth}(A)$ 是算法 A. 7. 1 在执行过程中产生的 A 中最大的元素的绝对值. 在最坏情况下, 可以证明 $\text{growth}(A)/\|A\|$ 近似为 2^{n-1} . 其表明算法 A. 7. 1 是一个不稳定算法, 因为即使对适中的 n (比如 $n=200$), 甚至当 $\kappa(A)$ 是适中的时, 式 (A. 8. 5) 的右端项也可能会很大. 然而在实践中, 很少观察到大的增长因子, 因此断言算法 A. 7. 1 对所有实际问题是稳定的.

另一方面, 没有转轴的高斯消元法无疑是不稳定的. 如果省掉算法 A. 7. 1 中可能的行交换, 那么算法甚至对某些良态矩阵都得不到分解, 诸如

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

当式 (A. 7. 2) 中的 A 是对称正定的时, Cholesky 分解与三角回代相结合的算法是求解问题的稳定算法.

附录 B 阅读材料

参考文献[13]中有丰富的例子和练习,也有工程中出现的大量问题.这里选取了几个刻画最优解的典型例子,供读者阅读.

半定规划除了在最大割问题中的典型应用外,在其他组合问题中也有很广泛的应用,这里给出半定规划在 MAX-2-SAT 问题中的应用.

B. 1 KKT 条件和对偶理论的应用实例

B. 1. 1 KKT 条件的力学解释

KKT 条件在力学背景下有很直观的解释(当然,这也是 Lagrange 对此进行研究的初衷).下面用一个简单的例子来说明其力学解释.

如图 B. 1. 1 所示的系统,用 3 个弹簧将滑块 A 和 B 以及左右两面墙壁连在一起.两个滑块的位置用 $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$ 表示,其中 x_1 是滑块 A 的位置(质心), x_2 是滑块 B 的位置(质心).左面墙壁位置设为 0,右面墙壁位置设为 l . 滑块的宽为 $w > 0$,滑块不可能穿入墙内或者另一个滑块内.

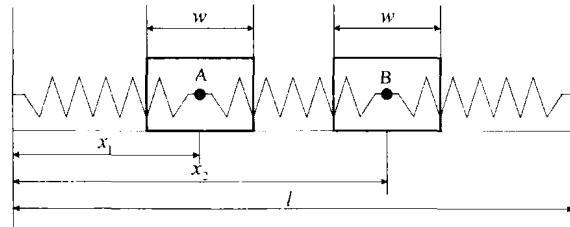


图 B. 1. 1 弹簧-滑块系统

弹簧的弹性势能可以表示成滑块位置的函数,即

$$f(x_1, x_2) = \frac{1}{2}k_1 x_1^2 + \frac{1}{2}k_2(x_2 - x_1)^2 + \frac{1}{2}k_3(l - x_2)^2$$

其中 $k_i > 0$ 分别是 3 个弹簧的弹性系数,且 x_1, x_2 满足不等式

$$x_1 - \frac{w}{2} \geq 0, \quad -x_1 + x_2 - w \geq 0, \quad -x_2 + \frac{w}{2} + l \geq 0$$

系统的平衡位置 \mathbf{x}^* 即为势能函数在上述约束条件下的最小点.

这些约束称为运动约束,代表的意义是两个滑块的宽度 $w > 0$,而且滑块不能穿入另外一个滑块或者墙壁.受力平衡点 \mathbf{x}^* 就是二次规划问题

$$\begin{aligned}
 & \text{minimize} && \frac{1}{2} [k_1 x_1^2 + k_2 (x_2 - x_1)^2 + k_3 (l - x_2)^2] \\
 & \text{subject to} && x_1 - \frac{w}{2} \geq 0 \\
 & && -x_1 + x_2 - w \geq 0 \\
 & && -x_2 - \frac{w}{2} + l \geq 0
 \end{aligned} \tag{B. 1. 1}$$

的解。

记问题的 Lagrange 乘子是 $\lambda_1, \lambda_2, \lambda_3$, 那么二次规划问题(B. 1. 1)的 KKT 条件由以下几部分组成: 运动约束、非负约束 $\lambda_i \geq 0$ 和互补条件

$$\left. \begin{aligned} \lambda_1(x_1 - w/2) &= 0 \\ \lambda_2(-x_1 + x_2 - w) &= 0 \\ \lambda_3(-x_2 - w/2 + l) &= 0 \end{aligned} \right\} \tag{B. 1. 2}$$

以及稳定点条件

$$\begin{bmatrix} k_1 x_1 - k_2 (x_2 - x_1) \\ k_2 (x_2 - x_1) - k_3 (l - x_2) \end{bmatrix} + \lambda_1 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \lambda_3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{0} \tag{B. 1. 3}$$

如图 B. 1. 2 所示, 施加在每个滑块上的力, 包括由弹簧产生的弹力以及滑块之间和滑块与墙壁间接触时所产生的力, 总和必为 0. 图中上面一行显示的是 Lagrange 乘子, 即滑块与滑块、滑块与墙壁之间的接触力; 下面一行显示了弹簧的弹力. 这里的 Lagrange 乘子可以解释为滑块与滑块、滑块与墙壁之间所产生的接触力(contact force), 因此可以将等式(B. 1. 3)理解为在两个滑块上作用的力分别达到平衡. 式(B. 1. 3)中第一个等式表示作用在滑块 A 上的力的总和为零, 其中 $-k_1 x_1$ 为左边的弹簧作用在滑块 A 上的弹力, $k_2 (x_2 - x_1)$ 为中间的弹簧所产生的弹力, λ_1 为左面墙壁所施加的接触力, $-\lambda_2$ 为滑块 B 所作用的接触力. 接触力必然垂直于接触面并向外(约束条件中的 $\lambda_1 \geq 0, -\lambda_2 \leq 0$ 即说明了该事实), 并且当滑块与接触面发生接触的时候接触力才可能非零(这是互补条件(B. 1. 2)中前两个等式所代表的意思). 也可以类似地解释等式(B. 1. 3)中的第二个等式. 互补条件中的最后一个等式代表的意思是, 如果滑块 B 和右面的墙壁没有发生接触, 那么 λ_3 为 0.

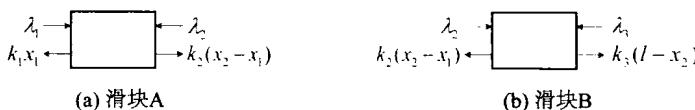


图 B. 1. 2 系统的受力分析

在这个模型中, 势能函数和运动约束都为凸函数, 而且如果两面墙之间有足够的空间容纳两个滑块, 即 $2w \leq l$, 那么 Slater 约束规范自然成立. 因此由 KKT 条件可以得到系统(B. 1. 1)的最优解, 同样这也是系统的受力平衡点.

B. 1. 2 KKT 条件的应用实例

考虑凸优化问题

$$\begin{aligned}
 & \underset{\mathbf{x} \geq \mathbf{0}}{\text{maximize}} \quad \sum_{i=1}^n \log(a_i + x_i) \\
 & \text{subject to} \quad \sum_{i=1}^n x_i = 1
 \end{aligned} \tag{B.1.4}$$

其中 $a_i > 0$ 表示噪声功率. 该问题是信息论中产生的, 是给 n 个通信信道分配功率时碰到的问题. 变量 \mathbf{x} 代表分配给第 i 个信道的发送功率, $\log(a_i + x_i)$ 给出了信道的容量或者通信速率, 因此问题是将全部功率 1 分配给信道来极大化整个系统的通信速率, 即信道容量.

对于不等式约束 $\mathbf{x} \geq \mathbf{0}$ 引入 Lagrange 乘子 $\boldsymbol{\lambda}^* \in \mathbb{R}^n$, 对于等式约束引入乘子 $\nu^* \in \mathbb{R}$, 得到 KKT 条件

$$\begin{aligned}
 \mathbf{x}^* \geq \mathbf{0}, \quad \sum_{i=1}^n x_i^* = 1, \quad \boldsymbol{\lambda}^* \geq \mathbf{0}, \quad \lambda_i^* x_i^* = 0, \quad i = 1, 2, \dots, n \\
 -\frac{1}{a_i + x_i^*} - \lambda_i^* + \nu^* = 0, \quad i = 1, 2, \dots, n
 \end{aligned}$$

可以直接解这些方程得到 \mathbf{x}^* , $\boldsymbol{\lambda}^*$ 和 ν^* . 由于 λ_i^* 充当着最后一个方程的松弛变量, 因此消去 λ_i^* 得到

$$\mathbf{x}^* \geq \mathbf{0}, \quad \sum_{i=1}^n x_i^* = 1 \tag{B.1.5a}$$

$$\nu^* \geq \frac{1}{a_i + x_i^*}, \quad i = 1, 2, \dots, n \tag{B.1.5b}$$

$$\left(\nu^* - \frac{1}{a_i + x_i^*} \right) x_i^* = 0, \quad i = 1, 2, \dots, n \tag{B.1.5c}$$

如果 $\nu^* < 1/a_i$, 不等式 (B.1.5b) 成立的必要条件是 $x_i^* > 0$, 而这时由等式 (B.1.5c) 得 $\nu^* = 1/(a_i + x_i^*)$. 因此, 当 $\nu^* < 1/a_i$ 时, $x_i^* = 1/\nu^* - a_i$. 如果 $\nu^* \geq 1/a_i$, 又因为 $x_i^* > 0$, 则蕴含着 $\nu^* \geq 1/a_i > 1/(a_i + x_i^*)$, 这违背了互补条件, 故 $x_i^* > 0$ 是不可能的. 因此, 当 $\nu^* \geq 1/a_i$ 时, $x_i^* = 0$. 这样有

$$x_i^* = \begin{cases} \frac{1}{\nu^*} - a_i, & \nu^* < \frac{1}{a_i} \\ 0, & \nu^* \geq \frac{1}{a_i} \end{cases}$$

或者更简洁地表示为 $x_i^* = \max(0, 1/\nu^* - a_i)$. 将该表达式代入式 (B.1.5a), 得

$$\sum_{i=1}^n \max(0, 1/\nu^* - a_i) = 1$$

上式的右端是 $1/\nu^*$ 的逐段线性函数, 间断点为 a_i . 该方程有唯一解, 且易于确定该解.

通常称该解法为注水 (water-filling) 算法. 具体地, 将 a_i 看作第 i 块地的地面高度, 用深为 $1/\nu^*$ 的水来淹没该区域 (如图 B.1.3 所示), 则用去的全部水量是 $\sum_{i=1}^n \max(0, 1/\nu^* - a_i)$.

可以增加淹没水平, 直到使用的全部水量等于 1, 则第 i 块地上水的深度就是最优解 x_i^* . 每一块地的高度是给定的, 记为 a_i . 区域将要被淹没的水平是 $1/\nu^*$, 总共使用的水量等于 1. 每一块地上水的高度 (阴影所示) 是最优解 x_i^* .

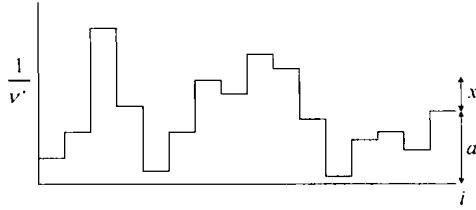


图 B.1.3 注水算法图示

B.1.3 对偶理论的应用实例

例 B.1.1 (熵极大化) 考虑

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && \sum_{i=1}^n x_i \log x_i \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{1}^T \mathbf{x} = 1 \end{aligned} \quad (\text{B.1.6})$$

其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, 且目标函数是负熵. 这个问题是熵极大化问题, 它的 Lagrange 对偶问题是

$$\begin{aligned} & \underset{\boldsymbol{\lambda} \in \mathbb{R}^m, \nu \in \mathbb{R}}{\text{maximize}} && -\mathbf{b}^T \boldsymbol{\lambda} - \nu - e^{-\nu-1} \sum_{i=1}^n e^{-a_i^T \boldsymbol{\lambda}} \\ & \text{subject to} && \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \quad (\text{B.1.7})$$

对于该问题而言, 假如弱 Slater 条件(7.9.2)成立, 即存在 $\mathbf{x}' > \mathbf{0}$ 满足 $\mathbf{A}\mathbf{x}' \leq \mathbf{b}$ 且 $\mathbf{1}^T \mathbf{x}' = 1$, 则强对偶定理(定理 7.9.1)成立且问题(B.1.7)的最优解 $(\boldsymbol{\lambda}^*, \nu^*)$ 存在.

假设已经求解了对偶问题(B.1.7). Lagrange 函数在 $(\boldsymbol{\lambda}^*, \nu^*)$ 处是

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*, \nu^*) = \sum_{i=1}^n x_i \log x_i + \boldsymbol{\lambda}^{*T} (\mathbf{A}\mathbf{x} - \mathbf{b}) + \nu^* (\mathbf{1}^T \mathbf{x} - 1)$$

该函数是严格凸的, 且有下界, 因此有唯一解 \mathbf{x}^* , 其由

$$x_i^* = \frac{1}{\exp(a_i^T \boldsymbol{\lambda}^* + \nu^* + 1)}, \quad i = 1, 2, \dots, n$$

给出, 其中 a_i 是 \mathbf{A} 的列. 如果 \mathbf{x}^* 不是原问题的可行解, 则可以断言原始问题(B.1.6)的最优值不可达.

例 B.1.2 (单个等式约束下可分离函数的极小化) 考虑

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} && \sum_{i=1}^n f_i(x_i) \\ & \text{subject to} && \mathbf{a}^T \mathbf{x} = b \end{aligned} \quad (\text{B.1.8})$$

其中 $\mathbf{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$ 且 $f: \mathbb{R} \rightarrow \mathbb{R}$ 是可微严格凸的. 目标函数之所以称为可分离是因为它是单变量 x_1, x_2, \dots, x_n 的函数之和. 假定目标函数的定义域和约束集相交, 即存在 f 的定义域内的点 \mathbf{x}_0 使得 $\mathbf{a}^T \mathbf{x}_0 = b$. 这蕴含着问题有唯一解 \mathbf{x}^* .

Lagrange 函数

$$\mathcal{L}(\mathbf{x}, \nu) = \sum_{i=1}^n f_i(x_i) + \nu(\mathbf{a}^T \mathbf{x} - b) = -b\nu + \sum_{i=1}^n [f_i(x_i) + \nu a_i x_i]$$

也是可分离的,因此对偶函数

$$\begin{aligned}
 g(\nu) &= -b\nu + \inf_{\mathbf{x}} \sum_{i=1}^n [f_i(x_i) + \nu a_i x_i] \\
 &= -b\nu + \sum_{i=1}^n \inf_{x_i} (f_i(x_i) + \nu a_i x_i) \\
 &= -b\nu - \sum_{i=1}^n f_i^*(-\nu a_i)
 \end{aligned}$$

其中 $f_i^*(y) = \sup_{x_i} (yx_i - f_i(x_i))$ 称为 f_i 的共轭函数 (conjugate function). 这样, 对偶问题是关于标量 $\nu \in \mathbb{R}$ 的极大化问题

$$\underset{\nu \in \mathbb{R}}{\text{maximize}} \quad -b\nu - \sum_{i=1}^n f_i^*(-\nu a_i) \quad (\text{B. 1. 9})$$

有一些简单的方法可以求解具有一个变量的凸问题, 诸如二分法. 假设已经得到对偶问题 (B. 1. 9) 的解 ν^* . 因为每个 f_i 是严格凸的, 从而函数 $\mathcal{L}(\mathbf{x}, \nu^*)$ 关于 \mathbf{x} 严格凸, 因此有唯一极小点 $\bar{\mathbf{x}}$. 又知 \mathbf{x}^* 极小化 $\mathcal{L}(\mathbf{x}, \nu^*)$, 因此必有 $\bar{\mathbf{x}} = \mathbf{x}^*$. 解方程组 $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \nu^*) = 0$ 即可得到 \mathbf{x}^* . 因为原始问题的变量是可分离的, 所以解方程组即转化为解 n 个单变量方程 $f_i'(\mathbf{x}_i^*) = -\nu^* a_i$.

B. 2 MAX-2-SAT 问题的半定规划松弛

SDP 松弛和随机化的方法可以解决许多问题. 实际上, 除了最大割问题外, Goemans 和 Williamson 还考虑了其在 MAX-2-SAT (maximum-2-satisfiability) 问题中的应用. 为了更好地介绍该问题, 首先需要以下的专业术语来了解隶属于计算机科学领域的逻辑理论中的可满足性问题 (satisfiability problem):

- 文字或变量 (如 x_1, \bar{x}_1).
- 布尔子句 (如 $C_1 = \{x_1, \bar{x}_2\}$).
- 可满足子句 (如 $x_1 = T$ 使子句 $C_1 = \{x_1, \bar{x}_2\}$ 满足).

每个可满足子句 C_i 会产生一个权值 w_i . **MAX-SAT** 问题是为了找出这些文字的指派 (真或假) 使总权值最大. 所谓的 **MAX-2-Sat** 问题就是每个子句至多包含两个文字的 MAX-SAT 问题. 利用二次型表述可以对 MAX-2-SAT 问题进行建模. 假设有 n 个文字 x_1, x_2, \dots, x_n . 设 y_0, y_1, \dots, y_n 为辅助变量, 对于 $i=0, 1, \dots, n$, $y_i \in \{-1, +1\}$. 考虑

$$v(x_i) = \frac{1 + y_0 y_i}{2}$$

于是

$$v(\bar{x}_i) = \frac{1 - y_0 y_i}{2}$$

此外

$$\begin{aligned}
 v(x_i \vee x_j) &= 1 - v(\bar{x}_i) v(\bar{x}_j) \\
 &= 1 - \frac{1 - y_0 y_i}{2} \frac{1 - y_0 y_j}{2}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{4} (3 + y_0 y_i + y_0 y_j - y_0^2 y_i y_j) \\
&= \frac{1 + y_0 y_i}{4} + \frac{1 + y_0 y_j}{4} + \frac{1 - y_0^2 y_i y_j}{4}
\end{aligned}$$

现在,对于一个给定的 MAX-2-SAT 实例 (C, w) ,其中 $w_j (\geq 0)$ 是与子句 $C_j \in C$ 对应的权值,可将问题表述为

$$\begin{aligned}
&\text{maximize} \quad \sum_{C_j \in C} w_j v(C_j) \\
&\text{subject to} \quad y_j \in \{-1, +1\}, \quad j = 0, 1, \dots, n
\end{aligned}$$

另外,该问题也可以用二次型表述为

$$\begin{aligned}
&\text{maximize} \quad \sum_{i < j} [a_{ij}(1 - y_i y_j) + b_{ij}(1 + y_i y_j)] \\
&\text{subject to} \quad y_j^2 = 1, \quad j = 0, 1, \dots, n
\end{aligned}$$

其中 a_{ij} 和 b_{ij} 都为非负数. 显然,它的 SDP 松弛型为

$$\begin{aligned}
&\text{maximize} \quad \sum_{i < j} [a_{ij}(1 - x_{ij}) + b_{ij}(1 + x_{ij})] \\
&\text{subject to} \quad \mathbf{X} \geq \mathbf{0}, \quad x_{ii} = 1, \quad i = 1, 2, \dots, n
\end{aligned}$$

如果 \mathbf{X}^* 是上述问题的最优解,再让 $\mathbf{y}(\xi) := \text{sign}(\xi)$,其中 $\xi \sim \mathcal{N}(0, \mathbf{X}^*)$. 在引理 7.8.1 中令 $x_i := -x$,可得

$$1 + \frac{2}{\pi} \arcsin x \geq \alpha(1 + x), \quad \forall x \in [-1, 1]$$

因此

$$\begin{aligned}
&\mathbb{E} \left[\sum_{i < j} [a_{ij}(1 - y_i(\xi) y_j(\xi)) + b_{ij}(1 + y_i(\xi) y_j(\xi))] \right] \\
&= \sum_{i < j} \left[a_{ij} \left(1 - \frac{2}{\pi} \arcsin x_{ij}^* \right) + b_{ij} \left(1 + \frac{2}{\pi} \arcsin x_{ij}^* \right) \right] \\
&\geq \alpha \sum_{i < j} [a_{ij}(1 - x_{ij}^*) + b_{ij}(1 + x_{ij}^*)] \\
&\geq \alpha \times \text{MAX-2-SAT 值}
\end{aligned}$$

也就是说,MAX-2-SAT 问题的逼近因子与最大割问题的相同,都是 0.878 56. 所以,这种分析方法主要依赖于目标函数的二次型的非对角线元素的非正性. 在其他的一些应用中,情况也许会不一样. 沿着这条分析主线的一个非常重要的结果应归于 Nesterov. 考虑

$$\begin{aligned}
&\text{maximize} \quad \sum_{i, j} q_{ij} x_i x_j \\
&\text{subject to} \quad x_j^2 = 1, \quad j = 1, 2, \dots, n
\end{aligned}$$

其中 $Q \in \mathcal{S}^n_+$. 它的 SDP 松弛为

$$\begin{aligned}
&\text{maximize} \quad \mathbf{Q} \cdot \mathbf{X} \\
&\text{subject to} \quad \mathbf{X} \geq \mathbf{0}, \quad x_{ii} = 1, \quad i = 1, 2, \dots, n
\end{aligned}$$

假设 \mathbf{X}^* 是 SDP 松弛问题的最优解. 那么,通过设定 $\xi \in \mathcal{N}(0, \mathbf{X}^*)$ 和 $\mathbf{x}(\xi) := \text{sign}(\xi)$,可产生一个期望值为

$$E[\mathbf{x}(\xi)^T Q \mathbf{x}(\xi)] = \mathbf{Q} \cdot \frac{2}{\pi} \arcsin \mathbf{X}^*$$

的随机解,其中 $\arcsin \mathbf{X}^*$ 是作用于矩阵元素的函数,即 $(\arcsin \mathbf{X}^*)_{ij} := \arcsin x_{ij}^*$. 注意:

$$\arcsin x = x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \dots$$

和两个正定矩阵的 Hadamard 乘积(逐分量乘积)仍然是半正定的,因此有

$$\arcsin \mathbf{X}^* \geq \mathbf{X}^*$$

这意味着解的期望是

$$\begin{aligned} \frac{2}{\pi} \mathbf{Q} \cdot \arcsin \mathbf{X}^* &\geq \frac{2}{\pi} \mathbf{Q} \cdot \mathbf{X}^* \\ &\geq \frac{2}{\pi} \times \text{二次型的最优值} \end{aligned}$$

用这种方法可以得到逼近因子为

$$\frac{2}{\pi} \approx 0.636\ 62 < \alpha \approx 0.878\ 56$$

的解.

参考文献

- [1] 陈宝林. 最优化理论与算法[M]. 2 版. 北京: 清华大学出版社, 2005.
- [2] 黄红选, 韩继业. 数学规划[M]. 北京: 清华大学出版社, 2006.
- [3] 张建中, 许绍吉. 线性规划[M]. 北京: 科学出版社, 1997.
- [4] 袁亚湘. 非线性优化计算方法[M]. 北京: 科学出版社, 2007.
- [5] 邢文训, 谢金星. 现代优化计算方法[M]. 2 版. 北京: 清华大学出版社, 2005.
- [6] J W Demmel. 应用数值线性代数[M]. 王国荣, 译. 北京: 人民邮电出版社, 2007.
- [7] G Dantzig. Linear Programming and Extensions[M]. Princeton, N J: Princeton University Press, 1963.
- [8] L Kantorovich. Mathematical methods in the organization and planning of production[J]. Management Science, 1960, 6:550-559.
- [9] D Gale, H Kuhn, A Tucker. Activity analysis of production and allocation. Linear Programming and the Theory of Games[C]//T Koopmans. New York: John Wiley and Sons, 1951:317-329.
- [10] V Klee, G Minty. How good is the simplex algorithm? [C]//O Shisha. Inequalities-III. New York: Academic Press, 1972:159-175.
- [11] L G Khachiyan. A polynomial algorithm in linear programming[J]. Soviet Mathematics Doklady, 1979, 20:191-194.
- [12] N Karmarkar. A new polynomial-time algorithm for linear programming[J]. Combinatorics, 1984, 4:373-395.
- [13] S Boyd, L Vandenberghe. Convex Optimization[M]. Cambridge: Cambridge University Press, 2004.
- [14] R Fletcher. Practical Methods of Optimization[M]. 2nd ed. New York: John Wiley & Sons, 1987.
- [15] H W Kuhn, A W Tucker. Nonlinear programming[C]//J Neyman. Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, C A: University of California Press, 1951:481 -492.
- [16] D G Luenberger. Optimization in Vector Space[M]. New York: John Wiley & Sons, Inc., 1969.
- [17] D G Luenberger. Linear and Nonlinear Programming[M]. 2nd ed. Massachusetts: Addison-Wesley Inc. , 1984.
- [18] J Nocedal, S J Wright. Numerical Optimization[M]. 2nd ed. New York: Springer, 2006.
- [19] R J Vanderbei. Linear programming: foundations and extensions[M]. Boston: Kluwer Academic Publishers, 2001.
- [20] R T Rockafellar. Lagrange multipliers and optimality[J]. SIAM Review, 1993,

- 35:183-238.
- [21] Y Nesterov, A Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*[M]. Philadelphia:SIAM, 1994.
- [22] M X Goemans, D P Williamson. Improved approximation algorithms for maximum cut and satisability problems using semidenite programming[J]. *Journal of the ACM*, 1995, 42(6):1115-1145.
- [23] S P Boyd, L E Ghaoui, E Feron, et al. *Linear Matrix Inequality in Systems and Control Theory*, SIAM Frontier Series[M]. Philadelphia:SIAM, 1995.
- [24] G H Golub, C F Van Loan. *Matrix Computations*[M]. 3rd ed. Baltimore: The Johns Hopkins University Press,1996.
- [25] P Gahinet, A Nemirovski, A J Laub, et al. *LMI Control Toolbox*[M]. Boston: The Mathworks Inc. ,1995.
- [26] W W Hager. Updating the inverse of a matrix[J]. *SIAM Review*, 1989, 31(2): 221-239.
- [27] J Barzilai, J Borwein. Two point step size gradient methods[J]. *IMA Journal of Numerical Analysis*,1988,8:141-148.
- [28] E Birgin, J Martinez, M Raydan. Nonmonotone spectral projected gradient methods on convex sets[J]. *SIAM Journal on Optimization*, 2000,10:1196-1211.
- [29] Y Yuan. On the truncated conjugate-gradient method[J]. *Mathematical Programming*, 2000,87:561-573.
- [30] T G Kolda, R M Lewis, V Torczon. Optimization by direct search:new perspectives on some classical and modern methods[J]. *SIAM Review*, 2003,45(3):385-482.
- [31] N V Sahinidis, M Tawarmalani. *BARON 9. 0. 4: Global Optimization of Mixed-Integer Nonlinear Programs*. 2010.
- [32] P E Gill, W Murray. Modification of matrix factorizations after a rankone change [M]//A H Jacobs. *The State of the Art in Numerical Analysis*. London: Academic Press,1978.
- [33] P E Gill, W Murray. Numerically stable methods for quadratic programming[J]. *Mathematical Programming*, 1978,14:349-372.
- [34] P E Gill, W Murray, M A Saunders, et al. Sparse matrix methods in optimization[J]. *SIAM J. Sci. Stat. Comp.* ,1984,5:562-589.
- [35] J B Rosen. The gradient projection method for nonlinear programming, Part I: Linear constraints[J]. *Journal on SIAM*, 1960,8:181-217.
- [36] R Courant. Variational methods for the solution of problems with equilibrium and vibration[J]. *Bull. Amer. Math. Soc.* , 1943,49:1-23.
- [37] K R Frisch. The logarithmic potential method of convex programming, Technical Report[R]. Oslo: University Institute of Economics,1955.
- [38] C W Carroll. The created response surface technique for optimizing nonlinear restrained systems[J]. *Operations Research*. 1961,9:169-184.

-
- [39] A V Fiacco, G P McCormick. Nonlinear Programming: Sequential Unconstrained Minimization Techniques[M]. New York, N Y: John Wiley & Sons, 1968.
 - [40] M J D Powell. A method for nonlinear constraints in minimization problems[M]// R Fletcher. Optimization. New York, N Y: Academic Press, 1969: 283-298.
 - [41] M R Hestenes. Multiplier and gradient methods[J]. Journal of Optimization Theory and Applications, 1969, 4: 303-320.
 - [42] L McLinden. An analogue of Moreau's proximation theorem, with applications to the nonlinear complementarity problem[J]. Pacific Journal of Mathematics, 1980, 88: 101-161.
 - [43] I I Dikin. Iterative solution of problems of linear and quadratic programming[J]. Soviet Mathematics Doklady, 1967, 8: 674-675.
 - [44] N Megiddo. Pathways to the optimal set in linear programming[C]// N Megiddo. Progress in Mathematical Programming: Interior-Point and Related Methods. New York: Springer-Verlag, 1988: 131-158.
 - [45] M J Todd. Potential reduction methods in mathematical programming[J]. Mathematical Programming, Series B, 1997, 76: 3-45.
 - [46] S Mehrotra. On the implementation of a primal-dual interior point method[J]. SIAM Journal on Optimization, 1992, 2: 575-601.
 - [47] R D C Monteiro, I Adler. Interior path following primal-dual algorithms. Part I: linear programming[J]. Mathematical Programming, 1989, 44: 27-41.
 - [48] R D C Monteiro, I Adler. Interior path following primal-dual algorithms. Part II: convex quadratic programming[J]. Mathematical Programming, 1989, 44: 43-66.
 - [49] M H Wright. Interior methods for constrained optimization[J]. Acta Numerica, 1992, 1: 341-407.
 - [50] Y Ye. Interior Point Algorithms: Theory and Analysis. Wiley-Interscience Series in Discrete Mathematics and Optimization[M]. New York: John Wiley Sons, Inc., 1997.
 - [51] M H Wright. The interior-point revolution in optimization: history, recent developments, and lasting consequences[J]. Bulletin of the American Mathematical Society, 2004, 42(1): 39-56.
 - [52] E D Andersen, J Gondzio, C Mészáros, et al. Implementation of interior-point methods for large scale linear programming[C]// T Terlaky. Interior Point Methods in Mathematical Programming. Kluwer: Springer, 1996: 189-252.

索引^{*}

一 画

一阶最优性条件 first-order optimality conditions, 见 KKT 条件, 4.1

无约束优化的 unconstrained optimization, 4.1

稳定点 stationary point, 4.1

一维搜索 linear search, 4.2

精确 exact, 4.2

非精确 inexact, 4.2, 4.3

可接受点 acceptable point, 4.3

Armijo 法则 Armijo rule, 4.3

回溯 backtracking, 4.3, 5.1, 9.4

Goldstein 测试 Goldstein test, 4.3

Wolfe 条件 Wolfe condition, 4.3, 5.3

强 Wolfe 条件 strong Wolfe condition, 4.3, 5.3

强 Wolfe 一维搜索 strong Wolfe linear search, 4.3

划界阶段 bracketing phase, 4.4

覆盖 bracket, 4.4

分割阶段 secting phase, 4.4

插值 interpolation, 4.4

恰当覆盖 right bracket, 4.4, 8.4

黄金分割法 golden section search, 4.4

Fibonacci 分割法 Fibonacci section search, 4.4

二 画

二阶最优性条件 second-order optimality conditions, 7.4

必要的 necessary, 7.4

充分的 sufficient, 7.4

无约束优化 unconstrained optimization, 4.1

二次规划 quadratic programming, 8.1, 8.2

积极集 active set, 8.2

* 名词术语后面的数字为相应章节。

积极集法 active set methods, 8.2

原始的 primal, 8.2

对偶的 dual, 8.2

原始-对偶的 primal-dual, 8.2

阻滞约束 blocking constraint, 8.2

凸的 convex, 8.2

循环 cycling, 8.2

广义消元法 generalized elimination method, 8.1

正交分解法 orthogonal factorization method, 8.1

零空间法 null-space method, 8.1

最优积极集 optimal active set, 8.1

最优化条件 optimality conditions, 8.1

第 I 阶段 phase I, 8.2

值空间法 range-space method, 8.1

终止性 termination, 8.2

更新分解 updating factorizations, 8.2

三 画

大范围收敛 global convergence, 4.2

广义逆 general inverse, 5.4, 7.2

四 画

凸性 convexity, 2.1

函数的 of functions, 4.1

严格 strict, 4.1

集合的 of sets, 2.1

凸锥 convex cone, 2.1

超平面 hyperplane, 2.1

正(负)闭半空间 positive(negative) closed half spaces, 2.1

多面集 polytope, 2.1

多面锥 polyhedral cone, 2.3, 7.3

极点 extreme point, 2.1

凸规划 convex programming, 1.2

隐 hidden, 6.2

分枝定界法 branch-and-bound method, 3.4

枚举法 enumeration method, 3.4

- 定界 bounding, 3.4
 剪枝 pruning, 3.4
 广度优先法 breadth-first search, 3.4
 当前最好解 best-so-far, 3.4
 枚举树 enumeration tree, 3.4
 深度优先法 depth-first search, 3.4
 分枝割算法 branch-and-cut algorithm, 3.4
 牛顿法 Newton method, 5.1
 用于乘子罚函数 for multiplier penalty function, 9.2
 用于对数障碍函数 for log-barrier function, 9.5
 单变量的 in one variable, 6.2, A.8
 截断 truncated, 5.5
 修正的 modified, 5.1
 增加 I 的倍数 adding a multiple of the unit matrix, 5.1
 特征值修正 eigenvalue modification, 5.1
 收敛速度 rate of convergence, 5.1
 水平集 level set, 4.3, 7.5
 互补性 complementarity, 2.3, 7.2, 7.6, 7.8
 互补条件 complementarity condition, 7.2
 严格的 strict, 7.2
 内点法 interior-point method, 2.2, 8.6, 9.1, 9.5
 中心路径 central path, 9.5
 路径跟踪算法 path-following algorithm, 9.5
 贴近性度量 proximity measure, 9.5

五 画

- 可行域 feasible region, 7.0
 原始 primal, 7.0
 原始-对偶 primal-dual, 8.6
 严格 strictly, 9.5
 可行序列 feasible sequences, 7.3
 极限方向 limiting directions of, 7.3
 可行方向 feasible direction, 7.3
 序列 sequential, 7.3
 线性化 linearized, 7.3
 可行方向法 feasible direction method, 7.1
 对偶 dual, 2.3, 3.3, 7.7, 7.8, 9.2
 对偶变量 dual variables, 3.3, 7.7

对偶函数 dual function, 3.3, 7.7
 对偶问题 dual problem, 2.3, 7.7, 7.8, 9.2
 对偶间隙 dual gap, 2.3, 3.3, 7.7
 原始函数 primal function, 7.6
 扰动问题 perturbation problem, 7.6
 对偶性 duality, 2.3, 3.3, 7.7, 9.5, 7.8
 线性规划 linear programming, 2.3, 9.5
 整数规划 integer programming, 3.3
 约束优化 constrained optimization, 7.7
 半定规划 SDP, 7.8
 单纯形乘子 simplex multiplier, 2.3
 边际价格 marginal price, 2.3
 互补性 complementarity, 2.3 7.1 7.7 7.8
 对偶单纯形法 dual simplex method, 2.3, 3.4
 正则性假定 regularity assumption, 7.3, 7.4
 半定规划 semidefinite programming, 7.9

六 画

优化 optimization, 1.2, 5.1, 7.1, 9.3, 9.4
 无约束 unconstrained, 1.2, 5.1
 约束 constrained, 1.2, 7.1
 连续 continuous, 1.2
 离散 discrete, 1.2
 光滑 smooth, 1.2
 非光滑 non-smooth, 1.2, 9.3, 9.4
 约束 constraints, 2.1, 7.1, 7.4, 7.6, 8.6, 8.2, 9.2, 9.3
 等式 equality, 7.0
 不等式 inequality, 7.0
 界 bounds, 7.7, 8.6, 9.2, 9.3
 法向量 normal vector, 2.1, 7.1
 积极(紧) active (binding), 7.1, 8.2
 严格(强) strictly (strongly), 7.4
 全局极小点 global minimizer, 1.2, 1.3, 4.1, 7.2, 7.4
 全局解 global solution, 见全局极小点
 全局优化 global optimization, 4.3
 网络 network, 3.1
 有向图 digraph, 3.1
 子网络 subnetwork, 3.1

- 生成树 spanning tree, 3.1
 二部图 bipartite graphs, 3.2
 网络单纯形法 network simplex method, 3.1
 树解 tree solution, 3.1
 根节点 root node, 3.1
 整性定理 integrality theorem, 3.1
 收敛速度 convergence, rate of, 4.2
 收敛阶 order of convergence, 4.2
 二次终止性 quadratic termination, 4.2
 线性 linear, 4.2, 5.1
 速率常数 rate constant, 4.2
 二次 quadratic, 4.2, 5.1
 次线性 sublinear, 4.2
 超线性 superlinear, 4.2
 共轭 conjugacy, 5.2
 共轭方向法 conjugate direction method, 5.2
 扩展子空间极小化 expanding subspace minimization, 5.2
 终止 termination of, 5.2
 共轭梯度法 conjugate gradient method, 5.2
 聚类 clustering, 5.2
 条件数 condition number, 5.2
 预条件的 preconditioned, 5.2
 收敛速度 rate of convergence, 5.2
 Krylov 子空间 Krylov subspace, 5.2
 预条件子 preconditioner, 5.2
 重新开始 restart, 5.2
 有限差分 finite difference, 4.2, 5.3, 8.3
 中心差分公式 central-differencing formula, 4.2
 向前差分公式 forward-difference formula, 4.2
 削度逼近 gradient approximation, 4.2
 Hessian 阵逼近 Hessian approximation, 5.3
 约束优化 constrained optimization, 1.2, 7.1
 线性 linear, 7.1, 8.1~8.5
 非线性 nonlinear, 9.1~9.4
 约束规范 constrain qualification, 7.3
 线性无关 linear independence (LICQ), 7.3
 线性 linear (LCQ), 7.3
 Slater, 7.6, 7.8, 7.9
 价值函数 merit function 9.4

ℓ_1 价值函数与参数选取 ℓ_1 merit function and parameter choosing, 9.3, 9.4

ℓ_2 价值函数与参数选取 ℓ_2 merit function and parameter choosing, 9.4

精确 exact, 9.3

定义 definition of, 见 罚函数, 9.1~9.4

非光滑性 nonsmoothness of, 9.3

用于可行方法的 for feasible methods, 9.4

用于 SQP 的 for SQP, 9.4

七 画

运筹学 operational research, 1.0

灵敏度 sensitivity, 2.3, 7.2, 7.6

灵敏度分析 sensitivity analysis, 1.0, 7.2

局部极小点 local minimizer, 1.2, 1.3, 4.1, 7.2, 7.4

局部解 local solution, 见 局部极小点

局部收敛 local convergence, 4.2

运输问题 transportation problem, 3.2

Hitchcock 运输问题 Hitchcock transportation problem, 3.2

指派问题 assignment problem, 3.2

拟牛顿逼近 Hessian 阵 quasi-Newton approximate Hessian, 5.3, 9.4

拟牛顿法 quasi-Newton method, 5.3

割线方程 secant equation, 5.3

曲率条件 curvature condition, 5.3

Broyden 族 Broyden family, 5.3

大范围收敛 global convergence, 5.3

收敛速率 rate of convergence, 5.3

对称秩一法 Symmetric-Rank-one (SR1) method, 5.3

DFP 法 DFP method, 5.3

BFGS 法 BFGS method, 5.3

性质 properties, 5.3

序列极小化技术 sequential minimization technique, 9.0

八 画

规划 programming, 1.0, 1.2, 2.1, 2.2, 2.3, 3.1, 3.3

数学 mathematical, 1.0

线性 linear, 1.2, 2.1, 2.2, 2.3, 3.1

非线性 nonlinear programming, 1.2

- 整数线性 integer linear, 1.2, 3.3
 混合整数 mixed integer, 1.2, 3.3
 凸 convex, 1.2, 1.3, 7.4-7.8
 函数 function, 1.4, 4.4, 5.1, 8.1, 8.3, 9.3, 9.4
 等高线 contour, 1.1
 方向导数 directional derivative, 1.4, 9.3
 梯度 gradient, 1.4
 既约 reduced, 8.1, 8.3
 Hessian 阵 Hessian matrix, 1.4
 既约 reduced, 8.1, 8.3
 斜率 slope, 1.4
 曲率 curvature, 1.4
 (扩展的)Rosenbrock (extended) Rosenbrock, 1.4, 4.4, 5.1, 9.4
 线性 linear, 1.4
 二次 quadratic function, 1.4
 范数 norm, 1.4
 线性规划 linear programming, 1.2, 2.1, 3.1, 9.5
 标准形 standard form, 2.1
 松弛/盈余变量 slack/surplus variables, 2.1
 自由变量 free variables, 2.1
 基本可行解 basic feasible solution, 2.1
 基变量 basic variables, 2.1
 基矩阵 basis matrix, 2.1
 对偶可行基本解 dual feasible basic solution, 2.3
 人工变量 artificial variables, 2.2
 对偶问题 dual problem, 3.3
 可行多面集 feasible polytope, 3.1
 极点 extreme point, 2.1
 基本定理 fundamental theorem, 2.1
 线性矩阵不等式 linear matrix inequality, 7.10
 单纯形法 simplex method, 2.2, 3.1
 既约线性规划 reduced linear programming, 2.2
 基本指标集 basic index set, 2.2
 复杂度 complexity of, 2.2
 循环 cycling, 2.2
 避免 avoidance of, 2.2
 Bland 法则 Bland rule, 2.2
 退化步 degenerate steps, 2.2
 基变量 basic variables, 2.1

- 一次迭代的描述 describing of one iteration, 2.1
 规范形 canonical form, 2.2
 既约(相对)费用系数 reduced (relative) cost coefficients, 2.2
 转轴元 pivot element, 2.2
 单纯形表 simplex tableau, 2.2
 发现 discovery of, 2.1
 进基指标 entering index, 2.2
 离基指标 leaving index, 2.2
 有限终止 finite termination of, 2.2
 初始化 initialization, 2.2
 两阶段法 two-phase method, 2.2
 修正单纯形法 revised simplex method, 2.2
- 图 graph, 3.1
 线搜索法 line search method, 4.2
 搜索方向 search direction, 4.2
 下降性 descent property, 4.2
 下降方向 descent direction, 4.2
 一维搜索 line search, 4.2, 4.4
 夹角条件 angle criterion, 4.3
 大范围收敛 global convergence of, 4.3

九 画

- 转运问题 transshipment problem, 3.2
 源(供给)节点 source (supply) nodes, 3.2
 宿(需求)节点 sink (demand) nodes, 3.2
 信赖域 trust region, 6.1
 半径 radius, 6.1
 球形的 spherical, 6.1
 盒子形状的 box-shape, 6.1
 信赖域法 trust region method, 4.2, 5.4, 6.1~6.3
 与线搜索法的对比 contrast with line search method, 4.2, 5.1, 6.1
 大范围收敛 global convergence, 5.1
 局部收敛 local convergence, 5.1
 牛顿法的变形 Newton variant, 5.1, 6.1
 调整半径的策略 strategy for adjusting radius, 6.1
 真实下降量 actual reduction, 6.1
 预估下降量 predicted reduction, 6.1
 子问题 subproblem, 6.2

- 精确解 exact solution, 6.2
 难的情况 hard case, 6.2
 几乎精确解 nearly exact solution of, 6.2
 近似解 approximate solution of, 6.2, 6.3
 柯西点 Cauchy point, 6.2
 计算 calculation of, 6.3
 在大范围收敛中的作用 role in global convergence, 6.3
 折线法 dogleg method, 5.4, 6.2, 9.4
 折线轨道 dogleg trajectory, 6.2
 LM 轨道 Levenberg-Marquardt trajectory, 6.2
 罚函数 penalty function, 见价值函数, 9.0 9.1
 约束违反度 constraint violation, 9.4
 精确的 exact, 9.0 9.3
 ℓ_1 , 9.3
 Courant (二次) Courant (quadratic), 9.1
 困难 difficulty of minimizing, 9.1
 Hessian 阵 Hessian of, 9.1
 乘子 multiplier, 9.2
 外部 exterior, 9.1
 捷径法 shortcut method, 9.1
 罚参数 penalty parameter, 9.1~9.4
 逐步二次规划 sequential quadratic programming, 9.0, 9.4
 推导 derivation, 9.4
 子问题 subproblem, 9.4
 最小二乘乘子 least-squares multipliers, 9.4
 局部算法 local algorithm, 9.4
 QP 乘子 QP multipliers, 9.4
 收敛速度 rate of convergence, 9.4
 $S\ell_1QP$ 法 S.1QP method, 9.4
 线搜索算法 linear search algorithm, 9.4
 信赖域算法 trust-region algorithm, 9.4

十 画

- 消元 elimination, 2.1, 8.1, 8.3
 乘子法 method of multipliers, 9.2
 动机 motivation, 9.2
 框架 framework for, 9.2
 不等式约束 inequality constraints, 9.2

十一画

混合整数规划 mixed integer programming problems, 3. 4

梯度投影法 gradient projection method, 8. 4, 8. 6, 9. 2

十二画

博弈论 game theory, 1. 1, 2. 4

收益矩阵 payoff matrix, 1. 1

链式法则 chain rule, 1. 4, 7. 3, 8. 3

割平面 cutting plane, 3. 4

最小费用网络流问题 minimum-cost network flow problem, 3. 0

流平衡约束 flow conservation constraint, 3. 1

点弧关联矩阵 node arc incidence matrix, 3. 1

流 flows, 3. 1

原始 primal, 3. 1

平衡 balanced, 3. 1

可行 feasible, 3. 1

最大流问题 maximum-flow problem, 3. 2

流的值, value of the flow, 3. 2

割 cut, 3. 2

割集 cut-set, 3. 2

最大流最小割定理 max-flow min-cut theorem, 3. 2

最短路问题 shortest path problem, 3. 2

标号 label, 3. 2

值函数 value function, 3. 2

动态规划原理 principle of dynamic programming, 3. 2

Bellman 方程 Bellman equation, 3. 2

逐次近似法 method of successive approximations, 3. 2

标号校正算法 label-correcting algorithm, 3. 2

标号设置算法 label-setting algorithm, 3. 2

Dijkstra 算法 Dijkstra algorithm, 3. 2

最速下降方向 steepest descent direction, 5. 1

最速下降法 steepest descent method, 5. 1

收敛速度 rate of convergence, 5. 1

最小二乘问题 least-square problems, 5. 4

线性 linear, 5. 4

正规方程组 normal equations, 5. 4

- 通过 QR 分解求解 solution via QR factorization, 5.4
非线性 nonlinear, 5.4
应用 applications of, 1.1, 5.4
小残量问题 small-residual problems, 5.4
LM 法 Levenberg-Marquardt method, 见 LM 法, 5.4
统计解释 statistical justification, 5.4
目标函数的结构 structue of objective function, 5.4

十三画

- 锯齿形 zigzagging, 5.1, 8.0, 8.5
障碍函数 barrier functions, 9.1
障碍法 barrier methods, 9.1
障碍参数 barrier parameter, 9.1
滤子 filter, 9.4

十四画

- 算法的复杂度 complexity of algorithms, 2.2
多项式时间 polynomial-time, 2.2
指数时间 exponential-time, 2.2

十五画

- 增广 Lagrange 函数 augmented Lagrangian function, 9.2

十六画

- 鞍点 saddle point, 4.1, 7.6
整数线性规划 integer linear programming, 3.0, 3.3
线性规划松弛 linear programming relaxation, 3.2, 3.4
集合分割问题 set-partitioning problem, 3.3
集合覆盖问题 set-covering problem, 3.3
旅行商问题 traveling salesman problem, 3.3

其 他

- Cholesky 分解 Cholesky factorization, 4.1, 5.1, 5.3, 5.4, 6.2, 8.1

-
- GN 法 Gauss-Newton method, 5.4
 - 与线性最小二乘的联系 connection to linear least squares, 5.4
 - 线搜索 line search in, 5.4
 - Gomory 割平面法 Gomory cutting plane method, 3.4
 - Karush-Kuhn-Tucker 条件 Karush-Kuhn-Tucker condition, 6.2, 7.2, 8.1, 9.4
 - Lagrange 乘子 Lagrangian multipliers, 7.1, 7.2
 - 估计 estimates of, 8.4, 9.1, 9.2, 9.4
 - 协态变量 co-state variable, 7.10
 - Lagrange 函数 Lagrangian function, 7.1
 - 约束优化的 for constrained optimization, 7.1, 7.5
 - Hessian 阵 Hessian of, 7.4, 9.4
 - 投影 Hessian 阵 projected Hessian of, 9.4
 - Lagrange 矩阵 Lagrangian matrix, 8.1
 - LM 法 Levenberg-Marquardt method, 5.4
 - 作为信赖域法 as trust-region method, 5.4
 - Maratos 效应 Maratos effect, 9.0, 9.4
 - 例子 example of, 9.4
 - 放松试探步的条件 nonmonotone watchdog strategy, 9.4
 - 二阶校正步 second-order correction step, 9.4
 - Newton-Lagrange 法, 见逐步二次规划, 9.3
 - Sherman-Morrison 公式 Sherman-Morrison formula, 5.3
 - Taylor 级数 Taylor series, 1.4
 - Taylor 定理 Taylor theorem, 1.4, 4.1, 7.3, 7.4, 9.3
 - Taylor 多项式 Taylor polynomial, 1.4
 - 余项 remainder, 1.4