

高等学校研究生教材

实用最优化方法

(第三版)

唐焕文 秦学志 编著

大连理工大学出版社

图书在版编目(CIP)数据

实用最优化方法/唐焕文,秦学志编著. —3版. —大连:大连理工大学出版社,2004.1

ISBN 7-5611-0901-6

I. 实 … II. ①唐… ②秦… III. 最优化方法 IV. O224

中国版本图书馆 CIP 数据核字(2000)第 96254 号

大连理工大学出版社出版

地址:大连市凌水河 邮政编码:116024

电话:0411-4708842 传真:0411-4701466 邮购:0411-4707961

E-mail:dutp@mail.dlptt.ln.cn URL:http://www.dutp.cn

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:140mm×203mm 印张:13.75 字数:344千字

印数:7 001~10 000

1994年11月第1版

2004年1月第3版

2004年1月第4次印刷

责任编辑:王 纪

责任校对:杨 莉

封面设计:孙宝福

定 价:19.80元

本书由

国家自然科学基金(10071010,90103033)

资助出版

The published book is financed by

**National Natural Science Foundation of
China, Grant No. 10071010,90103033**

第三版前言

本书第二版出版之后,深受读者欢迎,被 30 多所高校选用作教材或主要参考书,重印之后也很快销售一空。为满足读者需要,我们根据最近几年在教学中发现的问题和科研实践,对原书进行了修订和增补,主要变动如下:

1. 对第 4、5、8 章的内容进行了适当增补。

极大熵方法是求解一类非光滑优化和多约束优化问题的有效方法,因此我们增写了 4.9 节,对极大熵方法做了简要介绍。

目标规划是一种常用的有效的多目标优化方法,在处理某些多目标规划问题时,有明显的优点,因此,增写了 5.4 节,介绍目标规划的概念、模型及解法。

进化算法是近十多年来很受重视的一类随机优化方法,趁这次新版之机,我们对第 8 章做了较大的增补,增写了 8.4 和 8.5 节,较详细地介绍了进化规划和模拟退火方法及其在蛋白质空间结构预测中的应用。

2. 对原书中已发现的错漏和不当之处进行了订正,增补了部分习题和答案,并删去了一些较次要的内容。

本书修订和增补工作分工如下:

唐焕文:第 1~5、8 章;秦学志:第 6、7 章;第 9 章由唐焕文(9.1~9.3、9.5 节)、秦学志(9.2 节)和郭崇慧(9.1、9.4 节)合作修订。

中科院应用数学研究所赖炎连研究员和大连理工大学应用数学系冯恩民教授对本书第二版给予了充分的肯定,并提出了很好

的改进建议。庞丽萍老师多次使用本书作为研究生教材,并提出了很好的修改意见。研究生计明军、赵晶、李斌为本书的修订做了大量细致的工作。对上面提到的专家和各位同志,特在此表示衷心的感谢。

本书得到国家自然科学基金(10071010,90103033)资助,特此致谢。

本书虽经几次修订,但错漏和不当之处恐怕仍会存在,敬请读者批评指正。

唐焕文

2004 年 1 月

第二版前言

呈现在读者面前的这本《实用最优化方法》是《最优化方法》一书的修订本。原书在 1994 年 11 月出版之后,受到读者的欢迎,被 20 多所兄弟院校选用作为教材或主要参考书,到 1999 年 9 月已销售一空,仍有不少读者来信、来电求购。为满足教学需要和读者要求,在大连理工大学研究生院和出版社领导的大力支持下,我们在较短的时间内,完成了对原书的修订和增补工作。主要变动如下:

1. 增写了第 8 章和第 9 章。

进化算法是近十年来备受重视的一类优化方法,具有广阔的应用前景。为满足读者的学习需要,我们在第八章中对此做了简要介绍。

在教学过程中,常有学生要求介绍优化方法如何应用。在这一版中,我们从自己的科研实践中,选择了五个例子,增写了第 9 章,介绍优化方法的具体应用,供读者参考。

2. 对原书中已发现的错漏和不当之处进行了订正,并增补了部分习题和答案。

如果使用本书作为教材,我们建议主要讲授前 8 章。预计 50 学时可讲完前 5 章的大部分内容及第八章,70 学时可讲完前 8 章的绝大部分内容;第 9 章可作简要介绍,主要供学生自学、参考。

本书修订和增补分工如下:

唐焕文:第 1~4、5、8 八章;秦学志:第 6、7 章;第 9 章由唐焕文(9.2、9.3、9.5 节)、秦学志(9.2 节)和郭崇慧(9.1、9.4 节)合作

编写。

感谢编者的许多研究生对本书出版所给予的支持和帮助。

由于时间匆促,本书难免会有错漏和不当之处,请广大读者批评指正。

唐焕文

1999 年 12 月

第一版前言

最优化方法是运筹学的一个重要的组成部分,在自然科学、社会科学、生产实际、工程设计和现代化管理中有着重要的实用价值,因此,在最近的 40 多年中得到了十分迅速的发展和广泛的应用。

本书是根据高校工科研究生数学课程教学指导小组审订通过的《工学硕士研究生最优化方法课程教学基本要求》,在对原编著《实用数学规划导论》一书进行修改补充后写成的。主要对象是理工院校的工科研究生。考虑到应用数学、计算数学、运筹学等应用理科专业和管理工程、系统工程等专业本科生的教学需要以及从事运筹优化应用的工程技术人员、管理人员的需要,本书还增写了部分超出基本要求的内容(用 * 标记者),供有关专业选用。全书共分七章,比较系统地介绍了线性规划、无约束优化方法、约束优化方法、多目标规划、整数规划和动态规划的数学模型、基本概念、基本理论和计算方法。期望通过本书的学习,能使读者较好地理解定量优化的思想,掌握一些基本而常用的优化方法,并能运用优化的观点和方法分析解决实践中经常遇到的一些较典型的优化问题。

在编写本书时,我们还着重考虑了以下几点:

1. 努力做到深入浅出,通俗易懂,适于教学和自学,使具有微积分和线性代数知识的广大读者都能读懂本书的大部分内容。为了帮助大家学习,在附录中列出了阅读本书需要用到的主要数学基础知识,供读者复习和查阅。但是,为了叙述的完整,本书中也含有少部分较难的内容,需要用到较深的数学知识或较复杂的数学推导,有困难的读者初学时,可以略去。

2. 正如在《自然杂志》1981 年举行的应用数学座谈会上许多同志所指出的那样:目前在我国,适合工科师生和工程技术人员阅读的应用数学书籍还比较少,有的书,纯数学的味道太浓,搞应用的人学不下去,学了也不会应用;有的书,又写得太浅,不够用。本书企图弥补这方面的不足,考虑到从事应用的读者的需要,着重介绍最优化的基本原理和在实际应用中比较有效的计算方法。为了便于理解和应用,对选入的每种算法,尽可能把它的基本思想阐述清楚,然后阐明其理论依据,并给出具体的计算步骤、程序框图和数值计算例子。在选择算法时,我们遵循下面的原则:第一,所选算法便于在计算机上实现;第二,在应用中比较有效的重要算法都尽可能选入,并注意选入一些较新的重要方法。

3. 要掌握最优化方法,需要有一些数值计算及应用的例题帮助理解;还需要独立完成一定数量的习题,其中部分习题要上机计算。为此,我们在每一章都配有一定数量的例题和较多的习题,供大家阅读和练习,部分习题附有提示或答案。

在编者从事运筹、优化的学习、教学和科研工作中,经常得到中科院院士、著名力学家、工程优化专家钱令希教授、著名运筹优化专家越民义、桂湘云、吴方、韩继业教授和著名计算数学家徐利治教授等的大力支持和热情指导,《实用数学规划导论》一书就是在他们的热情支持和帮助下编写和出版的。韩继业教授还亲自在北京大学概率统计专业使用过该书。本书的修改得到上述各位先生和大连理工大学研究生院、出版社领导的关心与支持,特在此表示衷心的感谢。

如果使用本书作为工科研究生教材,预计 50 学时左右可讲完前五章,70 学时可讲完全书。

本书由唐焕文(前 5 章)和秦学志(后两章)合作编写,全书由唐焕文修改定稿。限于编者水平,书中难免存在不当之处,请广大读者批评指正。

唐焕文

1994 年 5 月

目 录

第 1 章 引 论	1
1.1 最优化问题举例	1
1.2 最优化的基本概念	7
1.3 凸集和凸函数.....	14
习题一	27
第 2 章 线性规划	31
2.1 引言 线性规划的标准形式.....	31
2.2 线性规划的基本定理.....	36
2.3 单纯形法.....	41
2.4 关于单纯形法的说明和补充.....	55
2.5 线性规划的对偶理论与对偶单纯形法.....	71
* 2.6 线性规划的多项式算法	80
习题二	87
第 3 章 无约束优化方法	94
3.1 引言 下降递推算法.....	94
3.2 一维搜索.....	98
3.3 求多变量函数极值的基本下降法	112
3.4 共轭方向法和共轭梯度法	119
3.5 变尺度法	129
3.6 直接搜索法	138

习题三	149
第 4 章 约束优化方法	153
4.1 引言 Kuhn-Tucker 条件	153
4.2 惩罚函数法	158
4.3 碰壁函数法	165
* 4.4 可行方向法	171
4.5 梯度投影法	180
* 4.6 既约梯度法	188
4.7 乘子法	196
4.8 二次逼近法	205
4.9 极大熵方法	218
习题四	227
第 5 章 多目标规划	236
5.1 多目标规划问题举例	236
5.2 多目标规划问题的解集和像集	240
5.3 处理多目标规划问题的一些方法	244
5.4 目标规划	256
习题五	266
* 第 6 章 整数规划	270
6.1 整数规划问题举例	270
6.2 整数线性规划的解法概述	276
6.3 分枝定界法	280
6.4 割平面法	286
6.5 隐枚举法	291
6.6 匈牙利法	296

习题六.....	300
第 7 章 动态规划.....	304
7.1 多阶段决策问题	304
7.2 动态规划的基本原理	306
7.3 函数空间迭代法和策略空间迭代法	314
7.4 应用举例	322
习题七.....	329
第 8 章 进化算法.....	333
8.1 引言 进化算法概述	333
8.2 进化策略	337
8.3 遗传算法	344
8.4 进化规划	350
8.5 模拟退火方法	354
习题八.....	363
第 9 章 优化方法应用举例.....	365
9.1 经济预测中的人工神经网络方法	365
9.2 稠油油层热物性参数的计算模型及算法	372
9.3 一类水火电系统的最优开发模型	380
9.4 干线公路网等级结构优化的目标规划模型及 算法	389
9.5 一个多目标动态投入产出优化模型及算法	397
部分习题答案或提示.....	405
参考文献.....	411
附录 最优化方法中常用的数学基础知识汇编.....	416

符号说明

$\min_{x \in \Omega} f(x)$ $f(x)$ 在 Ω 上的最小值

$\max_{x \in \Omega} f(x)$ $f(x)$ 在 Ω 上的最大值

s. t. 受约束于, 是“subject to”的缩写

$A \subset B$ 集合 A 包含于集合 B

$A \supset B$ 集合 A 包含集合 B

$x \in A$ x 属于集合 A , 即 x 是集合 A 的一个元素

$x \notin A$ x 不属于集合 A , 亦可写成 $x \notin A$

$A \cup B$ 集合 A 与 B 的并集

$A \cap B$ 集合 A 与 B 的交集

\approx 近似等于

\emptyset 空集合

$N(x_0, \epsilon)$ 或 $N_\epsilon(x_0)$ 以点 x_0 为中心, ϵ 为半径的邻域

$\binom{n}{k}$ 或 C_n^k 二项式系数, 即从 n 个元素中每次取出 k 个元素

所有不同的组合数

\triangleq 定义、恒等

\mathbf{R} 实数域

$[x]$ 不超过 x 的最大整数

$f(x) \in C$ $f(x)$ 是连续函数

$f(x) \in C^k$ $f(x)$ 具有 k 阶连续偏导数

\mathbf{R}^n n 维实欧氏空间

$f: D \subset \mathbf{R}^n \rightarrow \mathbf{R}$ $f(\mathbf{x})$ 是定义在 \mathbf{R}^n 中区域 D 上的实值函数

$\|\mathbf{x}\|$ 向量 \mathbf{x} 的欧氏范数, 即 $\|\mathbf{x}\| = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$

(\mathbf{x}, \mathbf{y}) 或 $\mathbf{x}^T \mathbf{y}$ 向量 \mathbf{x}, \mathbf{y} 的内积

$\det(A)$ 或 $|A|$ 矩阵 A 的行列式

$r(A)$ 矩阵 A 的秩

$\nabla f(\mathbf{x})$ $f(\mathbf{x})$ 的梯度, $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T$

$H(\mathbf{x})$ 或 $\nabla^2 f(\mathbf{x})$ $f(\mathbf{x})$ 的海赛阵, $H(\mathbf{x}) \triangleq \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right)_{n \times n}$

$\min\{x_1, x_2, \dots, x_n\}$ 数 x_1, x_2, \dots, x_n 中最小者

$\max\{x_1, x_2, \dots, x_n\}$ 数 x_1, x_2, \dots, x_n 中最大者

$\inf_{\mathbf{x} \in X} f(\mathbf{x})$ $f(\mathbf{x})$ 在 X 上的下确界

$\sup_{\mathbf{x} \in X} f(\mathbf{x})$ $f(\mathbf{x})$ 在 X 上的上确界

\mathbf{x}^* 最优解

f^* 目标函数的最优值

第 1 章 引 论

本章首先介绍如何从实际问题中抽象出最优化问题,然后介绍最优化的一些基本概念和有关的预备知识,为以后的学习打下基础。

1.1 最优化问题举例

利用最优化的理论和方法解决生产实际和自然科学中的具体问题,一般分为两个步骤:

(1) 建立数学模型。即对所要解决的具体问题进行分析研究,加以简化,形成最优化问题。

(2) 进行数学加工和求解。这个过程主要包括以下各项工作:将所得的最优化问题进行整理和变换,使之成为易于求解的形式;选择或提出解决该问题的适当的计算方法;编制计算程序并上机计算;分析计算结果,看其是否符合实际。

在这一节中,我们将通过几个简单的实例来说明怎样由实际问题经过抽象,建立它的数学模型。

【例 1.1】 某化工厂生产 A 、 B 、 C 、 D 四种化工产品,生产每种产品一吨所消耗的工时和产值如下表:

产 品	A	B	C	D
工 时 (小时)	100	300	400	75
产 值 (千元)	1	5	10	0.5

要求全厂年产值在 1000 万元以上,求当消耗的总工时最少时,该厂生产各种产品的数量。

解 设该厂全年生产 A 、 B 、 C 、 D 四种产品的数量分别为 x_1, x_2, x_3, x_4 (单位均为吨),消耗的总工时为 y ,则

$$y = 100x_1 + 300x_2 + 400x_3 + 75x_4$$

消耗的总工时 y 最少是我们所追求的目标,所以把 $y = f(x_1, x_2, x_3, x_4)$ 称为目标函数。而变量 x_1, x_2, x_3, x_4 的值是我们需要确定的,称 x_1, x_2, x_3, x_4 为决策变量或设计变量。

因为要求全厂的年产值在 1000 万元以上,所以决策变量 x_1, x_2, x_3, x_4 还需要满足以下的限制条件:

$$x_1 + 5x_2 + 10x_3 + 0.5x_4 \geq 10000 \quad (1)$$

$$x_i \geq 0, i = 1, 2, 3, 4 \quad (2)$$

条件(1)、(2)称为限制条件或约束条件。所以上述问题的数学模型可概述如下。

在约束条件(1)、(2)之下,确定变量 x_1, x_2, x_3, x_4 的数值,使目标函数 $y = f(x_1, x_2, x_3, x_4)$ 取得最小值。为叙述简便,以下文中常将此问题简写为

$$\begin{cases} \min & y = 100x_1 + 300x_2 + 400x_3 + 75x_4 \\ \text{s. t.} & x_1 + 5x_2 + 10x_3 + 0.5x_4 \geq 10000 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad (1-1)$$

其中 min 是 minimize 的缩写, s. t. 是 subject to 的缩写,中文意思是“受约束于”或“约束条件”。由于在最优化问题(1-1)中,目标函数 $y = f(x_1, x_2, x_3, x_4)$ 是变量 x_1, x_2, x_3, x_4 的线性函数,约束条件是 x_1, x_2, x_3, x_4 的线性不等式(或等式),所以把(1-1)称为线性规划问题,线性规划(Linear Programming)以后简记为 LP。 ■

【例 1.2】 运输问题

假设某种物资有 m 个产地, n 个销地。第 i 个产地的产量为 a_i ($i = 1, 2, \dots, m$); 第 j 个销地的需要量为 b_j ($j = 1, 2, \dots, n$), 其中

解 我们来建立上述问题的数学模型, 设由产地 i 运往销地 j 的货物数量为 x_{ij} , s 为运输的总吨公里数, 则上述问题可归结为如下的线性规划问题:

运输问题是线性规划中的一个典型问题,也是线性规划模型中最早被研究的一类问题。早在 1939 年,前苏联数学家康托洛维奇(Л. В. Канторович)在列宁格勒大学所作的题为“生产组织与计划中的数学方法”的报告中,就讨论了交通运输和生产计划中所提出的线性规划问题,并给出了一种解法——解乘数法。

设有 m 种资源: A_1, A_2, \dots, A_m , 拟生产 n 种产品: B_1, B_2, \dots, B_n 。用 a_{ij} 表示生产一个单位的第 j 种产品所需要第 i 种资源的数量, 用 b_i 表示第 i 种资源的最大数量, 用 c_j 表示第 j 种产品的单价, 用 x_j 表示第 j 种产品的生产数量。则 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 就代表一个生产计划, 我们的问题是: 要设法安排一个生产计划, 使每种产

品都完成或超额完成国家下达的产量计划, 即有 $x_j \geq e_j (j = 1, 2, \dots, n)$, 而使总产值最高, 其中 $e_j (j = 1, 2, \dots, n)$ 为国家下达的生产第 j 种产品的产量指标。

解 在这个问题中, 决策变量为 x_1, x_2, \dots, x_n 。设总产值为 y , 则上述问题可化为如下的线性规划问题:

$$\begin{cases} \max & y = \sum_{j=1}^n c_j x_j \\ \text{s. t.} & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m \\ & x_j \geq e_j, \quad j = 1, 2, \dots, n \end{cases} \quad (1-3)$$

采用向量、矩阵记号, 则(1-3)可写成

$$\begin{cases} \max & y = \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{A} \mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{e} \end{cases} \quad (1-3')$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\mathbf{c} = (c_1, c_2, \dots, c_n)^T$, $\mathbf{b} = (b_1, b_2, \dots, b_m)^T$, $\mathbf{e} = (e_1, e_2, \dots, e_n)^T$, $\mathbf{A} = (a_{ij})_{m \times n}$, $\mathbf{x} \geq \mathbf{e}$ 表示向量 \mathbf{x} 的每一个分量 x_j 都大于或等于向量 \mathbf{e} 的每一个相应的分量 e_j , T 表示向量或矩阵的转置。

【例 1.4】合理下料问题

设要用某类钢板下 m 种零件 A_1, A_2, \dots, A_m 的毛坯料。根据既省料又容易操作的原则, 人们在一块钢板上, 已设计出 n 种不同的下料方案, 设在第 j 种下料方案中, 可得到零件 A_i 的个数为 a_{ij} , 第 i 种零件的需要量为 $b_i (i = 1, 2, \dots, m)$ 。问应如何下料, 才能既满足需要, 又使所用钢板的总数最少? 试建立数学模型。

解 设采用第 j 种方案下料的钢板数为 x_j , 所用钢板的总数为 y , 则上述问题可化为如下的最优化问题:

$$\left\{ \begin{array}{l} \min \quad y = \sum_{j=1}^n x_j \\ \text{s. t.} \quad \sum_{j=1}^n a_{ij}x_j \geq b_i, \quad i = 1, 2, \dots, m \\ \quad \quad \quad (\text{第 } i \text{ 种零件的总数不少于需要量}) \\ \quad \quad \quad x_j \geq 0, \\ \quad \quad \quad x_j \in I, \quad j = 1, 2, \dots, n \end{array} \right. \quad (1-4)$$

其中 $I = \{0, 1, 2, \dots, n, \dots\}$ 。在问题(1-4)中, 由于 x_j 表示用第 j 种方案下料的钢板数, 所以应有 $x_j \in I$, 即 x_j 应取整数值, 这样的问题(1-4)称为整数线性规划问题。 ■

【例 1.5】 汽轮机叶片的优化设计

设汽轮机叶片面积的变化规律为

$$F(x) = F_0 - ax^m$$

其中 F_0 为叶片根部面积, 根据热力学计算可求得各级叶片的平均半径 R_p 和叶片高度 L , 欲确定参数 F_0, a, m 的值, 使其在强度许可的条件下, 叶片的重量最轻, 试建立数学模型。

解 强度条件为 $\sigma_p \leq [\sigma]$, 其中 $[\sigma]$ 是简单拉伸的许用应力, σ_p 是叶片截面上的最大拉应力。

$$\sigma_p = \gamma \omega^2 L R_p \left[1 - \left(1 - \frac{F_1}{F_0} \right) \left(\frac{1}{m+1} + \frac{m}{\frac{D_p}{L}(m+1)(m+2)} \right) \right] \quad (1-5)$$

上式中的 γ, ω, L, R_p 为已知常数, $D_p = 2R_p, F_1 = F_0 - aL^m$ 为叶片顶部面积, γ 为密度, 则叶片的重量为

$$G = \int_0^L \gamma g F(x) dx = \gamma g \left[F_0 L - \frac{a}{m+1} L^{m+1} \right] = f(F_0, a, m)$$

所以汽轮机叶片的优化设计问题归结为在约束条件 $\sigma_p \leq [\sigma], F_0 \geq (F_0)_{\min}$ 之下, 求参数 F_0, a, m 的值, 使目标函数 $G = f(F_0, a, m)$

取得最小值。这里的目标函数和约束函数都是变量 F_0, a, m 的非线性函数, 所以称它为非线性规划问题。非线性规划 (Nonlinear programming) 以后常简写为 NLP。

【例 1.6】 在船体的数学放样中, 常会遇到下面的问题, 已知平面上 n 个点: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, 我们希望用下面的样条函数来拟合这 n 个点。

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4(x - x_1)_\#^3 + \dots + a_{n+3}(x - x_n)_\#^3$$

其中

$$(x - x_i)_\#^3 = \frac{1}{2} [|x - x_i|^3 + (x - x_i)^3], \quad i = 1, 2, \dots, n$$

为了保证求出的曲线 $y = f(x)$ 与连结已知 n 个点所成的曲线具有相同的拐点, 应要求 $\gamma_i f''(x_i) \geq 0, i = 2, 3, \dots, n-1$, 其中

$$\gamma_i = \frac{2}{x_{i+1} - x_{i-1}} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right)$$

所以上述问题可归结为

$$\begin{cases} \min & s = \sum_{i=1}^n [f(x_i) - y_i]^2 \\ \text{s. t.} & \gamma_i f''(x_i) \geq 0, \quad i = 2, 3, \dots, n-1 \end{cases} \quad (1-6)$$

【例 1.7】 投资决策问题

设在一段时间 (比如三年) 内, 有 B 亿元的资金可用于投资, 有 m 个项目 A_1, A_2, \dots, A_m 可供挑选。若对项目 A_i 进行投资, 需花费资金 a_i 亿元, 可获得收益 c_i 亿元, 试确定最佳的投资方案。

解 引入布尔变量

$$x_i = \begin{cases} 1, & \text{若对 } A_i \text{ 投资,} \\ 0, & \text{若对不 } A_i \text{ 投资,} \end{cases} \quad i = 1, 2, \dots, m$$

则约束条件为

$$\begin{cases} \sum_{i=1}^m a_i x_i \leq B \\ x_i \in I = \{0, 1\}, i = 1, 2, \dots, m \end{cases} \quad (1-7)$$

最佳的投资方案应该是：投资少，收益大。即在约束(1-7)之下，求 $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$ ，使目标函数

$$f_1(\mathbf{x}) = \sum_{i=1}^m a_i x_i$$

取最小值，而使另一个目标函数

$$f_2(\mathbf{x}) = \sum_{i=1}^m c_i x_i$$

取最大值，易见，这是一个包含两个目标函数的线性规划问题，称为多目标线性规划问题。

工程技术、现代化管理和自然科学中还有许多问题可以归结为最优化问题来解决，限于篇幅，这里就不介绍了，有兴趣的读者可参看文献 1~4。

1.2 最优化的基本概念

20 世纪的 30 年代末期，由于军事和工业生产发展的需要，提出了一些不能用古典的微分法和变分法解决的最优化问题，在许多学者和广大科技工作者的共同努力下，逐渐产生、发展和形成了一些新的数学方法——最优化方法。近代解决最优化问题的方法，大致可分为两类。

(1) 间接最优化(或解析最优化)方法，就是把所研究的问题(例如，物理、力学、化学和工程问题)用数学方程描述出来，然后用解析方法求其最优解。上一节描述的问题就可以用这一类方法来求解。

(2) 直接最优化(或试验最优化)方法。在某些情况下，所研究

对象本身的机理尚不清楚,无法用数学方程描述,这时可设法在数学原理的指导之下,直接通过少量试验,根据试验结果的比较而求其最优解。这种方法被称为直接最优化方法,它可以用来解没有解析表达式或解析式子很复杂的最优化问题。优选法就是解决这一类问题的重要的数学方法之一。

1.2.1 最优化问题的提法和基本概念

设 $x = (x_1, x_2, \dots, x_n)^T$ 为 n 维实欧氏空间 R^n 内的一点, $f(x)$, $g_i(x)$ ($i = 1, 2, \dots, m$), $h_i(x)$ ($i = m + 1, \dots, p$) 为给定的 n 元函数,则一般的最优化问题的提法是:在约束条件

$$g_i(x) \leq 0, i = 1, 2, \dots, m$$

和

$$h_i(x) = 0, i = m + 1, \dots, p$$

之下,求向量 x ,使函数 $f(x)$ 取极小值(或极大值)。这里 $f(x)$ 称为目标函数, $g_i(x) \leq 0$ 称为不等式约束条件, $h_i(x) = 0$ 称为等式约束条件, $x = (x_1, x_2, \dots, x_n)^T$ 称为设计变量或决策变量。上一节讲述的七个例子,其数学模型都可以写成上述问题的形式。今后我们把上述的最优化问题简写为

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, i = 1, 2, \dots, m \\ & h_i(x) = 0, i = m + 1, \dots, p \end{cases} \quad (2-1)$$

令 $R = \{x | g_i(x) \leq 0, i = 1, 2, \dots, m; h_i(x) = 0, i = m + 1, \dots, p\}$,称 R 为问题(2-1)的可行集或容许集,称 $x \in R$ 为(2-1)的可行解或容许解。

定义 2.1 若有 $x^* \in R$,使得 $\forall x \in R$,均有 $f(x^*) \leq f(x)$,则称 x^* 为问题(2-1)的(全局)最优解(点)或全局极小点。

若 $\exists x^* \in R$,使得 $\forall x \in R, x \neq x^*$,均有 $f(x^*) < f(x)$ 成立,则称 x^* 为问题(2-1)的严格全局极小点。

若 $x^* \in R$ 具有性质:存在 x^* 的一个邻域 $N(x^*, \epsilon)$,使得 $\forall x$

$\in R \cap N(x^*, \varepsilon)$, 均有 $f(x^*) \leq f(x)$, 则称 x^* 为问题(2-1)的一个局部最优解(点)或局部极小点, 其中 $\varepsilon > 0$ 为实数, $N(x^*, \varepsilon) \triangleq \{x \mid \|x - x^*\| < \varepsilon\}$, 称为 x^* 的 ε 邻域, $\|\cdot\|$ 为向量模。

若 $\exists x^* \in R$, $\varepsilon > 0$, 使得 $\forall x \in R \cap N(x^*, \varepsilon)$, $x \neq x^*$, 均有 $f(x^*) < f(x)$, 则称 x^* 为问题(2-1)的严格局部极小点。

显然, 问题(2-1)的一个全局最优解也是它的一个局部最优解, 但局部最优解不一定是全局最优解。

求解最优化问题(2-1), 就是要求(2-1)的全局最优解, 但在一般情况下, 我们往往只能求出它的一个局部最优解。因此, 在本书中, 所介绍的最优化方法也主要是求其局部最优解的数值方法。

定理 2.1 设 $f(x)$, $g_i(x)$, $i = 1, 2, \dots, m$, $h_i(x)$, $i = m + 1, \dots, p$ 在 \mathbf{R}^n 上连续, 则问题(2-1)的可行集 R 为闭集; 它的全局最优解集合亦为闭集。

证明留作习题。

定理 2.2 (一阶必要条件)

设函数 $f(x) \in C^1$ 定义在 $\Omega \subset \mathbf{R}^n$ 上, x^* 是 Ω 的一个内点, 若 x^* 为 $f(x)$ 的一个极小点, 则 $\nabla f(x^*) = 0$ 。其中

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)^T$$

为 $f(x)$ 的梯度。

证明留给读者练习。

满足条件 $\nabla f(x^*) = 0$ 的点 x^* 称为 $f(x)$ 的稳定点或驻点。

1.2.2 二维最优化问题的几何意义

考察如下的二维最优化问题:

$$\begin{cases} \min & z = f(x_1, x_2) \\ \text{s. t.} & g_i(x_1, x_2) \leq 0, i = 1, 2, \dots, m \end{cases} \quad (2-2)$$

$z = f(x_1, x_2)$ 在三维欧氏空间中表示一张曲面, $f(x_1, x_2) =$

C (常数) 称为等高线或等值线, 让常数 C 依次取一系列的数值 c_1, c_2, \dots , 就得到一族等值线 (参见图 1-1)。从等值线族的图上可大致看出函数值的变化情况。

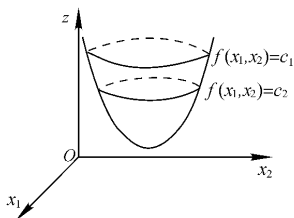


图 1-1

【例 2.1】 研究如下问题：

$$\left\{ \begin{array}{l} \min \quad z = f(x_1, x_2) = (x_1 - 1.5)^2 + x_2^2 \\ \text{s. t.} \quad g_1(x_1, x_2) = x_1^2 + x_2^2 - 1 \leq 0 \\ \quad \quad g_2(x_1, x_2) = 2x_1 + x_2 - 1 \geq 0 \\ \quad \quad g_3(x_1, x_2) = x_1 \geq 0 \\ \quad \quad g_4(x_1, x_2) = x_2 \geq 0 \end{array} \right. \quad (2-3)$$

画出(2-3)的可行集 R , 如图 1-2 所示 (见图中阴影部分)。

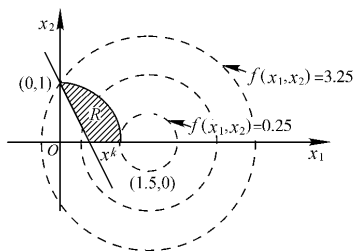


图 1-2

令 $z = 0.5, 1, 2, 3.25, \dots$, 作出目标函数的等值线族(图中虚线)。从图中容易看出: 问题(2-3)的最优解为 $x^* = (1, 0)^T$, 最优值为 $f(x^*) = 0.25$, 像上面这样, 通过作图来求得最优解的方法称为图解法。其步骤可概述如下:

(1) 画出问题的可行集 R 的图形。

(2) 让常数 C 取一系列的数值 c_1, c_2, \dots , 作出目标函数的等值线族。

(3) 通过观察等值线族与可行集 R , 确定使目标函数取得最小值的可行点 x^* , 即为所求的最优解。

1.2.3 最优化问题分类

1. 线性规划(Linear Programming)

若 $f(x), g_i(x), h_i(x)$ 都是 x 的线性函数, 则上述问题(2-1)称为线性规划问题, 简称为 LP 问题。LP 问题的标准形式, 可以有几种形式略有不同而实质却是一样的表述形式。

(1) 在线性约束条件

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, 2, \dots, m$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n$$

之下, 求 x , 使目标函数 $f(x) = \sum_{i=1}^n c_i x_i$ 取最小(或最大)值。其中 a_{ij}, c_j, b_i 是已知常数, 且 $b_i \geq 0, i = 1, 2, \dots, m, m \leq n$ 。

(2) 已知 $c = (c_1, c_2, \dots, c_n)^T, A = (a_{ij})_{m \times n}, b = (b_1, b_2, \dots, b_m)^T \geq 0$, 在约束条件 $Ax = b, x \geq 0$ 之下, 求 x , 使 $f(x) = c^T x$ 取最小(或最大)值。

(3) 在约束条件 $\sum_{i=1}^n x_i p_i = b, x \geq 0$ 之下, 求 x , 使 $f(x) = c^T x$ 取最小(或最大)值, 其中 $p_i (i = 1, 2, \dots, n)$ 是矩阵 A 的第 i 列。

2. 二次规划(Quadratic Programming)

在式(2-1)中,若 $g_i(x)$ ($i = 1, 2, \dots, m$), $h_i(x)$ ($i = m + 1, \dots, p$) 都是 x 的线性函数,而 $f(x)$ 是 x 的二次函数,则上述问题(2-1)称为二次规划问题,常简称为 QP 问题。

3. 非线性规划(Nonlinear Programming)

在式(2-1)中,若函数 $f(x)$, $g_i(x)$ ($i = 1, 2, \dots, m$) 和 $h_i(x)$ ($i = m + 1, \dots, p$) 中至少有一个是 x 的非线性函数,则上述问题(2-1)称为非线性规划问题,今后常简称为 NLP 问题,显然二次规划是最简单的一种非线性规划。一般的非线性规划问题,根据约束条件的情况可分为:

(1) 无约束非线性规划问题,即没有任何约束条件限制的非线性规划问题,有时也称之为无约束优化问题或简称为无约束问题,常写成

$$\min f(x), \quad x \in \mathbf{R}^n$$

(2) 约束非线性规划问题,即在(2-1)中,至少有一个约束条件限制的非线性规划问题,有时也称之为约束优化问题或简称为有约束问题。

4. 整数规划(Integer Programming)

在(2-1)中,如果还要求某些设计变量或全部设计变量取非负的整数值,则上述问题(2-1)称为整数规划问题。整数规划(Integer Programming)常简称为 IP。如果所有的设计变量都要求取整数值,这类问题称为纯整数规划问题或全整数规划问题。如果只要求部分设计变量取整数值,而其余的设计变量为连续的非负变量,这类问题称为混合型整数规划问题。如果在整数规划问题中,要求其中的部分设计变量或所有的设计变量取值为 0 或 1,这类问题称为 0-1 规划问题。

整数规划也可以分为:整数线性规划和整数非线性规划。关于整数规划的系统介绍可参看文献 5 ~ 7。

5. 几何规划(Geometric Programming)

几何规划是由 C. Zener 在 1961 年首先叙述的,是作为解决工程中的优化设计问题而提出的。在(2-1)中,若目标函数和约束函数都是设计变量 x_1, x_2, \dots, x_n 的广义多项式,即形如

$$\varphi(x) = \sum_{i=1}^m c_i \prod_{j=1}^n x_j^{r_{ij}}$$

的函数,其中 c_i, r_{ij} 可取任意实数值,而 $x_i > 0$,则称非线性规划

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 1, \quad i = 1, 2, \dots, M \\ & x > 0 \end{cases} \quad (2-4)$$

为几何规划,这里 $f(x), g_i(x)$ ($i = 1, 2, \dots, M$) 都是 x 的广义多项式,当所有的 c_i 为正数时,称 $\varphi(x)$ 为正项式(Posynomial)或正定多项式。在(2-4)中,若 $f(x), g_i(x)$ 均为正项式,则称它为正规几何规划,否则称为符号几何规划。关于几何规划的简要介绍见文献 8。

6. 多目标规划(Multiobjective Programming)

在(2-1)中,若目标函数 $f(x) = (f_1(x), f_2(x), \dots, f_p(x))^T$, $p \geq 2$,即 $f(x)$ 是 x 的一个向量函数,则称问题(2-1)为多目标规划问题,例如 1.1 中例 1.7 为一个多目标线性规划问题(其中的 $p = 2$)。由于解决实际问题的需要,近二十多年来,多目标规划得到了迅速的发展,获得了极其广泛的推广、应用。参见文献 8, 41 ~ 44。

上面简要介绍了线性规划、二次规划、非线性规划、几何规划、整数规划和多目标规划,此外,还有动态规划、不可微规划、参数规划和随机规划等,这里就不介绍了,有兴趣的同志可参看文献 9 ~ 13。

本书将在第 2 章介绍线性规划,第 3 章介绍无约束非线性规划,第 4 章介绍约束非线性规划,第 5 章介绍多目标规划,第 6 章介

绍整数规划,第7章介绍动态规划,第8章介绍进化算法,第9章介绍最优化方法若干应用实例。

1.3 凸集和凸函数

凸集和凸函数的理论一般称为凸分析,是最优化的理论基础。本节介绍凸集和凸函数最基本的知识。

1.3.1 凸 集

直观地看,三角形、矩形、圆形和球体等,其形体都是凸的。易见,凡是凸的形体,都有这样的几何特征:连接形体中任意两点的直线段,都在该形体之中(参见图 1-3),于是我们可以这样来定义凸集。

定义 3.1 设 $\Omega \subset \mathbf{R}^n$, 如果对于任意的点 $x, y \in \Omega$, 连接点 x, y 的线段上的一切点都在 Ω 中, 即对于 $0 \leq \mu \leq 1$ 的一切 μ , 总有 $\mu x + (1 - \mu)y \in \Omega$, 则称 Ω 为一个凸集。

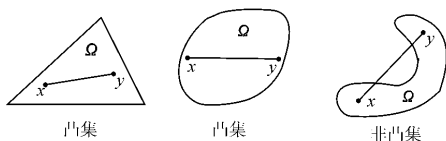


图 1-3

显然, 单个点 x 组成的集合、整个空间 \mathbf{R}^n 都是凸集, 我们规定空集 \emptyset 为凸集。

【例 3.1】 满足线性规划问题的约束条件 $Ax = b, x \geq 0$ 的一切点 x 所组成的集合 R 是一个凸集。

证明 由题意得: $R = \{x | Ax = b, x \geq 0\}$, 对于任意的 $x, y \in R$, 都有

$$Ax = b, x \geq 0$$

$$Ay = b, y \geq 0$$

因此,对任意的 $\lambda \in [0, 1]$, 总有

$$\begin{aligned} A[\lambda x + (1 - \lambda)y] &= \lambda Ax + (1 - \lambda)Ay \\ &= \lambda b + (1 - \lambda)b = b \end{aligned}$$

及 $\lambda x + (1 - \lambda)y \geq 0$

故 $\lambda x + (1 - \lambda)y \in R$

所以 R 为凸集。 ■

【例 3.2】 设 $c \neq 0$ 是已给定的 n 维向量, b 是已给常数, 集合 $H = \{x | x \in \mathbf{R}^n, c^T x = b\}$ 称为超平面, 它也是一个凸集。

证明 因为对任意的 $x, y \in H$, 有 $c^T x = b; c^T y = b$, 所以对于 $[0, 1]$ 中的任意 μ , 总有

$$\begin{aligned} c^T[\mu x + (1 - \mu)y] &= \mu c^T x + (1 - \mu)c^T y \\ &= \mu b + (1 - \mu)b = b \end{aligned}$$

即 $\mu x + (1 - \mu)y \in H$, 因此, 集合 H 也是一个凸集。 ■

前面讲过的以 $x^{(0)}$ 为中心的 ε 邻域

$$N(x^{(0)}, \varepsilon) = \{x | \|x - x^{(0)}\| < \varepsilon\}$$

易证它是凸集。当 $n = 2$ 时, $N(x^{(0)}, \varepsilon)$ 表示以 $x^{(0)}$ 为圆心, 以 ε 为半径的圆内所有点的集合。

若向量模 $\|x - x^{(0)}\| = [\sum_{i=1}^n (x_i - x_i^{(0)})^2]^{\frac{1}{2}}$, 它表示两点 x 与 $x^{(0)}$ 之间的距离, 称为欧氏距离或欧氏范数。

定义 3.2 设实数 $\alpha_i \geq 0$ ($i = 1, 2, \dots, p$), $\sum_{i=1}^p \alpha_i = 1$, $x^{(i)} \in \mathbf{R}^n$ ($i = 1, 2, \dots, p$), 则称 $x = \sum_{i=1}^p \alpha_i x^{(i)}$ 为点 $x^{(1)}, x^{(2)}, \dots, x^{(p)}$ 的一个凸组合。

定理 3.1 集合 $\Omega \subset \mathbf{R}^n$ 为凸集的充要条件是: 点 $x^{(i)} \in \Omega$

($i = 1, 2, \dots, p$) 的任意凸组合仍包含在 Ω 中。

证明 1° 由定义 2 和凸集定义可知定理 3.1 的充分性显然成立。

2° 必要性。即设 Ω 为凸集, 要证 Ω 中的点的任何凸组合仍包含在 Ω 中。

用数学归纳法证明。

当 $p = 1$ 时, 结论显然成立。

设 $p = n$ 时, 结论成立。那么, 当 $p = n + 1$ 时, 令

$$\mathbf{x} = \alpha_1 \mathbf{x}^{(1)} + \alpha_2 \mathbf{x}^{(2)} + \cdots + \alpha_n \mathbf{x}^{(n)} + \alpha_{n+1} \mathbf{x}^{(n+1)}$$

不失一般性, 可设 $\alpha_{n+1} \neq 1$, 于是 $1 - \alpha_{n+1} > 0$, 上式可改写为

$$\mathbf{x} = (1 - \alpha_{n+1})\mathbf{z} + \alpha_{n+1}\mathbf{x}^{(n+1)}$$

其中

$$\mathbf{z} = \frac{\alpha_1}{1 - \alpha_{n+1}} \mathbf{x}^{(1)} + \cdots + \frac{\alpha_n}{1 - \alpha_{n+1}} \mathbf{x}^{(n)}$$

由于 $\frac{\alpha_i}{1 - \alpha_{n+1}} \geq 0, i = 1, 2, \dots, n$, 且 $\sum_{i=1}^n \frac{\alpha_i}{1 - \alpha_{n+1}} = 1$, 由归纳法假设知 $\mathbf{z} \in \Omega$, 又因为 Ω 是凸集, 所以

$$\mathbf{x} = (1 - \alpha_{n+1})\mathbf{z} + \alpha_{n+1}\mathbf{x}^{(n+1)} \in \Omega$$

定理 3.2 任意一组凸集的交集仍为凸集。

证明留给读者练习。

定义 3.3 包含集合 $\Omega \subset \mathbf{R}^n$ 的所有凸集的交集称为 Ω 的凸包, 记为 $\text{Co}(\Omega)$ 或 $H(\Omega)$ 。

由定理 3.2 可知 $\text{Co}(\Omega)$ 为凸集, 它实际上是 \mathbf{R}^n 中包含 Ω 的最小凸集。

定义 3.4 设 $\Omega \subset \mathbf{R}^n$, 如果对于任何 $\mathbf{x} \in \Omega$ 及所有的 $\alpha \geq 0$, 都有 $\alpha \mathbf{x} \in \Omega$, 则称集合 Ω 为一个锥。一个同时为凸集的锥, 称为凸锥。

1.3.2 凸函数

定义 3.5 设 $f(x)$ 是定义在非空凸集 $\Omega \subset \mathbf{R}^n$ 上的函数, 若对任意 $x, y \in \Omega$, 不等式

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad (3-1)$$

对于 $0 \leq \lambda \leq 1$ 中的一切 λ 都成立, 则称 $f(x)$ 为 Ω 上的凸函数。

若对于 $0 < \lambda < 1$ 中的一切 λ 和 $x \neq y (x, y \in \Omega)$, 都有

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y) \quad (3-2)$$

成立, 则称 $f(x)$ 为 Ω 上的严格凸函数。

如果将 (3-1) 式中的不等号“ \leq ”反向, 将 (3-2) 式中的严格不等号“ $<$ ”反向, 则可以类似地定义凹函数和严格凹函数, 详细的叙述请读者写出。

一元凸函数 $f(x)$ 有明显的几何意义: 过函数图象上任意两点 A, B 的弦线段, 处处都在函数图象的上方, 而凹函数的情形则正好相反, 参见图 1-4。

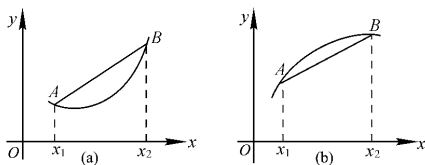


图 1-4

容易证明: 若 $f(x)$ 是凸集 Ω 上的凸函数, 则 $g(x) = -f(x)$ 是 Ω 上的凹函数, 反之亦真。

例如, $f(x) = x^2 + 4$ 是一个一元严格凸函数, 而 $g(x) = 4 - x_1^2 - x_2^2$ 是一个二元严格凹函数。

为了以后应用的需要, 下面来研究凸函数的性质。

定理 3.3 设 $f_1(x), f_2(x)$ 是凸集 $\Omega \subset \mathbf{R}^n$ 上的凸函数, 则

$f_1(x) + f_2(x)$ 在 Ω 上也是凸的。

证明 设 $x^{(1)}, x^{(2)} \in \Omega, \lambda \in [0, 1]$, 则

$$\begin{aligned} & f_1(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) + f_2(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) \\ & \leq \lambda f_1(x^{(1)}) + (1 - \lambda)f_1(x^{(2)}) + \lambda f_2(x^{(1)}) + (1 - \lambda)f_2(x^{(2)}) \\ & = \lambda[f_1(x^{(1)}) + f_2(x^{(1)})] + (1 - \lambda)[f_1(x^{(2)}) + f_2(x^{(2)})] \end{aligned}$$

即 $f_1(x) + f_2(x)$ 也是 Ω 上的一个凸函数。 ■

定理 3.4 设 $f(x)$ 是凸集 $\Omega \subset \mathbf{R}^n$ 上的凸函数, 则对任意常数 $a > 0$, 函数 $af(x)$ 也是凸的。

推论 设 $f_1(x), f_2(x), \dots, f_p(x)$ 是凸集 $\Omega \subset \mathbf{R}^n$ 上的凸函数,

$\lambda_i \geq 0$ ($i = 1, 2, \dots, p$) 则非负的线性组合 $\sum_{i=1}^p \lambda_i f_i(x)$ 在 Ω 上也是凸的。

定理 3.5 设 $f(x)$ 是凸集 $\Omega \subset \mathbf{R}^n$ 上的凸函数, 则对任一个实数 c , 水平集

$$\Omega_c = \{x | x \in \Omega, f(x) \leq c\}$$

是凸集。

证明 设 $x^{(1)} \in \Omega_c, x^{(2)} \in \Omega_c$, 则 $x^{(1)}, x^{(2)} \in \Omega$, 且 $f(x^{(1)}) \leq c, f(x^{(2)}) \leq c$. 因为集 Ω 是凸集, 所以

$$\lambda x^{(1)} + (1 - \lambda)x^{(2)} \in \Omega, \lambda \in [0, 1]$$

又因为 $f(x)$ 是 Ω 上的凸函数, 所以

$$\begin{aligned} f(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) & \leq \lambda f(x^{(1)}) + (1 - \lambda)f(x^{(2)}) \\ & \leq \lambda c + (1 - \lambda)c = c \end{aligned}$$

即 $\lambda x^{(1)} + (1 - \lambda)x^{(2)} \in \Omega_c$, 因此 Ω_c 是凸集。 ■

推论 设 $f_i(x)$ ($i = 1, 2, \dots, p$) 为凸集 $\Omega \subset \mathbf{R}^n$ 上的凸函数, c_i ($i = 1, 2, \dots, p$) 为实常数, 则 Ω 中同时满足 $f_1(x) \leq c_1, f_2(x) \leq c_2, \dots, f_p(x) \leq c_p$ 的点所构成的集合 Ω_c 为凸集。

因为 $\Omega_c = \{x | x \in \Omega, f_i(x) \leq c_i, i = 1, 2, \dots, p\} = \bigcap_{i=1}^p \Omega_{c_i}$,

其中 $\Omega_{c_i} = \{x | x \in \Omega, f_i(x) \leq c_i\}$, 由定理 3.5 可知 Ω_{c_i} 是凸集, 而由定理 3.2 知凸集的交集为凸集, 所以 Ω_c 为凸集。

定理 3.6 定义在凸集 $\Omega \subset \mathbf{R}^n$ 上的可微函数 $f(x)$ 为凸函数的充要条件是: 对于所有的 $x, y \in \Omega$ 都有

$$f(y) \geq f(x) + (\nabla f(x))^T(y - x) \quad (3-3)$$

证明 1° 必要性 设 $f(x)$ 是凸函数, 要证 (3-3) 式成立。

因为 $f(x)$ 是凸函数, 所以对于任意 $x, y \in \Omega$ 和任意 $\lambda \in (0, 1)$, 有

$$f(\lambda y + (1 - \lambda)x) \leq \lambda f(y) + (1 - \lambda)f(x) \quad (3-4)$$

由可微函数的定义(见附录)知

$$\begin{aligned} f(\lambda y + (1 - \lambda)x) &= f(x + \lambda(y - x)) \\ &= f(x) + \lambda(\nabla f(x))^T(y - x) + \lambda \|y - x\| \beta(x; \lambda(y - x)) \end{aligned} \quad (3-5)$$

其中当 $\lambda \rightarrow 0$ 时, $\beta(x; \lambda(y - x)) \rightarrow 0$ 。

由式 (3-4)、(3-5) 可得

$$\begin{aligned} &(\nabla f(x))^T(y - x) + \|y - x\| \beta(x; \lambda(y - x)) \\ &\leq f(y) - f(x) \end{aligned}$$

令 $\lambda \rightarrow 0$, 即得 (3-3) 式。

2° 充分性 设 $\forall x, y \in \Omega$, (3-3) 式成立, 要证 $f(x)$ 是凸函数。

令 $z = \lambda x + (1 - \lambda)y$, $\lambda \in [0, 1]$ 。因为

$$\begin{aligned} f(x) &\geq f(z) + (\nabla f(z))^T(x - z) \\ f(y) &\geq f(z) + (\nabla f(z))^T(y - z) \end{aligned}$$

$$\begin{aligned} \text{所以} \quad &\lambda f(x) + (1 - \lambda)f(y) \\ &\geq f(z) + (\nabla f(z))^T[\lambda(x - z) + (1 - \lambda)(y - z)] \\ &= f(z) = f(\lambda x + (1 - \lambda)y) \end{aligned}$$

即

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

故 $f(x)$ 是凸函数。

只要将(3-3)式中的“ \leq ”号改为“ $<$ ”，就可得到严格凸函数的充要条件，其叙述和证明与定理 3.6 类似，留给读者练习。

从几何上看，若 $f(x)$ 是凸函数，则函数图象总是在切线（或切平面）的上方。定理 3.6 说明：曲线（面）上任一点的切线（切平面）上的点的纵坐标 $N(y) = f(x) + (\nabla f(x))^T(y - x)$ 总不会大于曲线（面）在该点的纵坐标 $f(y)$ （参看图 1-5）。

定理 3.7 设 $\Omega \subset \mathbb{R}^n$ 为含有内点的凸集， $f(x) \in C^2$ ，则 f 在 Ω 上为凸函数的充要条件是： $f(x)$ 的（海赛）Hesse 矩阵 $F(x) = \nabla^2 f(x)$ 在整个 Ω 上是半正定的。

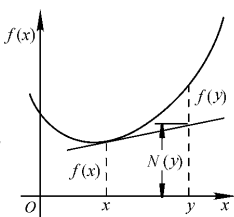


图 1-5

函数 $f(x)$ 在一点 x 的海赛矩阵是指下述的 n 阶对称矩阵

$$F(x) = \nabla^2 f(x) = \left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right)_{n \times n}$$

例如 $f(x) = 5x_1^2 - 6x_1x_2 + 5x_2^2$ ，其海赛阵为

$$F(x) = \nabla^2 f(x) = \begin{pmatrix} 10 & -6 \\ -6 & 10 \end{pmatrix}$$

若 $f(x) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{j=1}^n b_j x_j + c$ ， $a_{ij} = a_{ji}$ 。令

$$A = (a_{ij})_{n \times n}, \quad b = (b_1, b_2, \dots, b_n)^T$$

则

$$f(x) = \frac{1}{2} x^T A x + b^T x + c$$

$$\nabla f(x) = A x + b, \quad \nabla^2 f(x) = A$$

证明 1° 充分性 即若 $f(x)$ 的海赛阵 F 在整个 Ω 上是半正定的，则 $f(x)$ 是凸函数。

因为 $f \in C^2$ ，即 $f(x)$ 是二阶连续可微函数，则对任意 $x, y \in$

Ω , 根据 Taylor 公式必有

$$f(y) = f(x) + (\nabla f(x))^T(y - x) + \frac{1}{2}(y - x)^T F(x + \alpha(y - x))(y - x) \quad (3-6)$$

其中 $\alpha \in (0, 1)$, 由于 F 在 Ω 上处处是半正定的, 故

$$\begin{aligned} & (y - x)^T F(x + \alpha(y - x))(y - x) \\ &= (y - x)^T F(\alpha y + (1 - \alpha)x)(y - x) \geq 0 \end{aligned}$$

而 $\alpha y + (1 - \alpha)x \in \Omega$

所以对任意 $x, y \in \Omega$, 必有

$$f(y) \geq f(x) + (\nabla f(x))^T(y - x)$$

成立, 由定理 3.6 可知, $f(x)$ 在 Ω 上是凸函数。

2° 必要性 即若 $f(x)$ 在含有内点的凸集 Ω 上是凸函数, 则 $f(x)$ 的海赛阵 $F(x)$ 在整个 Ω 上是半正定的。

用反证法证明。设海赛阵 F 在某点 x 不是半正定的, 我们来导出矛盾。分两种情况:

(1) 当 x 是 Ω 的一个内点时, 由矩阵知识得知, 若 $F(x)$ 不是半正定的, 则必存在 $y \in \Omega$, 使

$$(y - x)^T F(x)(y - x) < 0$$

由于海赛阵 F 的连续性, 可以选择某个这样的 y 使得对于所有的 $\alpha \in [0, 1]$, 有

$$(y - x)^T F(x + \alpha(y - x))(y - x) < 0$$

而由 (3-6) 式, 这意味着对 $x, y \in \Omega$, 有

$$f(y) < f(x) + (\nabla f(x))^T(y - x)$$

这与 $f(x)$ 在 Ω 上为凸函数相矛盾。

(2) 当 x 是 Ω 的边界点时

由海赛阵 $F(x)$ 的连续性, 必可找到 x 的一个小邻域 $N(x, \epsilon)$, 使在该邻域内, F 也不是半正定的, 而在 $N(x, \epsilon)$ 内, 必含有 Ω 的内点, 这就转化为情况 (1)。

所以在整个 Ω 上,海赛阵 $F(x)$ 在某点 x 不是半正定的假设是错误的,即 $f(x)$ 的海赛阵 $F(x)$ 处处是半正定的。 ■

显然,当 $n = 1$ 时, $f(x)$ 的海赛阵 $F(x) = f''(x)$, 此时定理 3.7 为: 若 $f \in C^2$, 则 $f(x)$ 为凸函数的充要条件是 $f''(x) \geq 0$, 其中 $x \in (a, b)$ 。

定理 3.8 设 $\Omega \subset \mathbf{R}^n$ 为非空开凸集, $f \in C^2$, 若 $\nabla^2 f(x)$ 在 Ω 上处处正定, 则 $f(x)$ 是 Ω 上的严格凸函数。

应该指出: 定理 3.8 的逆定理一般是不成立的。即 $f(x)$ 在 Ω 上处处正定仅是 $f(x)$ 为严格凸函数的充分条件, 但非必要条件。例如 $f(x) = x^4$ 为严格凸函数, 但 $F(x) = f''(x) = 12x^2$, $F(0) = 0$, 即在 $x = 0$ 点, 海赛阵 $F(x)$ 不是正定的。

【例 3.3】 讨论 $f(x) = 5x_1^2 - 6x_1x_2 + 5x_2^2$ 在凸集 $\Omega \subset \mathbf{R}^n$ 上是否为严格凸函数。因为

解

$$F(x) = \nabla^2 f(x) = \begin{pmatrix} 10 & -6 \\ -6 & 10 \end{pmatrix}$$

中所有的顺序主子式为

$$10 > 0, \quad \begin{vmatrix} 10 & -6 \\ -6 & 10 \end{vmatrix} = 64 > 0$$

所以, $f(x)$ 的海赛阵 $F(x) = \nabla^2 f(x) = \begin{pmatrix} 10 & -6 \\ -6 & 10 \end{pmatrix}$ 在 Ω 上是处处正定的。

因此, $f(x)$ 是 Ω 上的严格凸函数。 ■

【例 3.4】 $f(x) = \frac{1}{2}x^T Ax + b^T x + c$, 其中 $A = (a_{ij})_{n \times n}$ 为对称阵, $b = (b_1, b_2, \dots, b_n)^T$ 为给定向量, c 为常数, 则

(1) 当 A 为半正定时 (记为 $A \geq 0$), $f(x)$ 是凸函数。

(2) 当 A 为正定时 (记为 $A > 0$), $f(x)$ 为严格凸函数。

证明 因为 $F(x) = \nabla^2 f(x) = A$, 由定理 3.7 和定理 3.8 即可得到所需结论。 ■

1.3.3 凸规划

考虑如下的非线性规划问题:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, i = 1, 2, \dots, m \\ & h_i(x) = 0, i = m+1, m+2, \dots, p \end{cases} \quad (3-7)$$

易见, 在(3-7)中, 总可以把等式约束 $h_i(x) = 0$ 改写为不等式约束。这是因为

$$h_i(x) = 0 \Leftrightarrow h_i(x) \leq 0, \text{ 且 } -h_i(x) \leq 0$$

因此, 不失一般性, 我们可以考虑仅包含不等式约束的优化问题:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, i = 1, 2, \dots, l \end{cases} \quad (3-8)$$

定义 3.6 在问题(3-8)中, 若 $f(x), g_i(x) (i = 1, 2, \dots, l)$ 均为可行集 R 上的凸函数, 则称这样的问题为凸规划问题。

定理 3.9 凸规划问题(3-8)的可行集 R 是凸集。

证明 由于 $R = \{x | g_i(x) \leq 0, i = 1, 2, \dots, l\}$ 。所以对于任意 $x^{(1)}, x^{(2)} \in R$, 有 $g_i(x^{(1)}) \leq 0$ 和 $g_i(x^{(2)}) \leq 0 (i = 1, 2, \dots, l)$ 。因为 $g_i(x)$ 为凸函数, 所以对任意 $\lambda \in [0, 1]$, 必有

$$g_i(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) \leq \lambda g_i(x^{(1)}) + (1 - \lambda)g_i(x^{(2)}) \leq 0$$

其中 $i = 1, 2, \dots, l$ 。

因此, $\lambda x^{(1)} + (1 - \lambda)x^{(2)} \in R$, 即 R 为凸集。 ■

定理 3.10 对于凸规划问题(3-8), 目标函数 $f(x)$ 的任一局部极小点都是 $f(x)$ 在非空可行集 R 上的全局极小点。

证明 设 $x^* \in R$ 为 $f(x)$ 的任意一个局部极小点, 要证 x^* 是 $f(x)$ 在 R 上的全局极小点。

用反证法证明。假设 x^* 不是 $f(x)$ 在 R 上的全局极小点。则一定存在 $\bar{x} \in R$, 使 $f(\bar{x}) < f(x^*)$ 。由定理 3.9 知 R 为凸集, 从而对任意 $\lambda \in [0, 1]$, 必有

$$\lambda \bar{x} + (1 - \lambda)x^* \in R$$

又因为 $f(x)$ 为凸函数, 所以对 $\lambda \in (0, 1)$, 有

$$\begin{aligned} f(\lambda \bar{x} + (1 - \lambda)x^*) &\leq \lambda f(\bar{x}) + (1 - \lambda)f(x^*) \\ &= \lambda[f(\bar{x}) - f(x^*)] + f(x^*) < f(x^*) \end{aligned}$$

成立, 上式对 $0 < \lambda < 1$ 中的一切 λ 均成立, 故当 λ 足够小时, $y = \lambda \bar{x} + (1 - \lambda)x^* \in R$ 将足够地接近于 x^* , 而这与 x^* 是 $f(x)$ 的一个局部极小点矛盾。

所以 $f(x)$ 的任一局部极小点也是 $f(x)$ 在 R 上的全局极小点。 ■

定理 3.10 的重要性在于, 它使求解凸规划问题(3-8)的全局极值问题与局部极值问题统一了起来。只要求得了凸规划问题(3-8)的任一局部极小点, 它也就是全局极小点, 因而使问题大为简化。

应该指出: 在一般情况下, 问题(3-8)的全局极小点不一定是惟一的, 但是用与上面类似的方法, 可以证明:

定理 3.11 对于凸规划问题(3-8), 若 $f(x)$ 在非空可行集 R 上是严格凸函数, 则问题(3-8)的全局极小点是惟一的。

证明留作习题。

1.3.4 拟凸函数

定义 3.7 设 $\Omega \subset \mathbf{R}^n$ 为非空凸集, $f: \Omega \rightarrow \mathbf{R}^1$, 若 $\forall x, y \in \Omega$, 均有

$$f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}, \forall \lambda \in (0, 1)$$

成立, 则称 $f(x)$ 在 Ω 上为拟凸函数。

若 $\forall x, y \in \Omega, f(x) \neq f(y)$, 均有

$$f(\lambda x + (1 - \lambda)y) < \max\{f(x), f(y)\}, \forall \lambda \in (0, 1)$$

成立, 则称 $f(x)$ 在 Ω 上为严格拟凸函数。

若 $-f$ 是 Ω 上的(严格)拟凸函数, 则称 f 为 Ω 上的(严格)拟凹函数。

注意 严格拟凸函数不一定是拟凸函数。

命题 凸函数必为拟凸函数, 但反之不真。

定理 3.12 设 $\Omega \subset \mathbf{R}^n$ 为非空凸集, $f: \Omega \rightarrow \mathbf{R}^1$, 则 $f(x)$ 是 Ω 上的拟凸函数的充要条件是: $\forall c \in \mathbf{R}^1$, 水平集 $H_c(f) = \{x | x \in \Omega, f(x) \leq c\}$ 是凸集。

定理 3.13 设 $\Omega \subset \mathbf{R}^n$ 为非空凸集, $f: \Omega \rightarrow \mathbf{R}^1$ 为可微函数, 则 f 是 Ω 上的拟凸函数的充要条件是, 若

$$f(x^{(1)}) \leq f(x^{(2)}), \forall x^{(1)}, x^{(2)} \in \Omega$$

$$\text{则} \quad \nabla f(x^{(2)})^T (x^{(1)} - x^{(2)}) \leq 0$$

以上命题和定理的证明留给读者练习。

1.3.5 伪凸函数

定义 3.8 设 $\Omega \subset \mathbf{R}^n$ 为非空开凸集, $f: \Omega \rightarrow \mathbf{R}^1$ 为可微函数, 如果对于

$$\nabla f(x^{(1)})^T (x^{(1)} - x^{(2)}) \leq 0, \quad \forall x^{(1)}, x^{(2)} \in \Omega$$

必有 $f(x^{(1)}) \leq f(x^{(2)})$, 则称 $f(x)$ 是 Ω 上的伪凸函数。如果对于

$$\nabla f(x^{(1)})^T (x^{(1)} - x^{(2)}) \leq 0, \quad \forall x^{(1)}, x^{(2)} \in \Omega, x^{(1)} \neq x^{(2)}$$

必有 $f(x^{(1)}) < f(x^{(2)})$, 则称 $f(x)$ 是 Ω 上的严格伪凸函数。

若 $-f(x)$ 是 Ω 上的(严格)伪凸函数, 则称 $f(x)$ 是 Ω 上的(严格)伪凹函数。

易证可微的凸函数是伪凸函数, 严格伪凸函数必是伪凸函数。

注意 这里定义的(严格)伪凸函数一定是可微的, 去掉可微性要求的广义伪凸函数的定义及性质可参见文献 54。

下面的定理说明了伪凸函数和拟凸函数之间的关系。

定理 3.14 设 $\Omega \subset \mathbf{R}^n$ 为非空凸集, $f: \Omega \rightarrow \mathbf{R}^1$ 为可微函数, 若 f 是 Ω 上的伪凸函数, 则 f 是 Ω 上的严格拟凸函数和拟凸函数。

证明可参见文 11 或 34。

1.3.6 广义凸规划

考虑优化问题:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, \quad i = 1, 2, \dots, m \end{cases} \quad (3-8)$$

若问题(3-8)的可行集 R 是凸集, $f(x)$ 是 R 上的(严格)拟凸函数或(严格)伪凸函数, 则称问题(3-8)为广义凸规划问题。

定理 3.15 设(3-8)的可行集 R 是凸集, $f(x)$ 是 R 上的严格拟凸函数, 则广义凸规划(3-8)的任一局部最优解 x^* 也是它的全局最优解。

证明 用反证法证明。假设 x^* 不是(3-8)的全局最优解, 则必有 $\bar{x} \in R, \bar{x} \neq x^*$, 使

$$f(\bar{x}) < f(x^*)$$

因 $f(x)$ 为严格拟凸函数, 所以, $\forall \lambda \in (0, 1)$, 有

$$f(\lambda \bar{x} + (1 - \lambda)x^*) < \max\{f(\bar{x}), f(x^*)\} = f(x^*)$$

当 λ 足够小时, $\lambda \bar{x} + (1 - \lambda)x^* \in R \cap N(x^*, \varepsilon), \varepsilon > 0$ 因此, 上式与 x^* 为(3-8)的任一局部最优解矛盾。 ■

定理 3.16 设(3-8)的可行集 R 是开凸集, $f(x)$ 是 R 上的伪凸函数, $x^* \in R$, 若 $\nabla f(x^*) = \mathbf{0}$, 则 x^* 是(3-8)的全局最优解。

证明 由 $\nabla f(x^*) = \mathbf{0}$ 及 $x^* \in R$, 必有

$$\nabla f(x^*)^T (x^* - x) = 0, \quad \forall x \in R$$

因 $f(x)$ 是伪凸函数, 由定义及上式知

$$f(x^*) \leq f(x), \quad \forall x \in R$$

所以 x^* 是(3-8)的全局最优解。 ■

习 题 一

试建立下列问题的数学模型(1 ~ 6)。

1. 合理下料问题

制造产品 D , 需要 A, B, C 三种轴类零件, 其规格和数量如下表:

零件种类	零件规格 (米)	每件产品 D 所需零件数目
A	3.1	1
B	2.1	2
C	1.2	4

各类零件都用 5.5 米长的圆钢来下料, 如果计划生产 D 种产品 100 件, 问如何下料, 才能使所用的圆钢根数最少?

2. 物资调运问题

甲、乙两煤矿, 供应 A, B, C 三个城市, 各煤矿产量和各城市需要量如下表:

煤 矿	日产量 (t)	城 市	日需要量 (t)
甲	200	A	100
		B	150
乙	250	C	200

各煤矿与各城市之间的距离如下表:

煤 矿	距 离 (km)	城 市		
		A	B	C
甲		90	70	100
乙		80	65	80

问应该如何调运煤, 才能在保证各城市需要的情况下, 使运输的总吨公里数最少?

3. 某工地, 在第 1 个月至第 20 个月内需要某种物资, 其第 i 个月的需要

量为 b_i , 自产量为 x_i , 最大生产能力为 a_i , 月末的库存量为 y_i , 最大库存容量为 V , 单位产品的生产费用为 C_0 , 库存费用为 C_1 , 问应如何安排各月的生产量和库存量, 才能使生产费用与库存费用之和为最少?

4. 饮食问题

某医院要为病人确定最经济的饮食方案, 假定 n 种食品中, 第 i 种食品的单价为 c_i 。某病人每天需要第 j 种营养的数量至少为 b_j 个单位。用 a_{ij} 表示每个单位的第 i 种食品含有第 j 种营养成分的数量。问怎样选配食品, 才能保证在满足 m 种营养成分需要的条件下, 使食品的总成本最少?

5. 某工厂生产 n 种产品: A_1, A_2, \dots, A_n , 生产一台第 i 种产品 A_i , 需消耗工时 a_i (h), 能源 b_i (吨标准煤), 可获利润 c_i (元)。该厂全年能耗的限额为 b (吨标准煤), 全年消耗工时的限额为 a (h), 欲确定该厂全年生产各种产品的数量是多少时, 才能使该厂的年利润最高?

6. 设要选定一个供应中心的位置。这个中心是为城市中有固定位置的 m 个用户服务的, 供应的商品可以是水、电、牛奶或其他货物。供应中心的定位准则是使中心到用户的某个距离函数值最小。问如何选定供应中心的位置, 使其到各个用户的最大距离为最小。

7. 证明定理 2.1 及定理 2.2。

8. 画出如下优化问题的可行集 R 和目标函数的等值线图。

$$\begin{cases} \min & f(x) = 2x_1^2 + x_2^2 - 2x_2 \\ \text{s. t.} & x_1^2 + 2x_1 + x_2^2 \leq 8 \\ & x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{cases}$$

9. 求下列优化问题目标函数的梯度和稳定点。

$$(1) \min f(x) = 2x_1^2 + x_2^2 - 8x_1$$

$$(2) \min f(x) = x_1^2 + 2x_2^2 - 4x_1 - 4x_2 - 1$$

10. 证明 n 维欧氏空间的闭超球:

$$S(x^{(0)}, r) = \{x | x \in \mathbf{R}^n, \|x - x^{(0)}\| \leq r\}$$

是凸集。其中 $x^{(0)} \in \mathbf{R}^n$ 为已知向量, $r > 0$ 为常数。

11. 设 $\Omega_1, \Omega_2 \subset \mathbf{R}^n$ 为凸集, 则

$$(1) \Omega_1 + \Omega_2 = \{z | z = x + y, x \in \Omega_1, y \in \Omega_2\};$$

$$(2) \Omega_1 - \Omega_2 = \{z | z = x - y, x \in \Omega_1, y \in \Omega_2\};$$

$$(3) \lambda\Omega_1 = \{z | z = \lambda x, x \in \Omega_1, \lambda \in \mathbf{R}\}$$

均为凸集。

12. 证明定理 3.2。

13. 求下列函数的 Hesse 阵,并说明 $f(x)$ 在凸集 $\Omega \subset \mathbf{R}^n$ 上是否为凸函数,严格凸函数?

$$(1) f(x) = x_1^2 - x_1 + x_2^2 + x_1x_2$$

$$(2) f(x) = x_1^2 + 2x_1x_2 + 2x_2^2 + 4x_2x_3 + 4x_3^2$$

$$(3) f(x) = x_1^2 - 3x_2^2 - 2x_1x_2 + 2x_1x_3 - 6x_2x_3$$

(以下凡带 * 的题,均为较难的题,供选作)

* 14. 设 $f(x)$ 定义在非空凸集 $\Omega \subset \mathbf{R}^n$ 上,证明 $f(x)$ 为凸函数的充要条件是: $\forall x^{(1)}, \dots, x^{(p)} \in \Omega$, 有

$$f\left(\sum_{i=1}^p \lambda_i x^{(i)}\right) \leq \sum_{i=1}^p \lambda_i f(x^{(i)})$$

其中 $\lambda_i \geq 0$ ($i = 1, 2, \dots, p$), $\sum_{i=1}^p \lambda_i = 1$, $p \geq 2$ 。

15. 设 $f(x) = a + b^T x$, 其中 b 为已知的 n 维列向量, a 为已给常数, x 为 n 维列向量,则称 $f(x)$ 为仿射函数,证明仿射函数既是凸函数,又是凹函数。

16. 指出下列函数的凸凹性,并说明理由。

$$(1) f(x) = x_1^2 + 2x_1x_2 + 2x_2^2 - 10x_1 + 5x_2$$

$$(2) f(x) = -x_1^2 - 5x_2^2 + 2x_1x_2 + 10x_1 - 16x_2$$

$$(3) f(x) = x_1x_2 + 2x_1^2 + x_2^2 + 2x_3^2 - 6x_1x_3$$

17. 设 $\Omega = \{(x_1, x_2)^T | -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1\}$, $f(x) = 10 - 2(x_2 - x_1^2)^2$ 在 Ω 上是凹函数吗?为什么?

18. 设 $g: \mathbf{R}^n \rightarrow \mathbf{R}$ 是凹函数, $f(x) = 1/g(x)$ 证明 $f(x)$ 是 $\Omega = \{x | g(x) > 0\}$ 上的凸函数。

19. 举出一个局部极小点不是全局极小点的二元函数的例子。

20. 设目标函数 $f(x)$ 在非空可行集 R 上是严格凸的,则凸规划问题(3-8)的全局极小点是惟一的。

21. 考察问题

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & x \in \Omega \end{aligned}$$

设 $\Omega \subset \mathbf{R}^n$ 为非空凸集, $f(\mathbf{x})$ 为 Ω 上的凸函数, 证明 $f(\mathbf{x})$ 在 Ω 上取得极小值的那些点构成的集合是凸集。

22. 分析下列问题是否为凸规划问题

$$\begin{aligned}
 (1) \quad & \begin{cases} \min & f(\mathbf{x}) = x_1^2 + 2x_1x_2 + 2x_2^2 + 10x_1 + 1 \\ \text{s. t.} & x_1^2 + x_2^2 \leq 16 \\ & x_1 + x_2 \leq 6 \\ & 2x_1 - x_2 \leq 3 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases} \\
 (2) \quad & \begin{cases} \min & f(\mathbf{x}) = x_1^4 - 4x_1x_2 + 2x_2^2 \\ \text{s. t.} & x_1 + x_2 \leq 5 \\ & x_1 - x_2 \leq 2 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}
 \end{aligned}$$

第 2 章 线 性 规 划

前面讲过,线性规划就是由目标函数为决策变量的线性函数和约束条件为线性等式或线性不等式所组成的数学规划,所以它是数学规划中较简单的一类,因而也是最基本的一类数学规划问题。本章主要讨论标准形式的线性规划问题的解法,着重介绍最为常用、且行之有效的解法——单纯形法及对偶单纯形法,最后对 LP 的多项式算法作一简要介绍。

2.1 引言 线性规划的标准形式

线性规划问题的约束条件的一般形式为

$$\begin{aligned} \text{(I)} \quad & \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ \vdots \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \end{cases} \\ \text{(II)} \quad & \begin{cases} a_{i+1,1}x_1 + a_{i+1,2}x_2 + \cdots + a_{i+1,n}x_n = b_{i+1} \\ \vdots \\ a_{t1}x_1 + a_{t2}x_2 + \cdots + a_{tn}x_n = b_t \end{cases} \\ \text{(III)} \quad & \begin{cases} a_{t+1,1}x_1 + a_{t+1,2}x_2 + \cdots + a_{t+1,n}x_n \geq b_{t+1} \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geq b_m \end{cases} \end{aligned}$$

例如 1.1 的例 1.1,其约束条件为

$$\begin{aligned} x_1 + 5x_2 + 10x_3 + 0.5x_4 &\geq 10000 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

它属于(Ⅲ)的形式。而在下面的问题

$$\begin{cases} \min & f(\mathbf{x}) = 3x_1 - x_2 \\ \text{s. t.} & x_1 + x_2 = 10 \\ & x_1 + 2x_2 + x_3 = 25 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

中,其约束条件属于(Ⅱ), (Ⅲ)的形式。

我们指出:一般的线性规划问题,总可以化为如下的标准形式。标准形式的线性规划问题是指在约束条件

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\ x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0 \end{cases} \quad (1-1)$$

之下,求一组数 x_1, x_2, \cdots, x_n , 使目标函数

$$y = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

取最小(大)值,其中 $a_{ij}, b_i, c_j (i = 1, 2, \cdots, m, j = 1, 2, \cdots, n)$ 是给定的常数,且 $b_i \geq 0, i = 1, 2, \cdots, m$ 。

上述线性规划问题可简写为

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{cases} \quad (1-2)$$

其中 $\mathbf{A} = (a_{ij})_{m \times n}$;

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)^T;$$

$$\text{rank}(\mathbf{A}) = m \leq n, \mathbf{c} = (c_1, c_2, \cdots, c_n)^T;$$

$$\mathbf{b} = (b_1, b_2, \cdots, b_m)^T, \text{ 且 } \mathbf{b} \geq \mathbf{0}。$$

下面通过一些例子来说明如何把非标准形式的线性规划问题化为上述的标准形式。

【例 1.1】 将下述线性规划问题化为标准形式。

$$\begin{cases} \max & w = 7x + 12y \\ \text{s. t.} & 9x + 4y \leq 360 \\ & 4x + 5y \leq 200 \\ & 3x + 10y \leq 300 \\ & x \geq 0, y \geq 0 \end{cases}$$

令 $x_4 = x$, $x_5 = y$, 并引入松弛变量 $x_1, x_2, x_3 \geq 0$, 则上述问题可化为下面的 LP 问题:

$$\begin{cases} \max & w = 7x_4 + 5x_5 \\ \text{s. t.} & x_1 + 9x_4 + 4x_5 = 360 \\ & x_2 + 4x_4 + 5x_5 = 200 \\ & x_3 + 3x_4 + 10x_5 = 300 \\ & x_1 \geq 0, x_2 \geq 0, \dots, x_5 \geq 0 \end{cases}$$

对于 1.1 中的例 1.1, 引入剩余变量 $x_5 \geq 0$, 则它可化为如下的 LP 问题:

$$\begin{cases} \min & y = 100x_1 + 300x_2 + 400x_3 + 75x_4 \\ \text{s. t.} & x_1 + 5x_2 + 10x_3 + 0.5x_4 - x_5 = 10000 \\ & x_1 \geq 0, x_2 \geq 0, \dots, x_5 \geq 0 \end{cases}$$

由上面的例子看出: 对于不等式约束, 通过引入松弛变量或剩余变量总可以化为等式约束, 而对于 $b_i < 0$ 的情况, 可以在该式两边乘以 -1 , 使之化为 $b'_i = -b_i > 0$ 的情况。

下面来讨论对自由变量的处理办法。

如果在一个线性规划问题中, 有一部分变量不要求取非负的值, 则这些变量就称为自由变量。例如在线性规划问题(1-2)中, 若不要求 $x_1 \geq 0$, 则 x_1 就是自由变量, 可以用两种办法把它化为标准形式。

1. 令 $x_1 = u_1 - v_1$, 而 $u_1, v_1 \geq 0$, 将它代入上述的 LP 问题中, 则目标函数和约束函数的线性性质仍然保持, 问题转化为 $n + 1$ 个变量 $x_2 \geq 0, x_3 \geq 0, \dots, x_n \geq 0, u_1 \geq 0, v_1 \geq 0$ 的 LP 问题。

2. 从约束方程之一解出 x_1 , 代入其他的约束方程及目标函数, 则问题转化为 $n - 1$ 个变量和 $m - 1$ 个约束的 LP 问题。 ■

【例 1.2】 研究问题

$$\begin{cases} \min & w = x_1 + 3x_2 + 4x_3 \\ \text{s. t.} & \begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

在这里, x_1 为自由变量, 从第一个约束方程 $x_1 + 2x_2 + x_3 = 5$ 中解出 x_1 , 得

$$x_1 = 5 - 2x_2 - x_3$$

将 x_1 的表达式代入目标函数 w 及第二个约束方程中, 得到与之等价的 LP 问题:

$$\begin{cases} \min & w = 5 + x_2 + 3x_3 \\ \text{s. t.} & x_2 + x_3 = 4 \\ & x_2 \geq 0, x_3 \geq 0 \end{cases}$$

对于一般的非标准形式的线性规划问题, 可以像上面的例 1.1 和例 1.2 那样, 加以类似的处理, 就可化为标准形式, 这里就不多讲了。下面主要是讨论标准形式的线性规划问题, 即在约束条件

$$\begin{cases} x_j \geq 0, j = 1, 2, \dots, n \\ \sum_{j=1}^n a_{ij}x_j = b_i, i = 1, 2, \dots, m \end{cases} \quad (1-3)$$

之下, 求向量 $x = (x_1, x_2, \dots, x_n)^T$, 使目标函数 $y = \sum_{i=1}^n c_i x_i$ 取最小(大)值。其中 $m < n$, $b_i \geq 0$ 。而变量的个数 n , 称为线性规划的维数; 等式约束方程的数目 m , 称为线性规划的阶数。满足约束条件 (1-2) 的点 $x = (x_1, x_2, \dots, x_n)^T$ 称为可行点或可行解, 也叫容许解。

令 $A = (a_{ij})_{m \times n}$, 则 (1-3) 中的约束方程组可写成

$$Ax = b \quad (1-4)$$

其中 $x = (x_1, x_2, \dots, x_n)^T$, $b = (b_1, b_2, \dots, b_m)^T$ 。

设 A 的秩为 m , 则可从 A 的 n 列中选出 m 列, 使它们线性无关, 为了书写简便, 不妨设 A 的前 m 列是线性无关的, 即设 $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T (j=1, 2, \dots, m)$ 是线性无关的。令

$$B = (a_1, a_2, \dots, a_m) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{pmatrix}$$

易见矩阵 B 是非奇异的, 因此方程组

$$Bx_B = b \quad (1-5)$$

有惟一解 $x_B = B^{-1}b$, 其中 x_B 是一个 m 维列矢量。

令 $x^T = (x_B^T, 0^T)$, 我们就得到(1-4)的一个解 x 。这里 x 的前 m 个分量等于 x_B 的相应分量, 而后面的 $n-m$ 个分量均为零。

我们称 B 为基或基底, 称这样得到的解矢量 x 为约束方程组(1-4)关于基底 B 的基本解(basic solution), 而与 B 的列相应的 x 的分量 x_i 称为基本变量(basic variable)。当基本解中有一个或一个以上的基本变量 x_i 为零时, 则称这个解为退化的基本解。

当一个可行解 x 又是基本解时, 称它为基本可行解(basic feasible solution), 若它是退化的, 则称它为退化的基本可行解。

显然, 一个基本可行解 x 是一个不超过 m 个正 x_i 的可行解, 而一个非退化的基本可行解 x 是恰有 m 个正 x_i 的可行解。

【例 1.3】 对线性规划问题:

$$\begin{cases} \max & w = 10x_1 + 11x_2 \\ \text{s. t.} & 3x_1 + 4x_2 + x_3 = 9 \\ & 5x_1 + 2x_2 + x_4 = 8 \\ & x_1 - 2x_2 + x_5 = 1 \\ & x_i \geq 0, i = 1, 2, 3, 4, 5 \end{cases}$$

令 $x_1 = x_2 = 0$, 得 $x_3 = 9, x_4 = 8, x_5 = 1$, 即 $x = (0, 0, 9, 8, 1)^T$, 这是一个可行解, 而且是一个基本可行解, 因为 x 满足约束条件, 而 x_3, x_4, x_5 是基本变量, x_1, x_2 是非基本变量, 在这里基向量为

$$a_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a_5 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

即基底 $B = (a_3, a_4, a_5) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, a_3, a_4, a_5 这三个向量显然是线性无关的。

最后我们指出, 一个 m 阶 n 维的线性规划问题, 其基本可行解的个数不超过 C_n^m 个, 而

$$C_n^m = \frac{n!}{m!(n-m)!}$$

这是因为从 n 列中选取 m 个线性无关的列向量的取法至多只有 C_n^m 种。

设 $x = (x_1, x_2, \dots, x_n)^T$ 为线性规划问题的一个可行解, 当它使目标函数 $f(x)$ 达到最小(大)值时, 我们称其为最优可行解, 简称为最优解或解, 而目标函数所达到的最小(大)值称为线性规划问题的值或最优值。

2.2 线性规划的基本定理

在这一节里, 我们来介绍线性规划的基本定理, 从而确定基本可行解在求解线性规划问题中的重要作用, 并阐明基本可行解与可行集的极点之间的关系, 为下一节讲述单纯形法奠定基础。

定义 2.1 称 $H^+ = \{x | x \in \mathbf{R}^n, c^T x \geq b\}$ 为正闭半空间, $H^- = \{x | x \in \mathbf{R}^n, c^T x \leq b\}$ 为负闭半空间。 H^+ 和 H^- 统称为闭半

空间。

定义 2.2 有限个闭半空间的交集称为多面凸集,亦即集合 $S = \{x | Ax \leq b\}$ (其中 $A = (a_{ij})_{m \times n}$) 称为多面凸集或凸多胞形。

易证 H^+, H^- 为凸集,由 1.3 节的定理 3.2 知有限个凸集的交集仍为凸集,所以 S 为凸集。

定义 2.3 有界且非空的多面凸集称为多面凸体或凸多面体。

定义 2.4 设 S 为凸集, $x \in S$, 若对于 x 找不到 $x_1, x_2 \in S$ ($x_1 \neq x_2$), 使 $x = \alpha x_1 + (1 - \alpha)x_2$ ($0 < \alpha < 1$) 成立,则称 x 为凸集 S 的极点,或顶点。

定理 2.1 设 $A = (a_{ij})_{m \times n}$, 其秩为 m , 且 $m < n$, $x = (x_1, x_2, \dots, x_n)^T$, $b = (b_1, b_2, \dots, b_m)^T$, 则向量 x 为凸集

$$R = \{x | Ax = b, x \geq 0\} \quad (2-1)$$

的一个极点的充要条件是 x 为

$$Ax = b, \quad x \geq 0 \quad (2-2)$$

的一个基本可行解。

证明 1° 必要性。即设 x 为 R 的一个极点,则 x 是 (2-2) 的一个基本可行解。

设 x 有 k 个分量大于零,不妨设前 k 个分量 $x_i > 0$ ($i = 1, 2, \dots, k$), 用 a_i 表示 A 的第 i 个列向量,于是

$$\sum_{i=1}^k x_i a_i = b$$

我们来证明列向量 a_1, a_2, \dots, a_k 是线性无关的,从而 $x = (x_1, x_2, \dots, x_k, 0, \dots, 0)^T$ 就是一个基本可行解。

用反证法。假设 a_1, a_2, \dots, a_k 线性相关,则必有 k 个不全为零的数 y_1, y_2, \dots, y_k , 使

$$\sum_{i=1}^k y_i a_i = 0$$

令 $y = (y_1, y_2, \dots, y_k, 0, \dots, 0)^T$, 则

$$Ay = 0$$

因为 $x_i > 0, i = 1, 2, \dots, k$, 所以可选取足够小的 $\varepsilon > 0$, 使得

$$x + \varepsilon y \geq 0, \quad x - \varepsilon y \geq 0$$

显然
$$x = \frac{1}{2}(x + \varepsilon y) + \frac{1}{2}(x - \varepsilon y)$$

因而 x 可以表示为 R 中两个不同点的凸组合, 这是不可能的。因为 x 是 R 的一个极点。所以列向量 a_1, a_2, \dots, a_k 线性无关, 于是 x 一定是 R 的一个基本可行解。

2° 充分性。即设 $x = (x_1, x_2, \dots, x_k, 0, \dots, 0)^T$ 是 (2-2) 的一个基本可行解, 因而列向量 a_1, a_2, \dots, a_k 线性无关, 则 x 一定是 R 的一个极点。

用反证法, 假设 x 不是 R 的一个极点, 则 R 中必有两个不同的点 y, z , 使

$$x = \alpha y + (1 - \alpha)z \quad (0 < \alpha < 1)$$

即
$$x_j = \alpha y_j + (1 - \alpha)z_j, \quad j = 1, 2, \dots, n$$

当 $j = k + 1, \dots, n$ 时, 因为 $x_j = 0$, 所以上式变为

$$0 = \alpha y_j + (1 - \alpha)z_j, \quad j = k + 1, \dots, n$$

又因为 $y_j \geq 0, z_j \geq 0, 0 < \alpha < 1$, 所以

$$y_j = z_j = 0, \quad j = k + 1, \dots, n$$

又因为 $y \in R, z \in R$, 所以 $Ay = b, Az = b$

即有
$$y_1 a_1 + y_2 a_2 + \dots + y_k a_k = b$$

$$z_1 a_1 + z_2 a_2 + \dots + z_k a_k = b$$

两式相减得

$$\sum_{j=1}^k (y_j - z_j) a_j = 0$$

由于 $y \neq z$, 故 $y_j - z_j$ 不全为零, 于是 a_1, a_2, \dots, a_k 线性相关, 这与充分性的题设相矛盾。因此 x 是 R 是一个极点。 ■

定理 2.2 线性规划的基本定理

设给定的线性规划问题为

$$\begin{cases} \min & w = \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geqslant \mathbf{0} \end{cases} \quad (2-3)$$

其中 $A = (a_{ij})_{m \times n}$, 其秩为 m , 且 $m < n$ 。

(1) 若(2-3)有可行解, 那么它必有基本可行解;

(2) 若(2-3)有最优可行解, 则它必有最优的基本可行解。

定理 2.2 说明: 在寻求线性规划问题(2-3)的最优解时, 只需要研究基本可行解就行了。也就是说, 如果(2-3)有最优解, 则一定在凸集 $R = \{\mathbf{x} | \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}$ (可行集)的极点上达到。

证明 (1) 用 $\mathbf{a}_i (i = 1, 2, \dots, n)$ 表示 A 的第 i 列, 设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 是(2-3)的一个可行解, 于是有

$$\sum_{i=1}^n x_i \mathbf{a}_i = \mathbf{b}$$

设 \mathbf{x} 有 p 个分量 $x_i > 0$, 为简单起见, 不妨设 \mathbf{x} 的前 p 个分量 $x_i > 0, i = 1, 2, \dots, p$, 因此有

$$\sum_{i=1}^p x_i \mathbf{a}_i = \mathbf{b} \quad (2-4)$$

下面根据列矢量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 是否线性相关分两种情况进行证明。

情形 1 当 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性无关时, 显然 $p \leqslant m$; 若 $p = m$, 则可行解 \mathbf{x} 就是基本可行解, 无需再证。

若 $p < m$, 由于 A 的秩为 m , 所以可从 A 的剩下的 $n - p$ 个列向量 \mathbf{a}_j 中找出 $m - p$ 个向量, 与 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 一起组成一个线性无关向量组。令与这 $m - p$ 个向量对应的变量为零, 这样就得到(2-3)的一个退化的基本可行解 \mathbf{x} 。

情形 2 当 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_p$ 线性相关时, 则存在 p 个不全为零的

数 y_1, y_2, \dots, y_p , 使

$$\sum_{i=1}^p y_i a_i = 0 \quad (2-5)$$

我们总可以假定至少有一个 $y_i > 0$, 令

$$\varepsilon = \min_{1 \leq i \leq p} \left\{ \frac{x_i}{y_i} \mid y_i > 0 \right\}$$

用 ε 乘(2-5)式, 并从(2-4)式中减去它, 就得到

$$\sum_{i=1}^p (x_i - \varepsilon y_i) a_i = b \quad (2-6)$$

令 $y = (y_1, y_2, \dots, y_p, 0, \dots, 0)^T$, 则 $x - \varepsilon y$ 满足 $A(x - \varepsilon y) = b$, $x - \varepsilon y \geq 0$, 即 $x - \varepsilon y$ 是(2-3)的一个可行解, 而且至多有 $p - 1$ 个分量 $x_i - \varepsilon y_i$ 大于零。如果这些大于零的分量 $x_i - \varepsilon y_i$ 所对应的列向量 a_i 是线性无关的, 则问题转化为情形 1, 因而定理的结论成立。反之, 则可重复上述过程, 直到剩下的大于零的分量 x_i 所对应的列矢量组是线性无关时为止。

对于定理 2.2 的(2)的证明, 可以像(1)的证明一样类似地给出, 作为习题留给读者。 ■

推论 1 若由(2-3)确定的可行集

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

是非空的, 则它至少有一个极点。

由定理 2.2 的(1)和定理 2.1 即可推出。

推论 2 如果一个线性规划问题(2-3)存在有限的最优解, 则至少存在一个可行集 R 的极点的有限最优解。

推论 3 对应于(2-3)的可行集 R 至多具有有限多个极点。

因为 2.1 中已经指出: 一个 m 阶 n 维的线性规划问题, 其基本可行解的个数不超过 C_n^m 个。

而 $C_n^m = \frac{n!}{m!(n-m)!}$ 是有限的, 所以可行集 R 至多具有有限多个极点。

上面的定理和推论指出：一个线性规划问题(2-3)的可行集 R 如果是非空的,那么该线性规划问题的基本可行解 x 一定存在;如果线性规划问题(2-3)的最优解存在,那么它一定在(2-3)的可行集 R 的一个极点(即(2-3)的基本可行解)上达到。因此,寻求线性规划问题(2-3)的解,只要在基本可行解(R 的极点)中寻找就行了,而极点的个数是有限的,这就从理论上保证了我们有可能在有限步内求得线性规划问题的最优解。

2.3 单纯形法

单纯形法(Simplex Method)是求解线性规划问题的一种通用的有效算法,于1947年由 G. B. Dantzig 首先提出。几十年的计算实践说明,它不仅是求解线性规划的基本方法,而且是整数规划和非线性规划某些算法的基础。为了保证单纯形法是有限步收敛的,必须避免迭代过程的循环。避免循环的办法有:① 摄动法^[14,15];② 字典序法^[18];③ Bland 法^[57]。本节介绍单纯形法的基本原理和计算方法,下一节对单纯形法作进一步的说明和补充。

2.3.1 单纯形法的基本思想

前面已经指出:一个线性规划问题若有最优解,则一定有最优的基本可行解,而且基本可行解的个数是有限的。因此,一个求解线性规划问题的直观想法是,把所有的基本可行解求出来,并求出其相应的目标函数值,相互比较,即可求得其中相应目标函数值最小(大)的最优解了。当线性规划问题的阶数 m 与维数 n 很小时,这种方法(枚举法)是可行的,比如 $n \leq 6$, $m \leq 5$, 需要计算的基本可行解的个数不会超过 20 个,是可行的。当数 n 和 m 较大时,其计算量迅速增长,以至成为不可能。比如当 $n = 8$, $m \leq 7$ 时,需要

计算的基本可行解的个数就有可能达到 70 个;而当 $n = 10, m \leq 9$ 时,要求的基本可行解的个数,有可能达到 252 个。因此我们需要寻求其他的计算量小的方法,这就导致了单纯形法和其他算法的研究。

单纯形法的基本思想是:给出一种规则,使由 LP 问题一个基本可行解(极点)转移到另一个基本可行解,目标函数值是减小的,而且两个基本可行解之间的转换是容易实现的。经过有限次迭代,即可求得所需的最优基本可行解。

2.3.2 基本解的转换

设线性规划问题为

$$(P) \begin{cases} \min & w = \sum_{i=1}^n c_i x_i \\ \text{s. t.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\ & x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0 \end{cases} \quad (3-1)$$

其中 $m < n, b_i \geq 0, i = 1, 2, \cdots, m$ 。

令

$$A = (a_{ij})_{m \times n}, \mathbf{x} = (x_1, x_2, \cdots, x_n)^T$$

则(3-1)可写为

$$A\mathbf{x} = \mathbf{b} \quad (3-1')$$

假定 A 的秩为 m ,不妨设 A 的前 m 列线性无关。则(3-1)式总可化为

我们称(3-3)为约束方程组的规范形式。常简写为

其相应的基本解为

令 $\mathbf{x}_B = (x_1, x_2, \dots, x_m)^T$, $\mathbf{x}_N = (x_{m+1}, \dots, x_n)^T$, 则 $x_i (i = 1, 2, \dots, m)$ 为基本变量, $x_i (i = m+1, \dots, n)$ 为非基本变量, 记 $\mathbf{y}_0 = (y_{10}, \dots, y_{m0})^T$, 因而 (3-3) 也可写为

假定我们要由基本解 $x = (y_{10}, \dots, y_{m0}, 0, \dots, 0)^T$ 转换到另一个基本解,例如要把基本变量 x_1, x_2, \dots, x_m 中的某一个 x_p 变为非基本变量,而把某一个非基本变量 x_q 变为基本变量。即在规范形式的方程组(3-3)中,用非基本变量 x_q 来代替基本变量 x_p , $1 \leq p \leq m$, 当且仅当 $y_{pq} \neq 0$ 时,才能做到这一点。用 y_{pq} 除第 p 个方程,使 x_q 的系数变为 1,然后用其他行中的每一行减去第 p 行的一个适当的倍数,使其他各行中 x_q 的系数变为零,这样就完成了用 x_q 代替 x_p 的过程。于是得到新的规范形式约束方程组的系数为

我们称系数 y_{pq} 为一次转换所取的主元素。下面通过一个具体例子来说明上述过程。

【例 3.1】 设规范形式的约束方程组为

$$\begin{cases} x_1 & + x_4 + x_5 - x_6 = 5 \\ x_2 & + 2x_4 - 3x_5 + x_6 = 3 \\ x_3 - x_4 + 2x_5 - x_6 = -1 \end{cases}$$

求基本变量为 x_4, x_5, x_6 的基本解。

不难看出：基本解为 $\mathbf{x} = (5, 3, -1, 0, 0, 0)^T$ ，基本变量为

x_1, x_2, x_3 。

(1) 用非基本变量 x_4 代替基本变量 x_1 ，用黑体字表示一次转换所取的主元，在这里主元为 1。

(2) 用 x_5 代替 x_2 ，所取的主元为 -5 。

(3) 用 x_6 代替 x_3 ，所取的主元为 $-1/5$ 。

计算表格如下：

x_1	x_2	x_3	x_4	x_5	x_6	b
1	0	0	1	1	-1	5
0	1	0	2	-3	1	3
0	0	1	-1	2	-1	-1
1	0	0	1	1	-1	5
-2	1	0	0	-5	3	-7
1	0	1	0	3	-2	4
3/5	1/5	0	1	0	-2/5	18/5
2/5	-1/5	0	0	1	-3/5	7/5
-1/5	3/5	1	0	0	-1/5	-1/5
1	-1	-2	1	0	0	4
1	-2	-3	0	1	0	2
1	-3	-5	0	0	1	1

这样我们就由旧的基本解

$$\mathbf{x} = (5, 3, -1, 0, 0, 0)^T$$

经过转换得到了一个新的基本解

$$\mathbf{x}^* = (0, 0, 0, 4, 2, 1)^T$$

其基本变量为 x_4, x_5, x_6 。

上面所述的把(3-1)的一个基本解转换为另一个基本解的过程也可以作另外一种解释。

用 a_1, a_2, \dots, a_n 表示 A 的列向量, 则(3-1)可写成

$$\sum_{i=1}^n x_i a_i = b$$

其中 $a_i \in \mathbf{R}^m$, $b \in \mathbf{R}^m$ 。

设 a_1, a_2, \dots, a_m 为 \mathbf{R}^m 的一组基, 由于 $b \in \mathbf{R}^m$, 所以有

$$\sum_{i=1}^m x_i a_i = b$$

令 $x = (x_1, x_2, \dots, x_m, 0, 0, \dots, 0)^T$, 则 x 就是(3-1)的一个基本解, 对于 a_1, a_2, \dots, a_n 中的任一个矢量 a_j , 有表达式

$$a_j = \sum_{i=1}^m y_{ij} a_i$$

假定我们要把某一个非基矢量 a_q 引入基, 而将某个基矢量 a_p 替换出去。称 a_q 为进基矢量, a_p 为离基矢量。由于

$$a_q = \sum_{i=1}^m y_{iq} a_i = \sum_{i=1, i \neq p}^m y_{iq} a_i + y_{pq} a_p$$

当 $y_{pq} \neq 0$ 时, 有

$$a_p = \frac{1}{y_{pq}} \left(a_q - \sum_{i=1, i \neq p}^m y_{iq} a_i \right)$$

则 $a_1, \dots, a_{p-1}, a_q, a_{p+1}, \dots, a_m$ 为 \mathbf{R}^m 的一组新基(见习题二第8题), 任一矢量 $a_j \in \mathbf{R}^m$ 的表达式为

$$\begin{aligned} a_j &= \sum_{i=1, i \neq p}^m y_{ij} a_i + y_{pj} a_p \\ &= \sum_{i=1, i \neq p}^m y_{ij} a_i + \frac{y_{pj}}{y_{pq}} \left(a_q - \sum_{i=1, i \neq p}^m y_{iq} a_i \right) \\ &= \sum_{i=1, i \neq p}^m \left(y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq} \right) a_i + \frac{y_{pj}}{y_{pq}} a_q \end{aligned}$$

令 $y_{ij} - \frac{y_{pj}}{y_{pq}}y_{iq} = y'_{ij} (i \neq p)$, $y_{pj}/y_{pq} = y'_{pj}$, 则 $a_j = \sum_{i=1}^m y'_{ij}a_i$ 。而

$$\begin{cases} y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}}y_{iq}, & i \neq p \\ y'_{pj} = y_{pj}/y_{pq} \end{cases} \quad (3-5)$$

公式(3-5)与(3-4)是一样的。这样我们就完成了从一个基本解到另一个基本解的转换。

2.3.3 基本可行解的转换

在上一段,我们说明了从(3-1)的一个基本解转换到另一个基本解的方法。如果转换前的基本解同时又是一个可行解(满足(3-2)),那么经过转换后,一般来说,可行性将不再保持,即(3-2)将不满足,由于进基矢量和离基矢量在上一段是可以任意确定的。下面我们来说明:只要按照某种规则来确定离基矢量,即按照某种规则来确定哪个基本变量变为非基本变量,就可以使可行性得到保持,从而使我们可以从一个基本可行解转换到另一个基本可行解。

非退化假定:在以下的讨论中,我们总是假定:

$$Ax = b, \quad x \geq 0$$

的每一个基本可行解都是非退化的基本可行解。

这个假定贯穿在整个单纯形法的推导过程之中。但是这个假定应看做是仅仅为了方便而提出的,因为所有的推导论证都可以推广到包含退化的情况。

离基矢量的确定。

设 $x = (x_1, x_2, \dots, x_m, 0, \dots, 0)^T$ 为 LP 问题(P)的一个基本可行解,在非退化的假定下,必有 $x_i > 0$ ($i = 1, 2, \dots, m$)。假定进基矢量为 a_k , $k > m$, 由于 a_1, a_2, \dots, a_m 为 R^m 的一组基,所以

$$a_k = \sum_{i=1}^m y_{ik} a_i$$

用 ε ($\varepsilon > 0$) 乘上式得

$$\sum_{i=1}^m \varepsilon y_{ik} a_i = \varepsilon a_k \quad (3-6)$$

因为 x 是一个基本可行解, 所以有

$$\sum_{i=1}^m x_i a_i = b \quad (3-7)$$

由式(3-7) 减去式(3-6) 得

$$\sum_{i=1}^m (x_i - \varepsilon y_{ik}) a_i + \varepsilon a_k = b$$

由于 $x_i > 0$, 所以只要 $\varepsilon > 0$ 足够小, 就可使 $x_i - \varepsilon y_{ik} \geq 0$, 因而

$$\tilde{x} = (x_1 - \varepsilon y_{1k}, \dots, x_m - \varepsilon y_{mk}, 0, \dots, 0, \varepsilon, 0, \dots, 0)^T$$

是 (P) 的一个可行解, 但不一定是基本解。

如果 $\min_{1 \leq i \leq m} \left\{ \frac{x_i}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{x_r}{y_{rk}} (1 \leq r \leq m)$ 存在, 那么取

$$\varepsilon = \min_{1 \leq i \leq m} \left\{ \frac{x_i}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{x_r}{y_{rk}} \quad (3-8)$$

则 \tilde{x} 至多有 m 个分量大于零, 因而 \tilde{x} 是一个基本可行解, 这时取 a_r ($1 \leq r \leq m$) 为离基矢量, 就可实现从一个基本可行解 x 到另一个基本可行解 \tilde{x} 的转换。如果所有的 $y_{ik} \leq 0$ ($i = 1, 2, \dots, m$), 那么可任取 $\varepsilon > 0$, 而 \tilde{x} 的 $m+1$ 个分量将随 ε 的增大而增大(或保持常数), 而 \tilde{x} 总是 (P) 的可行解, 因而 (P) 的可行集

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

是无界的。

【例 3.2】 设 LP 问题的约束条件为

$$\begin{cases} x_1 & + 2x_4 + 4x_5 + 6x_6 = 4 \\ & x_2 & + x_4 + 2x_5 + 3x_6 = 3 \\ & & x_3 - x_4 + 2x_5 + x_6 = 1 \\ x_1 \geq 0, x_2 \geq 0, \dots, x_6 \geq 0 \end{cases}$$

易见

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

为一组基。 $x = (4, 3, 1, 0, 0, 0)^T$ 为一个基本可行解。假定我们选择 a_4 为进基矢量, 下面来确定离基矢量 a_r 。

计算比值 $\frac{x_i}{y_{ik}} = \frac{y_{i0}}{y_{ik}}$, 这里 $k = 4, i = 1, 2, 3$ 。因为

$$\frac{y_{10}}{y_{14}} = \frac{4}{2} = 2, \quad \frac{y_{20}}{y_{24}} = \frac{3}{1} = 3$$

$$\varepsilon = \min_{1 \leq i \leq 3} \left\{ \frac{y_{i0}}{y_{i4}} \mid y_{i4} > 0 \right\} = \frac{y_{10}}{y_{14}} = 2$$

即 $r = 1$, 所以离基矢量为 a_1 , 也就是说要用非基本变量 x_4 来代替基本变量 x_1 , 新的表格为

a_1	a_2	a_3	a_4	a_5	a_6	b
1/2	0	0	1	2	3	2
-1/2	1	0	0	0	0	1
1/2	0	1	0	4	4	3

因此例 3.2 的 LP 问题的相应基本可行解为 $x = (0, 1, 3, 2, 0, 0)^T$ 。

2.3.4 最优基本可行解的确定

上面我们已经说明了从任一个基本可行解出发, 只要 y_{ik} 至少有一个为正, 按照使

$$\min_{1 \leq i \leq m} \left\{ \frac{y_{i0}}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{y_{r0}}{y_{rk}}, \quad 1 \leq r \leq m$$

的下标 r 决定离基矢量 a_r , 就可以实现从一个基本可行解到另一个基本可行解的转换。下面, 我们来说明如何确定进基矢量 a_k , 使从一个基本可行解转换到另一个基本可行解时, 目标函数值是减小的。

设线性规划问题 (P) 的约束方程组已化为如下的规范形式:

$$x_i + \sum_{j=m+1}^n y_{ij}x_j = y_{i0}, \quad i = 1, 2, \dots, m \quad (3-9)$$

其中 $y_{i0} > 0$, $i = 1, 2, \dots, m$ 。而 $x = (y_{i0}, \dots, y_{m0}, 0, \dots, 0)^T$ 为问题 (P) 的一个基本可行解, 其对应的目标函数值为

$$z_0 = \mathbf{c}_B^T \mathbf{x}_B = \sum_{k=1}^m c_k y_{k0}$$

其中 $\mathbf{c}_B = (c_1, c_2, \dots, c_m)^T$, 而与问题 (P) 的任一可行解 x 对应的目标函数值为

$$z = \mathbf{c}^T \mathbf{x} = \sum_{k=1}^n c_k x_k \quad (3-10)$$

利用(3-9) 可将(3-10) 改写为

$$\begin{aligned} z &= \sum_{k=1}^m c_k (y_{k0} - \sum_{j=m+1}^n y_{kj}x_j) + \sum_{j=m+1}^n c_j x_j \\ &= \sum_{k=1}^m c_k y_{k0} + \sum_{j=m+1}^n c_j x_j - \sum_{j=m+1}^n (\sum_{k=1}^m c_k y_{kj}) x_j \\ &= z_0 + \sum_{j=m+1}^n (c_j - \sum_{k=1}^m c_k y_{kj}) x_j \\ &= z_0 + \sum_{j=m+1}^n (c_j - z_j) x_j \end{aligned} \quad (3-11)$$

其中

$$z_0 = \mathbf{c}_B^T \mathbf{x}_B, \quad z_j = \mathbf{c}_B^T \mathbf{y}_j = \sum_{k=1}^m c_k y_{kj} \quad (3-12)$$

公式(3-11) 说明了 LP 问题 (P) 的任一可行解与它的一个基本可行解的目标函数值之间的关系。若对于某个 $j(m+1 \leq j \leq n)$, 使

$c_j - z_j$ 是负的, 那么当 x_j 由 0 增加到某一正值时, 目标函数 z 就减小, 这样我们就可以得到一个较优的解。 ■

定理 3.1 已知一个非退化的基本可行解, 其目标函数值为 z_0 , 假定对于某一个 j , 有 $c_j - z_j < 0$, 那么存在一个可行解, 使其对应的目标函数值 $z < z_0$ 。如果能用非基矢量 a_j 代替原基中的某一矢量 a_r , 而产生一个新的基本可行解, 则这个新的解将使 $z < z_0$ 。如果不能用 a_j 来代替而产生一个基本可行解, 则可行集 R 是无界的, 而且目标函数值可任意的小 (趋向 $-\infty$)。

证明 设 $x = (x_1, x_2, \dots, x_m, 0, \dots, 0)^T$ 是一个非退化的基本可行解, 相应的目标函数值为 z_0 , 假定 $c_j - z_j < 0$ ($m+1 \leq j \leq n$), 那么用上一段讲述的方法可以构造一个形式为

$$\begin{aligned}\tilde{x} &= (x_1 - \varepsilon y_{1k}, \dots, x_m - \varepsilon y_{mk}, 0, 0, \dots, 0, \varepsilon, 0, \dots, 0)^T \\ &= (x'_1, x'_2, \dots, x'_m, 0, \dots, 0, x'_j, 0, \dots, 0)^T\end{aligned}$$

的新的可行解, 且 $x'_j > 0$ 。将它代入式(3-11)得

$$z - z_0 = (c_j - z_j)x'_j < 0$$

所以, 当 $\varepsilon > 0$ 时, 对任意这样的解总有

$$z < z_0$$

成立。

如果至少有一个 $y_{ik} > 0$, 象上一段讲的那样, 选取 $\varepsilon > 0$, 就能用非基矢量 a_j 来代替原基中的某一矢量 a_r , 从而得到一个新的基本可行解; 如果所有的 $y_{ik} \leq 0$, 那么 \tilde{x} 的 $m+1$ 个分量将随 ε 的增大而增大 (或保持常数), 但至少有一个分量随之增大而增大, 因此可行集 R 是无界的, 而且目标函数值无下界, 即 $z \rightarrow -\infty$ 。 ■

定理 3.1 说明, 若 $c_j - z_j < 0$, 则目标函数值就可以继续减小。剩下的问题是: 对于所有的 j , $c_j - z_j \geq 0$ 是否能保证这个基本可行解就是最优解? 下面的定理 3.2 回答了这个问题。

定理 3.2 如果某一个基本可行解,对所有的 j ,都有 $c_j - z_j \geq 0$,那么这个解就是最优解。

证明 因为对任一可行解 x ,必有 $x_i \geq 0, 1 \leq i \leq n$,所以,若对所有的 j ,总有 $c_j - z_j \geq 0$,则 $z \geq z_0$ 。 ■

由于 $c_j - z_j = r_j$ 在单纯形法中起着重要的作用,所以我们称它为相对成本系数或检验数,有的文献也称为判别数。

2.3.5 单纯形法的计算步骤和例子

(1) 把一般的线性规划问题化为标准形式。

(2) 建立初始单纯形表。

(3) 若所有的检验数 $r_j = c_j - z_j \geq 0$,就得到了一个最优解,运算结束;否则转到第4步。

(4) 当有多于一个的 $r_j < 0$ 时,可选其中任一列矢量 a_j 为进基矢量,通常是选使 $\min\{r_j | r_j < 0\} = r_k$ 的 a_k 为进基矢量。

(5) 对所有的 $y_{ik} > 0$, 计算比值 $\frac{y_{i0}}{y_{ik}}$, 设 $\min\left\{\frac{y_{i0}}{y_{ik}} \mid y_{ik} > 0\right\} = \frac{y_{r0}}{y_{rk}}$, 则主元素为 y_{rk} , 当 $A = (I_m, N)$ 时, x_r 为离基变量, 转第6步。

若所有的 $y_{ik} \leq 0$, 则 LP 问题的可行集 R 是无界的, 而且目标函数是无界的, 运算结束。

(6) 以 y_{rk} 为主元素, 进行一次 Gauss 消元(或称为旋转), 从而求得一个新的基本可行解, 返回到第3步。

下面给出用单纯形法求解线性规划问题的程序框图(图2-1)。

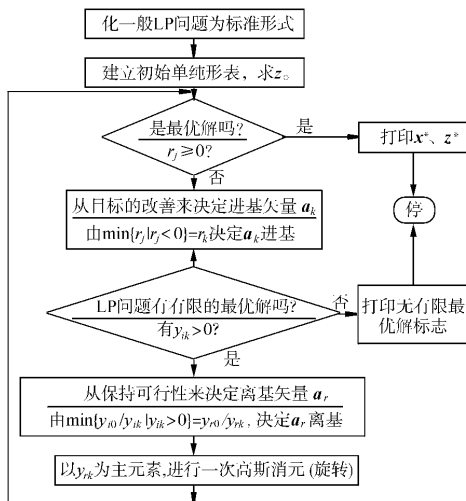


图 2-1 单纯形法程序框图

【例 3】 用单纯形法求解 LP 问题

$$\begin{cases} \min & z = -(3x_1 + x_2 + 3x_3) \\ \text{s. t.} & \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 3 \\ 2 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 2 \\ 5 \\ 6 \end{pmatrix}, \quad \mathbf{x} \geq \mathbf{0} \end{cases}$$

解 (1) 化为标准形式:

$$\begin{cases} \min & z = -(3x_1 + x_2 + 3x_3) \\ \text{s. t.} & \begin{aligned} 2x_1 + x_2 + x_3 + x_4 &= 2 \\ x_1 + 2x_2 + 3x_3 + x_5 &= 5 \\ 2x_1 + 2x_2 + x_3 + x_6 &= 6 \\ x_i &\geq 0, i = 1, 2, 3, 4, 5, 6 \end{aligned} \end{cases}$$

(2) 建立初始单纯形表:

基矢量	a_1	$a_2 \downarrow$	a_3	$\uparrow a_4$	a_5	a_6	b
a_4	2	1	1	1	0	0	2
a_5	1	2	3	0	1	0	5
a_6	2	2	1	0	0	1	6
检验数	-3	-1	-3	0	0	0	0

$z_0 = c_B^T x_B = 0$, 在检验数行与 b 列的交叉处, 填上 $(-z_0)$ 。

(3) 计算检验数 $r_j = c_j - z_j$ 。

因为 $z_j = c_B^T y_j$, $j = 1, 2, 3$, 此处 $c_B^T = (c_4, c_5, c_6) = (0, 0, 0)$, 所以 $z_j = 0$, $i = 1, 2, 3$ 。由 $r_j = c_j - z_j$ 得

$$r_1 = -3, r_2 = -1, r_3 = -3$$

(4) 决定进基矢量 a_k 。

由于 $r_j < 0$, $j = 1, 2, 3$ 。所以 a_1, a_2, a_3 均可选做进基矢量, 为减少除法运算, 我们选 a_2 为进基矢量。

(5) 决定离基矢量 a_r 和主元素 y_{rk} 。

首先计算比值 y_{i0}/y_{ik} , 这里 $k = 2$, $i = 1, 2, 3$ 。因为

$$\frac{y_{10}}{y_{12}} = \frac{2}{1} = 2, \frac{y_{20}}{y_{22}} = \frac{5}{2} = 2.5, \frac{y_{30}}{y_{32}} = \frac{6}{2} = 3$$

故
$$\min \left\{ \frac{y_{i0}}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{y_{10}}{y_{12}}$$

因此主元素为 y_{12} , a_4 为离基矢量。

(6) 以 y_{12} 为主元素(在表中用黑体字标记), 进行一次 Gauss 消元, 以求得一个新的基本可行解。具体计算可列表如下:

基矢量	a_1	a_2	$\downarrow a_3$	a_4	$a_5 \uparrow$	a_6	b
a_2	2	1	1	1	0	0	2
a_5	-3	0	1	-2	1	0	1
a_6	-2	0	-1	-2	0	1	2
检验数	-1	0	-2	1	0	0	2

再像上面的第 3 步一样计算检验数 r_j , 填入上表中。因为

$$\mathbf{c}_B^T = (-1, 0, 0)$$

$$\mathbf{y}_1 = \begin{pmatrix} 2 \\ -3 \\ -2 \end{pmatrix}, \mathbf{y}_3 = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \mathbf{y}_4 = \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix}$$

故

$$z_1 = \mathbf{c}_B^T \mathbf{y}_1 = -2, z_3 = -1, z_4 = -1$$

$$r_1 = c_1 - z_1 = -3 - (-2) = -1$$

$$r_3 = -3 - (-1) = -2$$

$$r_4 = 0 - (-1) = 1, z_0 = \mathbf{c}_B^T \mathbf{y}_0 = -2$$

注意表中所填的数是目标函数的负值, 即 $-z_0$ 。

由于 $\min\{r_j | r_j < 0\} = r_3$, 所以进基矢量为 \mathbf{a}_3 。再计算比值 y_{i0}/y_{ik} , 这里 $k = 3, i = 1, 2, 3$ 。因为

$$y_{10}/y_{13} = 2/1 = 2, y_{20}/y_{23} = 1/1 = 1$$

故

$$\min \left\{ \frac{y_{i0}}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{y_{20}}{y_{23}}$$

因此主元为 y_{23} , 离基矢量为 \mathbf{a}_5 。在表中用黑体字来指示所选的主元。

基矢量	$\downarrow \mathbf{a}_1$	$\uparrow \mathbf{a}_2$	\mathbf{a}_3	\mathbf{a}_4	\mathbf{a}_5	\mathbf{a}_6	\mathbf{b}
\mathbf{a}_2	5	1	0	3	-1	0	1
\mathbf{a}_3	-3	0	1	-2	1	0	1
\mathbf{a}_6	-5	0	0	-3	2	0	4
检验数	-7	0	0	-3	2	0	4

因

$$\mathbf{c}_B^T = (-1, -3, 0)$$

故

$$z_1 = -5 + 9 = 4, r_1 = -7$$

$$z_4 = 3, r_4 = -3, z_5 = -2, r_5 = 2, z_0 = -4$$

因此进基矢量为 \mathbf{a}_1 。

计算比值 y_{i0}/y_{ik} , 这里 $k = 1, y_{10}/y_{11} = 1/5$, 而 y_{21}, y_{31} 均为负, 故主元素为 $y_{21} = 5$, 离基矢量为 \mathbf{a}_2 。

计算表格为:

基向量	a_1	a_2	a_3	a_4	a_5	a_6	b
a_1	1	1/5	0	3/5	-1/5	0	1/5
a_3	0	3/5	1	-1/5	2/5	0	8/5
a_6	0	1	0	-1	0	1	4
检验数	0	7/5	0	6/5	3/5	0	27/5

$$c_B^T = (-3, -3, 0), z_2 = -12/5, r_2 = -1 + 12/5 = 7/5$$

$$z_4 = -6/5, r_4 = 0 - (-6/5) = 6/5$$

$$z_5 = -3/5, r_5 = 0 - (-3/5) = 3/5, z_0 = -27/5$$

因为 $r_j \geq 0, j = 1, 2, \dots, 6$

所以最优解为 $x^* = (1/5, 0, 8/5, 0, 0, 4)^T$, 最优值为

$$z^* = -27/5$$

2.4 关于单纯形法的说明和补充

上一节介绍了单纯形法的基本原理和计算步骤,这一节将对单纯形法作一些说明和补充。

2.4.1 初始基本可行解的确定

用单纯形法求解线性规划问题,首先需要确定一个初始基本可行解,以便使迭代计算能够进行下去。这里给出两种处理方法,它们都是通过引入人工变量的办法来获得一个初始基本可行解的。

1. 大 M 法

设要求解的线性规划问题的标准形式如下:

$$\begin{cases} \min & w = c^T x \\ \text{s. t.} & Ax = b \\ & x \geq 0 \end{cases} \quad (4-1)$$

为了获得一个初始基本可行解,引入人工变量 y_1, y_2, \dots, y_m , 令 $y = (y_1, y_2, \dots, y_m)^T$, 考察另一个线性规划问题:

$$\begin{cases} \min & z = c^T x + ME^T y \\ \text{s. t.} & Ax + y = b \\ & x \geq 0, y \geq 0 \end{cases} \quad (4-2)$$

其中 $M > 0$ 是一个充分大的数, $E = (1, 1, \dots, 1)^T \in \mathbf{R}^m$ 。问题 (4-1) 与 (4-2) 的解之间有如下的关系。

定理 4.1 设 $\begin{bmatrix} x^* \\ y^* \end{bmatrix}$ 是问题 (4-2) 的最优解, 若 $y^* = 0$, 则 x^* 是问题 (4-1) 的最优解; 若 $y^* \neq 0$, 则问题 (4-1) 没有可行解。反之, 若 x^* 是问题 (4-1) 的最优解, 则 $\begin{bmatrix} x^* \\ 0 \end{bmatrix}$ 是问题 (4-2) 的最优解。

证明 设 x 是问题 (4-1) 的任一可行解, 则 $\begin{bmatrix} x \\ 0 \end{bmatrix}$ 是问题 (4-2) 的一个可行解, 于是当 $y^* = 0$ 时, 有

$$z^* = c^T x^* \leq c^T x = w$$

所以 x^* 是问题 (4-1) 的最优解, 且 $z^* = w^*$ 。

若 $y^* \neq 0$, 假设问题 (4-1) 有一可行解 x , 则 $\begin{bmatrix} x \\ 0 \end{bmatrix}$ 是问题 (4-2) 的一个可行解, 于是

$$z^* = c^T x^* + ME^T y^* \leq c^T x$$

因为 $y^* \neq 0$, 当 $M > 0$ 充分大时, 上式不可能成立, 所以这时问题 (4-1) 没有可行解。

设 $\begin{bmatrix} x \\ y \end{bmatrix}$ 是问题 (4-2) 的任一可行解, 下面分两种情况来证明

$\begin{bmatrix} x^* \\ 0 \end{bmatrix}$ 是问题 (4-2) 的最优解。

1° 若 $y = 0$, 则 x 是问题(4-1)的一个可行解, 于是 $w^* = c^T x^* \leq c^T x$ 。

2° 若 $y \neq 0$, 由于 $M > 0$ 可以充分大, 所以必有

$$c^T x^* \leq c^T x + ME^T y$$

因此 $\begin{pmatrix} x^* \\ 0 \end{pmatrix}$ 是问题(4-2)的最优解, 最优值为 $z^* = w^* = c^T x^*$ 。

上述定理说明: 为了获得问题(4-1)的最优解, 可以直接去求解问题(4-2), 而问题(4-2)的初始基本可行解是明显的。其中的 M 较难确定, 但在手算时不需要具体确定, 只要把它作为一个足够大的正数来处理就行了(参看下例)。

【例 4.1】 用单纯形法求解线性规划问题:

$$\begin{cases} \min & w = -x_1 - 2x_2 - 3x_3 + x_4 \\ \text{s. t.} & x_1 + 2x_2 + 3x_3 = 15 \\ & 2x_1 + x_2 + 5x_3 = 20 \\ & x_1 + 2x_2 + x_3 + x_4 = 10 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad (4-3)$$

解 1° 添加人工变量 $x_5 \geq 0, x_6 \geq 0$, 构造如下的线性规划问题:

$$\begin{cases} \min & z = -x_1 - 2x_2 - 3x_3 + x_4 + M(x_5 + x_6) \\ \text{s. t.} & x_1 + 2x_2 + x_3 + x_4 = 10 \\ & x_1 + 2x_2 + 3x_3 + x_5 = 15 \\ & 2x_1 + x_2 + 5x_3 + x_6 = 20 \\ & x_i \geq 0, i = 1, 2, 3, 4, 5, 6 \end{cases} \quad (4-4)$$

2° 建立初始单纯形表, 确定初始的基本可行解为 $x = (0, 0, 0, 10, 15, 20)^T$, 其相应的目标函数值为

$$z_0 = 10 + M(15 + 20) = 10 + 35M$$

序号	基 矢量	c_B	y_0	y_1	y_2	$y_3 \downarrow$	y_4	y_5	$\uparrow y_6$
				- 1	- 2	- 3	1	M	M
1	y_4	1	10	1	2	1	1	0	0
2	y_5	M	15	1	2	3	0	1	0
3	y_6	M	20	2	1	5	0	0	1
函数值与 检验数			- 10 - $35M$	- 2 - $3M$	- 4 - $3M$	- 4 - $8M$	0	0	0

3° 确定进基矢量 y_k 。因为

$$z_j = c_B^T y_j, \quad r_j = c_j - z_j$$

故

$$z_1 = 3M + 1, \quad r_1 = -2 - 3M$$

$$z_2 = 3M + 2, \quad r_2 = -4 - 3M$$

$$z_3 = 8M + 1, \quad r_3 = -4 - 8M$$

$$z_4 = 1, r_4 = 0, z_5 = M, r_5 = 0, z_6 = M, r_6 = 0$$

因为 $\min\{r_j | r_j < 0\} = r_3$, 故进基矢量为 y_3 。

4° 确定主元素 y_{rk} 和离基矢量 y_r 。因为

$$y_{10}/y_{13} = 10, y_{20}/y_{23} = 5, y_{30}/y_{33} = 4$$

$$\min \left\{ \frac{y_{i0}}{y_{i3}} \mid y_{i3} > 0 \right\} = \frac{y_{30}}{y_{33}}$$

所以主元素为 $y_{33} = 5$, 离基矢量为 y_6 。

5° 以 y_{33} 为主元素, 进行一次 Gauss 消元(旋转), 然后返回第 3 步, 并重复上述过程, 直到求得最优解, 具体计算过程见下表:

序号	基变量	c_B	y_0	y_1	$\downarrow y_2$	y_3	y_4	$\uparrow y_5$	y_6	y_2 进基 y_5 离基 主元素为 $7/5$
				-1	-2	-3	1	M	M	
1	y_4	1	6	$3/5$	$9/5$	0	1	0	$-1/5$	
2	y_5	M	3	$-1/5$	$7/5$	0	0	1	$-3/5$	
3	y_3	3	4	$2/5$	$1/5$	1	0	0	$1/5$	
函数值与检验数			$-3M+6$	$\frac{M-2}{5}$	$-\frac{7M+16}{5}$	0	0	0	$\frac{8M+4}{5}$	
1	y_4	1	$15/7$	$6/7 \downarrow$	0	0	$1 \uparrow$	$-9/7$	$4/7$	y_1 进基 y_4 离基 主元素为 $y_{11} = \frac{6}{7}$
2	y_2	-2	$15/7$	$-1/7$	1	0	0	$5/7$	$-3/7$	
3	y_3	-3	$25/7$	$3/7$	0	1	0	$-1/7$	$2/7$	
函数值与检验数			$90/7$	$-6/7$	0	0	0	$\frac{7M+16}{7}$	$\frac{7M-4}{7}$	
1	y_1	-1	$5/2$	1	0	0	$7/6$	$-3/2$	$2/3$	
2	y_2	-2	$5/2$	0	1	0	$1/6$	$1/2$	$-1/3$	
3	y_3	-3	$5/2$	0	0	1	$-1/2$	$1/2$	0	
函数值与检验数			15	0	0	0	4	$M+1$	M	

所以 LP 问题(4-4) 的最优解为 $\tilde{x}^* = (5/2, 5/2, 5/2, 0, 0, 0)^T$, 最优值为 $z^* = -15$ 。于是 LP 问题(4-3) 的最优解为 $x^* = (5/2, 5/2, 5/2, 0)^T$, 最优值为 $w^* = -15$ 。

2. 二阶段法

设要求解的 LP 问题为(4-1), 引入人工变量 y_1, y_2, \dots, y_m , 令 $y = (y_1, y_2, \dots, y_m)^T$, 构造一个辅助的 LP 问题

$$\begin{cases} \min & z = \sum_{i=1}^m y_i \\ \text{s. t.} & Ax + y = b \\ & x \geq 0, y \geq 0 \end{cases} \quad (4-5)$$

问题(4-1) 与(4-5) 的解之间有如下关系。 ■

定理 4.2 若问题(4-5) 的最优基本可行解为 $\begin{bmatrix} x^* \\ 0 \end{bmatrix}$, 则 x^* 为问题(4-1) 的一个基本可行解; 若问题(4-5) 的最优基本可行解为

$\begin{pmatrix} x^* \\ y^* \end{pmatrix}$, 且 $y^* \neq 0$, 则问题(4-1) 没有可行解。

证明 前一结论显然成立, 下面只证后者(用反证法)。设 x 为问题(4-1) 的一个可行解, 则 $\begin{pmatrix} x \\ 0 \end{pmatrix}$ 为问题(4-5) 的一个可行解, 于是

$$z^* = \sum_{i=1}^m y_i^* \leq z_0 = 0$$

因为 $y^* \geq 0$, 且 $y^* \neq 0$, 所以

$$z^* = \sum_{i=1}^m y_i^* > 0$$

这与上式 $z^* = \sum_{i=1}^m y_i^* \leq 0$ 矛盾, 故问题(4-1) 没有可行解。 ■

定理 4.2 说明: 为了获得问题(4-1) 的一个初始基本可行解, 可以先求解辅助问题(4-5), 由它的最优基本可行解, 即可得到问题(4-1) 的一个基本可行解。所以可将求解问题(4-1) 分为两个阶段:

阶段 1: 求解辅助 LP 问题(4-5), 它的初始基本可行解是明显的。

阶段 2: 求解原 LP 问题(4-1)。

上述方法通常称为二阶段法。

【例 4.2】 求解如下 LP 问题:

$$\begin{cases} \min & w = 4x_1 + x_2 + x_3 \\ \text{s. t.} & 2x_1 + x_2 + 2x_3 = 4 \\ & 3x_1 + 3x_2 + x_3 = 3 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

解 阶段 1: 求解辅助 LP 问题

$$\begin{cases} \min & z = y_1 + y_2 \\ \text{s. t.} & 2x_1 + x_2 + 2x_3 + y_1 = 4 \\ & 3x_1 + 3x_2 + x_3 + y_2 = 3 \\ & x_i \geq 0 \ (i = 1, 2, 3), \ y_i \geq 0 \ (i = 1, 2) \end{cases}$$

用单纯形法易得最优解为 $x_1 = 1/2, x_2 = 0, x_3 = 3/2, y_1 = y_2 = 0$, 最优值为 $z^* = 0$ 。

阶段 2: 求解原 LP 问题。

阶段 1 最后一步的单纯形表为

基 矢 量	a_1	a_2	a_3	a_4	a_5	b
a_3	0	$-3/4$	1	$3/4$	$-1/2$	$3/2$
a_1	1	$5/4$	0	$-1/4$	$1/2$	$1/2$
r_j, z	0	0	0	1	1	0

原 LP 问题的初始单纯形表是将上表中人工变量所在的两列去掉, 并将目标函数值 $z = 0$ 去掉, 将原 LP 问题的目标函数值 w 计算出来, 将 $-w$ 填在 z 的位置即得

基 矢 量	a_1	a_2	a_3	b
a_3	0	$-3/4$	1	$3/2$
a_1	1	$5/4$	0	$1/2$
r_j, z	0	$-13/4$	0	$-7/2$

此处, $c_B = (1, 4)^T$, $x_B = (3/2, 1/2)^T$, 故 $w = 7/2$

$$r_2 = c_2 - z_2 = 1 - (1, 4) \begin{pmatrix} -3/4 \\ 5/4 \end{pmatrix} = 1 - \frac{17}{4} = -\frac{13}{4}$$

然后按单纯形法进行迭代计算, 只需一步即可求得原问题的最优解为 $x^* = (0, 2/5, 9/5)^T$ 及最优值 $w^* = 11/5$ 。例 4.2 的详细计算过程留给读者练习。

关于如何求初始的基本可行解, 还有人在继续研究, 可参看文献 55 和文献 56。

2.4.2 单纯形法的矩阵形式

前一节讲的单纯形法比较通俗易懂,但书写较繁,进行理论分析不大方便,这里介绍一下单纯形法的矩阵形式。

设 LP 问题为

$$\begin{cases} \min & z = \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases} \quad (4-6)$$

其中矩阵 A 的秩 $r(A) = m < n$, $\mathbf{b} \geq \mathbf{0}$ 。

设 B 为基,不妨设它是由 A 的前 m 列组成的,把 $A, \mathbf{x}, \mathbf{c}$ 分块为

$$A = (B, N), \mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ \mathbf{c}_N \end{bmatrix}$$

于是问题(4-6)可改写为

$$\begin{cases} \min & z = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N \\ \text{s. t.} & \mathbf{Bx}_B + \mathbf{Nx}_N = \mathbf{b} \\ & \mathbf{x}_B, \mathbf{x}_N \geq \mathbf{0} \end{cases} \quad (4-6')$$

令 $\mathbf{x}_N = \mathbf{0}$, 则得 $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$, 于是得到与基 B 对应的基本可行解(假设它是可行的)为

$$\mathbf{x}^{(0)} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{pmatrix}$$

与 $\mathbf{x}^{(0)}$ 对应的目标函数值为 $z_0 = \mathbf{c}_B^T \mathbf{x}_B = \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b}$ 。

对任意的 \mathbf{x}_N , 由(4-6')可得

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx}_N$$

于是得到与它相应的目标函数值为

$$\begin{aligned} z &= \mathbf{c}_B^T (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{Nx}_N) + \mathbf{c}_N^T \mathbf{x}_N \\ &= \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N \\ &= z_0 + \mathbf{r}_N^T \mathbf{x}_N \end{aligned} \quad (4-7)$$

其中 $z_0 = \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b}$ 是与基本可行解 $\mathbf{x}^{(0)}$ 对应的目标函数值。

$$\mathbf{r}_N^T = \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} = \mathbf{c}_N^T - \boldsymbol{\pi}^T \mathbf{N} \quad (4-8)$$

$$\boldsymbol{\pi}^T = \mathbf{c}_B^T \mathbf{B}^{-1} \quad (4-9)$$

称 $\boldsymbol{\pi}$ 为单纯形乘子矢量, \mathbf{r}_N 是与非基变量对应的检验数构成的矢量, 用 \mathbf{r}_B 表示与基变量对应的检验数构成的矢量, 由前述可知

$\mathbf{r}_B = \mathbf{0}$ 。称 $\mathbf{r} = \begin{bmatrix} \mathbf{r}_B \\ \mathbf{r}_N \end{bmatrix}$ 为相对成本矢量或检验矢量。由它的分量即可确定哪个矢量应该进基。于是与基 B 对应的矩阵形式的单纯形表可写成

$$T(B) = \left\{ \begin{array}{c|c|c} \mathbf{I}_m & \mathbf{B}^{-1} \mathbf{N} & \mathbf{B}^{-1} \mathbf{b} \\ \hline \mathbf{0} & \mathbf{c}_N^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} & -\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} \end{array} \right\} \quad (4-10)$$

因为

$$\begin{aligned} \mathbf{B}^{-1} \mathbf{A} &= \mathbf{B}^{-1} (\mathbf{B}, \mathbf{N}) = (\mathbf{I}_m, \mathbf{B}^{-1} \mathbf{N}) \\ \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A} - \mathbf{c}^T &= \mathbf{c}_B^T (\mathbf{I}_m, \mathbf{B}^{-1} \mathbf{N}) - (\mathbf{c}_B^T, \mathbf{c}_N^T) \\ &= (\mathbf{c}_B^T, \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N}) - (\mathbf{c}_B^T, \mathbf{c}_N^T) \\ &= (\mathbf{0}, \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{N} - \mathbf{c}_N^T) \end{aligned}$$

所以矩阵形式的单纯形表也可写成为

$$T(B) = \left\{ \begin{array}{c|c} \mathbf{B}^{-1} \mathbf{A} & \mathbf{B}^{-1} \mathbf{b} \\ \hline \mathbf{c}^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A} & -\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{b} \end{array} \right\} \quad (4-10')$$

而检验矢量 \mathbf{r} 可表示为

$$\mathbf{r}^T = \mathbf{c}^T - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{A} = \mathbf{c}^T - \boldsymbol{\pi}^T \mathbf{A} \quad (4-11)$$

2.4.3 修正单纯形法

对单纯形法进行仔细分析就会发现：在迭代过程中，我们所关心的主要是如下的数据：

$$\begin{aligned} (1) \quad r_j &= c_j - \mathbf{c}_B^T \mathbf{y}_j = c_j - \mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{a}_j = c_j - \boldsymbol{\pi}^T \mathbf{a}_j \\ \min \quad \{r_j | r_j < 0\} &= r_k \end{aligned}$$

$$(2) \quad B^{-1}a_k = (y_{1k}, y_{2k}, \dots, y_{mk})^T$$

$$x_B = B^{-1}b = (y_{10}, y_{20}, \dots, y_{m0})^T$$

用 $r_j (j = 1, 2, \dots, n)$ 来判别现行解是否为最优解, 若 $r \geq 0$ 不成立, 则利用这两组数来确定主元素 y_{rk} 和进基、离基矢量, 进行换基迭代, 就可得到新基 \bar{B} 。而这些数据, 只要知道 B^{-1} , 就可由问题的初始数据计算出来。同时, 当基 B 变换到新基 \bar{B} 时, \bar{B}^{-1} 容易由 B^{-1} 求得, 下面来导出具体的计算公式。

不妨设 $B = (a_1, a_2, \dots, a_m)$, 新基 \bar{B} 为

$$\bar{B} = (a_1, a_2, \dots, a_{r-1}, a_k, a_{r+1}, \dots, a_m)$$

因

$$B^{-1}B = B^{-1}(a_1, a_2, \dots, a_m)$$

$$= (B^{-1}a_1, B^{-1}a_2, \dots, B^{-1}a_m) = I_m,$$

所以

$$B^{-1}\bar{B} = B^{-1}(a_1, a_2, \dots, a_{r-1}, a_k, a_{r+1}, \dots, a_m)$$

$$= (B^{-1}a_1, \dots, B^{-1}a_{r-1}, B^{-1}a_k, B^{-1}a_{r+1}, \dots, B^{-1}a_m)$$

因

$$B^{-1}a_k = y_k = (y_{1k}, y_{2k}, \dots, y_{mk})^T$$

第 r 列

因

$$B^{-1}\bar{B} = \begin{pmatrix} 1 & & & y_{1k} & & \\ & \ddots & & \vdots & & \\ & & 1 & y_{r-1,k} & & \\ & & & y_{rk} & & \\ & & & y_{r+1,k} & 1 & \\ & & & \vdots & & \ddots \\ & & & y_{mk} & & & 1 \end{pmatrix} \triangleq B_{rk} \quad (4-12)$$

由式(4-12) 易得

$$E_{rk} \triangleq B_{rk}^{-1} = \begin{matrix} & \text{第 } r \text{ 列} \\ \left(\begin{array}{ccccccc} 1 & & & -y_{1k}/y_{rk} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -y_{r-1,k}/y_{rk} & & & \\ & & & 1/y_{rk} & & & \\ & & & -y_{r+1,k}/y_{rk} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -y_{mk}/y_{rk} & & & 1 \end{array} \right) \end{matrix}$$

所以 $(B^{-1}\bar{B})^{-1} = \bar{B}^{-1}B = B_{rk}^{-1} = E_{rk}$, 由此得到

$$\bar{B}^{-1} = E_{rk}B^{-1} \quad (4-13)$$

$$x_{\bar{B}} = \bar{B}^{-1}b = E_{rk}B^{-1}b = E_{rk}x_B \quad (4-14)$$

综上所述,就可给出修正单纯形法的计算步骤如下:

设已求得 LP 问题(4-6)的一个初始基本可行解,与它相对应的基 B (称为可行基)的逆为 B^{-1} , 求出 $x_B = B^{-1}b = y_0 = (y_{10}, y_{20}, \dots, y_{m0})^T$ 。

(1) 求单纯形乘子矢量 $\pi^T = c^T B^{-1}$ 及现行的检验矢量 r 。

$$r_j = c_j - \pi^T a_j, \quad j = 1, 2, \dots, n$$

若 $r \geq 0$, 则计算结束,现行解为最优解;否则转第 2 步。

(2) 令 $r_k = \min\{r_j | r_j < 0\}$, 则 a_k 为进基矢量,计算

$$y_k = B^{-1}a_k = (y_{1k}, y_{2k}, \dots, y_{mk})^T$$

若所有的 $y_{ik} \leq 0$, 则 LP 问题没有有限的最优解,计算结束;若至少有一个 $y_{ik} > 0$, 则转第 3 步。

(3) 由 $\min\{y_{i0}/y_{ik} | y_{ik} > 0\} = y_{r0}/y_{rk}$ 决定 a_r 离基,主元素为 y_{rk} 。

(4) 形成初等变换矩阵 E_{rk} 。

(5) 根据(4-13)将 B^{-1} 修改为 $\bar{B}^{-1} = E_{rk}B^{-1}$ 。

(6) 根据(4-14)将 x_B 修改为 $x_{\bar{B}} = E_{rk}x_B$, 然后返回第 1 步。

【例 4.3】 用修正单纯形法求解问题

$$\left\{ \begin{array}{ll} \min & z = -(3x_1 + x_2 + 3x_3) \\ \text{s.t.} & \begin{array}{ll} 2x_1 + x_2 + x_3 + x_4 & = 2 \\ x_1 + 2x_2 + 3x_3 + x_5 & = 5 \\ 2x_1 + 2x_2 + x_3 + x_6 & = 6 \\ x_i \geq 0, i = 1, 2, \dots, 6 \end{array} \end{array} \right. \quad (4-15)$$

解 易见问题(4-15)的初始可行基为

$$B = (a_4, a_5, a_6) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

显然

$$B^{-1} = B$$

$$c_B = (0, 0, 0)^T, x_B = (x_4, x_5, x_6)^T = (2, 5, 6)^T$$

第一次迭代

$$(1) \pi^T = c_B^T B^{-1} = (0, 0, 0), \pi^T a_j = 0, j = 1, 2, \dots, 6$$

$$r_1 = c_1 = -3, r_2 = -1, r_3 = -3$$

$$r_j = 0, j = 4, 5, 6$$

$$(2) r_k = r_1 = r_3 = -3, \text{这里取 } k = 1, \text{确定 } a_1 \text{ 进基。}$$

$$y_1 = B^{-1}a_1 = (2, 1, 2)^T$$

$$(3) \text{因 } y_{10}/y_{11} = 2/2 = 1, y_{20}/y_{21} = 5, y_{30}/y_{31} = 3,$$

$$\text{所以 } y_{r0}/y_{r1} = y_{10}/y_{11} = 1, r = 1, a_4 \text{ 是离基矢量,}$$

主元素为 $y_{11} = 2$ 。

$$(4) \text{形成 } E_{11} = \begin{pmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

$$(5) \text{将 } B^{-1} \text{ 修改为 } E_{11}B^{-1} = \begin{pmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

$$(6) \text{将 } x_B \text{ 修改为 } E_{11}x_B = (1, 4, 4)^T.$$

第二次迭代

$$(1) \pi^T = c_B^T B^{-1} = (-3, 0, 0) B^{-1} = (-3/2, 0, 0)$$

$$r_1 = r_5 = r_6 = 0, r_2 = c_2 - \pi^T a_2 = -1 - (-3/2) = 1/2$$

$$r_3 = -3 - (-3/2) = -3/2, r_4 = 3/2$$

(2) $r_k = r_3$, 即 $k = 3$, 故 a_3 是进基矢量。

$$y_3 = B^{-1} a_3 = (1/2, 5/2, 0)^T$$

$$(3) \text{ 因 } y_{10}/y_{13} = 1/(1/2) = 2, y_{20}/y_{23} = 4/(5/2) = 8/5,$$

故 $y_{r0}/y_{r3} = y_{20}/y_{23} = 8/5$, 即 $r = 2$, a_5 是离基矢量, 主元素为 $y_{23} = 5/2$ 。

$$(4) \text{ 形成 } E_{23} = \begin{pmatrix} 1 & -1/5 & 0 \\ 0 & 2/5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(5) 将 B^{-1} 修改为

$$E_{23} B^{-1} = \begin{pmatrix} 1 & -1/5 & 0 \\ 0 & 2/5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3/5 & -1/5 & 0 \\ -1/5 & 2/5 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

(6) 将 x_B 修改为

$$E_{23} x_B = \begin{pmatrix} 1 & -1/5 & 0 \\ 0 & 2/5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 1/5 \\ 8/5 \\ 4 \end{pmatrix}$$

第三次迭代

$$\begin{aligned} \pi^T &= c_B^T B^{-1} = (-3, -3, 0) \begin{pmatrix} 3/5 & -1/5 & 0 \\ -1/5 & 2/5 & 0 \\ -1 & 0 & 1 \end{pmatrix} \\ &= (-6/5, -3/5, 0) \end{aligned}$$

$$r_1 = r_3 = r_6 = 0, r_2 = c_2 - \pi^T a_2 = -1 - (-12/5) = 7/5$$

$$r_4 = 0 - (-6/5) = 6/5, r_5 = 0 - (-3/5) = 3/5$$

由于 $r_j \geq 0$ ($j = 1, 2, \dots, 6$), 所以现行解

$$\mathbf{x}^* = (1/5, 0, 8/5, 0, 0, 4)^T$$

即为最优解, 问题(4-15)的最优值为 $z^* = -27/5$, 这与 2.3 中例 3.3 的结果相同。

由上述可见: 修正单纯形法的存贮量和计算量都要比单纯形法少, 当 n 远大于 m 时更为明显。所以在计算机上算题时, 一般多采用修正单纯形法。一般的求解 LP 问题的程序, 也是用修正单纯形法编写的。但在手算时, 修正单纯形法的优点并不明显。

2.4.4 退化与循环

在上一节的讨论中, 我们总是假定: 线性规划问题(4-6)的基本可行解都是非退化的, 这时称(4-6)是非退化的, 因此所有的基变量 $x_i = y_{i0} > 0$ ($i \in J_B$), 其中 J_B 为基变量的指标集。但在某些 LP 问题中, 常有一些基变量 x_i 取零值, 这时基本可行解就是退化的。在选主元素和确定离基矢量时, 取

$$\varepsilon = \min\{y_{i0}/y_{ik} \mid y_{ik} > 0\} = y_{r0}/y_{rk}$$

时, 就可能出现 $\varepsilon = 0$, 因而

$$\tilde{\mathbf{x}} = (x_1 - \varepsilon y_{1k}, \dots, x_m - \varepsilon y_{mk}, 0, \dots, 0, \varepsilon, 0, \dots, 0)^T$$

$$= (x_1, \dots, x_m, 0, \dots, 0, 0, 0, \dots, 0)^T = \mathbf{x}$$

$$\tilde{z} = f(\tilde{\mathbf{x}}) = z_0 = f(\mathbf{x})$$

即在新的基本可行解 $\tilde{\mathbf{x}}$ 处, 目标函数值 \tilde{z} 在与原基本可行解 \mathbf{x} 处的目标函数值相同, 如果这种情况不断发生, 就可能使可行基重复出现, 使迭代计算循环不止, 以致得不到需要的最优解。1951 年, A. J. Hoffman 举出了第一个在迭代过程中可能出现循环的例子, 这是一个包含 11 个变量, 3 个约束方程的 LP 问题。1955 年, E. M. L. Beale 给出了一个较简单的例子, 它包含 7 个变量, 3 个约束方程, 我们将在下面介绍。

在用单纯形法进行迭代计算时, 上面都采用如下规则:

1° 若有两个以上检验数 $r_j < 0$ 时, 令

$$r_k = \min\{r_j | r_j < 0\}$$

则选 a_k 为进基矢量。

2° 若 $\min\{y_{i0}/y_{ik} | y_{ik} > 0\} = y_{r_1 0}/y_{r_1 k} = \cdots = y_{r_t 0}/y_{r_t k}$, 且 $r_1 < r_2 < \cdots < r_t$, 则选 $y_{r_1 k}$ 为主元素, a_{r_1} 为离基矢量。

用上述规则, 前一例经 9 次迭代后, 开始循环; 后一例经 6 次迭代后, 开始循环。

为了避免迭代过程出现循环, 弥补单纯形法的缺陷, 保证单纯形法在有限步内收敛, 已经提出三种处理办法, 下面作一简介。

1. 摄动法

对 LP 问题(4-1), 设 $\varepsilon > 0$ 充分小, 用 y_i 表示 A 的第 i 列, 令 $b(\varepsilon) = b + \varepsilon y_1 + \varepsilon^2 y_2 + \cdots + \varepsilon^n y_n$ 。用 $b(\varepsilon)$ 代替(4-1)中的 b , 得到摄动后的问题:

$$\begin{cases} \min & w = c^T x \\ \text{s. t.} & Ax = b(\varepsilon) \\ & x \geq 0 \end{cases} \quad (4-16)$$

设 x 是(4-1)的一个基本可行解, 它对应的基是 $B = (y_1, y_2, \cdots, y_m)$, 则

$$x_1 y_1 + x_2 y_2 + \cdots + x_m y_m = b$$

如果 x 是退化的, 那么它的基变量中就有一部分为零, 也就是说, 当 b 由向量组 y_1, y_2, \cdots, y_m 用非负系数线性表示时, 其中有的系数等于零。若对 b 作一微小变动, 即用 $b(\varepsilon)$ 代替 b , 使 $b(\varepsilon)$ 用非负系数由 y_1, y_2, \cdots, y_m 线性表示时, 其系数全部为正的, 则 x 就是非退化的。进而若变动 b 能使所有的基都具有上述性质时, 则变动 b 以后的 LP 问题(4-16)就是非退化的, 因而可用前述方法求得其最优解, 然后再把 $b(\varepsilon)$ 变回来, 就可得到原问题(4-1)的解, 这就是摄动法的基本思想。该法由 A. Charnes 于 1952 年首先提出^[58]。其详细介绍可参看文献 15 和文献 18。

2. 字典序法

字典序法是由 G. B. Dantzig 等人于 1954 年首先提出的,它可以看做是摄动法的一种变形,可参看文献 15 和文献 18。

3. Bland 法^[57]

1976 年, R. G. Bland 在第九届国际数学规划会上,给出了一种避免循环的新方法。他证明:按如下两条规则去做,就可避免迭代计算中的循环。

1° 若 $\min\{i | r_i < 0\} = k$, 则选 y_k 为进基矢量。

2° 若 $\min\{y_{i0}/y_{ik} | y_{ik} > 0\} = y_{r0}/y_{rk}$, 则选 y_{rk} 为主元素, y_r 为离基矢量,若有多于一个指标可选时,总选取指标最小的。

上述的三种方法中以 Bland 法较为简单,但迭代次数要多一些。

下面来介绍 E. M. Beale 于 1955 年给出的一个在迭代过程中出现循环的例子。

【例 4.4】 求解如下的 LP 问题:

$$\left\{ \begin{array}{ll} \min & z = -3x_1/4 + 150x_2 - x_3/50 + 6x_4 \\ \text{s. t.} & x_1/4 - 60x_2 - x_3/25 + 9x_4 + x_5 = 0 \\ & x_1/2 - 90x_2 - x_3/50 + 3x_4 + x_6 = 0 \\ & x_3 + x_7 = 1 \\ & x_i \geq 0, \quad i = 1, 2, \dots, 7 \end{array} \right.$$

(4-17)

解 易见 $x^{(0)} = (0, 0, 0, 0, 0, 0, 1)^T$ 是(4-17)的一个退化的基本可行解,基 $B = (y_5, y_6, y_7)$, 若用前述的单纯形法求解(采用规则 1°, 2°), 在迭代过程中将出现循环:

$$\begin{aligned} (5, 6, 7) &\rightarrow (1, 6, 7) \rightarrow (1, 2, 7) \rightarrow (3, 2, 7) \rightarrow \\ &\quad (3, 4, 7) \rightarrow (5, 4, 7) \rightarrow (5, 6, 7) \end{aligned}$$

其中(5, 6, 7)是基 $B = (y_5, y_6, y_7)$ 的简写,其余可类推。

如果我们保留规则 1°, 将规则 2° 修改为

3° 若 $\min\{y_{i0}/y_{ik} \mid y_{ik} > 0\} = y_{r_1 0}/y_{r_1 k} = \cdots = y_{r_t 0}/y_{r_t k}$, 且 $r_1 < r_2 < \cdots < r_t$, 则选 $y_{r_t k}$ 为主元素, y_{r_t} 为离基矢量。

采用规则 1°, 3°, 其迭代计算过程列在下表中。

基矢量	$y_1 \downarrow$	y_2	y_3	y_4	y_5	$y_6 \uparrow$	y_7	b	说 明
y_5	1/4	-60	-1/25	9	1	0	0	0	y_1 进基
$\leftarrow y_6$	1/2	-90	-1/50	3	0	1	0	0	y_6 离基
y_7	0	0	1	0	0	0	1	1	主元素
检验数	-3/4	150	-1/50	6	0	0	0	0	为 1/2
y_5	0	-15	-3/100 \downarrow	15/2	1	-1/2	0 \uparrow	0	y_3 进基
y_1	1	-180	-1/25	6	0	2	0	0	y_7 离基
$\leftarrow y_7$	0	0	1	0	0	0	1	1	主元素
检验数	0	15	-1/20	21/2	0	3/2	0	0	为 $y_{33} = 1$
y_5	0	-15	0	15/2	1	-1/2	3/100	3/100	
y_1	1	-180	0	6	0	2	1/25	1/25	
y_3	0	0	1	0	0	0	1	1	
检验数	0	15	0	21/2	0	3/2	1/20	1/20	

从表中可见：经二次迭代，所有的检验数 $r_j \geq 0$, $j = 1, 2, \dots, 7$ 。所以(4-17)的最优解为

$$\mathbf{x}^* = (1/25, 0, 1, 0, 3/100, 0, 0)^T$$

最优值为 $z^* = -1/20$ 。

在第一次迭代中，因为

$$y_{10}/y_{11} = 0/(1/4) = 0, \quad y_{20}/y_{21} = 0/(1/2) = 0$$

所以

$$\min\{y_{i0}/y_{i1} \mid y_{i1} > 0\} = y_{10}/y_{11} = y_{20}/y_{21} = 0$$

按规则 3°, 主元素应选 $y_{21} = 1/2$, 离基矢量为 y_6 。

若采用 Bland 规则, 则迭代次数要多一些。

2.5 线性规划的对偶理论与对偶单纯形法

每一个线性规划问题, 都有一个被称为对偶的线性规划问题与它相对应, 二者之间有密切的联系。因为它们可以看做是对同一

个问题从不同的角度所进行的分析与研究。它们是根据同样的条件和数据构成的两个不同的问题。一个是求目标函数的最小值,另一个是求另一个目标函数的最大值,它们之间存在着确定的关系,当有限的最优解存在时,二者的最优值必定相等。研究两个互为对偶的线性规划问题的解之间的这些关系就构成了线性规划的对偶理论,对偶理论对线性规划的解法也有重要的作用。因此对偶理论是线性规划的重要内容之一。限于时间与篇幅,这里只能作初步的介绍,有兴趣的读者可参看文献 14 和文献 15。

2.5.1 对偶线性规划问题

设有线性规划问题:

$$\begin{cases} \min & f(x) = c^T x \\ \text{s. t.} & Ax \geq b \\ & x \geq 0 \end{cases} \quad (5-1)$$

问题(5-1)的对偶线性规划问题定义为

$$\begin{cases} \max & g(\lambda) = b^T \lambda \\ \text{s. t.} & \lambda^T A \leq c^T \\ & \lambda \geq 0 \end{cases} \quad (5-2)$$

其中 $A = (a_{ij})_{m \times n}$, $x, c \in \mathbf{R}^n$, $b, \lambda \in \mathbf{R}^m$ 。

通常我们称(5-1)为原始规划问题,(5-2)为对偶规划问题。

由上述定义可以看出:从已知的原始问题(5-1)构成对偶问题(5-2)的方法是:

1. 交换常矢量 c, b 的位置,变矢量 x 用 λ 替换;
2. 改变约束不等式的不等号方向;
3. 把 \min 换为 \max
4. 交换 A 与变矢量的位置,并按需要做适当的转置。

任何一个线性规划问题的对偶问题可以通过先把原始问题化为上述形式来求得。

【例 5.1】 求标准形式的线性规划问题的对偶问题。

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases} \quad (5-3)$$

与(5-3)等价的线性规划问题是

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} \geq \mathbf{b} \\ & -\mathbf{Ax} \geq -\mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases}$$

其系数阵为 $\begin{pmatrix} \mathbf{A} \\ -\mathbf{A} \end{pmatrix}$, 因此, 上述问题又可写成

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \begin{pmatrix} \mathbf{A} \\ -\mathbf{A} \end{pmatrix} \mathbf{x} \geq \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix} \\ & \mathbf{x} \geq \mathbf{0} \end{cases}$$

其对偶问题为

$$\begin{cases} \max & (\mathbf{u}^T, \mathbf{v}^T) \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix} \\ \text{s. t.} & (\mathbf{u}^T, \mathbf{v}^T) \begin{pmatrix} \mathbf{A} \\ -\mathbf{A} \end{pmatrix} \leq \mathbf{c}^T \\ & \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \end{cases} \quad (5-4)$$

或写成

$$\begin{cases} \max & (\mathbf{u}^T \mathbf{b} - \mathbf{v}^T \mathbf{b}) \\ \text{s. t.} & (\mathbf{u}^T - \mathbf{v}^T) \mathbf{A} \leq \mathbf{c}^T \\ & \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \end{cases}$$

令 $\mathbf{u} - \mathbf{v} = \mathbf{w}$, 则得

$$\begin{cases} \max & \mathbf{w}^T \mathbf{b} \\ \text{s. t.} & \mathbf{w}^T \mathbf{A} \leq \mathbf{c}^T \end{cases} \quad (5-4')$$

而 \mathbf{w} 为自由变矢量, 即没有非负性要求。

通常称(5-2)为(5-1)的对称形式的对偶线性规划问题, 而称(5-4')为(5-3)的非对称形式的对偶线性规划问题。由上述可见: 如果在原始问题中, 某个约束是等式约束, 那么在对偶问题中, 变矢量 λ 的相应分量就是自由变量, 反之亦真。 ■

【例 5.2】 求下列线性规划问题的对偶问题:

$$\begin{cases} \min & z = 2x_1 + 3x_2 + 5x_2 + 2x_4 + 3x_5 \\ \text{s. t.} & x_1 + x_2 + 2x_3 + x_4 + 3x_5 \geq 4 \\ & 2x_1 - 2x_2 + 3x_3 + x_4 + x_5 \geq 3 \\ & x_i \geq 0, i = 1, 2, \dots, 5 \end{cases}$$

其对偶问题为

$$\begin{cases} \max & w = 4\lambda_1 + 3\lambda_2 \\ \text{s. t.} & \lambda_1 + 2\lambda_2 \leq 2 \\ & \lambda_1 - 2\lambda_2 \leq 3 \\ & 2\lambda_1 + 3\lambda_2 \leq 5 \\ & \lambda_1 + \lambda_2 \leq 2 \\ & 3\lambda_1 + \lambda_2 \leq 3 \\ & \lambda_1, \lambda_2 \geq 0 \end{cases}$$

【例 5.3】 设种某一作物, 整个生产过程中至少需要氮肥 32kg, 磷肥 24kg, 钾肥 42kg, 已知含有氮磷钾的四种化肥的单价和含量如下表。问应选用这四种化肥各多少 kg, 才能既满足作物的需要, 又使施肥的总成本最低? 试建立数学模型。

含 量 (kg) 成 分 \ 化 肥	甲	乙	丙	丁
氮	0.03	0.3	0	0.15
磷	0.05	0	0.2	0.10
钾	0.14	0	0	0.07
单价(元 /kg)	0.4	1.5	1.0	1.3

我们来写出这个问题的原始问题和对偶问题,并说明对偶问题的实际背景。

用 x_1, x_2, x_3, x_4 分别表示甲、乙、丙、丁四种化肥的用量, z 表示施肥的总成本,则原始问题可写成

$$\left\{ \begin{array}{l} \min \quad z = 0.4x_1 + 1.5x_2 + x_3 + 1.3x_4 \\ \text{s. t.} \quad 0.03x_1 + 0.3x_2 + 0.15x_4 \geq 32 \\ \quad \quad 0.05x_1 + 0.2x_3 + 0.1x_4 \geq 24 \\ \quad \quad 0.14x_1 + 0.07x_4 \geq 42 \\ \quad \quad x_i \geq 0, i = 1, 2, 3, 4 \end{array} \right.$$

而对偶问题则为

$$\left\{ \begin{array}{l} \max \quad y = 32\lambda_1 + 24\lambda_2 + 42\lambda_3 \\ \text{s. t.} \quad 0.03\lambda_1 + 0.05\lambda_2 + 0.14\lambda_3 \leq 0.4 \\ \quad \quad 0.3\lambda_1 \leq 1.5 \\ \quad \quad \quad \quad 0.2\lambda_2 \leq 1 \\ \quad \quad 0.15\lambda_1 + 0.1\lambda_2 + 0.07\lambda_3 \leq 1.3 \\ \quad \quad \lambda_i \geq 0, i = 1, 2, 3 \end{array} \right.$$

上面的对偶问题可以解释为某肥料公司计划生产氮磷钾三种单成分的化肥。公司要为这三种化肥确定单价(设为 $\lambda_1, \lambda_2, \lambda_3$),要使其利润最高,同时又要与生产甲、乙、丙、丁四种化肥的工厂竞争,因此要求用单成分化肥合成的甲肥、乙肥、丙肥、丁肥的单价分别不超过 0.4 元、1.5 元、1 元和 1.3 元,这样就得到了上面的对偶问题。

2.5.2 对偶定理

上面定义的对偶问题,从表面上看,似乎是人们的一种随意的规定,其实不然。上面的例子已经指出了对偶问题是有其实际意义的。下面我们再从理论上来说明原始问题与它的对偶问题之间深刻的内在联系。

引理 1 若 x 和 w 分别是(5-3), (5-4') 的可行解, 则 $c^T x \geq w^T b$ 。

证明 因为

$$x \geq 0, w^T A \leq c^T \quad (5-5)$$

故 $w^T b = w^T (Ax) = (w^T A)x \leq c^T x$ ■

推论 设 x^*, w^* 分别为(5-3), (5-4') 的可行解, 且 $c^T x^* = w^{*T} b$, 则 x^*, w^* 分别是(5-3), (5-4') 的最优解。

证明 用反证法证明, 假设 x^* 不是(5-3) 的最优解, 则有可行解 \tilde{x} , 使 $c^T \tilde{x} < c^T x^*$ 。

由于 $c^T x^* = w^{*T} b$, 所以 $c^T \tilde{x} < w^{*T} b$, 这与引理 1 矛盾, 因此 x^* 是(5-3) 的最优解。类似地可以证明 w^* 是(5-4') 的最优解。 ■

线性规划的对偶定理 若(5-3) 或(5-4') 二者有一个有有限的最优解, 则另一个也有有限的最优解, 而且二者的目标函数值相同。若任一问题的目标函数值无界, 则另一个问题没有可行解。

证明 设 x^* 是(5-3) 的最优基本可行解, B 为对应的最优基。则检验矢量 r 应满足(参见 2.4 节(4-11))

$$r^T = c^T - \pi^T A \geq 0 \quad (5-6)$$

其中 $\pi^T = c_B^T B^{-1}$ 。因此, $\pi^T A \leq c^T$, 即 π 是对偶问题(5-4') 的可行解, 又因为

$$\pi^T b = (c_B^T B^{-1})b = c_B^T (B^{-1}b) = c_B^T x_B^* = c^T x^*$$

由推论知 $w^* = \pi$ 是(5-4') 的一个有限的最优解。

另一种情形的证明从略。

若问题(5-3)的目标函数值无界,即 $\min(c^T x) = -\infty$ 。下面用反证法证明(5-4')没有可行解。

假设(5-4')有可行解 w ,则由引理1知:对问题(5-3)的任何可行解 x ,有 $c^T x \geq w^T b$,这与 $c^T x$ 无下界相矛盾,所以(5-4')没有可行解。 ■

2.5.3 对偶单纯形法

上面讲的单纯形法是从标准形式的线性规划问题(5-3)(即原始问题)的一个基本可行解出发,逐次进行迭代,使目标函数值逐次减小,直到获得最优的基本可行解为止。下面对单纯形法给出另一种解释。

定理 5.1 设 x 为(5-3)的任一基本解,对应的基为 B ,令 $w^T = c_B^T B^{-1}$ 。若 x, w 分别是(5-3)和(5-4')的可行解,则 x, w 也分别是(5-3)和(5-4')的最优解。

证明 因为 $w = (c_B^T B^{-1})^T$ 是(5-4')的可行解,所以

$$w^T A = (c_B^T B^{-1}) A \leq c^T$$

因此 $c^T - c_B^T B^{-1} A \geq 0$,即 $r \geq 0$,这说明 x 是(5-3)的最优(基本可行)解。又因

$$w^T b = c_B^T (B^{-1} b) = c_B^T x_B = c^T x$$

由上一段中引理1的推论知: w 也是(5-4')的最优解。 ■

由上述可见:与基本可行解 x 相应的所有检验数 $r_j \geq 0$ ($j = 1, 2, \dots, n$) 和 $w = (c_B^T B^{-1})^T$ 是对偶问题(5-4')的可行解等价。据此,我们可对单纯形法给出如下的解释:从(5-3)的一个基本可行解 x 出发迭代到另一个基本可行解,同时使它对应的对偶规划的解 $w = (c_B^T B^{-1})^T$ 的不可行性逐步消失(即:使检验数逐步变为非负),直到 w 是(5-4')的可行解为止,这时 x 就是(5-3)的最优解。

定义 5.1 若 x 是(5-3)的一个基本解(设对应的基为 B),且

它对应的检验矢量 $r \geq 0$ (即 $w = (c_B^T B^{-1})^T$ 为 (5-4') 的可行解), 则称 x 为 (5-3) 的对偶可行解或正则解。

对偶单纯形法的基本思想是: 从原始问题 (5-3) 的一个对偶可行的基本解开始, 逐次进行迭代, 在保持对偶可行性的条件下, 逐步使原始问题 (5-3) 的基本解 x 的不可行性消失 (即使 $x \geq 0$), 直到获得 (5-3) 的一个基本可行解为止, 而它即为原始问题的最优解。

对偶单纯形法的计算步骤如下:

(1) 把一般的线性规划问题化为 (5-3) 的形式, 这时不要求 $b \geq 0$ 。

(2) 列出初始单纯形表, 求出 (5-3) 的一个对偶可行的基本解

$$x = \begin{pmatrix} x_B \\ 0 \end{pmatrix}$$

(3) 若 $x_B \geq 0$, 则现行解 x 为最优解, 计算结束; 否则 $x_B \not\geq 0$, 令 $x_{B_i} = \min \{x_{B_j} \mid j = 1, 2, \dots, m\}$, 由此确定 a_{B_i} 为离基矢量。

(4) 用 y_{ij} 表示单纯形表中第 B_i 行的各个元素, 若所有的 $y_{ij} \geq 0$ ($j = 1, 2, \dots, n$), 则对偶问题 (5-4') 的目标函数无上界, 原始问题 (5-3) 没有可行解; 否则必有某个 j , 使 $y_{ij} < 0$, 令

$$\varepsilon = \min_{1 \leq j \leq n} \{(z_j - c_j)/y_{ij} \mid y_{ij} < 0\} = (z_k - c_k)/y_{ik} \quad (5-7)$$

其中

$$z_j = w^T a_j, \quad j = 1, 2, \dots, n \quad (5-8)$$

$$w^T = c_B^T B^{-1} \quad (5-9)$$

由 (5-7) 决定 a_k 为进基矢量, y_{ik} 为主元素。

(5) 用 a_k 代替 a_{B_i} 形成一个新基, 再用这个新基来确定相应的对偶可行的基本解, 即以 y_{ik} 为主元素, 进行一次旋转变换, 然后返回第 3 步。

下面举例说明。

【例 5.4】 用对偶单纯形法求解问题

$$\begin{cases} \min & z = 3x_1 + 4x_2 + 5x_3 \\ \text{s. t.} & x_1 + 2x_2 + 3x_3 \geq 5 \\ & 2x_1 + 2x_2 + x_3 \geq 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases} \quad (5-10)$$

解 (1) 引入剩余变量, 将(5-10) 写成为

$$\begin{cases} \min & z = 3x_1 + 4x_2 + 5x_3 \\ \text{s. t.} & -x_1 - 2x_2 - 3x_3 + x_4 = -5 \\ & -2x_1 - 2x_2 - x_3 + x_5 = -6 \\ & x_i \geq 0, i = 1, 2, \dots, 5 \end{cases} \quad (5-11)$$

(2) 令 $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, 则基本变量为 x_4, x_5 , 非基本变量为 x_1, x_2, x_3 , 初始单纯形表为

基 矢 量	$a_1 \downarrow$	a_2	a_3	a_4	$a_5 \uparrow$	b
a_4	-1	-2	-3	1	0	-5
$\leftarrow a_5$	-2	-2	-1	0	1	-6
检 验 数	3	4	5	0	0	0

其中 $r_j = c_j - z_j$, $z_j = \mathbf{w}^T \mathbf{a}_j$, $j = 1, 2, \dots, 5$ 。

因初始的对偶可行解为 $\mathbf{x} = (0, 0, 0, -5, -6)^T$, 故

$$\mathbf{x}_B = \begin{pmatrix} -5 \\ -6 \end{pmatrix}, \mathbf{w}^T = \mathbf{c}_B^T \mathbf{B}^{-1} = (0, 0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (0, 0)$$

(3) 由于 $\mathbf{x}_B \not\geq \mathbf{0}$, 选取 $x_{B_2} = x_5 = -6 < 0$, 所以 a_5 为离基向量。

$$\begin{aligned} (4) \quad \text{因} \quad \varepsilon &= \min_{1 \leq j \leq 5} \{ (z_j - c_j) / y_{2j} \mid y_{2j} < 0 \} \\ &= 3/2 = (z_1 - c_1) / y_{21} \end{aligned}$$

所以 a_1 为进基矢量, 主元素为 $y_{21} = -2$, 在表中用黑体字标记。

(5) 以 $y_{21} = -2$ 为主元, 进行一次旋转变换, 就可得到用 a_1

代替 a_5 形成一个新基 B 。

基 矢 量	a_1	$a_2 \downarrow$	a_3	$a_4 \uparrow$	a_5	b
$\leftarrow a_4$	0	-1	-5/2	1	-1/2	-2
a_1	1	1	1/2	0	-1/2	3
检 验 数	0	1	7/2	0	3/2	-9

再返回第 3 步,重复上述计算,可得离基矢量为 a_4 ,进基矢量为 a_2 ,主元素为 $y_{12} = -1$ 。新的单纯形表为

基 矢 量	a_1	a_2	a_3	a_4	a_5	b
a_2	0	1	5/2	-1	1/2	2
a_1	1	0	-2	1	-1	1
检 验 数	0	0	1	1	1	-11

因为这时 $x_B = (1, 2)^T \geq 0$, 所以 $x^* = (1, 2, 0, 0, 0)^T$ 是可行的,它即为原始问题(5-10)的最优解,而最优目标函数值为 $z^* = 11$ 。

从上述算法和例 5.4 可见:对偶单纯形法的执行必须从一个对偶可行解开始,这可通过人工约束法来实现,参见文献 20。

* 2.6 线性规划的多项式算法

1979 年,前苏联数学家哈奇扬(Л. Г. Хачиян)发表了《线性规划的多项式算法》一文,文中给出了一种解线性规划的方法——椭球算法^[59],其计算复杂性为 $O(n^6 L^2)$ 。1984 年,在美国贝尔实验室工作的印度数学家卡玛卡尔(N. Karmarkar)给出了线性规划的一个新的多项式算法,其计算复杂性为 $O(n^{3.5} L^2)$,大大改进了哈奇扬的结果,引起了人们的极大兴趣。本节对 LP 的多项式算法作一简介。

2.6.1 多项式算法的概念

考虑解某种问题(例如线性规划问题)的一种算法,这种问题通常包含许多具体问题(例如每一个具体的 LP 问题),称每一个具体问题为一个实例。一种算法当然可以求解这种问题中的每一个实例。衡量一种算法好坏的一条重要标准是该算法的计算次数或计算时间。显然,解不同的实例,所需的计算次数或计算时间是不同的。因为计算次数与实例的规模 n (例如 LP 问题中变量的个数 n ,约束条件的数目 m)和输入数据的位数 L (或称为输入数据长度)有关。如果存在 n 和 L 的一个多项式 $P(n, L)$,使得该问题的任何实例都可以在计算时间(或次数) $T = O(P(n, L))$ 之内解出,则称该问题存在多项式时间算法,简称多项式算法,而 $O(P(n, L))$ 称为算法的计算复杂性。一个问题仅当存在多项式算法时,它才可以有效地用计算机来求解,而这种算法就认为是好的算法。对规模为 n ,输入数据长度为 L 的问题,通常把计算时间 T 破坏多项式界的算法,称为非多项式算法。例如,一个算法的计算时间 $T = O(2^n)$,它就是一个指数型算法。而哈奇扬算法和卡玛卡尔算法都是多项式算法,一般认为:指数型算法不是好算法。

在文献中还会遇到:

(1) P 类问题。已找到多项式算法的所有问题称为 P 类问题,简称 P 问题。

(2) NP 类问题,也叫不确定性问题。如果有一个用于求解此问题的算法,对于它可以找到一个多项式算法来验证该算法所得的结果是否为此问题的解,则称此问题是属于 NP 类问题,简称 NP 问题。

如所熟知:求解 LP 问题的单纯形法简单、实用,是大家乐于采用的方法。但是,1972 年, V. Klee 和 G. Minty 构造了一个反例,它含有 n 个变量, $m = 2n$ 个不等式约束,若用单纯形法来求解,选

择适当的初始点,必须检验约束条件中不等式组所确定的凸多面体的所有极点,才能获得最优解,其计算复杂性为 $O(2^n)$ 。因此,从计算复杂性的标准来看,单纯形法不是好算法。于是就产生了一个问题:能否找到一种解 LP 问题的多项式算法?而哈奇扬的文章恰好解决了这个问题,他找到了一种解 LP 问题的多项式算法——椭圆算法。哈奇扬工作的重要性在于:他第一个证明了 LP 问题存在多项式算法。

2.6.2 线性规划与线性不等式组的关系

哈奇扬提出的是一种求解严格整系数线性不等式组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n < b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n < b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n < b_m \end{cases} \quad (6-1)$$

的算法。可以证明,求 LP 问题的最优解可以转化为解线性不等式组的问题。

设要求解的 LP 问题是:

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{cases} \quad (6-2)$$

它的对偶问题是

$$\begin{cases} \max & \mathbf{w}^T \mathbf{b} \\ \text{s. t.} & \mathbf{A}^T \mathbf{w} \leq \mathbf{c}^T \\ & \mathbf{w} \geq \mathbf{0} \end{cases} \quad (6-3)$$

在 2.5 中讲过:若 \mathbf{x}, \mathbf{w} 分别是 (6-2) 和 (6-3) 的可行解,则必有 $\mathbf{c}^T \mathbf{x} \geq \mathbf{w}^T \mathbf{b}$ 。若 $\mathbf{x}^*, \mathbf{w}^*$ 分别是 (6-2) 和 (6-3) 的可行解,且 $\mathbf{c}^T \mathbf{x}^* = \mathbf{w}^{*T} \mathbf{b}$,则 $\mathbf{x}^*, \mathbf{w}^*$ 分别是 (6-2) 和 (6-3) 的最优解。因此,要求解 LP 问题 (6-2),可以先解如下的线性不等式组:

$$\begin{cases} Ax \geq b \\ x \geq 0 \\ A^T w \leq c^T \\ w \geq 0 \\ c^T x \leq w^T b \end{cases} \quad (6-4)$$

若已求得(6-4)的一组解(x^* , w^*), 则 x^* 就是 LP 问题(6-2)的一个最优解。若(6-4)没有解, 则问题(6-2)没有最优解。所以求 LP 问题的最优解可以转化为求解线性不等式组(6-4), 而不等式组(6-4)总可以改写为如下的形式:

$$\begin{cases} -Ax \leq -b \\ -x \leq 0, \\ A^T w \leq c^T \\ -w \leq 0 \\ c^T x - w^T b \leq 0 \end{cases} \quad (6-4')$$

为了书写简便, 不妨将(6-4')改写为

$$a_i^T x \leq b_i, \quad i = 1, 2, \dots, m \quad (6-5)$$

其中 a_i 为 A 的 i 个列向量。应当指出, 这里的 A , x 和 b 与(6-4')中的 A , x 和 b 是不同的。

哈奇扬证明: 如果整系数不等式组

$$a_i^T x < b_i + 2 \cdot 2^{-L}, \quad i = 1, 2, \dots, m \quad (6-6)$$

有解, 则整系数不等式组(6-5)也有解。其中

$$L = \left[\sum_{i=1}^m \sum_{j=1}^n \log_2(|a_{ij}| + 1) + \sum_{i=1}^m \log_2(|b_i| + 1) + \log_2 nm \right] + 1 \quad (6-7)$$

$[a]$ 表示不超过 a 的整数部分。 L 大致等于把不等式组(6-5)的所有系数都化为二进制数时的位数, 称 L 为问题(6-5)的输入长度, 它大体上说明了问题(6-5)的规模大小。

综上所述可见,求解 LP 问题(6-2) 可以转化为求解不等式组(6-1)。

2.6.3 哈奇扬算法

哈奇扬的工作主要就是给出了一种求解严格整系数线性不等式组(6-1) 的多项式算法,这种算法也是一种迭代法。在迭代过程中的每一步,都要产生一个点 $x^{(k)}$ 和以 $x^{(k)}$ 为中心的椭球 $E_k = E(x^{(k)}, B_k)$,其中 B_k 是与第 k 个椭球方程

$$(x - x^{(k)})^T B_k (x - x^{(k)}) \leq 1$$

对应的 n 阶方阵,而 $x^{(1)} = 0$, $B_1 = I_n$, E_1 则是一个以 $x^{(1)}$ 为球心的 n 维单位超球。

哈奇扬算法的计算步骤如下:

(1) 按(6-7) 式算出不等式组(6-1) 对应的输入长度 L ,令 $x^{(1)} = 0$, $B_1 = 2^{2L} I_n$ (其中 I_n 为 n 阶单位阵),得到初始点 $x^{(1)}$ 和初始椭球(实际上是一个超球) $E_1 = E(x^{(1)}, B_1)$ 后,令 $k = 1$, 转第 2 步。

(2) 如果 $k > 6n^2 L$, 计算结束,(6-1) 没有解;否则转第 3 步。

(3) 如果 $x^{(k)}$ 是(6-1) 的解,计算结束;否则至少有一个 i ,使 $a_i^T x^{(k)} \geq b_i$, 令

$$a_r^T x^{(k)} - b_r = \max_{1 \leq i \leq m} \{a_i^T x^{(k)} - b_i \mid a_i^T x^{(k)} - b_i \geq 0\} \quad (6-8)$$

转第 4 步。

(4) 按下面的公式,计算下一个迭代点 $x^{(k+1)}$ 和以它为中心的椭球 E_{k+1} 。

$$x^{(k+1)} = x^{(k)} - \frac{1}{n+1} \frac{B_k a_r}{\sqrt{a_r^T B_k a_r}} \quad (6-9)$$

$$B_{k+1} = \frac{n^2}{n^2 - 1} \left(B_k - \frac{2}{n+1} \frac{(B_k a_r)(B_k a_r)^T}{a_r^T B_k a_r} \right) \quad (6-10)$$

其中 a_r 由(6-8) 式确定。

令 $k = k + 1$, 返回第 2 步。

由上述可见: 哈奇扬算法并不复杂, 但由于 L 往往很大, 而且它没有充分利用具体 LP 问题的特性, 所以迭代次数比用单纯形法要多得多。哈奇扬算法的实用价值虽然不大, 但在理论上的意义是很大的, 它解决了 LP 理论上的一个重要问题, 即 LP 问题是否存在多项式算法。关于哈奇扬算法的详细介绍和严格证明可参看文献 18 和文献 59。

2.6.4 卡玛卡尔(Karmarkar) 算法

首先考虑特殊形式的 LP 问题:

$$\left\{ \begin{array}{ll} \min & z = \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{0} \\ & \sum_{i=1}^n x_i = 1 \\ & \mathbf{x} \geqslant \mathbf{0} \end{array} \right. \quad (6-11)$$

令 $\Omega = \{\mathbf{x} | \mathbf{Ax} = \mathbf{0}\}$, $S = \{\mathbf{x} | \sum_{i=1}^n x_i = 1, \mathbf{x} \geqslant \mathbf{0}\}$ 。其中 A 为 $m \times n$ 矩阵, $\mathbf{x}, \mathbf{c} \in \mathbf{R}^n$ 。并假设:

(1) 目标函数的最小值 $z^* = 0$ 。

(2) $R = \Omega \cap S \neq \emptyset$, 单纯形 S 的中心 $\mathbf{a}_0 = (1, 1, \dots, 1)^T/n \in R$ 。

为以后叙述方便, 我们称上述问题为 Karmarkar 标准型问题。下面给出求解标准型问题的计算步骤。

算法一 (主算法)

给定 $\alpha \in (0, \beta)$, $\mathbf{a}_0 = (1, 1, \dots, 1)^T/n \in R$, 其中 $\beta = \sqrt{(n-1)/n}$, 计算精度 $\epsilon > 0$, 选取 $q > 0$, 使 $2^{-q} \leqslant \epsilon$ 。

(1) 令 $\mathbf{x}^{(0)} = \mathbf{a}_0$, $k = 0$ 。

(2) 若 $\mathbf{c}^T \mathbf{x}^{(k)} / (\mathbf{c}^T \mathbf{a}_0) \leqslant 2^{-q}$, 则 $\mathbf{x}^{(k)}$ 为最优解, 计算结束; 否则

按某种规则修改 α , 使修改后的 $\alpha \in (0, \beta)$, 然后转第 3 步。

(3) 令 $D = \text{diag}[x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]$, 这里 $\text{diag}[\]$ 为对角矩阵。

(4) 令 $B = \begin{pmatrix} AD \\ e^T \end{pmatrix}$, 这里 $e^T = (1, 1, \dots, 1) \in \mathbf{R}^n$ 。

(5) 求 $c_p = [I_n - B^T(BB^T)^{-1}B]Dc$, 其中 I_n 为 n 阶单位矩阵。

(6) 求 $\bar{C} = c_p / \|c_p\|$ (即将矢量 c_p 单位化)。

(7) 求 $b' = a_0 - \alpha \bar{C} / \sqrt{n(n-1)}$ 。

(8) 令 $x^{(k+1)} = D \cdot b' / (e^T \cdot D \cdot b')$, $k = k + 1$, 返回第 2 步。

几点说明:

(1) α 可取为常数, 例如取 $\alpha = 1/4$; 也可采用如下规则: 取 $\alpha = 1/8$, 然后令 $\alpha = (\alpha + \beta)/2$, 逐次进行修改。

(2) 由上述计算步骤可见: 每次迭代, 计算量都集中在第 5 步上, 因为求矩阵 (BB^T) 的逆占去了主要的计算量。用 Gauss 消去法求一个矩阵的逆, 需进行 $O(n^3)$ 次算术运算。因此, 如何减少求 $(BB^T)^{-1}$ 的计算量成为快速实现卡玛卡尔算法的一个关键。卡玛卡尔给出一种技巧, 可以使求 $(BB^T)^{-1}$ 的算术运算次数减少 $n^{0.5}$ 。

上面的算法一是针对标准型问题给出的。下面来讨论将一般的 LP 问题

$$\begin{cases} \min & z = c^T x \\ \text{s. t.} & Ax = b \\ & x \geq 0 \end{cases} \quad (6-12)$$

化为标准型问题的方法。

设 (6-12) 的可行集 R 有界, 那么可以选取充分大的正整数 M , 使得对任意 $x \in R$, 有

$$x_1 + x_2 + \dots + x_n \leq M$$

引入松弛变量 $x_{n+1} \geq 0$, 可使上式成为

$$x_1 + x_2 + \dots + x_n + x_{n+1} = M \quad (6-13)$$

作线性变换

$$y_i = x_i/M, \quad i = 1, 2, \dots, n, n+1$$

则(6-13)变为

$$y_1 + y_2 + \dots + y_{n+1} = 1$$

而问题(6-12)变为

$$\begin{cases} \min & \bar{z} = \bar{c}^T y \\ \text{s. t.} & A_1 y = b_1 \\ & e^T y = 1 \\ & y \geq 0 \end{cases} \quad (6-14)$$

其中 $y \in \mathbf{R}^{n+1}$ 。

因为 $A_1 y = b_1$ 可改写为 $A_1 y - b_1 = A_1 y - b_1 e^T y = 0$, 即 $(A_1 - b_1 e^T) y = 0$, 所以问题(6-14)又可改写为

$$\begin{cases} \min & \tilde{z} = \tilde{c}^T y \\ \text{s. t.} & A_2 y = 0 \\ & e^T y = 1 \\ & y \geq 0 \end{cases} \quad (6-15)$$

其中 $A_2 = A_1 - b e^T$, 这样就把问题(6-12)化成了标准型问题所要求的形式。在上述方法中, 由于 M 很大, 在计算机上实现比较困难。在实际计算中, 若在第 i 个方程中, 所有的系数 $a_{ij} > 0$ ($j = 1, 2, \dots, n$), 且 $b_i > 0$, 则可作线性变换:

$$y_j = a_{ij} x_j / b_j, \quad j = 1, 2, \dots, n$$

将第 i 个方程 $\sum_{j=1}^n a_{ij} x_j = b_j$ 化为 $\sum_{j=1}^n y_j = 1$ 。

在标准型问题中, 还假设 a_0 是问题(6-11)的可行解, 且 $z^* = 0$, 在卡玛卡尔的文章中也介绍了处理的办法, 有兴趣的读者可参看文献 19 和 60。

习 题 二

1. 将下面的线性规划问题化为标准形式。

$$(1) \begin{cases} \min & f(x) = 10x_1 + 4x_2 \\ \text{s. t.} & 3x_1 + 4x_2 \leq 10 \\ & 5x_1 + 2x_2 \leq 8 \\ & x_1 - 2x_2 \leq 3 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(2) \begin{cases} \min & z = 3x_1 - 4x_2 + 2x_3 - x_4 \\ \text{s. t.} & 3x_1 + x_2 + x_3 \leq 7 \\ & 4x_1 + x_2 + 6x_3 - x_4 \geq 6 \\ & -x_1 - x_2 + x_3 + x_4 = -4 \\ & x_1 \geq 1, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

2. 作出习题 1 (1) 的可行集 R 的图形, 并作出目标函数 $f(x)$ 分别等于 $-10, -5, 4, 10, 20$ 时的等值线。

3. 证明 LP 问题

$$\begin{cases} \min & w = x_1 - 2x_2 + 2x_3 + x_4 \\ \text{s. t.} & x_1 + x_2 + x_3 + x_4 = 4 \\ & x_1 + x_2 - x_3 - x_4 = 6 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

的可行集 R 是一个空集。

4. 设线性规划问题为

$$\begin{cases} \min & z = 3x_1 - 4x_2 - x_3 \\ \text{s. t.} & 3x_1 + 4x_2 + x_3 + x_4 = 9 \\ & 5x_1 + 2x_2 + x_5 = 8 \\ & x_1 - 2x_2 - x_4 + x_6 = -1 \\ & x_i \geq 0, i = 1, 2, \dots, 6 \end{cases}$$

(1) 求它的一个基本解, 可行解, 基本可行解 (要求它们不相同);

(2) 试求基本变量为 x_1, x_2, x_3 的一个基本可行解。

5. 用图解法求如下优化问题的最优解。

$$\begin{cases} \max & w = 3x_1 - x_2 \\ \text{s. t.} & 2x_1 + x_2 \geq 2 \\ & x_1 + 3x_2 \leq 6 \\ & x_1 \geq 0, 4 \geq x_2 \geq 0 \end{cases}$$

6. 试构造一个线性规划问题,使它有无限多个最优解。

* 7. 证明定理 2.2 的(2)。

8. 设 $a_i \in \mathbf{R}^m (i = 1, 2, \dots, n)$, a_1, a_2, \dots, a_m 为 \mathbf{R}^m 的一组基,则

$$a_j = \sum_{i=1}^m y_{ij} a_i, \quad j = 1, 2, \dots, n$$

证明: 当且仅当 $y_{pq} \neq 0$ 时, $a_1, \dots, a_{p-1}, a_q, a_{p+1}, \dots, a_m$ 也是 \mathbf{R}^m 的一组基。其中 $1 \leq p < q \leq n$ 。

9. 把如下的线性规划问题化为标准形式,并求解

$$\begin{cases} \min & w = x_1 + 4x_2 + x_3 \\ \text{s. t.} & 2x_1 - 2x_2 + x_3 = 4 \\ & x_1 - x_3 = 1 \\ & x_2 \geq 0, \quad x_3 \geq 0 \end{cases}$$

10. 用单纯形法求解如下的 LP 问题。

$$(1) \begin{cases} \min & w = x_1 - 3x_2 \\ \text{s. t.} & -x_1 + 2x_2 \leq 6 \\ & x_1 + x_2 \leq 5 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(2) \begin{cases} \min & z = -(10x_1 + 11x_2) \\ \text{s. t.} & 3x_1 + 4x_2 \leq 9 \\ & 5x_1 + 2x_2 \leq 8 \\ & x_1 - 2x_2 \leq 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(3) \begin{cases} \min & w = x_1 - 2x_2 + 3x_3 \\ \text{s. t.} & -2x_1 + x_2 + 3x_3 = 2 \\ & 2x_1 + 3x_2 + 4x_3 = 10 \\ & x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$(4) \begin{cases} \max & w = 3x_1 - x_2 \\ \text{s. t.} & 2x_1 + x_2 \geq 2 \\ & x_1 + 3x_2 \leq 3 \\ & x_2 \leq 4 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(5) \begin{cases} \max & z = 10x_1 + 7x_2 + 4x_3 + 3x_4 + x_5 \\ \text{s. t.} & 2x_1 + 6x_2 + x_3 \leq 7 \\ & 2x_1 + 3x_2 + 4x_3 + x_4 + x_5 \leq 8 \\ & x_1 + 2x_2 + 3x_4 + x_5 \leq 5 \\ & x_i \geq 0, i = 1, 2, \dots, 5 \end{cases}$$

11. 用单纯形法求解如下的 LP 问题:

$$(1) \begin{cases} \min & w = 4x_1 + x_2 + x_3 \\ \text{s. t.} & 2x_1 + x_2 + 2x_3 = 4 \\ & 3x_1 + 3x_2 + x_3 = 3 \\ & x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$(2) \begin{cases} \min & w = 2x_1 + x_2 - x_3 - x_4 \\ \text{s. t.} & x_1 - x_2 + 2x_3 - x_4 = 2 \\ & 2x_1 + x_2 - 3x_3 + x_4 = 6 \\ & x_1 + x_2 + x_3 + x_4 = 7 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

12. 用修正单纯形法求解如下的 LP 问题:

$$(1) \begin{cases} \min & z = -x_1 - 2x_2 - 3x_3 + x_4 \\ \text{s. t.} & x_1 + 2x_2 + x_3 + x_4 = 10 \\ & x_1 + 2x_2 + 3x_3 + x_5 = 15 \\ & 2x_1 + x_2 + 5x_3 + x_6 = 20 \\ & x_i \geq 0, i = 1, 2, 3, 4, 5, 6 \end{cases}$$

$$(2) \begin{cases} \min & z = -2x_1 - x_2 - 4x_3 - 5x_4 \\ \text{s. t.} & x_1 + 2x_2 + 2x_3 + x_4 \leq 2 \\ & 2x_1 - x_2 + x_3 - x_4 \geq 1 \\ & -x_1 + x_2 + 2x_3 - x_4 \geq 1 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$(3) \begin{cases} \min & z = 4x_1 + 12x_2 + 2x_3 + 5x_5 + 10x_6 \\ \text{s. t.} & 2x_1 + x_2 + x_3 \geq 30 \\ & x_2 + 3x_4 + 2x_5 + x_6 \geq 60 \\ & 2x_3 + x_5 + 2x_6 + 4x_7 \geq 48 \\ & x_i \geq 0, i = 1, 2, \dots, 7 \end{cases}$$

* 13. 试举出一个在用单纯形法进行迭代求解过程中,会出现退化的基本可行解的线性规划问题。

14. 设某工厂生产 A_1, A_2, A_3, A_4 四种产品,都要用车床及钻床,已知它们可供利用的时间,生产一件产品所需使用设备的时间和产值如下表所示,问 A_1, A_2, A_3, A_4 各生产多少,才能使总产值最高?

设 备	生产一件产品所需的时间(h)				可供利用的时间(h)
	A_1	A_2	A_3	A_4	
车 床	1	2	2	0.5	100
钻 床	3	1	3	2	100
产 值 (元)	4	3	7	2	

15. 种植 A 种作物,根据当地的气候、土壤等情况,全部生产过程中至少需氮 32kg,磷以 24kg 为宜,钾不得超过 42 kg。现有四种肥料,其单价和所含氮磷钾的数量如下表所示。问应施用这四种肥料各多少,才能既满足作物对氮磷钾的需要,又使施肥的总成本最低?

所 含 成 分 数 量 成 分	肥 料			
	甲	乙	丙	丁
氮	0.03	0.3	0	0.15
磷	0.05	0	0.2	0.1
钾	0.14	0	0	0.07
单价(元 /kg)	0.4	1.5	1.0	1.3

16. 某工厂用甲、乙两台机床,加工 A,B,C 三种零件,已知在一生产周期内甲只能工作 80 机时,乙只能工作 100 机时。一生产周期内要加工 A,B,C 的件数分别为 70,50,20。两台机床加工每个零件的时间和成本如下表,问应如何安排两台机床生产一周期的加工任务,才能使成本最低?

时 间 和 成 本 机 床	零 件 A		B		C	
	时间(机时 / 件)	成本(元 / 件)	时间(机时 / 件)	成本(元 / 件)	时间(机时 / 件)	成本(元 / 件)
甲	1	2	1	3	1	5
乙	1	3	2	4	3	6

17. 画出用修正单纯形法求解线性规划问题的程序框图,编写计算程

序,并求解习题 11。

18. 用 Bland 法求解 Beale 问题(即 2.4 中的例 4)。

* 19. 假定一个 LP 问题有有限的最小值 z^* , 若用单纯形法求它的最优解, 设在进行到第 k 步时出现退化的基本可行解, 但这时并未达到最优解(即目标函数 $z_k > z^*$), 并且只有一个基变量为零($x_i = 0$)。求证: 这个基本可行解在以后各步中不会重新出现。

20. 写出下列问题的对偶问题:

$$(1) \begin{cases} \min & z = 5x_1 - 3x_2 \\ \text{s. t.} & x_1 + x_2 - 2x_3 \geq 1 \\ & 3x_1 - x_2 + x_3 \geq 3 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

$$(2) \begin{cases} \min & (2x_1 - x_2) \\ \text{s. t.} & 2x_1 - x_2 - x_3 \geq 3 \\ & x_1 - x_2 + x_4 \geq 2 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases}$$

$$(3) \begin{cases} \min & (2x_1 + x_2 + 4x_3) \\ \text{s. t.} & x_1 + x_2 + 2x_3 = 3 \\ & 2x_1 + x_2 + 3x_3 = 5 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

$$(4) \begin{cases} \min & (3x_1 - 2x_2 + 4x_4) \\ \text{s. t.} & 2x_1 - x_2 + 4x_3 + x_4 \leq 5 \\ & x_1 + x_2 + 2x_3 \leq 4 \\ & 2x_1 - x_2 + x_3 \geq 1 \\ & x_i \geq 0, i = 1, 2, 3, 4. \end{cases}$$

21. 求线性规划问题

$$\begin{cases} \min & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{a} \end{cases}$$

的对偶问题, 其中 $\mathbf{a} \geq \mathbf{0}$ 。

* 22. 试构造一个原始问题的例子, 使它本身及其对偶问题均没有可行

解。

23. 用对偶单纯形法求解如下的线性规划问题:

$$(1) \begin{cases} \min & z = x_1 + 2x_2 \\ \text{s. t.} & x_1 + 2x_2 \geq 4 \\ & x_1 \leq 5 \\ & 3x_1 + x_2 \geq 6 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(2) \begin{cases} \min & z = 2x_1 + x_2 \\ \text{s. t.} & 3x_1 + x_2 \geq 3 \\ & 4x_1 + 3x_2 \geq 6 \\ & x_1 + 2x_2 \geq 2 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

24. 考虑 Karmarkar 标准形式的 LP 问题:

$$\begin{cases} \min & z = \mathbf{c}^T \mathbf{x} \\ \text{s. t.} & \mathbf{Ax} = \mathbf{0} \\ & \mathbf{e}^T \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0} \end{cases} \quad (1)$$

其中 $A = (a_{ij})_{m \times n}$, $\mathbf{e}^T = (1, 1, \dots, 1) \in \mathbf{R}^n$, $\text{rank}(A) = m < n$ 。

令 $\Omega = \{x \in \mathbf{R}^n | \mathbf{Ax} = \mathbf{0}\}$, $S = \{x \in \mathbf{R}^n | \mathbf{e}^T x = 1, x \geq 0\}$ 。

设 \bar{x} 是 (1) 的内点解, 令 $D = \text{diag}(\bar{x})$, $B = \begin{pmatrix} AD \\ \mathbf{e}^T \end{pmatrix}$ 。

证明矩阵 $P = I_n - B^T(B B^T)^{-1}B$ 具有如下性质:

$$(1) P = P^T, \quad P P = P; \quad (2) P \geq 0$$

25. 用 Karmarkar 算法求解如下 LP 问题:

$$\begin{cases} \min & z = -x_1 - 2x_2 + 4x_5 \\ \text{s. t.} & x_2 - x_3 = 0 \\ & 2x_1 - 2x_2 + 4x_3 - 4x_5 = 0 \\ & x_1 + 2x_2 + x_4 - 4x_5 = 0 \\ & x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ & x_i \geq 0, i = 1, 2, \dots, 5 \end{cases}$$

第 3 章 无约束优化方法

本章讨论无约束非线性规划问题的解法,包括一维搜索、基本下降算法、共轭梯度法、变尺度法、直接搜索法等。这些方法在以后将经常用到,也是学习约束优化方法的基础。

3.1 引言 下降递推算法

第一章讲过,无约束非线性规划问题的提法是:求目标函数 $f(x)$, $x \in \mathbf{R}^n$ 的极小(或极大)值。而求 $f(x)$ 的极大值的问题可以化为求 $-f(x)$ 的极小值问题。第一章我们还指出:若 $f(x) \in C^1$, 则 x^* 为 $f(x)$ 的一个极小点的必要条件是

$$\nabla f(x^*) = \mathbf{0}$$

或

$$\frac{\partial f(x^*)}{\partial x_i} = 0, i = 1, 2, \dots, n \quad (1-1)$$

这是 n 个未知量、 n 个方程的方程组,并且一般是非线性的。只有在比较特殊的情况下,方程组(1-1)是可以求出准确解的。在一般情况下,都不能用解析方法求得准确解,只能用数值方法逐步求其近似解,这就是本章所要介绍的求解无约束最优化问题的各种迭代方法,简称为无约束优化方法。

迭代法的基本思想是:首先给出 $f(x)$ 的极小点 x^* 的一个初始估计 $x^{(0)}$ (称为初始点),然后,计算一系列的点 $x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots$, 希望点列 $\{x^{(k)}\}$ 的极限就是 $f(x)$ 的一个极小点 x^* 。怎样

产生这个点列呢?也就是说在有了点 $x^{(k)}$ 之后,如何求得点 $x^{(k+1)}$ 呢?我们这样来考虑,因为 $x^{(k+1)} - x^{(k)}$ 是一个向量,而一个向量是由它的方向和长度确定的,即

$$x^{(k+1)} - x^{(k)} = \lambda_k s^{(k)}$$

或

$$x^{(k+1)} = x^{(k)} + \lambda_k s^{(k)} \quad (1-2)$$

其中 $s^{(k)}$ 是一个向量, λ_k 是一个正实数(称为步长)。当 λ_k 与 $s^{(k)}$ 确定以后,由 $x^{(k)}$ 就可以惟一的确定 $x^{(k+1)}$,这样就可以确定逼近极小点的一个序列 $\{x^{(k)}\}$,通常称它为一个极小化序列。从而也就确定了一个算法,而各种不同算法的差别就在于选取方向 $s^{(k)}$ 和步长 λ_k 的方法不同,特别是选取 $s^{(k)}$ 的方法不同。那么,应该按照什么原则来选取 $s^{(k)}$ 和 λ_k 呢?

(1) 要求极小化序列对应的函数值是逐次减少的,至少是不增的,即

$$f(x^{(0)}) \geq f(x^{(1)}) \geq \dots \geq f(x^{(k)}) \geq \dots$$

具有这种性质的算法,称为下降递推算法或下降算法,以后我们讨论的算法,一般都是下降算法。

(2) 要求极小化序列具有这样的性质,或者序列 $\{x^{(k)}\}$ 中的某一点 $x^{(N)}$ 本身是 $f(x)$ 的极小点,或者 $\{x^{(k)}\}$ 有一个极限 x^* ,它是函数 $f(x)$ 的一个极小点。具有上述性质的算法称为收敛的。这个要求是很基本的,因为如果极小化序列不收敛于极小点,那么,这个序列就没有什么用处了。但是这个要求却并不容易做到。

因此,当提出一种算法时,往往要对其收敛性进行研究,但这个工作是困难的。事实上有许多方法在经过长时间的实际应用之后,其收敛性才得到证明;有的算法,尽管其收敛性尚未得到证明,但在某些实际问题的应用中显示出是很有效的,因而人们仍在不断地使用它。另一方面,任何一种算法,也只能对满足一定条件的目标函数来说是收敛的,但是,当应用于不满足这些条件的函数

时,有时也能收到很好的效果。由于我们的主要目的是向大家介绍一些解决实际问题的最优化方法,所以对算法的收敛性一般不进行讨论。此外,当目标函数有多个极小值时,我们求得的往往是局部极小值,这时可以改变初始点重新计算。如果经过几次计算,求得的是同一个极小点,我们就认为它是全局极小点。

一般把最优化算法的迭代过程分为如下四步:

(1) 选择初始点 $x^{(0)}$;

(2) 如果 $x^{(k)}$ 已求得,且 $x^{(k)}$ 不是极小点,设法选取一个方向 $s^{(k)}$,使目标函数 $f(x)$ 沿 $s^{(k)}$ 是下降的,至少是不增的, $s^{(k)}$ 称为搜索方向;

(3) 在方向 $s^{(k)}$ 确定以后,在射线 $x^{(k)} + \lambda s^{(k)}$ ($\lambda \geq 0$) 上选取适当的步长 λ_k ,使 $f(x^{(k)} + \lambda_k s^{(k)}) \leq f(x^{(k)})$,如此确定出下一个点 $x^{(k+1)} = x^{(k)} + \lambda_k s^{(k)}$ (在多数算法中, λ_k 的选取是使 $f(x)$ 的值下降最多,即沿射线 $x^{(k)} + \lambda s^{(k)}$ 求 $f(x)$ 的极小值,这是关于单变量 λ 的函数求极小的问题)。这种确定步长 λ_k 的方法称为一维搜索或线搜索。

(4) 检验所得的新点 $x^{(k+1)}$ 是否为极小点,或满足精度要求的近似极小点,检验的方法因算法而不同,例如,有些算法看梯度 $\nabla f(x)$ 的模是否小于某个预先指定的小正数 ϵ ,即若 $\|\nabla f(x^{(k+1)})\| < \epsilon$,就认为 $x^{(k+1)}$ 是 $f(x)$ 的近似极小点,迭代结束,否则继续进行迭代。

给定一个迭代算法,我们不仅要求它是收敛的,而且希望由这个算法产生的点列 $\{x^{(k)}\}$,以较快的速度收敛于最优解 x^* 。收敛的快慢通常用收敛的阶来度量。

定义 1.1 设 x^* 为问题 $\min f(x)$, $x \in \mathbf{R}^n$ 的最优解,由算法 A 产生的序列 $\{x^{(k)}\}$ 收敛于 x^* ,即

$$\lim_{k \rightarrow +\infty} x^{(k)} = x^* \quad (1-3)$$

若存在一个与 k 无关的常数 $\beta \in (0, 1)$, 某个正整数 k_0 , 使当 $k > k_0$

时,

$$\|x^{(k+1)} - x^*\| \leq \beta \|x^{(k)} - x^*\| \quad (1-4)$$

成立,则称序列 $\{x^{(k)}\}$ 为线性收敛,有时也说算法 A 是线性收敛的。

不失一般性,可以假定(1-4)式对于 $k \geq 0$ 都成立,于是有

$$\begin{aligned} \|x^{(k)} - x^*\| &\leq \beta \|x^{(k-1)} - x^*\| \leq \beta^2 \|x^{(k-2)} - x^*\| \\ &\leq \dots \leq \beta^k \|x^{(0)} - x^*\| \end{aligned}$$

上式说明:点 $x^{(k)}$ 到点 x^* 的距离,当 $k \rightarrow +\infty$ 时,大致以公比为 β 的等比序列(又称几何序列)减小,因此线性收敛速度相当于相应的等比序列的收敛速度。

定义 1.2 设由算法 A 产生的序列 $\{x^{(k)}\}$ 收敛于最优解 x^* ,如果存在一个与 k 无关的常数 $\beta > 0$ 和 $\alpha > 1$,某个正整数 k_0 ,使当 $k > k_0$ 时,

$$\|x^{(k+1)} - x^*\| \leq \beta \|x^{(k)} - x^*\|^\alpha \quad (1-5)$$

成立,则称序列 $\{x^{(k)}\}$ 的收敛阶为 α ,或称序列 $\{x^{(k)}\}$ 是 α 阶收敛的,有时也说算法 A 是 α 阶收敛的。

当 $\alpha = 2$ 时,称序列 $\{x^{(k)}\}$ 是二阶收敛的;

当 $1 < \alpha < 2$ 时,称序列 $\{x^{(k)}\}$ 是超线性收敛的。

一般来说,线性收敛是比较慢的,超线性收敛就比较快了,而二阶收敛就更快了。如果一个算法具有超线性收敛速度,就认为它是一个比较好的算法(从计算速度这个角度看)。

设由算法 A 产生的序列 $\{x^{(k)}\}$ 收敛于最优解 x^* ,那么在计算机上迭代到哪一点,我们才能肯定它就是所求的近似最优解呢?显然,当 $\|x^{(k)} - x^*\| < \epsilon$ 时(其中 $\epsilon > 0$ 是预先给定的小正数), $x^{(k)}$ 就是所求的近似最优解。但 x^* 是未知的,因此 $\|x^{(k)} - x^*\|$ 无法计算。不过,当 $\|x^{(k)} - x^*\|$ 很小时, $\|x^{(k+1)} - x^{(k)}\|$ 也必然很小,所以,可以用

$$\|x^{(k+1)} - x^{(k)}\| < \epsilon \quad (1-6)$$

作为计算结束的检验条件, ε 是根据计算要求给定的一个小正数, 以后我们称为计算精度或终止限。但是只用 (1-6) 来作为结束准则是不够的, 因为 $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, 并不能保证 $\|x^{(k)} - x^*\| < \varepsilon$, 所以还需加上一个条件:

$$|f(x^{(k+1)}) - f(x^{(k)})| < \varepsilon \quad (1-7)$$

3.2 一 维 搜 索

从上一节的讲述中我们看到: 在大多数无约束最优化算法中, 为了确定极小化点列 $\{x^{(k)}\}$, 要沿逐次确定的搜索方向 $s^{(k)}$, 在射线 $x^{(k)} + \lambda s^{(k)}$ 上, 确定适当的步长 λ_k , 使 $f(x^{(k+1)}) \leq f(x^{(k)})$, 这里 $x^{(k+1)} = x^{(k)} + \lambda_k s^{(k)}$ 。在不少算法中, 步长 λ_k 是由求 $\varphi(\lambda) = f(x^{(k)} + \lambda s^{(k)})$ 的极小值确定的, 即 $f(x^{(k)} + \lambda_k s^{(k)}) = \min_{\lambda \geq 0} f(x^{(k)} + \lambda s^{(k)})$, 这种确定步长 λ_k 的方法称为精确一维搜索或简称一维搜索。本节介绍几种常用的精确一维搜索方法, 最后对不精确的一维搜索作一简介。

设要求一元函数 $\varphi(x)$ 的极小点问题为

$$\min \varphi(x) \quad (2-1)$$

若 $\varphi(x) \in C^1$, 则在极小点 x^* 处, 应有

$$\varphi'(x^*) = 0 \quad (2-2)$$

精确一维搜索方法, 可以分为两类:

(1) 不用导数的方法, 如 0.618 法、分数法、“成功 — 失败”法等。

(2) 使用导数的方法, 如 Newton 法、抛物线法、三次插值法等。

下面介绍的几种方法, 有的是直接求解问题 (2-1), 有的是通过求解问题 (2-2) 来求解问题 (2-1)。

3.2.1 牛顿(Newton)法

牛顿法的基本思想是：用 $\varphi(x)$ 在已知点 x_0 处的二阶 Taylor 展开式来近似代替 $\varphi(x)$ ，即取 $\varphi(x) \approx g(x)$ ，其中 $g(x) = \varphi(x_0) + \varphi'(x_0)(x - x_0) + (1/2)\varphi''(x_0)(x - x_0)^2$ ，用 $g(x)$ 的极小点 \hat{x} 作为 $\varphi(x)$ 的近似极小点，参看图 3-1。

由于 $g'(x) = \varphi'(x_0) + \varphi''(x_0)(x - x_0)$ ，令 $g'(x) = 0$ ，得

$$x_1 = x_0 - \frac{\varphi'(x_0)}{\varphi''(x_0)} \quad (2-3)$$

类似的，若已知 x_k 点，则有

$$x_{k+1} = x_k - \frac{\varphi'(x_k)}{\varphi''(x_k)} \quad k = 0, 1, 2, \dots \quad (2-3')$$

按公式 (2-3') 进行迭代计算，可求得一个序列 $\{x_k\}$ ，这种求一元函数极小值的一维搜索方法称为牛顿法。当 $|\varphi'(x_k)| < \varepsilon$ 时（其中 $\varepsilon > 0$ 为计算精度），则迭代结束， x_k 为 $\varphi(x)$ 的近似极小点，即 $x^* \approx x_k$ 。

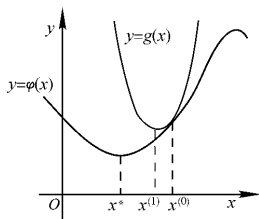


图 3-1

牛顿法的优点是收敛速度快，可以证明，它至少是二阶收敛的^[16]。缺点是：需计算二阶导数；要求初始点

x_0 选得好，即要求 $x_0 \in N(x^*, \delta)$ ， $\delta > 0$ ，否则可能不收敛。

【例 2.1】 求 $f(x) = x^4 - 4x^3 - 6x^2 - 16x + 4$ 的极小值。

$$\begin{aligned} \text{解} \quad f'(x) &= 4x^3 - 12x^2 - 12x - 16 \\ &= 4(x - 4)(x^2 + x + 1) \end{aligned}$$

令 $f'(x) = 0$ ，得 $x_1 = 4, x_2 = \frac{-1 \pm \sqrt{3}i}{2}, i = \sqrt{-1}$ 。

因此，在实数域内，只有 $x_1 = 4$ 使 $f'(x) = 0$ 。

因为 $f''(x) = 12x^2 - 24x - 12 = 12(x^2 - 2x - 1)$

$$f''(4) = 84 > 0$$

所以 $f(x)$ 在 $x_1 = 4$ 处取得极小值 $f(4) = -156$ 。

下面再用牛顿法来求 $f(x)$ 的极小点。

取 $x_0 = 6$, $\varepsilon = 10^{-3}$, 则

$$x_1 = 6 - \frac{f'(6)}{f''(6)} = 6 - \frac{86}{69} = 6 - 1.25 = 4.75$$

$$\begin{aligned} x_2 &= 4.75 - \frac{f'(4.75)}{f''(4.75)} = 4.75 - \frac{84.94}{144.75} = 4.75 - 0.587 \\ &= 4.163 \end{aligned}$$

$$\begin{aligned} x_3 &= 4.163 - \frac{f'(4.163)}{f''(4.163)} = 4.163 - \frac{14.666}{96.055} \\ &= 4.163 - 0.153 = 4.010 \end{aligned}$$

$$x_4 = 4.010 - \frac{f'(4.010)}{f''(4.010)} = 4.010 - \frac{0.8436}{84.7212} \approx 4.00004$$

$x_4 = 4.00004$ 已经很接近 $f(x)$ 的极小点 $x^* = 4$ 。可见 Newton 法收敛是比较快的。

应当指出: Newton 法产生的序列即使收敛,其极限也不一定是 $f(x)$ 的极小点,而只能保证它是 $f(x)$ 的驻点,驻点可能是极小点,也可能是极大点,也可能既不是极小点,又不是极大点。例如,对函数 $f(x) = 3x^4 - 16x^3 + 30x^2 - 24x + 8$,不难求出它有两个驻点 $x = 1$, $x = 2$ 。而 $x = 2$ 是 $f(x)$ 的极小点, $x = 1$ 则不是 $f(x)$ 的极值点。若取 $x_0 = 1.5$, 则不难验证,当 $n \rightarrow \infty$ 时 $x_n \rightarrow 1$ 。因此,为了保证 Newton 法收敛到极小点,应要求 $f''(x_n) > 0$, 至少对足够大的 n 如此。

3.2.2 抛物线法

抛物线法的基本思想也是用二次函数 $g(x)$ 来近似地代替原来的函数 $f(x)$, 并以它的极小点作为函数 $f(x)$ 的近似极小点。它

与 Newton 法不同的地方在于: Newton 法是用 $f(x)$ 在 x_0 点的二阶 Taylor 展开式作为 $g(x)$ 来逼近 $f(x)$, 即利用 x_0 点的函数值 $f(x_0)$ 及其一、二阶导数值 $f'(x_0), f''(x_0)$ 来构造二次函数 $g(x)$, 而抛物线法则是利用 $f(x)$ 在三个点 x_0, x_1, x_2 处的函数值来构造一个二次函数 $y = g(x) = a_0 + a_1x + a_2x^2$, 使它满足

$$\begin{cases} y_1 = g(x_0) = f(x_0) \\ y_2 = g(x_1) = f(x_1) \\ y_3 = g(x_2) = f(x_2) \end{cases} \quad (2-4)$$

则 $g'(x) = a_1 + 2a_2x$ 。令 $g'(x) = 0$, 得

$$\bar{x} = -a_1/2a_2 \quad (2-5)$$

(2-5) 就是计算 $f(x)$ 的近似极小点的公式, 为了求出近似极小点 \bar{x} , 只需算出 a_1, a_2 即可。

令 $f(x_0) = f_0, f(x_1) = f_1, f(x_2) = f_2$, 由 (2-4) 得

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 = f_0 \\ a_0 + a_1x_1 + a_2x_1^2 = f_1 \\ a_0 + a_1x_2 + a_2x_2^2 = f_2 \end{cases}$$

消去 a_0 得

$$\begin{cases} a_1(x_1 - x_0) + a_2(x_1^2 - x_0^2) = f_1 - f_0 \\ a_1(x_2 - x_1) + a_2(x_2^2 - x_1^2) = f_2 - f_1 \end{cases} \quad (2-6)$$

或

$$\begin{cases} a_1 + a_2(x_1 + x_0) = (f_1 - f_0)/(x_1 - x_0) \\ a_1 + a_2(x_2 + x_1) = (f_2 - f_1)/(x_2 - x_1) \end{cases}$$

消去 a_1 得

$$a_2 = [(f_2 - f_1)/(x_2 - x_1) - (f_1 - f_0)/(x_1 - x_0)]/(x_2 - x_0) \quad (2-7)$$

令

$$\begin{cases} (f_1 - f_0)/(x_1 - x_0) = b_1 \\ (f_2 - f_1)/(x_2 - x_1) = b_2 \end{cases} \quad (2-8)$$

则得

$$a_2 = (b_2 - b_1)/(x_2 - x_0) \quad (2-7')$$

故

$$a_1 = b_1 - a_2(x_1 + x_0) \quad (2-9)$$

由(2-8)(2-7')(2-9)式即可求得 a_2 和 a_1 , 代入(2-5)式或利用

$$\bar{x} = \frac{1}{2} \left[x_1 + x_0 - \frac{b_1}{a_2} \right] \quad (2-10)$$

即可求得 $f(x)$ 的近似极小点 \bar{x} 。

注意:

(1) 通常假设 $f(x) \in C^1$, $x_1 < x_0 < x_2$, $f(x_0) < f(x_1)$, $f(x_0) < f(x_2)$, 这时可以证明 $a_2 > 0$, 因而求得的 \bar{x} 确是 $g(x)$ 的极小点, 且 $\bar{x} \in [x_1, x_2]$ 。

(2) 迭代的结束准则可以采用以下几种:

(I) 若 $|f(\bar{x}) - f(x_0)| < \varepsilon$ (其中 $\varepsilon > 0$ 为已给的计算精度), 则结束, 取 $x^* \approx \bar{x}$, 否则, 在点 \bar{x}, x_0, x_1, x_2 中, 选取使 f 值最小的点为新的 x_0 , 并使新的 x_1, x_2 各是新的 x_0 近旁的左右两点, 继续进行迭代, 直到获得近似最优解为止。

(II) 若 $|x_0 - \bar{x}| < \varepsilon$, 则迭代结束, 取 $x^* \approx \bar{x}$; 否则继续进行迭代。

(III) 若 $|f(\bar{x}) - g(\bar{x})| < \varepsilon$, 则迭代结束, 取 $x^* \approx \bar{x}$, 否则继续进行迭代。

还可以采用其他的结束准则。

(3) 应当指出: 上面介绍的抛物线法并不能保证算法一定收敛, 在迭代过程中可能出现上一次迭代点 x_k 与下一次迭代点 x_{k+1} 充分接近, 而 x_{k+1} 并不是 $f(x)$ 的近似极小点的退化情况^[35]。但是若已知由抛物线法产生的点列 $\{x_k\}$ 收敛于 $f(x)$ 的极小点 x^* , 则可以证明: 在一定的条件下, 抛物线法是超线性收敛的, 其收敛阶

约为 1.3, 证明见文献 16, 32, 35。

3.2.3 三次插值法

三次插值法是用 a, b 两点处的函数值 $f(a), f(b)$ 和导数值 $f'(a), f'(b)$ 来构造三次插值多项式 $g(x)$, 并以 $g(x)$ 的极小点作为 $f(x)$ 的近似极小点。一般来说, 三次插值法比抛物线法的收敛速度要快些。

为了保证 $f(x)$ 的极小点 $x^* \in [a, b]$, 假定 $f(x) \in C^1$ 。

$$a < b, f'(a) < 0, f'(b) > 0 \quad (2-11)$$

设插值多项式为

$$g(x) = \alpha(x-a)^3 + \beta(x-a)^2 + \gamma(x-a) + \delta \quad (2-12)$$

由插值条件知 $\alpha, \beta, \gamma, \delta$ 应满足

$$\begin{cases} \delta = f(a) \\ \gamma = f'(a) \\ \alpha(b-a)^3 + \beta(b-a)^2 + \gamma(b-a) + \delta = f(b) \\ 3\alpha(b-a)^2 + 2\beta(b-a) + \gamma = f'(b) \end{cases} \quad (2-13)$$

下面来求 $f(x)$ 的近似极小点。

$$g'(x) = 3\alpha(x-a)^2 + 2\beta(x-a) + \gamma$$

令 $g'(x) = 0$, 分两种情况进行讨论。

(1) 当 $\alpha = 0$ 时, 插值多项式 $g(x)$ 为二次多项式,

$$\hat{x} = a - \frac{\gamma}{2\beta} \quad (2-14)$$

(2) 当 $\alpha \neq 0$ 时,

$$\hat{x} = a + \frac{-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha} \quad (2-15)$$

为了保证所求的根为 $g(x)$ 的极小点, 在该点还需要满足

$$g''(x) = 6\alpha(x-a) + 2\beta > 0 \quad (2-16)$$

将(2-16)代入上式得

$g''(\hat{x}) = 2(-\beta \pm \sqrt{\beta^2 - 3\alpha\gamma}) + 2\beta = \pm 2\sqrt{\beta^2 - 3\alpha\gamma} > 0$
故(2-15)式中的根号前应取正号。

注意到：当 $\alpha \neq 0$ 时，

$$\hat{x} - a = \frac{-\beta + \sqrt{\beta^2 - 3\alpha\gamma}}{3\alpha} = \frac{-\gamma}{\beta + \sqrt{\beta^2 - 3\alpha\gamma}}$$

当 $\alpha = 0$ 时，上式可化为 $\hat{x} - a = -\gamma/2\beta$ ，所以(2-14)与(2-15)可合写为

$$\hat{x} = a - \frac{\gamma}{\beta + \sqrt{\beta^2 - 3\alpha\gamma}} \quad (2-17)$$

这就是计算 $f(x)$ 的近似极小点的公式。为了求出 \hat{x} ，我们来推导计算 α, β, γ 的公式。

由(2-13)的第三式得

$$\alpha(b-a)^2 + \beta(b-a) = \frac{f(b) - \delta}{b-a} - \gamma$$

令

$$u \triangleq \frac{f(b) - \delta}{b-a} - \gamma \quad (2-18)$$

得

$$\alpha(b-a)^2 + \beta(b-a) = u \quad (2-19)$$

从(2-13)的第四式得

$$3\alpha(b-a)^2 + 2\beta(b-a) = f'(b) - \gamma$$

令

$$v = f'(b) - \gamma \quad (2-20)$$

得

$$3\alpha(b-a)^2 + 2\beta(b-a) = v \quad (2-21)$$

从(2-19)和(2-21)中消去 α ，得

$$\beta = \frac{3u - v}{b-a} \quad (2-22)$$

将(2-22)代入(2-19)，得

$$\alpha = \frac{v - 2u}{(b-a)^2} \quad (2-23)$$

将求得的 α, β, γ 代入(2-17)式，即可求得 $f(x)$ 的近似极小点 \hat{x} 。

若 $|f'(\hat{x})| < \varepsilon$ (其中 $\varepsilon > 0$ 为要求的精度), 则取 $x^* \approx \hat{x}$; 否则继续进行迭代, 当 $f'(\hat{x}) > 0$ 时, 用 \hat{x} 代替 b , 否则用 \hat{x} 代替 a , 直到求得所要的近似极小点为止。

3.2.4 平分法(对分法)

设 $f(x) \in C^1$, x^* 为 $f(x)$ 的极小点。

当 $x < x^*$ 时, 若 $f'(x) < 0$, 则 $f(x)$ 单调减少; 当 $x > x^*$ 时, 若 $f'(x) > 0$, 则 $f(x)$ 单调增加。如果我们找到了一个区间 $[a, b]$, 且有 $f'(a) < 0$ 和 $f'(b) > 0$, 则在 a, b 之间 ($a < b$), 必有 $f(x)$ 的一个极小点 x^* , 使 $f'(x^*) = 0$ 。我们可以这样来求 x^* :

取 $x_0 = (a + b)/2$, 若 $f'(x_0) > 0$, 则 $f(x)$ 在 $[a, x_0]$ 中有极小点, 用 $[a, x_0]$ 代替 $[a, b]$, 再取 $x_1 = (a + x_0)/2$; 若 $f'(x_0) < 0$, 则 $f(x)$ 在 $[x_0, b]$ 中有极小点, 用 $[x_0, b]$ 代替 $[a, b]$, 再取 $x_1 = (x_0 + b)/2$; ...; 经过 k 次迭代, 设所得的小区间为 $[a_k, b_k]$ 。

若 $|b_k - a_k| < \delta$, 或 $|f'(x_k)| < \varepsilon$ 时, 取 $x^* \approx x_k = (a_k + b_k)/2$, 迭代结束; 否则继续进行迭代。 ■

【例 2.2】 设 $f(x) = 3x^4 - 16x^3 + 30x^2 - 24x + 8$, 求 $f(x)$ 的极小点。

$$\begin{aligned}\text{解} \quad f'(x) &= 12x^3 - 48x^2 + 60x - 24 \\ &= 12(x^3 - 4x^2 + 5x - 2)\end{aligned}$$

因为 $f'(0) = -24$, $f'(3) = 48$, 所以在 $[0, 3]$ 内有 $f(x)$ 的极小点 x^* , 取 $a_0 = 0$, $b_0 = 3$, 于是 $x_0 = 3/2$, 又因为 $f'(3/2) = -3/2$, 取 $[3/2, 3]$ 代替 $[0, 3]$, 即令 $a_1 = 3/2$, $b_1 = 3$, 所以

$$x_1 = (3/2 + 3)/2 = 9/4, \quad f'(x_1) = f'(9/4) = 4.68$$

取 $a_2 = 3/2$, $b_2 = 9/4$, 所以

$$x_2 = (3/2 + 9/4)/2 = 15/8 = 1.875$$

$$f'(x_2) = f'(1.875) = -1.0284$$

取 $a_3 = 15/8$, $b_3 = 9/4$, 则

$$x_3 = 2.0625$$

$$f'(x_3) = f'(2.0625) = 0.8472$$

取 $a_4 = 15/8$, $b_4 = 2.0625$, 则

$$x_4 = 1.96875$$

$$f'(x_4) = f'(1.96875) = -0.35196$$

取 $a_5 = 1.96875$, $b_5 = 2.0625$, 则

$$x_5 = 2.015625, \dots \rightarrow x^* = 2$$

这时 $|b_5 - a_5| = 0.09375$, 所以可取 $x^* \approx 2.016$ 。

平分法的优点是：① 每一步的计算量小，程序简单；② 总能收敛于一个局部极小点 x^* 。

缺点是收敛速度很慢。

3.2.5 “成功 - 失败”法

设要求解的问题为 $\min_{a \leq x \leq b} F(x)$, 令

$$f(x) = \begin{cases} F(x), & x \in [a, b] \\ +\infty, & x \notin [a, b] \end{cases}$$

容易看出

$$\min_{a \leq x \leq b} F(x) = \min_{x \in \mathbf{R}} f(x), \quad \text{而 } \mathbf{R} = (-\infty, +\infty)$$

因此, 不失一般性, 我们总可以研究 $f(x)$ 在 $(-\infty, +\infty)$ 上的无条件极值问题, 即

$$\min_{x \in \mathbf{R}} f(x)$$

以下方法叫做“成功 - 失败”法：

(1) 取定初始点 $x_0 \in \mathbf{R}$, 搜索步长 $h > 0$ 及精度 $\varepsilon > 0$;

(2) 计算 $x_1 = x_0 + h$, $f(x_1)$;

(3) 若 $f(x_1) < f(x_0)$, 称为搜索成功, 下一次搜索就大步前进, 用 x_1 代替 x_0 , $2h$ 代替 h , 继续进行搜索。

若 $f(x_1) \geq f(x_0)$, 称为搜索失败, 下一次搜索就小步后退。首先看是否有 $|h| < \varepsilon$? 若 $|h| < \varepsilon$, 则取 $x^* \approx x_0$, 计算结束; 否则, 用 $-h/4$ 代替 h , 返回第 2 步, 继续进行搜索。

计算框图见图 3-2。

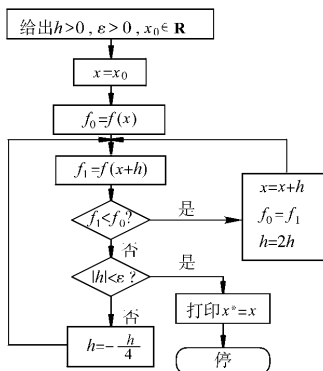


图 3-2

注意初始搜索步长 $h > 0$ 不能选得太小。

3.2.6 0.618 法

考察一维搜索问题：

$$\begin{cases} \min & \varphi(\lambda) \\ \text{s. t.} & a_1 \leq \lambda \leq b_1 \end{cases} \quad (2-24)$$

其中 $\varphi(\lambda)$ 为凸函数(实际上当 $\varphi(\lambda)$ 为拟凸函数时, 下述结论仍然可用, 参看文献 [11])。

0.618 法也叫黄金分割法, 是不用导数的一维搜索方法中比较常用的一种方法, 它所需要进行的试验或函数计算次数略多于分数法^[11]。

设 $\varepsilon > 0$ 为允许的最后的搜索区间长度, 令 $\alpha = 0.618$, 则 0.618 法的计算步骤如下:

1° 计算

$$\lambda_1 = a_1 + (1 - \alpha)(b_1 - a_1), \varphi(\lambda_1)$$

$$\mu_1 = a_1 + \alpha(b_1 - a_1), \quad \varphi(\mu_1)$$

令 $k = 1$ 。

2° 若 $b_k - a_k < \varepsilon$, 则计算结束, 最优解 $\lambda^* \in [a_k, b_k]$, 可取 $\lambda^* \approx (1/2)(b_k + a_k)$; 否则, 若 $\varphi(\lambda_k) > \varphi(\mu_k)$, 则转 3°; 若 $\varphi(\lambda_k) \leq \varphi(\mu_k)$, 则转 4°。

3° 令 $a_{k+1} = \lambda_k, b_{k+1} = b_k$, 再令 $\lambda_{k+1} = \mu_k, \mu_{k+1} = a_{k+1} + \alpha(b_{k+1} - a_{k+1})$, 计算 $\varphi(\mu_{k+1})$, 转 5°。

4° 令 $a_{k+1} = a_k, b_{k+1} = \mu_k$, 再令 $\mu_{k+1} = \lambda_k, \lambda_{k+1} = a_{k+1} + (1 - \alpha)(b_{k+1} - a_{k+1})$, 计算 $\varphi(\lambda_{k+1})$, 转 5°。

5° 令 $k = k + 1$, 返回 2°。 ■

【例 2.3】 求解 $\min \varphi(\lambda) = \lambda^2 + 2\lambda, \text{ s. t. } -3 \leq \lambda \leq 5$, 显然 $\varphi(\lambda)$ 在 $[-3, 5]$ 上是凸函数。取 $\varepsilon = 0.2$ 。

$$1^\circ \quad \lambda_1 = -3 + 0.382 \times 8 = 0.056$$

$$\mu_1 = -3 + 0.618 \times 8 = 1.944$$

$$\varphi(\lambda_1) = \varphi(0.056) = 0.115$$

$$\varphi(\mu_1) = \varphi(1.944) = 7.667$$

2° 因为 $\varphi(\lambda_1) < \varphi(\mu_1)$, 所以

$$a_2 = a_1 = -3, b_2 = \mu_1 = 1.944$$

新的搜索区间为 $[-3, 1.944]$ 。

$$\lambda_2 = a_2 + 0.382 \times (1.944 + 3) = -1.112$$

$$\varphi(\lambda_2) = -0.987, \mu_2 = \lambda_1 = 0.056$$

重复上述计算到 $k = 9$, 得到

$$a_9 = -1.112, b_9 = -0.936$$

因为 $|b_k - a_k| = 0.176 < \varepsilon = 0.2$, 所以计算结束, 取

$$x^* \approx (a_9 - b_9)/2 = (1/2)(-1.112 - 0.936) = -1.024$$

而函数 $\varphi(\lambda) = \lambda^2 + 2\lambda$ 的准确极小点为 $\lambda^* = -1$ 。

由上例可见, 0.618 法收敛速度是比较慢的, 它的优点是不要
求 $\varphi(\lambda)$ 可微, 且每次迭代, 只需计算一个函数值, 因此, 计算量小,
程序简单。

3.2.7 确定初始搜索区间和初始点的方法

前面讲的一维搜索方法, 有的需要从一个初始的搜索区间出
发, 逐次进行迭代计算, 大多数算法, 都要从一个初始点出发, 逐次
进行迭代。下面给出一种进退算法, 它可以同时确定初始的搜索区
间和初始点。设要求解的问题为 (2.1)。

(1) 选定初始点的一个估计值 t_0 , 初始步长 $h > 0$, 计算 $\varphi(t_0)$ 。

(2) 令 $t_2 = t_0 + h$, 计算 $\varphi(t_2)$ 。

(3) 若 $\varphi(t_2) \leq \varphi(t_0)$, 转第 4 步; 否则, 若 $\varphi(t_2) > \varphi(t_0)$, 令
 $h = -h$, 转第 4 步。

(4) 令 $t_1 = t_0 + h$, 计算 $\varphi(t_1)$ 。

(5) 若 $\varphi(t_1) \leq \varphi(t_0)$, 则令 $h = 2h$, $t_2 = t_0$, $t_0 = t_1$, 转第 4 步;
否则, 若 $\varphi(t_1) > \varphi(t_0)$, 转第 6 步。

(6) 令 $a = \min(t_1, t_2)$, $b = \max(t_1, t_2)$, 则 $[a, b]$ 即为所求
的初始搜索区间, 而 $t_0 = (b + a)/2$ 可作为所求的初始点, 计算结
束。

3.2.8 不精确一维搜索

在实际计算中, 一般做不到精确的一维搜索, 实际上, 也没有
必要做到这一点。因为精确的一维搜索需要付出较高的代价, 而对
加速收敛作用不大。所以不精确一维搜索方法受到了广泛的重视
和欢迎。这里对实用的不精确一维搜索方法作简要的介绍。

设要求 $\min \varphi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$, 在不精确一维搜索中, 通
常要求 $f(\mathbf{x}^{(k+1)})$ 比 $f(\mathbf{x}^{(k)})$ 下降一定的数量, 而且在新点 $\mathbf{x}^{(k+1)} =$
 $\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$ 处沿方向 $\mathbf{s}^{(k)}$ 的方向导数值比在 $\mathbf{x}^{(k)}$ 点沿 $\mathbf{s}^{(k)}$ 的方向导

数值大一定的数量,这里“一定的数量”是用 $-\nabla f(\mathbf{x}^{(k)})^T \mathbf{s}^{(k)} > 0$ 来度量的。通常采用的是 Wolfe-Powell 不精确一维搜索准则,简称 Wolfe-Powell 准则。即对给定的常数 $c_1, c_2, 0 < c_1 < c_2 < 1$,要求 λ_k 满足如下条件:

$$1^\circ \quad f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) \geq -c_1 \lambda_k \nabla f(\mathbf{x}^{(k)})^T \mathbf{s}^{(k)} \quad (2-25)$$

$$2^\circ \quad \nabla f(\mathbf{x}^{(k+1)})^T \mathbf{s}^{(k)} \geq c_2 \nabla f(\mathbf{x}^{(k)})^T \mathbf{s}^{(k)} \quad (2-26)$$

根据计算经验,常取 $c_1 = 0.1, c_2 = 0.5$ 。

不精确一维搜索算法一(直接法)的计算步骤如下:

设点 $\mathbf{x}^{(k)}$ 、搜索方向 $\mathbf{s}^{(k)}$ 已求得,为书写简便起见,令

$$f_k \triangleq f(\mathbf{x}^{(k)}), f_{k+1} \triangleq f(\mathbf{x}^{(k+1)})$$

$$g_k \triangleq \nabla f(\mathbf{x}^{(k)}), g_{k+1} \triangleq \nabla f(\mathbf{x}^{(k+1)})$$

求出 f_k, g_k 。

(1) 给定 $c_1 \in (0, 1), c_2 \in (c_1, 1)$, 令 $a = 0, b = -\infty, \lambda = 1, j = 0$ 。

(2) 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}$, 计算 f_{k+1}, g_{k+1} , 若 λ 满足条件 1° , 2° , 则令 $\lambda_k = \lambda$, 计算结束; 否则令 $j = j + 1$, 若 λ 不满足条件 1° , 则转第 3 步; 若 λ 满足条件 1° , 不满足条件 2° , 则转第 4 步。

(3) 令 $b = \lambda, \lambda = (\lambda + a)/2$, 返回第 2 步。

(4) 令 $a = \lambda, \lambda = \min\{2\lambda, (\lambda + b)/2\}$, 返回第 2 步。

在上述算法中,第 3、4 步的放大与缩小的系数 2 与 $1/2$ 也可改取为 $1/\beta, \beta (0 < \beta < 1)$ 或 $\beta_2 > 1, 0 < \beta_1 < 1$ 。 ■

【例 2.4】 设 $f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, 已求得 $\mathbf{x}^{(k)} = (0, 0)^T, \mathbf{s}^{(k)} = (1, 0)^T$, 试确定在 $\mathbf{x}^{(k)}$ 点, 沿方向 $\mathbf{s}^{(k)}$ 的步长 λ_k , 使条件 $1^\circ, 2^\circ$ 成立。

$$\text{解} \quad \nabla f(\mathbf{x}) = \begin{bmatrix} -400(x_2 - x_1^2)x_1 - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix}$$

$$f_k = f(0, 0) = 1, \quad g_k = (-2, 0)^T, \quad g_k^T \mathbf{s}^{(k)} = -2$$

(1) 给定 $c_1 = 0.1$, $c_2 = 0.5$, 令 $a = 0$, $b = +\infty$, $\lambda = 1$, $j = 0$;

(2) 因为 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)} = (1, 0)^T$, 所以 $f_{k+1} = f(1, 0) = 100$, 从而

$$f_k - f_{k+1} = 1 - 100 = -99 < -c_1 \lambda \mathbf{g}_k^T \mathbf{s}^{(k)} = 0.2$$

即条件 1° 不成立, 转第 3 步;

(3) 令 $b = 1$, $\lambda = (1 + 0)/2 = 0.5$, 返回第 2 步重算。直到 $j = 3$, $\lambda = 0.125$, 这时, $\mathbf{x}^{(k+1)} = (0.125, 0)^T$, $f_{k+1} = 0.790$, $\mathbf{g}_{k+1} = (-0.969, -3.125)^T$,

$$\begin{aligned} f_k - f_{k+1} &= 1 - 0.79 = 0.21 > -c_1 \lambda \mathbf{s}_k^T \mathbf{s}^{(k)} \\ &= -0.1 \times 0.125 \times (-2) = 0.025 \end{aligned}$$

$$\mathbf{g}_{k+1}^T \mathbf{s}^{(k)} = -0.969 > c_2 \mathbf{g}_k^T \mathbf{s}^{(k)} = 0.5 \times (-2) = -1$$

所以条件 1°, 2° 均成立, $\lambda_k = 0.125$, 而 $\mathbf{x}^{(k+1)} = (0.125, 0)^T$ 。

计算过程列在下表中。其中“-”号表示条件不成立, “+”号表示条件成立。

j	$\mathbf{x}^{(k)}$	f_k	λ	$\mathbf{x}^{(k+1)}$	f_{k+1}	$\mathbf{g}_k^T \mathbf{s}^{(k)}$	$\mathbf{g}_{k+1}^T \mathbf{s}^{(k)}$	条件 1°	条件 2°
0	$(0, 0)^T$	1	1	$(1, 0)^T$	100	-2		-	
1	$(0, 0)^T$	1	0.5	$(0.5, 0)^T$	6.25	-2		-	
2	$(0, 0)^T$	1	0.25	$(0.25, 0)^T$	0.953	-2		-	
3	$(0, 0)^T$	1	0.125	$(0.125, 0)^T$	0.790	-2	-0.969	+	+

将例 2.3 和例 2.4 进行比较, 即可看出: 不精确一维搜索方法有明显的优点: 方法简单, 收敛速度比精确一维搜索的直接法快得多。这里只迭代了四次, 就求得了满足条件 1°, 2° 的步长 λ_k 。

不精确一维搜索算法二(二次插值法)

设 $\varphi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$, 若已知 (λ_1, φ_1) , (λ_2, φ_2) 和 (λ_1, φ'_1) , 其中 $\varphi_i = \varphi(\lambda_i)$ ($i = 1, 2$), $\varphi'_1 = \varphi'(\lambda_1)$ 。求通过 (λ_1, φ_1) , (λ_2, φ_2) 两点和在点 (λ_1, φ_1) 处具有相同切线的二次曲线方程 $g(\lambda) = a\lambda^2 +$

$b\lambda + c$, 则

$$g(\lambda_1) = \varphi_1, g(\lambda_2) = \varphi_2, g'(\lambda_1) = \varphi'_1$$

因为 $g'(\lambda) = 2\lambda a + b$, 所以

$$\hat{\lambda} = -b/2a \quad (2-27)$$

只要算出 a 和 b , 即可求得 $\hat{\lambda}$, 用 $\hat{\lambda}$ 作为 $\varphi(\lambda)$ 的近似极小点, 可以证明

$$\hat{\lambda} = \lambda_1 + \frac{1}{2}(\lambda_2 - \lambda_1) / \left[1 + \frac{\varphi_1 - \varphi_2}{(\lambda_2 - \lambda_1)\varphi'_1} \right] \quad (2-28)$$

证明留做习题。

算法二的计算步骤如下:

(1) 给定 $c_1 \in (0, 1)$, $c_2 \in (c_1, 1)$, $T > 0$ (例如取 $T = 0.1$), 令 $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = +\infty$, $j = 0$ 。

(2) 若 λ_2 满足条件 $1^\circ, 2^\circ$, 则令 $\lambda_k = \lambda_2$, 计算结束; 否则转第 3 步。

(3) 计算 $\varphi_1, \varphi_2, \varphi'_1 = \nabla f(\mathbf{x}^{(k)} + \lambda_1 \mathbf{s}^{(k)})^T \mathbf{s}^{(k)}$ 和 $\hat{\lambda}$, 令 $j = j + 1$ 。

(4) 若 $\lambda_1 < \hat{\lambda} < \lambda_2$, 则令 $\lambda_3 = \lambda_2$, $\lambda_2 = \hat{\lambda}$, 返回第 2 步; 若 $\lambda_2 \leq \hat{\lambda} < \lambda_3$, 则令 $\lambda_1 = \lambda_2$, $\lambda_2 = \hat{\lambda}$, 返回第 2 步; 否则转第 5 步。

(5) 若 $\hat{\lambda} \leq \lambda_1$, 则令 $\hat{\lambda} = \lambda_1 + T\Delta\lambda$; 若 $\hat{\lambda} \geq \lambda_3$, 则令 $\hat{\lambda} = \lambda_3 - T\Delta\lambda$ (其中 $\Delta\lambda = \lambda_3 - \lambda_1$), 令 $\lambda_2 = \hat{\lambda}$, 返回第 2 步。

对不精确一维搜索有兴趣的读者可参看有关文献, 例如文献 26, 35, 61。

3.3 求多变量函数极值的基本下降法

本节介绍求多个变量函数的无约束极小值的一些基本的下降算法: 最速下降法、Newton 法和阻尼 Newton 法。掌握这些算法有两方面的意义: ① 这些算法为我们提供了一些简单而直接的求解

方案; ② 更重要的也许是他们为求解无约束极值问题的难度和收敛速度提供了可供进行比较的标准, 为建立更有效的算法奠定了基础。

3.3.1 最速下降法

最速下降法, 也叫梯度法, 是人们用来求多个变量函数极值问题的最早的一种方法。后来提出的不少方法都是人们企图改进这种算法的结果。下面作简要介绍。

1. 基本想法

设 $f(x) \in C^1$ 。由于函数 $f(x)$ 沿负梯度方向 $-\nabla f(x)$ 下降最快, 所以选取它作为每次迭代的搜索方向, 即 $s^{(k)} = -\nabla f(x^{(k)})$ 。因此梯度法的计算步骤是: 任给一个初始点 $x^{(0)}$, 计算 $s^{(0)} = -\nabla f(x^{(0)})$, 从 $x^{(0)}$ 出发沿 $s^{(0)}$ 方向在半直线 $x^{(0)} + \lambda s^{(0)} (\lambda \geq 0)$ 上求得 $f(x)$ 的第一个近似极小点 $x^{(1)}$, $x^{(1)} = x^{(0)} + \lambda_0 s^{(0)}$, 再以 $x^{(1)}$ 为起点, 重复上述过程求得 $f(x)$ 的第二个近似极小点 $x^{(2)} = x^{(1)} + \lambda_1 s^{(1)}$, \dots , 直到得到满意的结果为止。

2. 最速下降方向

为什么在 $x^{(0)}$ 的附近, $f(x)$ 沿 $-\nabla f(x^{(0)})$ 的方向可使函数值下降最快?

设 p 为任给的一个单位向量, ε 为正实数, 则 $f(x)$ 在 $x^{(0)}$ 处的 Taylor 展开式为

$$f(x^{(0)} + \varepsilon p) = f(x^{(0)}) + \varepsilon (\nabla f(x^{(0)}))^T p + o(\varepsilon)$$

其中 $o(\varepsilon)$ 是 ε 的高阶无穷小, 而

$$\begin{aligned} (\nabla f(x^{(0)}))^T p &= \|\nabla f(x^{(0)})\| \cdot \|p\| \cos(\nabla f(x^{(0)}), p) \\ &= \|\nabla f(x^{(0)})\| \cos(\nabla f(x^{(0)}), p) \end{aligned}$$

所以当 $(\nabla f(x^{(0)}), p) = \pi$ 时, 即

$$p = -\nabla f(x^{(0)}) / \|\nabla f(x^{(0)})\|$$

时, $f(x)$ 在 $x^{(0)}$ 点附近下降最快。因此, 称 p 为 $f(x)$ 在 $x^{(0)}$ 点的最

速下降方向,最速下降法的名称就由此而来。

3. 梯度法的迭代步骤

(1) 给定初始点 $\mathbf{x}^{(0)}$, 精度 $\varepsilon > 0$, 令 $k = 0$;

(2) 计算 $\nabla f(\mathbf{x}^{(k)})$;

(3) 若 $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, 则迭代结束, 取 $\mathbf{x}^* = \mathbf{x}^{(k)}$, 否则转到(4);

(4) 这时 $\|\nabla f(\mathbf{x}^{(k)})\| \geq \varepsilon$, 用精确一维搜索求

$$\varphi(\lambda) = f(\mathbf{x}^{(k)} - \lambda \nabla f(\mathbf{x}^{(k)}))$$

的一个极小点 λ_k , 使 $f(\mathbf{x}^{(k)} - \lambda_k \nabla f(\mathbf{x}^{(k)})) < f(\mathbf{x}^{(k)})$;

(5) 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \nabla f(\mathbf{x}^{(k)})$, $k = k + 1$, 返回(2)。

4. 计算框图

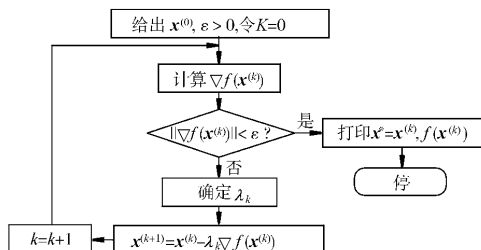


图 3-3

5. 收敛性定理

设 $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 连续可微, $\mathbf{x}^{(0)} \in \mathbf{R}^n$, 如果水平集 $L = \{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ 有界, 则梯度法或在有限步迭代后停止, 或者得到点列 $\{\mathbf{x}^{(k)}\}$, 它的任何极限点都是 $f(\mathbf{x})$ 的稳定点。

证明可参看文献 21, 27, 34。若进一步假设 $f(\mathbf{x})$ 为凸函数, 则应用梯度法, 或在有限步迭代后达到 $f(\mathbf{x})$ 的最小点; 或者点列 $\{\mathbf{x}^{(k)}\}$ 的任何极限点都是 $f(\mathbf{x})$ 的最小点。

6. 优缺点评述

优点: (1) 方法简单, 每迭代一次的工作量较小, 所需要的存储量也少; (2) 从一个不好的初始点 $\mathbf{x}^{(0)}$ 出发, 也能保证算法的收敛性。

缺点: (1) 在极小点附近收敛得很慢。从表面上看, 这似乎与“最速下降”矛盾, 其实不然, 因为梯度是函数的局部性质, 从局部看在一点附近下降得快, 但从总体上来看可能走许多弯路。事实上, 若 $\mathbf{x}^{(k+1)}$ 是沿方向 $\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$ 用一维搜索求得的, 即 λ_k 为 $\varphi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$ 的极小点, 那么

$$\left. \frac{d\varphi}{d\lambda} \right|_{\lambda=\lambda_k} = (\nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}))^T \mathbf{s}^{(k)} = 0$$

即 $(\nabla f(\mathbf{x}^{(k+1)}))^T \nabla f(\mathbf{x}^{(k)}) = 0$ 。

这说明梯度法逐次下降的方向是彼此正交的, 因此在极小点附近, 梯度法产生的点列逼近函数的极小点越来越慢。

(2) 梯度法的收敛速度与变量的尺度关系很大。例如, 设 $f(x_1, x_2) = x_1^2 + x_2^2$, 取 $\mathbf{x}^{(0)} = (2, 2)^T$, 则 $\nabla f(\mathbf{x}^{(0)}) = (4, 4)^T$, 沿 $-\nabla f(\mathbf{x}^{(0)})$ 进行一维搜索可得 $\lambda_0 = 0.5$, 因为 $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \lambda_0 \nabla f(\mathbf{x}^{(0)}) = (0, 0)^T$, 即迭代一次就达到了极小点 $(0, 0)^T$ 。但是, 对于 $g(x_1, x_2) = x_1^2 + 25x_2^2$, 取 $\mathbf{x}^{(0)} = (2, 2)^T$, 则 $\nabla g(\mathbf{x}^{(0)}) = (4, 100)^T$, 沿 $-\nabla g(\mathbf{x}^{(0)})$ 进行一维搜索, 可得 $\lambda_0 = 0.02$, 所以 $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \lambda_0 \nabla g(\mathbf{x}^{(0)}) = (1.92, 0)^T$ 。再从 $\mathbf{x}^{(1)}$ 出发, 继续进行迭代, 可得 $\mathbf{x}^{(2)}, \dots$, 约需进行 10 次迭代, 才能达到极小点 $(0, 0)^T$ 。由此可见梯度法收敛的快慢与变量的尺度关系很大。

(3) 梯度法关于小的扰动是不稳定的, 而小的扰动在计算过程中是容易产生的, 例如舍入误差的存在, 或一维搜索时步长确定得不准确, 这样就可能破坏方法的收敛性。

3.3.2 Newton 法

在 3.2 中讲过求解一维问题的 Newton 法, 它容易推广到多维

的情况。这个方法也是求解无约束极值问题的最古老的算法之一，已发展成为一类算法——Newton 型方法。

1. 基本想法与迭代公式

与一维问题类似，在局部，用一个二次函数 $\varphi(x)$ 近似地代替目标函数 $f(x)$ ， $f(x) \approx \varphi(x)$ ，然后用 $\varphi(x)$ 的极小点作为 $f(x)$ 的近似极小点。

设 $\mathbf{x}^{(k)}$ 为 $f(x)$ 的一个近似极小点，将 $f(x)$ 在 $\mathbf{x}^{(k)}$ 点进行 Taylor 展开，并略去高于二次的项，则得

$$f(\mathbf{x}) \approx \varphi(\mathbf{x}) = f(\mathbf{x}^{(k)}) + (\nabla f(\mathbf{x}^{(k)}))^T (\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k)})^T \nabla^2 f(\mathbf{x}^{(k)}) (\mathbf{x} - \mathbf{x}^{(k)}) \quad (3-1)$$

容易求得

$$\nabla \varphi(\mathbf{x}) = \nabla f(\mathbf{x}^{(k)}) + \nabla^2 f(\mathbf{x}^{(k)}) (\mathbf{x} - \mathbf{x}^{(k)})$$

令 $\nabla \varphi(\mathbf{x}) = \mathbf{0}$ ，得 $\varphi(x)$ 的极小点为

$$\hat{\mathbf{x}} = \mathbf{x}^{(k)} - (\nabla^2 f(\mathbf{x}^{(k)}))^{-1} \nabla f(\mathbf{x}^{(k)}) \quad (3-2)$$

取 $\hat{\mathbf{x}}$ 作为 $f(x)$ 的近似极小点，这样就得到 Newton 法的迭代公式：

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\nabla^2 f(\mathbf{x}^{(k)}))^{-1} \nabla f(\mathbf{x}^{(k)}) \quad (3-2')$$

从上述可以看出：对于二次函数，如果它的极小点存在，则用 Newton 法只需迭代一次就可得到它的极小点。

2. 举 例

求 $f(x_1, x_2) = x_1^2 + 25x_2^2$ 的极小点。

解 取初始点 $\mathbf{x}^{(0)} = (2, 2)^T$ ，则

$$\nabla f(\mathbf{x}^{(0)}) = (4, 100)^T, \quad \nabla^2 f(\mathbf{x}^{(0)}) = \begin{pmatrix} 2 & 0 \\ 0 & 50 \end{pmatrix}$$

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - (\nabla^2 f(\mathbf{x}^{(0)}))^{-1} \nabla f(\mathbf{x}^{(0)}) \\ &= \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 1/2 & 0 \\ 0 & 1/50 \end{pmatrix} \begin{pmatrix} 4 \\ 100 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned}$$

即 $\mathbf{x}^{(1)}$ 为 $f(x_1, x_2)$ 的极小点。迭代一次就求得了 $f(x_1, x_2)$ 的极小点 $\mathbf{x}^* = (0, 0)^T$ 。

3. 收敛速度及优缺点评述

设 \mathbf{x}^* 为 $f(\mathbf{x})$ 的极小点, 可以证明^[16], 在适当的条件下, Newton 法的收敛速度估计式为

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| < M \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2 \quad (3-3)$$

即 Newton 法至少是二阶收敛的。所以, Newton 法的突出优点是收敛速度快, 而缺点是: (1) 要求 $f(\mathbf{x})$ 二阶连续可微; (2) 在迭代中, 要计算 $[\nabla^2 f(\mathbf{x})]^{-1}$ 是困难的; (3) 从 (3-3) 式可看出, 初始点 $\mathbf{x}^{(0)}$ 不能离极小点 \mathbf{x}^* 太远, 否则迭代可能不收敛。为了克服这个缺点, 人们对算法作了修正, 提出了“阻尼 Newton 法”, 又称为修正 Newton 法。

3.3.3 阻尼 Newton 法

在 Newton 法中, 步长 λ_k 总是取为 1。在阻尼 Newton 法中, 每步迭代沿方向

$$\mathbf{s}^{(k)} = -(\nabla^2 f(\mathbf{x}^{(k)}))^{-1} \nabla f(\mathbf{x}^{(k)})$$

进行一维搜索来决定 λ_k , 即取 λ_k , 使

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) = \min_{\lambda \geq 0} f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$$

而用迭代公式

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k (\nabla^2 f(\mathbf{x}^{(k)}))^{-1} \nabla f(\mathbf{x}^{(k)}) \quad (3-4)$$

来代替 (3-2')。阻尼牛顿法保持了牛顿法快速收敛的优点, 又不要求初始点选得很好, 因而在实际应用中取得了较好的效果。当然 (3-4) 式也要求海赛矩阵 $\nabla^2 f(\mathbf{x}^{(k)})$ 的逆阵。

1. 迭代程序

(1) 给定初始点 $\mathbf{x}^{(0)}$, 精度 $\epsilon > 0$, 令 $k = 0$ 。

(2) 计算 $\nabla f(\mathbf{x}^{(k)})$, 若 $\|\nabla f(\mathbf{x}^{(k)})\| < \epsilon$ 成立, 则迭代停止, $\mathbf{x}^{(k)}$ 即为所求; 否则转到 (3)。

- (3) 计算 $[\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$ 及 $\mathbf{s}^{(k)} = -[\nabla^2 f(\mathbf{x}^{(k)})]^{-1} \nabla f(\mathbf{x}^{(k)})$ 。
- (4) 沿 $\mathbf{s}^{(k)}$ 进行一维搜索, 决定步长 λ_k 。
- (5) 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$, $k = k + 1$, 返回(2)。

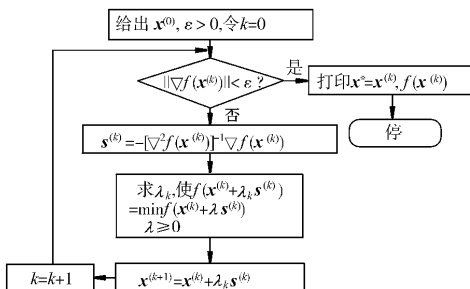


图 3-4

2. 计算框图(图 3-4)

3. 收敛定理

设 $f(\mathbf{x}) \in C^2$, $\nabla^2 f(\mathbf{x})$ 为正定矩阵, 水平集 $\{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\}$ 有界, 则由阻尼牛顿法得到的点列 $\{\mathbf{x}^{(k)}\}$ 有如下的性质:

- (1) $\{f(\mathbf{x}^{(k)})\}$ 为严格单调下降数列;
- (2) $\{\mathbf{x}^{(k)}\}$ 的任一极限点 $\hat{\mathbf{x}}$ 必为 $f(\mathbf{x})$ 的极小点。

此收敛定理的证明参看文献 27。

4. 算法举例

设 $f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$, 求 $\min f(\mathbf{x})$ 。

解 取 $\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, 则 $f(\mathbf{x}^{(0)}) = -3$

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4 - 2x_2, \quad \frac{\partial f}{\partial x_2} = 4x_2 - 2x_1$$

$$\text{因 } \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -4 \\ 2 \end{pmatrix}, \quad \nabla^2 f(\mathbf{x}) = \begin{pmatrix} 2 & -2 \\ -2 & 4 \end{pmatrix}$$

$$\text{故 } (\nabla^2 f(\mathbf{x}))^{-1} = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

$$\text{于是 } \mathbf{s}^{(0)} = -(\nabla^2 f(\mathbf{x}^{(0)}))^{-1} \nabla f(\mathbf{x}^{(0)})$$

$$= - \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} -4 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

$$\varphi(\lambda) = f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)}) = f(1 + 3\lambda, 1 + \lambda)$$

$$= (1 + 3\lambda)^2 + 2(1 + \lambda)^2 -$$

$$4(1 + 3\lambda) - 2(1 + 3\lambda)(1 + \lambda)$$

$$= 5\lambda^2 - 10\lambda - 3$$

$$\varphi'(\lambda) = 10\lambda - 10, \text{ 令 } \varphi'(\lambda) = 0, \text{ 得 } \lambda_0 = 1, \text{ 所以}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{s}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$\nabla f(\mathbf{x}^{(1)}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

即迭代一次就达到了 $f(\mathbf{x})$ 极小点 $\mathbf{x}^* = (4, 2)^T$, 而对应的极小值为

$$f(\mathbf{x}^*) = f(4, 2) = -8$$

3.4 共轭方向法和共轭梯度法

上一节讲的最速下降法, 计算步骤简单, 但收敛速度太慢, 而牛顿法和阻尼牛顿法收敛速度快, 但要计算二阶偏导数矩阵及其逆阵, 计算量太大。因此人们希望找到一种方法, 它兼有这两种方法的优点, 又能克服它们的缺点。共轭方向法就是这样的一类方法, 它比最速下降法的收敛速度要快得多, 同时又避免了像牛顿法所要求的海赛矩阵的计算、存贮和求逆。

共轭方向法, 主要是其中的共轭梯度法, 它对一般目标函数的无约束优化问题的求解具有较高的效率, 因此在无约束优化算法

中占有重要的地位,是目前最常用的方法之一。由于它的计算公式简单,存储量少,可以用来求解比较大的问题,特别是用于最优控制问题时,效果很好,引起了人们的重视和兴趣。

由于最速下降方向不是理想的搜索方向,要提高收敛速度,就要寻找其他的搜索方向。为了对建立新的搜索方向提供依据,就需要一个判断算法优劣的标准。由于二次函数是一种简单的非线性函数,而一般的目标函数 $f(x)$ 在极小点附近的性态(当 $f(x) \in C^2$ 时)又近似于二次函数,所以人们常常把二次函数作为衡量一个算法优劣的尺子。如果一种算法对于二次函数的效果尚且不好,就难以指望它对于一般函数会有更好的效果了。因此在研究算法时,人们常常是首先对二次函数进行仔细分析,然后再推广到一般的目标函数。

3.4.1 共轭方向和它的一些性质

定义 4.1 设 A 为 n 阶对称矩阵, p, q 为 n 维列向量,若

$$p^T A q = 0 \quad (4-1)$$

则称向量 p 和 q 为 A -正交,或关于 A -共轭。

如果 $A = I_n$,则(4-1)成为 $p^T q = 0$,这就是通常意义下的正交性,故 A -共轭或 A -正交是正交概念的推广。

如果对于有限个向量 p_1, p_2, \dots, p_m ,有 $p_i^T A p_j = 0$ ($i \neq j, j = 1, 2, \dots, m$) 成立,则称这个向量组为 A -正交(或共轭)向量组,也称它们是一组 A 共轭方向。

定理 4.1 设 V 为 n 维欧氏空间,向量 $p_i \in V$ ($i = 1, 2, \dots, n$) 是线性无关的, $q \in V$, 若 q 与 p_1, p_2, \dots, p_n 都正交,则 $q = 0$ 。

证明 因为 p_1, p_2, \dots, p_n 线性无关,所以它们构成 V 的一组基,于是有

$$q = k_1 p_1 + k_2 p_2 + \dots + k_n p_n$$

注意到 q 与 p_1, \dots, p_n 都正交,所以

$$(p_i, q) = 0, i = 1, 2, \dots, n$$

由于内积

$$\begin{aligned}(q, q) &= (q, k_1 p_1 + \dots + k_n p_n) \\ &= k_1 (q, p_1) + k_2 (q, p_2) + \dots + k_n (q, p_n) = 0\end{aligned}$$

所以 $q = 0$ ■

定理 4.2 设 A 为 n 阶正定矩阵, p_1, p_2, \dots, p_n 为 A -共轭的 n 维非零向量组, 则此向量组必为线性无关的。

证明 设 $k_1 p_1 + k_2 p_2 + \dots + k_n p_n = 0$

用 $p_i^T A$ 左乘上式得

$$k_i p_i^T A p_i = 0, i = 1, 2, \dots, n$$

因为 $p_i \neq 0$, 且 A 为正定矩阵, 所以 $p_i^T A p_i > 0$, 于是必有

$$k_i = 0, i = 1, 2, \dots, n$$

因此 p_1, p_2, \dots, p_n 为线性无关的向量组。 ■

下面来研究 n 元二次函数的无约束优化问题:

$$\min f(x) = c + b^T x + \frac{1}{2} x^T H x \quad (4-2)$$

其中 c 为常数, x, b 为 n 维列向量, H 为 n 阶对称正定矩阵。这时 $\nabla f(x) = b + Hx$, $\nabla^2 f(x) = H$, $f(x)$ 有惟一的极小点, 在极小点 x^* 处,

$$\nabla f(x^*) = b + Hx^* = 0$$

则 $x^* = -H^{-1}b$

对 n 元二次函数 $f(x)$, 我们有下述结果。

定理 4.3 设 H 为 n 阶对称正定矩阵, $s^{(0)}, s^{(1)}, \dots, s^{(n-1)} \in \mathbb{R}^n$ 是一组 H 共轭方向, 对问题 (4-2), 若从任一初始点 $x^{(0)} \in \mathbb{R}^n$ 出发, 依次沿方向 $s^{(0)}, s^{(1)}, \dots, s^{(n-1)}$ 进行精确一维搜索, 则至多经过 n 次迭代, 即可求得 $f(x)$ 的最小点。

证明 从 $x^{(0)}$ 点出发, 依次沿方向 $s^{(0)}, s^{(1)}, \dots, s^{(n-1)}$ 进行精

确一维搜索得到的迭代点是

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}, \quad k = 0, 1, \dots, n-1$$

其中 λ_k 使

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) = \min_{\lambda \geq 0} f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)}), \quad k = 0, 1, \dots, n-1 \quad (4-3)$$

于是可得到

$$\begin{aligned} \nabla f(\mathbf{x}^{(k+1)}) &= H\mathbf{x}^{(k+1)} + \mathbf{b} = H(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) + \mathbf{b} \\ &= \nabla f(\mathbf{x}^{(k)}) + \lambda_k H\mathbf{s}^{(k)}, \quad k = 0, 1, \dots, n-1 \end{aligned}$$

由上式进一步可得到

$$\begin{aligned} \nabla f(\mathbf{x}^{(k+1)}) &= \nabla f(\mathbf{x}^{(k-1)}) + \lambda_{k-1} H\mathbf{s}^{(k-1)} + \lambda_k H\mathbf{s}^{(k)} = \dots \\ &= \nabla f(\mathbf{x}^{(j)}) + \sum_{i=j}^k \lambda_i H\mathbf{s}^{(i)}, \quad j = 0, 1, \dots, k \end{aligned}$$

对上式右乘 $\mathbf{s}^{(j)}$, 得

$$\begin{aligned} \nabla f(\mathbf{x}^{(k+1)})^T \mathbf{s}^{(j)} &= \nabla f(\mathbf{x}^{(j)})^T \mathbf{s}^{(j)} + \sum_{i=j}^k \lambda_i (\mathbf{s}^{(i)})^T H\mathbf{s}^{(j)} \\ j &= 0, 1, \dots, k \end{aligned}$$

因为 $\mathbf{s}^{(0)}, \dots, \mathbf{s}^{(n-1)}$ 是 H 共轭方向, 所以

$$\begin{aligned} \nabla f(\mathbf{x}^{(k+1)})^T \mathbf{s}^{(j)} &= \nabla f(\mathbf{x}^{(j)})^T \mathbf{s}^{(j)} + \lambda_j (\mathbf{s}^{(j)})^T H\mathbf{s}^{(j)} \\ &= (\nabla f(\mathbf{x}^{(j)}) + \lambda_j H\mathbf{s}^{(j)})^T \mathbf{s}^{(j)} \end{aligned}$$

或

$$\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{s}^{(j)} = \nabla f(\mathbf{x}^{(j+1)})^T \mathbf{s}^{(j)}, \quad j = 0, 1, \dots, k \quad (4-4)$$

又由(4-3)知

$$\left. \frac{df(\mathbf{x}^{(j)} + \lambda \mathbf{s}^{(j)})}{d\lambda} \right|_{\lambda=\lambda_j} = 0, \quad j = 0, 1, \dots, n-1$$

而 $\left. \frac{df(\mathbf{x}^{(j)} + \lambda \mathbf{s}^{(j)})}{d\lambda} \right|_{\lambda=\lambda_j} = \nabla f(\mathbf{x}^{(j+1)})^T \mathbf{s}^{(j)}$, 因此

$$\nabla f(\mathbf{x}^{(j+1)})^T \mathbf{s}^{(j)} = 0, \quad j = 0, 1, \dots, n-1 \quad (4-5)$$

综合上式和(4-4)式可得: 对 $k = 0, 1, \dots, n-1$, 有

$$\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{s}^{(j)} = \mathbf{0}, j = 0, 1, \dots, k \quad (4-6)$$

特别地, 当 $k = n - 1$ 时, 我们得到

$$\nabla f(\mathbf{x}^{(n)})^T \mathbf{s}^{(j)} = 0, j = 0, 1, \dots, n - 1 \quad (4-7)$$

由定理 4.2 知: $\mathbf{s}^{(0)}, \dots, \mathbf{s}^{(n-1)}$ 是线性无关的, 所以由 (4-7) 式和定理 4.1 知

$$\nabla f(\mathbf{x}^{(n)}) = \mathbf{0}$$

即 $\mathbf{x}^{(n)}$ 是 $f(\mathbf{x})$ 的稳定点, 又因 $H > 0$, 故 $f(\mathbf{x})$ 为严格凸函数, 因此 $\mathbf{x}^{(n)}$ 是问题 (4-2) 的最优解。这说明: 至多经过 n 次迭代, 即可求得 $f(\mathbf{x})$ 的最小点

3.4.2 共轭方向的生成与共轭梯度法

由上述可见, 只要能选取一组 H 共轭的方向 $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n-1)}$, 就可以用上述方法在 n 步之内求得 n 元二次函数 $f(\mathbf{x})$ 的极小点。我们称上述算法为共轭方向法。这种算法对于形如 (4-2) 的二次函数, 具有有限步收敛的性质, 因为应用共轭方向法, 不超过 n 次迭代, 即可求得 n 元二次函数的极小点 \mathbf{x}^* , 所以有的书也称这种有限步收敛性为二次收敛, 即称它是一种二次收敛的算法。

共轭方向的选取有很大的任意性, 而对应于不同的一组共轭方向就有不同的共轭方向法。作为一种迭代算法, 我们自然希望共轭方向能在迭代过程中逐次生成。下面介绍一种生成共轭方向的方法, 它是利用每次一维最优化所得到的点 $\mathbf{x}^{(i)}$ 处的梯度来生成共轭方向, 这种方法称为共轭梯度法, 其具体作法如下:

任给初始点 $\mathbf{x}^{(0)}$, 令 $\mathbf{s}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$, 由 (4-3) 式定出 λ_0 , 令

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{s}^{(0)}$$

$$\mathbf{s}^{(1)} = -\nabla f(\mathbf{x}^{(1)}) + \mu_{10} \mathbf{s}^{(0)}$$

其中 μ_{10} 为待定系数, 要使 $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}$ 是 H 共轭的, 即使

$$(\mathbf{s}^{(1)})^T H \mathbf{s}^{(0)} = 0$$

成立, 由此可求得

$$\mu_{10} = \frac{(\nabla f(\mathbf{x}^{(1)}))^T \mathbf{H} \mathbf{s}^{(0)}}{(\mathbf{s}^{(0)})^T \mathbf{H} \mathbf{s}^{(0)}} \quad (4-8)$$

将 μ_{10} 代入 $\mathbf{s}^{(1)}$ 的表达式中, 就可以求出搜索方向 $\mathbf{s}^{(1)}$, 令

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{s}^{(1)}$$

$$\mathbf{s}^{(2)} = -\nabla f(\mathbf{x}^{(2)}) + \mu_{20} \mathbf{s}^{(0)} + \mu_{21} \mathbf{s}^{(1)}$$

其中 λ_1 由 (4-3) 式确定。而 μ_{20} 和 μ_{21} 为待定系数, 要使 $\mathbf{s}^{(2)}$ 与 $\mathbf{s}^{(0)}$, $\mathbf{s}^{(1)}$ 分别是 \mathbf{H} 共轭的, 即要

$$(\mathbf{s}^{(2)})^T \mathbf{H} \mathbf{s}^{(0)} = 0, \quad (\mathbf{s}^{(2)})^T \mathbf{H} \mathbf{s}^{(1)} = 0$$

由此可求得

$$\mu_{20} = \frac{(\nabla f(\mathbf{x}^{(2)}))^T \mathbf{H} \mathbf{s}^{(0)}}{(\mathbf{s}^{(0)})^T \mathbf{H} \mathbf{s}^{(0)}}, \quad \mu_{21} = \frac{(\nabla f(\mathbf{x}^{(2)}))^T \mathbf{H} \mathbf{s}^{(1)}}{(\mathbf{s}^{(1)})^T \mathbf{H} \mathbf{s}^{(1)}}, \quad \dots$$

类似地可得

$$\begin{cases} \mathbf{s}^{(i)} = -\nabla f(\mathbf{x}^{(i)}) + \sum_{j=0}^{i-1} \mu_{ij} \mathbf{s}^{(j)}, & 0 \leq i \leq n-1 \\ \mu_{ij} = \frac{(\nabla f(\mathbf{x}^{(i)}))^T \mathbf{H} \mathbf{s}^{(j)}}{(\mathbf{s}^{(j)})^T \mathbf{H} \mathbf{s}^{(j)}}, & j = 0, 1, \dots, i-1 \end{cases} \quad (4-9)$$

应用上述方法就可生成 n 个关于 \mathbf{H} 共轭的方向 $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n-1)}$, 但是仔细分析一下上述公式就会发现, 在求 μ_{ij} 时, 需要用到 $f(\mathbf{x})$ 的海赛矩阵 \mathbf{H} , 这是我们所不希望的。因此, 需要对上述公式作进一步的化简, 使在生成共轭方向 $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n-1)}$ 时, 不需要直接用到海赛矩阵 \mathbf{H} 。因为

$$\nabla f(\mathbf{x}) = \mathbf{b} + \mathbf{H} \mathbf{x}$$

为以后书写简便, 令

$$\nabla f(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \quad \nabla f(\mathbf{x}^{(i)}) = \mathbf{g}(\mathbf{x}^{(i)}) = \mathbf{g}_i$$

故

$$\mathbf{g}_i = \mathbf{b} + \mathbf{H} \mathbf{x}^{(i)}, \quad \mathbf{g}_{i+1} = \mathbf{b} + \mathbf{H} \mathbf{x}^{(i+1)}$$

$$\mathbf{g}_{i+1} - \mathbf{g}_i = \mathbf{H}(\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}) \quad (4-10)$$

但是

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \lambda_i \mathbf{s}^{(i)}$$

所以
$$\mathbf{g}_{i+1} - \mathbf{g}_i = \lambda_i \mathbf{H} \mathbf{s}^{(i)} \quad (4-11)$$

因此(4-9)式中 μ_{ij} 可改写为

$$\mu_{ij} = \frac{\mathbf{g}_i^T (\mathbf{g}_{j+1} - \mathbf{g}_j)}{(\mathbf{s}^{(j)})^T (\mathbf{g}_{j+1} - \mathbf{g}_j)}, j = 0, 1, \dots, i-1 \quad (4-12)$$

注意到: $\mathbf{s}^{(0)} = -\mathbf{g}_0$ 及 $\mathbf{g}_1^T \mathbf{g}_0 = 0$, 由(4-12)式易得

$$\mu_{10} = \|\mathbf{g}_1\|^2 / \|\mathbf{g}_0\|^2 \quad (4-13)$$

在(4-6)式中, 令 $j = 0, k = 1$, 得

$$\mathbf{g}_2^T (-\mathbf{g}_0) = 0 \quad \text{或} \quad \mathbf{g}_2^T \mathbf{g}_0 = 0$$

利用这一结果, 并注意到 $\mathbf{g}_2 \perp \mathbf{g}_1$, 易得 $\mu_{20} = 0$ 。

由(4-9)式的第一个式子得

$$\mathbf{s}^{(1)} = -\mathbf{g}_1 + \mu_{10} \mathbf{s}^{(0)} = -\mathbf{g}_1 - \mu_{10} \mathbf{g}_0$$

在(4-12)式中, 当 $i = 2, j = 1$ 时, 并将上式代入得

$$\mu_{21} = \frac{\mathbf{g}_2^T (\mathbf{g}_2 - \mathbf{g}_1)}{(\mathbf{s}^{(1)})^T (\mathbf{g}_2 - \mathbf{g}_1)} = \|\mathbf{g}_2\|^2 / \|\mathbf{g}_1\|^2 \quad (4-14)$$

类似地可得

$$\mu_{ij} = 0, \quad 0 \leq j \leq i-2 \quad (4-15)$$

$$\mu_{i,i-1} = \|\mathbf{g}_i\|^2 / \|\mathbf{g}_{i-1}\|^2 \quad (4-16)$$

其中 $\|\mathbf{g}_i\| \neq 0, i = 0, 1, \dots, n-1$ 。因此

$$\mathbf{s}^{(i)} = -\mathbf{g}_i + \mu_{i,i-1} \mathbf{s}^{(i-1)} \quad (4-17)$$

由(4-16)(4-17)式求得的方向 $\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n-1)}$ 是关于 \mathbf{H} 共轭的方向, 按照这些公式确定共轭方向的算法就是通常所说的 F-R 共轭梯度法, 它是由 Fletcher 和 Reeves 于 1964 年提出的。因为公式中不显含 $f(\mathbf{x})$ 的海赛矩阵, 所以可毫无困难地将此方法应用于一般的目标函数。

3.4.3 共轭梯度法的计算步骤和程序框图

计算步骤如下:

1. 给定初始点 $\mathbf{x}^{(0)}$ 及精度 $\varepsilon > 0$;
2. 计算 $\mathbf{g}_0 = \nabla f(\mathbf{x}^{(0)})$, 令 $\mathbf{s}^{(0)} = -\mathbf{g}_0$, $k = 0$;
3. 求 $\min_{\lambda \geq 0} f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$ 决定 λ_k , 计算

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}, \quad \mathbf{g}_{k+1} = \nabla f(\mathbf{x}^{(k+1)})$$
4. 若 $\|\mathbf{g}_{k+1}\| < \varepsilon$, 迭代结束, 否则转 5;
5. 若 $k < n - 1$, 计算

$$\mu_{k+1} \triangleq \mu_{k+1,k} = \|\mathbf{g}_{k+1}\|^2 / \|\mathbf{g}_k\|^2$$

$$\mathbf{s}^{(k+1)} = -\mathbf{g}_{k+1} + \mu_{k+1} \mathbf{s}^{(k)}$$

令 $k = k + 1$, 转回 3;

若 $k = n - 1$, 令 $\mathbf{x}^{(0)} = \mathbf{x}^{(n)}$ 。转回 2。

程序框图如图 3-5 所示。

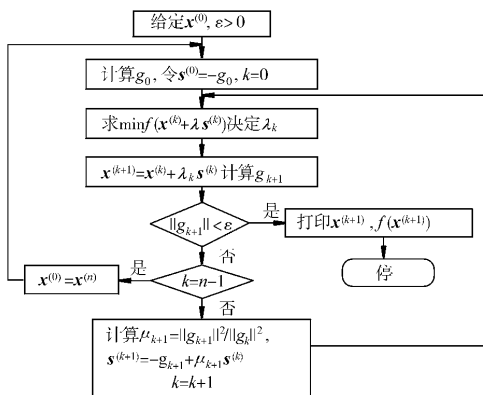


图 3-5

应该注意的是:由于 n 维问题的共轭方向只有 n 个,在 n 步之后,继续进行计算是无意义的;而且,舍入误差的积累也越来越大。因此,在实际应用时,多是采取计算 n 步,就以所得的近似极小点 $\mathbf{x}^{(n)}$ 为初始点,重新开始迭代。实际计算说明:这样做一般都可取得比较好的效果。上述算法的计算程序可在文献 46 中找到。

3.4.4 应用举例

【例 4.1】 求二次函数 $f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$ 的极小点。

解 取初始点 $\mathbf{x}^{(0)} = (1, 1)^T$, 则 $f(\mathbf{x}^{(0)}) = -3$,

$$\text{因 } \frac{\partial f}{\partial x_1} = 2x_1 - 4 - 2x_2, \quad \frac{\partial f}{\partial x_2} = 4x_2 - 2x_1$$

$$\text{故 } \mathbf{g}_0 = \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -4 \\ 2 \end{pmatrix}, \quad \mathbf{s}^{(0)} = -\mathbf{g}_0 = \begin{pmatrix} 4 \\ -2 \end{pmatrix}$$

$$\text{因 } \mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)} = \begin{pmatrix} 1 + 4\lambda \\ 1 - 2\lambda \end{pmatrix}$$

$$\begin{aligned} \text{故 } f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)}) &= f(1 + 4\lambda, 1 - 2\lambda) = 40\lambda^2 - 20\lambda - 3 \\ \frac{df}{d\lambda} &= 80\lambda - 20 \end{aligned}$$

令 $\frac{df}{d\lambda} = 0$, 得

$$\lambda_0 = 0.25$$

$$\text{因 } \mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{s}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.25 \begin{pmatrix} 4 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}$$

$$f(\mathbf{x}^{(1)}) = -5.5, \quad \mathbf{g}_1 = \nabla f(\mathbf{x}^{(1)}) = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$$

$$\mu_1 = \|\mathbf{g}_1\|^2 / \|\mathbf{g}_0\|^2 = \frac{1}{4}$$

$$\mathbf{s}^{(1)} = -\mathbf{g}_1 + \mu_1 \mathbf{s}^{(0)} = -\begin{pmatrix} -1 \\ -2 \end{pmatrix} + \frac{1}{4} \begin{pmatrix} 4 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1.5 \end{pmatrix}$$

又因
$$\mathbf{x}^{(1)} + \lambda \mathbf{s}^{(1)} = \begin{pmatrix} 2 + 2\lambda \\ 0.5 + 1.5\lambda \end{pmatrix}$$

故
$$\begin{aligned} f(\mathbf{x}^{(1)} + \lambda \mathbf{s}^{(1)}) &= f(2 + 2\lambda, \frac{1}{2}(1 + 3\lambda)) \\ &= \frac{5}{2}\lambda^2 - 5\lambda - \frac{11}{2} \end{aligned}$$

$$\frac{df}{d\lambda} = 5\lambda - 5, \quad \text{令 } \frac{df}{d\lambda} = 0, \text{ 得 } \lambda_1 = 1$$

于是
$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{s}^{(1)} = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix} + 1 \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$f(\mathbf{x}^{(2)}) = -8, \quad \nabla f(\mathbf{x}^{(2)}) = (0, 0)^T, \quad \text{即 } \|\mathbf{g}_2\| = 0$$

因此, $\mathbf{x}^{(2)} = (4, 2)^T$ 为 $f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$ 的极小点。 ■

【例 4.2】 求 $f(x_1, x_2) = x_1^3 + x_2^3 - 3x_1x_2$ 的极小点。

解
$$\frac{\partial f}{\partial x_1} = 3x_1^2 - 3x_2, \quad \frac{\partial f}{\partial x_2} = 3x_2^2 - 3x_1$$

取初始点 $\mathbf{x}^{(0)} = (2, 2)^T$, 从而有

$$\nabla f(\mathbf{x}^{(0)}) = \mathbf{g}_0 = (6, 6)^T$$

$$\mathbf{s}^{(0)} = -\mathbf{g}_0 = \begin{pmatrix} -6 \\ -6 \end{pmatrix}, \quad \mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)} = \begin{pmatrix} 2 - 6\lambda \\ 2 - 6\lambda \end{pmatrix}$$

所以
$$f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)}) = 4(1 - 3\lambda)^2(1 - 12\lambda)$$

$$\frac{df}{d\lambda} = -72(1 - 3\lambda)(1 - 6\lambda)$$

$$\text{令 } \frac{df}{d\lambda} = 0, \text{ 得 } \lambda = \frac{1}{3} \text{ 或 } \frac{1}{6}, \text{ 易见当 } \lambda_0 = \frac{1}{6} \text{ 时, } f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)})$$

取得极小值。而

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{s}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \nabla f(\mathbf{x}^{(1)}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

又因
$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1, \quad \frac{\partial^2 f}{\partial x_2^2} = 6x_2, \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = -3$$

故 $\nabla^2 f(\mathbf{x}) = \begin{pmatrix} 6x_1 & -3 \\ -3 & 6x_2 \end{pmatrix}$, $\nabla^2 f(1,1) = \begin{pmatrix} 6 & -3 \\ -3 & 6 \end{pmatrix}$ 是正定的。因此 $\mathbf{x}^{(1)} = (1,1)^T$ 为 $f(x_1, x_2)$ 的严格极小点。

当 $\lambda_0 = \frac{1}{3}$ 时, $\mathbf{x}^{(1)} = (0, 0)^T$ 不是 $f(x_1, x_2)$ 的极小点。

3.5 变尺度法 (Variable Metric Method)

变尺度法是由 W. C. Davidon 于 1959 年首先提出, 1963 年为 R. Fletcher 和 M. J. D. Powell 所发展, 形成为著名的 DFP 变尺度算法, 目前变尺度算法已发展成为一大类算法, 被认为是求解无约束优化问题最有效的算法之一, 得到了广泛的研究和应用。

3.5.1 基本思想

大家知道: 最速下降法计算方法简便, 但收敛速度太慢; 牛顿法收敛速度快, 但计算量太大, 实现困难。能否设计一种算法, 它既具有快速收敛的优点, 又能不计算二阶偏导数矩阵及其逆阵, 就可以构造出每次迭代的搜索方向 $s^{(k)}$ 呢? 分析最速下降法和阻尼牛顿法的计算公式, 发现它们可以用

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \mathbf{H}_k \nabla f(\mathbf{x}^{(k)}) \quad (5-1)$$

来统一描述。

若 $\mathbf{H}_k = \mathbf{I}$ (\mathbf{I} 为单位矩阵), 则得最速下降法的计算公式。

若 $\mathbf{H}_k = [\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$, 则得阻尼牛顿法的计算公式。特别地, 若步长 $\lambda_k = 1$, 则得到牛顿法的迭代公式。

为了保持牛顿法的优点, 我们希望 \mathbf{H}_k 能近似地等于 $[\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$, 并且希望 \mathbf{H}_k 能在迭代计算中逐次生成, 即

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{C}_k \quad (5-2)$$

\mathbf{C}_k 称为修正矩阵, \mathbf{C}_k 不同, 就可以得到不同的算法。

为了研究使 $H_k \approx [\nabla^2 f(\mathbf{x}^{(k)})]^{-1}$ 的条件, 我们来考察 n 元二次函数

$$f(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \mathbf{x}^T \mathbf{G} \mathbf{x} / 2$$

其中 $\mathbf{b}, \mathbf{x} \in \mathbf{R}^n$, \mathbf{G} 为 n 阶对称正定矩阵, a 为常数。

$$\nabla f(\mathbf{x}) = \mathbf{b} + \mathbf{G} \mathbf{x}$$

令 $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$, $\mathbf{g}_k = \mathbf{g}(\mathbf{x}^{(k)})$, 则 $\mathbf{g}_k = \mathbf{b} + \mathbf{G} \mathbf{x}^{(k)}$, $\mathbf{g}_{k+1} = \mathbf{b} + \mathbf{G} \mathbf{x}^{(k+1)}$ 。所以

$$\mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{G}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

令

$$\begin{cases} \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \Delta \mathbf{x}_k \\ \mathbf{g}_{k+1} - \mathbf{g}_k = \Delta \mathbf{g}_k \end{cases} \quad (5-3)$$

则上式可写成

$$\mathbf{G} \Delta \mathbf{x}_k = \Delta \mathbf{g}_k \quad (5-4)$$

或

$$\Delta \mathbf{x}_k = \mathbf{G}^{-1} \Delta \mathbf{g}_k \quad (5-5)$$

对于一般的 n 元非二次函数 $f(\mathbf{x}) \in C^2$, 可以将 $f(\mathbf{x})$ 在点 $\mathbf{x}^{(k+1)}$ 进行 Taylor 展开, 取其前三项, 则可得

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k+1)}) + \nabla f(\mathbf{x}^{(k+1)})^T (\mathbf{x} - \mathbf{x}^{(k+1)}) + (\mathbf{x} - \mathbf{x}^{(k+1)})^T \mathbf{G}_{k+1} (\mathbf{x} - \mathbf{x}^{(k+1)}) / 2$$

其中 $\mathbf{G}_{k+1} = \nabla^2 f(\mathbf{x}^{(k+1)})$ 。

$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{(k+1)}) + \mathbf{G}_{k+1} (\mathbf{x} - \mathbf{x}^{(k+1)})$$

或

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}_{k+1} + \mathbf{G}_{k+1} (\mathbf{x} - \mathbf{x}^{(k+1)})$$

所以

$$\mathbf{g}_{k+1} - \mathbf{g}_k \approx \mathbf{G}_{k+1} \Delta \mathbf{x}_k$$

即

$$\mathbf{G}_{k+1} \Delta \mathbf{x}_k \approx \Delta \mathbf{g}_k \quad (5-4')$$

或

$$\Delta \mathbf{x}_k \approx \mathbf{G}_{k+1}^{-1} \Delta \mathbf{g}_k \quad (5-5')$$

称(5-4)和(5-5)为拟牛顿方程。为了使 $H_k \approx \mathbf{G}_k^{-1}$, 应要求 H_{k+1} 满足拟牛顿方程(5-4)或(5-5), 即

$$\mathbf{H}_{k+1} \Delta \mathbf{g}_k = \Delta \mathbf{x}_k \quad (5-6)$$

或

$$B_{k+1}\Delta x_k = \Delta g_k \quad (5-7)$$

其中 $B_{k+1} = H_{k+1}^{-1}$ 。

3.5.2 对称秩 1 算法

为了使迭代计算简单易行,修正矩阵 C_k 应选取尽可能简单的形式。而简单又可以从不同的角度来理解,通常是要求 C_k 的秩越小越好。

若要求 C_k 是秩为 1 的对称矩阵,则可设

$$C_k = \alpha_k u u^T \quad (5-8)$$

其中 $\alpha_k \neq 0$ 为待定常数, $u = (u_1, u_2, \dots, u_n)^T \neq \mathbf{0}$ 。

将(5-8)式代入(5-2)式得

$$H_{k+1} = H_k + \alpha_k u u^T$$

将上式代入(5-6)得

$$H_k \Delta g_k + \alpha_k u (u^T \Delta g_k) = \Delta x_k$$

由于 $\alpha_k, u^T \Delta g_k$ 为数量,所以 u 与 $\Delta x_k - H_k \Delta g_k$ 成正比,不妨取

$$u = \Delta x_k - H_k \Delta g_k$$

故

$$\alpha_k = \frac{1}{u^T \Delta g_k} = \frac{1}{\Delta g_k^T (\Delta x_k - H_k \Delta g_k)}$$

其中

$$\Delta g_k^T (\Delta x_k - H_k \Delta g_k) \neq 0$$

$$H_{k+1} = H_k + \frac{(\Delta x_k - H_k \Delta g_k)(\Delta x_k - H_k \Delta g_k)^T}{\Delta g_k^T (\Delta x_k - H_k \Delta g_k)} \quad (5-9)$$

(5-9) 式称为对称秩 1 公式。由对称秩 1 公式确定的变尺度算法称为对称秩 1 变尺度算法。此算法是简单的,但也存在明显的缺点:

(1) 当 H_k 正定时,由(5-9)式确定的 H_{k+1} 不一定是正定的,因此不能保证 $s^{(k)} = -H_k g_k$ 是下降方向。(2) (5-9)式的分母可能为零或近似为零,前者将使算法失效,后者将引起计算不稳定。

3.5.3 DFP 算法

DFP 算法是最先被研究的一种变尺度算法,也是目前仍在广

泛使用的一种算法,它是一种秩 2 对称算法,一般的秩 2 对称算法的修正矩阵公式可以写成

$$C_k = \alpha_k \mathbf{u} \mathbf{u}^T + \beta_k \mathbf{v} \mathbf{v}^T \quad (5-10)$$

其中 \mathbf{u}, \mathbf{v} 为待定的 n 维向量, α_k, β_k 为待定常数,将上式代入(5-2)式,再代入(5-6)式可得

$$\alpha_k \mathbf{u} (\mathbf{u}^T \Delta \mathbf{g}_k) + \beta_k \mathbf{v} (\mathbf{v}^T \Delta \mathbf{g}_k) = \Delta \mathbf{x}_k - \mathbf{H} \Delta \mathbf{g}_k$$

满足上式的 $\alpha_k, \beta_k, \mathbf{u}, \mathbf{v}$ 有无数多种取法,比较简单的一种取法是

$$\mathbf{u} = \mathbf{H}_k \Delta \mathbf{g}_k, \quad \mathbf{v} = \Delta \mathbf{x}_k$$

$$\alpha_k = -\frac{1}{\mathbf{u}^T \Delta \mathbf{g}_k} = -\frac{1}{\Delta \mathbf{g}_k^T \mathbf{H}_k \Delta \mathbf{g}_k}, \quad \beta_k = \frac{1}{\mathbf{v}^T \Delta \mathbf{g}_k} = \frac{1}{\Delta \mathbf{x}_k^T \Delta \mathbf{g}_k}$$

$$\therefore \quad \mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\Delta \mathbf{x}_k \Delta \mathbf{x}_k^T}{\Delta \mathbf{x}_k^T \Delta \mathbf{g}_k} - \frac{\mathbf{H}_k \Delta \mathbf{g}_k (\mathbf{H}_k \Delta \mathbf{g}_k)^T}{\Delta \mathbf{g}_k^T \mathbf{H}_k \Delta \mathbf{g}_k} \quad (5-11)$$

这就是 DFP 变尺度算法的计算公式。

1. 变尺度算法的计算步骤

(1) 给定初始 $\mathbf{x}^{(0)}$ 、计算精度 $\varepsilon > 0$ 和初始矩阵 $\mathbf{H}_0 = \mathbf{I}$ (单位阵), 令 $k = 0$;

(2) 计算 $\mathbf{s}^{(k)} = -\mathbf{H}_k \mathbf{g}_k$, 沿 $\mathbf{s}^{(k)}$ 进行精确一维搜索, 求出步长 λ_k , 使

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}) = \min_{\lambda \geq 0} f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$$

令

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$$

(3) 若 $\|\mathbf{g}_{k+1}\| < \varepsilon$, 则取 $\mathbf{x}^* = \mathbf{x}^{(k+1)}$, 计算结束; 否则由公式(5-11)计算 \mathbf{H}_{k+1} , 若 $k \neq n-1$, 则令 $k = k+1$, 返回(2); 若 $k = n-1$, 令 $\mathbf{x}^{(0)} = \mathbf{x}^{(k+1)}$, $k = 0$, 返回(2)。

DFP 法的计算程序见文献 46。

2. 算法应用举例

【例 5.1】 求函数 $f(\mathbf{x}) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$ 的极小点。

解 1° 取初始点 $\mathbf{x}^{(0)} = (1, 1)^T$, 则 $f(\mathbf{x}^{(0)}) = -3$ 。而

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4 - 2x_2, \quad \frac{\partial f}{\partial x_2} = 4x_2 - 2x_1$$

令 $k = 0$;

$$2^\circ \quad \mathbf{s}^{(0)} = -\mathbf{H}_0 \nabla f(\mathbf{x}^{(0)}) = -\mathbf{g}_0 = (4, -2)^T$$

$$\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)} + (1 + 4\lambda, 1 - 2\lambda)^T$$

所以

$$f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(0)}) = f(1 + 4\lambda, 1 - 2\lambda)$$

$$= 40\lambda^2 - 20\lambda - 3$$

因为 $\frac{df}{d\lambda} = 80\lambda - 20$, 令 $\frac{df}{d\lambda} = 0$, 得 $\lambda_0 = 0.25$, 所以

$$\mathbf{x}^{(1)} = (2, 0.5)^T, \quad \mathbf{g}_1 = (-1, -2)^T, \quad f_1 = -5.5$$

3° 因为 $\Delta \mathbf{x} = \mathbf{x}^{(1)} - \mathbf{x}^{(0)} = (1, -0.5)^T$, $\Delta \mathbf{g}_0 = (3, -4)^T$,

所以

$$\mathbf{H}_1 = \mathbf{H}_0 + \frac{\Delta \mathbf{x}_0 (\Delta \mathbf{x}_0)^T}{(\Delta \mathbf{x}_0)^T \Delta \mathbf{g}_0} - \frac{(\mathbf{H}_0 \Delta \mathbf{g}_0) (\mathbf{H}_0 \Delta \mathbf{g}_0)^T}{\Delta \mathbf{g}_0^T \mathbf{H}_0 \Delta \mathbf{g}_0}$$

$$= \begin{pmatrix} 21/25 & 19/50 \\ 19/50 & 41/100 \end{pmatrix}$$

这时 $k = 1$, 返回(2), 继续进行迭代。因为

$$\mathbf{s}^{(1)} = -\mathbf{H}_1 \mathbf{g}_1 = (8/5, 6/5)^T$$

$$\mathbf{x}^{(1)} + \lambda \mathbf{s}^{(1)} = (2 + 8\lambda/5, 0.5 + 6\lambda/5)^T$$

所以 $f(\mathbf{x}^{(1)} + \lambda \mathbf{s}^{(1)}) = 8\lambda^2/5 - 4\lambda - 5.5$

令 $\frac{df}{d\lambda} = 16\lambda/5 - 4 = 0$, 得步长 $\lambda_1 = 5/4$, 所以 $\mathbf{x}^{(2)} = (4, 2)^T$, 这时 $\nabla f(\mathbf{x}^{(2)}) = (0, 0)^T$, 因此 $\mathbf{x}^{(2)}$ 即为 $f(\mathbf{x})$ 的极小点。

3. DFP 算法的性质和优缺点

首先我们指出: 若 \mathbf{H}_k 为对称正定矩阵, 且 $\mathbf{g}_k = \nabla f(\mathbf{x}^{(k)}) \neq 0$, 则 DFP 公式(5-11)后两项的分母均不为零, 因此 DFP 公式(5-11)总是有意义的。证明留做习题。

DFP 算法的优点是: (1) 若目标函数 $f(\mathbf{x})$ 为 n 元二次严格凸

函数,当初始矩阵取 $H_0 = I$ (单位阵) 时,由 DFP 法产生的搜索方向 $s^{(0)}, s^{(1)}, \dots, s^{(k)}$ 是关于 $G = \nabla^2 f(x)$ 的共轭方向,因此,最多经过 n 步迭代,就能达到最小点。即算法具有二次收敛性。(2) 如果 $f(x) \in C^1$ 为严格凸函数,则 DFP 算法是全局收敛的。(3) 若 H_k 为对称正定矩阵,且 $g_k \neq 0$, 则由公式(5-11) 确定的 H_{k+1} 也是对称正定的。关于上述三点的详述和论证可参看文献 24 ~ 26, 61。

DFP 算法的缺点是:(1) 需要的存贮量较大,大约需要 $O(n^2)$ 个存贮单元,因此对于大型问题,不如使用共轭梯度法方便。(2) 数值计算的稳定性不如后面将要介绍的 BFGS 算法。(3) 在使用不精确一维搜索时,它的计算效果不如 BFGS 算法的效果好。

3.5.4 吴桂变尺度算法类

1970 年, Huang 在文献 62 中提出修正矩阵 C_k 实质上仅包含三个独立参数的一类变尺度算法,后来被称为 Huang 类算法。1981 年,吴方,桂湘云在文献 63 中,用三个简单的条件刻划了 Huang 类算法中的修正矩阵,并提出了一个比 Huang 的算法类更广的,含有 $n+1$ 个独立参数的变尺度算法类,同时证明了这个算法类具有 Huang 类算法的一切性质。这是一个很广的变尺度算法类,我们称为吴桂变尺度算法类,简称吴桂算法。下面作简要介绍。

定义 5.1 设 A, C 为两个有相同行数的矩阵或向量,用 $C|A$ 表示有矩阵(包括向量) D ,使得 $A = CD$;特别,若用 (a, b) 表示由列向量 a, b 组成的二列矩阵,则 $(a, b)|A$ 表示有列向量 c, d , 使 $A = ac^T + bd^T$ 。

在吴桂算法中,不要求 H_k 是对称的。并引入一个数量参数 ρ_k , 将拟牛顿方程推广为更一般的形式

$$H_{k+1}\Delta g_k = \rho_k \Delta x_k \quad (5-12)$$

设 $f(x) \in C^1$, 给定初始点 $x^{(0)}$ 、数列 $\{\rho_k\}$ 和计算精度 $\epsilon > 0$, 则吴桂算法的计算步骤如下:

(1) 计算 $\mathbf{g}_0 = \nabla f(\mathbf{x}^{(0)})$, 若 $\|\mathbf{g}_0\| < \varepsilon$, 则 $\mathbf{x}^* = \mathbf{x}^{(0)}$, 计算结束, 否则取一个满足 $\mathbf{g}_0^T \mathbf{H}_0 \mathbf{g}_0 \neq 0$ 的初始矩阵 \mathbf{H}_0 , 例如取 $\mathbf{H}_0 = \mathbf{I}$, 令 $k = 0$;

(2) 以 $\mathbf{x}^{(k)}$ 为起点, 沿方向 $\mathbf{s}^{(k)} = -\mathbf{H}_k^T \mathbf{g}_k$ 进行精确一维搜索, 决定步长 λ_k , 令

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}^{(k)}$$

(3) 若 $\|\mathbf{g}_{k+1}\| < \varepsilon$, 则 $\mathbf{x}^* = \mathbf{x}^{(k+1)}$, 计算结束, 否则转第(4)步;

(4) 若 $k < n$, 令 $\Delta \mathbf{x}_k = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, $\Delta \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, 取 \mathbf{H}_{k+1} 为满足(5-12)式及条件

$$\mathbf{g}_{k+1}^T \mathbf{H}_{k+1} \mathbf{g}_{k+1} \neq 0 \quad (5-13)$$

和

$$(\mathbf{H}_k^T \Delta \mathbf{g}_k, \Delta \mathbf{x}_k) | (\mathbf{H}_{k+1} - \mathbf{H}_k)^T \quad (5-14)$$

的方阵, 令 $k = k + 1$, 返回第(2)步, 否则当 $k = n$ 时, 令 $\mathbf{x}^{(0)} = \mathbf{x}^{(n)}$, 返回第(1)步。

由(5-14)式知: 有列向量 $\mathbf{a}_k, \mathbf{b}_k$, 使

$$(\mathbf{H}_{k+1} - \mathbf{H}_k)^T = \mathbf{H}_k^T \Delta \mathbf{g}_k \mathbf{a}_k^T + \Delta \mathbf{x}_k \mathbf{b}_k^T$$

或

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{a}_k \Delta \mathbf{g}_k^T \mathbf{H}_k + \mathbf{b}_k \Delta \mathbf{x}_k^T \quad (5-15)$$

将(5-15)式代入(5-12)式得

$$\mathbf{a}_k (\Delta \mathbf{g}_k^T \mathbf{H}_k \Delta \mathbf{g}_k) + \mathbf{b}_k (\Delta \mathbf{x}_k^T \Delta \mathbf{g}_k) = \rho_k \Delta \mathbf{x}_k - \mathbf{H}_k \Delta \mathbf{g}_k \quad (5-16)$$

因为 $\mathbf{g}_{k+1}^T \Delta \mathbf{x}_k = \mathbf{g}_{k+1}^T (\lambda_k \mathbf{s}^{(k)}) = \lambda_k \mathbf{g}_{k+1}^T \mathbf{s}^{(k)} = 0$

$$\Delta \mathbf{g}_k^T \Delta \mathbf{x}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^T \Delta \mathbf{x}_k = -\mathbf{g}_k^T \Delta \mathbf{x}_k = -\mathbf{g}_k^T (\lambda_k \mathbf{s}^{(k)})$$

$$= \lambda_k \mathbf{g}_k^T \mathbf{H}_k^T \mathbf{g}_k = \lambda_k \mathbf{g}_k^T \mathbf{H}_k \mathbf{g}_k \neq 0$$

所以(5-16)式中的 \mathbf{b}_k 可由 \mathbf{a}_k, ρ_k 惟一确定, 在 \mathbf{H}_{k+1} 的计算公式(5-15)中, 实质上仅含有 $n + 1$ 个独立参数(n 维列向量 \mathbf{a}_k , 数量 ρ_k)。

在文献 63 中,还证明了该类算法的一些重要性质,我们简介如下。

1° 对于二次凸函数,吴桂算法中的任一算法都是共轭方向算法。因此至多迭代 n 次,就一定能达到二次凸函数的最优点。

2° 设 $f(x) \in C^1$, 对于给定的初始点 $x^{(0)}$ 和初始方阵 H_0 , 吴桂算法类中与同一组 $\{\rho_k\}$ 相应的任意两个不同算法产生相同的点列 $\{x^{(k)}\}$; 也就是说其他各组参数在理论上对点列不产生影响。关于上述性质的严格论述见文献 63。

3.5.5 Huang 类算法

在上述的吴桂算法类中,若 H_{k+1} 除满足条件(5-13)和(5-14)以外,还要求满足条件

$$(H_k \Delta g_k, \Delta x_k) | (H_{k+1} - H_k) \quad (5-17)$$

则由文献 63 中的引理 2.2 可知:有纯量 $\alpha_k, \beta_k, \gamma_k$ 和 δ_k , 使

$$H_{k+1} = H_k - \frac{H_k \Delta g_k (\alpha_k H_k^T \Delta g_k + \beta_k \Delta x_k)^T}{(\alpha_k H_k^T \Delta g_k + \beta_k \Delta x_k)^T \Delta g_k} + \rho_k \frac{\Delta x_k (\gamma_k H_k^T \Delta g_k + \delta_k \Delta x_k)^T}{(\gamma_k H_k^T \Delta g_k + \delta_k \Delta x_k)^T \Delta g_k} \quad (5-18)$$

这就是 Huang 变尺度算法类的迭代公式。这里的 H_{k+1} 表面上含有五个参数 $\alpha_k, \beta_k, \gamma_k, \delta_k$ 和 ρ_k , 但实质上只含有三个独立的参数 $\alpha_k/\beta_k, \gamma_k/\delta_k$ 和 ρ_k , Huang 类算法的矩阵 H_{k+1} 是由条件(5-13), (5-14)和(5-17)确定的,所以它是吴桂算法类的一个子类。

3.5.6 自调节(Self-Scaling) 变尺度算法类

若把条件(5-17)进一步加强为

$$H_k = H_k^T \text{ 和 } H_{k+1} = H_{k+1}^T \quad (5-19)$$

易见条件(5-17)可由式(5-14)和(5-19)推出,则由文献 63 的引理 2.5 可知:在假设 $\Delta g_k^T H_k \Delta g_k \neq 0$ 之下,有纯量 α_k , 使

$$H_{k+1} = H_k - \frac{H_k \Delta g_k \Delta g_k^T H_k}{\Delta g_k^T H_k \Delta g_k} + \rho_k \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} + \alpha_k (\Delta g_k^T H_k \Delta g_k) v_k v_k^T \quad (5-20)$$

其中

$$v_k = \frac{\Delta x_k}{\Delta x_k^T \Delta g_k} - \frac{H_k \Delta g_k}{\Delta g_k^T H_k \Delta g_k} \quad (5-21)$$

这就是 Oren—Luenberger 于 1974 年提出的自调节变尺度算法类,它含有两个参数 α_k 和 ρ_k 。

若 $\rho_k = 1$,就得到 Fletcher—Broyden 算法类

$$H_{k+1} = H_k - \frac{H_k \Delta g_k \Delta g_k^T H_k}{\Delta g_k^T H_k \Delta g_k} + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} + \alpha_k (\Delta g_k^T H_k \Delta g_k) v_k v_k^T \quad (5-22)$$

它包含一个参数 α_k 。当 $\alpha_k = 0$ 时,就得到前面讲过的 DFP 变尺度算法的计算公式。

3.5.7 BFGS 变尺度算法

在式(5-22)中,若令 $\alpha_k = 1$,则得到熟知的 BFGS 变尺度算法的计算公式

$$H_{k+1} = H_k - \frac{H_k \Delta g_k \Delta g_k^T H_k}{\Delta g_k^T H_k \Delta g_k} + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} + (\Delta g_k^T H_k \Delta g_k) v_k v_k^T \quad (5-23)$$

或写成

$$H_{k+1} = H_k + \frac{\mu_k \Delta x_k \Delta x_k^T - H_k \Delta g_k \Delta x_k^T - \Delta x_k \Delta g_k^T H_k}{\Delta x_k^T \Delta g_k} \quad (5-23')$$

其中

$$\mu_k = 1 + \Delta g_k^T H_k \Delta g_k / \Delta x_k^T \Delta g_k \quad (5-24)$$

如何由式(5-23)和(5-21)得到式(5-23')和(5-24),留给读者作为

练习。

BFGS 算法是由 Broyden, Fletcher, Goldfarb 和 Shanno 等人在 1970 年给出的。它与 DFP 算法同属于 Broyden 算法类,因此,它与 DFP 算法具有类似的性质,也具有前面讲过的 DFP 算法的那些优点,但它的数值稳定性要比 DFP 算法好,而且在使用不精确一维搜索时,也能证明它是超线性收敛的。因此 BFGS 算法被认为是目前最好的一种算法,得到了极为广泛的应用。其计算程序可参看文献 64。

最后我们指出:在(5-22)式中,若令

$$\alpha_k = \frac{\Delta \mathbf{x}_k^T \Delta \mathbf{g}_k}{\Delta \mathbf{x}_k^T \Delta \mathbf{g}_k + \Delta \mathbf{g}_k^T \mathbf{H}_k \Delta \mathbf{g}_k} \quad (5-25)$$

就得到 Hoshino 变尺度算法的修正公式。若令

$$\alpha_k = \frac{\mathbf{g}_{k+1}^T \mathbf{H}_k \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{H}_k \mathbf{g}_k} \quad (5-26)$$

则又可得到一个新的变尺度算法的修正公式。使用上述修正公式(5-25)和(5-26)的变尺度算法,其计算效果均比较好,可与 BFGS 算法比美。

3.6 直接搜索法

上面所讲的几种方法,都要利用目标函数的一阶或二阶偏导数,但在实际问题中,所遇到的目标函数往往比较复杂,有的甚至难于写出其明显的解析表达式,因此,它们的导数很难求得,甚至根本无法求得。这时,就必须采用不用导数的方法,即求多变量函数极值的直接搜索法,这类方法的特点是方法简单,适用范围较广,但由于没有利用函数的分析性质,其收敛速度一般是比较慢的。本节介绍三种比较常用的直接搜索法。

3.6.1 Hooke-Jeeves 方法

这是一种简单而且容易实现的实用算法,由 Hooke 和 Jeeves 在 1961 年提出。这种方法由两类“移动”构成:一类称为探测搜索,其目的是探求下降的有利方向;另一类称为模式搜索,其目的是沿着有利方向进行加速。所以,也称为步长加速法或模式搜索法。

设要求解的问题为

$$\min f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

1. 计算步骤

设初始点和初始步长分别为 $\mathbf{x}^{(1)}$ 和 d, e_1, e_2, \dots, e_n 为坐标向量,加速因子和计算精度分别为 $\alpha > 0$ 和 $\varepsilon > 0$ 。则 Hooke-Jeeves 方法的计算步骤如下:

1° 令 $\mathbf{y}^{(1)} = \mathbf{x}^{(1)}$, $k = j = 1$;

2° 若 $f(\mathbf{y}^{(j)} + d\mathbf{e}_j) < f(\mathbf{y}^{(j)})$, 则称为试验成功, 令 $\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} + d\mathbf{e}_j$, 转 3°; 否则, 若 $f(\mathbf{y}^{(j)} + d\mathbf{e}_j) \geq f(\mathbf{y}^{(j)})$, 称为试验失败, 此时, 若 $f(\mathbf{y}^{(j)} - d\mathbf{e}_j) < f(\mathbf{y}^{(j)})$, 令 $\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} - d\mathbf{e}_j$, 转 3°; 若 $f(\mathbf{y}^{(j)} - d\mathbf{e}_j) \geq f(\mathbf{y}^{(j)})$, 令 $\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)}$, 转 3°;

3° 若 $j < n$, 令 $j = j + 1$, 返回 2°; 否则 $j = n$, 若 $f(\mathbf{y}^{(n+1)}) \geq f(\mathbf{x}^{(k)})$, 转 5°; 若 $f(\mathbf{y}^{(n+1)}) < f(\mathbf{x}^{(k)})$, 则转 4°;

4° 令 $\mathbf{x}^{(k+1)} = \mathbf{y}^{(n+1)}$, $\mathbf{y}^{(1)} = \mathbf{x}^{(k+1)} + \alpha(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$, $k = k + 1$, 再令 $j = 1$, 返回 2°;

5° 若 $d \leq \varepsilon$, 则计算结束, 取 $\mathbf{x}^* \approx \mathbf{x}^{(k)}$; 否则, 令 $d = d/2$, $\mathbf{y}^{(1)} = \mathbf{x}^{(k)}$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$, $k = k + 1$, 再令 $j = 1$, 返回 2°。

上述的 Hooke-Jeeves 方法是不用线搜索的, 使用线搜索的 Hooke-Jeeves 方法, 可参看文献 11。

在上述方法中, 容易看出: 第 1 步是准备步骤, 第 2, 3 步是一种探测搜索, 探求下降的有利方向; 第 4 步是沿着找到的有利方向 $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ 加速前进, 第 5 步判断是否可以结束。

2. Hooke-Jeeves 方法的程序框图

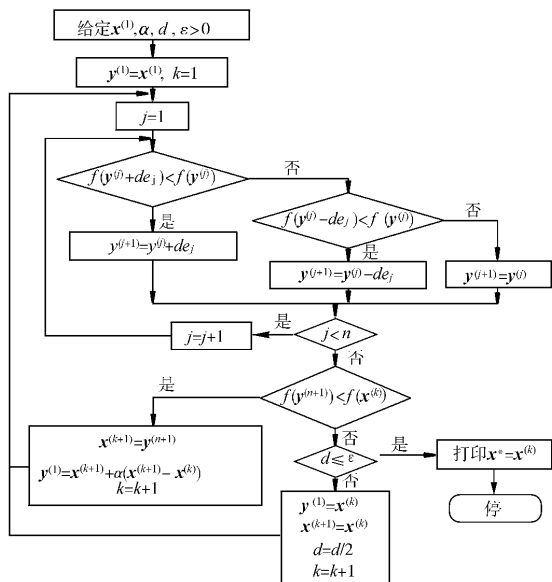


图 3-6

3. 计算例子

【例 6.1】 求函数 $f(x) = x_1^2 - 2x_1x_2 + 2x_2^2 - 4x_1$ 的极小点。

解 取 $x^{(1)} = (1, 1)^T$ 为初始点, $d = 1$, $\varepsilon = 0.1$, $\alpha = 1$.

第一次迭代

1° 令 $y^{(1)} = x^{(1)} = (1, 1)^T$, $k = j = 1$.

2° 因为 $f(y^{(1)} + de_1) = f(2, 1) = -6 < f(y^{(1)}) = -3$, 所以 $y^{(2)} = (2, 1)^T$.

3° 由于 $j = 1 < n = 2$, 所以再回到第 2 步进行计算:

因 $f(\mathbf{y}^{(2)} + d\mathbf{e}_2) = f(2, 2) = -4 > -6 = f(\mathbf{y}^{(2)})$

所以试验失败。又因为

$$f(\mathbf{y}^{(2)} - d\mathbf{e}_2) = f(2, 0) = -4 > -6$$

所以 $\mathbf{y}^{(3)} = \mathbf{y}^{(2)} = (2, 1)^T$ 。这时, $j = 2 = n$, 且 $f(\mathbf{y}^{(3)}) = -6 < f(\mathbf{x}^{(1)}) = f(1, 1) = -3$ 。

4° 令 $\mathbf{x}^{(2)} = \mathbf{y}^{(3)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $\mathbf{y}^{(1)} = \mathbf{x}^{(2)} + \alpha(\mathbf{x}^{(2)} - \mathbf{x}^{(1)}) = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ 。
 $f(\mathbf{y}^{(1)}) = f(3, 1) = -7$ 。第一次迭代完成, 这时 $k = 2$, 再令 $j = 1$, 回到第 2 步再进行计算。

第二次迭代

因 $f(\mathbf{y}^{(1)} + d\mathbf{e}_1) = f(4, 1) = -6 > f(\mathbf{y}^{(1)}) = -7$

$$f(\mathbf{y}^{(1)} - d\mathbf{e}_1) = f(2, 1) = -6 > f(\mathbf{y}^{(1)}) = -7$$

故 $\mathbf{y}^{(2)} = \mathbf{y}^{(1)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$

又因为 $j = 1 < 2$, 再返回第 2 步。

因 $f(\mathbf{y}^{(2)} + d\mathbf{e}_2) = f(3, 2) = -7 = f(\mathbf{y}^{(2)})$

$$f(\mathbf{y}^{(2)} - d\mathbf{e}_2) = f(3, 0) = -3 > -7 = f(\mathbf{y}^{(2)})$$

故 $\mathbf{y}^{(3)} = \mathbf{y}^{(2)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$

这时 $j = 2 = n$, 且

$$f(\mathbf{y}^{(3)}) = f(3, 1) = -7 < f(\mathbf{x}^{(2)}) = f(2, 1) = -6$$

于是令

$$\mathbf{x}^{(3)} = \mathbf{y}^{(3)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \mathbf{y}^{(1)} = \mathbf{x}^{(3)} + \alpha(\mathbf{x}^{(3)} - \mathbf{x}^{(2)}) = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

这时 $k = 3$, 再令 $j = 1$, 回到第 2 步重新进行计算。

第三次迭代

因 $f(\mathbf{y}^{(1)} + d\mathbf{e}_1) = f(5, 1) = -3 > f(\mathbf{y}^{(1)}) = -6$

$$f(\mathbf{y}^{(1)} - d\mathbf{e}_1) = f(3, 1) = -7 < f(\mathbf{y}^{(1)})$$

所以 $\mathbf{y}^{(2)} = (3, 1)^T$ 。

又因 $j = 1 < 2 = n$, 所以再回到第 2 步进行计算, 由于

$$f(\mathbf{y}^{(2)} + d\mathbf{e}_2) = f(3, 2) = -7 = f(\mathbf{y}^{(2)})$$

$$f(\mathbf{y}^{(2)} - d\mathbf{e}_2) = f(3, 0) = -3 > -7 = f(\mathbf{y}^{(2)})$$

所以 $\mathbf{y}^{(3)} = \mathbf{y}^{(2)} = (3, 1)^T$ 。由于这时 $j = 2 = n$, 且

$$f(\mathbf{y}^{(3)}) = f(3, 1) = f(\mathbf{x}^{(3)})$$

转到第 5 步。

令 $d = d/2$, $\mathbf{y}^{(1)} = \mathbf{x}^{(3)}$, $\mathbf{x}^{(4)} = \mathbf{x}^{(3)}$, 这时 $k = 4$, 再令 $j = 1$, 回到第 2 步进行计算。

第四次迭代

$$\text{因 } f(\mathbf{y}^{(1)} + d\mathbf{e}_1) = f(3.5, 1) = -6.75 < f(\mathbf{y}^{(1)}) = -7$$

$$f(\mathbf{y}^{(1)} - d\mathbf{e}_1) = f(2.5, 1) = -6.75 < f(\mathbf{y}^{(1)}) = -7$$

故 $\mathbf{y}^{(2)} = (3, 1)^T$

又因 $j = 1 < 2 = n$, 回到第 2 步, 由于

$$f(\mathbf{y}^{(2)} + d\mathbf{e}_2) = f(3, 1.5) = -7.5 < f(\mathbf{y}^{(2)})$$

故令 $\mathbf{y}^{(3)} = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$, 由于 $j = 2 = n$, 且 $f(\mathbf{y}^{(3)}) = -7.5 < f(\mathbf{x}^{(4)})$,

所以令 $\mathbf{x}^{(5)} = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$, $\mathbf{y}^{(1)} = \mathbf{x}^{(5)} + \alpha(\mathbf{x}^{(5)} - \mathbf{x}^{(4)}) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, 这时 $k = 5$, $f(\mathbf{y}^{(1)}) = -7$, 再令 $j = 1$, 回到第 2 步进行计算, …… , 直到求得近似于最优解 $\mathbf{x}^* = (4, 2)^T$ 的解为止。

* 3.6.2 单纯形法

无约束极小化的单纯形法与线性规划的单纯形法不同, 它是在 1962 年由 Spendley, Hext 和 Himsworth 首先提出的, 1965 年被 Nelder 和 Mead 所改进。

1. 基本思想与有关概念

单纯形法与前面讲述的直接法不同, 它不是沿某一个方向进

行搜索,而是对 n 维空间的 $n+1$ 个点(它们构成一个单纯形的顶点或极点)上的函数值进行比较,丢掉其中最“坏”的点,代之以新点,构成一个新的单纯形,这样来逐步逼近函数的极小点。为简明起见,我们来考虑问题:

$$\min f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2)^T$$

图 3-7

在平面上取不在同一直线上的三点

p_0, p_1, p_2 , 用 f_0, f_1, f_2 来表示相应点上的函数值,三点中函数值最大的点记为 p_h ,最小的点记为 p_l ,次大的点记为 p_q 。显然,函数值越大,这个点越“坏”,函数值越小,相应的点越“好”。因此“坏”点 p_h 应丢掉,那么新点应如何选取呢?

设 p_c 为线段 $p_l p_q$ 的中点,在直线段 $p_h p_c$ 的延长线上取点 p_r , 使 $p_h p_r = 2 p_h p_c$ 。

称点 p_r 为 p_h 关于 p_c 点的反射点,这种作法称为反射(Reflection)。

算出函数 f 在点 p_r 处的值 f_r , 若

$$f_r \geq f_h \quad (6-1)$$

说明前进太远,需要后退,我们称之为压缩或收缩(Contraction),即在 p_h 和 p_r 之间选

一点 p_s , 若 $f_s < f_h$, 得一新点 p_s , 否则将边 $p_h p_l, p_q p_l$ 缩小一半,得一新单纯形 $\{p_l, p_1, p_2\}$; 若

$$f_r < f_h \quad (6-2)$$

说明前进的还不够远,可以继续前进,我们称之为扩张或延伸(Expansion),即在 $p_h p_r$ 的延长线上取一点 p_e , 若 $f_e \leq f_r$, 取 p_e 为新点,否则取 p_r 为新点(有时要求将条件(6-1)改为更严格的条件 $f_r \leq f_l$, 才进行扩张)。总之,我们可以求得一个新点,从而形成一个新的单纯形 $\{p_q, p_l, p_s\}$, 然后再重复上述过程,直到求得极小点

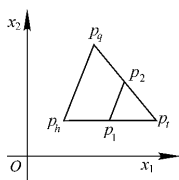
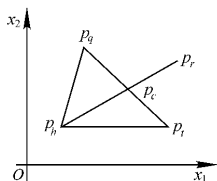


图 3-8

为止。

2. 迭代步骤

对问题 $\min f(x)$, $x \in \mathbf{R}^n$, 在 n 维空间 \mathbf{R}^n 中适当选取 $n+1$ 个点 $x^{(0)}, x^{(1)}, \dots, x^{(n)}$, 构成一个单纯形。通常选取为正规单纯形(即边长相等的单纯形), 一般地可以要求这 $n+1$ 个点使向量组 $x^{(1)} - x^{(0)}, x^{(2)} - x^{(0)}, \dots, x^{(n)} - x^{(0)}$ 线性无关。

1° 计算函数值 $f(x^{(i)})$, $i = 0, 1, \dots, n$, 决定坏点 $x^{(h)}$ 和好点 $x^{(l)}$, 于是

$$f_h = f(x^{(h)}) = \max\{f(x^{(0)}), \dots, f(x^{(n)})\}$$

$$f_l = f(x^{(l)}) = \min\{f(x^{(0)}), \dots, f(x^{(n)})\}$$

2° 算出除点 $x^{(h)}$ 外的 n 个点 $x^{(0)}, \dots, x^{(h+1)}, \dots, x^{(n)}$ 的中心 $x^{(c)}$, 即

$$x^{(c)} = \frac{1}{n} \left(\sum_{i=0}^n x^{(i)} - x^{(h)} \right) \quad (6-3)$$

并求出反射点

$$x^{(r)} = 2x^{(c)} - x^{(h)} \quad (6-4)$$

3° 若 $f_r = f(x^{(r)}) \geq f_h$ (6-5)

则进行压缩, 即令 $x^{(s)} = x^{(h)} + \lambda(x^{(r)} - x^{(h)}) = (1-\lambda)x^{(h)} + \lambda x^{(r)}$, 并求出 $f_s = f(x^{(s)})$, 然后转 5°。这里 $\lambda \in (0, 1)$ 为给定的压缩系数, 可取 $\lambda = 1/4$ 或 $\lambda = 3/4$, 一般要求 $\lambda \neq 0.5$, 若 $f_r < f_h$, 转 4°。

4° 进行扩张, 即令

$$x^{(e)} = x^{(h)} + \mu(x^{(r)} - x^{(h)}) = \mu x^{(r)} + (1-\mu)x^{(h)}$$

其中 $\mu > 1$ 为给定的扩张系数, 可取 $\mu \in [1.2, 2]$ (扩张条件 $f_r < f_h$ 也可换为 $f_r \leq f_l$)。

计算 $f_e = f(x^{(e)})$, 若 $f_e \leq f_r$, 则令 $x^{(s)} = x^{(e)}$, $f_s = f_e$; 否则, 令 $x^{(s)} = x^{(r)}$, $f_s = f_r$ 。

5° 若 $f_s < f_h$, 用 $x^{(s)}$ 替换 $x^{(h)}$, f_s 替换 f_h , 把这样得到的新点 $x^{(s)}$ 和其他 n 个点一起构成一个新的单纯形, 重新确定 $x^{(l)}$, $x^{(h)}$, 然后返回 2°; 若 $f_s \geq f_h$, 转 6°。

$$6^{\circ} \quad \text{若} \quad \frac{f_h - f_l}{|f_l|} < \varepsilon \quad (6-6)$$

或

$$\sum_{i=1}^n [f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(l)})]^2 < \varepsilon \quad (6-7)$$

 成立, 计算结束, 取 $\mathbf{x}^* \approx \mathbf{x}^{(l)}$, $f^* \approx f_l$; 否则缩短边长, 令

$$\mathbf{x}^{(i)} = (\mathbf{x}^{(i)} + \mathbf{x}^{(l)})/2, \quad i = 0, 1, \dots, n \quad (6-8)$$

返回 1°, 继续进行计算。

3. 程序框图

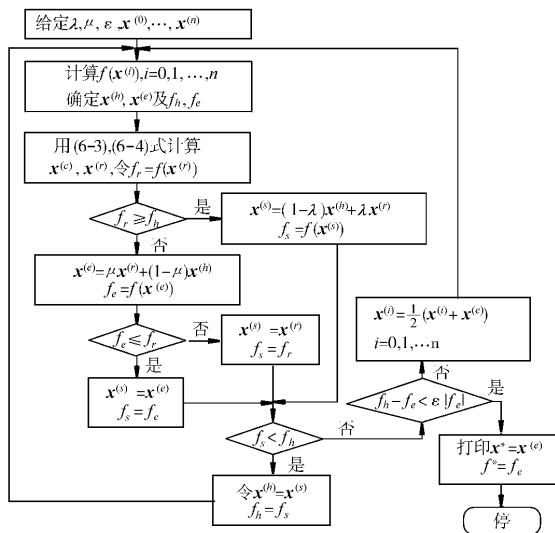


图 3-9

框图说明:

1° $\varepsilon > 0$ 为允许误差, 结束准则可以采用(6-6)式, 如框图3-9所示, 也可采用(6-7)式或其他的结束准则。

2° 初始单纯形的顶点 $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$ 可以直接给定, 也可以自动生成, 例如: 给定初始点 $\mathbf{x}^{(0)}$ 及步长 d 后, 令

$$\mathbf{x}^{(i)} = \mathbf{x}^{(0)} + d\mathbf{e}_i, \quad i = 1, 2, \dots, n \quad (6-9)$$

其中 \mathbf{e}_i 为第 i 个坐标的单位向量; 也可用其他的规则生成 \mathbf{e}_i , 参看文献 23, 29。

4. 算法举例

【例 6.2】 用单纯形法求解问题:

$$\min f(\mathbf{x}) = 4(x_1 - 5)^2 + (x_2 - 6)^2$$

解 取 $\mathbf{x}^{(0)} = (8, 9)^T$, $\mathbf{x}^{(1)} = (10, 9)^T$, $\mathbf{x}^{(2)} = (8, 11)^T$, 扩张系数 $\mu = 2$, 压缩系数 $\lambda = 0.75$ 。因为

$$\begin{aligned} f(\mathbf{x}^{(0)}) &= 45, f(\mathbf{x}^{(1)}) = 109, f(\mathbf{x}^{(2)}) = 61 \\ \text{故 } \mathbf{x}^{(h)} &= \mathbf{x}^{(1)} = (10, 9)^T, \mathbf{x}^{(l)} = \mathbf{x}^{(0)} = (8, 9)^T \\ f_h &= 109, f_l = 45, f_q = 61 \end{aligned}$$

中心点 $\mathbf{x}^{(c)} = (\mathbf{x}^{(0)} + \mathbf{x}^{(2)})/2 = (8, 10)^T$, 而反射点 $\mathbf{x}^{(r)} = 2\mathbf{x}^{(c)} - \mathbf{x}^{(h)} = (6, 11)^T$, 因为

$$f_r = f(\mathbf{x}^{(r)}) = 29 < f_l = 45$$

所以要进行扩张:

$$\begin{aligned} \mathbf{x}^{(e)} &= \mathbf{x}^{(h)} + \mu(\mathbf{x}^{(r)} - \mathbf{x}^{(h)}) = 2\mathbf{x}^{(r)} - \mathbf{x}^{(h)} = (2, 13)^T \\ f_e &= f(\mathbf{x}^{(e)}) = 85 > f_r \end{aligned}$$

因此, 令 $f_s = f_r = 29$, $\mathbf{x}^{(s)} = \mathbf{x}^{(r)} = (6, 11)^T$, 于是得到一个新的单纯形, 其顶点为

$$\mathbf{x}^{(0)} = (8, 9)^T, \mathbf{x}^{(1)} = (6, 11)^T, \mathbf{x}^{(2)} = (8, 11)^T$$

而 $f(\mathbf{x}^{(0)}) = 45$, $f(\mathbf{x}^{(1)}) = 29$, $f(\mathbf{x}^{(2)}) = 61$, 所以

$$\mathbf{x}^{(h)} = \mathbf{x}^{(2)} = (8, 11)^T, \mathbf{x}^{(l)} = \mathbf{x}^{(1)} = (6, 11)^T$$

中心点为 $\mathbf{x}^{(c)} = (\mathbf{x}^{(0)} + \mathbf{x}^{(1)})/2 = (7, 10)^T$, 反射点为

$$\mathbf{x}^{(r)} = 2\mathbf{x}^{(c)} - \mathbf{x}^{(h)} = (6, 9)^T$$

$$f_r = f(\mathbf{x}^{(r)}) = 13 < f_e = 29$$

进行扩张:

$$\mathbf{x}^{(e)} = 2\mathbf{x}^{(r)} - \mathbf{x}^{(h)} = (4, 7)^T, f_e = f(\mathbf{x}^{(e)}) = 5 < f_r$$

所以令 $f_s = f_e = 5$, $\mathbf{x}^{(s)} = \mathbf{x}^{(e)} = (4, 7)^T$, 于是又得到一个新的单纯形, 其顶点为

$$\mathbf{x}^{(0)} = (8, 9)^T, \mathbf{x}^{(1)} = (6, 11)^T, \mathbf{x}^{(2)} = (4, 7)^T$$

而 $f(\mathbf{x}^{(0)}) = 45, f(\mathbf{x}^{(1)}) = 29, f(\mathbf{x}^{(2)}) = 5$

所以 $\mathbf{x}^{(h)} = \mathbf{x}^{(0)} = (8, 9)^T$, $\mathbf{x}^{(l)} = \mathbf{x}^{(2)} = (4, 7)^T$, 中心点为 $\mathbf{x}^{(c)} = (\mathbf{x}^{(1)} + \mathbf{x}^{(2)})/2 = (5, 9)^T$, 反射点为 $f_r = f(\mathbf{x}^{(r)}) = 45 = f_h$, 所以要进行压缩:

$$\mathbf{x}^{(s)} = 0.25 \begin{pmatrix} 8 \\ 9 \end{pmatrix} + 0.75 \begin{pmatrix} 2 \\ 9 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 9 \end{pmatrix}$$

$$f_s = f(\mathbf{x}^{(s)}) = 18 < f_h$$

于是得一个新的单纯形, 其顶点为

$$\mathbf{x}^{(0)} = (3.5, 9)^T, \mathbf{x}^{(1)} = (6, 11)^T, \mathbf{x}^{(2)} = (4, 7)^T$$

继续进行迭代, 直到求得近似最优解为止。

易见极小点为 $\mathbf{x}^* = (5, 6)^T$, 极小值为 $f^* = f(\mathbf{x}^*) = 0$ 。

3.6.3 Powell 方法(方向加速法)

Powell 方法是由 M. J. D. Powell 于 1964 年首先提出的一种直接搜索方法。可以证明: 在一定条件下, 它是一种共轭方向法, 因此, 对 n 元正定二次函数, 至多迭代 n 次, 即可求得其极小点。Powell 方法被认为是直接搜索法中比较有效的一种方法。

1. Powell 方法的基本思想

从选定的初始点 $\mathbf{x}^{(0)}$ 出发, 先依次沿每个坐标方向求函数 $f(\mathbf{x})$ 的极小点 $\mathbf{x}^{(n)}$, 然后沿方向 $\mathbf{s} = \mathbf{x}^{(n)} - \mathbf{x}^{(0)}$ 再求一次极小, 得

一个新点,仍记为 $\mathbf{x}^{(0)}$,考虑到方向 s 可能比坐标方向更好,为组成下一次迭代要用的 n 个方向,我们丢掉一个坐标方向 s_1 ,加进方向 s ,然后重复上述过程得点 $\mathbf{x}^{(n)}$,又得到一个新方向 $s = \mathbf{x}^{(n)} - \mathbf{x}^{(0)}$,再用它来代替一个坐标方向 s_2, \dots ,如此迭代 n 次,即可得到一组彼此 H 共轭的方向(H 为二次目标函数的海赛阵)。但这样得到的 n 个方向 s_1, s_2, \dots, s_n ,有时是线性相关或近似线性相关的,因此,人们又对它作了一些改进,得到了修正 Powell 方法。修正 Powell 方法虽然不再具有二次收敛性,但它的效果还是比较令人满意的。下面来介绍改进后的 Powell 方法的计算步骤。

2. Powell 方法的计算步骤

1° 给定初始点 $\mathbf{x}^{(0)}$,计算精度 $\varepsilon > 0$, n 个初始的线性无关的搜索方向(一般取为 n 个坐标轴方向)为 $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ 。令

$$\mathbf{s}_j = \mathbf{e}_{j+1}, \quad j = 0, 1, \dots, n-1, \quad k = 0$$

2° 进行一维搜索,决定 λ_k ,使得

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{s}_k) = \min f(\mathbf{x}^{(k)} + \lambda \mathbf{s}_k)$$

令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{s}_k$, 若 $k < n$, 令 $k = k + 1$, 转向 2°; 否则转向 3°;

3° 若 $\|\mathbf{x}^{(n)} - \mathbf{x}^{(0)}\| < \varepsilon$, 计算结束,取 $\mathbf{x}^* \approx \mathbf{x}^{(n)}$; 否则求整数 j ($0 \leq j \leq n-1$), 使

$$\Delta = f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(j+1)}) = \max_{1 \leq i \leq n-1} [f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(i+1)})]$$

4° 令 $f_1 = f(\mathbf{x}^{(0)})$, $f_2 = f(\mathbf{x}^{(n)})$, $f_3 = f(2\mathbf{x}^{(n)} - \mathbf{x}^{(0)})$, 若 $2\Delta < f_1 - 2f_2 + f_3$, 则方向 s_0, s_1, \dots, s_{n-1} 不变,令 $\mathbf{x}^{(0)} = \mathbf{x}^{(n)}$, $k = 0$, 返回 2°; 否则令

$$\mathbf{s}_n = \frac{\mathbf{x}^{(n)} - \mathbf{x}^{(0)}}{\|\mathbf{x}^{(n)} - \mathbf{x}^{(0)}\|}, \quad \text{或 } \mathbf{s}_n = \mathbf{x}^{(n)} - \mathbf{x}^{(0)}$$

$$\mathbf{s}_i = \mathbf{s}_{i+1}, \quad i = j, j+1, \dots, n-1$$

转向 5°;

5° 求 λ_n , 使得

$$f(\mathbf{x}^{(n)} + \lambda_n \mathbf{s}_n) = \min f(\mathbf{x}^{(n)} + \lambda \mathbf{s}_n)$$

令 $\mathbf{x}^{(0)} = \mathbf{x}^{(n)} + \lambda_n \mathbf{s}_n$, $k = 0$, 返回 2° 。

Powell 方法的计算程序见文献 46, 64。

3. 算法应用举例

【例 6.3】 用 Powell 方法求解

$$\min f(\mathbf{x}) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$$

解 1° 取初始点 $\mathbf{x}^{(0)} = (1, 1)^T$, $\mathbf{s}_0 = \mathbf{e}_1 = (1, 0)^T$, $\mathbf{s}_1 = \mathbf{e}_2 = (0, 1)^T$, $\varepsilon = 10^{-2}$ 。

2° 进行一维搜索, 求得 $\lambda_0 = 2$, $\lambda_1 = 1/2$, $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{s}_0 = (3, 1)^T$, $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{s}_1 = (3, 1.5)^T$;

3° 显然 $\|\mathbf{x}^{(2)} - \mathbf{x}^{(0)}\| > \varepsilon$, 因为 $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(1)}) = 4$, $f(\mathbf{x}^{(1)}) - f(\mathbf{x}^{(2)}) = 0.5$, 所以 $\Delta = f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(1)}) = 4$, $j = 0$;

4° 因为 $f_1 = f(\mathbf{x}^{(0)}) = -3$, $f_2 = f(\mathbf{x}^{(2)}) = -7.5$, $f_3 = f(2\mathbf{x}^{(2)} - \mathbf{x}^{(0)}) = f(5, 2) = -7$, 所以 $f_1 - 2f_2 + f_3 = 5 < 2\Delta = 8$, 因此, 令 $\mathbf{s}_2 = \mathbf{x}^{(2)} - \mathbf{x}^{(0)} = (2, 0.5)^T$, $\mathbf{s}_0 = (0, 1)^T$, $\mathbf{s}_1 = (2, 0.5)^T$;

5° 由 $\min f(\mathbf{x}^{(2)} + \lambda \mathbf{s}_2)$, 求得 $\lambda_2 = 2/5$, 所以 $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \lambda_2 \mathbf{s}_2 = (19/5, 17/10)^T$ 。返回 2° , 继续进行迭代, ..., 直到求得所需要的近似极小点为止。此题中 $\mathbf{x}^* = (4, 2)^T$, 极小值 $f^* = -8$ 。

习 题 三

1. 考虑问题: $\min f(x) = |x|$, $x \in \mathbf{R}$ 。构造一个算法:

$$x_{k+1} = \begin{cases} x_k/2, & \text{若 } x_k \leq 1 \\ (x_k - 1)/2 + 1, & \text{若 } x_k > 1 \end{cases}$$

(1) 证明这是一个下降算法;

(2) 取 $x_0 = a$, 试求 $\lim_{k \rightarrow +\infty} x_k$ 。

2. 判断下面点列 $\{\mathbf{x}^{(k)}\}$ 的收敛速度。

$$(1) x^{(k)} = \frac{1}{k}; (2) x^{(k)} = \left(\frac{1}{k}\right)^k; (3) x^{(k)} = \frac{1}{2^k}$$

3. 用 Newton 法求 $f(x) = x^4 - 4x^3 - 6x^2 - 16x + 4$ 的极小点。分别取初始点为 $x_0 = 2.5$, $x_0 = 3$ 。

4. 画出用 Newton 法求函数 $f(x)$ 的极小点的程序框图, 并编写计算程序, 求 $f(x) = 3x^4 - 16x^3 + 30x^2 - 24x + 8$ 的极小值, 取初始点 $x_0 = 3$ 。

5. 证明 3.2 中, 满足方程组(2-4) 的二次插值多项式为

$$g(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)$$

$f(x)$ 的近似极小点为

$$\bar{x} = \frac{1}{2} \frac{f(x_0)(x_1^2 - x_2^2) + f(x_1)(x_2^2 - x_0^2) + f(x_2)(x_0^2 - x_1^2)}{f(x_0)(x_1 - x_2) + f(x_1)(x_2 - x_0) + f(x_2)(x_0 - x_1)}$$

6. 用抛物线法求 $f(x) = x^4/4 - 4x^3/3 + 5x^2/2 - 2x$ 的极小点。

7. 求 $f(x) = e^x - 2x - 1$ 的极小点, 要求精确到小数点后 4 位数字。

8. 画出用平分法求函数 $f(x)$ 的极小点的程序框图, 并求解第 6 题, 取 $a_0 = 0$, $b_0 = 4$ 。

9. 编写用“成功 - 失败”法求函数 $f(x)$ 的极小点的计算程序, 并求解

$$\min f(x) = x^5 + 2x^4 - 4x^3 + x^2 + x + 2$$

10. 求函数 $f(x) = \begin{cases} 4x^3 - 3x^3, & x \geq 0, \\ 4x^3 + 3x^4, & x < 0 \end{cases}$ 的极小点。

11. 编写用 0.618 法求函数 $f(x)$ 的极小点的计算程序, 并求解

$$\min f(x) = e^{-x} + x^2$$

$$\text{s. t. } -2 \leq x \leq 3$$

12. 对第 9 题中的函数 $f(x)$, 求初始搜索区间和初始点。

13. 设所给条件和要求与 3.2 例 2.4 相同, 试用 3.2 节 3.2.8 的算法二, 求在 $\mathbf{x}^{(k)}$ 点沿方向 $\mathbf{s}^{(k)}$ 的一维搜索步长 λ_k 。

14. 导出 3.2 公式(2-28)。

15. 编写不精确一维搜索算法一、二的计算程序, 并计算下面的问题:

$$\min \varphi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{s}^{(k)})$$

其中 $f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, $\mathbf{x}^{(k)} = (-1, 1)^T$, $\mathbf{s}^{(k)} = (1, 1)^T$ 。

16. 用最速下降法求 $f(x_1, x_2) = 3x_1^2 + 2x_2^2 - 4x_1 - 6x_2$ 的极小值。

17. 编写用最速下降法求函数 $f(x)$ 的极小点的计算程序。

18. 分别用 Newton 法和阻尼 Newton 法求函数 $f(x_1, x_2) = 4x_1^2 + x_2^2 - 8x_1 - 4x_2$ 的极小点。

19. 编写用阻尼 Newton 法求函数 $f(x)$ 的极小点的计算程序。并用此程序求函数

$$f(x) = x_1^2 + x_2^2 + x_1x_2 - 3x_1$$

的极小点。

20. 求 Rosenbrock 函数 $f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$ 的极小点。取初始点 $x^{(0)} = (-1.2, 1)^T$ 。

21. 求函数 $f(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$ 的极小点。取初始点 $x^{(0)} = (0, 3)^T$ 。

22. 设 $C = \alpha u u^T$, 其中 $\alpha \neq 0$ 为实常数, $u = (u_1, u_2, \dots, u_n)^T \neq 0$, 证明 C 为秩 1 对称矩阵。

23. 用共轭梯度法求下列函数的极小点。

(1) $f(x) = 4x_1^2 + 4x_2^2 - 4x_1x_2 - 12x_2$, 取初始点 $x^{(0)} = (-1/2, 1)^T$;

(2) $f(x) = x_1 + 2x_2 + 5x_1x_2 + x_1^2 + 13x_2^2$;

(3) $f(x) = x_1^2 - 2x_1x_2 + 2x_2^2 + x_3^2 - x_2x_3 + 2x_1 + 3x_2 - x_3$, 取初始点 $x^{(0)} = (0, 0, 0)^T$ 。

24. 编写用共轭梯度法求函数 $f(x)$ 的极小点的计算程序。

25. 用几种不同的方法求函数 $f(x) = (x_1^3 - x_2)^2 + 2(x_2 - x_1)^4$ 的极小点, 如果所得结果不同, 试说明理由。

26. 求函数 $f(x) = (x_1 - 1)^2 + 5(x_2 - x_1^2)^2$ 的极小点。取初始点 $x^{(0)} = (2, 0)^T$ 。

* 27. 求函数 $f(x) = x_1 + 2x_2^2 + \exp(x_1^2 + x_2^2)$ 的极小点。取初始点 $x^{(0)} = (1, 0)^T$ 。

28. 用 DFP 算法求 $f(x) = -x_1 + x_2 + 2x_1x_2 + x_1^2 + 3x_2^2$ 的极小点。

29. 用 BFGS 算法求 $f(x) = x_1^2 + x_1x_2 + x_2^2$ 的极小点, 取初始点 $x^{(0)} = (3, 2)^T$ 。

30. 编写用 DFP 算法求函数 $f(x)$ 的极小点的计算程序。并计算习题 26,

31. 编写用 BFGS 方法求函数 $f(\mathbf{x})$ 的极小点的计算程序, 并计算习题 26, 27。

32. 证明: 满足条件(5-14)和(5-19)的矩阵 H_k 和 H_{k+1} 必满足条件(5-17)。

33. 试由公式(5-23)和(5-21)导出公式(5-23')和(5-24)。

34. 编写用 Hoshino 算法求函数 $f(\mathbf{x})$ 的极小点的计算程序, 并计算习题 26, 27。

35. 用 Hooke-Jeeves 方法求解:

(1) $\min f(\mathbf{x}) = 3x_1^2 + 5x_2^2$, 取 $\mathbf{x}^{(1)} = (2, 1)^T$, $\alpha = 1$, $d = 1$, $\epsilon = 0.1$ 。

(2) $\min f(\mathbf{x}) = 4x_1^2 + x_2^2 - 40x_1 - 12x_2 + 136$, 取 $\mathbf{x}^{(1)} = (4, 8)^T$, $\alpha = 1$, $d = 1$, $\epsilon = 0.2$ 。

(3) $\min f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, 取 $\mathbf{x}^{(1)} = (-1.2, 1.0)^T$ 。

36. 编写用 Hooke-Jeeves 方法求函数极小点的计算程序, 并计算习题 35(2), (3)。

37. 求解如下优化问题:

(1) $\min f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1 - x_1x_2 + 3$, 取 $\mathbf{x}^{(1)} = (0, 0)^T$ 。

(2) $\max f(\mathbf{x}) = 3x_2^2 - x_1^2 + 5x_1x_2 + x_1 + 2x_2$ 。

(3) $\min f(\mathbf{x}) = 4(1 - x_1)^2 + 5(x_2 - x_1^2)^2$, 取 $\mathbf{x}^{(1)} = (2, 0)^T$ 。

38. 编写用单纯形法求函数 $f(\mathbf{x})$ 的极小点的计算程序, 并计算习题 35(3), 37(3)。

39. 用 Powell 方法求解:

(1) $\min f(\mathbf{x}) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2$, 取 $\mathbf{x}^{(0)} = (1, 1)^T$;

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

40. 编写用 Powell 方法求函数极小点的计算程序, 并计算习题 39(1), (2)。

第 4 章 约束优化方法

约束优化问题是应用中经常遇到的一类数学规划问题,它的解法是人们最为关心的,因此研究的人比较多,提供的方法也较多,但目前尚没有一种对一切问题都普遍有效的算法,而且求得的解多是局部最优解。

约束优化方法大体可分为以下四类:

(1) 用线性规划或二次规划来逐次逼近非线性规划的方法,如 SLP 法, SQP 法等;

(2) 把约束优化问题转化为无约束优化问题来求解的方法,如 SUMT 外点法, SUMT 内点法, 乘子法等;

(3) 对约束优化问题不预先作转换,直接进行处理的分析方法,如可行方向法,梯度投影法,既约梯度法等;

(4) 对约束优化问题不预先作转换的直接搜索方法,如复形法,随机试验法等。

本章主要介绍求解约束优化问题的一些比较有效的算法,包括 SUMT 内点法, SUMT 外点法,可行方向法,梯度投影法,既约梯度法,乘子法和二次逼近法等。对其他方法有兴趣的读者可参看文献 11, 16, 29~38。

4.1 引言 Kuhn-Tucker 条件

一般的约束优化问题是指:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, p \\ & \mathbf{x} \in X \subset \mathbf{R}^n \end{cases} \quad (1-1)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 用 S 表示问题 (1-1) 的可行域, 即

$$S = \{\mathbf{x} | \mathbf{x} \in X, g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m, \\ h_j(\mathbf{x}) = 0, j = 1, 2, \dots, p\} \quad (1-2)$$

以后若不另加说明, 我们总假定函数 f, g_i, h_j 具有解法所要求的各阶偏导数。

下面我们来叙述一点 $\mathbf{x}^* \in S$ 为问题 (1-1) 的(局部)极小点的必要条件和充分条件, 因为许多方法都与这些条件有密切的联系。详细的讨论, 可参看文献 11, 16, 36。

设 \mathbf{x}^0 为可行点, 即 $\mathbf{x}^0 \in S$, 定义

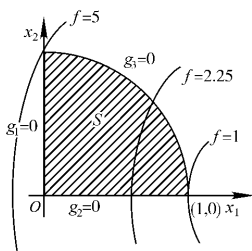
$$I(\mathbf{x}^0) = \{i | g_i(\mathbf{x}^0) = 0, 1 \leq i \leq m\} \quad (1-3)$$

当 $i \in I(\mathbf{x}^0)$ 时, 对应的约束 $g_i(\mathbf{x}) \leq 0$ 称为在 \mathbf{x}^0 处的紧约束, 或起作用约束, 而 $I(\mathbf{x}^0)$ 称为 \mathbf{x}^0 点紧约束的指标集, 显然

$$I(\mathbf{x}^0) \subset \{1, 2, \dots, m\}$$

【例 1.1】 考察问题

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 2)^2 + x_2^2 \\ \text{s. t.} & g_1(\mathbf{x}) = -x_1 \leq 0 \\ & g_2(\mathbf{x}) = -x_2 \leq 0 \\ & g_3(\mathbf{x}) = x_1^2 + x_2^2 - 1 \leq 0 \end{cases} \quad (1-4)$$



可行集 S 如图 4-1 所示。易见最优点 $\mathbf{x}^* = (1, 0)^T$, 它是两个约束 $g_2(\mathbf{x}) = 0$ 和 $g_3(\mathbf{x}) = 0$ 的交点, 所以

这里 $I(\mathbf{x}^*) = \{2, 3\}$, $g_2(\mathbf{x}) \leq 0$ 和 $g_3(\mathbf{x}) \leq 0$ 为 \mathbf{x}^* 点的紧约束。

图 4-1

$f = 1, f = 2.25, f = 5$ 为等值线。

定理 1.1 最优性的一阶必要条件

设 \mathbf{x}^* 是问题(1-1)的一个可行解, $X \subset \mathbf{R}^n$ 为非空开集, $f, g_i (i \in I(\mathbf{x}^*))$ 在 \mathbf{x}^* 点可微, $g_i (i \notin I(\mathbf{x}^*))$ 在 \mathbf{x}^* 点连续, $\nabla g_i(\mathbf{x}^*), i \in I(\mathbf{x}^*), \nabla h_j(\mathbf{x}^*), j = 1, 2, \dots, p$ 线性无关, 若 \mathbf{x}^* 是(1-1)的局部最优解, 则有实数 u_i, v_j (其中 $i \in I(\mathbf{x}^*), j = 1, 2, \dots, p$) 使得

$$\nabla f(\mathbf{x}^*) + \sum_{i \in I} u_i \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^p v_j \nabla h_j(\mathbf{x}^*) = \mathbf{0} \quad (1-5a)$$

$$u_i \geq 0, i \in I(\mathbf{x}^*) \quad (1-5b)$$

如果加上 $i \notin I$ 时, g_i 在 \mathbf{x}^* 点可微, 则上述条件可改写为

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m u_i \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^p v_j \nabla h_j(\mathbf{x}^*) = \mathbf{0} \quad (1-5a')$$

$$u_i \geq 0, i = 1, 2, \dots, m \quad (1-5b')$$

$$u_i g_i(\mathbf{x}^*) = 0, i = 1, 2, \dots, m \quad (1-5c)$$

上述条件(1-5)称为 Kuhn-Tucker 最优性**必要条件**, 满足条件(1-5)的点称为 Kuhn-Tucker 点, 简称为 K-T 点。条件(1-5c)称为互补松弛条件。这就导致我们引入 Lagrange 函数

$$L(\mathbf{x}, \mathbf{u}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^m u_i g_i(\mathbf{x}) + \sum_{j=1}^p v_j h_j(\mathbf{x}) \quad (1-6)$$

其中

$$\mathbf{u} = (u_1, u_2, \dots, u_m)^T, \mathbf{v} = (v_1, v_2, \dots, v_p)^T$$

称为 Lagrange 乘子向量。

由定理 1.1 可知: 若 \mathbf{x}^* 是问题(1-1)的最优解, 则存在 $\mathbf{u}^* \geq \mathbf{0}, \mathbf{v}^*$, 使 $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*)$ 是 $L(\mathbf{x}, \mathbf{u}, \mathbf{v})$ 的稳定点。

Kuhn-Tucker 条件的几何解释

为便于说明, 假设(1-1)中没有等式约束, 即 $p = 0$ 的情况。令

$c = \{y | y = \sum_{i \in I} u_i \nabla g(x^*), i \in I, u_i \geq 0, u \neq 0\}$, 易见 C 是由紧约束梯度张成的锥。于是 Kuhn-Tucker 条件

$$-\nabla f(x^*) = \sum_{i \in I} u_i \nabla g(x^*), u_i \geq 0 (i \in I) \text{ 和 } u^* \neq 0$$

可以解释为: 若 x^* 是问题(1-1)的最优解, 则 $-\nabla f(x^*)$ 必位于点 x^* 的上述锥 C 之中。即 $-\nabla f(x^*) \in C$ 。

图 4-2 画出了两个点 $x^{(1)}$ 和 $x^{(2)}$, 注意 $-\nabla f(x^{(1)})$ 位于由 $x^{(1)}$ 点的紧约束的梯度张成的锥中, 因而 $x^{(1)}$ 是 K-T 点; 另一方面, $-\nabla f(x^{(2)})$ 位于由 $x^{(2)}$ 点的紧约束的梯度张成的锥之外, 于是 $x^{(2)}$ 不是 K-T 点。

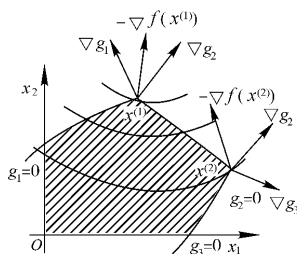


图 4-2

定理 1.2 最优性的充分条件:

设 $X \subset \mathbf{R}^n$ 为非空开集, $x^* \in S$, 在 x^* 点, K-T 条件成立, 即存在数 $u_i \geq 0 (i \in I(x^*))$ 和 $v_i (i = 1, 2, \dots, p)$, 使得

$$\nabla f(x^*) + \sum_{i \in I} u_i \nabla g_i(x^*) + \sum_{i=1}^p v_i \nabla h_i(x^*) = 0$$

令 $J = \{i | v_i > 0\}$, $K = \{i | v_i < 0\}$, 设 f 在 x^* 点伪凸, $g_i (i \in I)$ 在 x^* 点拟凸, $h_i (i \in J)$ 在 x^* 点拟凸, $h_i (i \in K)$ 在 x^* 点拟凹, 则 x^* 是问题(1-1)的全局最优解。

关于定理 1.1 和定理 1.2 的证明可参看文献 11, 关于函数在一点伪凸、拟凸和拟凹的定义见第一章 1.3 节。这里我们仅仅指出: 若函数为可微凸函数, 则它一定是伪凸和拟凸的, 若函数为凹函数, 则它一定是拟凹的。

【例 1.2】 考察问题

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s. t.} & g_1(\mathbf{x}) = x_1^2 + x_2^2 - 5 \leq 0 \\ & g_2(\mathbf{x}) = -x_1 \leq 0 \\ & g_3(\mathbf{x}) = -x_2 \leq 0 \\ & h_1(\mathbf{x}) = x_1 + 2x_2 - 4 = 0 \end{cases}$$

易见 $\mathbf{x}^{(0)} = (2, 1)^T \in S$, 且只有一个不等式约束 $g_1(\mathbf{x}) \leq 0$ 是紧约束。我们来验证 $\mathbf{x}^{(0)}$ 点是 K-T 点。

$$\text{因} \quad \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$$

$$\nabla g_1(\mathbf{x}^{(0)}) = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$\nabla h_1(\mathbf{x}^{(0)}) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

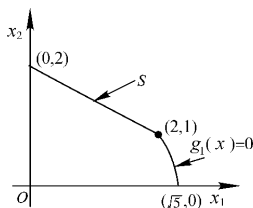


图 4-3

所以由式(1-5a)得

$$\begin{pmatrix} -2 \\ -2 \end{pmatrix} + u_1 \begin{pmatrix} 4 \\ 2 \end{pmatrix} + v_1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \mathbf{0}$$

$$\text{或} \quad \begin{cases} 4u_1 + v_1 = 2 \\ 2u_1 + 2v_1 = 2 \end{cases}$$

所以 $u_1 = 1/3$, $v_1 = 2/3$, $u_2 = u_3 = 0$, 因此 $\mathbf{x}^{(0)}$ 是 K-T 点。

容易看出, f , g_1 为凸函数, h_1 为线性函数, 所以在 $\mathbf{x}^{(0)}$ 点, 定理 1.2 中的条件是满足的, 因此 $\mathbf{x}^{(0)}$ 点也是此问题的全局最优解, 最优值为 $f(\mathbf{x}^{(0)}) = 2$ 。

下面再来研究另一点 $\mathbf{x}^{(1)} = (0, 2)^T$ 。这时, $I(\mathbf{x}^{(1)}) = \{2\}$, 即紧约束为 $g_2(\mathbf{x}) \leq 0$ 。

$$\nabla f(\mathbf{x}^{(1)}) = \begin{pmatrix} -6 \\ 0 \end{pmatrix}, \quad \nabla g_2(\mathbf{x}^{(1)}) = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \nabla h_1(\mathbf{x}^{(1)}) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

由式(1-5a) 得

$$\begin{pmatrix} -6 \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + v_1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0$$

$$\text{或} \quad \begin{cases} v_1 - u_2 = 6 \\ 2v_1 = 0 \end{cases}$$

$$\text{所以} \quad v_1 = 0, u_2 = -6 < 0$$

因此,在 $x^{(1)}$ 点, K-T 条件不成立, 所以 $x^{(1)}$ 不是 K-T 点, 当然不可能是最优点。 ■

4.2 惩罚函数法

考察约束优化问题

$$\begin{cases} \min & f(x) \\ \text{s. t.} & g_i(x) \leq 0, i = 1, 2, \dots, m \\ & h_j(x) = 0, j = 1, 2, \dots, p \\ & x \in X \subset \mathbf{R}^n, x = (x_1, x_2, \dots, x_n)^T \end{cases} \quad (2-1)$$

其中 $f, g_i (i = 1, 2, \dots, m), h_j (j = 1, 2, \dots, p) \in C^1$ 。下面将要介绍的惩罚函数法和碰壁函数法是早期求解约束优化问题(2-1) 的一类重要而常用的方法, 其基本思想是把约束问题(2-1) 转化为一个或一系列无约束问题来求解, 所以也称为序列无约束极小化技术, 简称为 SUMT 法, 通常把惩罚函数法称为 SUMT 外点法, 碰壁函数法称为 SUMT 内点法。

4.2.1 惩罚函数的性质和构造

对问题(2-1) 定义惩罚函数

$$F(x, M) = f(x) + Mp(x) \quad (2-2)$$

其中 $M > 0$ 为常数, 称为惩罚因子, $p(x)$ 是定义在 \mathbf{R}^n 上的一个函数, 称为惩罚项, 它满足:

- (1) $p(x)$ 是连续的;
- (2) 对任意 $x \in \mathbf{R}^n$, 有 $p(x) \geq 0$;
- (3) 当且仅当 $x \in S$ 时, $p(x) = 0$ 。

而 S 是 (2-1) 的可行集, 即

$$S = \{x | g_i(x) \leq 0, i = 1, 2, \dots, m; \\ h_j(x) = 0, j = 1, 2, \dots, p, x \in X\}$$

通常对等式约束, 定义

$$g_j^+(x) = (h_j(x))^2, j = 1, 2, \dots, p \quad (2-3)$$

对不等式约束, 定义

$$g_{i+p}^+(x) = \begin{cases} 0, & g_i(x) \leq 0, \\ (g_i(x))^2, & g_i(x) > 0, \end{cases} i = 1, 2, \dots, m \quad (2-4)$$

令 $L = p + m$, 于是, 惩罚函数为

$$F(x, M_k) = f(x) + M_k \sum_{i=1}^L g_i^+(x) \quad (2-5)$$

其中 $M_k > 0$, 且 $M_1 < M_2 < \dots < M_k < M_{k+1} < \dots$

$$\lim_{k \rightarrow \infty} M_k = +\infty$$

容易验证, 这样定义的惩罚项 $p(x) = \sum_{i=1}^L g_i^+(x)$ 满足上述的三条性质。

(1) 显然 $g_i^+(x)$ ($i = 1, 2, \dots, L$) 是连续的, 因此 $p(x) = \sum_{i=1}^L g_i^+(x)$ 连续。

(2) 对于所有的 $x \in \mathbf{R}^n$, 显然有 $g_i^+(x) \geq 0$, 所以 $p(x) \geq 0$ 。

(3) 当 $x \in S$ 时, $g_i^+(x) = 0$ ($i = 1, 2, \dots, L$), 所以 $p(x) = \sum_{i=1}^L g_i^+(x) = 0$; 反之, 若 $p(x) = \sum_{i=1}^L g_i^+(x) = 0$, 由于 $g_i^+(x) \geq 0$, 所以 $g_i^+(x) = 0$ ($i = 1, 2, \dots, L$)。

由上述分析可见, 当 x 满足约束条件, 即 $x \in S$ 时, 则 $g_i^+(x) =$

$0 (i = 1, 2, \dots, L)$; 当约束条件被破坏, 即 $x \notin S$ 时, 则至少有一个 $i (1 \leq i \leq L)$, 使 $g_i^+(x) > 0$, 从而 $p(x) > 0$ 。约束条件被破坏得越厉害, 则 $p(x) > 0$ 取值越大, 从而 $F(x, M) = f(x) + Mp(x)$ 也就越大, 即对于约束条件被破坏是一种惩罚, M 越大, 则惩罚得越厉害。反之, 当约束条件满足时, 则 $p(x) = 0$, 这时不管 $M > 0$ 取多么大的值, $F(x, M) \equiv f(x)$, 就是说, 当约束条件满足时, 不受惩罚, 由此可见惩罚项及惩罚函数的意义。

关于惩罚因子 M_k 的取法, 根据计算经验可以选取 $M_{k+1} = M_k c$, $c \in [2, 50]$, 常取 $c \in [4, 10]$ 。

4.2.2 计算方法和程序框图

用惩罚函数法求解约束优化问题(2-1)的计算步骤如下:

- (1) 选取 $M_1 > 0$, 精度 $\varepsilon > 0$, $c \geq 2$, 初始点 $x^{(0)}$, 令 $k = 1$;
- (2) 以 $x^{(k-1)}$ 为初始点, 求解无约束优化问题

$$\min F(x, M_k) = f(x) + M_k \sum_{i=1}^L g_i^+(x) \quad (2-6)$$

设其最优解为 $x^{(k)} = x(M_k)$;

- (3) 令

$$\begin{aligned} \tau_1 &= \max_{1 \leq i \leq p} \{ |h_i(x^{(k)})| \}, \tau_2 = \max_{1 \leq i \leq m} \{ g_i(x^{(k)}) \} \\ \tau &= \max \{ \tau_1, \tau_2 \} \end{aligned}$$

- (4) 若 $\tau < \varepsilon$, 则迭代结束, 取 $x^* = x^{(k)}$; 否则令 $M_{k+1} = cM_k$, $k = k + 1$, 转回第(2)步。

在(2-6)式中, 惩罚项 $p(x) = \sum_{i=1}^L g_i^+(x)$, 可以采用(2-3)和(2-4)式的定义方法, 也可用其他的办法来定义, 只要能保证 $p(x)$ 具有前面所述的三条性质就行。

上述算法的结束准则 $\tau < \varepsilon$, 也可以改为: 若 $M_k p(x^{(k)}) < \varepsilon$, 则取 $x^* \approx x^{(k)}$, 迭代结束; 否则令 $M_{k+1} = cM_k$, $k = k + 1$, 继续

进行迭代。

根据上面的计算步骤,可画出 SUMT 外点法的程序框图如图 4-4 所示。

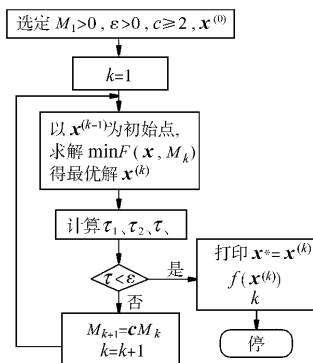


图 4-4

4.2.3 收敛性分析

设 $\mathbf{x}^{(k)} = \mathbf{x}(M_k)$ 为无约束优化问题(2-6)的最优解。分两种情况进行讨论

(1) 如果某一个 $M_{k_0} (k_0 \geq 1)$ 对应的无约束问题(2-6)的最优解 $\mathbf{x}^{(k_0)} \in S$, 则 $\mathbf{x}^{(k_0)}$ 为原来的约束优化问题(2-1)的最优解。

证明 因为对一切 $\mathbf{x} \in S$, 有 $g_i^+(\mathbf{x}) = 0, i = 1, 2, \dots, L$ 所以

$$\begin{aligned}
 f(\mathbf{x}) &= f(\mathbf{x}) + M_{k_0} \sum_{i=1}^L g_i^+(\mathbf{x}) \\
 &= F(\mathbf{x}, M_{k_0}) \geq F(\mathbf{x}^{(k_0)}, M_{k_0}) \\
 &= f(\mathbf{x}^{(k_0)}) + M_{k_0} \sum_{i=1}^L g_i^+(\mathbf{x}^{(k_0)}) = f(\mathbf{x}^{(k_0)})
 \end{aligned}$$

因而 $\mathbf{x}^{(k_0)}$ 是问题(2-1)的最优解。

(2) 如果第一种情况总不发生,这时就得到一个无穷点列

$$\{\mathbf{x}^{(k)}\}, \mathbf{x}^{(k)} \in S, k = 1, 2, \dots$$

这时,可以证明在某些条件下, $\{\mathbf{x}^{(k)}\}$ 的任何极限点 \mathbf{x}^* 都是原来约束优化问题(2-1)的最优解。为此,我们先证明以下两个引理。

引理 1 对 $M_{k+1} > M_k$, 下列不等式成立。

$$(1) F(\mathbf{x}^{(k)}, M_k) \leq F(\mathbf{x}^{(k+1)}, M_{k+1}) \quad (2-7)$$

$$(2) p(\mathbf{x}^{(k)}) \geq p(\mathbf{x}^{(k+1)}) \quad (2-8)$$

$$(3) f(\mathbf{x}^{(k)}) \leq f(\mathbf{x}^{(k+1)}) \quad (2-9)$$

$$\begin{aligned} \text{证明} \quad (1) \quad & F(\mathbf{x}^{(k+1)}, M_{k+1}) = f(\mathbf{x}^{(k+1)}) + M_{k+1}p(\mathbf{x}^{(k+1)}) \\ & \geq f(\mathbf{x}^{(k+1)}) + M_k p(\mathbf{x}^{(k+1)}) \\ & \geq f(\mathbf{x}^{(k)}) + M_k p(\mathbf{x}^{(k)}) = F(\mathbf{x}^{(k)}, M_k) \end{aligned}$$

这就证明了(2-7)。

(2) 因

$$\begin{aligned} f(\mathbf{x}^{(k)}) + M_k p(\mathbf{x}^{(k)}) & \leq f(\mathbf{x}^{(k+1)}) + M_k p(\mathbf{x}^{(k+1)}) \\ f(\mathbf{x}^{(k+1)}) + M_{k+1} p(\mathbf{x}^{(k+1)}) & \leq f(\mathbf{x}^{(k)}) + M_{k+1} p(\mathbf{x}^{(k)}) \end{aligned}$$

将上面两式相加,并消去相同的项,得

$$M_k p(\mathbf{x}^{(k)}) + M_{k+1} p(\mathbf{x}^{(k+1)}) \leq M_k p(\mathbf{x}^{(k+1)}) + M_{k+1} p(\mathbf{x}^{(k)})$$

$$\text{故} \quad (M_{k+1} - M_k) p(\mathbf{x}^{(k+1)}) \leq (M_{k+1} - M_k) p(\mathbf{x}^{(k)})$$

又因 $M_{k+1} > M_k$, 所以 $p(\mathbf{x}^{(k+1)}) \leq p(\mathbf{x}^{(k)})$, 即式(2-8)成立。

(3) 因

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) + M_k p(\mathbf{x}^{(k+1)}) & \geq f(\mathbf{x}^{(k)}) + M_k p(\mathbf{x}^{(k)}) \\ M_k p(\mathbf{x}^{(k+1)}) & \leq M_k p(\mathbf{x}^{(k)}) \end{aligned}$$

综合上面二式的结果,即得

$$f(\mathbf{x}^{(k+1)}) \geq f(\mathbf{x}^{(k)})$$

引理 2 设 \mathbf{x}^* 是约束优化问题(2-1)的一个最优解,则对每一个 k , 有

$$f(\mathbf{x}^*) \geq F(\mathbf{x}^{(k)}, M_k) \geq f(\mathbf{x}^{(k)}) \quad (2-10)$$

证明 因为 \mathbf{x}^* 是问题(2-1)的一个最优解,所以

$$\mathbf{x}^* \in S, p(\mathbf{x}^*) = 0$$

因此

$$\begin{aligned} f(\mathbf{x}^*) &= f(\mathbf{x}^*) + M_k p(\mathbf{x}^*) \geq f(\mathbf{x}^{(k)}) + M_k p(\mathbf{x}^{(k)}) \\ &= F(\mathbf{x}^{(k)}, M_k) \geq f(\mathbf{x}^{(k)}) \end{aligned}$$

定理 2.1 设 $f(\mathbf{x}) \in C$, $g_i(\mathbf{x}) \in C$ ($i = 1, 2, \dots, m$), $h_j(\mathbf{x}) \in C$ ($j = 1, 2, \dots, p$), $\{\mathbf{x}^{(k)}\}$ 是由惩罚函数法产生的一个序列,则这个序列的任一个极限点都是约束优化问题(2-1)的一个最优解。

证明 设 $\{\mathbf{x}^{(k_i)}\}$ 是 $\{\mathbf{x}^{(k)}\}$ 的一个收敛的子序列,其极限为 $\bar{\mathbf{x}}$, 即

$$\lim_{i \rightarrow \infty} \mathbf{x}^{(k_i)} = \bar{\mathbf{x}}$$

于是由 $f(\mathbf{x}) \in C$ 得知

$$\lim_{i \rightarrow \infty} f(\mathbf{x}^{(k_i)}) = f(\bar{\mathbf{x}}) \quad (2-11)$$

要证明 $\bar{\mathbf{x}}$ 是问题(2-1)的最优解,即要证 $\bar{\mathbf{x}} \in S$, 且

$$f(\bar{\mathbf{x}}) = \min_{\mathbf{x} \in S} f(\mathbf{x}) = f(\mathbf{x}^*) \triangleq f^*$$

(1) 由引理1和引理2知 $F(\mathbf{x}^{(k)}, M_k)$ 是一个不减且有上界 f^* 的序列,故

$$\lim_{i \rightarrow \infty} F(\mathbf{x}^{(k_i)}, M_{k_i}) = F^* \leq f^* \quad (2-12)$$

用(2-12)减去(2-11)得

$$\lim_{i \rightarrow \infty} M_{k_i} p(\mathbf{x}^{(k_i)}) = F^* - f(\bar{\mathbf{x}})$$

因为,当 $i \rightarrow \infty$ 时, $M_{k_i} \rightarrow \infty$, 且 $p(\mathbf{x}^{(k_i)}) \geq 0$, 所以

$$\lim_{i \rightarrow \infty} p(\mathbf{x}^{(k_i)}) = 0$$

又因为 $g_i(\mathbf{x}) \in C$, $h_j(\mathbf{x}) \in C$, 因而 $p(\mathbf{x}) \in C$, 所以 $p(\bar{\mathbf{x}}) = 0$, 故 $\bar{\mathbf{x}} \in S$ 。

(2) 由引理 2 知 $f(\mathbf{x}^{(k_i)}) \leq f^*$, 令 $i \rightarrow \infty$, 得

$$f(\bar{\mathbf{x}}) \leq f^* = f(\mathbf{x}^*)$$

而 $f(\mathbf{x}^*) = \min_{\mathbf{x} \in S} f(\mathbf{x})$, 所以 $f(\bar{\mathbf{x}}) \geq f(\mathbf{x}^*)$, 于是必有

$$f(\bar{\mathbf{x}}) = f(\mathbf{x}^*)$$

综合(1)和(2), 即知 $\bar{\mathbf{x}}$ 是问题(2-1)的最优解。 ■

4.2.4 算法应用举例

【例 2.1】 用 SUMT 外点法求解

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s. t.} & h(\mathbf{x}) = x_1 + x_2 - 4 = 0 \end{cases}$$

解 令 $F(\mathbf{x}, M) = (x_1 - 3)^2 + (x_2 - 2)^2 + M(x_1 + x_2 - 4)^2$

$$\frac{\partial F}{\partial x_1} = 2(x_1 - 3) + 2M(x_1 + x_2 - 4)$$

$$\frac{\partial F}{\partial x_2} = 2(x_2 - 2) + 2M(x_1 + x_2 - 4)$$

令 $\frac{\partial F}{\partial x_1} = 0, \frac{\partial F}{\partial x_2} = 0$, 得

$$x_1 = \frac{5M + 3}{2M + 1}, \quad x_2 = \frac{3M + 2}{2M + 1}$$

因 $\frac{\partial^2 F}{\partial x_1^2} = 2(M + 1), \frac{\partial^2 F}{\partial x_2^2} = 2(M + 1), \frac{\partial^2 F}{\partial x_1 \partial x_2} = 2M$

故 $\nabla^2 F = \begin{pmatrix} 2(M + 1) & 2M \\ 2M & 2(M + 1) \end{pmatrix}$

由于 $M > 0$, 所以 $\nabla^2 F$ 是正定的, 因此 $F(\mathbf{x}, M)$ 在

$$\mathbf{x}(M) = \left(\frac{5M + 3}{2M + 1}, \frac{3M + 2}{2M + 1} \right)^T$$

处取得极小值。令 $M \rightarrow +\infty$, 得 $\mathbf{x}^* = \lim_{M \rightarrow +\infty} \mathbf{x}(M) = (5/2, 3/2)^T$,

易见 $h(\mathbf{x}^*) = 0$, 所以 \mathbf{x}^* 即为 $f(\mathbf{x})$ 在约束条件 $h(\mathbf{x}) = 0$ 下的极小点。而极小值为 $f(\mathbf{x}^*) = 1/2$ 。 ■

【例 2.2】 用外点法求解

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 2)^2 \\ \text{s. t.} & h(\mathbf{x}) = x_1 + x_2 - 4 \leq 0 \end{cases}$$

解 令 $h^+(\mathbf{x}) = \begin{cases} 0, & \text{当 } h(\mathbf{x}) \leq 0 \\ (h(\mathbf{x}))^2, & \text{当 } h(\mathbf{x}) > 0 \end{cases}$

$$p(\mathbf{x}, M) = (x_1 - 3)^2 + (x_2 - 2)^2 + Mh^+(\mathbf{x})$$

$$\frac{\partial p}{\partial x_1} = \begin{cases} 2(x_1 - 3) + 2M(x_1 + x_2 - 4), & \text{当 } h(\mathbf{x}) > 0 \\ 2(x_1 - 3), & \text{当 } h(\mathbf{x}) \leq 0 \end{cases}$$

$$\frac{\partial p}{\partial x_2} = \begin{cases} 2(x_2 - 2) + 2M(x_1 + x_2 - 4), & \text{当 } h(\mathbf{x}) > 0 \\ 2(x_2 - 2), & \text{当 } h(\mathbf{x}) \leq 0 \end{cases}$$

令 $\nabla p = \mathbf{0}$, 得

$$\begin{cases} (x_1 - 3) + M(x_1 + x_2 - 4) = 0 \\ (x_2 - 2) + M(x_1 + x_2 - 4) = 0 \end{cases}$$

所以 $x_1 = \frac{5M+3}{2M+1}$, $x_2 = \frac{3M+2}{2M+1}$, 令 $M \rightarrow +\infty$, 得

$$\mathbf{x}^* = \lim_{M \rightarrow +\infty} \mathbf{x}(M) = (5/2, 3/2)^T$$

显然有 $h(\mathbf{x}^*) \leq 0$, 所以 $f(\mathbf{x})$ 在 \mathbf{x}^* 点取得极小值 $f(\mathbf{x}^*) = 1/2$ 。 ■

4.3 碰壁函数法

碰壁函数法适用于形如

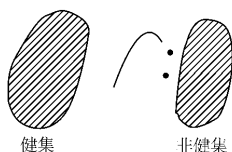
$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in S \end{cases} \quad (3-1)$$

的约束优化问题, 其中可行集 S 的内部(用 $\text{int}S$ 表示)非空, 而且 $\text{int}S$ 中的点可以任意地接近于 S 的任一点, 从直观上来看, 就是 S 不能包含孤立点和孤立的弧段, 这样的集合 S 称为健集(Robust set)。图 4-5 给出了一个健集与非健集的例子。在这种要求下, 显然

S 中不能包含有等式约束, 即 S 只能是形如

$$S = \{x | g_i(x) \leq 0, i = 1, 2, \dots, m\}$$

(3-2)



的约束集。

碰壁函数法是从一个可行点 $x^{(0)}$ 出

图 4-5

发, 在可行点之间进行迭代的一种方法。为了使迭代点保持为可行点, 在约束集 S 的边界上建造一道“围墙”, 它阻挡迭代点列离开可行集 S 。碰壁项是定义于 $\text{int}S$ 的一个函数 $B(x)$, 它满足条件:

- (1) $B(x)$ 是连续的;
- (2) $B(x) \geq 0$;
- (3) 当 x 趋近于 S 的边界时, $B(x) \rightarrow \infty$ 。

通常定义碰壁项为

$$B(x) = \sum_{i=1}^m g_i^+(x) \quad (3-3)$$

$$g_i^+(x) = -\frac{1}{g_i(x)}, i = 1, 2, \dots, m \quad (3-4)$$

或

$$g_i^+(x) = -\ln(-g_i(x)), i = 1, 2, \dots, m \quad (3-5)$$

而碰壁函数定义为

$$F(x, r_k) = f(x) + r_k B(x) \quad (3-6)$$

其中 $r_k > 0$, 且 $r_1 > r_2 > r_3 > \dots > r_k > r_{k+1} > \dots, \lim_{k \rightarrow \infty} r_k = 0$ 。

容易看出: 当 x 靠近可行域 S 的边界时, $g_1(x), \dots, g_m(x)$ 中, 至少有一个 $i (1 \leq i \leq m)$, 使 $g_i(x) \rightarrow 0$ 。因此, $g_i^+(x) \rightarrow +\infty$, 从而 $B(x) \rightarrow +\infty$ 。这样求解问题(3-1), 就可转化为求解问题

$$\begin{cases} \min & F(x, r) = f(x) + rB(x) \\ \text{s. t.} & x \in \text{int}S \end{cases} \quad (3-7)$$

其中 $r > 0$, 虽然从形式上看, 问题(3-7) 仍是一个约束问题, 但

是,由于在 S 的边界附近,它的目标函数值趋近于无穷大,所以只要从 S 的一个内点开始迭代,并注意控制一维搜索的步长,就可使 $\mathbf{x}^{(k)}$ 不越出可行域,因而不必直接地与约束打交道,也就是说,从计算的观点来看,它是无约束问题。

4.3.1 SUMT 内点法的计算步骤

(1) 选取 $r_1 > 0$ (例如取 $r_1 = 1$), $c \geq 2$ (一般取 $c \in [2, 50]$, 常取 $c \in [4, 10]$), 精度 $\varepsilon > 0$ 。

(2) 求可行集 S 的一个内点 $\mathbf{x}^{(0)} \in \text{int}S$; 令 $k = 1$ 。

(3) 以 $\mathbf{x}^{(k-1)}$ 为初始点,使用求解无约束优化问题的方法求解

$$\begin{cases} \min & F(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k B(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in \text{int}S \end{cases}$$

设其最优解为 $\mathbf{x}^{(k)} = \mathbf{x}(r_k)$ 。

(4) 检查 $\mathbf{x}^{(k)}$ 是否满足结束准则,若 $\mathbf{x}^{(k)}$ 满足下面的结束准则,

则迭代结束,取 $\mathbf{x}^* \approx \mathbf{x}^{(k)}$; 否则取 $r_{k+1} < r_k$ (例如取 $r_{k+1} = \frac{r_k}{c}$), 令 $k = k + 1$, 返回第(3)步。

关于结束准则,常用的有以下四种,可以选用其中之一,或要求其中两种同时满足才停止迭代。准则(4)用得更广泛一些。

(1) $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \varepsilon$, $\|\cdot\|$ 为 \mathbf{R}^n 中的某种范数, $\varepsilon > 0$ 为计算精度。

(2) $|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k-1)})| < \varepsilon$ 。

(3) $|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k-1)})| / |f(\mathbf{x}^{(k)})| < \varepsilon_1$, 其中 $\varepsilon_1 > 0$ 为精度。

(4) $r_k B(\mathbf{x}^{(k)}) < \varepsilon$ 。

4.3.2 算法应用举例

【例 3.1】 用内点法求解

$$\begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s. t.} & x_1 \geq 1 \end{cases}$$

解 易见最优解为 $\mathbf{x}^* = (1, 0)^T$, $f^* = f(\mathbf{x}^*) = 1$ 。

令 $F(\mathbf{x}, r) = x_1^2 + x_2^2 - r \ln(x_1 - 1)$, 则

$$\frac{\partial F}{\partial x_1} = 2x_1 - \frac{r}{x_1 - 1}, \quad \frac{\partial F}{\partial x_2} = 2x_2$$

令 $\frac{\partial F}{\partial x_1} = 0$, $\frac{\partial F}{\partial x_2} = 0$, 得 $2x_1^2 - 2x_1 - r = 0$, $2x_2 = 0$, 从而

$$x_1 = \frac{2 \pm \sqrt{4 + 8r}}{4} = \frac{1 \pm \sqrt{1 + 2r}}{2}, \quad x_2 = 0$$

因为 $r > 0$, 若根号前取负号, 将使 $x_1 < 0$, 所以根号前只能取正号。于是

$$x_1 = \frac{1 + \sqrt{1 + 2r}}{2}$$

当 $r \rightarrow 0$ 时, 得 $\mathbf{x}^* = (1, 0)^T$ 为最优解, 而最优值为 $f^* = 1$ 。

下面来讨论 SUMT 内点法的收敛性。 ■

4.3.3 收敛性分析

首先介绍两个引理, 其证明与 4.2 中引理 1, 2 的证明类似, 留给读者练习。

引理 1 设 $r_{k+1} < r_k$, 对于由 SUMT 内点法产生的序列 $\{\mathbf{x}^{(k)}\}$, 下列不等式成立:

$$(1) F(\mathbf{x}^{(k+1)}, r_{k+1}) \leq F(\mathbf{x}^{(k)}, r_k)$$

$$(2) B(\mathbf{x}^{(k+1)}) \geq B(\mathbf{x}^{(k)})$$

$$(3) f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$$

引理 2 设 \mathbf{x}^* 是问题(3-1)的一个最优解, 其中 S 由(3-2)式确定, 则对每一个 k , 有

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^{(k)}) \leq F(\mathbf{x}^{(k)}, r_k)$$

定理 3.1 设 $f(\mathbf{x}), g_i(\mathbf{x}) \in C$ ($i = 1, 2, \dots, m$), 序列 $\{\mathbf{x}^{(k)}\}$

是由 SUMT 内点法产生的, 则它的任一个极限点都是问题(3-1)的一个最优解, 其中 S 由(3-2) 式确定。

证明 设 $\{\mathbf{x}^{(k_i)}\}$ 是 $\{\mathbf{x}^{(k)}\}$ 的任一个收敛子序列, 其极限为 $\bar{\mathbf{x}}$, 由于 $f(\mathbf{x}) \in C$, 所以

$$\lim_{i \rightarrow \infty} f(\mathbf{x}^{(k_i)}) = f(\bar{\mathbf{x}})$$

令 $f^* = f(\mathbf{x}^*)$ 为问题(3-1) 的最优值, 由引理 1.2 知: $\{F(\mathbf{x}^{(k)}, r_k)\}$ 是一个非增且有下界 f^* 的序列, 所以 $\lim_{k \rightarrow \infty} F(\mathbf{x}^{(k)}, r_k) = F^* \geq f^*$ 。由于 $f(\mathbf{x}) \in C$, 所以对任给的小正数 ε , 存在 $\delta > 0$, 在 \mathbf{x}^* 点的小邻域 $N(\mathbf{x}^*, \delta)$ 内, 总可以找到 S 的一个内点 $\tilde{\mathbf{x}} \in \text{int}S \cap N(\mathbf{x}^*, \delta)$, 使得 $f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) < \varepsilon/2$ 。显然

$$F(\mathbf{x}^{(k)}, r_k) \leq f(\tilde{\mathbf{x}}) + r_k B(\tilde{\mathbf{x}})$$

$$F(\mathbf{x}^{(k)}, r_k) - f(\mathbf{x}^*) \leq f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) + r_k B(\tilde{\mathbf{x}})$$

因为 $\lim_{k \rightarrow \infty} r_k = 0$, 所以对上面给定的 $\varepsilon > 0$, 存在 k_0 , 当 $k > k_0$ 时, 总有 $r_k B(\tilde{\mathbf{x}}) < \varepsilon/2$, 于是可得

$$F(\mathbf{x}^{(k)}, r_k) - f(\mathbf{x}^*) < \varepsilon/2 + \varepsilon/2 = \varepsilon$$

由 ε 的任意性, 可得 $F^* = f^*$, 又因为

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^{(k_i)}) \leq F(\mathbf{x}^{(k_i)}, r_{k_i})$$

令 $i \rightarrow \infty$, 即得

$$f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}) \leq F^*$$

故综合上述即得

$$f(\mathbf{x}^*) = f(\bar{\mathbf{x}})$$

这就说明 $\bar{\mathbf{x}}$ 是问题(3-1) 的一个最优解。 ■

4.3.4 混合罚函数法

前面介绍的 SUMT 外点法和内点法, 它们的共同优点是:

① 方法简单, 程序易于编制; ② 对目标函数和约束函数要求不高, 适用范围较广。因此受到工程技术人员的欢迎。但也存在一些

缺点: ① 为了求解一个约束优化问题, 需要求解一系列的无约束优化问题, 计算量大, 而且参数 M_k 和 r_k 的选取对方法的收敛速度影响比较大; ② 在迭代过程中, 外点法中的参数 M_k 不断增大, 内点法中的参数 r_k 不断减小, 使得求解无约束极小问题变得十分困难; ③ 用外点法求得的近似解 $x^{(k)}$ 往往不是可行解, 只能近似地满足约束条件, 在某些实际问题中, 这样的结果是不能用的; 而内点法中, 要求初始点 $x^{(0)}$ 位于可行域的内部, 有时这是困难的, 而且内点法不能求解包含有等式约束的优化问题, 这也是一个很大的限制。为了克服这最后一个缺点, 人们往往将内点法和外点法结合起来使用, 当初始点 $x^{(0)}$ 给定以后, 对 $x^{(0)}$ 满足的那些不等式约束, 按内点法来构造碰壁项 $B(x)$, 对 $x^{(0)}$ 不满足的那些不等式约束和等式约束, 按外点法构造惩罚项 $p(x)$, 即取混合罚函数为

$$F(x, r_k) = f(x) + r_k B(x) + \frac{1}{r_k} p(x) \quad (3-8)$$

其中

$$B(x) = \sum_{i \in I_1} g_i^+(x), \quad p(x) = \sum_{i \in I_2} g_i^+(x) + \sum_{j=1}^p (h_j(x))^2$$

$$I_1 = \{i | g_i(x^{(k-1)}) < 0, i \in I\}, \quad I_2 = \{i | g_i(x^{(k-1)}) \geq 0, i \in I\}$$

$$I = \{1, 2, \dots, m\} = I_1 \cup I_2$$

$$r_0 > r_1 > r_2 > \dots > r_k > r_{k+1} > \dots$$

且

$$\lim_{k \rightarrow \infty} r_k = 0$$

$g_i^+(x)$ 可按外点法和内点法中所介绍的方法确定。

Lootsma 建议取

$$r_0 = \max\{0.01, \frac{|v^*|}{100}\}, \quad r_{k+1} = r_k c$$

其中 $0 < c < 1$, 常取 $c \in [0.1, 0.25]$, v^* 为约束优化问题

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & h_j(\mathbf{x}) = 0, j = 1, 2, \dots, p \\ & \mathbf{x} \in \mathbf{R}^n \end{cases} \quad (3-9)$$

的最优值 $f(\mathbf{x}^*)$ 的一个估计值。

上述方法称为混合罚函数法,其计算步骤如下:

(1) 选取初始点 $\mathbf{x}^{(0)}$, 参数 $r_0 > 0$, $0 < c < 1$, 计算精度 $\varepsilon_i > 0$, $i = 1, 2$, 令 $k = 1$;

(2) 以 $\mathbf{x}^{(k-1)}$ 为初始点, 求解无约束优化问题 $\min F(\mathbf{x}, r_k)$, 得最优解 $\mathbf{x}^{(k)} = \mathbf{x}(r_k)$;

(3) 若 $\mathbf{x}^{(k)}$ 满足结束准则, 那么迭代结束, 取 $\mathbf{x}^{(k)}$ 为近似最优解; 否则令 $r_{k+1} = cr_k$, $k = k + 1$, 返回(2)。

关于结束准则, 有以下几种可供选用:

(1) $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \varepsilon_1$, $|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k-1)})| < \varepsilon_2$;

(2) $r_k B(\mathbf{x}^{(k)}) < \varepsilon_1$, $\frac{1}{r_k} p(\mathbf{x}^{(k)}) < \varepsilon_2$;

(3) $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty < \varepsilon_1 \|\mathbf{x}^{(k)}\|_\infty + \varepsilon_2$, 且
 $|F(\mathbf{x}^{(k)}, r_k) - F(\mathbf{x}^{(k-1)}, r_{k-1})| < \varepsilon_1 |F(\mathbf{x}^{(k)}, r_k)| + \varepsilon_2$

其中, $\|\mathbf{x}\|_\infty = \max_i |x_i|$ 。

常取 $\varepsilon_1 = \varepsilon_2 = 10^{-5}$ 。

* 4.4 可行方向法

最早的一个可行方向法是在 1960 年由 G. Zoutendijk 提出的。现在, 可行方向法已发展成为求解约束优化问题的一类重要的方法, 其基本思想是: 在给定一个可行点 $\mathbf{x}^{(k)}$ 之后, 用某种方法确定一个改进的可行方向 \mathbf{d}_k , 然后沿方向 \mathbf{d}_k , 求解一个有约束的线搜索问题, 得极小点 λ_k , 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}_k$, 如果 $\mathbf{x}^{(k+1)}$ 仍不

是最优解, 则重复上述步骤。各种不同的可行方向法的主要区别在于: 选择可行方向 d_k 的策略不同, 大体上可以分为三类:

(1) 用求解一个线性规划问题来确定 d_k , 如 Zoutendijk 方法, Frank-Wolfe 方法和 Topkis-Veinott 方法等。

(2) 利用投影矩阵来直接构造一个改进的可行方向 d_k , 如 Rosen 的梯度投影法和 Rosen-Polak 方法等。

(3) 利用既约梯度, 直接构造一个改进的可行方向 d_k , 如 Wolfe 的既约梯度法及其各种改进, 凸单纯形法等。

本节主要介绍用解线性规划来确定改进的可行方向这类方法, 其余两类可行方向法将在 4.5 和 4.6 中介绍, 以下说到可行方向法时一般是指第一类可行方向法。

4.4.1 可行方向与改进方向

考察非线性规划问题

$$\begin{cases} \min & f(x) \\ \text{s. t.} & x \in S \subset \mathbf{R}^n \end{cases} \quad (4-1)$$

其中 S 为非空的可行集。

定义 非零向量 d 称为在点 $x \in S$ 的一个可行方向, 如果存在一个数 $\delta > 0$, 使得对 $\forall \lambda \in (0, \delta)$, 都有 $x + \lambda d \in S$ 。

$d \neq 0$ 称为在点 $x \in S$ 的一个改进的可行方向, 如果存在 $\delta > 0$, 使得对 $\forall \lambda \in (0, \delta)$, 都有 $f(x + \lambda d) < f(x)$, 且 $x + \lambda d \in S$ 。

定理 4.1 设 x 是问题

$$\begin{cases} \min & f(x) \\ \text{s. t.} & Ax \leq b, Ex = e \\ & x \in \mathbf{R}^n \end{cases} \quad (4-2)$$

的一个可行解, 假定 $A_1 x = b_1$, $A_2 x < b_2$, 其中 $A^T = (A_1^T, A_2^T)$, $b^T = (b_1^T, b_2^T)$, A 是 $m \times n$ 矩阵, E 是 $l \times n$ 矩阵, $b \in \mathbf{R}^m$, $e \in \mathbf{R}^l$,

则一个非零向量 d 是在 x 点的可行方向, 当且仅当 $A_1 d \leq 0$, $Ed = 0$; 如果 $\nabla f(x)^T d < 0$, 则 d 是一改进方向。其中 $f(x)$ 在 x 点可微。

证明 1° 设 $A_1 d \leq 0$, $Ed = 0$, 要证 d 为 $x \in S$ 的可行方向。这里 S 表示(4-2)的可行集。

因为 $A_1 x = b_1$, $A_1 d \leq 0$, $Ex = e$, $Ed = 0$, 所以对任何 $\lambda > 0$, 都有 $A_1(x + \lambda d) = b_1 + \lambda A_1 d \leq b_1$, $E(x + \lambda d) = Ex + \lambda Ed = e$, 又因 $A_2 x < b_2$, 所以存在一个 $\delta > 0$, 使得对任意 $\lambda \in (0, \delta)$ 时, 有

$$A_2(x + \lambda d) = A_2 x + \lambda A_2 d \leq b_2$$

因此对于任意 $\lambda \in (0, \delta)$ 都有 $x + \lambda d \in S$, 即 d 为 x 点的可行方向。

2° 设 $d \neq 0$ 为点 $x \in S$ 的可行方向, 要证 $A_1 d \leq 0$, $Ed = 0$ 。因为

$$A_1(x + \lambda d) = A_1 x + \lambda A_1 d = b_1 + \lambda A_1 d \leq b_1$$

$$\lambda \in (0, \delta), \delta > 0$$

所以 $A_1 d \leq 0$

因为 $E(x + \lambda d) = Ex + \lambda Ed = e + \lambda Ed = e$

$$\lambda \in (0, \delta), \delta > 0$$

所以 $Ed = 0$

综合上述, 定理 4.1 的第一部分得证。

3° 设 $\nabla f(x)^T d < 0$, 要证 d 为改进方向。

因为 $f(x)$ 在 $x \in S$ 点可微, 所以

$$f(x + \lambda d) = f(x) + \lambda \nabla f(x)^T d + \lambda \|d\| \alpha(x; \lambda d)$$

其中, 当 $\lambda \rightarrow 0$ 时, $\alpha(x; \lambda d) \rightarrow 0$ 。

因为 $\nabla f(x)^T d < 0$, 故存在 $\delta > 0$, 使得当 $\lambda \in (0, \delta)$ 时, 有 $f(x + \lambda d) < f(x)$ 成立, 即 d 为改进方向。■

定理 4.2 设 x 是问题

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & \mathbf{x} \in \mathbf{R}^n \end{cases} \quad (4-3)$$

的一个可行解, 令 $S = \{\mathbf{x} | \mathbf{x} \in \mathbf{R}^n, g_i(\mathbf{x}) \leq 0, i = 1, \dots, m\}$, $I = \{i | g_i(\mathbf{x}) = 0\}$, 即 I 是 $\mathbf{x} \in S$ 点紧约束的指标集, 设 f 和 $g_i (i \in I)$ 在 \mathbf{x} 点可微, $g_i (i \notin I)$ 在 \mathbf{x} 点连续, 如果 $\nabla f(\mathbf{x})^T \mathbf{d} < 0$, $\nabla g_i(\mathbf{x})^T \mathbf{d} < 0 (i \in I)$, 则 \mathbf{d} 是一改进的可行方向。

证明 因为对 $i \in I$, $g_i(\mathbf{x}) < 0$, 且 $g_i(\mathbf{x})$ 在 \mathbf{x} 点连续, 所以存在 $\delta_1 > 0$, 使得对任意 $\lambda \in (0, \delta_1)$, 有 $g_i(\mathbf{x} + \lambda \mathbf{d}) \leq 0$; 因为对 $i \in I$, g_i 在 \mathbf{x} 点可微, 所以

$$g_i(\mathbf{x} + \lambda \mathbf{d}) = g_i(\mathbf{x}) + \lambda \nabla g_i(\mathbf{x})^T \mathbf{d} + \lambda \|\mathbf{d}\| \alpha(\mathbf{x}; \lambda \mathbf{d})$$

其中, 当 $\lambda \rightarrow 0$ 时, $\alpha(\mathbf{x}; \lambda \mathbf{d}) \rightarrow 0$, 而 $\nabla g_i(\mathbf{x})^T \mathbf{d} < 0$, 因此, 存在 $\delta_2 > 0$, 使得对任意 $\lambda \in (0, \delta_2)$, 有

$$g_i(\mathbf{x} + \lambda \mathbf{d}) \leq g_i(\mathbf{x}) = 0$$

用类似的办法, 可以证明: 存在 $\delta_3 > 0$, 使得对任意 $\lambda \in (0, \delta_3)$, 有 $f(\mathbf{x} + \lambda \mathbf{d}) < f(\mathbf{x})$ 成立。

综合上述, 并令 $\delta = \min_{1 \leq i \leq 3} \{\delta_i\}$, 即知: 当 $\lambda \in (0, \delta)$ 时, 总有

$$g_i(\mathbf{x} + \lambda \mathbf{d}) \leq 0, i = 1, 2, \dots, m$$

$$f(\mathbf{x} + \lambda \mathbf{d}) < f(\mathbf{x})$$

所以 \mathbf{d} 是一改进的可行方向。 ■

上述的两个定理, 指出了改进可行方向的特征。下面再举一个例子来加以说明。

【例 4.1】 考察如下的问题:

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 2)^2 \\ \text{s. t.} & -x_1 + 2x_2 \leq 4 \\ & 3x_1 + 2x_2 \leq 12 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \end{cases}$$

解 令 $x = (2, 3)^T$, 注意到在点 x , 前两个约束是紧约束, 所以

$$I = \{1, 2\}, \quad A_1 = \begin{pmatrix} -1 & 2 \\ 3 & 2 \end{pmatrix}$$

因而 $d \neq 0$ 要成为一个在点 x 处的可行方向, 当且仅当 $A_1 d \leq 0$, 即

$$-d_1 + 2d_2 \leq 0$$

$$3d_1 + 2d_2 \leq 0$$

满足上面两个不等式的方向 d 的集合为 $D = \{d | A_1 d \leq 0\}$,

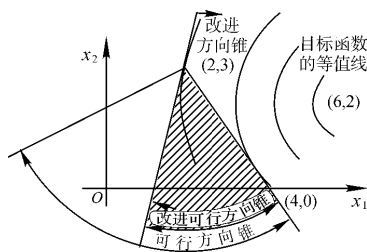


图 4-6

其所形成的可行方向锥在图 4-6 中指出。而满足条件

$$\nabla f(x)^T d = (-8, 2) \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -8d_1 + 2d_2 < 0$$

的方向 d 是一改进方向, 其集合 $F = \{d | \nabla f(x)^T d < 0\}$ 形成改进方向锥, 它们的交集 $D \cap F$ 就是改进的可行方向锥, 如图 4-6 中所指出的。

4.4.2 Zoutendijk 的可行方向法

对可行方向法来说, 关键的一步是如何具体地生成改进的可

行方向, 对于线性约束问题(4-2), 由定理 4.1 得知: 可由求解如下的规划问题 P_1 , P_2 或 P_3 来得到 d 。

设 x 为(4-2)的可行点。

$$\begin{aligned} \text{问题 } P_1 & \begin{cases} \min & z = \nabla f(x)^T d \\ \text{s. t.} & A_1 d \leq 0 \\ & Ed = 0 \\ & -1 \leq d_j \leq 1, j = 1, \dots, n \end{cases} \\ \text{问题 } P_2 & \begin{cases} \min & z = \nabla f(x)^T d \\ \text{s. t.} & A_1 d \leq 0 \\ & Ed = 0 \\ & d^T d \leq 1. \end{cases} \\ \text{问题 } P_3 & \begin{cases} \min & z = \nabla f(x)^T d \\ \text{s. t.} & A_1 d \leq 0 \\ & Ed = 0 \\ & \nabla f(x)^T d \geq -1 \end{cases} \end{aligned}$$

其中 $d = (d_1, d_2, \dots, d_n)^T$ 。最后一个约束是为了保证解有界的规格化约束。

对于变量 d_1, d_2, \dots, d_n 来说, 问题 P_2 的约束是二次的, 但可设法将其化简。因为对上述的每一个问题, $d = 0$ 都是可行解, 且目标函数值 $z = 0$, 所以上述问题的最优目标函数值 $z^* \leq 0$, 若 $z^* < 0$, 由定理 4.1 可知: $d^* \neq 0$ 是一改进的可行方向; 而 $z^* = 0$, 当且仅当 x 是 K-T 点^[11]。

1. 线性约束的 Zoutendijk 方法的计算步骤

1° 求(4-2)的一初始可行解 $x^{(1)}$, 令 $k = 1$, 转 2°。

2° 对于可行点 $x^{(k)}$, 设 $A_1 x^{(k)} = b_1$, $A_2 x^{(k)} < b_2$, $A^T = (A_1^T, A_2^T)$, $b^T = (b_1^T, b_2^T)$, 求解问题 P_1 (或 P_2, P_3), 得最优解 d_k , 如果 $\nabla f(x^{(k)})^T d_k = 0$, 计算结束, $x^{(k)}$ 是 K-T 点; 否则转 3°。

3° 求解线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(k)} + \lambda \mathbf{d}_k) \\ \text{s. t.} & 0 \leq \lambda \leq \lambda_{\max} \end{cases} \quad (4-4)$$

其中

$$\lambda_{\max} = \begin{cases} \min \{ \hat{b}_i / \hat{d}_i \mid \hat{d}_i > 0 \}, & \text{若 } \hat{\mathbf{d}} \not\leq \mathbf{0} \\ +\infty, & \text{若 } \hat{\mathbf{d}} \leq \mathbf{0} \end{cases} \quad (4-5)$$

$$\hat{\mathbf{b}} = \mathbf{b}_2 - \mathbf{A}_2 \mathbf{x}^{(k)}, \quad \hat{\mathbf{d}} = \mathbf{A}_2 \mathbf{d}_k \quad (4-6)$$

设 λ_k 为(4-4)的最优解, 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}_k$, $k = k + 1$, 返回 2°。

注意:

1° 由于计算总会有误差, 所以一般来说, $\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_k = 0$ 不可能准确地成立, 可取一个允许误差 $\varepsilon_1 > 0$, 若 $|\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}_k| < \varepsilon_1$, 计算即可结束。

2° 若可行点 $\mathbf{x}^{(k)}$ 为内点, 即 $\mathbf{A}\mathbf{x}^{(k)} < \mathbf{b}$, 这时不必去求解线性规划, 可取 $\mathbf{d}_k = -\nabla f(\mathbf{x}^{(k)})$, 若 $|\mathbf{z}_k| < \varepsilon_2$, 计算结束, 其中 $\varepsilon_2 > 0$ 为允许误差。

2. 非线性不等式约束的 Zoutendijk 方法的计算步骤

这时改进的可行方向可通过求解如下的线性规划问题来得到:

$$\begin{cases} \min & z \\ \text{s. t.} & \nabla f(\mathbf{x})^T \mathbf{d} - z \leq 0 \\ & \nabla g_i(\mathbf{x})^T \mathbf{d} - z \leq 0, \quad i \in I(\mathbf{x}) \\ & -1 \leq d_j \leq 1, \quad j = 1, 2, \dots, n \end{cases} \quad (4-7)$$

其中 \mathbf{x} 为(4-3)的可行点, $I(\mathbf{x}) = \{i \mid g_i(\mathbf{x}) = 0\}$ 。

令 $(\bar{z}, \bar{\mathbf{d}})$ 为(4-7)的最优解, 易见 $\bar{z} \leq 0$ 。若 $\bar{z} < 0$, 由定理 4.2 知 $\bar{\mathbf{d}}$ 是一改进的可行方向; 否则可以证明, 在一定条件下, \mathbf{x} 是 K-T 点的充要条件为 $\bar{z} = 0$ 。

1° 选取允许误差 $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, 求一初始可行点 $\mathbf{x}^{(1)}$, 令 $k = 1$, 转 2°。

2° 确定指标集 $I(\mathbf{x}^{(k)}) = \{i | g_i(\mathbf{x}^{(k)}) = 0\}$ 。

3° 若 $I(\mathbf{x}^{(k)}) = \emptyset$ (空集), 且 $\|\nabla f(\mathbf{x}^{(k)})\| \leq \epsilon_1$, 计算结束, 取 $\mathbf{x}^* = \mathbf{x}^{(k)}$; 若 $I(\mathbf{x}^{(k)}) = \emptyset$, 但 $\|\nabla f(\mathbf{x}^{(k)})\| > \epsilon_1$, 令 $\mathbf{d}_k = -\nabla f(\mathbf{x}^{(k)})$, 转 6°; 若 $I(\mathbf{x}^{(k)}) \neq \emptyset$, 转 4°;

4° 在 (4-7) 中, 令 $\mathbf{x} = \mathbf{x}^{(k)}$, 求出线性规划问题 (4-7) 的最优解 $(\bar{z}_k, \bar{\mathbf{d}}_k)$ 。

5° 若 $|\bar{z}_k| < \epsilon_2$, 计算结束, 取 $\mathbf{x}^* = \mathbf{x}^{(k)}$; 否则令 $\mathbf{d}_k = \bar{\mathbf{d}}_k$, 转 6°。

6° 求出线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(k)}) + \lambda \mathbf{d}_k \\ \text{s. t.} & 0 \leq \lambda \leq \lambda_{\max} \end{cases}$$

的最优解 λ_k , 其中 $\lambda_{\max} = \max\{\lambda | \mathbf{x}^{(k)} + \lambda \mathbf{d}_k \in S\}$, 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}_k$, $k = k + 1$, 返回 2°。

3. 算法应用举例

【例 4.2】 用 Zoutendijk 方法求解如下问题:

$$\begin{cases} \min & f(\mathbf{x}) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ \text{s. t.} & x_1 + x_2 \leq 2 \\ & x_1 + 5x_2 \leq 5 \\ & -x_1 \leq 0, -x_2 \leq 0 \end{cases}$$

解 取初始点 $\mathbf{x}^{(1)} = (0, 0)^T$, $\nabla f(\mathbf{x}) = \begin{pmatrix} 4x_1 - 2x_2 - 4 \\ 4x_2 - 2x_1 - 6 \end{pmatrix}$

第一次迭代

1° 搜索方向 因为

$$\nabla f(\mathbf{x}^{(1)}) = \begin{pmatrix} -4 \\ -6 \end{pmatrix}, I(\mathbf{x}^{(1)}) = \{3, 4\}$$

所以确定改进可行方向 \mathbf{d} 的线性规划问题为

$$\begin{cases} \min & z = -4d_1 - 6d_2 \\ \text{s. t.} & 0 \leq d_1 \leq 1, 0 \leq d_2 \leq 1 \end{cases}$$

其最优解为 $\mathbf{d}^{(1)} = (1, 1)^T$, 最优值为 $z_1 = -10$ 。

2° 线搜索 因为

$$\mathbf{x}^{(1)} + \lambda \mathbf{d}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda \\ \lambda \end{pmatrix}$$

故 $f(\mathbf{x}^{(1)} + \lambda \mathbf{d}^{(1)}) = -10\lambda + 2\lambda^2$

由式(4-5)和(4-6)得 $\lambda_{\max} = 5/6$, 求解线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(1)} + \lambda \mathbf{d}^{(1)}) = -10\lambda + 2\lambda^2 \\ \text{s. t.} & 0 \leq \lambda \leq 5/6 \end{cases}$$

得 $\lambda_1 = 5/6$, 故 $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{d}^{(1)} = (5/6, 5/6)^T$ 。

第二次迭代

1° 搜索方向 因为

$$\nabla f(\mathbf{x}^{(2)}) = (-7/3, -13/3)^T, I(\mathbf{x}^{(2)}) = \{2\}$$

所以 $\mathbf{d}^{(2)}$ 由求解下面的问题确定:

$$\begin{cases} \min & z = -7d_1/3 - 13d_2/3 \\ \text{s. t.} & d_1 + 5d_2 \leq 0, \\ & -1 \leq d_i \leq 1, i = 1, 2 \end{cases}$$

其最优解为 $\mathbf{d}^{(2)} = (1, -1/5)^T$, 最优值为 $z_2 = -22/15$ 。

2° 线搜索 因为

$$\mathbf{x}^{(2)} + \lambda \mathbf{d}^{(2)} = (5/6 + \lambda, 5/6 - \lambda/5)^T$$

$$f(\mathbf{x}^{(2)} + \lambda \mathbf{d}^{(2)}) = -125/8 - 22\lambda/15 + 62\lambda^2/25, \lambda_{\max} = 5/12$$

所以线搜索问题为

$$\begin{cases} \min & f = -125/8 - 22\lambda/15 + 62\lambda^2/25 \\ \text{s. t.} & 0 \leq \lambda \leq 5/12 \end{cases}$$

其最优解为 $\lambda_2 = 55/186$, 因此 $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \lambda_2 \mathbf{d}^{(2)} = (35/31, 24/31)^T$ 。

第三次迭代 因为

$$\nabla f(\mathbf{x}^{(3)}) = (-32/31, -160/31)^T, I(\mathbf{x}^{(3)}) = \{2\}$$

所以确定搜索方向的问题为

$$\begin{cases} \min & z = -32d_1/31 - 160d_2/31 \\ \text{s. t.} & d_1 + 5d_2 \leq 0 \\ & -1 \leq d_i \leq 1, i = 1, 2 \end{cases}$$

其最优解为 $d^{(3)} = (1, -1/5)^T$, 最优值为 $z_3 = 0$ 。故 $x^{(3)}$ 为 K-T 点。可以验证此例中 $x^* = x^{(3)}$ 确是最优解, $f(x^*) = -7.16$ 。

上述 Zoutendijk 的可行方向法可能出现“锯齿现象”, 使得收敛速度很慢, 甚至不收敛于 K-T 点。1972 年, Wolfe 曾举出一个例子, 说明由上述的 Zoutendijk 算法产生的点列 $\{x^{(k)}\}$ 不收敛于该问题的 K-T 点(参看文献 11)。因此, 早在 1967 年, Topkis 和 Veinott 提出了一种修正的可行方向法。可以证明: 在一定条件下, 由这种算法产生的点列 $\{x^{(k)}\}$ 是收敛于 K-T 点的, 参见文献 11。

4.5 梯度投影法

本节介绍 Rosen 的梯度投影法, 这是他在 1960 年针对线性约束的优化问题首先提出来的, 1961 年 Rosen 又将他的方法推广到处理非线性约束的情况, 后来这类方法又得到了进一步的发展, 成为求解非线性规划问题的一类重要的方法。

4.5.1 理论基础

考虑线性约束的优化问题

$$\begin{cases} \min & f(x) \\ \text{s. t.} & Ax \leq b \\ & Ex = e \end{cases} \quad (5-1)$$

其中 A 是 $m \times n$ 矩阵, E 是 $l \times n$ 矩阵, $b \in \mathbf{R}^m$, $e \in \mathbf{R}^l$, $x \in \mathbf{R}^n$, $f: \mathbf{R}^n \rightarrow \mathbf{R}$, $f \in C^1$ 。用 Ω 表示(5-1)的可行域。

梯度投影法的基本思想是当迭代点 $x^{(k)}$ 在可行域 Ω 的内部时, 取 $d = -\nabla f(x^{(k)})$ 为迭代方向; 当点 $x^{(k)}$ 在 Ω 的边界上时, 取

— $\nabla f(x^{(k)})$ 在这些边界面的交集上的投影为迭代的方向。

为了具体说明上述方法,首先引入投影矩阵的概念,并研究它的一些性质。

定义 5.1 称 $n \times n$ 矩阵 P 为投影矩阵,如果

$$P = P^T, \text{ 且 } PP = P$$

引理 1 设 P 为 $n \times n$ 矩阵

(1) 若 P 为投影矩阵,则 P 是半正定的;

(2) P 是投影矩阵,当且仅当 $I - P$ 是投影矩阵,其中 I 为 n 阶单位阵;

(3) 设 P 是投影矩阵,令 $Q = I - P$, 则

$$L = \{Px | x \in \mathbb{R}^n\} \text{ 与 } L^\perp = \{Qx | x \in \mathbb{R}^n\}$$

为互相正交的线性子空间,并且任一点 $x \in \mathbb{R}^n$ 可惟一地表示为

$$x = p + q, p \in L, q \in L^\perp$$

证明 (1) 设 $x \in \mathbb{R}^n$ 是任一向量,因为 P 是投影矩阵,所以

$$x^T Px = x^T PPx = x^T P^T Px = \|Px\|^2 \geq 0$$

因此 P 是半正定的。

(2) 设 P 为投影矩阵,则

$$(I - P)^T = I - P^T = I - P$$

$$(I - P)(I - P) = I - P - P + PP = I - P$$

因此,由投影矩阵的定义可知 $I - P$ 亦为投影矩阵。

现在设 $I - P$ 为投影矩阵,因为 $P = I - (I - P)$, 所以 P 亦为投影矩阵。

(3) 因为对于任意的 $x, y \in \mathbb{R}^n$, 有 $\alpha x + \beta y \in \mathbb{R}^n$, 且 $\alpha Px + \beta Py = P(\alpha x + \beta y) \in L$, 其中 α, β 为实数, 所以 L 是线性子空间, 同理可知 L^\perp 亦为线性子空间。由 $P^T Q = P^T (I - P) = P - P = O$, 可知 $(Px)^T (Qy) = x^T (P^T Q) y = 0$, 所以 L 与 L^\perp 正交。

因为对任意的 $x \in \mathbb{R}^n$, 有

$$x = Ix = (P + Q)x = Px + Qx = p + q$$

其中 $p \in L, q \in L^\perp$ 。下面再证明此表示式是惟一的。

设有 $p, p' \in L, q, q' \in L^\perp$, 使

$$x = p + q, x = p' + q'$$

则有

$$p - p' = q' - q$$

且

$$p - p' \in L, q' - q \in L^\perp$$

由于 L 与 L^\perp 正交, 所以 $p - p' = q' - q = 0$, 即

$$p = p', \quad q = q' \quad \blacksquare$$

引理 2 设 x 为问题(5-1)的一个可行解, 且使 $A_1 x = b_1, A_2 x < b_2$, 而 $A^T = (A_1^T, A_2^T), b^T = (b_1^T, b_2^T)$, 设 $f(x)$ 在 x 点可微, $M^T = (A_1^T, E^T)$ 满秩, 则

(1) $P = I - M^T(MM^T)^{-1}M$ 为投影矩阵;

(2) 若 $P \nabla f(x) \neq 0$, 则 $d = -P \nabla f(x)$ 是 x 点的一个改进的可行方向。

证明 (1) 因为 $Q = M^T(MM^T)^{-1}M$ 满足

$$Q^T = M^T(MM^T)^{-1}M = Q$$

$$QQ = (M^T(MM^T)^{-1}M)(M^T(MM^T)^{-1}M) = Q$$

所以, Q 为投影矩阵, 由引理 1 知 $P = I - Q$ 亦为投影矩阵。

$$\begin{aligned} (2) \text{ 因为 } \quad \nabla f(x)^T d &= -\nabla f(x)^T P \nabla f(x) \\ &= -\nabla f(x)^T P^T P \nabla f(x) \\ &= -\|P \nabla f(x)\|^2 < 0 \end{aligned}$$

所以 d 是一改进方向。又因为

$$\begin{aligned} Md &= -MP \nabla f(x) \\ &= -[M - MM^T(MM^T)^{-1}M] \nabla f(x) \\ &= -(M - M) \nabla f(x) = 0 \end{aligned}$$

即 $A_1 d = 0, E d = 0$ 。所以当任意的 $\lambda > 0$ 时, 有 $A_1(x + \lambda d) = b_1, E(x + \lambda d) = e$, 当 $\lambda > 0$ 足够小时, 总有 $A_2(x + \lambda d) < b_2$, 因此 d 是 x 点的一个改进的可行方向。 \blacksquare

由上述可见: 若 $P \nabla f(x) \neq \mathbf{0}$, 则 $d = -P \nabla f(x)$ 是一个改进的可行方向, 当 $P \nabla f(x) = \mathbf{0}$ 时, 又怎么办呢? 我们有下面的定理。

定理 5.1 设 x 是问题(5-1) 的一个可行解, 且使 $A_1 x = b_1$, $A_2 x < b_2$, 而 $A^T = (A_1^T, A_2^T)$, $b^T = (b_1^T, b_2^T)$, $M^T = (A_1^T, E^T)$ 满秩, 令

$$P = I - M^T(MM^T)^{-1}M$$

$$w = - (MM^T)^{-1}M \nabla f(x), w^T = (u^T, v^T)$$

设 $P \nabla f(x) = \mathbf{0}$,

(1) 若 $u \geq \mathbf{0}$, 则 x 是一个 K-T 点;

(2) 若 $u \not\geq \mathbf{0}$, 令 u_j 是 u 的一个负分量, $\hat{M}^T = (\hat{A}_1^T, E^T)$

其中 \hat{A}_1 是由 A_1 中去掉第 j 行后得到的矩阵, 令

$$\hat{P} = I - \hat{M}^T(\hat{M}\hat{M}^T)^{-1}\hat{M}, \quad d = -\hat{P} \nabla f(x)$$

则 d 是一个改进的可行方向。

证明 (1) 因为

$$\begin{aligned} P \nabla f(x) &= [I - M^T(MM^T)^{-1}M] \nabla f(x) \\ &= \nabla f(x) + M^T w \\ &= \nabla f(x) + A_1^T u + E^T v = \mathbf{0} \end{aligned} \quad (5-2)$$

所以, 若 $u \geq \mathbf{0}$, 则由 K-T 条件知 x 为一个 K-T 点。

(2) 首先证明: $\hat{P} \nabla f(x) \neq \mathbf{0}$ (用反证法)。

假设 $\hat{P} \nabla f(x) = \mathbf{0}$ 。并令

$$\hat{w} = - (\hat{M}\hat{M}^T)^{-1}\hat{M} \nabla f(x)$$

$$\begin{aligned} \text{可得} \quad \hat{P} \nabla f(x) &= [I - \hat{M}^T(\hat{M}\hat{M}^T)^{-1}\hat{M}] \nabla f(x) \\ &= \nabla f(x) + \hat{M}^T \hat{w} = \mathbf{0} \end{aligned} \quad (5-3)$$

由于 $A_1^T u + E^T v$ 可以写成 $\hat{M}^T \bar{w} + u_j a_j^T$, 于是由(5-2) 式可得

$$\nabla f(x) + \hat{M}^T \bar{w} + u_j a_j^T = \mathbf{0} \quad (5-4)$$

其中 a_j 是 A_1 的第 j 行。

将上面两式相减可得

$$\hat{M}^T(\hat{\mathbf{w}} - \bar{\mathbf{w}}) - u_j \mathbf{a}_j^T = 0$$

而 $u_j < 0$, 所以上式与 M 满秩的假设相矛盾, 这就证明了:
 $\hat{P} \nabla f(\mathbf{x}) \neq 0$ 。

易证 \hat{P} 为投影矩阵, 因此由

$$\nabla f(\mathbf{x})^T \mathbf{d} = - \nabla f(\mathbf{x})^T \hat{P} \nabla f(\mathbf{x}) = - \|\hat{P} \nabla f(\mathbf{x})\|^2 < 0$$

可知, $\mathbf{d} = -\hat{P} \nabla f(\mathbf{x})$ 是一个改进方向。

下面证明 \mathbf{d} 是一个可行方向。

注意到: $\hat{M}\hat{P} = \mathbf{0}$, 因此

$$\begin{pmatrix} \hat{A}_1 \\ E \end{pmatrix} \mathbf{d} = \hat{M} \mathbf{d} = -\hat{M}\hat{P} \nabla f(\mathbf{x}) = \mathbf{0} \quad (5-5)$$

由 4.5 节的定理 5.1 知: 若 $A_1 \mathbf{d} \leq \mathbf{0}$, $E \mathbf{d} = \mathbf{0}$, 则 \mathbf{d} 是可行方向。由 (5-5) 式看出: 要证明 \mathbf{d} 是一个可行方向, 只要证明 $a_j \mathbf{d} \leq 0$ 就够了。

用 $a_j \hat{P}$ 乘 (5-4) 式, 并注意到 $\hat{P} \hat{M}^T = 0$, 就可推出:

$$a_j \hat{P} \nabla f(\mathbf{x}) + a_j \hat{P} (\hat{M}^T \bar{\mathbf{w}} + u_j \mathbf{a}_j^T) = -a_j \mathbf{d} + u_j a_j \hat{P} \mathbf{a}_j^T = 0 \quad (5-6)$$

由引理 1 可知: \hat{P} 是半正定的, 因此 $a_j \hat{P} \mathbf{a}_j^T \geq 0$, 而 $u_j < 0$, 所以 (5-6) 式意味着: $a_j \mathbf{d} \leq 0$ 。 ■

4.5.2 计算步骤

在问题 (5-1) 中, 设

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{pmatrix}$$

$$E = \begin{pmatrix} a_{m+1,1} & a_{m+1,2} & \cdots & a_{m+1,n} \\ a_{m+2,1} & a_{m+2,2} & \cdots & a_{m+2,n} \\ \vdots & \vdots & & \vdots \\ a_{m+l,1} & a_{m+l,2} & \cdots & a_{m+l,n} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_{m+1}^T \\ \mathbf{a}_{m+2}^T \\ \vdots \\ \mathbf{a}_{m+l}^T \end{pmatrix}$$

其中

$$\mathbf{a}_j = (a_{i1}, a_{i2}, \dots, a_{in})^T, \quad i = 1, 2, \dots, m+l$$

令

$$B = \begin{pmatrix} \mathbf{b} \\ \mathbf{e} \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_{m+l} \end{pmatrix}, \quad J_k = J(\mathbf{x}^{(k)}) = \{j | \mathbf{a}_j^T \mathbf{x}^{(k)} = B_j\}$$

J_k 称为在 $\mathbf{x}^{(k)}$ 点的指标集。用 M_k 表示以 $\mathbf{a}_j^T (j \in J_k)$ 为行所组成的矩阵。

1. 选取 $\mathbf{x}^{(1)}$ 为(5-1)的一个可行解, 给定计算精度 $\varepsilon > 0$, 令 $k = 1$ 。

2. 计算 $\nabla f(\mathbf{x}^{(k)})$, $J_k = \{j | \mathbf{a}_j^T \mathbf{x}^{(k)} = B_j\}$, 若 $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, 则 $\mathbf{x}^{(k)}$ 为近似的 K-T 点, 计算结束; 否则

(1) 若 $J_k = \emptyset$ (空集), 令 $P = I$,

(2) 若 $J_k \neq \emptyset$, 令 $P = I - M_k^T (M_k M_k^T)^{-1} M_k$ 。

3. 若 $P \nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$, 令 $d_k = -P \nabla f(\mathbf{x}^{(k)})$, 转 4; 若 $P \nabla f(\mathbf{x}^{(k)}) = \mathbf{0}$, 令

$$\mathbf{w} = - (M_k M_k^T)^{-1} M_k \nabla f(\mathbf{x}^{(k)}) = \begin{pmatrix} u \\ v \end{pmatrix}$$

(1) 若 $u \geq 0$, 则 $\mathbf{x}^{(k)}$ 为 K-T 点, 计算结束,

(2) 若 u 有某个分量 $u_j < 0$, 令 \hat{M}_k 是在 M_k 中去掉与 u_j 对应的第 j 行而得到的矩阵。

令 $\hat{P} = I - \hat{M}_k^T (\hat{M}_k \hat{M}_k^T)^{-1} \hat{M}_k$, $d_k = -\hat{P} \nabla f(\mathbf{x}^{(k)})$, 转 4。

4. 计算

$$\lambda_M = \begin{cases} \min \left\{ \frac{\mathbf{B}_i - \mathbf{a}_i^T \mathbf{x}^{(k)}}{\mathbf{a}_i^T \mathbf{d}_k} \mid i \in J_k, \mathbf{a}_i^T \mathbf{d}_k > 0 \right\} \\ +\infty, \quad \text{若对任意 } i, \mathbf{a}_i^T \mathbf{d}_k \leq 0 \end{cases} \quad (5-7)$$

设 λ_k 是线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(k)} + \lambda \mathbf{d}_k) \\ \text{s. t.} & 0 \leq \lambda \leq \lambda_M \end{cases} \quad (5-8)$$

的解, 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \mathbf{d}_k$, $k = k + 1$, 返回 2。

4.5.3 计算例子

例 用上述的梯度投影法求解如下的优化问题:

$$\begin{cases} \min & f(\mathbf{x}) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ \text{s. t.} & x_1 + x_2 \leq 2 \\ & x_1 + 5x_2 \leq 5 \\ & -x_1 \leq 0, -x_2 \leq 0 \end{cases} \quad (5-9)$$

解 $\nabla f(\mathbf{x}) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6)^T$

因为本题没有等式约束, 所以 $M = A_1$, $\mathbf{w} = \mathbf{u}$ 。

第一次迭代 取 $\mathbf{x}^{(1)} = (0, 0)^T$ 。

1° 确定搜索方向

在 $\mathbf{x}^{(1)}$ 点, $\nabla f(\mathbf{x}^{(1)}) = (-4, -6)^T$, 且只有非负约束是紧约束, 因此

$$M_1 = A_1 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad J_1 = \{3, 4\}$$

$$P = I - A_1^T (A_1 A_1^T)^{-1} A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$P \nabla f(\mathbf{x}^{(1)}) = (0, 0)^T$$

$$\mathbf{u} = - (A_1 A_1^T)^{-1} A_1 \nabla f(\mathbf{x}^{(1)}) = (-4, -6)^T$$

取 $u_4 = -6$, 从 A_1 中去掉与 u_4 对应的行得到

$$\hat{A}_1 = (-1, 0), \quad \hat{P} = I - \hat{A}_1^T (\hat{A}_1 \hat{A}_1^T)^{-1} \hat{A}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$d_1 = -\hat{P} \nabla f(x^{(1)}) = - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

2° 进行线搜索 因为

$$x = x^{(1)} + \lambda d_1 = (0, 6\lambda)^T$$

所以 $f(x^{(1)} + \lambda d_1) = 72\lambda^2 - 36\lambda$, $\lambda_M = \min\{2/6, 5/30\} = 1/6$

求解线搜索问题

$$\begin{cases} \min & f(x^{(1)} + \lambda d_1) = 72\lambda^2 - 36\lambda \\ \text{s. t.} & 0 \leq \lambda \leq 1/6 \end{cases}$$

得最优解 $\lambda_1 = 1/6$, 于是得 $x^{(2)} = x^{(1)} + \lambda_1 d_1 = (0, 1)^T$ 。

第二次迭代

1° 确定搜索方向

在 $x^{(2)}$ 点, $\nabla f(x^{(2)}) = (-6, -2)^T$, 且 $J_2 = \{2, 3\}$, $M_2 =$

$$A_2 = \begin{pmatrix} 1 & 5 \\ -1 & 0 \end{pmatrix}, \quad P = I - A_2^T (A_2 A_2^T)^{-1} A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad P \nabla f(x^{(2)}) = (0, 0)^T, \quad u = - (A_2 A_2^T)^{-1} A_2 \nabla f(x^{(2)}) = (2/5, -28/5)^T.$$

因为 $u_3 = -28/5 < 0$, 所以应从 A_2 中去掉行 $(-1, 0)$, 得到 $\hat{A}_2 = (1, 5)$, 于是

$$\hat{P} = I - \hat{A}_2^T (A_2 \hat{A}_2^T)^{-1} \hat{A}_2 = \begin{pmatrix} 25/26 & -5/26 \\ -5/26 & 1/26 \end{pmatrix}$$

而 $d_2 = -\hat{P} \nabla f(x^{(2)}) = (70/13, -14/13)^T$, 由于 d_2 模的大小是不重要的, 不妨取 $d_2 = (5, -1)^T$ 。

2° 进行线搜索 因为

$$x^{(2)} + \lambda d_2 = (5\lambda, 1 - \lambda)^T$$

所以 $f(x^{(2)} + \lambda d_2) = 62\lambda^2 - 28\lambda - 4$

$$\lambda_M = \min\{1/4, 1\} = 1/4$$

求解线搜索问题

$$\begin{cases} \min & 62\lambda^2 - 28\lambda - 4 \\ \text{s. t.} & 0 \leq \lambda \leq 1/4 \end{cases}$$

得最优解 $\lambda_2 = 7/31$, 因此

$$\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \lambda_2 \mathbf{d}_2 = (35/31, 24/31)^T$$

第三次迭代

在 $\mathbf{x}^{(3)}$ 点, $\nabla f(\mathbf{x}^{(3)}) = (-32/31, -160/31)^T$, $J_3 = \{2\}$, $\mathbf{M}_3 = \mathbf{A}_3 = (1, 5)$,

$$\mathbf{P} = \mathbf{I} - \mathbf{A}_3^T (\mathbf{A}_3 \mathbf{A}_3^T)^{-1} \mathbf{A}_3 = 1/26 \begin{pmatrix} 25 & -5 \\ -5 & 1 \end{pmatrix}$$

$$\mathbf{d}_3 = -\mathbf{P} \nabla f(\mathbf{x}^{(3)}) = (0, 0)^T$$

$$\mathbf{u} = -(\mathbf{A}_3 \mathbf{A}_3^T)^{-1} \mathbf{A}_3 \nabla f(\mathbf{x}^{(3)}) = 32/31 > 0$$

所以 $\mathbf{x}^{(3)}$ 是 K-T 点, 又因为 $f(\mathbf{x})$ 是凸函数, 因此 $\mathbf{x}^{(3)}$ 也是问题 (5-9) 的全局最优解。 ■

* 4.6 既约梯度法

(The Method of Reduced Gradient)

在这一节, 我们介绍产生改进可行方向的另一种方法——既约梯度法。这种方法是 1963 年首先由 Wolfe 提出来的, 当时是为了解决具有线性约束条件的非线性规划问题。1969 年, 由 Abadie 与 Carpentier 推广到非线性约束条件的问题。既约梯度法的基本思想是: 利用约束条件将所考察问题中的某些变量用其他的一组独立变量来表示, 从而使问题的维数降低, 并且利用既约梯度, 直接构造出一个改进的可行方向 \mathbf{d} , 然后沿此方向进行线搜索, 从而求得一个新点, 这样一步步来逼近原问题的最优解。

4.6.1 线性约束情形

考察如下的线性约束优化问题：

$$\begin{cases} \min & f(x) \\ \text{s. t.} & Ax = b \\ & x \geq 0 \end{cases} \quad (6-1)$$

其中 A 为 $m \times n$ 矩阵, $\text{rank}(A) = m$, $x \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, $m \leq n$, 设 $f(x) \in C^1$, (6-1) 的可行集 Ω 的每一个顶点都是非退化的, 即每一个顶点 x 都有 m 个正分量。

1. 理论基础

定理 6.1 设 x 是问题(6-1)的一个可行解, 使得 $x^T = (x_B^T, x_N^T)$, $x_B > 0$, 其中 $A = (B, N)$, 而基矩阵 B 是 $m \times m$ 可逆矩阵, 用 I 表示基变量的指标集。令

$$r^T = \nabla f(x)^T - \nabla_B f(x)^T B^{-1} A = (r_B^T, r_N^T)$$

$d^T = (d_B^T, d_N^T)$ 是按式(6-2)和(6-3)形成的方向, 对非基变量 x_j , $j \in I$, 令

$$d_j = \begin{cases} -r_j, & \text{若 } r_j \leq 0 \\ -x_j r_j, & \text{若 } r_j > 0 \end{cases} \quad (6-2)$$

$$d_B = -B^{-1} N d_N \quad (6-3)$$

若 $d \neq 0$, 则 d 是一个改进的可行方向; 而 $d = 0$ 的充要条件是 x 为 K-T 点。

证明

1° 证明 $d \neq 0$ 是可行方向。

因 d 是一个可行方向, 当且仅当 $Ad = 0$, 且若 $x_j = 0$, 则 $d_j \geq 0$, $j = 1, 2, \dots, n$ 。由式(6-3)知

$$Ad = Bd_B + Nd_N = B(-B^{-1}Nd_N) + Nd_N = 0$$

若 x_j 是基变量, 则 $x_j > 0$, 若 x_j 不是基变量, 则由(6-2)可知

d_j 仅当 $x_j > 0$ 时, 可以是负的, 于是, 若 $x_j = 0$, 则 $d_j \geq 0$, 所以 d 是一个可行方向。

2° 证明 $d \neq 0$ 是一个改进方向。

$$\begin{aligned}\nabla f(x)^T d &= \nabla_B f(x)^T d_B + \nabla_N f(x)^T d_N \\ &= [\nabla_N f(x)^T - \nabla_B f(x)^T B^{-1} N] d_N \\ &= r_N^T d_N = \sum_{j \in I} r_j d_j\end{aligned}$$

其中 I 为基变量的指标集。

由 (6-2) 知, $\nabla f(x)^T d \leq 0$, 当 $d \neq 0$ 时, 显然有 $\nabla f(x)^T d < 0$, 由此及 4.4 的定理 4.1 可知 d 是一个改进方向。

3° 证明 $d = 0$ 的充要条件是 x 为 K-T 点。

注意到: x 是 K-T 点当且仅当存在向量 $u^T = (u_B^T, u_N^T) \geq (0, 0)$ 和 v , 使得

$$[\nabla_B f(x)^T, \nabla_N f(x)^T] + v^T(B, N) - (u_B^T, u_N^T) = (0, 0) \quad (6-4)$$

$$u_B^T x_B = 0, \quad u_N^T x_N = 0$$

因为 $x_B > 0$, 且 $u_B \geq 0$, 所以 $u_B^T x_B = 0$ 当且仅当 $u_B = 0$ 。由此及式 (6-4) 可得

$$v^T B = -\nabla_B f(x)^T$$

于是得

$$v^T = -\nabla_B f(x)^T B^{-1}$$

将它代入 (6-4) 可得

$$u_N^T = \nabla_N f(x)^T - \nabla_B f(x)^T B^{-1} N = r_N^T$$

因此, K-T 条件化简为

$$r_N \geq 0 \text{ 和 } r_N^T x_N = 0 \quad (6-5)$$

但是由 d 的定义, 注意到 $d = 0$ 的充要条件是 $r_N \geq 0$, 且 $r_N^T x_N = 0$ 。由此即得 x 是 K-T 点的充要条件为 $d = 0$ 。 ■

2. 计算步骤

1° 选定计算精度 $\epsilon > 0$, 初始点 $x^{(0)}$ 满足

$$Ax^{(0)} = b, x^{(0)} = (x_B^{(0)}, x_N^{(0)}), x_B^{(0)} > 0, x_N^{(0)} \geq 0$$

其中 $A = (B, N)$, B 为 m 阶可逆矩阵。用 I_k 表示对应于 B 中向量的指标集, 令 $k = 0$ 。

2° 计算 $(r_N^{(k)})^T = \nabla_N f(x^{(k)})^T - \nabla_B f(x^{(k)})^T B^{-1}N$, $r_N^{(k)}$ 称为既约梯度, 它有 $n - m$ 个分量。对 $j \in I_k$, 令

$$d_j = \begin{cases} -r_j^{(k)}, & \text{若 } r_j^{(k)} \leq 0 \\ -x_j^{(k)} r_j^{(k)}, & \text{若 } r_j^{(k)} > 0 \end{cases}$$

$$d_B = -B^{-1}Nd_N, \quad d = \begin{pmatrix} d_B \\ d_N \end{pmatrix}$$

若 $\|d\| < \epsilon$, 则计算结束, $x^{(k)}$ 为 K-T 点, 否则转 3°。

3° 令

$$\begin{cases} l_1 = \max \{ \lambda | x_B^{(k)} + \lambda d_B \geq 0 \} \\ l_2 = \max \{ \lambda | x_N^{(k)} + \lambda d_N \geq 0 \} \\ l = \min \{ l_1, l_2 \} \end{cases} \quad (6-6)$$

求解线搜索问题:

$$\begin{cases} \min & f(x^{(k)} + \lambda d) \\ \text{s. t.} & 0 \leq \lambda \leq l \end{cases} \quad (6-7)$$

得最优解 λ_k 。

4° 令 $x^{(k+1)} = x^{(k)} + \lambda_k d$, 若 $\lambda_k < l_1$, 令 $k = k + 1$, 转 2°。

若 $\lambda_k = l_1$, 求出 I_k 中使 $x_i^{(k+1)} = 0$ 的那些下标 i , 将它们从 I_k 中去掉, 换 \bar{I}_k 中 $x_j^{(k+1)}$ 值最大的那些 j , 得出新的 I_k 及相应的 B, N , 令 $k = k + 1$, 转 2°。

3. 算法应用举例

【例 6.1】 用既约梯度法求解

$$\begin{cases} \min & f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ \text{s. t.} & x_1 + x_2 + x_3 = 2 \\ & x_1 + 5x_2 + x_4 = 5 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{cases} \quad (6-8)$$

解 取初始可行点 $x^{(1)} = (0, 0, 2, 5)^T$

$$\nabla f(\mathbf{x}) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6, 0, 0)^T$$

第一次迭代

$$\text{因为 } I_1 = \{3, 4\}, \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}$$

$$(\mathbf{r}_N^{(1)})^T = (-4, -6) - (0, 0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix} = (-4, -6)$$

$$\text{故 } \mathbf{d}_N = (4, 6)^T, \mathbf{d}_B = - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} 4 \\ 6 \end{pmatrix} = \begin{pmatrix} -10 \\ -34 \end{pmatrix}$$

$$\mathbf{d} = (4, 6, -10, -34)^T$$

$$\text{因 } \mathbf{x}_B^{(1)} + \lambda \mathbf{d}_B = \begin{pmatrix} 2 \\ 5 \end{pmatrix} + \lambda \begin{pmatrix} -10 \\ -34 \end{pmatrix} = \begin{pmatrix} 2 - 10\lambda \\ 5 - 34\lambda \end{pmatrix}$$

$$\mathbf{x}_N^{(1)} + \lambda \mathbf{d}_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 4\lambda \\ 6\lambda \end{pmatrix}$$

$$\text{故 } l_1 = \max \{ \lambda | \mathbf{x}_B^{(1)} + \lambda \mathbf{d}_B \geq 0 \} = 5/34$$

$$l_2 = +\infty, l = \min \{ l_1, l_2 \} = 5/34$$

解线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(1)} + \lambda \mathbf{d}) = 56\lambda^2 - 52\lambda \\ \text{s. t.} & 0 \leq \lambda \leq 5/34 \end{cases}$$

得最优解为 $\lambda_1 = 5/34$, 因此

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{d} = (10/17, 15/17, 9/17, 0)^T$$

第二次迭代

在 $\mathbf{x}^{(2)} = (10/17, 15/17, 9/17, 0)^T$ 点, $I_2 = \{2, 3\}$, 与 I_2 相应的 \mathbf{B}, \mathbf{N} 为

$$\mathbf{B} = \begin{pmatrix} 1 & 1 \\ 5 & 0 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

而

$$\mathbf{B}^{-1} = \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix}$$

$$\nabla f(\mathbf{x}^{(2)}) = (-58/17, -62/17, 0, 0)^T$$

$$\begin{aligned}
 (\mathbf{r}_N^{(2)})^T &= (-58/17, 0) - (-62/17, 0) \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \\
 &= (-228/85, 62/85)
 \end{aligned}$$

所以 $\mathbf{d}_N = (228/85, 0)^T$

$$\mathbf{d}_B = - \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 228/85 \\ 0 \end{pmatrix} = \begin{pmatrix} -228/425 \\ -912/425 \end{pmatrix}$$

$$\mathbf{d} = (228/85, -228/425, -912/425, 0)^T$$

因为

$$\begin{aligned}
 \mathbf{x}_B^{(2)} + \lambda \mathbf{d}_B &= \begin{pmatrix} 15/17 \\ 9/17 \end{pmatrix} + \lambda \begin{pmatrix} -228/425 \\ -912/425 \end{pmatrix} \\
 &= \begin{pmatrix} 15/17 - (228/425)\lambda \\ 9/17 - (912/425)\lambda \end{pmatrix}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{x}_N^{(2)} + \lambda \mathbf{d}_N &= \begin{pmatrix} 10/17 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 228/85 \\ 0 \end{pmatrix} \\
 &= \begin{pmatrix} 10/17 + (228/85)\lambda \\ 0 \end{pmatrix}
 \end{aligned}$$

所以 $l_1 = \max\{\lambda | \mathbf{x}_B^{(2)} + \lambda \mathbf{d}_B \geq 0\} = 75/304$

$$l_2 = \max\{\lambda | \mathbf{x}_N^{(2)} + \lambda \mathbf{d}_N \geq 0\} = +\infty$$

$$l = \min\{l_1, l_2\} = 75/304$$

解线搜索问题

$$\begin{cases} \min & f(\mathbf{x}^{(2)} + \lambda \mathbf{d}) = \frac{31 \times 103968}{180625} \lambda^2 - \frac{51984}{7225} \lambda \\ \text{s. t.} & 0 \leq \lambda \leq 75/304 \end{cases}$$

得最优解 $\lambda_2 = 25/124$ 。因此

$$\begin{aligned}
 \mathbf{x}^{(3)} &= \mathbf{x}^{(2)} + \lambda_2 \mathbf{d} \\
 &= \begin{pmatrix} 10/17 \\ 15/17 \\ 9/17 \\ 0 \end{pmatrix} + \frac{25}{124} \begin{pmatrix} 228/85 \\ -228/425 \\ -912/425 \\ 0 \end{pmatrix} = \begin{pmatrix} 35/31 \\ 24/31 \\ 3/31 \\ 0 \end{pmatrix}
 \end{aligned}$$

第三次迭代

在 $\mathbf{x}^{(3)}$ 点, $I_3 = \{2, 3\}$, 因此

$$\mathbf{B} = \begin{pmatrix} 1 & 1 \\ 5 & 0 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \mathbf{B}^{-1} = \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix}$$

$$\nabla f(\mathbf{x}^{(3)}) = (-32/31, -160/31, 0, 0)^T$$

因为

$$\begin{aligned} (\mathbf{r}_N^{(3)})^T &= (-32/31, 0) - (-160/31, 0) \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix} \begin{pmatrix} 10 \\ 11 \end{pmatrix} \\ &= (0, 32/31) \end{aligned}$$

故 $\mathbf{d}_N = (0, 0)^T$, $\mathbf{d}_B = - \begin{pmatrix} 0 & 1/5 \\ 1 & -1/5 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
 $\mathbf{d} = (0, 0, 0, 0)^T$, 因此, $\mathbf{x}^{(3)} = (35/31, 24/31, 3/31, 0)^T$ 为 K-T 点, 此处即为问题 (6-8) 的最优解, 而最优值为 $f^* = f(\mathbf{x}^{(3)}) = -7.16$ 。 ■

4.6.2 非线性约束情形

考察如下的非线性约束优化问题:

$$\begin{cases} \min & f(\mathbf{x}), \\ \text{s. t.} & g_i(\mathbf{x}) = 0, i = 1, 2, \dots, m \\ & a_i \leq x_i \leq b_i, i = 1, 2, \dots, n, m < n \end{cases} \quad (6-9)$$

一般的非线性约束优化问题总可化为上述形式。因为当约束条件是不等式时, 可以通过引入松弛变量变成为等式约束, 当 x_i 无上界(或下界)时, 可令 $b_i = +\infty$ (或 $a_i = -\infty$)。假设 $f, g_i \in C^1 (i = 1, 2, \dots, m)$ 。与线性约束情形类似, 对任一可行点 $\mathbf{x} \in \Omega$, 假定

$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$ 满足如下的条件:

(1) $\mathbf{x}_B \in \mathbf{R}^m, \mathbf{x}_N \in \mathbf{R}^{n-m}$,

(2) 若用 I_B 表示对应于 B 中向量的指标集, I_N 表示对应于 N

中向量的指标集, 则 $a_i < x_i < b_i, i \in I_B$ 。

(3) 令 $g(x) = (g_1(x), \dots, g_m(x))^T$

$$\nabla_{Bg_i}(x) = \left(\frac{\partial g_i(x)}{\partial x_j}, j \in I_B \right)^T, i = 1, 2, \dots, m$$

则 $\nabla_B g(x) = (\nabla_{Bg_1}(x), \dots, \nabla_{Bg_m}(x))$ 为在 x 点的 $m \times m$ 非奇异矩阵。

$$\nabla_N g_i(x) = \left(\frac{\partial g_i(x)}{\partial x_j}, j \in I_N \right)^T, i = 1, 2, \dots, m$$

$$\nabla_N g(x) = (\nabla_N g_1(x), \dots, \nabla_N g_m(x))$$

$$\nabla g(x) = (\nabla_B g(x), \nabla_N g(x))$$

$$r(x_N) = \nabla_N f(x) - \nabla_N g(x) [\nabla_B g(x)]^{-1} \nabla_B f(x)$$

(6-10)

称 $r(x_N)$ 为 $f(x)$ 的既约梯度。

广义既约梯度法 (GRG) 计算步骤如下:

(1) 给定计算精度 $\varepsilon > 0$, 选取初始可行解 $x^{(0)} \in \Omega$, 分解 $x^{(0)}$

$$= \begin{bmatrix} x_B^{(0)} \\ x_N^{(0)} \end{bmatrix}, \text{使其满足条件(1)(2)(3), 令 } k = 0.$$

(2) 计算既约梯度

$$r_N^{(k)} = r(x_N^{(k)}) = \nabla_N f(x^{(k)}) - \nabla_N g(x^{(k)}) [\nabla_B g(x^{(k)})]^{-1} \nabla_B f(x^{(k)})$$

按下面的公式计算 d_N 的分量。

若 $x_j^{(k)} = a_j, r_j^{(k)} > 0$, 或 $x_j^{(k)} = b_j, r_j^{(k)} < 0$, 令 $d_j^{(k)} = 0, j \in I_N$; 否则令 $d_j^{(k)} = -r_j^{(k)}, j \in I_N$ 。

(3) 若 $\|d_N^{(k)}\| < \varepsilon$, 计算结束, $x^{(k)}$ 为 K-T 点; 否则转第(4)步。

(4) 解非线性方程组

$$g_i(x_B, \bar{x}_N^{(k)}) = 0, i = 1, 2, \dots, m \quad (6-11)$$

其中 $\bar{x}_N^{(k)}$ 是按下式确定的, 试取步长 $\lambda_k > 0$, 则

$$\bar{x} = \begin{cases} b_j, & \text{若 } x_j^{(k)} + \lambda_k d_j^{(k)} > b_j, j \in I_N \\ a_j, & \text{若 } x_j^{(k)} + \lambda_k d_j^{(k)} < a_j, j \in I_N \\ x_j^{(k)} + \lambda_k d_j^{(k)}, & \text{其他情形, } j \in I_N \end{cases} \quad (6-12)$$

求解(6-11)的方法有多种,比如可用牛顿法求解。为书写方便,用 y 代替 x_B , 则(6-11)可改写为

$$g(y, \bar{x}_N^{(k)}) = 0 \quad (6-11')$$

选定计算精度 $\varepsilon_1 > 0$ 和适当大的正整数 L 。

牛顿法的子程序如下:

(1) 令 $y^{(1)} = x_B^{(k)}$, $i = 1$ 。

(2) 计算

$$y^{(i+1)} = y^{(i)} - [\nabla_B g(y^{(i)}, \bar{x}_N^{(k)})]^{-1} g(y^{(i)}, \bar{x}_N^{(k)}), \quad i = 1, 2, \dots \quad (6-13)$$

若 $a_j \leq y_j^{(i+1)} \leq b_j$, $j \in I_B$, 且

$$f(y^{(i+1)}, \bar{x}_N^{(k)}) < f(y^{(1)}, \bar{x}_N^{(k)}) = f(x^{(k)})$$

$\|g(y^{(i+1)}, \bar{x}_N^{(k)})\| < \varepsilon_1$, 转(4); 否则转(3)。

(3) 若 $i < L$, 令 $i = i + 1$, 返回(2), 否则令 $\lambda_k = \lambda_k/2$,

$\bar{x}_N^{(k)} = \bar{x}_N^{(k)} + \lambda_k d_N^{(k)}$, 并检查 $a_N \leq \bar{x}_N^{(k)} \leq b_N$ 是否成立, 若不成立, 则按式(6-12)进行修正, 直到成立为止, 然后返回(1)。

(4) 令 $x^{(k+1)} = \begin{pmatrix} y^{(i+1)} \\ \bar{x}_N^{(k)} \end{pmatrix}$, 对 $x^{(k+1)}$ 选择新基 B , 使 $x_B^{(k+1)}$,

$x_N^{(k+1)}$ 满足条件(1)(2)(3), 然后令 $k = k + 1$, 返回主程序第2步。

由上述可见, 用于非线性约束优化问题的广义既约梯度法 (GRG) 是比较复杂的, 在实际计算时, 还有许多细节需要很好解决, 才能取得好的效果。但是值得注意的是, 1970 年以来的计算实践说明, 在非线性约束优化算法的比较性研究中, GRG 算法一直是比较优秀的重要方法之一。

4.7 乘子法

乘子法是由 Powell 和 Hestenes 于 1969 年彼此独立地对等式

约束的优化问题首先提出来的。1973年, Rockafellar 将其推广到不等式约束的优化问题, 十多年来获得了迅速的发展, 成为求解约束优化问题的一类重要而有效的方法。D. P. Bertsekas 在乘子法方面做过较多的工作, 1976年发表了一个综述^[65], 1982年出版了《约束优化与 Lagrange 乘子方法》的专著^[31], 对乘子法的理论作了系统的论述。1978年, D. A. Pierre 和 M. J. Lowe 给出了一个乘子法的 Fortran 程序, 1984年北京工业大学的史明仁也在该校学报上发表了乘子法的一个程序。在这一节, 我们对乘子法作一简介。

4.7.1 Hestenes 的乘子方法

惩罚函数法的主要缺点之一是: 当罚因子 M 越来越大时, 惩罚目标函数 $F(x, M)$ 的海赛矩阵越来越病态, 使无约束优化方法的计算难以进行下去, 而乘子法则克服了这个缺点。

下面来分析一下罚因子 M 之所以必须无限增大的原因。

考察等式约束的优化问题:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & h_i(x) = 0, i = 1, 2, \dots, p \end{cases} \quad (7-1)$$

令 $h(x) = (h_1(x), \dots, h_p(x))^T$; 引入罚函数

$$F(x, M) = f(x) + M \sum_{i=1}^p [h_i(x)]^2 \quad (7-2)$$

若 x^* 是问题(7-1)的最优解, 则 $h_j(x^*) = 0, j = 1, 2, \dots, p$ 。于是罚函数在 x^* 点的梯度为

$$\nabla_x F(x^*, M) = \nabla f(x^*) + 2M \sum_{j=1}^p h_j(x^*) \nabla h_j(x^*) = \nabla f(x^*)$$

而由 K-T 条件应有

$$\nabla_x L(x^*, \mu^*) = \nabla f(x^*) + \sum_{j=1}^p \mu_j \nabla h_j(x^*) = 0 \quad (7-3)$$

所以,一般说来, $\nabla f(x^*) \neq \mathbf{0}$, 只有当 x^* 是 $f(x)$ 的无约束优化问题的稳定点时, 才有 $\nabla f(x^*) = \mathbf{0}$ 成立。由于在一般情况下, $\nabla f(x^*) = \mathbf{0}$ 是不成立的, 所以找不到一个有限的 M , 使 $\nabla_x F(x^*, M) = \mathbf{0}$ 成立, 只能期望 $\lim_{M \rightarrow \infty} \nabla F(x^*, M) = \mathbf{0}$ 。

Hestenes 经过分析, 首先引入 增广 Lagrange 函数:

$$\varphi(x, \mu) = f(x) + \sum_{j=1}^p \mu_j h_j(x) + \frac{c}{2} \sum_{j=1}^p [h_j(x)]^2 \quad (7-4)$$

其中 $\mu_1, \mu_2, \dots, \mu_p$ 为 Lagrange 乘子, 这里将 M 改写为 $\frac{c}{2}$ 。(7-4) 式也可改写为

$$\varphi(x, \mu) = L(x, \mu) + \frac{c}{2} \sum_{j=1}^p [h_j(x)]^2 \quad (7-4')$$

其中

$$L(x, \mu) = f(x) + \sum_{j=1}^p \mu_j h_j(x)$$

为(7-1)的 Lagrange 函数。

由 4.1 知: 存在 μ^* , 使 (x^*, μ^*) 为 $L(x, \mu)$ 的稳定点, 即

$$\nabla_x L(x^*, \mu^*) = \mathbf{0} \quad (7-5)$$

而附加项 $\frac{c}{2} \sum_{j=1}^p [h_j(x)]^2$ 在 x^* 点的梯度为零。因此

$$\nabla_x \varphi(x^*, \mu^*) = \mathbf{0}$$

这说明 x^* 是 $\varphi(x, \mu^*)$ 的稳定点。可以证明^[35]: 存在 $c_1 > 0$, 当 $c \geq c_1$ 时, x^* 将是 $\varphi(x, \mu^*)$ 的极小点, 而求解问题(7-1)便可转化为对某个 μ^* , 求 $\varphi(x, \mu^*)$ 的无约束极小点了。如何求得这个 μ^* 呢? 我们可以这样做: 首先给定一个 足够大的正数 c , 然后在迭代过程中, 逐步调整 $\mu^{(k)}$, 如何调整呢? 假设已求得 $\mu^{(k)}$, 求解 $\min \varphi(x, \mu^{(k)})$ 得最优解 $x^{(k)}$, 则

$$\nabla_x \varphi(x^{(k)}, \mu^{(k)}) = \nabla f(x^{(k)}) + \sum_{j=1}^p \mu_j^{(k)} \nabla h_j(x^{(k)}) +$$

$$c \sum_{j=1}^p h_j(\mathbf{x}^{(k)}) \nabla h_j(\mathbf{x}^{(k)}) = \mathbf{0}$$

即

$$\nabla f(\mathbf{x}^{(k)}) + \sum_{j=1}^p (\mu_j^{(k)} + ch_j(\mathbf{x}^{(k)})) \nabla h_j(\mathbf{x}^{(k)}) = \mathbf{0} \quad (7-6)$$

为了使 $\mu^{(k)}$, $\mathbf{x}^{(k)}$ 分别逼近 μ^* , \mathbf{x}^* , 将(7-6)式与(7-3)式进行比较, 于是我们应采用公式

$$\mu_j^{(k+1)} = \mu_j^{(k)} + ch_j(\mathbf{x}^{(k)}), \quad j = 1, 2, \dots, p \quad (7-7)$$

来调整 $\mu^{(k)}$ 。那么, 迭代到什么时候才可以结束呢? 下面的定理为我们提供了依据。

定理 7.1 设 $\mathbf{x}^{(k)}$ 是无约束优化问题

$$\min \quad \varphi(\mathbf{x}, \mu^{(k)}) = f(\mathbf{x}) + \sum_{j=1}^p \mu_j^{(k)} h_j(\mathbf{x}) + \frac{c}{2} \sum_{j=1}^p [h_j(\mathbf{x})]^2 \quad (7-8)$$

的最优解, 则 $\mathbf{x}^{(k)}$ 是问题(7-1)的最优解, $\mu_1^{(k)}, \mu_2^{(k)}, \dots, \mu_p^{(k)}$ 为其相应的 Lagrange 乘子的充要条件是 $h_j(\mathbf{x}^{(k)}) = 0, j = 1, 2, \dots, p$ 。

证明 若 $\mathbf{x}^{(k)}$ 是问题(7-1)的最优解, 则显然有 $h_j(\mathbf{x}^{(k)}) = 0$ ($j = 1, 2, \dots, p$)。

反之, 若 $\mathbf{x}^{(k)}$ 是问题(7-8)的最优解, 且 $h_j(\mathbf{x}^{(k)}) = 0$ ($j = 1, 2, \dots, p$), 则对于任意 \mathbf{x} , 都有

$$\varphi(\mathbf{x}, \mu^{(k)}) \geq \varphi(\mathbf{x}^{(k)}, \mu^{(k)}) = f(\mathbf{x}^{(k)})$$

特别, 当 \mathbf{x} 为问题(7-1)的任一可行解时, 有 $f(\mathbf{x}) = \varphi(\mathbf{x}, \mu^{(k)}) \geq f(\mathbf{x}^{(k)})$, 所以 $\mathbf{x}^{(k)}$ 是问题(7-1)的最优解, 即 $\mathbf{x}^* = \mathbf{x}^{(k)}$ 。同时应有

$$\nabla_{\mathbf{x}} \varphi(\mathbf{x}^{(k)}, \mu^{(k)}) = \nabla f(\mathbf{x}^*) + \sum_{j=1}^p \mu_j^{(k)} \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$

这就说明 $\mu^{(k)}$ 是与 $\mathbf{x}^{(k)} = \mathbf{x}^*$ 对应的 Lagrange 乘子向量。 ■

由上述定理得知: 迭代计算可采用如下的结束准则: 选定计算精度 $\varepsilon > 0$, 若 $\mathbf{x}^{(k)}$ 满足 $\|\mathbf{h}(\mathbf{x}^{(k)})\| < \varepsilon$, 则停止迭代, 否则继续

进行迭代。若在迭代过程中, $\mu^{(k)}$ 收敛太慢或不收敛, 可适当增大 c 。

综上所述, 用乘子法求解问题(7-1) 的迭代步骤如下:

1° 给定初始点 $x^{(0)}$, 初始乘子向量 $\mu^{(1)}$ (若无其他信息, 可取 $\mu^{(1)} = \mathbf{0}$), 计算精度 $\varepsilon > 0$, 取 $c > 0$, $0 < r < 1$ (比如取 $r = 0.25$), $\alpha > 1$ (比如取 $\alpha = 2 \sim 10$), 令 $k = 1$ 。

2° 以 $x^{(k-1)}$ 为初始点, 求解 $\min \varphi(x, \mu^{(k)})$, 得解 $x^{(k)}$, 其中 $\varphi(x, \mu)$ 由(7-4) 式确定。

3° 若 $\|h(x^{(k)})\| < \varepsilon$, 计算结束, 取 $x^{(k)}$ 为(7-1) 的最优解; 否则计算 $\beta = \|h(x^{(k)})\| / \|h(x^{(k-1)})\|$, 若 $\beta \leq r$, 转到第 4° 步, 否则令 $c = \alpha c$, 转到第 4° 步。

4° 计算 $\mu_j^{(k+1)} = \mu_j^{(k)} + ch_j(x^{(k)})$ ($j = 1, 2, \dots, p$), 令 $k = k + 1$, 返回第 1° 步。

4.7.2 Powell 的乘子方法

Powell 在 1969 年与 Hestenes 几乎同时, 但又各自独立地提出了一种类似的乘子方法。他考虑含有两组参数的罚函数:

$$M(x, \sigma, \alpha) = f(x) + \frac{1}{2} \sum_{j=1}^p \sigma_j [h_j(x) + \alpha_j]^2 \quad (7-9)$$

在上式中, 若令 $\sigma_j = c$, $\alpha_j = \mu_j/c$, 则得

$$\begin{aligned} M(x, \sigma, \alpha) &= f(x) + \frac{c}{2} \sum_{j=1}^p \left([h_j(x)]^2 + 2 \frac{\mu_j}{c} h_j(x) + \left(\frac{\mu_j}{c} \right)^2 \right) \\ &= f(x) + \frac{c}{2} \sum_{j=1}^p [h_j(x)]^2 + \sum_{j=1}^p \mu_j h_j(x) + \frac{1}{2c} \sum_{j=1}^p \mu_j^2 \end{aligned}$$

将上式与(7-4) 式对比, 可得

$$M(x, \sigma, \alpha) = \varphi(x, \mu) + \frac{1}{2c} \sum_{j=1}^p \mu_j^2$$

即 $M(x, \sigma, \alpha)$ 与 $\varphi(x, \mu)$ 仅差一个常数项。因此, Powell 方法是一种比 Hestenes 的方法更广泛一些的乘子方法, 但两者的出发点不同。Powell 方法基于下面的事实。

定理 7.2 设对某两组参数 σ 和 $\alpha, x^*(\sigma, \alpha)$ 是 $M(x, \sigma, \alpha)$ 的无约束极小点, 则 $x^*(\sigma, \alpha)$ 也是问题

$$\begin{cases} \min & f(x) \\ \text{s. t.} & h_j(x) = h_j(x^*(\sigma, \alpha)), j = 1, 2, \dots, p \end{cases} \quad (7-10)$$

的极小点。

证明 (用反证法) 假设 $x^*(\sigma, \alpha)$ 不是问题(7-10)的极小点, 则存在可行点 \bar{x} , 使

$f(\bar{x}) < f(x^*(\sigma, \alpha)), h_j(\bar{x}) = h_j(x^*(\sigma, \alpha)), j = 1, 2, \dots, p$
于是有

$$\begin{aligned} M(\bar{x}, \sigma, \alpha) &= f(\bar{x}) + \sum_{j=1}^p \sigma_j [h_j(\bar{x}) + \alpha_j]^2 \\ &< f(x^*(\sigma, \alpha)) + \sum_{j=1}^p \sigma_j [h_j(x^*(\sigma, \alpha)) + \alpha_j]^2 \\ &= M(x^*(\sigma, \alpha), \sigma, \alpha) \end{aligned}$$

上式与 $x^*(\sigma, \alpha)$ 是 $M(x, \sigma, \alpha)$ 的无约束极小点相矛盾, 所以 $x^*(\sigma, \alpha)$ 是问题(7-10)的极小点。 ■

由定理 7.2 得知: 若能找到两组参数 σ 和 α , 使

$$h_j(x^*(\sigma, \alpha)) = 0, j = 1, 2, \dots, p \quad (7-11)$$

则这个点 $x^*(\sigma, \alpha)$ 就是问题(7-1)的最优解。为此我们期望产生一系列参数值 $\sigma^{(k)}$ 与 $\alpha^{(k)}$, 使

$$\lim_{k \rightarrow \infty} h_j(x(\sigma^{(k)}, \alpha^{(k)})) = 0, j = 1, 2, \dots, p$$

则当 $k \rightarrow \infty$ 时, 就可使 $x(\sigma^{(k)}, \alpha^{(k)})$ 的极限 $x^*(\sigma, \alpha)$ 满足(7-11)式。

为了得到 $\alpha_j^{(k)} (j = 1, 2, \dots, p)$ 的调整公式, 求出 $M(x, \sigma, \alpha)$ 对 x 的梯度

$$\nabla_x M(x, \sigma, \alpha) = \nabla f(x) + \sum_{j=1}^p \sigma_j [h_j(x) + \alpha_j] \nabla h_j(x) \quad (7-12)$$

将(7-12)与(7-3)对比,并注意到 $\alpha_j = \mu_j/c, \sigma_j = c$, 即知: 对固定的充分大的 $\sigma_j (j = 1, 2, \dots, p)$, 可采用公式

$$\alpha_j^{(k+1)} = \alpha_j^{(k)} + h_j(\mathbf{x}(\boldsymbol{\sigma}, \boldsymbol{\alpha}^{(k)})), \quad j = 1, 2, \dots, p \quad (7-13)$$

来调整 $\alpha_j^{(k)}$ 。若 $\alpha_j^{(k)}$ 收敛太慢或不收敛, 可增大 $\sigma_j, j = 1, 2, \dots, p$ 。

关于 Powell 乘子方法的计算步骤与前述的 Hestenes 乘子方法类似, 不再详述。

4.7.3 Rockafellar 的乘子方法

1973 年, Rockafellar 将乘子方法推广到不等式约束的优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \geq 0, i = 1, 2, \dots, p \end{cases} \quad (7-14)$$

他的办法是, 引入松弛变量 $z_i (i = 1, 2, \dots, p)$, 将不等式约束化为等式约束, 即

$$g_i(\mathbf{x}) - z_i^2 = 0, \quad i = 1, 2, \dots, p \quad (7-15)$$

然后再利用前面讲述的等式约束优化问题的结果推出相应的结果, 考虑在等式约束(7-15)下的增广 Lagrange 函数:

$$\Phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{j=1}^p \mu_j [g_j(\mathbf{x}) - z_j^2] + \frac{c}{2} \sum_{j=1}^p [g_j(\mathbf{x}) - z_j^2]^2 \quad (7-16)$$

将 $\Phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu})$ 关于 \mathbf{z} 求极小, 令

$$\begin{aligned} \varphi(\mathbf{x}, \boldsymbol{\mu}) &= \min_{\mathbf{z}} \Phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}) \\ \nabla_{\mathbf{z}} \Phi(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}) &= \mathbf{0} \end{aligned}$$

得

$$z_j [cz_j^2 - (\mu_j + cg_j(\mathbf{x}))] = 0, \quad j = 1, 2, \dots, p$$

若 $\mu_j + cg_j(\mathbf{x}) \leq 0$, 则 $z_j^2 = 0$;

若 $\mu_j + cg_j(\mathbf{x}) > 0$, 则 $z_j^2 = \frac{1}{c}(\mu_j + cg_j(\mathbf{x}))$, 因此

$$g_j(\mathbf{x}) - z_j^2 = \begin{cases} g_j(\mathbf{x}), & \text{若 } \mu_j + cg_j(\mathbf{x}) \leq 0 \\ -\mu_j/c, & \text{若 } \mu_j + cg_j(\mathbf{x}) > 0 \end{cases} \quad (7-17)$$

于是当 $\mu_j + cg_j(\mathbf{x}) \leq 0$ 时, 有

$$\begin{aligned} & \mu_j [g_j(\mathbf{x}) - z_j^2] + \frac{c}{2} [g_j(\mathbf{x}) - z_j^2]^2 \\ &= \mu_j g_j(\mathbf{x}) + \frac{c}{2} (g_j(\mathbf{x}))^2 \\ &= \frac{1}{2c} [(\mu_j + cg_j(\mathbf{x}))^2 - \mu_j^2] \end{aligned}$$

当 $\mu_j + cg_j(\mathbf{x}) > 0$ 时, 有

$$\begin{aligned} & \mu_j [g_j(\mathbf{x}) - z_j^2] + \frac{c}{2} [g_j(\mathbf{x}) - z_j^2]^2 \\ &= -\frac{\mu_j^2}{c} + \frac{c}{2} \left(-\frac{\mu_j}{c} \right)^2 = -\frac{1}{2c} \mu_j^2 \end{aligned}$$

所以

$$\begin{aligned} \varphi(\mathbf{x}, \mu) &= \min_z \Phi(\mathbf{x}, \mathbf{z}, \mu) \\ &= f(\mathbf{x}) + \frac{1}{2c} \sum_{j=1}^p \{ [\min(0, \mu_j + cg_j(\mathbf{x}))]^2 - \mu_j^2 \} \end{aligned} \quad (7-18)$$

利用等式约束优化问题乘子迭代的修正公式(7-7)和(7-17)可得乘子迭代公式为

$$\mu_j^{(k+1)} = \min \{ 0, \mu_j^{(k)} + cg_j(\mathbf{x}^{(k)}) \} \quad (7-19)$$

而结束准则可采用

$$\left(\sum_{j=1}^p [\min(g_j(\mathbf{x}), -\mu_j/c)]^2 \right)^{1/2} < \varepsilon \quad (7-20)$$

其中 $\varepsilon > 0$ 为计算精度。

对于一般的非线性最优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & h_i(\mathbf{x}) = 0, i = 1, 2, \dots, m \\ & g_j(\mathbf{x}) \geq 0, j = 1, 2, \dots, p \end{cases} \quad (7-21)$$

与上述类似,可令

$$\begin{aligned} \varphi(\mathbf{x}, \lambda, \mu) = & f(\mathbf{x}) + \sum_{i=1}^m \lambda_i h_i(\mathbf{x}) + \frac{c}{2} \sum_{i=1}^m (h_i(\mathbf{x}))^2 + \\ & \frac{1}{2c} \sum_{j=1}^p \{ [\min(0, \mu_j + c g_j(\mathbf{x}))]^2 - \mu_j^2 \} \end{aligned} \quad (7-22)$$

这时的乘子迭代公式与(7-7)式和(7-19)式类似,而迭代计算的结束准则可采用

$$\sum_{i=1}^m h_i^2(\mathbf{x}^{(k)}) + \sum_{j=1}^p \left(\min(g_j(\mathbf{x}^{(k)}), -\frac{\mu_j^{(k)}}{c}) \right)^2 < \epsilon^2 \quad (7-23)$$

4.7.4 方法应用举例

【例 7.1】 用乘子法求解问题

$$\begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s. t.} & x_1 - 1 \geq 0 \end{cases} \quad (7-24)$$

解 显然问题(7-24)的最优解为 $\mathbf{x}^* = (1, 0)^T$ 。下面用乘子法来求解。

增广 Lagrange 函数为

$$\begin{aligned} \varphi(\mathbf{x}, \mu) = & x_1^2 + x_2^2 + \frac{1}{2c} \{ [\min(0, \mu + c(x_1 - 1))]^2 - \mu^2 \} \\ = & \begin{cases} x_1^2 + x_2^2 + \mu(x_1 - 1) + \frac{c}{2}(x_1 - 1)^2, & x_1 \leq 1 - \frac{\mu}{c} \\ x_1^2 + x_2^2 - \frac{\mu^2}{2c}, & x_1 > 1 - \frac{\mu}{c} \end{cases} \end{aligned}$$

为求解 $\min \varphi(\mathbf{x}, \mu)$,一般需用无约束优化的数值方法,这里 $\varphi(\mathbf{x}, \mu)$ 比较简单,可用解析法求解。

$$\frac{\partial \varphi}{\partial x_1} = \begin{cases} 2x_1 + \mu + c(x_1 - 1), & x_1 \leq 1 - \frac{\mu}{c}, \\ 2x_1, & x_1 > 1 - \frac{\mu}{c}, \end{cases} \quad \frac{\partial \varphi}{\partial x_2} = 2x_2$$

令 $\nabla \varphi(\mathbf{x}, \mu) = \mathbf{0}$, 得 $\varphi(\mathbf{x}, \mu)$ 的极小点为

$$x_1 = \frac{c - \mu}{c + 2}, \quad x_2 = 0$$

取 $c = 4$, $\mu^{(1)} = 0$, 则 $x_1^{(1)} = 2/3$, $x_2^{(1)} = 0$, 于是

$$\mu^{(2)} = \min\{0, 0 + 4(2/3 - 1)\} = -4/3$$

一般的 $\mathbf{x}^{(k)} = \left(\frac{4 - \mu^{(k)}}{6}, 0 \right)^T$

$$\mu^{(k+1)} = \min\{0, \mu^{(k)} + 4(x_1^{(k)} - 1)\} = \frac{\mu^{(k)} - 4}{3}$$

所以, 当 $k \rightarrow +\infty$ 时, $\mu^{(k)} \rightarrow -2$, $\mathbf{x}^{(k)} \rightarrow (1, 0)^T = \mathbf{x}^*$ 。

作数值计算可得

$$\mu^{(2)} = -4/3, \mathbf{x}^{(2)} = (8/9, 0)^T = (0.88889, 0)^T$$

$$\mu^{(3)} = -16/9, \mathbf{x}^{(3)} = (26/27, 0)^T = (0.96296, 0)^T$$

$$\mu^{(4)} = -52/27, \mathbf{x}^{(4)} = (80/81, 0)^T = (0.98765, 0)^T$$

⋮

若用外点法, 计算结果如下表:

M_k	$\mathbf{x}^{(k)T} = \left(\frac{M_k}{M_k + 1}, 0 \right)$	M_k	$\mathbf{x}^{(k)T} = \left(\frac{M_k}{M_k + 1}, 0 \right)$
1	(0.5, 0)	5	(0.8333, 0)
2	(0.66667, 0)	6	(0.85714, 0)
3	(0.75, 0)	7	(0.87500, 0)
4	(0.8, 0)	8	(0.88889, 0)

从上述计算结果可见, 乘子法比外点法收敛要快得多。

4.8 二次逼近法

由于线性规划和二次规划都比较容易求解, 所以人们很自然

想到：把要求解的一般的非线性约束优化问题线性化，然后用线性规划方法来逐步求其近似解，这种方法称为线性逼近法或序列线性规划法，简称为 SLP 法。但是线性逼近法逼近的精度差，收敛速度慢，而近 20 年来二次规划的解法已取得很大进展，有了比较有效的算法，因此近十多年来用二次规划来逐步逼近非线性规划的方法（称为二次逼近法或序列二次规划法，简称为 SQP 法）获得了突出的进展，成为当前世界上最流行的重要约束优化算法之一。本节对二次规划和 SQP 法作简要介绍，详细论述见文献 32, 35, 36，计算程序见文献 42。

4.8.1 二次规划 (QP)

1. 二次规划的概念

所谓二次规划是指在变量 x_1, x_2, \dots, x_n 的线性等式和线性不等式约束下，求二次函数 $Q(x)$ 的极小值问题：

$$\begin{cases} \min & Q(x) = \frac{1}{2}x^T G x + g^T x \\ \text{s. t.} & a_i^T x = b_i, i = 1, 2, \dots, m \\ & a_i^T x \leq b_i, i = m + 1, \dots, p \end{cases} \quad (8-1)$$

其中 G 为 n 阶对称矩阵， g, a_1, a_2, \dots, a_p 均为 n 维列向量，假设 a_1, \dots, a_m 线性无关。而 $x = (x_1, x_2, \dots, x_n)^T, b_1, b_2, \dots, b_p$ 为已知常数， $m \leq n, p \geq m$ 。用 S 表示 (8-1) 的可行集。

二次规划问题是最简单的一类非线性规划问题，也是最早被研究的一类非线性约束优化问题。它不仅有自己的实际背景，而且是 SQP 法的基础。

问题 (8-1) 的约束可能不相容，也可能没有有限的最小值，这时称 QP 问题无解。若 $G \geq 0$ ，则问题 (8-1) 就是一个凸 QP 问题，它的任何局部最优解，也是全局最优解，简称整体解；若 $G > 0$ ，则 (8-1) 是一个正定（严格凸）QP 问题，只要存在整体解，则它是惟

一的;若 G 不定,则问题(8-1) 是一个一般的 QP 问题。

【例 8.1】 考察二次规划问题:

$$\begin{cases} \min & z = x_1^2 + x_2^2 - 2x_1 - 4x_2 \\ \text{s. t.} & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \end{cases} \quad (8-2)$$

将 $z = Q(x)$ 写成 $z = Q(x) = \frac{1}{2}x^T Gx + g^T x$, 其中

$$g = (-2, -4)^T, x = (x_1, x_2)^T$$

而 $G = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, 可行集

$$S = \{x | x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0\}$$

显然, $G > 0$, 所以(8-2) 是一个严格凸二次规划问题, 其惟一整体解为 $x^* = (0, 1)^T$, 最优值 $z^* = -3$ 。

定义 设 \bar{x} 是问题(8-1) 的可行解, 若某个 $i \in \{1, 2, \dots, p\}$, 使得 $a_i^T \bar{x} = b_i$ 成立, 则称它为 \bar{x} 点处的有效约束, 称在 \bar{x} 点处所有有效约束的指标组成的集合 $J = J(\bar{x}) = \{i | a_i^T \bar{x} = b_i\}$ 为 \bar{x} 点处的有效约束指标集, 简称为 \bar{x} 点处的有效集。

显然, 对于任何可行点 \bar{x} , 所有等式约束都是有效约束, 只有不等式约束才可能是非有效约束。在例 1 中, 易见 $J(x^*) = \{1, 2\}$, 即在 x^* 点处 $x_1 + x_2 = 1$ 和 $x_1 \geq 0$ 是有效约束, 而 $x_2 \geq 0$ 不是有效约束。

2. 正定二次规划问题解的性质

在问题(8-1) 中假设 $G > 0$, 为以后讨论方便, 令

$$E = \{1, 2, \dots, m\}, \quad I = \{m+1, \dots, p\}$$

根据本章 4.1 的定理 1.1 和定理 1.2 易得

定理 8.1 点 x^* 是正定二次规划问题(8-1) 的严格整体解的充要条件是 x^* 为 K-T 点, 即有乘子向量 $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*, \lambda_{m+1}^*, \dots, \lambda_p^*)^T$, 使得

$$\begin{cases} Gx^* + g + \sum_{i \in E} \lambda_i^* a_i + \sum_{i \in I} \lambda_i^* a_i = 0 \\ a_i^T x^* - b_i = 0, \quad i \in E \\ a_i^T x^* - b_i \leq 0, \quad i \in I \\ \lambda_i^* \geq 0, i \in I; \quad \lambda_i^* = 0, i \in I \setminus J^* \end{cases} \quad (8-3)$$

其中 J^* 为 x^* 处的有效集。

K-T 点 x^* 与其相应的乘子向量 λ^* , 称为问题(8-1) 的 K-T 对。

定理 8.2 若正定 QP 问题(8-1) 有可行解, 则它必有最优解, 且最优解是惟一的。

解的惟一性显然, 解的存在性证明见文献 28。

定理 8.3 设 x^* 是正定 QP 问题(8-1) 的最优解, 且在 x^* 点处的有效集为 J^* , 则 x^* 是如下等式约束问题的惟一解。

$$\begin{cases} \min Q(x) = \frac{1}{2} x^T G x + g^T x \\ \text{s. t. } a_i^T x = b_i, \quad i \in J^* \end{cases} \quad (8-4)$$

证明 因为 x^* 是(8-1) 的最优解, 由定理 8-1 知存在 λ^* , 满足(8-3) 式, 易见(8-3) 式蕴含有

$$Gx^* + g + \sum_{i \in J^*} \lambda_i^* a_i = 0 \quad (8-5)$$

根据上式再对问题(8-4) 应用定理 8-1, 即知定理 8.3 的结论成立。 ■

3. 仅含等式约束的正定二次规划

下面讨论仅含等式约束的正定 QP 问题:

$$\begin{cases} \min Q(x) = \frac{1}{2} x^T G x + g^T x \\ \text{s. t. } a_i^T x = b_i, \quad i \in E = \{1, 2, \dots, m\} \end{cases} \quad (8-6)$$

这里假设 a_1, a_2, \dots, a_m 线性无关, G 为 n 阶对称正定矩阵, $m \leq n$ 。

令 $A = (a_1, \dots, a_m)$, $b = (b_1, \dots, b_m)^T$, 则有

定理 8.4 在上述假设下,问题(8-6)的 K-T 对 (x^*, λ^*) 是存在惟一的,且 (x^*, λ^*) 为(8-6)的 K-T 对的充要条件是它们满足如下的方程组:

$$\begin{pmatrix} G & A \\ A^T & O \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} -g \\ b \end{pmatrix} \quad (8-7)$$

证明 由定理 8.1 知: x^* 和 λ^* 为问题(8-1)的解及其相应乘子向量的充要条件是它们满足

$$\begin{cases} Gx + g + \sum_{i \in E} \lambda_i a_i = 0 \\ a_i^T x - b_i = 0, \quad i \in E \end{cases}$$

注意到 $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*)^T$, 将上式写成矩阵形式,即为(8-7)。因此问题(8-6)与方程组(8-7)等价。

由于 a_1, a_2, \dots, a_m 线性无关,且 $m \leq n$,所以问题(8-6)总有可行解,由定理 8.2 知,(8-6)有惟一解 x^* ,易见与 x^* 相应的乘子向量 λ^* 也是存在惟一的。

上述定理说明:可以通过求解方程组(8-7)来获得问题(8-6)的解。 ■

【例 8.2】 求解

$$\begin{cases} \min & Q(x) = \frac{1}{2}(x_1^2 + x_2^2 + x_3^2) \\ \text{s. t.} & x_1 + 2x_2 - x_3 = 4 \\ & x_1 - x_2 + x_3 = -2 \end{cases} \quad (8-8)$$

这里 $n = 3, m = 2, G = I_3, g = 0$ 。

$$\text{解} \quad A = (a_1, a_2) = \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ -2 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

与(8-7) 相应的方程组为

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 1 & 2 & -1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 4 \\ -2 \end{pmatrix}$$

求解上面方程组, 得到其惟一解 $x_1 = 2/7$, $x_2 = 10/7$, $x_3 = -6/7$, $\lambda_1 = -4/7$, $\lambda_2 = 2/7$, 因此原问题(8-8) 的最优解为 $\mathbf{x}^* = (2/7, 10/7, -6/7)^T$, 相应的乘子向量为 $\lambda^* = (-4/7, 2/7)^T$ 。

4. 一般正定二次规划的有效集法

(1) 基本思想

对一般正定 QP 问题(8-1), 由定理 8-3 得知: 只要能找到解 \mathbf{x}^* 满足的有效集 J^* , 就可通过求解等式约束的 QP 问题得到 \mathbf{x}^* 。这就启发我们, 可以先求出问题(8-1) 的一个可行点 $\mathbf{x}^{(1)}$, 并求出 $\mathbf{x}^{(1)}$ 点处的有效集 J_1 。然后按照使目标函数值减少的原则, 对有效集不断进行调整, 直到得到 J^* 为止, 从而获得问题(8-1) 的最优解 \mathbf{x}^* 。

(2) 方法的导出

设 $\mathbf{x}^{(1)}$ 为(8-1) 的初始可行解, 其有效集为 J_1 , 它含有 t 个元素, 不妨设 $J_1 = \{1, 2, \dots, m, m+1, \dots, t\}$, 且 $\mathbf{a}_i (i \in J_1)$ 线性无关。设问题

$$\begin{cases} \min Q(x) = \frac{1}{2}x^T Gx + g^T x \\ \text{s. t. } a_i^T x = b_i, \quad i \in J_1 \end{cases} \quad (8-9)$$

的最优解为 $\bar{x}^{(1)}$, 相应的乘子向量为 $\lambda^{(1)}$ 。下面分别对 $\bar{x}^{(1)} \neq x^{(1)}$ 和 $\bar{x}^{(1)} = x^{(1)}$ 进行讨论。

1° $\bar{x}^{(1)} \neq x^{(1)}$ 。因为 $\bar{x}^{(1)}$ 和 $x^{(1)}$ 分别是问题(8-9)的严格整体解和可行解, 所以 $Q(\bar{x}^{(1)}) < Q(x^{(1)})$ 。

a. 若 $\bar{x}^{(1)}$ 为原问题(8-1)的可行解, 即 $\bar{x}^{(1)} \in S$, 令 $x^{(2)} = \bar{x}^{(1)}$, 并求出 $x^{(2)}$ 点处的有效集 J_2 , 用 $x^{(2)}$ 和 J_2 代替 $x^{(1)}$ 和 J_1 , 重复前述步骤。

b. 若 $\bar{x}^{(1)} \notin S$, 而 $x^{(1)} \in S$, 令 $d_1 = \bar{x}^{(1)} - x^{(1)}$, 易见, 从 $x^{(1)}$ 出发沿方向 d_1 前进到 $\bar{x}^{(1)}$ 的过程中, $Q(x)$ 是减少的, 而且在到达 $\bar{x}^{(1)}$ 之前必然会遇到某个不等式约束 $a_i^T x \leq b_i$, $t+1 \leq i \leq p$ 的边界, 设最先遇到的那个约束的指标为 l , 相应的交点记为 $x^{(2)}$, 则有 $Q(x^{(2)}) < Q(x^{(1)})$, $x^{(2)}$ 点处的有效集记为 J_2 , 则 $x^{(2)}$ 满足 $a_i^T x = b_i$, $i \in J_2$, 以 $x^{(2)}$ 和 J_2 代替 $x^{(1)}$ 和 J_1 , 重复前述步骤。

2° $\bar{x}^{(1)} = x^{(1)}$ 。分两种情形进行讨论。

a. 若乘子 $\lambda_{m+1}^{(1)}, \dots, \lambda_t^{(1)} \geq 0$, 则由定理 8.1 知 $\bar{x}^{(1)} = x^{(1)}$ 是问题(8-1)的最优解, 计算结束。

b. 若 $\lambda_{m+1}^{(1)}, \dots, \lambda_t^{(1)}$ 中有负值, 令 $\lambda_q^{(1)} = \min \{\lambda_i^{(1)} | i = m+1, \dots, t\}$, 则 $\lambda_q^{(1)} < 0$ 。经分析后可知: 这时应取 $x^{(2)} = \bar{x}^{(1)}$, $J_2 = J_1 \setminus \{q\}$ 。再用 $x^{(2)}$ 和 J_2 代替 $x^{(1)}$ 和 J_1 , 重复前述步骤, 这时将出现情形 1° ($\bar{x}^{(2)} \neq x^{(2)}$)。

从上面的分析中得知: 在方法的计算过程中, 需要在已知 $x^{(k)}$ 为

$$\begin{cases} \min & Q(x) = \frac{1}{2}x^T Gx + g^T x \\ \text{s. t.} & a_i^T x = b_i, \quad i \in J_k \end{cases} \quad (8-10)$$

的可行点的条件下,求问题(8-10)的解 $\bar{x}^{(k)}$ 。若令 $d_k = \bar{x}^{(k)} - x^{(k)}$,则可把问题转化为寻求 d_k 。事实上,由 $\bar{x}^{(k)} = x^{(k)} + d_k$ 可得

$$\begin{aligned} Q(\bar{x}^{(k)}) &= \frac{1}{2}(x^{(k)} + d_k)^T G(x^{(k)} + d_k) + g^T(x^{(k)} + d_k) \\ &= \frac{1}{2}d_k^T Gd_k + (Gx^{(k)} + g)^T d_k + Q(x^{(k)}) \end{aligned} \quad (8-11)$$

而 $\bar{x}^{(k)}$ 满足约束 $a_i^T x = b_i, i \in J_k$ 与 d_k 满足 $a_i^T d_k = 0, i \in J_k$ 等价,因此 d_k 是正定QP问题

$$\begin{cases} \min & q(x) = \frac{1}{2}d^T Gd + (Gx^{(k)} + g)^T d \\ \text{s. t.} & a_i^T d = 0, \quad i \in J_k \end{cases} \quad (8-12)$$

的最优解。由 d_k 即可求得(8-10)的最优解 $\bar{x}^{(k)}$ 。

(3) 正定二次规划问题(8-1)有效集法的计算步骤

1° 选取问题(8-1)的初始可行点 $x^{(1)}$,确定 $x^{(1)}$ 点处的有效集 J_1 ,使 $a_i (i \in J_1)$ 线性无关,令 $k = 1$ 。

2° 求解仅含等式约束的正定二次规划问题(8-12),设其解为 $d^{(k)}$ 。

3° 若 $d^{(k)} = 0$,则计算相应的乘子 $\lambda_i^{(k)} (i \in J_k)$,转4°;否则转5°。

4° 若 $\forall i \in J_k \cap I$,都有 $\lambda_i^{(k)} \geq 0$,则 $x^{(k)}$ 为问题(8-1)的最优解,计算结束;否则求出 $\lambda_q^{(k)} = \min\{\lambda_i^{(k)} | i \in J_k \cap I\}$,令 $x^{(k+1)} = x^{(k)}$, $J_{k+1} = J_k \setminus \{q\}$, $k = k + 1$,返回2°。

5° 若 $\bar{x}^{(k)} = x^{(k)} + d^{(k)}$ 满足 $a_i^T x \leq b_i, i \in I \setminus J_k$,则令 $x^{(k+1)} = \bar{x}^{(k)}$,并求出 $x^{(k+1)}$ 点处的有效集 J_{k+1} ,令 $k = k + 1$,返回2°;否

则转 6°。

6° 计算步长 $\alpha_k = \min \{ -(\mathbf{a}_i^T \mathbf{x}^{(k)} - b_i) / \mathbf{a}_i^T \mathbf{d}^{(k)} \mid i \in I \setminus J_k, \mathbf{a}_i^T \mathbf{d}^{(k)} > 0 \}$, 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$, 并求 $\mathbf{x}^{(k+1)}$ 点处的有效集 J_{k+1} , 令 $k = k + 1$, 返回 2°。

【例 8.3】 求解如下的 QP 问题

$$\begin{cases} \min & Q(x) = x_1^2 + x_2^2 - 2x_1 - 4x_2 \\ \text{s. t.} & a_1^T x - b_1 = -x_1 \leq 0 \\ & a_2^T x - b_2 = -x_2 \leq 0 \\ & a_3^T x - b_3 = x_1 + x_2 - 1 \leq 0 \end{cases} \quad (8-13)$$

解 易见 $\mathbf{x}^{(1)} = (0, 0)^T$ 为可行解, $\mathbf{x}^{(1)}$ 处的有效集为 $J_1 = \{1, 2\}$ 。

第一次迭代

1° 取初始可行点 $\mathbf{x}^{(1)} = (0, 0)^T$, $J_1 = \{1, 2\}$ 。

2° 写出并求解与问题(8-12)相应的问题。

因为 $G = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, $\mathbf{g} = \begin{pmatrix} -2 \\ -4 \end{pmatrix}$, 所以与(8-12)相应的问题为

$$\begin{cases} \min & z = d_1^2 + d_2^2 - 2d_1 - 4d_2 \\ \text{s. t.} & -d_1 = 0, \quad -d_2 = 0 \end{cases}$$

它的解显然为 $d_1^{(1)} = d_2^{(2)} = 0$ 。

3° $\mathbf{d}^{(1)} = \mathbf{0}$, 其相应乘子向量可仿照例 2 求得为

$$\lambda^{(1)} = (-2, -4)^T$$

4° 计算 $\lambda_q^{(1)} = \min(\lambda_1^{(1)}, \lambda_2^{(2)}) = \lambda_2^{(1)} = -4$, 令 $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} = (0, 0)^T$, $J_2 = J_1 \setminus \{2\} = \{1\}$, 转入第二次迭代。

第二次迭代

求解问题

$$\begin{cases} \min & z = d_1^2 + d_2^2 - 2d_1 - 4d_2 \\ \text{s. t.} & -d_1 = 0 \end{cases}$$

显然 $d_1^{(2)} = 0$, $d_2^{(2)}$ 可任取, 若取 $d_2^{(2)} = 2$, 则 $\mathbf{d}^{(2)} = (0, 2)^T$ 。因为 $\mathbf{d}^{(2)} \neq \mathbf{0}$, 所以按算法转 5°, 令 $\bar{\mathbf{x}}^{(2)} = \mathbf{x}^{(2)} + \mathbf{d}^{(2)} = (0, 2)^T$ 。由于 $\bar{\mathbf{x}}^{(2)}$ 不是可行点, 所以转 6°, 计算步长

$$\alpha_2 = \min \{ -(\mathbf{a}_i^T \mathbf{x}^{(2)} - b_i) / \mathbf{a}_i^T \mathbf{d}^{(2)} \mid i = 2, 3, \mathbf{a}_i^T \mathbf{d}^{(2)} > 0 \} = \frac{1}{2}$$

令 $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \alpha_2 \mathbf{d}^{(2)} = (0, 1)^T$, 求出 $J_2 = \{1, 3\}$, 转入第三次迭代。

第三次迭代

求解问题

$$\begin{cases} \min & z = d_1^2 + d_2^2 - 2d_1 - 4d_2 \\ \text{s. t.} & -d_1 = 0, \quad d_1 + d_2 = 0 \end{cases}$$

得解 $\mathbf{d}^{(3)} = (0, 0)^T$, $\lambda^{(3)} = (0, 2)^T$ 。所以 $\mathbf{x}^{(3)} = (0, 1)^T$ 即为问题 (8-13) 的最优解, 其相应的乘子向量为 $\lambda^* = (0, 0, 2)^T$ 。 ■

(4) 两点注记

1° 用有效集法求解一般正定 QP 问题, 需要先求出它的一个初始可行解 $\mathbf{x}^{(1)}$, $\mathbf{x}^{(1)}$ 可仿照线性规划中求初始可行解方法得到, 所以这是一种两阶段算法。为克服这一缺点, 减少求解正定 QP 问题的工作量。1983 年 Goldfarb 和 Idnani 提出了一种对偶方法, 其详细介绍见文献 36。

2° 上面介绍的有效集法对更一般的 QP 问题也是适用的, 因篇幅所限未能细述, 有兴趣的读者可参看文献 36 的第三章。

4.8.2 二次逼近法

考虑一般的约束优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & c_i(\mathbf{x}) = 0, \quad i \in E = \{1, 2, \dots, m\} \\ & c_i(\mathbf{x}) \leq 0, \quad i \in I = \{m+1, \dots, p\} \end{cases} \quad (8-14)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $f, c_i (i \in E \cup I): \mathbf{R}^n \rightarrow \mathbf{R}^1$ 。假设 $f \in$

C^2 , $c_1, \dots, c_p \in C^1$ 。用 Ω 表示问题(8-14)的可行集, 假设 $\Omega \neq \emptyset$ 。

1. 基本思想

二次逼近法的基本思想是: 把问题(8-14)转化为求解一系列的二次规划子问题, 因此也称为序列二次规划法, 简称为 SQP 法, 具体地说, 假定在第 k 次迭代开始时已知近似解 $\mathbf{x}^{(k)}$ 和近似乘子向量 $\lambda^{(k)}$, 根据它们, 可以给出第 k 个二次规划子问题 (P_k) , 求解 (P_k) 即可得到新的近似解 $\mathbf{x}^{(k+1)}$, 并确定相应的乘子向量 $\lambda^{(k+1)}$, \dots , 重复上述过程, 直到获得问题(8-14)所需的近似最优解为止。若令 $\mathbf{d}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, 则求解问题 (P_k) 来获得 $\mathbf{x}^{(k+1)}$ 可转化为求 $\mathbf{d}^{(k)}$ 的二次规划子问题。如何来得到相应的二次规划子问题呢? 可以设想, 对问题(8-14), 相应的二次规划问题的约束可以用将 $c_i(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 点线性化的方法得到, 即应考虑如下的 QP 问题

$$\begin{cases} \min & Q(\mathbf{d}) = \frac{1}{2} \mathbf{d}^T \mathbf{B}_k \mathbf{d} + \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} \\ \text{s. t.} & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + c_i(\mathbf{x}^{(k)}) = 0, \quad i \in E \\ & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + c_i(\mathbf{x}^{(k)}) \leq 0, \quad i \in I \end{cases} \quad (8-15)$$

但是这样做有时会遇到困难。例如, 对下面的例 8.4, 将导致相应的可行域为空集, 但原问题的可行域非空。

【例 8.4】 考察约束条件为

$$\begin{cases} c_1(x) = x - 1 \leq 0 \\ c_2(x) = -x^2 \leq 0 \end{cases}$$

的问题, 设 $x^0 = 3$, 则与(8-15)相应的约束条件为

$$\begin{cases} \nabla c_1(x^0)^T \mathbf{d} + c_1(x^0) = d + 2 \leq 0 \\ \nabla c_2(x^0)^T \mathbf{d} + c_2(x^0) = -6d - 9 \leq 0 \end{cases}$$

显然这两个约束是互相矛盾的, 即可行域是空集, 而原问题的可行域为 $\Omega = (-\infty, 1]$ 。为克服上述缺点, 可以这样做: 把(8-15)的约束分为两类:

$$I_1 = \{i | c_i(\mathbf{x}^{(k)}) \geq 0, i \in I\}, \quad I_2 = \{i | c_i(\mathbf{x}^{(k)}) < 0, i \in I\}$$

然后用如下约束(8-16)来代替问题(8-15)中的约束,其中 ξ 是一个待定参数。

$$\begin{cases} \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi c_i(\mathbf{x}^{(k)}) = 0, & i \in E \\ \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_1 \\ \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_2 \end{cases} \quad (8-16)$$

选择 ξ 的原则是使(8-16)的可行域 S 非空,且尽量接近问题(8-15)中的约束,具体办法是:考察以 ξ, d_1, \dots, d_n 为变量的LP问题(8-17)。

$$\begin{cases} \min & z = -\xi \\ \text{s. t.} & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi c_i(\mathbf{x}^{(k)}) = 0, & i \in E \\ & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_1 \\ & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_2 \\ & 0 \leq \xi \leq 1 \end{cases} \quad (8-17)$$

其中 $\mathbf{d} = (d_1, \dots, d_n)^T$ 。

易见 $\xi = 0, \mathbf{d} = \mathbf{0}$ 是问题(8-17)的一个可行解,所以它总有最优解 $(\xi^*, \mathbf{d}^*)^T$ 。把(8-16)式中的 ξ 取为 ξ^* ,就符合上述选择 ξ 的原则。当原来的二次规划问题(8-15)有可行解 $\bar{\mathbf{d}}$ 时,易见 $\xi = 1, \mathbf{d} = \bar{\mathbf{d}}$ 就是LP问题(8-17)的最优解,即 $\xi^* = 1, \mathbf{d}^* = \bar{\mathbf{d}}$ 。这时约束条件(8-16)与问题(8-15)的约束条件相同。

综合上述可见:确定搜索方向 $\mathbf{d}^{(k)}$ 的LP问题应取为

$$\begin{cases} \min & Q(\mathbf{x}) = \frac{1}{2} \mathbf{d}^T \mathbf{B}_k \mathbf{d} + \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} \\ \text{s. t.} & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi^* c_i(\mathbf{x}^{(k)}) = 0, & i \in E \\ & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + \xi^* c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_1 \\ & \nabla c_i(\mathbf{x}^{(k)})^T \mathbf{d} + c_i(\mathbf{x}^{(k)}) \leq 0, & i \in I_2 \end{cases} \quad (8-18)$$

其中 E, I_1, I_2 由前面所述各式确定, ξ^* 由求解LP问题(8-17)确定。

【例 8.5】 对例 8.4 中的问题,确定QP问题(8-18)的约束条件和可行域。

解 因为 $x^0 = 3$, $c_1(3) = 2 > 0$, $c_2(3) = -9 < 0$, 所以 $I_1 = \{1\}$, $I_2 = \{2\}$ 。注意到: $\nabla c_1(x) = 1$, $\nabla c_2(x) = -2x$, 所以与 (8-17) 相应的 LP 问题为

$$\begin{cases} \min & z = -\xi \\ \text{s. t.} & d + 2\xi \leq 0 \\ & -6d - a \leq 0 \\ & 0 \leq \xi \leq 1 \end{cases} \quad (8-19)$$

易见其最优解为 $(\xi^*, d^*)^T = (3/4, -3/2)^T$ 。因此相应于问题 (8-18) 的约束条件为

$$\begin{cases} \nabla c_1(x^0)^T d + \xi^* c_1(x^0) = d + 3/2 \leq 0 \\ \nabla c_2(x^0)^T d + c_2(x^0) = -6d - 9 \leq 0 \end{cases}$$

其可行解为 $d = -3/2$ 。 ■

2. 计算步骤

1° 选取初始点 $x^{(1)}$, 初始正定对称矩阵 B_1 (例如取 B_1 为单位阵), 计算精度 $\varepsilon > 0$, 令 $k = 1$ 。

2° 求解 LP 问题 (8-17), 设其最优解为 $(\xi^*, d^{*T})^T$ 。

3° 求解正定 QP 问题 (8-18) 得最优解 $d^{(k)}$ 及相应的乘子向量 $\mu^{(k)}$ 。若 $d^{(k)} = 0$, 则取 $x^{(k+1)} = x^{(k)} + d^{(k)}$ 为问题 (8-14) 的近似最优解, 计算结束; 否则转 4°。

4° 从 $x^{(k)}$ 出发, 沿方向 $d^{(k)}$, 对目标函数 $\varphi(\alpha) = w(x^{(k)} + \alpha d^{(k)}, \lambda^{(k)})$ 进行一维搜索, 确定步长 α_k , 其中

$$w(x, \lambda) = f(x) + \sum_{i \in E} \lambda_i |c_i(x)| + \sum_{i \in I} \lambda_i \max\{0, c_i(x)\}$$

$$\lambda^{(k)} = (\lambda_1^{(k)}, \dots, \lambda_m^{(k)}, \lambda_{m+1}^{(k)}, \dots, \lambda_p^{(k)})^T$$

由 $\mu^{(k)}$ 和 $\lambda^{(k-1)}$ 根据下式确定:

$$\begin{cases} \lambda_i^{(1)} = |\mu_i^{(1)}|, & i \in E \cup I \\ \lambda_i^{(k)} = \max\{|\mu_i^{(k)}|, \frac{1}{2}(\lambda_i^{(k-1)} + |\mu_i^{(k)}|)\}, & i \in E \cup I, k > 1 \end{cases}$$

5° 令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ 。

6° 若 $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon$, 则取 $\mathbf{x}^{(k+1)}$ 为问题(8-14)的近似最优解, 计算结束; 否则转 7°。

7° 按 BFGS 公式将 B_k 修正为 B_{k+1} , 即令

$$B_{k+1} = B_k - \frac{B_k \mathbf{s}^{(k)} (\mathbf{s}^{(k)})^T B_k}{(\mathbf{s}^{(k)})^T B_k \mathbf{s}^{(k)}} + \frac{\eta^{(k)} (\eta^{(k)})^T}{(\eta^{(k)})^T \mathbf{s}^{(k)}} \quad (8-21)$$

其中 $\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, $\eta^{(k)} = \theta \mathbf{y}^{(k)} + (1 - \theta) B_k \mathbf{s}^{(k)}$
 $\mathbf{y}^{(k)} = \nabla_x L(\mathbf{x}^{(k+1)}, \boldsymbol{\mu}^{(k)}) - \nabla_x L(\mathbf{x}^{(k)}, \boldsymbol{\mu}^{(k)})$

$$L(\mathbf{x}, \boldsymbol{\mu}^{(k)}) = f(\mathbf{x}) + \sum_{i=1}^p \mu_i^{(k)} c_i(\mathbf{x})$$

$$\boldsymbol{\mu}^{(k)} = (\mu_1^{(k)}, \dots, \mu_p^{(k)})^T$$

$$\theta = \begin{cases} 1, & \text{当 } (\mathbf{y}^{(k)})^T \mathbf{s}^{(k)} \geq 0.2 (\mathbf{s}^{(k)})^T B_k \mathbf{s}^{(k)} \text{ 时} \\ \frac{0.8 (\mathbf{s}^{(k)})^T B_k \mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^T B_k \mathbf{s}^{(k)} - (\mathbf{y}^{(k)})^T \mathbf{s}^{(k)}}, & \text{否则} \end{cases} \quad (8-22)$$

8° 令 $k = k + 1$, 转 2°。

4.9 极大熵方法

4.9.1 引言

极大熵方法是近年来出现的一种新的优化方法, 它的基本思想是: 利用最大熵原理推导出一个可微函数 $G_p(x)$ (通常称为极大熵函数), 用函数 $G_p(x)$ 来逼近最大值函数 $G(x) = \max_{1 \leq i \leq m} \{g_i(x)\}$, 就可把求解多约束优化问题转化为单约束优化问题, 把某些不可微优化问题转化为可微优化问题, 使问题简化。数值试验说明: 这种算法效果良好, 它为求解大型多约束优化问题和某些不可微问题提供了一种新的思路和途径。为了能真正有效地求解大规模约束优化问题, 还需要解决如下两个问题:

(1) 必须找到一种计算速度快、存贮容量小、适于求解大规模无约束优化问题的好算法与之配套。

(2) 研制一种性能优良的计算机软件,它具有计算速度快、存贮容量小、精度高、通用性强和较好的健壮性。这就需要很好地解决算法实现过程中的一些难点。例如,用差商代替导数如何保证计算精度;一维搜索算法、无约束优化方法与极大熵方法的配合;如何避免计算过程中可能发生的上溢和下溢等。

通过对适于求解大规模无约束优化问题的若干共轭梯度法、有限内存 BFGS 方法进行分析、研究和数值试验,证实有限内存 BFGS 方法确是一种较好的大系统优化方法,它具有与 BFGS 方法基本相同的快速收敛的优点,而占用的内存容量当变量数目 n 较大时比一般的 BFGS 方法少得多,将它与极大熵方法结合,取得了良好的数值结果。

4.9.2 极大熵方法

考虑一般的约束优化问题

$$\begin{cases} \min & F(\mathbf{x}) = \max_{1 \leq k \leq s} f_k(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & h_j(\mathbf{x}) = 0, j = 1, 2, \dots, l \\ & \mathbf{x} \in \mathbf{R}^n \end{cases} \quad (9-1a)$$

其中 $f_k(\mathbf{x}), g_i(\mathbf{x}), h_j(\mathbf{x}) : \mathbf{R}^n \rightarrow \mathbf{R}, f_k(\mathbf{x}), g_i(\mathbf{x}), h_j(\mathbf{x}) \in C^1$ 。若 $s \geq 2$, 一般来说,问题(9-1a)是一个不可微的优化问题,它的求解是困难的,当 $n, m + l$ 较大时难度更大。下面给出求解问题(9-1a)的一种有效的近似方法——极大熵方法。

令 $G(\mathbf{x}) = \max_{1 \leq i \leq m} \{g_i(\mathbf{x})\}, H(\mathbf{x}) = \max_{1 \leq j \leq l} \{h_j^2(\mathbf{x})\}$ 。易证下面的引理成立。

引理 1 问题(9-1a)与下面的问题(9-1b)等价。

$$\begin{cases} \min & F(\mathbf{x}) \\ \text{s. t.} & G(\mathbf{x}) \leq 0 \\ & H(\mathbf{x}) = 0 \\ & \mathbf{x} \in \mathbf{R}^n \end{cases} \quad (9-1b)$$

令

$$F_p(\mathbf{x}) = \frac{1}{p} \ln \sum_{i=1}^s \exp(pf_i(\mathbf{x})) \quad p > 0$$

$$G_q(\mathbf{x}) = \frac{1}{q} \ln \sum_{i=1}^m \exp(qg_i(\mathbf{x})) \quad q > 0$$

$$H_t(\mathbf{x}) = \frac{1}{t} \ln \sum_{j=1}^l \exp(th_j^2(\mathbf{x})) \quad t > 0$$

不难证明如下结果。

定理 9.1 (1) 当 $p \rightarrow +\infty$ 时, $F_p(\mathbf{x})$ 一致收敛于 $F(\mathbf{x})$, 且有

$$F(\mathbf{x}) \leq F_p(\mathbf{x}) \leq F(\mathbf{x}) + (\ln s)/p$$

(2) 当 $q \rightarrow +\infty$ 时, $G_q(\mathbf{x})$ 一致收敛于 $G(\mathbf{x})$, 且有

$$G(\mathbf{x}) \leq G_q(\mathbf{x}) \leq G(\mathbf{x}) + (\ln m)/q;$$

(3) 当 $t \rightarrow +\infty$ 时, $H_t(\mathbf{x})$ 一致收敛于 $H(\mathbf{x})$, 且有

$$H(\mathbf{x}) \leq H_t(\mathbf{x}) \leq H(\mathbf{x}) + (\ln l)/t \quad (9-2)$$

证明 (1) ~ (3) 的证明类似, 下面仅给出 (3) 的证明, (1)、

(2) 的证明从略。注意到 $h_j^2(\mathbf{x}) \leq H(\mathbf{x})$, 即知

$$\begin{aligned} H_t(\mathbf{x}) &\leq \frac{1}{t} \ln \sum_{j=1}^l \exp(tH(\mathbf{x})) = \frac{1}{t} \ln [l \exp(tH(\mathbf{x}))] \\ &= H(\mathbf{x}) + (\ln l)/t \quad t > 0 \end{aligned}$$

$$\begin{aligned} \text{又} \quad H_t(\mathbf{x}) - H(\mathbf{x}) &= \frac{1}{t} \ln \sum_{j=1}^l \exp(th_j^2(\mathbf{x})) - \frac{1}{t} \ln \exp(tH(\mathbf{x})) \\ &= \frac{1}{t} \left(\ln \sum_{j=1}^l \exp(t(h_j^2(\mathbf{x}) - H(\mathbf{x}))) \right) \end{aligned}$$

而 $h_j^2(\mathbf{x}) - H(\mathbf{x}) \leq 0$, 且至少有一个 j ($1 \leq j \leq l$), 使

$$h_j^2(\mathbf{x}) - H(\mathbf{x}) = 0$$

因此必有

$$\sum_{j=1}^l \exp(t(h_j^2(\mathbf{x}) - H(\mathbf{x}))) \geq 1$$

所以 $H_t(\mathbf{x}) - H(\mathbf{x}) \geq 0$, 即 $H(\mathbf{x}) \leq H_t(\mathbf{x})$ 。 ■

由(9-2) 易见当 $t \rightarrow +\infty$ 时, $H_t(\mathbf{x})$ 一致收敛于 $H(\mathbf{x})$ 。

根据引理 9.1 和定理 9.1 可知, 求问题(9-1a) 的近似最优解, 可转化为求解当 p, q, t 为正且充分大时如下的优化问题。

$$\left\{ \begin{array}{ll} \min & F_p(\mathbf{x}) \\ \text{s. t.} & G_q(\mathbf{x}) \leq 0 \\ & H_t(\mathbf{x}) \leq (r \ln l)/t \\ & \mathbf{x} \in \mathbf{R}^n \end{array} \right. \quad (9-3)$$

其中 $r \in (1, +\infty)$ 为常数。而问题(9-3) 仅含有两个不等式约束, 且目标函数和约束函数, 当 $f_k, g_i, h_j \in C^1$ 时, 均是连续可微的, 因而比问题(9-1a) 容易求解。利用增广 Lagrange 乘子法可进一步将问题(9-3) 转化为无约束优化问题来求解, 而无约束优化问题可用有限内存的 BFGS 方法求解, 这样就为求解大规模的约束优化问题和某些不可微问题提供了一种比较简单而有效的近似方法。下面通过构造问题(9-3) 的增广 Lagrange 函数来给出具体的算法。令

$$\begin{aligned} \varphi(\mathbf{x}, \mu) = f(\mathbf{x}) + \frac{1}{2c} \{ & [\max(0, \mu_1 + cG_q(\mathbf{x}))]^2 - \mu_1^2 + \\ & [\max(0, \mu_2 + c(H_t(\mathbf{x}) - (r \ln l)/t))]^2 - \mu_2^2 \} \end{aligned} \quad (9-4)$$

它就是问题(9-3) 的增广 Lagrange 函数。

算法的计算步骤:

1° 给定初始点 $\mathbf{x}^{(0)}$, 初始 Lagrange 乘子 $\mu_1^{(1)} = 0, \mu_2^{(1)} = 0, c > 0, p, q, t \in [10^3, 10^6], r \geq 1$, 计算精度 $\varepsilon > 0$, 令 $k = 1$ 。

2° 以 $\mathbf{x}^{(k-1)}$ 为初始点, 用有限内存 BFGS 方法求解 $\min \varphi(\mathbf{x},$

$\mu^{(k)}$, 设其解为 $x^{(k)}$, 其中 $\varphi(x, \mu)$ 由 (9-4) 确定。

3° 计算

$$\tau = \left\{ [\max(G_q(x^{(k)}), \mu_1^{(k)}/c)]^2 + \left(\max\left(H_t(x^{(k)}) - \frac{r \ln l}{t}, \frac{\mu_2^{(k)}}{c}\right) \right)^2 \right\}$$

若 $\tau \leq \varepsilon$, 计算结束, 取 $x^{(k)}$ 为问题 (9-1a) 的近似最优解; 否则计算

$$\beta = \left\{ G_q^2(x^{(k)}) + \left(H_t(x^{(k)}) - \frac{r \ln l}{t} \right)^2 \right\}^{1/2} / \left\{ G_q^2(x^{(k-1)}) + \left(H_t(x^{(k-1)}) - \frac{r \ln l}{t} \right)^2 \right\}^{1/2}$$

若 $\beta < 1/4$, 转 4°; 否则令 $c = 2c$, 转 4°。

4° 计算

$$\mu_1^{(k+1)} = \max(0, \mu_1^{(k)} + \mu_1^{(k)} + cG_q(x^{(k)}))$$

$$\mu_2^{(k+1)} = \max\left\{0, \mu_2^{(k)} + c\left(H_t(x^{(k)}) - \frac{r \ln l}{t}\right)\right\}$$

令 $k = k + 1$, 返回 2°。

称上述算法为算法 A, 它的可行性和有效性是人们关注的重要问题, 作者从理论分析和数值计算与应用两方面进行了研究。在文献 [86, 87] 中, 就问题 (9-1a) 是凸规划的情况, 证明当参数 $p, q, t \rightarrow +\infty$ 时, 近似问题 (9-3) 的最优解序列的任何极限点 x^* , 在一定的条件下, 都是问题 (9-1a) 的最优解。

4.9.3 数值结果

为使上述算法实际可行和有效, 必须研制一种性能优良的计算软件, 解决好算法实现过程中的一些难点。近年来作者已研制几个计算程序, 下面列举三个数值例子的结果。

【例 9.1】

$$\min f(x) = \sum_{i=1}^{n/2} [100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2]$$

$$x^* = (1, 1, \dots, 1)^T, f(x^*) = 0$$

分别计算了 $n = 10, 100, 200$, 均取得了较好的结果, 为节省篇幅, 下面仅列出 $n = 10$ 的数值结果。

初值取为 $x_i^{(0)} = 500, i = 1, 2, \dots, 10$, 则可得到如下近似最优解

$$\mathbf{x}^* = (1.0000229795, 1.0000459494, 0.9999234840, \\ 0.9998462224, 1.0000707678, 1.0001418558, \\ 1.0000706221, 1.0001415639, 0.9999032601, \\ 0.9998062873)^T$$

【例 9.2】

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\ \text{s. t.} \quad 0 \leq g_1(\mathbf{x}) \leq 92 \\ \quad \quad 90 \leq g_2(\mathbf{x}) \leq 110 \\ \quad \quad 20 \leq g_3(\mathbf{x}) \leq 25 \\ \quad \quad 78 \leq x_1 \leq 102 \\ \quad \quad 33 \leq x_2 \leq 45 \\ \quad \quad 27 \leq x_3 \leq 45 \\ \quad \quad 27 \leq x_4 \leq 45 \\ \quad \quad 27 \leq x_5 \leq 45 \end{array} \right.$$

其中

$$g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - \\ 0.0022053x_3x_5$$

$$g_2(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + \\ 0.0021813x_3^2$$

$$g_3(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + \\ 0.0019085x_3x_4$$

取 $p = q = t = 10^3, c = 10^6, \epsilon_1 = 10^{-3}, \epsilon = 10^{-8}$ (算法 A), 结果如下:

文献[46] 方法			算法 A	
变量	初 值	近似解	初 值	近似解
x_1	78.62	78.000	500.0	78.0019616
x_2	33.44	33.000	500.0	33.0014176
x_3	31.07	29.995	500.0	29.9937421
x_4	44.18	45.000	500.0	44.9974306
x_5	35.22	36.776	500.0	36.7741758
$f(\mathbf{x})$		- 30665.5		- 30665.9979268

【例 9.3】

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = \sum_{i=1}^{10} x_i \left(c_i + \ln \left(x_i / \sum_{j=1}^{10} x_j \right) \right) \\ \text{s. t.} \quad h_1(\mathbf{x}) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0 \\ \quad \quad h_2(\mathbf{x}) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0 \\ \quad \quad h_3(\mathbf{x}) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0 \\ \quad \quad x_i \geqslant 0, i = 1, 2, \dots, 10 \end{array} \right.$$

其中 $c_1 = -6.089$, $c_2 = -17.164$, $c_3 = -34.054$
 $c_4 = -5.914$, $c_5 = -24.721$, $c_6 = -14.986$
 $c_7 = -24.100$, $c_8 = -10.708$, $c_9 = -26.662$
 $c_{10} = -22.179$

取 $p = q = 10^3$, $c = 10^6$, $\varepsilon_1 = 0.01$, $\varepsilon = 10^{-6}$ 。初值为 $x_i^{(0)} = 0.1$, $i = 1, 2, \dots, 10$, 计算结果为:

	可变容差法 ^[46]	GRG 法 ^[46]	算法 A
$f(\mathbf{x})$	- 47.736	- 47.761	- 47.6785
x_1	0.0128	0.0406	0.0417
x_2	0.1433	0.1477	0.1446
x_3	0.8078	0.7832	0.7757
x_4	0.0062	0.0014	0.0201
x_5	0.4790	0.4853	0.4659
x_6	0.0033	0.0007	0.0196
x_7	0.0324	0.0274	0.0285
x_8	0.0281	0.0180	0.0243
x_9	0.0250	0.0375	0.0367
x_{10}	0.0817	0.0969	0.0981
$h_1(\mathbf{x})$	0	0	0
$h_2(\mathbf{x})$	-10^{-4}	10^{-4}	0
$h_3(\mathbf{x})$	0	5×10^{-4}	0

4.9.4 结束语

通过理论分析和数值计算与应用两个方面的研究和试算,得到的初步结论是:极大熵方法与有限内存 BFGS 方法的有机结合,为求解大系统的非线性约束优化问题提供了一种新的思路和途径,它具有计算速度快、存贮容量小、简单易用、具有大范围收敛性等优点,初步的数值计算与应用实例说明:这是一种比较有效的大系统优化方法。当然,现在的工作还是初步的,需要继续深入和完善。

为了便于读者学习和编写计算程序,下面以附录的形式对有限内存 BFGS 方法作一简介。

附录 有限内存 BFGS 方法

考虑无约束优化问题

$$\min f(\mathbf{x}) \quad (9-5)$$

其中 $\mathbf{x} \in \mathbf{R}^n, f: \mathbf{R}^n \rightarrow \mathbf{R}, f \in C^1$

拟牛顿方程可写成

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{H}_{k+1} \mathbf{y}_k, k = 1, 2, \dots \quad (9-6)$$

其中 $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \mathbf{d}_k$ 为搜索方向, α_k 为搜索步长, $\mathbf{d}_k = -\mathbf{H}_k \mathbf{g}_k$, BFGS 修正公式可写成^[37]

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right) \mathbf{H}_k \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \quad (9-7)$$

其中 $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ 。有限内存 BFGS 方法的基本出发点是减少内存。

记 $\rho_k = 1/\mathbf{s}_k^T \mathbf{y}_k \cdot \mathbf{V}_k = (\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T)$, 则(9-7)可改写成为

$$\begin{aligned} \mathbf{H}_{k+1} = & (\mathbf{V}_k^T \cdots \mathbf{V}_{k-i}^T) \mathbf{H}_{k-i} (\mathbf{V}_{k-i} \cdots \mathbf{V}_k) + \sum_{j=0}^{i-1} \rho_{k-i+j} \cdot \\ & \left(\prod_{l=0}^{i-j-1} \mathbf{V}_{k-l}^T \right) \mathbf{s}_{k-i+j} \mathbf{s}_{k-i+j}^T \left(\prod_{l=0}^{i-j-1} \mathbf{V}_{k-l}^T \right)^T + \rho_k \mathbf{s}_k \mathbf{s}_k^T \end{aligned} \quad (9-8)$$

令 $i = m, H_{k-m} = H_k^{(0)}$, 则得到有限内存 BFGS 方法的矩阵修正公式

$$H_{k+1} = (V_k^T \cdots V_{k-m}^T) H_k^{(0)} (V_{k-m} \cdots V_k) + \sum_{j=0}^{m-1} \rho_{k-m+j} \cdot \left(\prod_{l=0}^{m-j-1} V_{k-l}^T \right) s_{k-i+j} s_{k-i+j}^T \left(\prod_{l=0}^{m-j-1} V_{k-l} \right)^T + \rho_k s_k s_k^T \quad (9-9)$$

其中 $H_k^{(0)}$ 可取为

$$H_k^{(0)} = \frac{s_k^T y_k}{\|y_k\|_2^2} I \quad (9-10)$$

综上所述可知, 有限内存 BFGS 方法只需存贮 $s_i, y_i (i = k - m, \dots, k)$ 少数几个 n 维向量就够了。下面给出有限内存 BFGS 方法(简称算法 B)的计算步骤。

1° 给定 $x_1 \in \mathbf{R}^n, H_1 \in \mathbf{R}^{n \times n}$ 对称正定, 取非负整数 \hat{m} (一般取 $3 \leq \hat{m} \leq 8$), 取 $0 < b_1 \leq b_2 < 1$, 精度 $\varepsilon > 0$, 令 $k = 1$ 。

2° 若 $\|g_k\| \leq \varepsilon$, 则计算结束, 取最优解为 $x^* \approx x_k$; 否则, 计算 $d_k = -H_k g_k$ 。

3° 利用不精确线搜索确定步长 α_k , 令 $x_{k+1} = x_k + \alpha_k d_k$ 。

4° 令 $m = \min\{k, \hat{m}\}$, 按式(9-9)计算 H_{k+1} , 若 $k = 1$, 则 $H_1^{(0)} = H_1$; 否则 $H_k^{(0)}$ 由式(9-10)给定。

5° 令 $k = k + 1$, 转 2°。

与上述算法相应的不精确线搜索算法的计算步骤如下:

1° 给定 $0 < b_1 \leq b_2 < 1$, 令 $\alpha = 1, \alpha_1 = 0, f_1 = f(x), f'_1 = d^T \nabla f(x), \alpha_2 = +\infty, f'_2 = -1$ 。

2° 计算 $f = f(x + \alpha d)$, 若 $f_1 - f \geq -\alpha b_1 f'_1$, 则转 4°, 否则令 $\alpha_2 = \alpha, f_2 = f$ 。

3° 利用 f_1, f'_1, f_2 进行 2 次插值求 α , 即

$$\alpha = \alpha_1 + \frac{1}{2}(\alpha_2 - \alpha_1) / [1 + (f_1 - f_2) / (\alpha_2 - \alpha_1) f'_1]$$

转 2° 。

4° 计算 $f' = d^T \nabla f(x + \alpha d)$, 若 $|f'| \leq -b_2 f'_1$, 则结束; 否则, 若 $f' < 0$, 则转 6° , 否则令 $\alpha_2 = \alpha, f_2 = f, f'_2 = f'$ 。

5° 利用 f_1, f'_1, f_2, f'_2 进行 3 次插值求 α , 即

$$\alpha = \alpha_1 - f'_1(\alpha_2 - \alpha_1) / \left(\sqrt{(\beta - f'_1)^2 - f'_1 f'_2} - \beta \right)$$

其中 $\beta = 2f'_1 + f'_2 - 3(f_2 - f_1)/(\alpha_2 - \alpha_1)$ 转 2° 。

6° 若 $\alpha_2 = +\infty$, 则转 7° , 否则, 令 $\alpha_1 = \alpha, f_1 = f, f'_1 = f'$; 若 $f'_2 > 0$, 则转 5° , 否则转 3° 。

7° 利用 f_1, f'_1, f, f' 进行 3 次插值求 $\hat{\alpha}$, 即

$$\hat{\alpha} = \alpha - f(\alpha - \alpha_1) / \left(\sqrt{(\hat{\beta} - f')^2 - f' f'_1} + \hat{\beta} \right)$$

其中 $\hat{\beta} = 2f' + f'_1 - 3(f_1 - f)/(\alpha_1 - \alpha)$

令 $\alpha_1 = \alpha, f_1 = f, f'_1 = f', \alpha = \hat{\alpha}$, 转 2° 。

注 在 3° 和 5° 中要求 $\alpha \in [\alpha_1 + \lambda(\alpha_2 - \alpha_1), \alpha_2 - \tau(\alpha_2 - \alpha_1)]$, 其中 $\tau > 0$ (通常令 $\tau = 0.1$), 若 3° 和 5° 中求得的 α 不满足上式, 则令 $\alpha = \min\{\max\{\alpha, \alpha_1 + \tau(\alpha_2 - \alpha_1)\}, \alpha_2 - \tau(\alpha_2 - \alpha_1)\}$ 。在 7° 中要求 $\hat{\alpha} \in [\alpha + (\alpha - \alpha_1), \alpha + 9(\alpha - \alpha_1)]$, 若计算所得 $\hat{\alpha}$ 不满足上式, 令 $\hat{\alpha} = \min\{\max\{\hat{\alpha}, \alpha + (\alpha - \alpha_1), \alpha + 9(\alpha - \alpha_1)\}\}$ 。

关于 BFGS 方法及有限内存 BFGS 方法的详细论述可参看文献 32 和 37。

习 题 四

1. 考虑如下问题:

$$\begin{cases} \min & f(x) = 2x_1 - x_2 \\ \text{s. t.} & (x_1, x_2)^T \in S \end{cases}$$

其中

$$S = \{(x_1, x_2)^T | x_1^2 + x_2^2 \leq 1\} \cup \{(x_1, x_2)^T | x_1 \geq -1, 0 \leq x_2 \leq 1\}$$

- a. 画出此问题的可行集和等值线的图形;
 b. 利用图解法求出此问题的最优解及最优值;
 c. 分别对点 $\mathbf{x}^{(1)} = (1, 0)^T$, $\mathbf{x}^{(2)} = (-1, 1)^T$, $\mathbf{x}^{(3)} = (0, 0)^T$, $\mathbf{x}^{(4)} = (0, -1)^T$, 指出哪些约束是紧约束和松约束。

2. 考察如下问题:

$$\begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s. t.} & x_1^2 + x_2^2 \leq 5 \\ & x_1 + 2x_2 = 4 \\ & x_1, x_2 \geq 0 \end{cases}$$

验证在最优解 $\mathbf{x}^* = (4/5, 8/5)^T$ 处, K-T 条件成立, 并求在 \mathbf{x}^* 点的 Lagrange 乘子。

3. 利用 K-T 条件求解如下的优化问题:

$$\begin{cases} \min & f(x) = x_1^2 + x_2 \\ \text{s. t.} & g_1(x) = x_1^2 + x_2^2 - 9 \leq 0 \\ & g_2(x) = x_1 + x_2 - 1 \leq 0 \end{cases}$$

4. 求解如下的二次规划问题:

$$(1) \begin{cases} \min & Q(x) = x_1^2 + 2x_2^2 - 2x_1 + 4x_2 \\ \text{s. t.} & c(x) = x_1 + x_2 - 1 = 0 \end{cases}$$

$$(2) \begin{cases} \min & Q(x) = x_1^2 + 4x_2^2 - 2x_1 - x_2 \\ \text{s. t.} & x_1 + x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{cases}$$

5. 考虑如下的问题:

$$\begin{cases} \min & f(\mathbf{x}) = x_1 + \cos x_2 \\ \text{s. t.} & -x_1 \leq 0 \end{cases}$$

(1) 求此问题的最优解 \mathbf{x}^* 及最优值 $f(\mathbf{x}^*)$;

(2) 令 $\mathbf{x}^{(0)} = (0, 0)^T$, $\mathbf{d} = (0, \pm 1)^T$, 证明 $\nabla f(0)^T \mathbf{d} = 0$, 且 \mathbf{d} 为 $\mathbf{x}^{(0)}$ 点一下降方向。

* 6. 考虑如下的问题:

$$\begin{cases} \min & f(\mathbf{x}) = |x_1 - 1| + |x_2| \\ \text{s. t.} & g_1(\mathbf{x}) = x_1 + x_2 - 2 \leq 0 \\ & g_2(\mathbf{x}) = x_2 + 2 \leq 0 \\ & g_3(\mathbf{x}) = -x_1 \leq 0 \end{cases}$$

- (1) 画出可行域 S 的图形;
- (2) 依次令 $f(\mathbf{x}) = 0, 1/2, 1, 2, 3$, 作出目标函数 $f(\mathbf{x})$ 的等值线图;
- (3) 从几何上求此问题的最优解 \mathbf{x}^* 及最优值 $f(\mathbf{x}^*)$;
- (4) 问 \mathbf{x}^* 是不是 K-T 点? 为什么?

7. 用外点法求解下列问题:

$$(1) \begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s. t.} & x_1 \geq 1 \end{cases}$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = -x_1 x_2 \\ \text{s. t.} & g_1(\mathbf{x}) = -x_1 - x_2^2 + 1 \geq 0 \\ & g_2(\mathbf{x}) = x_1 + x_2 \geq 0 \end{cases}$$

$$(3) \begin{cases} \min & f(\mathbf{x}) = 4x_1 - x_2^2 - 12 \\ \text{s. t.} & h(\mathbf{x}) = 25 - x_1^2 - x_2^2 = 0 \\ & g_1(\mathbf{x}) = 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0 \\ & g_2(\mathbf{x}) = x_1 \geq 0 \\ & g_3(\mathbf{x}) = x_2 \geq 0 \end{cases}$$

8. 用内点法求解

$$(1) \begin{cases} \min & f(\mathbf{x}) = x_1 + x_2 \\ \text{s. t.} & g_1(\mathbf{x}) = -x_1^2 + x_2 \geq 0 \\ & g_2(\mathbf{x}) = x_1 \geq 0 \end{cases}$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s. t.} & 2x_1 + x_2 - 2 \leq 0 \\ & -x_2 + 1 \leq 0 \end{cases}$$

$$(3) \begin{cases} \min & f(\mathbf{x}) = (x_1 + 1)^3/3 + x_2 \\ \text{s. t.} & x_1 \geq 1, x_2 \geq 0 \end{cases}$$

9. 编写 SUMT 外点法的计算程序, 其中求解无约束优化问题用 DFP 算

法或 BFGS 算法,并计算习题 7(2)、(3)。

10. 编写 SUMT 内点法的计算程序,并计算习题 8(1)、(2)。

11. 确定箱形剖面简支梁的最佳尺寸。

已知梁长 $l = 600$ cm, 宽 $b = 60$ cm, 腹板厚 $t_s = 0.5$ cm, 均布载荷 $q = 1$ kg/m², 化为梁上的单位长度荷载为 $q_1 = 6$ kg/cm, 铝材弹性模量 $E = 7 \times 10^5$ kg/cm², 泊桑比 $\nu = 0.3$, 求翼板厚 t_F , 腹板高 h , 使梁自重最轻。由于梁自重最轻与梁横剖面积 A 最小等价, 所以可取 A 为目标函数, $t_F = x_1$, $h = x_2$ 为设计变量, 于是问题归结为求解:

$$\left\{ \begin{array}{l} \min \quad A = 2bx_1 + 2t_sx_2 \\ \text{s. t.} \quad \sigma_b = 4500/x_1x_2 \leq 700 \\ \quad \quad \tau = 1800/x_2 \leq 450 \\ \quad \quad \sigma_b \leq Ex_1^2/10000 \\ \quad \quad \delta = 56.2 \times 10^2/Ex_1x_2^2 \leq 1/400 \\ \quad \quad x_1, x_2 > 0 \end{array} \right.$$

12. 证明 4.3 的引理 1。

13. 证明 4.3 的引理 2。

14. 用混合罚函数法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{s. t.} \quad g_1(\mathbf{x}) = x_1^2/4 + x_2^2 - 1 \leq 0 \\ \quad \quad h_1(\mathbf{x}) = x_1 - 2x_2 + 1 = 0 \end{array} \right.$$

取初始点 $\mathbf{x}^{(0)} = (2, 2)^T$ 。

15. 编写混合罚函数法求解约束优化问题 (3-9) 的计算程序, 并计算习题 14。

16. 用 Zoutendijk 的可行方向法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_2 - 12x_3 \\ \text{s. t.} \quad x_1 + x_2 + x_3 = 2 \\ \quad \quad -x_1 + 2x_2 \leq 3 \\ \quad \quad x_1, x_2, x_3 \geq 0 \end{array} \right.$$

17. 用可行方向法求解

$$\left\{ \begin{array}{ll} \min & f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_2 - 12x_3 \\ \text{s. t.} & 2x_1^2 + x_2^2 \leq 15 \\ & -x_1 + 2x_2 + x_3 \leq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array} \right.$$

18. 用可行方向法求解如下问题(Kunzi, krelle and Dettli[1966]):

$$\left\{ \begin{array}{ll} \min & f(\mathbf{x}) = x_1^2/2 + x_2^2/2 - x_1 - 2x_2 \\ \text{s. t.} & 2x_1 + 3x_2 \leq 6 \\ & x_1 + 4x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{array} \right.$$

取 $\mathbf{x}^{(0)} = (0, 0)^T$ 。

$$19. \left\{ \begin{array}{ll} \min & f(\mathbf{x}) = (1 - x_1)^2 - 10(x_2 - x_1^2)^2 + x_1^2 - 2x_1x_2 \\ \text{s. t.} & x_1^2 + x_2^2 \leq 16 \\ & (x_2 - x_1)^2 + x_1 \leq 16 \\ & x_1 + x_2 \geq 2 \end{array} \right.$$

20. 画出用 Zoutendijk 方法求解问题(4-3)的程序框图,并编写计算程序。

21. 用梯度投影法求解

$$\left\{ \begin{array}{ll} \min & f(\mathbf{x}) = x_1^2/2 + x_2^2/2 - x_1 - 2x_2 \\ \text{s. t.} & 2x_1 + 3x_2 \leq 6 \\ & x_1 + 4x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{array} \right.$$

22. 用梯度投影法求解

$$\left\{ \begin{array}{ll} \min & f(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 + x_1x_2 - 2x_1x_3 + x_2x_3 - 4x_1 - 6x_2 \\ \text{s. t.} & x_1 + 2x_2 + x_3 \leq 4 \\ & x_1, x_2, x_3 \geq 0 \end{array} \right.$$

取 $\mathbf{x}^{(0)} = (0, 0, 0)^T$, $\epsilon = 10^{-3}$ 。

23. 用梯度投影法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \text{s. t.} \quad 2x_1 + x_2 + x_3 + 4x_4 = 7 \\ \quad \quad x_1 + x_2 + 2x_3 + x_4 = 6 \\ \quad \quad x_i \geqslant 0, i = 1, 2, 3, 4 \end{array} \right.$$

取 $\mathbf{x}^{(1)} = (2, 2, 1, 0)^T$, $\epsilon = 10^{-4}$ 。

24. 用梯度投影法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 + 2x_3^2 + 2x_2x_3 + \\ \quad \quad \quad 4x_1 + 6x_2 + 12x_3 \\ \text{s. t.} \quad x_1 + x_2 + x_3 \leqslant 6 \\ \quad \quad -x_1 - x_2 + 2x_3 \geqslant 2 \\ \quad \quad x_i \geqslant 0, i = 1, 2, 3 \end{array} \right.$$

取 $\mathbf{x}^{(1)} = (1, 1, 3)^T$, $\epsilon = 10^{-5}$ 。

25. 画出用梯度投影法求解线性约束优化问题的程序框图,并编写计算程序,求解习题 24。

26. 用梯度投影法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = (1 - x_1)^2 - 10(x_2 - x_1)^2 + \\ \quad \quad \quad x_1^2 - 2x_1x_2 + e^{-x_1 - x_2} \\ \text{s. t.} \quad 2x_1 + 5x_2 \leqslant 25 \\ \quad \quad -x_1 + 2x_2 \leqslant 8 \\ \quad \quad x_1, x_2 \geqslant 0 \end{array} \right.$$

27. 用既约梯度法求解

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 - 6x_1 - 2x_3 - 12x_2 \\ \text{s. t.} \quad x_1 + x_2 + x_3 = 2 \\ \quad \quad -x_1 + 2x_2 \leqslant 3 \\ \quad \quad x_1, x_2, x_3 \geqslant 0 \end{array} \right.$$

28. 用既约梯度法求解

$$\begin{cases} \min & f(\mathbf{x}) = x_1^3 + 2x_2^3 + x_1 - 2x_2 - x_1^2 \\ \text{s. t.} & x_1 + 2x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{cases}$$

29. 用既约梯度法求解

$$\begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 - 2x_1x_2 \\ \text{s. t.} & 2x_1 + x_2 - 4 = 0 \\ & 5x_1 - x_3 - 8 = 0 \\ & x_1, x_2, x_3 \geq 0 \end{cases}$$

30. 编写用既约梯度法求解线性约束优化问题的计算程序,并求解如下问题:

$$(1) \begin{cases} \min & f(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 + x_1x_2 - 2x_1x_3 + \\ & x_2x_3 - 4x_1 - 6x_2 \\ \text{s. t.} & x_1 + 2x_2 + x_3 \leq 4 \\ & x_i \geq 0, i = 1, 2, 3 \end{cases}$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = e^{-x_1-x_2} + x_1^2 + 2x_1x_2 + x_2^2 + 2x_1 + 6x_2 \\ \text{s. t.} & x_1 + x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{cases}$$

31. 求解

$$\begin{cases} \min & f(\mathbf{x}) = x_1^2 + 4x_2^2 \\ \text{s. t.} & x_1 + 2x_2 - x_3 = 1 \\ & -x_1 + x_2 + x_4 = 0 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

* 32. 画出用 GRG 法求解问题(6-9)的程序框图,编写计算程序。并用你编写的 GRG 法计算程序求解如下的问题:

$$(1) \begin{cases} \min & f(\mathbf{x}) = 4x_1 + 5x_2 \\ \text{s. t.} & g_1(\mathbf{x}) = -x_1^2 - 2x_1x_2 - 2x_2^2 + 4 \geq 0 \\ & g_2(\mathbf{x}) = -x_1^2 - x_2^2 + 4x_1 - 3 \geq 0 \end{cases}$$

$$(2) \left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = 5.357\,854\,7x_3^2 + 0.835\,689\,1x_1x_5 + \\ \quad \quad \quad 37.293\,239x_1 - 40\,792.141 \\ \text{s. t.} \quad 0 \leq 85.334\,407 + 0.005\,685\,8x_2x_3 + \\ \quad \quad \quad 0.000\,626\,2x_1x_4 - 0.002\,205\,3x_3x_5 \leq 92 \\ \quad \quad \quad 90 \leq 80.512\,49 + 0.007\,131\,7x_2x_5 + \\ \quad \quad \quad 0.002\,995\,5x_1x_2 + 0.002\,181\,3x_3^2 \leq 110 \\ \quad \quad \quad 20 \leq 9.300\,961 + 0.004\,702\,6x_3x_5 + \\ \quad \quad \quad 0.001\,254\,7x_1x_3 + 0.001\,908\,5x_3x_4 \leq 25 \\ \quad \quad \quad 78 \leq x_1 \leq 102, \quad 33 \leq x_2 \leq 45 \\ \quad \quad \quad 27 \leq x_3 \leq 45, \quad 27 \leq x_4 \leq 45 \\ \quad \quad \quad 27 \leq x_5 \leq 45 \end{array} \right.$$

33. 用乘子法求解

$$(1) \left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2 \\ \text{s. t.} \quad x_1 + x_2 = 1 \end{array} \right.$$

$$(2) \left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + 6x_1 + x_2^2 + 9 \\ \text{s. t.} \quad g_1(\mathbf{x}) = -x_1 \leq 0 \\ \quad \quad g_2(\mathbf{x}) = -x_2 \leq 0 \\ \quad \quad g_3(\mathbf{x}) = 4 - 2x_1 - x_2 \leq 0 \end{array} \right.$$

$$(3) \left\{ \begin{array}{l} \min \quad f(\mathbf{x}) = x_1^2 + 2x_2^2 - 2x_1x_2^2 \\ \text{s. t.} \quad x_1x_2 + x_1^2 + x_2^2 \leq 2 \\ \quad \quad x_1 \geq 0, x_2 \geq 0 \end{array} \right.$$

34. 画出用乘子法求解问题(7-21)的程序框图,编写计算程序。并求解如下问题:

$$(1) \left\{ \begin{array}{l} \max \quad f(\mathbf{x}) = 10x_1 + 4.4x_2^2 + 2x_3 \\ \text{s. t.} \quad x_1 + 4x_2 + 5x_3 \leq 32 \\ \quad \quad x_1 + 3x_2 + 2x_3 \leq 29 \\ \quad \quad x_3^2/2 + x_2^2 \geq 3 \\ \quad \quad x_1 \geq 2, x_2 \geq 0, x_3 \geq 0 \end{array} \right.$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = x_1^2 + x_2^2 - 16x_1 - 10x_2 \\ \text{s. t.} & g_1(\mathbf{x}) = 11 - x_1^2 + 6x_1 - 4x_2 \geq 0 \\ & g_2(\mathbf{x}) = x_1x_2 - 3x_2 - e^{x_1-3} + 1 \geq 0 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

35. 用序列二次规划法求解如下的优化问题:

$$(1) \begin{cases} \min & f(\mathbf{x}) = x_1x_2x_3x_4x_5 \\ \text{s. t.} & g_1(\mathbf{x}) = \sum_{i=1}^5 x_i^2 - 10 = 0 \\ & g_2(\mathbf{x}) = x_2x_3 - 5x_4x_5 = 0 \\ & g_3(\mathbf{x}) = x_1^3 + x_2^3 + 1 = 0 \end{cases}$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = -5x_1 - 5x_2 - 4x_3 - x_1x_3 - 6x_4 - 5x_5/(1+x_5) - \\ & 8x_6/(1+x_6) - 10(1 - 2e^{-x_7} + e^{-2x_7}) \\ \text{s. t.} & g_1(\mathbf{x}) = 2x_4 + x_5 + 0.8x_6 + x_7 - 5 = 0 \\ & g_2(\mathbf{x}) = x_2^2 + x_3^2 + x_5^2 + x_6^2 - 5 = 0 \\ & g_3(\mathbf{x}) = \sum_{i=1}^7 x_i - 10 \leq 0 \\ & g_4(\mathbf{x}) = \sum_{i=1}^4 x_i - 5 \leq 0 \\ & g_5(\mathbf{x}) = x_1 + x_3 + x_5 + x_6^2 - x_7^2 - 5 \leq 0 \\ & x_i \geq 0, i = 1, 2, \dots, 7 \end{cases}$$

36. 用极大熵方法求解如下问题:

- (1) 第 33 题第(3)题;
- (2) 第 34 题第(2)题;
- (3) 第 35 题第(1)题。

第 5 章 多目标规划

在许多实际工程问题中,衡量一个设计方案的好坏标准往往不只一个。例如:设计一个导弹,既要射程最远,又要燃料最省,还要重量最轻,精度最高。在把飞机外形设计包括在内的飞机最优设计问题中,不仅要求飞机的总重量最轻,还要求在耗油量一定的情况下,航程最远。在确定一个橡胶配方时,往往要同时考察强力、硬度、变形、伸长等多个指标。这一类问题称为多目标数学规划问题或多目标最优化问题,它们的早期来源是经济理论,现在已应用于工程优化设计、地区发展规划、数理经济学和环境保护问题等许多领域。

本章主要从实用的角度简要介绍多目标规划的一些基本概念与求解方法。对多目标规划的理论、方法和新近发展有兴趣的读者,可参看文献 41 ~ 44。

5.1 多目标规划问题举例

【例 1.1】 买糖问题

设商店有 A_1, A_2, A_3 三种糖果,单价分别为 4 元 /kg, 2.80 元 /kg 和 2.40 元 /kg。今要筹办一次节日茶话会,要求用于买糖的钱数不超过 20 元,糖的总量不少于 6kg, A_1, A_2 两种糖的总和不少于 3kg,问应如何确定最好的买糖方案?

解 设 x_1, x_2, x_3 分别为购买 A_1, A_2, A_3 三种糖的公斤数。用于买糖所花费的总钱数为 y_1 , 所买糖的总公斤数为 y_2 。我们自

然希望 y_1 取最小值, y_2 取最大值, 即

$$y_1 = 4x_1 + 2.8x_2 + 2.4x_3 \rightarrow \min$$

$$y_2 = x_1 + x_2 + x_3 \rightarrow \max$$

而约束条件为

$$4x_1 + 2.8x_2 + 2.4x_3 \leq 20$$

$$x_1 + x_2 + x_3 \geq 6$$

$$x_1 + x_2 \geq 3$$

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

易见, 这是一个包含两个目标的 LP 问题, 称为多目标 LP 问题。由于求 y_2 的最大值可转化为求 $(-y_2)$ 的最小值, 所以上述问题可归结为

$$\left\{ \begin{array}{ll} \text{V-min} & (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \\ \text{s. t.} & 4x_1 + 2.8x_2 + 2.4x_3 \leq 20 \\ & x_1 + x_2 + x_3 \geq 6 \\ & x_1 + x_2 \geq 3 \\ & x_1, x_2, x_3 \geq 0 \end{array} \right. \quad (1-1)$$

其中

$$f_1(\mathbf{x}) = y_1, f_2(\mathbf{x}) = -y_2, \mathbf{x} = (x_1, x_2, x_3)^T$$

符号 V-min 表示求向量函数 $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T$ 的最小值。而一般的多目标数学规划问题可以写成

$$\left\{ \begin{array}{ll} \text{V-min} & f(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \end{array} \right. \quad (1-2)$$

其中

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

$$f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x}))^T, p \geq 2$$

令

$$R = \{\mathbf{x} | g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m\}$$

称 R 为问题(1-2)的可行集或约束集, $x \in R$ 称为问题(1-2)的可行解或容许解。多目标规划问题与前面讲的 LP 与 NLP 问题的主要区别在于: 它所追求的目标不只一个, 而是 p 个 ($p \geq 2$)。 ■

【例 1.2】 生产计划问题

某纺织厂生产 A_1 、 A_2 、 A_3 三种布料, 该厂两班生产, 每周生产时间为 90h, 每周的能耗不得超过 150t 标准煤, 其他的数据如下表:

布 料 名	生产数量 (m/h)	利 润 (元 /m)	最大销量 (m/ 周)	能 耗 (t/1000m)
A_1	400	0.25	40000	1.2
A_2	500	0.20	48000	1.3
A_3	360	0.30	30000	1.4

问每周应生产三种布料各多少米, 才能使该厂的利润最多, 而能源消耗最少?

解 设该厂每周生产三种布料 A_1 、 A_2 、 A_3 的小时数分别为 x_1 、 x_2 、 x_3 , 总利润为 y_1 (元), 总能源消耗量为 y_2 (t 标准煤), 则上述问题可归结为如下的多目标 LP 问题:

$$\left\{ \begin{array}{l} \text{V-min } f(x) = (f_1(x), f_2(x))^T \\ \text{s.t. } x_1 + x_2 + x_3 \leq 90 \\ 1.2 \times 10^{-3} \times 400x_1 + 1.3 \times 10^{-3} \times 500x_2 + \\ 1.4 \times 10^{-3} \times 360x_3 \leq 150 \\ 400x_1 \leq 40\,000, 500x_2 \leq 48\,000 \\ 360x_3 \leq 30\,000 \\ x_1, x_2, x_3 \geq 0 \end{array} \right. \quad (1-3)$$

其中

$$\begin{aligned} f_1(x) &= -y_1, f_2(x) = y_2, x = (x_1, x_2, x_3)^T \\ y_1 &= 0.25 \times 400x_1 + 0.2 \times 500x_2 + 0.3 \times 360x_3 \end{aligned}$$

$$y_2 = 1.2 \times 10^{-3} \times 400x_1 + 1.3 \times 10^{-3} \times 500x_2 + 1.4 \times 10^{-3} \times 360x_3$$

【例 1.3】 光学系统的自动设计

一个光学系统,如照相机头,显微镜的物镜和目镜,望远镜及电影放映机的镜头等,一般是由若干个球面透镜组成的,制造这些透镜的材料是具有不同折射率的光学玻璃。各个镜面都有一定大小的半径和厚度,各镜面之间保持一定的间隔。一个光学系统中各个透镜镜面的半径、厚度、镜面间距及各透镜所用光学玻璃的折射率等,称为这一光学系统的结构参数。这些参数的大小,直接影响光学系统的成像质量。所谓一个光学系统的自动设计问题就是根据系统的要求,设法自动确定(由计算机计算)各个结构参数的大小,使系统的成像质量最好,而与成像质量有关的是要考察若干个像差指标 $f_i(\mathbf{x}) (i = 1, 2, \dots, m)$ 。对于一个光学系统,其像差指标值可以事先选定,设为 $f_i^* (i = 1, 2, \dots, m)$ 。因此,光学系统的优化设计问题可以化为如下的多目标规划问题:

$$\begin{cases} \text{V-min} & f(\mathbf{x}) = (f_1(\mathbf{x}) - f_1^*, \dots, f_m(\mathbf{x}) - f_m^*)^T \\ \text{s. t.} & \mathbf{x} \in A \end{cases} \quad (1-4)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 为结构参数, $\mathbf{x} \in A$ 表示对结构参数 x_1, x_2, \dots, x_n 的某些限制条件。

问题(1-4)是一个含有 m 个目标的多目标非线性规划问题。

由上面的几个例子可以看出:在许多实际问题中,各个目标的量纲一般是不相同的,所以有必要把每个目标事先规范化,例如,对第 j 个带量纲的目标 $F_j(\mathbf{x})$, 可令

$$f_j(\mathbf{x}) = F_j(\mathbf{x})/F_j$$

其中 $F_j = \min_{\mathbf{x} \in R} F_j(\mathbf{x})$;

$$R = \{\mathbf{x} | g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m)\}$$

这样, $f_j(\mathbf{x})$ 就是规范化的目标了。因此,在以后的叙述中,我们不

妨假设多目标规划问题(1-2) 中的目标都是已经规范化了的。

5.2 多目标规划问题的解集和像集

5.2.1 多目标规划问题的解集

考察如下的多目标规划问题：

$$\begin{cases} \text{V-min} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x}))^T \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \end{cases} \quad (2-1)$$

其中

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, p \geq 2$$

令

$$R = \{\mathbf{x} | g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m\}$$

定义 2.1 设 $\mathbf{x}^* \in R$, 若对任意 $\mathbf{x} \in R$ 及 $i = 1, 2, \dots, p$, 都有 $f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})$ 成立, 则称 \mathbf{x}^* 为问题(2-1) 的绝对最优解。而 $\mathbf{F}^* = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_p(\mathbf{x}^*))^T$ 称为绝对最优值。

例如：若 $f_1(x) = x^2$, $f_2(x) = x^2 + 1$, 则问题

$$\begin{cases} \text{V-min} & \mathbf{F}(x) = (f_1(x), f_2(x))^T \\ \text{s. t.} & -4 \leq x \leq 4 \end{cases} \quad (2-2)$$

的最优解为 $\mathbf{x}^* = \mathbf{0}$, 绝对最优值为 $\mathbf{F}^* = (0, 1)^T$ 。

若 $f_1(\mathbf{x}) = x_1^2 + x_2^2$, $f_2(\mathbf{x}) = x_1^2 + x_2^2 + 1$, 则问题

$$\begin{cases} \text{V-min} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \\ \text{s. t.} & -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1 \end{cases} \quad (2-3)$$

的绝对最优解为 $\mathbf{x}^* = (0, 0)^T$, 绝对最优值为 $\mathbf{F}^* = (0, 1)^T$ 。

若 $f_1(x) = x^2$, $f_2(x) = (x - 1)^2$, 则问题(2-2) 没有绝对最优解, 或者说它的绝对最优解不存在。因此, 要寻找另外意义下的“解”, 为此, 我们引入如下的符号：

(1) 符号 $<$

令

$$F(x^{(1)}) = (f_1(x^{(1)}), \dots, f_p(x^{(1)}))^T$$

$$F(x^{(2)}) = (f_1(x^{(2)}), \dots, f_p(x^{(2)}))^T$$

$F(x^{(1)}) < F(x^{(2)})$ 等价于 $f_j(x^{(1)}) < f_j(x^{(2)})$ ($j = 1, 2, \dots, p$)。

(2) 符号 \leq

$F(x^{(1)}) \leq F(x^{(2)})$ 等价于 $f_j(x^{(1)}) \leq f_j(x^{(2)})$ ($j = 1, 2, \dots, p$), 且至少存在某个 j_0 ($1 \leq j_0 \leq p$), 使 $f_{j_0}(x^{(1)}) < f_{j_0}(x^{(2)})$ 。

(3) 符号 \leq

$F(x^{(1)}) \leq F(x^{(2)})$ 等价于 $f_j(x^{(1)}) \leq f_j(x^{(2)})$ ($j = 1, 2, \dots, p$)。

定义 2.2 设 $x^* \in R$, 若不存在 $x \in R$, 满足 $F(x) \leq F(x^*)$, 则称 x^* 为问题(2-1)的有效解(或者称为 Pareto 解), 亦称为非劣解。

定义 2.3 设 $x^* \in R$, 若不存在 $x \in R$, 满足 $F(x) < F(x^*)$, 则称 x^* 为问题(2-1)的弱有效解(或者称为弱 Pareto 解), 亦称为弱非劣解。

我们用 R^* 表示问题(2-1)的绝对最优解所成的集合, 用 R_{pa}^* 表示问题(2-1)的有效解所成的集合, 用 R_{wp}^* 表示问题(2-1)的弱有效解所成的集合, 易见 $x^* \in R_{pa}^*$, 意味着找不到一个可行解 $x \in R$, 使 $F(x) = (f_1(x), f_2(x), \dots, f_p(x))^T$ 的每一个目标函数值都不比 $F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_p(x^*))^T$ 的相应目标函数值“坏”, 并且 $F(x)$ 至少有一个目标函数值比 $F(x^*)$ 的相应目标函数值“好”。即当 $x^* \in R_{pa}^*$ 时, x^* 在“ \leq ”意义下, 已找不到另一个可“改进”的可行解 x 了。图 5-1 说明了 R_{pa}^* 的几何意义。

$x^* \in R_{wp}^*$, 意味着找不到一个可行解 $x \in R$, 使得 $F(x) = (f_1(x), f_2(x), \dots, f_p(x))^T$ 的每一个目标函数值都比 $F(x^*) =$

$(f_1(x^*), f_2(x^*), \dots, f_p(x^*))^T$ 的相应的目标函数值严格地“好”，即当 $x^* \in R_{wp}^*$ 时， x^* 在“ $<$ ”意义下，已找不到另一个可“改进”的可行解 x 了。

如果用 $R_j^* (j = 1, 2, \dots, p)$ 表示单目标规划问题

$$(P_j) \begin{cases} \min & f_j(x) \\ \text{s. t.} & g_i(x) \leq 0, i = 1, 2, \dots, m \end{cases}$$

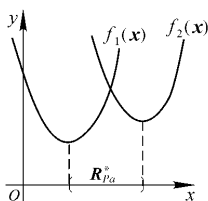


图 5-1

的最优解所成的集合。则显然有

$$R^* = \bigcap_{j=1}^p R_j^* \quad (2-4)$$

而 $R^*, R_{pa}^*, R_{wp}^*, R, R_1^*, R_2^*, \dots, R_p^*$ 之间的关系由下面的几个定理给出。

定理 2.1 $R_{pa}^* \subset R_{wp}^* \subset R$ 。

证明 $R_{wp}^* \subset R$ 是明显的，所以只需证 $R_{pa}^* \subset R_{wp}^*$ 。用反证法证明，假设有 $x \in R_{pa}^*$ ，但 $x \notin R_{wp}^*$ 。由定义 2.3 可知，必存在 $y \in R$ ，使 $F(y) < F(x)$ ，即对 $j = 1, 2, \dots, p$ ，有 $f_j(y) < f_j(x)$ 成立，所以 y 满足 $F(y) \leq F(x)$ ，由此推出 $x \notin R_{pa}^*$ ，这与假设相矛盾，因此应有 $x \in R_{wp}^*$ ，即 $R_{pa}^* \subset R_{wp}^*$ 成立。 ■

定理 2.2 $R_j^* \subset R_{wp}^*, j = 1, 2, \dots, p$ 。

定理 2.3 $R^* \subset R_{pa}^*$ 。

这两个定理的证明都不难，留作习题。由上述定理得知：

$$R^* \subset R_{pa}^* \subset R_{wp}^* \subset R \quad (2-5)$$

$$R_j^* \subset R_{wp}^*, j = 1, 2, \dots, p \quad (2-6)$$

定义 2.4 若在问题(2-1)中， $f_1(x), f_2(x), \dots, f_p(x)$ 和 $g_1(x), g_2(x), \dots, g_m(x)$ 均为凸函数，则称(2-1)为凸多目标规划。

一般来说,对于凸多目标规划(2-1),其 R_{wp}^* 和 R_{pa}^* 也不一定是凸集。

5.2.2 多目标规划问题的像集

在(2-1)中,取定一可行解 $x^{(0)} \in R$,得到对应的目标函数值 $F(x^{(0)}) = (f_1(x^{(0)}), f_2(x^{(0)}), \dots, f_p(x^{(0)}))^T$, $F(x^{(0)})$ 可视为 p 维欧氏空间 \mathbf{R}^p 中的一个点。一般的,对于任意 $x \in R$,都可得到 \mathbf{R}^p 中的一个点 $F(x)$,于是我们可以定义一个映射 F :

$$x \xrightarrow{F} F(x)$$

令

$$F(R) = \{F(x) | x \in R\}$$

称 $F(R)$ 为可行集 R 在映射 F 之下的像集。对任意 $x^{(0)} \in R$,称 $F(x^{(0)}) \in F(R)$ 为点 $x^{(0)}$ 在映射 F 之下的像。若某 $F^0 \in F(R)$,则至少存在一个点 $x^{(0)} \in R$,使 $F(x^{(0)}) = F^0$,称 $x^{(0)}$ 为 F^0 的一个原像。

一般来说,对于凸多目标规划(2-1),其像集 $F(R)$ 不一定是凸集。

定义 2.5 设 $\bar{F} \in F(R)$,若不存在 $F \in F(R)$,满足 $F \leq \bar{F}$,则称 \bar{F} 为像集 $F(R)$ 的有效点,用 E_{pa}^* 表示全体有效点所成的集合。

定义 2.6 设 $\bar{F} \in F(R)$,若不存在 $F \in F(R)$,满足 $F < \bar{F}$,则称 \bar{F} 为像集 $F(R)$ 的弱有效点,用 E_{wp}^* 表示全体弱有效点所成的集合。

易证:像集 $F(R)$ 的有效点一定是弱有效点,即恒有 $E_{pa}^* \subset E_{wp}^*$ 成立。

通过对像集 $F(R)$ 的研究,可以启发我们提出一些处理多目

标规划问题(2-1)的方法,对此有兴趣的读者可参看文献 43。

5.3 处理多目标规划问题的一些方法

本节介绍几种常用的处理多目标规划问题的方法。

5.3.1 约束法

在问题(2-1)中,从 p 个目标函数 $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})$ 中,如果能够确定出一个主要目标,例如 $f_1(\mathbf{x})$,而对其他的目标函数 $f_2(\mathbf{x}), \dots, f_p(\mathbf{x})$ 只要求满足一定的条件即可,例如要求

$$a_i \leq f_i(\mathbf{x}) \leq b_i, i = 2, 3, \dots, p$$

这样,我们就可以把其他目标当做约束来处理,则问题(2-1)可化为求解如下的非线性规划问题:

$$\begin{cases} \min & f_1(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & a_j \leq f_j(\mathbf{x}) \leq b_j, j = 2, 3, \dots, p \end{cases} \quad (3-1)$$

5.3.2 分层序列法

求解问题(2-1)的分层序列法是把其中的 p 个目标,按其重要程度排一个次序,例如不妨设问题(2-1)中的 p 个目标的次序已排好: $f_1(\mathbf{x})$ 最重要, $f_2(\mathbf{x})$ 次之, $f_3(\mathbf{x})$ 再次之, \dots , 最后一个目标为 $f_p(\mathbf{x})$ 。先求出问题

$$\begin{cases} \min & f_1(\mathbf{x}) \\ \text{s. t.} & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \end{cases} \quad (P_1)$$

的最优解 $\mathbf{x}^{(1)}$ 及最优值 f_1^* 。即

$$\min_{\mathbf{x} \in R} f_1(\mathbf{x}) = f_1^*$$

其中

$$R = \{\mathbf{x} | g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m\}$$

再求解问题

$$\begin{cases} \min & f_2(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in R_1 \end{cases} \quad (P_2)$$

其中

$$R_1 = R \cap \{\mathbf{x} | f_1(\mathbf{x}) \leq f_1^*\}$$

设问题 (P_2) 的最优解为 $\mathbf{x}^{(2)}$, 最优值为

$$f_2^* = \min_{\mathbf{x} \in R_1} f_2(\mathbf{x})$$

继续求解问题

$$\begin{cases} \min & f_3(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in R_2 \end{cases} \quad (P_3)$$

其中

$$R_2 = R_1 \cap \{\mathbf{x} | f_2(\mathbf{x}) \leq f_2^*\}$$

…… 如此继续下去, 直到求出第 p 个问题

$$\begin{cases} \min & f_p(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in R_{p-1} \end{cases} \quad (P_p)$$

的最优解 $\mathbf{x}^{(p)}$ 及最优值 f_p^* 。其中

$$R_{p-1} = R_{p-2} \cap \{\mathbf{x} | f_{p-1}(\mathbf{x}) \leq f_{p-1}^*\}$$

这样求得的 $\mathbf{x}^{(p)}$ 就是问题 (2-1) 在分层序列意义下的最优解, 即 $\mathbf{x}^* = \mathbf{x}^{(p)}$, 而

$$\mathbf{F}^* = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_p(\mathbf{x}^*))^T$$

为问题 (2-1) 的最优值, 可以证明: \mathbf{x}^* 是问题 (2-1) 的有效解, 即有如下定理。

定理 3.1 设 \mathbf{x}^* 是由分层序列法求得的问题 (2-1) 的最优解, 则必有 $\mathbf{x}^* \in R_{pa}^*$ 。

证明 用反证法证明。

假设 $\mathbf{x}^* \in R_{pa}^*$, 则必存在 $\mathbf{y} \in R$, 使 $F(\mathbf{y}) \leq F(\mathbf{x}^*)$ 。下面分两种情况讨论。

1° 若 $f_1(\mathbf{y}) < f_1(\mathbf{x}^*)$, 则因为 $f_1(\mathbf{x}^*) = f_1^*$, 于是我们得到了 (P_1) 的一个可行解 \mathbf{y} , 满足

$$f_1(\mathbf{y}) < f_1(\mathbf{x}^*) = f_1^*$$

这与 $f_1^* = \min_{\mathbf{x} \in R} f_1(\mathbf{x})$ 相矛盾。

2° 若 $f_j(\mathbf{x}) = f_j(\mathbf{x}^*)$, $j = 1, 2, \dots, j_0 - 1$, 而

$$f_{j_0}(\mathbf{y}) < f_{j_0}(\mathbf{x}^*) \quad (2 \leq j_0 \leq p)$$

这时, 由于 $f_j(\mathbf{x}^*) \leq f_j^* \quad (j = 1, 2, \dots, j_0 - 1)$, 所以

$$f_j(\mathbf{y}) = f_j(\mathbf{x}^*) \leq f_j^*, \quad j = 1, 2, \dots, j_0 - 1$$

因此, \mathbf{y} 是问题

$$\begin{cases} \min & f_{j_0}(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in R_{j_0-1} \end{cases} \quad (P_{j_0})$$

的一个可行解, 又因 $f_{j_0}(\mathbf{y}) < f_{j_0}(\mathbf{x}^*) = f_{j_0}^*$, 这与 $f_{j_0}^*$ 是问题 (P_{j_0}) 的最优值相矛盾。

综合 1°, 2°, 即知定理 3.1 的结论成立。 ■

对上述的分层序列法稍加分析即可看出: 若对某个问题 (P_i) , 其最优解是惟一的, 则问题 $(P_{i+1}), \dots, (P_p)$ 的最优解也是惟一的, 且 $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i+2)} = \dots = \mathbf{x}^{(p)} = \mathbf{x}^{(i)}$ 。因此, 常将分层序列法修改如下: 选取一组适当的小正数 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{p-1}$, 称为宽容值, 即按照各个目标函数的不同要求, 预先给定关于相应目标函数最优值的允许误差, 将问题 (P_j) 修改为

$$\begin{cases} \min & f_j(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in R_{j-1} = R_{j-2} \cap \{\mathbf{x} | f_{j-1}(\mathbf{x}) \leq f_{j-1}^* + \varepsilon_{j-1}\} \quad (P_j) \\ & j = 2, 3, \dots, p \end{cases}$$

再按上述的分层序列法依次求解各个问题 (P_2) , $(P_3), \dots$, (P_p) 。

5.3.3 评价函数法

评价函数法是一大类方法,它的基本思想是:针对多目标规划问题(2-1)构造一个评价函数 $h(F(x))$,然后求解问题

$$\begin{cases} \min & h(F(x)) \\ \text{s. t.} & x \in R \end{cases} \quad (3-2)$$

用(3-2)的最优解 x^* 作为问题(2-1)的最优解,由于可以用不同的方法来构造评价函数,因此有各种不同的评价函数方法。下面介绍常用的几种方法。

1. 理想点法

在问题(2-1)中,先分别求解 p 个单目标规划问题

$$\begin{cases} \min & f_i(x), \quad i = 1, 2, \dots, p \\ \text{s. t.} & g_j(x) \leq 0, \quad j = 1, 2, \dots, m \end{cases} \quad (3-3)$$

令 $f_i^* = \min_{x \in R} f_i(x), i = 1, 2, \dots, p$, 其中

$$R = \{x | g_i(x) \leq 0, i = 1, 2, \dots, m\}$$

构造评价函数

$$h(x) \triangleq h(F(x)) = \sqrt{\sum_{i=1}^p (f_i(x) - f_i^*)^2} \quad (3-4)$$

再求解问题(3-2),取其最优解 x^* 作为问题(2-1)的最优解。 $h(x)$ 也可以取为更一般的形式:

$$h(x) = \left[\sum_{i=1}^p (f_i(x) - f_i^*)^q \right]^{(1/q)} (q > 1 \text{ 为整数})$$

以上求得的向量 $F^* = (f_1^*, f_2^*, \dots, f_p^*)^T$, 只是一个理想点,一般不能达到它。上面介绍的是 1975 年 Салуквадзе 提出的方法,其中心思想是定义一种模,在这种模的意义下,找一个点,尽量

接近理想点 F^* , 使 $h(x) = \|F(x) - F^*\|_q \rightarrow \min$, 故称它为理想点法。

2. 平方和加权法

先求出各个单目标规划问题(3-3)的一个尽可能好的下界

$f_1^0, f_2^0, \dots, f_p^0$, 即

$$\min_{x \in R} f_i(x) \geq f_i^0, i = 1, 2, \dots, p$$

然后构造评价函数

$$h(x) = h(F(x)) = \sum_{i=1}^p \lambda_i (f_i(x) - f_i^0)^2 \quad (3-5)$$

其中 $\lambda_1, \lambda_2, \dots, \lambda_p$ 为选定的一组权系数, 它们满足

$$\sum_{i=1}^p \lambda_i = 1, \quad \lambda_i > 0, i = 1, 2, \dots, p \quad (3-6)$$

再求出问题(3-2)的最优解 x^* 作为(2-1)的最优解。权系数 $\lambda_1, \lambda_2, \dots, \lambda_p$ 的选取方法将在 5.3.5 中讲述。

3. 线性加权和法

对问题(2-1)中的 p 个目标函数 $f_1(x), f_2(x), \dots, f_p(x)$ 按

其重要程度给以适当的权系数 $\lambda_i \geq 0$ ($i = 1, 2, \dots, p$), 且 $\sum_{i=1}^p \lambda_i =$

1, 然后构造评价函数 $h(x) = h(F(x)) = \sum_{i=1}^p \lambda_i f_i(x)$ 作为新的目标函数, 再求解问题(3-2)得最优解 x^* , 以 x^* 作为原问题(2-1)的最优解。由于这种方法简单易行, 计算量小, 常为实际工作者采用。

4. min-max 法(极小 - 极大法)

在对策论中, 作决策时, 常常要考虑在最不利的情况下找出一个最有利的策略方案, 这就是所谓极小 - 极大方法。按照这种思想, 可以构造评价函数

$$h(F(x)) = \max_{1 \leq j \leq p} \{f_j(x)\} \quad (3-7)$$

然后求解问题

$$\begin{cases} \min & h(F(x)) = \min_{x \in R} \{ \max_{1 \leq j \leq p} f_j(x) \} \\ \text{s. t.} & x \in R \end{cases} \quad (3-2')$$

得最优解 x^* , 以 x^* 作为问题(2-1)的最优解, 也可选取一组适当的

的权系数 $\lambda_1, \lambda_2, \dots, \lambda_p$, 使它们满足 $\lambda_i \geq 0$ ($i = 1, 2, \dots, p$) 和 $\sum_{i=1}^p \lambda_i = 1$, 然后将评价函数定义为

$$h(F(x)) = \max_{1 \leq j \leq p} \{ \lambda_j f_j(x) \} \quad (3-8)$$

5. 乘法法

在问题(2-1)中, 设对任意 $x \in R$, 各目标函数值均满足 $f_j(x) > 0$, $j = 1, 2, \dots, p$ 。

现将目标函数分为两类, 不妨设其分别为

$$1^\circ \quad f_1(x), f_2(x), \dots, f_t(x) \rightarrow \min$$

$$2^\circ \quad f_{t+1}(x), f_{t+2}(x), \dots, f_p(x) \rightarrow \max$$

则可构造评价函数

$$h(F(x)) = \left[\prod_{j=1}^t f_j(x) \right] / \left[\prod_{j=t+1}^p f_j(x) \right] \quad (3-9)$$

然后求解问题(3-2), 即可得到问题(2-1)的最优解。

5.3.4 对评价函数法的补充说明

上一段通过构造评价函数, 把求解多目标规划问题(2-1)转化为求解单目标规划问题(3-2), 从而得到了原问题(2-1)在某种意义下的最优解 x^* 。我们要问: x^* 是问题(2-1)的有效解(或弱有效解)吗? 如果 x^* 既不是有效解, 也不是弱有效解, 那么 x^* 一定可以改进, 而上述解法(评价函数法)就没有什么意义了。下面将说明当 $h(F(x))$ 满足某种单调性质时, x^* 一定是问题(2-1)的有效解或弱有效解, 为此, 引入如下定义和定理。

定义 3.1 若对任意 $F, \bar{F} \in R^p$, 且 $F \leq \bar{F}$, 都有 $h(F) < h(\bar{F})$ 成立, 则称 $h(F)$ 是 F 的严格单调增函数。

定义 3.2 若对任意 $F, \bar{F} \in R^p$, 且 $F < \bar{F}$, 都有 $h(F) < h(\bar{F})$ 成立, 则称 $h(F)$ 是 F 的单调增函数。

定理 3.2 设 $F \in R^p$, 若 $h(F)$ 是 F 的严格单调增函数, 则问题 (3-2) 的最优解 $x^* \in R_{pa}^*$ 。

定理 3.3 设 $F \in R^p$, 若 $h(F)$ 是 F 的单调增函数, 则问题 (3-2) 的最优解 $x^* \in R_{wp}^*$ 。

这两个定理的证明是类似的, 我们只证明定理 3.2。

证明 用反证法证明。设 $x^* \in R_{pa}^*$, 则必存在 $y \in R$, 使 $F(y) \leq F(x^*)$ 。由于 $h(F)$ 是 F 的严格单调增函数, 所以有 $h(F(y)) < h(F(x^*))$, 而这与 x^* 是问题 (3-2) 的最优解相矛盾。 ■

下面来说明上一段讲的几种评价函数 $h(F)$ 均为严格单调增函数或单调增函数, 于是由定理 3.2 或定理 3.3 知 $x^* \in R_{pa}^*$ 或 $x^* \in R_{wp}^*$ 。

1. 理想点法

$$h(F) = \left[\sum_{j=1}^p (f_j(x) - f_j^*)^q \right]^{1/q}$$

其中 $q \geq 2$, 且为整数, $f_j(x) \geq f_j^*$, $j = 1, 2, \dots, p$ 。

若 $F \leq \bar{F}$, 由 $F \geq F^*$, $\bar{F} \geq F^*$, 可得

$$\bar{f}_j - f_j^* \geq f_j - f_j^* \geq 0, \quad j = 1, 2, \dots, p$$

且至少存在某个 $j_0 (1 \leq j_0 \leq p)$, 使

$$\bar{f}_{j_0} - f_{j_0}^* > f_{j_0} - f_{j_0}^*$$

因此 $h(F) < h(\bar{F})$, 即 $h(F)$ 为 F 的严格单调增函数。

2. 平方和加权法

$$h(F) = \sum_{j=1}^p \lambda_j (f_j - f_j^0)^2$$

其中 $\lambda_j > 0$ ($j = 1, 2, \dots, p$), 且 $\sum_{j=1}^p \lambda_j = 1$, $f_j \geq f_j^0$ ($j = 1, 2, \dots, p$). 用与上面类似的证法, 易证 $h(F)$ 为 F 的严格单调增函数, 具体过程请读者自己写出。

3. 线性加权和法

$$h(F) = \sum_{j=1}^p \lambda_j f_j$$

其中 $\lambda_j \geq 0$ ($j = 1, 2, \dots, p$), 且 $\sum_{j=1}^p \lambda_j = 1$ 。下面将证明:

1° 当 $\lambda_i > 0$ ($j = 1, 2, \dots, p$) 时, $h(F)$ 为严格单调增函数。

2° 当 $\lambda_j \geq 0$ ($j = 1, 2, \dots, p$) 时, $h(F)$ 为单调增函数。

证明 1° 因为 $\lambda_i > 0$ ($j = 1, 2, \dots, p$), 且 $F \leq \bar{F}$, 所以

$$\lambda_j f_j \leq \lambda_j \bar{f}_j, \quad j = 1, 2, \dots, p \quad (3-10)$$

由 $F \leq \bar{F}$ 可知: 至少存在某个 j_0 ($1 \leq j_0 \leq p$), 使

$$f_{j_0} < \bar{f}_{j_0}$$

于是

$$\lambda_{j_0} f_{j_0} < \lambda_{j_0} \bar{f}_{j_0} \quad (3-11)$$

由式(3-10)和(3-11)可得

$$h(F) = \sum_{j=1}^p \lambda_j f_j < \sum_{j=1}^p \lambda_j \bar{f}_j = h(\bar{F})$$

2° 因为 $\lambda_j \geq 0$ ($j = 1, 2, \dots, p$), 且 $F < \bar{F}$, 所以

$$\lambda_j f_j \leq \lambda_j \bar{f}_j, \quad j = 1, 2, \dots, p \quad (3-12)$$

又因为 $\sum_{j=1}^p \lambda_j = 1$, 所以至少存在某个 j_0 ($1 \leq j_0 \leq p$), 使 $\lambda_{j_0} > 0$ 。于是有

$$\lambda_{j_0} f_{j_0} < \lambda_{j_0} \bar{f}_{j_0} \quad (3-13)$$

由式(3-12)和(3-13)可得

$$h(F) = \sum_{j=1}^p \lambda_j f_j < \sum_{j=1}^p \lambda_j \bar{f}_j = h(\bar{F}) \quad \blacksquare$$

用类似的方法,可以证明:在 min-max 法中的评价函数 $h(F)$ 是 F 的单调增函数,在乘除法中的评价函数 $h(F)$ 是 F 的严格单调增函数。

综上所述,用上面的五种方法求得单目标规划问题(3-2)的最优解 x^* 都是问题(2-1)的有效解或弱有效解,即必有 $x^* \in R_{pa}^*$ 或 $x^* \in R_{wp}^*$ 。

5.3.5 确定权系数的方法

在这一段中,介绍评价函数 $h(F(x))$ 的表达式中包含的权系数 $\lambda_1, \lambda_2, \dots, \lambda_p$ 的确定方法。

1. 老手法

老手是指有关方面的专家、有经验的工人和干部等,邀请一批老手(N 个),请他们各自独立地填写如下的调查表(表 5-1)。

表 5-1

目标 权系数 老手	$f_1(x)$	$f_2(x)$	\dots	$f_p(x)$
	λ_1	λ_2	\dots	λ_p
1	λ_{11}	λ_{12}	\dots	λ_{1p}
\vdots	\vdots	\vdots		\vdots
N	λ_{N1}	λ_{N2}	\dots	λ_{Np}

表中的 λ_{ij} 是第 i 个老手对第 j 个目标 $f_j(x)$ 给出的权系数($i = 1, 2, \dots, N; j = 1, 2, \dots, p$)。在表 5-1 填好后,用下式可算出权系数 $\lambda_j(j = 1, 2, \dots, p)$ 的平均值(数学期望)。

$$\lambda_j = \frac{1}{N} \sum_{i=1}^N \lambda_{ij}, \quad j = 1, 2, \dots, p \quad (3-14)$$

然后对每个老手 $i(1 \leq i \leq N)$, 算出其估值 λ_{ij} 与平均值 λ_j 的偏

差,即

$$\delta_{ij} = |\lambda_{ij} - \lambda_j|, j = 1, 2, \dots, p \quad (3-15)$$

再请偏差最大的老手发表意见,通过充分讨论后,再对权系数作适当调整,以便获得较为可靠的数据。

2. α -方法

α -方法是由 Карпелевиц 等在 1975 年提出的,为便于理解,我们先介绍 $p = 2$ 情形的 α -方法。首先求出问题

$$\begin{cases} \min & f_i(\mathbf{x}), i = 1, 2 \\ \text{s. t.} & \mathbf{x} \in R \end{cases} \quad (P_1)$$

的最优解,设为 $\mathbf{x}^{(i)} (i = 1, 2)$ 。令

$$f_1^1 = f_1(\mathbf{x}^{(1)}), \quad f_2^1 = f_2(\mathbf{x}^{(1)})$$

$$f_1^2 = f_1(\mathbf{x}^{(2)}), \quad f_2^2 = f_2(\mathbf{x}^{(2)})$$

设过点 $(f_1^1, f_2^1)^T$ 和 $(f_1^2, f_2^2)^T$ 的直线方程为

$$\lambda_1 f_1 + \lambda_2 f_2 = \beta \quad (3-16)$$

其中系数 $\lambda_1, \lambda_2, \beta$ 待定,不妨假设

$$\lambda_1 + \lambda_2 = 1 \quad (3-17)$$

将点 $(f_1^1, f_2^1)^T$ 和 $(f_1^2, f_2^2)^T$ 的坐标代入(3-16)得

$$\begin{cases} \lambda_1 f_1^1 + \lambda_2 f_2^1 = \beta \\ \lambda_1 f_1^2 + \lambda_2 f_2^2 = \beta \end{cases} \quad (3-18)$$

若问题(2-1)不存在绝对最优解,则有

$$f_1^2 = f_1(\mathbf{x}^{(2)}) > f_1(\mathbf{x}^{(1)}) = f_1^1$$

$$f_2^1 = f_2(\mathbf{x}^{(1)}) > f_2(\mathbf{x}^{(2)}) = f_2^2$$

因为若 $f_2^2 = f_1^1$, 则说明 $\mathbf{x}^{(2)}$ 也是问题 (P_1) 的最优解。由于 $\mathbf{x}^{(2)}$ 已是问题 (P_2) 的最优解,因此 $\mathbf{x}^{(2)}$ 是问题(2-1)的绝对最优解。这与假设问题(2-1)不存在绝对最优解相矛盾。

由方程(3-17)和(3-18)联立,可求出权系数

$$\begin{cases} \lambda_1 = (f_2^1 - f_2^2) / [(f_1^2 - f_1^1) + (f_2^1 - f_2^2)] \\ \lambda_2 = (f_1^2 - f_1^1) / [(f_1^2 - f_1^1) + (f_2^1 - f_2^2)] \end{cases} \quad (3-19)$$

对于一般的具有 p ($p \geq 2$) 个目标的情况,可以完全类似地求出。

首先求出 p 个问题

$$\begin{cases} \min & f_i(\mathbf{x}), i = 1, 2, \dots, p \\ \text{s. t.} & \mathbf{x} \in R \end{cases} \quad (P_i)$$

的最优解,记为 $\mathbf{x}^{(i)}$ ($i = 1, 2, \dots, p$)。令

$$f_j^i = f_j(\mathbf{x}^{(i)}), j = 1, 2, \dots, p; i = 1, 2, \dots, p$$

设经过 p 个点 $(f_1^i, f_2^i, \dots, f_p^i)^T$ ($i = 1, 2, \dots, p$) 的超平面方程为

$$\lambda_1 f_1 + \lambda_2 f_2 + \dots + \lambda_p f_p = \beta$$

其中

$$\lambda_1 + \lambda_2 + \dots + \lambda_p = 1$$

于是有

$$\begin{cases} \sum_{j=1}^p \lambda_j f_j^i = \beta, & i = 1, 2, \dots, p \\ \sum_{j=1}^p \lambda_j = 1 \end{cases} \quad (3-20)$$

(3-20) 式是含有 $(p+1)$ 个变量 $\lambda_1, \lambda_2, \dots, \lambda_p, \beta$ 及 $(p+1)$ 个方程的线性方程组。当问题(2-1)不存在绝对最优解时, (3-20) 式有惟一的一组解, 这就是所要求的权系数(β 除外)。

5.3.6 逐步法

逐步法是英文“Step Method”的意译^[66], 这是一种迭代方法。在求解过程中的每一步, 分析者把计算结果告诉决策者, 决策

者对计算结果作出评价,若认为满意,则迭代终止;否则分析者根据决策者的意见再重复计算,如此循环地进行,直到求得满意的解为止。这种方法主要是针对如下的多目标 LP 问题设计的。

$$\begin{cases} \text{V-min} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x}))^T \\ \text{s. t.} & \mathbf{Ax} \leqslant \mathbf{b} \\ & \mathbf{x} \geqslant \mathbf{0} \end{cases} \quad (3-21)$$

其中

$$f_i(\mathbf{x}) = \sum_{j=1}^n c_{ij}x_j, \quad i = 1, 2, \dots, p$$

$$\mathbf{A} = (a_{ij})_{m \times n}, \quad \mathbf{b} = (b_1, b_2, \dots, b_m)^T$$

令 $R = \{\mathbf{x} | \mathbf{Ax} \leqslant \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}$, 逐步法的计算步骤如下。

1° 分别求解如下的 p 个单目标线性规划问题:

$$\begin{cases} \min & f_i(\mathbf{x}) = \sum_{j=1}^n c_{ij}x_j, \quad i = 1, 2, \dots, p \\ \text{s. t.} & \mathbf{x} \in R \end{cases} \quad (3-22)$$

设所得的最优解为 $\mathbf{x}^{(i)} (i = 1, 2, \dots, p)$, 相应的最优值为 $f_i^* (i = 1, 2, \dots, p)$ 。令

$$f_i^M \triangleq \max_j \{f_i(\mathbf{x}^{(j)})\}$$

2° 令

$$\alpha_i = \begin{cases} (f_i^M - f_i^*)/f_i^* \left[\sum_{j=1}^n c_{ij}^2 \right]^{1/2}, & \text{若 } f_i^* > 0 \\ (f_i^* - f_i^M)/f_i^* \left[\sum_{j=1}^n c_{ij}^2 \right]^{1/2}, & \text{若 } f_i^* < 0 \end{cases} \quad (3-23)$$

$$\lambda_i = \alpha_i / \sum_{j=1}^p \alpha_j, \quad i = 1, 2, \dots, p \quad (3-24)$$

由式(3-23)和(3-24)易见:

$$0 \leq \lambda_i \leq 1 \quad (i = 1, 2, \dots, p), \quad \sum_{i=1}^p \lambda_i = 1$$

3° 求出问题

$$\begin{cases} \min & t \\ \text{s. t.} & [f_i(\mathbf{x}) - f_i^*] \lambda_i \leq t, \quad i = 1, 2, \dots, p \\ & \mathbf{x} \in R, \quad t \geq 0 \end{cases} \quad (3-25)$$

的最优解 $\mathbf{x}^{(1)}$ 及 $f_1(\mathbf{x}^{(1)}), f_2(\mathbf{x}^{(1)}), \dots, f_p(\mathbf{x}^{(1)})$ 。

4° 将上面的计算结果 $\mathbf{x}^{(1)}, f_1(\mathbf{x}^{(1)}), \dots, f_p(\mathbf{x}^{(1)})$ 告诉决策者, 若决策者认为满意, 则取 $\mathbf{x}^{(1)}$ 为 (3-21) 的最优解, $F(\mathbf{x}^{(1)}) = (f_1(\mathbf{x}^{(1)}), f_2(\mathbf{x}^{(1)}), \dots, f_p(\mathbf{x}^{(1)}))^T$ 为最优值, 计算结束; 否则由决策者把某个目标 (例如第 j 个目标) 的值提高 Δf_j (称为宽容值), 则 (3-25) 中的约束集 R 应修改为 $R^{(1)}$, 即令 $R = R^{(1)}$, 其中

$$R^{(1)} = \{\mathbf{x} | \mathbf{x} \in R, f_j(\mathbf{x}) \leq f_j(\mathbf{x}^{(1)}) + \Delta f_j, \\ f_i(\mathbf{x}) \leq f_i(\mathbf{x}^{(1)}), \quad i = 1, 2, \dots, p, \quad i \neq j\}$$

且 $\lambda_j = 0$, 再求解问题 (3-25), 得到最优解 $\mathbf{x}^{(2)}$ 及 $f_1(\mathbf{x}^{(2)}), f_2(\mathbf{x}^{(2)}), \dots, f_p(\mathbf{x}^{(2)})$, 这样继续迭代下去, 直到求得一组决策者满意的解为止。由上述可见: 这是一种人机交互式的方法。

5.4 目标规划

目标规划是一种重要的多目标决策方法。由于它能把决策者的意愿反映到数学模型中去, 克服了线性与非线性规划的某些不足, 因而受到了人们的欢迎, 获得了越来越广泛的应用。

本节将通过几个简单的实例, 简要介绍目标规划的基本概念、建模方法和解法。

5.4.1 目标规划的基本概念

目标规划(Goal Programming)是运筹学的一个分支。1961年由美国学者 A. Charnes 和 W. W. Cooper 作为求解一个具有实际背景的不可行 LP 问题的工具而首先引入并命名的。1972 年, S. M. Lee 出版的《决策分析中的目标规划》一书是这一分支逐渐形成的一个重要标志。该书包含 15 章, 分为两个部分, 1 章 ~ 7 章为第一部分, 系统介绍了目标规划的原理、方法和计算程序, 8 章 ~ 15 章为第二部分, 主要介绍目标规划的应用, 包括在生产计划、财政决策、市场销售、广告计划、行政管理、政府决策分析、医院护理等方面的应用。1976 年, 出版了 J. P. Ignizio 著的《目标规划及其扩展》一书, 着重讨论了非线性目标规划的算法和应用。现在目标规划已经成为求解多目标数学规划问题的一种新的有力工具, 它能把决策者制定的目标融合到系统模型中去, 把人的主观意愿与客观可能统一在数学模型之中, 因而受到了大家的欢迎。

在目标规划中, 不像线性(或非线性)规划那样去直接求目标函数的最大(小)值, 而是寻求实际能够达到的值与理想目标值之间的偏差变量的最小值, 这些偏差变量表示目标的达成程度, 因此, 正偏差变量 d^+ 表示超过目标值的程度, 负偏差变量 d^- 表示未达到目标值的程度。

除了正负偏差变量 d^+, d^- 以外, 在目标规划中还要用到如下概念:

1. 绝对约束和目标约束

绝对约束是指必须严格满足的约束(条件)。例如线性规划与非线性规划中的约束都是绝对约束。绝对约束是硬约束, 对它的满

足与否,决定了解的可行性。

目标约束是决策者期望满足的一类约束,是一种软约束或称弹性约束。目标约束中决策值与目标值的差值用偏差变量来表示。这里的目标值指理想的目标值,即决策者期望达到的目标值,而决策值指实际达到的目标值。

2. 优先级别和权系数

不同目标有主次和轻重之分。一种差别是各个目标之间具有不同的优先级别,只有在优先级别高的目标已满足的前提下,才考虑优先级别低的目标,这种差别一般用优先因子来表示;另一种差别是各个目标之间重要程度的差别,是相对的,可用权系数的不同来表示。这时几个目标必须具有相同的度量单位。

3. 目标规划的目标函数

由于目标规划所追求的是尽可能接近或达到各个目标的理想目标值,也就是使各有关的偏差变量或其组合尽可能小,所以其目标函数只能是极小化。具体的表达形式有以下三种:

(1) 期望决策值不超过目标值,因此应有

$$\min d^+ \text{ 或 } \min \{g(d^+)\}$$

(2) 期望决策值不低于目标值,因此应有

$$\min d^- \text{ 或 } \min \{g(d^-)\}$$

(3) 期望决策值恰好达到目标值,因此应有

$$\min \{d^+ + d^-\} \text{ 或 } \min \{g(d^+ + d^-)\}$$

5.4.2 建模举例

【例 4.1】 某厂生产两种产品 A_1 、 A_2 ,每周需用的原材料、设备工时及获得的利润如表 5-2 所示。

表 5-2 周生产数据

产 品	原材料(kg/ 件)	设备工时(h/ 件)	利润(百元 / 件)
A_1	5	4	6
A_2	10	4	8
限额	60	40	

根据工厂实际情况,该厂决策者提出制订下周的生产计划应体现如下目标:

1. 由于原材料严重短缺,所以其使用限额不能突破;
2. 由于产品 A_2 销售疲软,所以要求 A_2 的产量不超过 A_1 的一半;
3. 考虑下周设备要进行检修,所以要求节约 4 小时设备工时;
4. 下周利润不少于 4800 元。

解 设产品 A_1 、 A_2 下周的产量分别为 x_1 、 x_2 件,则约束条件为

(1) 原材料使用限额不能超过 60 kg,所以

$$5x_1 + 10x_2 \leqslant 60 \quad (4-1a)$$

(2) A_2 的产量不超过 A_1 的一半,所以

$$x_1 - 2x_2 + d_1^- - d_1^+ = 0 \quad (4-1b)$$

其中 d_1^- 为负偏差变量, d_1^+ 为正偏差变量。它们分别表示产品 A_1 的产量不足或超过 A_2 产量的二倍的差值。

(3) 设备工时节约 4 小时,所以

$$4x_1 + 4x_2 + d_2^- - d_2^+ = 36 \quad (4-1c)$$

其中 d_2^- 、 d_2^+ 分别为设备工时不足和超过 36 小时的负偏差、正偏差变量。

(4) 利润不少于 4800 元,所以

$$6x_1 + 8x_2 + d_3^- - d_3^+ = 48 \quad (4-1d)$$

其中 d_3^-, d_3^+ 分别为利润不足和超过 4800 元的负偏差、正偏差变量。

上述的(4-1a)为绝对约束, (4-1b) ~ (4-1d)为目标约束, 此外还应满足如下的约束条件:

$$(5) x_1, x_2, d_i^-, d_i^+ \geq 0, i = 1, 2, 3.$$

下面来确定目标(达成)函数。

根据决策者的意愿, 原材料的使用限额 60 kg 不能突破, 已作为绝对约束处理。所以其目标有 3 个, 分为三个优先级别, 依次用优先因子 P_1, P_2, P_3 来表示, 它们应满足 $P_1 \geq P_2 \geq P_3$ 。所以目标函数可写成为

$$z = (P_1 d_1^-, P_2 d_2^+, P_3 d_3^-)^T$$

综合上述, 得到如下的数学模型:

$$\left\{ \begin{array}{ll} \min & z = (P_1 d_1^-, P_2 d_2^+, P_3 d_3^-)^T \\ \text{s. t.} & 5x_1 + 10x_2 \leq 60 \\ & x_1 - 2x_2 + d_1^- - d_1^+ = 0 \\ & 4x_1 + 4x_2 + d_2^- - d_2^+ = 36 \\ & 6x_1 + 8x_2 + d_3^- - d_3^+ = 48 \\ & x_1, x_2, d_i^-, d_i^+ \geq 0, i = 1, 2, 3 \end{array} \right. \quad (4-2)$$

易见, (4-2) 中的目标函数(也叫目标达成函数)是偏差变量的一个有序的向量函数, 所以这是一个多目标 LP 问题。由于问题(4-2)具有明显的层次性, 可用 5.3 节介绍的分层序列法求解。

【例 4.2】 某厂明年可用于投资的资金总额为 a 万元, 有 n 个项目 A_1, A_2, \dots, A_n 可供选择, 假设第 i 个项目 A_i 所需的资金为 a_i 万元, 消耗的能源为 b_i 吨标准煤, 耗水量为 q_i 吨, 能获得的收益为

c_i 万元,若该厂的能耗限额为 b 吨标准煤,用水限额为 q 吨,决策者追求的目标如下:

1. 总收益达到 c 万元;
2. 投资总金额不超过 a 万元;
3. 耗水量不超过 q 吨;
4. 能耗不超过 b 吨标准煤。

问应如何选择投资项目,才能尽可能依次达到上述各目标,试建立数学模型。

解 令

$$x_i = \begin{cases} 1, & \text{对项目 } A_i \text{ 投资,} \\ 0, & \text{否则,} \end{cases} \quad i = 1, 2, \dots, n$$

则上述问题的目标规划模型为

$$\left\{ \begin{array}{l} \min \quad z = (P_1 d_1^-, P_2 d_2^+, P_3 d_3^+, P_4 d_4^+)^T \\ \text{s. t.} \quad \sum_{i=1}^n c_i x_i + d_1^- - d_1^+ = c \\ \quad \quad \sum_{i=1}^n a_i x_i + d_2^- - d_2^+ = a \\ \quad \quad \sum_{i=1}^n q_i x_i + d_3^- - d_3^+ = q \\ \quad \quad \sum_{i=1}^n b_i x_i + d_4^- - d_4^+ = b \\ \quad \quad x_i = 0 \text{ 或 } 1, i = 1, 2, \dots, n \\ \quad \quad d_j^-, d_j^+ \geq 0, j = 1, 2, 3, 4 \end{array} \right. \quad (4-3)$$

其中 $d_i^-, d_i^+ (i = 1, 2, 3, 4)$ 分别为负偏差和正偏差变量,其具体意义请读者自己说明; P_1, P_2, P_3, P_4 分别为各优先级别相对应的优

先因子,它们应满足 $P_1 \gg P_2 \gg P_3 \gg P_4 > 0$ 。

5.4.3 解目标规划的图解法

图解法适用于只有两个决策变量的目标规划问题。它的优点是:(1)方法简便、直观,(2)能明显而形象地显示线性规划和线性目标规划的不同。下面通过两个例子来说明这种方法。

【例 4.3】 用图解法求解(4-1),即

$$\left\{ \begin{array}{ll} \min & z = (P_1 d_1^-, P_2 d_2^+, P_3 d_3^-)^T \\ \text{s. t.} & 5x_1 + 10x_2 \leq 60 \\ & x_1 - 2x_2 + d_1^- - d_1^+ = 0 \\ & 4x_1 + 4x_2 + d_2^- - d_2^+ = 36 \\ & 6x_1 + 8x_2 + d_3^- - d_3^+ = 48 \\ & x_1, x_2, d_i^-, d_i^+ \geq 0, i = 1, 2, 3 \end{array} \right. \quad (4-1)$$

解 (1) 画出满足(4-1)中绝对约束及决策变量非负约束(自然约束)的解空间 R_0 , 即图 5-2 中的区域: $\triangle OAB$ 。

(2) 去掉偏差变量 d_1^- , d_1^+ , 画出直线 $L_1: x_1 - 2x_2 = 0$, 即直线 OC , 其中 C 为 L_1 与 AB 的交点, 再标出第一个目标中偏差变量 d_1^- 变化时直线 L_1 的平移方面, 见图 5-2。

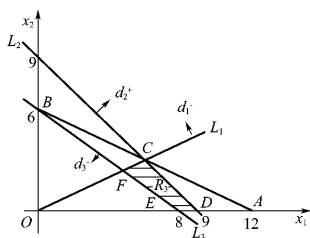


图 5-2

(3) 按优先级别的高低, 首先应考虑第一个目标, 此时要求 $\min d_1^-$, 所以满足第一个目

标要求的解空间(由 R_0 缩小)为 R_1 , 即 $\triangle OAC$ 。

(4) 去掉偏差变量 d_2^-, d_2^+ , 画出直线 $L_2: x_1 + x_2 = 9$ 及 d_2^+ 变化时直线 L_2 的平移方向。考虑第二个目标, 此时要求 $\min d_2^+$, 所以满足第二个目标要求的解空间为 R_2 , 即 $\triangle ODC$ (由 R_1 进一步缩小为 R_2)。

(5) 去掉偏差变量 d_3^-, d_3^+ , 画出直线 $L_3: 6x_1 + 8x_2 = 48$ 及 d_3^- 变化时直线 L_3 的平移方向。最后考虑第 3 个目标, 此时要求 $\min d_3^-$, 所以满足第三个目标要求的解空间为 R_3 , 即四边形 $EDCF$, 它就是所求的目标规划问题(4-1)的解集合。

容易求出 E, D, C, F 四点的坐标为 $E(8, 0), D(9, 0), C(6, 3), F(4.8, 2.4)$, 所以问题(4-1)的解可表为

$$\begin{aligned} \mathbf{x}^* &= \lambda_1(8, 0)^T + \lambda_2(9, 0)^T + \lambda_3(6, 3)^T + \lambda_4(4.8, 2.4)^T \\ &= (8\lambda_1 + 9\lambda_2 + 6\lambda_3 + 4.8\lambda_4, 3\lambda_3 + 2.4\lambda_4)^T \end{aligned}$$

其中 $\lambda_i \geq 0, i = 1, 2, 3, 4, \sum_{i=1}^4 \lambda_i = 1$ 。 ■

本题求得的解能满足(4-1)所有目标的要求, 即求得的解 \mathbf{x}^* , 使得 $\mathbf{z}^* = (0, 0, 0)^T$ 。这种情况并不常见, 多数情况下, 只能满足前几个目标的要求。

【例 4.4】 用图解法求解如下的目标规划问题

$$\left\{ \begin{array}{ll} \min & \mathbf{z} = (P_1 d_1^-, P_2 d_2^+)^T \\ \text{s. t.} & x_1 + x_2 \leq 2 \\ & x_2 + d_1^- - d_1^+ = 1 \\ & x_1 + 2x_2 + d_2^- - d_2^+ = 1 \\ & x_i \geq 0, d_i^-, d_i^+ \geq 0, i = 1, 2 \end{array} \right. \quad (4-4)$$

解 (1) 画出满足(4-4)中绝对约束及决策变量非负约束的解空间 R_0 , 即 $\triangle OAB$ (见图 5-3)

(2) 去掉偏差变量, 分别画出直线 $L_1: x_2 = 1$ 和直线 $L_2: x_1 + 2x_2 = 1$ 及偏差变量变化时直线的平移方向。

(3) 首先考虑第一个目标, 此时要求 $\min d_1^-$, 所以满足第一个目标要求的解空间为 R_1 , 即 $\triangle DCB$ 。

(4) 考虑第二个目标, 这时要求 $\min d_2^+$, 所以满足第二个目标要求的解空间为 R_2 , 即 $\triangle OEF$ (参见图 5-3)。

(5) 易见 $R_1 \cap R_2 = \emptyset$, 这时在 R_1 内无法满足 $d_2^+ = 0$, 所以, 只能在 R_1 内找一点, 使 d_2^+ 取最小值, 这一点就是点 $x^* = (0, 1)^T$, 即 D 点。这样得到目标规划问题的解 $x^* = (0, 1)^T$, 而 $z^* = (0, 1)^T$ 。

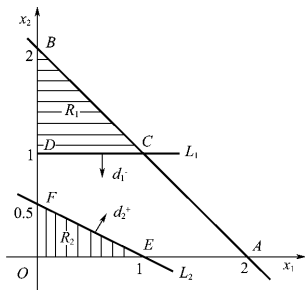


图 5-3

综合上述可见: 用图解法

求解目标规划问题, 可能会遇到如下两种情况:

1. 与例 4.3 类似, 可求得满足所有目标要求的解集 $R_t \neq \emptyset$ 。

2. 与例 4.4 类似, 所求得的解不能满足所有目标要求的解。这时, 我们要做的是寻找所谓的满意解, 使它尽可能满足优先级别较高的目标要求, 同时又使它对那些不能满足的优先级别较低目标的偏离程度尽可能小。

5.4.4 解目标规划的单纯形法

下面介绍的这一类求解线性目标规划问题的算法实质上都可视为单纯形法的某种推广, 因此, 我们称它为单纯形法。常用的有以下两种:

1. 序列线性(目标)规划法(SLP 法)

这种算法的基本思想是:按照目标规划问题的优先级别,把它分解为一系列的单目标问题(LP 问题),然后依次求解,最后单目标问题的最优解 x^* 就是原目标规划问题的最优解。这与上述的图解法处理问题的思想基本相同。下面通过一个简单的例子来说明。

【例 4.5】 用 SLP 法求解问题(4-4)

解 (1) 求解与第一优先级别对应的模型:

$$\begin{cases} \min & z_1 = d_1^- \\ \text{s. t.} & x_1 + x_2 \leq 2 \\ & x_2 + d_1^- - d_1^+ = 1 \\ & x_i \geq 0, i = 1, 2, d_1^-, d_1^+ \geq 0 \end{cases} \quad (4-5)$$

这是一个线性规划问题,可用单纯形法求得其最优解为 $x^{(1)} = (0, 1, 0, 0)^T$, 最优值 $z_1^* = d_1^- = 0$ 。

(2) 列出与第二优先级别对应的单目标 LP 模型:

$$\begin{cases} \min & z_2 = d_2^+ \\ \text{s. t.} & x_1 + x_2 \leq 1 \\ & x_2 + d_1^- - d_1^+ = 1 \\ & x_1 + 2x_2 + d_2^- - d_2^+ = 1 \\ & d_1^- = 0 \\ & x_i \geq 0, i = 1, 2, d_i^-, d_i^+ \geq 0, i = 1, 2 \end{cases} \quad (4-6)$$

其中 $d_1^- = 0$ 是根据前述求解结果增加的约束条件。

用单纯形法求解问题(4-6)得到其最优解为 $x^* = (0, 1, 0, 0, 0, 1)^T$, 最优值为 $z_2^* = d_2^+ = 1$ 。

根据上述结果,即可得到原目标规划问题(4-4)的解为 $x_1 = 0, x_2 = 1, d_1^- = d_1^+ = d_2^- = 0, d_2^+ = 1$, 最优值 $z^* = (0, 1)^T$, 这与

前面用图解求得的结果相同。

关于 SLP 法更详细的介绍见参考文献 41。

2. 多阶段单纯形法

这种算法是从线性规划的二阶段单纯形法经过修改和推广得到的一种方法,有兴趣的读者可参看参考文献 41。

习 题 五

1. 某化工厂拟生产两种产品 A 和 B , 它们都将造成环境污染, 其公害损失可折算成费用。其公害损失费用、生产设备费用和产品的最大生产能力如下表:

产 品	公 害 损 失 (万元 /t)	生产设备费 (万元 /t)	最大生产能力 (t/ 月)
A	4	2	5
B	1	5	6

已知每月市场的需求总量不少于 $7t$ 。问工厂应如何安排每月的生产计划, 在满足市场需要的前提下, 使公害损失和设备投资均达到最小; (1) 建立上述问题的数学模型; (2) 求解此问题(用线性加权和法, 取 $w_1 = 0.6$, $w_2 = 0.4$)。

2. 某厂生产 A 、 B 两种型号的摩托车, 每辆车的利润分别为 100 元和 80 元。平均生产时间分别为 $3h$ (A 种) 和 $2h$ (B 种)。该厂每周生产时间为 $120h$, 但可加班 $48h$, 在加班时间内生产每辆车的利润分别为: 90 元(A 种) 和 70 元(B 种)。市场每周需要 A 、 B 两种车各 30 辆以上, 问应如何安排每周的生产计划, 在尽量满足市场需要的前提下, 使利润最大, 而加班时间最少。试建立数学模型。

3. 证明定理 2.2。

4. 证明定理 2.3。

* 5. 证明 $R_{pa}^* \cup \{\bigcup_{j=1}^p R_j^*\} \subset R_{wp}^*$ 。

* 6. 试构造一个例子, 说明即使问题(2-1)为凸多目标规划, 但 R_{wp}^* , R_{pa}^* 也

不一定是凸集。

* 7. 证明 $E_{pa}^* \subset E_{wp}^*$ 。

* 8. 试构造一个例子,说明对于凸多目标规划(2-1),其像集合 $F(R)$ 不一定是凸集。

9. 证明定理 3.3。

10. 设多目标规划问题为

$$\begin{cases} \text{V-min } F(x) = (f_1(x), f_2(x))^T \\ \text{s.t. } x \geq 0, x \in R \end{cases}$$

其中

$$f_1(x) = (x-1)^2 + 1$$

$$f_2(x) = \begin{cases} -x + 4, & x \leq 3 \\ 1, & 3 < x \leq 4 \\ x - 3, & x > 4 \end{cases}$$

求 R_1^* , R_2^* , R_{wp}^* , R_{pa}^* 和 R^* 。

11. 求多目标规划问题

$$\begin{cases} \text{V-min } F(x) = (f_1(x), f_2(x))^T \\ \text{s.t. } \begin{aligned} x_2 + x_3 &= 3 \\ x_1 + x_2 + x_4 &= 5 \\ x_i &\geq 0, i = 1, 2, 3, 4 \end{aligned} \end{cases}$$

在分层序列意义下的最优解(设 $f_1(x)$ 比 $f_2(x)$ 重要),其中

$$f_1(x) = -2x_1 - x_2, \quad f_2(x) = -x_1$$

12. 用乘除法求解多目标规划问题:

$$\begin{cases} \max f_1(x) = 2x_1 + x_2 \\ \min f_2(x) = x_1 - x_2 \\ x_1 - x_2 \leq 4 \\ x_1 + x_2 \leq 8 \\ x_1 \geq 1, x_2 \geq 0 \end{cases}$$

13. 用线性加权和法求解问题:

$$\begin{cases} \text{V-min } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \\ \text{s. t. } 3x_1 + 8x_2 \leq 12 \\ \quad \quad x_1 + x_2 \leq 3 \\ \quad \quad 0 \leq x_1 \leq 1.5, x_2 \geq 0 \end{cases}$$

其中

$$f_1(\mathbf{x}) = -2x_1 - 8x_2, f_2(\mathbf{x}) = -6x_1 - x_2 \\ \lambda_1 = 2/3, \lambda_2 = 1/3$$

14. 求解如下的多目标规划问题:

$$(1) \begin{cases} \text{V-min } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^T \\ \text{s. t. } x_1 + x_2 \leq 2 \\ \quad \quad x_1 - x_2 \leq 1 \\ \quad \quad x_1, x_2 \geq 0 \end{cases}$$

其中

$$f_1(\mathbf{x}) = x_1 + x_2, f_2(\mathbf{x}) = x_1 - x_2, f_3(\mathbf{x}) = 3x_1 + 2x_2 \\ (2) \begin{cases} \text{V-min } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T \\ \text{s. t. } x_1 - x_2 \leq 4 \\ \quad \quad x_1 + x_2 \leq 8 \\ \quad \quad x_1, x_2 \geq 0 \end{cases}$$

其中

$$f_1(\mathbf{x}) = x_1^2 - x_2, f_2(\mathbf{x}) = 2x_2, \lambda_1 = 1/3, \lambda_2 = 2/3$$

15. 画出用分层序列法求解多目标规划问题(2-1)的框图,编写计算程序,并求解习题 14(1)(2)。

16. 某厂生产 A、B 两种布料,该厂两班生产,每周生产总时间为 80 小时,两种产品的预测销售量、生产率和盈利如下表:

产 品	预测销售量 (万米/周)	生产率 (m/h)	单位利润 (元/m)
A	7	1000	0.5
B	4.5	1000	0.3

该厂决策者提出目标如下,试建立数学模型。

(1)第一目标:充分利用现有生产能力,避免设备闲置;

(2) 第二目标: 周加班时间限制在 10 小时以内;

(3) 第三目标: 两种产品周生产量应满足预测销售量, 满足程度的权重之比等于它们单位利润之比;

(4) 第四目标: 尽量减少加班时间。

17. 在例 4.2 的投资决策问题中, 假设第 i 个项目 A_i 需用劳动力 p_i 个, 占地 g_i 亩, 在已有 4 个目标的基础上, 决策者希望再增加如下目标:

(1) 安置劳动力达到 p 个人;

(2) 占地面积不超过 g 亩。

试建立修改后的投资决策问题的目标规划模型。

18. 用图解法求解

$$\left\{ \begin{array}{ll} \min & z = (P_1 d_1^-, P_2 d_2^+, P_3 d_3^-, P_4 d_4^-)^T \\ \text{s. t.} & x_1 + 2x_2 + d_1^- - d_1^+ = 6 \\ & x_1 + 2x_2 + d_2^- - d_2^+ = 9 \\ & x_1 - 2x_2 + d_3^- - d_3^+ = 4 \\ & x_2 + d_4^- - d_4^+ = 2 \\ & x_1, x_2 \geq 0, d_i^-, d_i^+ \geq 0, i = 1, 2, 3, 4 \end{array} \right.$$

19. 用 SLP 法求解第 18 题。

20. 用 SLP 法求解 5.4 节的问题(4-2)。

* 第 6 章 整数规划

整数规划是一类要求决策变量取整数值的数学规划,若线性规划中的变量要求取整数值时,则称其为整数线性规划,若非线性规划中的变量要求取整数值时,则称其为整数非线性规划,要求变量仅取 0 或 1 值的数学规划称为 0—1 规划。只要求部分变量取整数值的数学规划称为混合型整数规划,要求全部变量取整数值的数学规划称为纯整数规划。

实践中有许多量具有不可分割的性质,如人数、机器数、方案数、项目数等;而开与关、取与舍、有与无等逻辑现象都需要用 0—1 变量来描述。因此,整数规划在许多领域中有重要的应用,如分配问题、工厂选址、线路设计、时间表问题、背包问题、旅行推销员问题等。1963 年,R. E. Gomory 提出了解整数线性规划的割平面算法,使整数规划逐渐成为一个独立的分支。

本章首先介绍整数规划的一些简单实例,然后着重介绍求解整数线性规划的几种常用的算法:分枝定界算法、割平面法、隐枚举法和匈牙利法。

6.1 整数规划问题举例

【例 1.1】 投资决策问题

设在今后五年内可用于投资的资金总额为 B 万元,有 n 个可供选择的投资项目,假定每个项目至多只能投资一次,第 i 个项目

所需的投资资金为 b_i 万元,将会获得的利润为 c_i 万元,问应如何选择投资项目,才能使获得的总利润最大。

解 令

$$x_i = \begin{cases} 1, & \text{若对第 } i \text{ 个项目投资} \\ 0, & \text{若不对第 } i \text{ 个项目投资} \end{cases} \quad i = 1, 2, \dots, n$$

设获得的总利润为 y , 则上述问题的数学模型为:

$$\begin{cases} \max & y = \sum_{i=1}^n c_i x_i \\ \text{s. t.} & \sum_{i=1}^n b_i x_i \leq B \\ & x_j = 0 \text{ 或 } 1, j = 1, 2, \dots, n \end{cases} \quad (1-1)$$

在问题(1-1)中,决策变量 x_1, x_2, \dots, x_n 只能取 0, 1 值,故称它为 0—1 规划。

【例 1.2】 背包问题

一个背包的容积为 v , 现有 n 件物品可供选择装入背包, 物品 j 的重量为 w_j , 体积为 $v_j (j = 1, 2, \dots, n)$, 问应装哪几件物品, 才能既不超过背包的容积, 又使所装的物品的总重量最大。

解 令

$$x_j = \begin{cases} 1, & \text{物品 } j \text{ 被装入背包,} \\ 0, & \text{物品 } j \text{ 不装入背包,} \end{cases} \quad j = 1, 2, \dots, n$$

设所装的物品的总重量为 w , 则上述问题的数学模型为:

$$\begin{cases} \max & w = \sum_{j=1}^n w_j x_j \\ \text{s. t.} & \sum_{j=1}^n v_j x_j \leq v \\ & x_j = 0 \text{ 或 } 1, j = 1, 2, \dots, n \end{cases} \quad (1-2)$$

容易看出: 背包问题的数学模型与例 1.1 的投资决策问题的数学模型是一样的, 它们都是 0—1 规划问题。

【例 1.3】 生产组织与计划问题

某工厂用 m 台机床 A_1, A_2, \dots, A_m 加工 n 种零件 B_1, B_2, \dots, B_n 。在一个生产周期内, 已知第 i 台机床 A_i 最多只能工作 a_i 个机时 ($i = 1, 2, \dots, m$), 工厂需加工零件 B_j 的数量为 b_j 个 ($j = 1, 2, \dots, n$)。用机床 A_i 加工一个零件 B_j 所需机时为 t_{ij} , 成本为 c_{ij} (元 / 个), 问在这个生产周期内, 应如何安排各机床的生产任务, 才能既完成加工任务, 又使总的加工成本最少。

解 设机床 A_i 在一个生产周期内加工零件 B_j 的个数为 x_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 总的加工成本为 y , 则上述问题的数学模型为:

$$\left\{ \begin{array}{ll} \min & y = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} & \sum_{j=1}^n t_{ij} x_{ij} \leq a_i, \quad i = 1, 2, \dots, m \\ & \sum_{i=1}^m x_{ij} \geq b_j, \quad j = 1, 2, \dots, n \\ & x_{ij} \geq 0, \text{ 且 } x_{ij} \in I, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \end{array} \right. \quad (1-3)$$

(机床 A_i 加工各个零件的机时不得超过 A_i 所能工作的机时)

(各机床加工零件 B_j 的数量不能少于 B_j 的需要量)

其中 $I = \{0, 1, 2, \dots, n, \dots\}$ 。

由于在问题(1-3)中, 决策变量 x_{ij} 只能取非负的整数值, 所以它是一个纯整数规划。 ■

【例 1.4】 设备的购置和安装问题

某工厂有 m 种设备 A_1, A_2, \dots, A_m , 已知这些设备的数量分别为 a_1, a_2, \dots, a_m 台。为扩大再生产, 该厂决定再购买部分设备, 已知 A_i 的单价为 p_i 元, 今该厂有资金 M 元, 可用于购买这些种类的设备。设该厂有 n 处可安装这些设备, B_j 处最多能安装 b_j 台, 将一台

设备 A_i 安装在 B_j 处,其经济效益为 c_{ij} (元),问应如何购置和安装这些设备,才能使总的经济效益最大。

解 用 x_{ij} 表示设备 A_i 安装在 B_j 处的台数, y_i 表示购置设备 A_i 的台数, z 表示总的经济效益,则上述问题的数学模型为:

$$\left\{ \begin{array}{l} \max \quad z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} \quad \sum_{j=1}^n x_{ij} \leq y_i + a_i, \quad i = 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} \leq b_j, \quad j = 1, 2, \dots, n \\ \sum_{i=1}^m p_i y_i \leq M \\ \sum_{i=1}^m (y_i + a_i) \leq \sum_{j=1}^n b_j \\ x_{ij}, y_i \geq 0, \text{ 且 } x_{ij}, y_i \in I, i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{array} \right. \quad (1-4)$$

其中 $I = \{0, 1, 2, \dots\}$ 。

易见问题(1-4)也是一个纯整数规划。 ■

【例 1.5】 工厂选址问题

设某商品有 n 个需求点,有 m 个地点可供选择建厂生产这种商品,每个地点最多只能建一个工厂,在 i 处建厂,生产能力为 $D_i(t)$,单位时间的固定成本为 a_i 元。需求点 j 的需求量为 $b_j(t)$,从地点 i 到需求点 j 的单位运费为 c_{ij} (元 /t),问应如何选择厂址和安排运输计划,才能获得总花费最少的方案。

解 设在单位时间内,从厂址 i 运往需求点 j 的产品数量为 $x_{ij}(t)$,引入布尔变量

$$y_i = \begin{cases} 1, & \text{若在 } i \text{ 处建厂} \\ 0, & \text{若不在 } i \text{ 处建厂} \end{cases}$$

设在单位时间内总的花费为 s 元, 则上述问题的数学模型为:

$$\left\{ \begin{array}{l} \min \quad s = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m a_i y_i \\ \text{s. t.} \quad \sum_{j=1}^n x_{ij} \leq D_i y_i, \quad i = 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} \geq b_j, \quad j = 1, 2, \dots, n \\ x_{ij} \geq 0, \quad y_i = 0 \text{ 或 } 1, \quad i = 1, 2, \dots, m \\ \phantom{x_{ij} \geq 0,} \phantom{y_i = 0 \text{ 或 } 1,} j = 1, 2, \dots, n \end{array} \right. \quad (1-5)$$

在问题(1-5)中, x_{ij} 为非负连续变量, y_i 为 0-1 型变量, 所以问题(1-5) 是一个混合型整数规划。 ■

【例 1.6】 人员安排问题

某宾馆一天中各时段需要的服务人数如下表所示。按规定, 服务员连续工作 8 小时为一班。现要求安排服务员的工作时间, 使所需服务员总数最少。

时段	始末时间	所需服务员人数
1	6:00 ~ 8:00	6
2	8:00 ~ 10:00	12
3	10:00 ~ 12:00	10
4	12:00 ~ 14:00	8
5	14:00 ~ 16:00	9
6	16:00 ~ 18:00	14
7	18:00 ~ 20:00	8
8	20:00 ~ 22:00	6
9	22:00 ~ 24:00	4

解 设 x_j 表示第 j 时段开始上班的服务员人数。由于每 2 小时为一时段, 所以第 j 时段开始时上班的服务员将在第 $j+3$ 时段结束时下班。因此只需考虑 x_1, x_2, \dots, x_6 。

相应的模型为:

$$\left\{ \begin{array}{l} \min \quad z = \sum_{i=1}^6 x_i \\ \text{s. t.} \quad x_1 \geq 6 \\ \quad \quad x_1 + x_2 \geq 12 \\ \quad \quad x_1 + x_2 + x_3 \geq 10 \\ \quad \quad x_1 + x_2 + x_3 + x_4 \geq 8 \\ \quad \quad x_2 + x_3 + x_4 + x_5 \geq 9 \\ \quad \quad x_3 + x_4 + x_5 + x_6 \geq 14 \\ \quad \quad x_4 + x_5 + x_6 \geq 8 \\ \quad \quad x_5 + x_6 \geq 6 \\ \quad \quad x_6 \geq 4 \\ \quad \quad x_i \geq 0, x_i \in I = \{0, 1, 2, \dots\}, i = 1, \dots, 6 \end{array} \right. \quad (1-6)$$

这是一个纯整数规划问题。 ■

【例 1.7】指派问题

欲指派 n 个人去做 n 件事。已知第 i 人做第 j 件事的费用为 $c_{ij} (i, j = 1, \dots, n)$, 要求拟定一个指派方案, 使每个人做一件事, 且使总费用最小。

解 设

$$x_{ij} = \begin{cases} 1, & \text{第 } i \text{ 人做第 } j \text{ 件事} \\ 0, & \text{否则} \end{cases} \quad i, j = 1, \dots, n$$

则模型为:

$$\left\{ \begin{array}{l} \min \quad z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} \quad \sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n \\ \quad \quad \quad (\text{第 } i \text{ 人做一件事}) \\ \quad \quad \quad \sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n \\ \quad \quad \quad (\text{第 } j \text{ 件事由一人去做}) \\ \quad \quad \quad x_{ij} = 0 \text{ 或 } 1, i, j = 1, \dots, n \end{array} \right. \quad (1-7)$$

这是一个 0—1 规划问题。 ■

6.2 整数线性规划的解法概述

上一节讲的几个例子都是整数线性规划问题,其一般形式可写为

$$\left\{ \begin{array}{l} \max \quad z = \mathbf{c}^T \mathbf{x} \\ \quad (\text{或 } \min) \\ \text{s. t.} \quad \mathbf{Ax} \leq \mathbf{b} \\ \quad \quad \mathbf{x} \geq \mathbf{0}, x_i \in I, i \in J \subset \{1, 2, \dots, n\} \end{array} \right. \quad (2-1)$$

其中

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, \mathbf{c} = (c_1, c_2, \dots, c_n)^T$$

$$\mathbf{b} = (b_1, b_2, \dots, b_m)^T, \mathbf{A} = (a_{ij})_{m \times n}, I = \{0, 1, 2, \dots\}$$

若 $J = \{1, 2, \dots, n\}$, 则(2-1)为纯整数规划问题;若 $J \neq \{1, 2, \dots, n\}$, 则(2-1)为混合型整数规划问题;若 $I = \{0, 1\}$, 则(2-1)为 0-1 规划问题。

整数规划(以后常简称为 IP)有着广泛的应用,从上节的几个例子我们也可以看出这点,但 IP 的求解比 LP 要困难得多。大家知

道,LP 已有有效的解法。因此,为了求解问题(2-1),一个容易想到的方法是,先去掉决策变量取整数值这一约束: $x_i \in I$ ($i \in J$),然后求出其相应的LP问题(称为(2-1)的松弛问题)的最优解 \tilde{x}^* 。若对任意 $i \in J$,都有 $\tilde{x}_i^* \in I$,则 \tilde{x}^* 就是原问题(2-1)的最优解,否则一个简单的试探性方法是,将最优解 \tilde{x}^* 取“整”,即令 $\bar{x}_i^* = [\tilde{x}_i^* + 0.5]$ ($i \in J$),以 \bar{x}^* 作为原IP问题(2-1)的最优解。但计算的实践说明:这样取“整”得到的 \bar{x}^* ,有时不是(2-1)的可行解;有时它虽然是可行解,但并不是(2-1)的最优解。若用 z^* 表示问题(2-1)的最优值,用 \bar{z} 表示(2-1)与 \bar{x}^* 对应的目标函数值,则差值 $|z^* - \bar{z}|$ 可能很大。只有当 \tilde{x}_i^* 很大时,比如 $\tilde{x}_i^* \geq 100$ ($i \in J$),才可能用 \bar{x}^* 作为问题(2-1)的近似最优解。

目前常用的求解IP问题的方法有:① 分枝定界法和隐数法;② 割平面法;③ 分解方法;④ 群论方法;⑤ 动态规划方法;⑥ 隐枚举法;⑦ 匈牙利法;等。分枝定界法是实际应用较多的一种方法,国外不少用于求解IP问题的商业计算机程序是用此法编制的,其他的一些算法比它要逊色一些,但又各具特点,适用于求解不同类型的整数规划问题。

求解整数线性规划问题的许多方法是基于如下的“松弛”与“分解”的思想。

6.2.1 松 弛

考察问题

$$\begin{cases} \max & z = c^T x \\ \text{s. t.} & Ax \leq b \\ & x \geq 0, x_i \in I, i \in J \subset \{1, 2, \dots, n\} \end{cases} \quad (P)$$

其中 $I = \{0, 1, 2, \dots\}$ 。

定义 2.1 在问题(P)中放弃某些约束条件所得到的问题(\tilde{P}),称为(P)的松弛问题。

下面我们用 R 表示 (P) 的可行集, 用 x^* 和 z^* 分别表示 (P) 的一个最优解和最优值; 用 \tilde{R} 表示 (\tilde{P}) 可行集, 用 \tilde{x}^* 和 \tilde{z}^* 表示 (\tilde{P}) 的一个最优解与最优值。对于任何松弛问题 (\tilde{P}) , 有如下重要性质:

1. $R \subset \tilde{R}$ 。
2. 若 (\tilde{P}) 没有可行解, 则 (P) 也没有可行解。
3. $z^* \leq \tilde{z}^*$ 。
4. 若 $\tilde{x}^* \in R$, 则 \tilde{x}^* 也是 (P) 的最优解。

对于求最小值的问题 (P) , 则有 $z^* \geq \tilde{z}^*$ 。

上述性质是明显的, 请读者自己证明。

通常的松弛方式是去掉决策变量取整数值这一约束: $x_i \in I$ ($i \in J$), 有时也采用去掉 $x \geq 0$ 或去掉 $Ax \leq b$ 的松弛方式。

6.2.2 分解

定义 2.2 用 $R(P)$ 表示问题 (P) 的可行集, 若条件

1. $\bigcup_{i=1}^m R(P_i) = R(P)$,
2. $R(P_i) \cap R(P_j) = \emptyset$ ($1 \leq i \neq j \leq m$)

成立, 则称问题 (P) 被分解为子问题 $(P_1), (P_2), \dots, (P_m)$ 之和。

一般是一分为二, 即 $m = 2$ 。

【例 2.1】 考察问题

$$\begin{cases} \max & z = 5x_1 + 8x_2 \\ \text{s. t.} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_1, x_2 \geq 0, x_i \in I, i = 1, 2 \end{cases} \quad (P_0)$$

去掉约束 $x_i \in I$ ($i = 1, 2$), 得到松弛问题 (\tilde{P}_0) , 其最优解为 $\tilde{x}^* = (2.25, 3.75)^T$, 最优值 $\tilde{z}^* = 41.25$, 在 (P_0) 的原有约束条件之

外,分别增加约束条件:(1) $x_2 \geq 4$, (2) $x_2 \leq 3$, 形成两个子问题 (P_1) 和 (P_2) , 则问题 (P_0) 被分解为子问题 (P_1) 和 (P_2) , 像这种把可行集 $R(P_0)$ 分割为较小的子集 $R(P_1)$ 和 $R(P_2)$ 的作法, 称为分枝。

概括地说, 求解一个整数 LP 问题 (P) , 其基本步骤是: 首先, 选定一种松弛方式, 将问题 (P) 松弛成为 (\tilde{P}) , 使其较易求解。若 (\tilde{P}) 没有可行解, 则 (P) 也没有可行解。若 (\tilde{P}) 的最优解 \tilde{x}^* 也是 (P) 的可行解, 即 $\tilde{x}^* \in R$, 则它就是 (P) 的最优解, 计算结束。若 $\tilde{x}^* \notin R$, 则下一步至少有两不同的途径: 一是设法改进松弛问题 (\tilde{P}) , 以期求得 (P) 的最优解, 割平面法就属于这一类, 它用 LP 问题作为松弛问题, 通过逐次生成割平面条件, 来不断地改进松弛问题, 使最后求得的松弛问题的最优解也是整数解, 从而也就是问题 (P) 的最优解。另一条途径是利用分解技术, 将 (P) 分解为两个或几个子问题之和, 这类算法又分为两类: 隐数法和分枝定界法, 它们都是用 LP 问题作为松弛问题, 不同之处仅在于探测(求解)子问题的先后次序不一样, 隐数法是按照后出现的子问题先探测, 先出现的子问题后探测, 即按“先入后出”的原则来确定求解子问题的先后顺序, 这种方法的计算程序比较简单, 计算过程中需要保存的中间信息较少, 但计算时间一般较长。分枝定界法是按照上界的大小来确定探测子问题的先后次序的, 上界大的子问题优先探测。这种方法, 选取子问题时具有灵活性, 计算程序要复杂一些, 需要保存的中间信息也要多一些, 但对于求极大值问题而言, 由于上界大的子问题中, 存在整数最优解的可能性较大, 因此计算时间往往要短些。由于这几种方法各有优缺点和不同的适用范围, 我们将在下面的几节中, 首先介绍分枝定界法, 然后介绍割平面法、隐枚举法和匈牙利法。

6.3 分枝定界法

6.3.1 分枝定界法的基本思想

分枝定界法可用于解纯整数规划和混合型整数规划,它的基本思想是:将要求解的 IP 问题(P)不断地分解为几个子问题的和,如果对每个子问题的可行域(简称为子域)能找到域内的最优解,或者明确原问题(P)的最优解肯定不在这个子域内,这样原问题在这个子域上就容易解决了。分成子问题是逐步进行的,这个过程称为分枝。对于每个子问题,仍然是求解它相应的松弛 LP 问题,若得到最优整数解或可以肯定原问题(P)的最优解不在这个子域内,则这个子域就查清了,不需再分枝;若该子问题最优解不是整数解,又不能确定原问题(P)的最优解是否在这个子域内,就把这个子域分为两部分,而把子问题的非整数最优解排除在外。如何来确定在所考察的子域内有没有所找的最优解呢?这是利用“界”来确定的。因为相应的松弛 LP 问题的可行域 \tilde{R} 包含整数 LP 问题的可行域 R ,即有 $R \subset \tilde{R}$,所以,对求最小值的问题有 $z^* \geq \tilde{z}^*$,对求最大值的问题有 $z^* \leq \tilde{z}^*$ 。而与任一个整数可行解对应的目标函数值是最优值的一个上(下)界。下面举例来进行具体说明。

【例 3.1】 求解整数规划问题

$$\begin{cases} \max & z = 20x_1 + 10x_2 \\ \text{s. t.} & 5x_1 + 4x_2 \leq 24 \\ & 2x_1 + 5x_2 \leq 13 \\ & x_i \geq 0, x_i \in I, i = 1, 2 \end{cases} \quad (3-1)$$

其中 $I = \{0, 1, 2, \dots\}$ 。

解 (1) 去掉变量取整数值约束 $x_i \in I$ ($i = 1, 2$), 得到松弛

问题(P_0):

$$\begin{cases} \max & z = 20x_1 + 10x_2 \\ \text{s. t.} & 5x_1 + 4x_2 \leq 24 \\ & 2x_1 + 5x_2 \leq 13 \\ & x_1, x_2 \geq 0 \end{cases} \quad (P_0)$$

(2) 求解松弛 LP 问题(P_0), 得最优解为 $x_1 = 4.8, x_2 = 0$, 最优值为 $z = 96$ 。由于 $x = 0$ 显然是(3-1)的一个可行解, 相应的目标函数值为 $z = 0$, 所以下界 $\underline{z} = 0$, 上界 $\bar{z} = 96, 0 \leq z^* \leq 96$ 。

若采用取“整”方法, 则得 $\bar{x}_1 = 5, \bar{x}_2 = 0$, 它不满足(3-1)的约束条件 $5x_1 + 4x_2 \leq 24$, 因此 $\bar{x} = (5, 0)^T \notin R$ (R 为(3-1)的可行域)。

(3) 在问题(P_0)的原有约束条件之外, 分别增加两个互相排斥的约束:

$$x_1 \leq 4; \quad x_1 \geq 5$$

形成两个子问题: P_1 和 P_2 。由于增加了两个互相排斥的约束: $x_1 \leq 4$ 和 $x_1 \geq 5$, 原来的非整数最优解 $x = (4.8, 0)^T$ 被排除了。问题 P_1 和 P_2 分别为

$$\begin{cases} \max & z = 20x_1 + 10x_2 \\ \text{s. t.} & 5x_1 + 4x_2 \leq 24 \\ & 2x_1 + 5x_2 \leq 13 \\ & x_1 \leq 4, x_1, x_2 \geq 0 \end{cases} \quad (P_1)$$

和

$$\begin{cases} \max & z = 20x_1 + 10x_2 \\ \text{s. t.} & 5x_1 + 4x_2 \leq 24 \\ & 2x_1 + 5x_2 \leq 13 \\ & x_1 \geq 5, x_2 \geq 0 \end{cases} \quad (P_2)$$

易见, (P_1) 和 (P_2) 是分别在(P_0)中增加约束条件 $x_1 \leq 4$ 或 $x_1 \geq 5$

5 而形成的。因此,也可以说,为了保证 $x_1 \in I$,把问题 (P_0) 分成了两枝,每一枝增加了一个约束条件,这样就保证了原来的非整数最优解不会重复出现。

但在问题 (P_2) 中,可行集 $R_2 = \emptyset$ (空集),因为满足 $x_1 \geq 5$, $x_2 \geq 0$ 的任何点 x 不可能满足约束 $5x_1 + 4x_2 \leq 24$, 所以问题 (P_2) 没有可行解,即 $R_2 = \emptyset$, 因此只需求解问题 (P_1) 。

(4) 用单纯形法易得问题 (P_1) 的最优解为 $x_1 = 4$, $x_2 = 1$, 最优值 $z = 90$ 。由于 $x_i \in I$ ($i = 1, 2$), 且 (P_2) 没有可行解, 所以 $x = (4, 1)^T$ 为问题 (3.1) 的最优解, $z = 90$ 为问题 (3.1) 的最优值。

6.3.2 分枝定界法的计算步骤和程序框图

设要求解的整数规划问题为

$$\begin{cases} \max & z = c^T x \\ \text{s. t.} & Ax \leq b \\ & x \geq 0, x_i \in I, i \in J \subset \{1, 2, \dots, n\} \end{cases} \quad (P)$$

用 R 表示 (P) 的可行集, x^* 和 z^* 分别表示问题 (P) 的最优解和最优值。用 (P_0) 表示在 (P) 中去掉变量取整数值的约束后而得到的松弛问题。下面将计算步骤和程序框图结合在一起, 介绍如下 (见图 6-1)。

几点说明:

(1) 在框图 6-1 中, 挑选未分析完的子问题, 一般采用上界大的子问题优先探测 (求解) 的原则。

(2) 如何由松弛 LP 问题的解中的非整数变量来形成新子问题, 通常可采用不同的规则来确定, 这里就不讨论了。

(3) 在分枝定界法中, 要多次求解 LP 问题, 一般多采用对偶单纯形法。

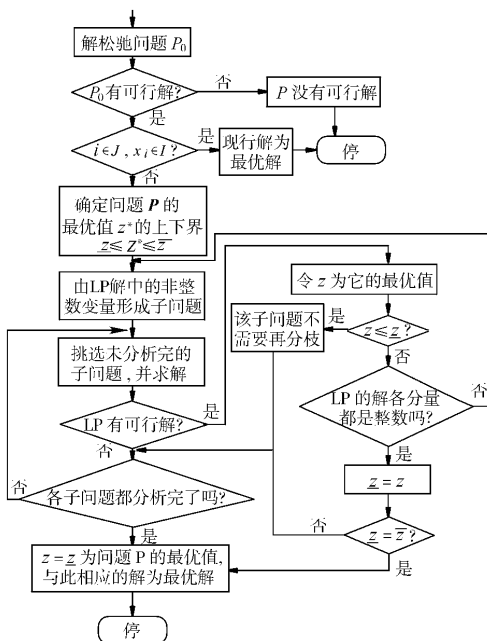


图 6-1 极大化 IP 问题的分枝定界法程序框图

6.3.3 分枝定界法的计算例子

【例 3.2】 用分枝定界法求解如下的整数线性规划问题 (P) ：

$$\begin{cases} \max & z = 5x_1 + 8x_2 \\ \text{s. t.} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_i \geq 0, x_i \in I, i = 1, 2 \end{cases} \quad (P)$$

解 (1) 解如下的松弛问题(P_0):

$$\begin{cases} \max & z = 5x_1 + 8x_2 \\ \text{s. t.} & x_1 + x_2 \leq 6 \\ & 5x_1 + 9x_2 \leq 45 \\ & x_1, x_2 \geq 0 \end{cases} \quad (P_0)$$

得最优解 $x_1 = 2.25, x_2 = 3.75$, 最优值 $z = 41.25$ 。由于原问题(P)中目标函数的系数为整数, $x_i \in I (i = 1, 2)$, 所以 $z^* \leq 41$, 即 $\bar{z} = 41$ 。令 $\underline{z} = 0$, 则有 $\underline{z} = 0 \leq z^* \leq 41 = \bar{z}$ 。我们将这些结果记录在图 6-2 中。

(2) 因为 x_2 的值不是整数, 所以在(P_0)的约束之外, 增加互相排斥的约束条件: $x_2 \geq 4$ 和 $x_2 \leq 3$ 形成两个子问题。今后为书写简便, 原有的约束将不写出, 只列出新增加的约束条件。

$$P_1: x_2 \geq 4; \quad P_2: x_2 \leq 3$$

(3) 挑选子问题 P_1 进行求解, 得 P_1 的最优解为 $x_1 = 1.8, x_2 = 4$, 最优值 $z = 41$ 。

为了便于记忆, 我们将上述计算结果画成一棵枚举树(参见图 6-2), 以后的计算结果将陆续画出。

(4) 因为 $z = 41 > \underline{z}$, 且变量 x_1 取非整数值, 所以在 P_1 的原有约束之外, 增加两个互相排斥的约束条件: $x_1 \geq 2, x_1 \leq 1$ 形成两个新的子问题:

$$P_3: x_1 \geq 2; \quad P_4: x_1 \leq 1$$

(5) 挑选子问题 P_3 进行求解, 易见 P_3 没有可行解, 即 $R(P_3) = \emptyset$ 。这样的子问题不必再进行分枝了, 我们称其为已查清的问题, 在枚举树的方框下面用“*”号标记, 表示此问题已查清。

(6) 挑选子问题 P_4 进行求解, 得最优解为 $x_1 = 1, x_2 = 4\frac{4}{9}$, 最优值 $z = 40\frac{5}{9}$ 。

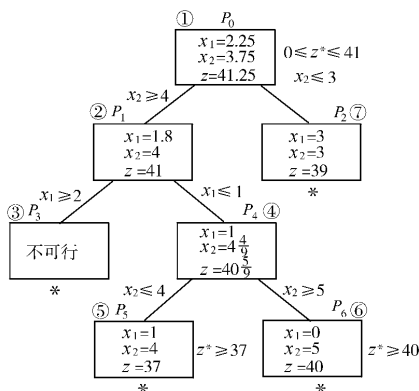


图 6-2 枚举树

(7) 因为 x_2 为非整数值, 再形成新的子问题:

$$P_5: x_2 \leq 4; \quad P_6: x_2 \geq 5$$

(8) 挑选子问题 P_5 进行求解, 得最优解 $x_1 = 1$ 和 $x_2 = 4$, 最优值 $z = 37$ 。由于 P_5 的解的分量均为整数, 所以 $\underline{z} = 37$, 于是 $z^* \geq 37$, 这样的子问题也不需再分枝了, 故在 P_5 的方框下面记上“*”号, 表示此问题已查清。

(9) 因为 $\underline{z} \neq \bar{z}$, 这时还剩下两个子问题 P_2 和 P_6 未查清, 挑选子问题 P_6 进行求解, 得最优解为 $x_1 = 0, x_2 = 5$, 最优值 $z = 40$ 。

因为 P_6 的解的各分量均为整数, 所以令 $\underline{z} = 40$, 并且在 P_6 的方框下面记一“*”号, 表示此问题已查清。

(10) 由于 $\underline{z} = 40 \neq \bar{z} = 41$, 这时只剩下一个子问题 P_2 未查清, 解之得最优解 $x_1 = 3, x_2 = 3$, 最优值 $z = 39$ 。

(11) 因为 $z = 39 < \underline{z} = 40$, 且各子问题均已分析完了。所以计算结束, $\underline{z} = 40$ 为问题 P 的最优值, 即 $z^* = 40$, 最优解为 $x^* = (0, 5)^T$ 。 ■

6.4 割平面法

解整数 LP 问题的割平面方法是 1963 年由柯莫瑞 (R. E. Gomory) 首先提出来的。从此以后, IP 逐渐成为一个独立的运筹学分支, 由于用割平面法求解 IP 问题常常会遇到收敛很慢的情况, 所以完全用它来求解 IP 问题的仍然不多, 但它在理论上是重要的, 被认为是 IP 的核心部分。本节简要介绍割平面法的基本思想和用它求解纯 IP 问题的具体步骤。希望进一步了解这方面内容的读者可参看文献 5 ~ 7。

考虑纯 IP 问题:

$$\begin{cases} \min & z = c^T x \\ \text{s. t.} & Ax \leq b \\ & x \geq 0, x_i \in I, i = 1, 2, \dots, n \end{cases} \quad (P)$$

去掉约束 $x_i \in I (i = 1, 2, \dots, n)$, 得到松弛问题 (\tilde{P}) , 若问题 (\tilde{P}) 的最优解 $\tilde{x} \in R$ (R 为问题 (P) 的可行域), 则 \tilde{x} 即为问题 (P) 的最优解, 计算结束, 若 $\tilde{x} \notin R$, 即 (\tilde{P}) 的最优解 \tilde{x} 的分量不全是整数, 则设法对 (\tilde{P}) 增加一个约束条件 (称它为割平面条件), 使这个非整数解 \tilde{x} 恰好被割去, 而原问题 (P) 的任何一个可行解都不被割去。我们把增添了割平面条件的问题, 记为 (P_1) , 再用对偶单纯形法求解问题 (P_1) , 若所得的最优解 $x^{(1)} \in R$, 则 $x^{(1)}$ 为 (P) 的最优解, 计算结束; 否则再增加一个割平面条件, 形成问题 (P_2) , …… , 如此继续进行下去, 直到求得问题 (P) 的最优解为止。下面我们通过一个例子来具体说明。

【例 4.1】 求解 IP 问题

$$\left\{ \begin{array}{ll} \min & z = -x_1 - 27x_2 \\ \text{s. t.} & -x_1 + x_2 \leq 1 \\ & 24x_1 + 4x_2 \leq 25 \\ & x_1, x_2 \geq 0, x_1, x_2 \in I \end{array} \right. \quad (4-1)$$

解 去掉约束 $x_1, x_2 \in I$ 后的问题, 记为 (\tilde{P}) 。引入松弛变量 $x_3 \geq 0, x_4 \geq 0$, 将问题 (\tilde{P}) 化成下列 LP 问题, 记为 (P_0) , 即

$$\left\{ \begin{array}{ll} \min & z = -x_1 - 27x_2 \\ \text{s. t.} & -x_1 + x_2 + x_3 = 1 \\ & 24x_1 + 4x_2 + x_4 = 25 \\ & x_i \geq 0, i = 1, 2, 3, 4 \end{array} \right. \quad (P_0)$$

用单纯形法求解问题 (P_0) , 计算过程如下:

基矢量	a_1	$a_2 \downarrow$	$a_3 \uparrow$	a_4	b
a_3	-1	1	1	0	1
a_4	24	4	0	1	25
r_j	-1	-27	0	0	0
基矢量	$a_1 \downarrow$	a_2	a_3	$a_4 \uparrow$	b
a_2	-1	1	1	0	1
a_4	28	0	-4	1	21
r_j	-28	0	27	0	27
基矢量	a_1	a_2	a_3	a_4	b
a_2	0	1	$\frac{6}{7}$	$\frac{1}{28}$	$\frac{7}{4}$
a_1	1	0	$-\frac{1}{7}$	$\frac{1}{28}$	$\frac{3}{4}$
r_j	0	0	23	1	48

从而,得到问题 (P_0) 的最优解: $x^{(0)} = \left(\frac{3}{4}, \frac{7}{4}\right)^T$ 。由于 $x^{(0)}$ 不是整数解,我们来确定割平面条件,从最终表中任选一个与非整数值变量对应的约束方程,比如

$$x_1 + \left(-\frac{1}{7}\right)x_3 + \frac{1}{28}x_4 = \frac{3}{4} \quad (4-2)$$

我们的目的是要求得最优的整数解,因此,将方程(4-2)中所有系数,包括常数项按

$$a = [a] + r$$

的形式,分解为整数与非负小数(分数)两部分之和,即将(4-2)改写为

$$(1+0)x_1 + \left(-1 + \frac{6}{7}\right)x_3 + \left(0 + \frac{1}{28}\right)x_4 = 0 + \frac{3}{4}$$

把整系数项和非整系数项左右分开得

$$x_1 - x_3 + 0 \cdot x_4 - 0 = \frac{3}{4} - 0 \cdot x_1 - \left(\frac{6}{7}\right)x_3 - \left(\frac{1}{28}\right)x_4 \quad (4-3)$$

由于 x_1, x_2 取整数值,由问题 (P_0) 中的约束可知: x_3, x_4 也应取整数值。于是:① 因为(4-3)式左边为整数,所以其右边也应为整数;② 因为 $x_3 \geq 0, x_4 \geq 0$,所以(4-3)式的右边之值不会超过 $3/4$ 。

综合上述两点可知:(4-3)式右边之整数值最大为零,即有

$$\frac{3}{4} - \frac{6}{7}x_3 - \frac{1}{28}x_4 \leq 0 \quad (4-4)$$

也可写成

$$-24x_3 - x_4 \leq -21 \quad (4-5)$$

我们称(4-2)为诱导方程,(4-4)式(或(4-5)式)为割平面条件。引入松弛变量 $x_5 \geq 0$,则(4-5)式变为

$$-24x_3 - x_4 + x_5 = -21 \quad (4-6)$$

称(4-6)式为割平面方程。把它加到前面的最终表中,用对偶单纯形法继续求解,过程如下:

基矢量	a_1	a_2	$a_3 \downarrow$	a_4	$a_5 \uparrow$	b
a_2	0	1	$\frac{6}{7}$	$\frac{1}{28}$	0	$\frac{7}{4}$
a_1	1	0	$-\frac{1}{7}$	$\frac{1}{28}$	0	$\frac{3}{4}$
a_5	0	0	-24	-1	1	-21
r_j	0	0	23	1	0	48
a_2	0	1	0	0	$\frac{1}{28}$	1
a_1	1	0	0	$\frac{1}{24}$	$-\frac{1}{168}$	$\frac{7}{8}$
a_3	0	0	1	$\frac{1}{24}$	$-\frac{1}{24}$	$\frac{7}{8}$
r_j	0	0	0	$\frac{1}{24}$	$\frac{23}{24}$	$\frac{223}{8}$

从而,又得到新的最优解 $x^{(1)} = (\frac{7}{8}, 1)^T$, 显然,这仍不是整数解。

从上述最终表中任选一个与非整数值变量对应的约束方程作为诱导方程,比如

$$x_3 + \frac{1}{24}x_4 - \frac{1}{24}x_5 = \frac{7}{8} \quad (4-7)$$

那么得到相应的割平面条件 $\frac{7}{8} - \frac{1}{24}x_4 - \frac{23}{24}x_5 \leq 0$, 即

$$-x_4 - 23x_5 \leq -21 \quad (4-8)$$

引入松弛变量 $x_6 \geq 0$, 得到割平面方程

$$-x_4 - 23x_5 + x_6 = -21 \quad (4-9)$$

把(4-9)式加到最终表中,用对偶单纯形法继续求解,过程如下:

基矢量	a_1	a_2	a_3	$a_4 \downarrow$	a_5	$a_6 \uparrow$	b
a_2	0	1	0	0	$\frac{1}{28}$	0	1
a_1	1	0	0	$\frac{1}{24}$	$-\frac{1}{168}$	0	$\frac{7}{8}$
a_3	0	0	1	$\frac{1}{24}$	$-\frac{1}{24}$	0	$\frac{7}{8}$
a_6	0	0	0	-1	-23	1	-21
r_j	0	0	0	$\frac{1}{24}$	$\frac{23}{24}$	0	$\frac{223}{8}$

(续表)

基矢量	a_1	a_2	a_3	a_4	a_5	a_6	b
a_2	0	1	0	0	$\frac{1}{28}$	0	1
a_1	1	0	0	0	$-\frac{27}{28}$	$\frac{1}{24}$	0
a_3	0	0	1	0	-1	$\frac{1}{24}$	0
a_4	0	0	0	1	23	-1	21
r_j	0	0	0	0	0	$\frac{1}{24}$	27

因此,我们得到最优解 $x^* = (0, 1)^T$, 最优值为 -27 。

下面我们从几何图形上作进一步的说明。由 (P_0) 中的约束条件得

$$x_3 = 1 + x_1 - x_2$$

$$x_4 = 25 - 24x_1 - 4x_2$$

将它们分别代入割平面条件(4-5)中, 化简后得: $x_2 \leq 1$ 。把这一约束条件加到问题 (\tilde{P}) 中去, 形成问题 (P_1) , 相当于将问题 (\tilde{P}) 的可行域 $R(\tilde{P})$ 割去一部分(包含非整数解的部分), 参见图 6-3, 即将 $R(\tilde{P})$ 中 $x_2 > 1$ 的部分割去,

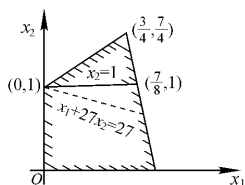


图 6-3

而割去的部分含 (\tilde{P}) 的最优解 $x_0 = \left(\frac{3}{4}, \frac{7}{4}\right)^T$ 且不含 \tilde{P} 的其他整数解。再由(4-6)式对应的最终表中的等式和(4-8)式, 整理可得第二个割平面条件等价于 $x_1 + 27x_2 \leq 27$, 把这一约束加到问题 (P_1) 中, 就相当于将问题 (P_1) 的可行域中包含问题 (P_1) 最优解且不含 (P_1) 的其他整数解的部分割去。由于在一般的 IP 问题中, 往往要经过多次切割, 才能达到这一目的, 因此, 割平面法常常是收敛很慢的, 如果与其他方法(如分枝定界法)配合使用, 则效果

会好些。

综合上述,可给出割平面法的迭代步骤如下:

1° 解纯 IP 问题(P)的松弛问题(\tilde{P}),若(\tilde{P})没有可行解,则(P)也没有可行解,停止;若(\tilde{P})的最优解 \tilde{x} 为整数解,则 \tilde{x} 即为问题(P)的最优解,停止;否则转 2°。

2° 求割平面方程。任选 \tilde{x} 的一个非整数分量 x_i (x_i 为基变量),由单纯形表的最终表得诱导方程

$$x_i + \sum_{j \in T} a_{ij} x_j = b_i \quad (T \text{ 为非基变量的下标集})$$

从而得到割平面方程

$$\sum_{j \in T} (-r_j) x_j + x_{n+t} = -r_i \quad (4-10)$$

其中

$$\begin{aligned} r_i &= b_i - [b_i] \\ r_j &= a_{ij} - [a_{ij}] \end{aligned}$$

3° 把割平面方程(4-10)加到最终表中,用对偶单纯形法继续求解。若其最优解为整数解,则它就是问题(P)的最优解,停止;否则,将这个最优解重新记为 \tilde{x} ,返回 2°。

6.5 隐枚举法

隐枚举法是求解 0—1 规划常用的一种方法。对于有 n 个变量的 0—1 规划问题,由于每个变量只取 0,1 两个值,故 n 个变量所有可能的 0—1 组合数有 2^n 个。若对这 2^n 个组合点逐点检查其可行性,并算出每个可行点上的目标函数值,再比较它们的大小以求得最优解和最优值,这种方法称为完全枚举法(或称为穷举法)。

完全枚举法只适用于变量个数较少的 0—1 规划问题。当 n 较大时,用这种方法求解,计算量将变得相当大,此时,某些问题的求解几乎是不可能的。本节介绍的隐枚举法就克服了这一缺点,它只需比较目标函数在一小部分组合点上的取值大小就能求得最优解

和最优值。

6.5.1 0—1 规划的标准形式

$$\begin{cases} \min & z = \mathbf{c}^T \mathbf{x} = \sum_{i=1}^n c_i x_i \\ \text{s. t.} & \mathbf{Ax} \leq \mathbf{b} \\ & x_i = 0 \text{ 或 } 1, i = 1, 2, \dots, n \end{cases}$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\mathbf{c} = (c_1, c_2, \dots, c_n)^T$, $\mathbf{b} = (b_1, b_2, \dots, b_m)^T$, $A = (a_{ij})_{m \times n}$, $c_i \geq 0, i = 1, 2, \dots, n$ 。

一般形式的 0—1 规划问题可按下列步骤化成标准形式：

(1) 若问题是求目标函数 $z_1 = \mathbf{c}^T \mathbf{x}$ 的极大值，则令 $z = -z_1$ ，于是，求极大值问题就变成了求极小值问题： $\min z = -\mathbf{c}^T \mathbf{x}$ ；

(2) 若目标函数中存在某个 $c_i < 0$ ，则可令 $x_i = 1 - y_i$ ，那么， $c_i x_i = c_i(1 - y_i) = c_i - c_i y_i$ ，这样， y_i 的系数 $-c_i$ 就是正数了；

(3) 若约束条件中存在不等式约束： $\sum_{j=1}^n a_{ij} x_j \geq b_i$ ，则将此不等式两边同时乘 -1 ，改写成 $\sum_{j=1}^n (-a_{ij}) x_j \leq -b_i$ ；

(4) 若约束条件中存在等式约束： $\sum_{j=1}^n a_{ij} x_j = b_i$ ，则可用两个不等式约束

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{和} \quad \sum_{j=1}^n (-a_{ij} x_j) \leq -b_i$$

来代替；

(5) 若约束条件中存在 k 个等式约束 ($k > 1$)，不妨设为

$\sum_{j=1}^n a_{ij} x_j = b_i, i = 1, 2, \dots, k$ ，则可用 $k+1$ 个不等式约束

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, 2, \dots, k$$

和

$$\sum_{i=1}^k \sum_{j=1}^n (-a_{ij})x_j \leq \sum_{i=1}^k (-b_i)$$

来代替。例如,下列两个等式约束

$$4x_1 - x_2 - 3x_3 = 1$$

$$2x_1 + 5x_2 - x_3 = 4$$

就可利用

$$4x_1 - x_2 - 3x_3 \leq 1$$

$$2x_1 + 5x_2 - x_3 \leq 4$$

$$-6x_1 - 4x_2 + 4x_3 \leq -5$$

来代替。

6.5.2 隐枚举法的基本步骤

为了明晰简便,隐枚举法的计算过程通常用枚举树图来表示,下列步骤就是针对枚举树图给出的。

0° 将 0—1 规划问题化为标准形式;检查所有变量均取零值的点(即零点)是否可行。若可行,则零点即为最优解,对应的目标值就是最优值,计算停止;否则,转 1°。

1° 令所有变量为自由变量。

2° 任选一自由变量 x_i , 令 x_i 为固定变量,则问题就被分成 $x_i = 1$ 和 $x_i = 0$ 两枝,再令所有自由变量取零值,得到每枝各一个试探解,转 3°。

3° (a) 若该枝的试探解可行,则将相应于该试探解的目标值标于该枝的旁边,并在该枝下方标记“—”; (b) 若该枝的试探解不可行,且存在一个不等式约束,将该枝的所有固定变量值代入后,所得的不等式中所有负系数之和大于右端常量,或当所有系数均为正数而最小的正数大于右端常量,则在该枝上不存在问题的可行解,在该枝下方标记“—”; (c) 若该枝的试探解不可行,且 z_0 与 c_0 之和大于已标记“—”的可行试探解目标值,其中, z_0 为该枝试探解的目标值, c_0 为目标函数中对应该枝自由变量的最小系数,则该

枝不存在问题的最优解,在该枝下方标记“—”。

4° 凡已标记“—”的枝称为已探明的枝。

5° (a) 若所有枝均已探明,则从标记“—”中找出问题的所有可行试探解,若无可行试探解,则问题无最优解,若存在可行试探解,比较所有可行试探解的目标值,选其最小者,从而得问题的最优解和最优值;(b) 若仍存在未探明的分枝,任选一未探明的枝,转 2°。

6.5.3 算法举例

【例 5.1】

$$\begin{cases} \min & z = 2x_1 + 5x_2 + 3x_3 + 4x_4 \\ \text{s. t.} & -4x_1 + x_2 + x_3 + x_4 \geq 0 \\ & -2x_1 + 4x_2 + 2x_3 + 4x_4 \geq 4 \\ & x_1 + x_2 - x_3 + x_4 \geq 1 \\ & x_i = 0 \text{ 或 } 1, i = 1, 2, 3, 4 \end{cases}$$

解 将该问题标准化,记为 (P_0) ,即

$$(P_0) \begin{cases} \min & z = 2x_1 + 5x_2 + 3x_3 + 4x_4 \\ \text{s. t.} & 4x_1 - x_2 - x_3 - x_4 \leq 0 & (5-1) \\ & 2x_1 - 4x_2 - 2x_3 - 4x_4 \leq -4 & (5-2) \\ & -x_1 - x_2 + x_3 - x_4 \leq -1 & (5-3) \\ & x_i = 0 \text{ 或 } 1, i = 1, 2, 3, 4 \end{cases}$$

检查零点 $(0, 0, 0, 0)^T$,知其不可行,任选一自由变量,如选 x_1 ,令 x_1 为固定变量,则问题 (P_0) 就分成了 $x_1 = 1$ 和 $x_1 = 0$ 两枝,分别记为问题 (P_1) 和问题 (P_2) ,即

$$(P_1) \begin{cases} \min & z = 2x_1 + 5x_2 + 3x_3 + 4x_4 \\ \text{s. t.} & 4x_1 - x_2 - x_3 - x_4 \leq 0 \\ & 2x_1 - 4x_2 - 2x_3 - 4x_4 \leq -4 \\ & -x_1 - x_2 + x_3 - x_4 \leq -1 \\ & x_1 = 1, x_i = 0 \text{ 或 } 1, i = 2, 3, 4 \end{cases}$$

$$(P_2) \begin{cases} \min & z = 2x_1 + 5x_2 + 3x_3 + 4x_4 \\ \text{s. t.} & 4x_1 - x_2 - x_3 - x_4 \leq 0 \\ & 2x_1 - 4x_2 - 2x_3 - 4x_4 \leq -4 \\ & -x_1 - x_2 + x_3 - x_4 \leq -1 \\ & x_1 = 0, x_i = 0 \text{ 或 } 1, i = 2, 3, 4 \end{cases}$$

考察问题 (P_1) , 即有试探解 $(1, 0, 0, 0)^T$, 知其不可行, 且将该枝固定变量 $x_1 = 1$ 代入约束(5-1)中得:

$-x_2 - x_3 - x_4 \leq -4$, 因为系数 $-1 - 1 - 1 > -4$, 所以, 该枝不存在问题的可行解, 不必再分。在枚举树图中此分枝下方标记“—”。考察问题

(P_2) , 即有试探解 $(0, 0, 0, 0)^T$, 知其不可行。那么再任选一个自由变量, 如选 x_2 , 令其为固定变量, 则问题(2)就分成了 $x_2 = 1$ 和 $x_2 = 0$ 两枝, 其试探解分别为 $(0, 1, 0, 0)^T$ 和 $(0, 0, 0, 0)^T$, 判断它们是否可行。

如此继续下去, 下面仅以枚举树图表示计算过程。

考察问题 (P_3) , 此时, 试探解为 $(0, 1, 0, 0)^T$, 将其代入约束(5-1)~(5-3)中, 可知是可行的。同理, 问题 (P_7) 的试探解 $(0, 0, 0, 1)^T$ 也是可行的。

考察问题 (P_5) , 此时, 试探解为 $(0, 0, 1, 0)^T$, 将其固定变量 $x_1 = 0, x_2 = 0, x_3 = 1$ 代入约束(5-3), 得 $-x_4 \leq -2$, 因为 x_4 的系数 $-1 > -2$, 所以, 此分枝上无可行解。

从枚举树图可知, 原问题的最优解为 $(0, 0, 0, 1)^T$, 最优值为4。

注: 对于 n 个变量的0-1规划问题, 为了较快地求得问题的最优解, 一般常重新排列 x_i 的顺序, 使目标函数中 x_i 的系数递增, 即使 $c_1 \leq c_2 \leq \dots \leq c_{n-1} \leq c_n$ 成立, 或在每分枝选固定变量时, 先

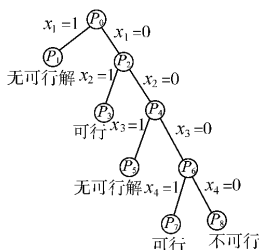


图 6-4

选其在目标函数中的系数较小的变量为固定变量。

6.6 匈牙利法

匈牙利法是匈牙利数学家考尼格(könig)提出的求解指派问题的一种方法。

从数学模型可知,指派问题是一种特殊的 0—1 规划问题和特殊的运输问题,因此可以采用前几节介绍的整数规划方法或用运输问题解法来求解,但由于其独特的模型结构,采用前述的方法求解时,要么计算繁杂,要么需在相当多的退化解之间进行转换,因而所需的计算时间较长。本节介绍的匈牙利法依据指派问题的特点较好地解决了这个问题。

6.6.1 指派问题标准形式

6.1 节例 1.7 给出的指派问题模型为其标准形式。记 $C = (c_{ij})_{n \times n}$ 为指派问题的系数矩阵, $c_{ij} \geq 0, i, j = 1, \dots, n$ 。在实际问题中,根据 c_{ij} 的具体意义, C 可以有不同的名称,如费用矩阵、成本矩阵、时间矩阵等。

问题的每一个可行解可用矩阵表示为

$$X = (x_{ij})_{n \times n} \quad (6-1)$$

其中 X 中每行元素之和或每列元素之和为 1, 且 $x_{ij} = 0$ 或 1, $i, j = 1, \dots, n$ 。

6.6.2 基本原理

匈牙利法的主要理论依据是下面的两个定理。

定理 6.1 设指派问题的系数矩阵为 $C = (c_{ij})_{n \times n}$, 若将 C 的第 i 行各元素减去 u_i , 将 C 的第 j 列各元素减去 $v_j, i, j = 1, \dots, n$, 则所得的新的系数矩阵 $C' = (c'_{ij})_{n \times n}$ 对应的指派问题的最优解与 C 对应的指派问题的最优解一致。

证明 因为

$$c'_{ij} = c_{ij} - u_i - v_j, i, j = 1, \dots, n$$

所以

$$\begin{aligned} \sum_{i,j=1}^n c'_{ij} x_{ij} &= \sum_{i,j=1}^n c_{ij} x_{ij} - \sum_{i,j=1}^n u_i x_{ij} - \sum_{i,j=1}^n v_j x_{ij} \\ &= \sum_{i,j=1}^n c_{ij} x_{ij} - \sum_{i=1}^n (u_i \sum_{j=1}^n x_{ij}) - \sum_{j=1}^n (v_j \sum_{i=1}^n x_{ij}) \\ &= \sum_{i,j=1}^n c_{ij} x_{ij} - \sum_{i=1}^n u_i - \sum_{j=1}^n v_j \end{aligned}$$

其中应用了 $\sum_{j=1}^n x_{ij} = \sum_{i=1}^n x_{ij} = 1$ 这个条件 ($i, j = 1, \dots, n$), 所以, 以 C' 为系数矩阵的指派问题与以 C 为系数矩阵的指派问题的最优解一致。 ■

定理 6.2 若一方阵中的一部分元素为零, 另一部分元素非零, 则覆盖方阵内所有零元素的最少直线数等于位于该方阵中位于不同行列的零元素的最多个数。

使用定理 6.1 变换系数矩阵, 使其含有许多零元素, 并保证变换后的系数矩阵各元素不小于零。这样若能找到 n 个位于不同行、不同列的零元素, 则在问题的解矩阵 X 中, 令这 n 个零对应的位置上元素为 1, 其余元素为 0, 便得到了指派问题的一个最优解。若不能找到 n 个位于不同行、不同列的零元素, 则再利用定理 6.1, 将系数矩阵中零元素的位置做适当调整, 不断地这样做下去, 直至找到 n 个位于不同行、不同列的零元素为止。

6.6.3 基本步骤

(1) 变换系数矩阵

使变换后的系数矩阵各行各列均出现零元素, 且每个元素不小于零。

如可将系数矩阵 C 中各行各列元素依次减去所在行或列的最

小元素。记变换后的矩阵为 $C' = (c'_{ij})_{n \times n}$ 。

(2) 找 C' 中位于不同行、不同列的零元素

(I) 找到 C' 中未加标记的零元素最少的行(列), 从该行(列)圈出一个零元素。若该行(列)中有几个零元素, 可以任圈其中一个, 用 ① 表示。

(II) 从由(I)得到的 ① 元素所在行、列中, 划去其余零元素, 用 \times 表示。

(III) 凡是 ① 或 \times 均为加了标记的元素。若 C' 中仍存在未加标记的 ① 元素, 返回(I), 否则结束。

标记的 ① 零元素一定位于不同行、不同列, 若其个数为 n , 则可得问题的最优解, 否则进行(3)。

(3) 找出能覆盖 C' 中所有零元素的最少直线集合

(I) 对没有 ① 的行打 \checkmark ;

(II) 对打 \checkmark 号的行上所有 \times 元素的列打 \checkmark 号;

(III) 对打 \checkmark 号的列上有 ① 的行打 \checkmark 号;

(IV) 重复(II)、(III), 直至找不出新的打 \checkmark 号的行、列为止;

(V) 对没有打 \checkmark 号的行划横线, 对所有打 \checkmark 号的列划竖线, 这些直线便为所求的直线集合。

(4) 零元素的移动

下面的步骤是基于定理 6.1 和步骤(3)得到的。

(I) 在未被直线覆盖的所有元素中, 找出最小元素;

(II) 对未划直线的行中各元素都减去这个最小元素;

(III) 对划直线的列中各元素都加上这个最小元素。

这样便得到一个新矩阵, 仍记为 C' , 返回(2)。

【例 6.1】 有四项工作需甲、乙、丙、丁四个人去完成。每项工作只由一个人去完成, 每个人只完成其中一项工作。已知每个人完成各项工作的时间(h)如下表所示。问应如何指派, 才能使总的耗时间最少?

时间 人	工作	I	II	III	IV
甲		10	8	12	9
乙		7	10	6	11
丙		9	7	11	10
丁		12	6	9	8

解

$$\begin{aligned}
 (1) \quad C &= \begin{pmatrix} 10 & 8 & 12 & 9 \\ 7 & 10 & 6 & 11 \\ 9 & 7 & 11 & 10 \\ 12 & 6 & 9 & 8 \end{pmatrix} \xrightarrow{(I)} \begin{pmatrix} 2 & 0 & 4 & 1 \\ 1 & 4 & 0 & 5 \\ 2 & 0 & 4 & 3 \\ 6 & 0 & 3 & 2 \end{pmatrix} \\
 &\xrightarrow{(II)} \begin{pmatrix} 1 & 0 & 4 & 0 \\ 0 & 4 & 0 & 4 \\ 1 & 0 & 4 & 2 \\ 5 & 0 & 3 & 1 \end{pmatrix}
 \end{aligned}$$

其中 (I) 步是将 C 中各行元素减去各行中的最小元素, (II) 步是将所得的矩阵中各列元素减去各列中的最小元素。

$$(2) \quad \begin{pmatrix} 1 & \infty & 4 & \textcircled{0} \\ \textcircled{0} & 4 & \infty & 4 \\ 1 & \textcircled{0} & 4 & 2 \\ 5 & \infty & 3 & 1 \end{pmatrix}$$

✓

$$(3) \quad \begin{pmatrix} 1 & \infty & 4 & \textcircled{0} \\ \textcircled{0} & 4 & \infty & 4 \\ 1 & \textcircled{0} & 4 & 2 \\ 5 & \infty & 3 & 1 \end{pmatrix} \begin{matrix} \checkmark \\ \checkmark \\ \checkmark \\ \checkmark \end{matrix}$$

$$(4) \begin{pmatrix} 1 & 1 & 4 & \textcircled{0} \\ \textcircled{0} & 5 & \textcircled{0} & 4 \\ \textcircled{0} & \textcircled{0} & 3 & 1 \\ 4 & \textcircled{0} & 2 & \textcircled{0} \end{pmatrix}$$

有 4 个 ① 位于不同行、不同列。因此,最优指派为:甲做第 IV 项工作,乙做第 III 项工作,丙做第 I 项工作,丁做第 II 项工作。

注:对于求极大值问题,如

$$\begin{cases} \max & z = \sum_{i,j=1}^n c_{ij}x_{ij} \\ \text{s. t.} & \sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n \\ & x_{ij} = 0 \text{ 或 } 1, i, j = 1, \dots, n \end{cases} \quad (6-2)$$

可令 $z_1 = -z$, 将 (6-2) 的目标函数转化为求极小值问题, 即

$$\min z_1 = - \sum_{i,j=1}^n c_{ij}x_{ij} \text{ 或 } \min z_2 = \sum_{i,j=1}^n (M - c_{ij})x_{ij} (M \geq \max_{i,j} \{c_{ij}\}),$$

再利用上述方法求解即可。 ■

习 题 六

1. 设有 n 件工作要完成, 恰好有 n 个人可以分别去完成其中的每一件。若第 i 个人完成第 j 件工作所需的时间为 t_{ij} , 问应如何分派, 才能使花费的总时间最少, 试建立数学模型。

2. 加工任务分配问题

设有 m 台同一类型的机床, 有 n 种零件各一个要在这些机床上进行加工, 一个第 j 种零件需加工 a_j 机时, 问应如何分配加工任务, 才能使各机床的负荷尽可能均衡, 试建立数学模型。

3. 板材合理下料问题

设有同一类型的钢板若干张, 要用它们切割成 m 种零件 A_1, A_2, \dots, A_m

的毛料,根据既省料又容易操作的原则,人们在一块钢板上,已经设计出 n 种不同的下料方案,设在第 j 种下料方案中,可下得第 i 种零件 A_i 的个数为 a_{ij} ,第 i 种零件的需要量为 $b_i (i = 1, 2, \dots, m)$ 。问应如何下料,才能既满足需要,又使所用的钢板总数最少?试建立数学模型。

4. 设钢筋长 10m,现需截取长 3m 和 4m 的两种钢筋各 50 根,问怎样截法方能使所用的钢筋根数最少。

5. 用分枝定界法求解如下的 IP 问题:

$$(1) \begin{cases} \max & z = 3x_1 + 2x_2 \\ \text{s. t.} & x_1 + x_2 \leq 3.5 \\ & 0 \leq x_1 \leq 2, 0 \leq x_2 \leq 2 \\ & x_i \in I, i = 1, 2 \end{cases}$$

$$(2) \begin{cases} \max & z = x_1 + x_2 \\ \text{s. t.} & 2x_1 + x_2 \leq 6 \\ & 4x_1 + 5x_2 \leq 20 \\ & x_i \in I, i = 1, 2 \end{cases}$$

$$(3) \begin{cases} \max & z = 6x_1 + 4x_2 + x_3 \\ \text{s. t.} & 3x_1 + 2x_2 + x_3 \leq 20 \\ & 6x_1 + 5x_2 + x_3 \leq 25 \\ & x_1 + 3x_2 + 3x_3 \leq 10 \\ & x_i \geq 0, x_i \in I, i = 1, 2, 3 \end{cases}$$

$$(4) \begin{cases} \max & z = x_1 + x_2 \\ \text{s. t.} & 14x_1 + 9x_2 \leq 51 \\ & -6x_1 + 3x_2 \leq 1 \\ & x_1, x_2 \geq 0, x_i \in I, i = 1, 2 \end{cases}$$

$$(5) \begin{cases} \min & z = -x_1 - 4x_2 \\ \text{s. t.} & -2x_1 + 3x_2 \leq 3 \\ & x_1 + 2x_2 \leq 8 \\ & x_1, x_2 \geq 0 \\ & x_1 \in I \end{cases}$$

6. 用割平面法求解如下的整数线性规划问题:

$$\begin{aligned}
 (1) \quad & \begin{cases} \min & z = -3x_1 - 2x_2 \\ \text{s. t.} & 2x_1 + 3x_2 \leq 14 \\ & 2x_1 + x_2 \leq 9 \\ & x_1, x_2 \geq 0, x_1, x_2 \in I \end{cases} \\
 (2) \quad & \begin{cases} \max & z = x_1 + x_2 \\ \text{s. t.} & 2x_1 + x_2 \leq 6 \\ & 4x_1 + 5x_2 \leq 20 \\ & x_i \geq 0, x_i \in I, i = 1, 2 \end{cases}
 \end{aligned}$$

7. 求解如下的 IP 问题:

$$\begin{aligned}
 (1) \quad & \begin{cases} \min & z = -4x_1 - 5x_2 \\ \text{s. t.} & 3x_1 + 2x_2 \leq 10 \\ & x_1 + 4x_2 \leq 11 \\ & x_i \geq 0, x_i \in I, i = 1, 2 \end{cases} \\
 (2) \quad & \begin{cases} \min & z = 10 - 2x_1 + x_2 + x_3 \\ \text{s. t.} & 2x_1 + 3x_2 - x_3 \leq 9 \\ & 2x_2 + 3x_3 \geq 4 \\ & 3x_1 + 3x_2 + 3x_3 = 6 \\ & x_i \in \{0, 1\}, i = 1, 2, 3 \end{cases} \\
 (3) \quad & \begin{cases} \min & z = 7x_1 + 3x_2 + 4x_3 \\ \text{s. t.} & x_1 + 2x_2 + 3x_3 \geq 8 \\ & 3x_1 + x_2 + x_3 \geq 5 \\ & x_i \geq 0, x_i \in I, i = 1, 2, 3 \end{cases}
 \end{aligned}$$

8. 用隐枚举法求解下列 0-1 规划问题:

$$\begin{aligned}
 (1) \quad & \begin{cases} \min & z = 7x_1 + 5x_2 + 6x_3 + 8x_4 + 9x_5 \\ \text{s. t.} & 3x_1 - x_2 + x_3 + x_4 - 2x_5 \geq 2 \\ & -x_1 - 3x_2 + x_3 + 2x_4 - x_5 \leq 0 \\ & -x_1 - x_2 + 3x_3 + x_4 + x_5 \geq 1 \\ & x_j = 0 \text{ 或 } 1, j = 1, 2, 3, 4, 5 \end{cases} \\
 (2) \quad & \begin{cases} \min & z = 4x_1 + 3x_2 + 2x_3 \\ \text{s. t.} & 2x_1 - 5x_2 + 3x_3 \leq 4 \\ & 4x_1 + x_2 + 3x_3 \geq 3 \\ & x_2 + x_3 \geq 1 \\ & x_j = 0 \text{ 或 } 1, j = 1, 2, 3 \end{cases}
 \end{aligned}$$

$$(3) \begin{cases} \max & z = 3x_1 - 2x_2 + 5x_3 \\ \text{s. t.} & x_1 + 2x_2 - x_3 \leq 2 \\ & x_1 + 4x_2 + x_3 \leq 4 \\ & x_1 + x_2 \leq 3 \\ & 4x_2 + x_3 \leq 6 \\ & x_i = 0 \text{ 或 } 1, i = 1, 2, 3 \end{cases}$$

$$(4) \begin{cases} \max & z = 2x_1 - x_2 + 5x_3 - 3x_4 + 4x_5 \\ \text{s. t.} & 3x_1 - 2x_2 + 7x_3 - 5x_4 + 4x_5 \leq 6 \\ & x_1 - x_2 + 2x_3 - 4x_4 + 2x_5 \leq 0 \\ & x_j = 0 \text{ 或 } 1, j = 1, 2, 3, 4, 5 \end{cases}$$

9. 某运输队有五辆汽车,待驶往五个目的地送货。一地的货物只需一辆汽车运送,其运费(元)如下表所示:

目 汽 车 的 地	1	2	3	4	5
A	100	120	140	110	130
B	140	200	230	150	210
C	80	300	100	90	70
D	120	160	200	130	250
E	70	140	190	150	220

- (1) 试求最优调运方案;
- (2) 若表中数字表示所得利润,则应如何调运?
- (3) 若目的地只有 A、B、C, (1)、(2) 的结果又如何?

10. 某厂拟用五台机床加工五种零件,其加工费(元)如下表所示,若每台机床只限加工一种零件,则应如何分配任务才能使总加工费最少?

机 零 件 床	1	2	3	4	5
1	4	1	8	4	2
2	9	8	4	7	7
3	8	4	6	6	3
4	6	5	7	6	2
5	5	5	4	3	1

第 7 章 动 态 规 划

动态规划(Dynamic Programming)是解决多阶段决策过程最优化问题的一种方法。1951 年,美国数学家贝尔曼(R. Bellman)研究了一类多阶段决策问题的特征,提出了解决这类问题的基本原理,在研究、解决了某些实际问题的基础上,于 1957 年发表了《动态规划》这一名著。

动态规划把比较复杂的问题划分成若干阶段,通过逐段求解,最终求得全局最优解。这种“分而治之,逐步改善”的方法已在一些较难解决的问题中显示出优越性,尤其是离散性问题,用动态规划的方法去处理,比用线性规划或非线性规划方法有时更为有效。迄今为止,动态规划已广泛应用于经济、生物、工程、军事等许多领域,并已取得了很好的效果。

本章首先介绍动态规划的基本概念、基本原理和基本方法,然后介绍动态规划中常用的两种迭代方法:函数空间迭代法和策略空间迭代法,最后给出一些实例,期望读者从这些实例中掌握求解动态规划的一些技巧。

7.1 多阶段决策问题

所谓多阶段决策问题是指这样一类活动过程:它可以分为若干个相互联系的阶段,在每一个阶段上都需要作出决策,而一个阶段的决策确定以后,将会影响以后各阶段的活动及其决策,当所有阶段的决策确定以后,就完全确定了该问题的活动过程。各个阶段

所确定的决策就构成了一个决策序列,称为一个策略。一般来说,由于每一阶段可供选择的决策往往不止一个,因此,对于整个过程,就会有许多可供选择的策略。若对应于一个策略,可以由一个量化的指标来确定这个策略所对应的活动过程的效果,那么,不同的策略就有各自的效果。在所有可供选择的策略中,对应效果最好的策略称为最优策略。把一个问题划分成若干个相互联系的阶段,选取其最优策略,这类问题就是多阶段决策问题。它要求决策者不能用孤立的观点看待各阶段的决策。

多阶段决策问题,不论其本身是否与时间有关,由于分为阶段来依次解决,这便具有了明显的时序性,而在各阶段中所采取的决策是随阶段而变动的,不同阶段采取不同决策,这便是“动态”的含义。阶段往往可以用时段来表示,但动态规划在一定条件下也可以解决一些与时间无关的静态最优化问题,只要人为地引入“时段”因素,就可以将其转化为一个多阶段决策问题。

下面给出几个多阶段决策问题的实例。

【例 1.1】 最短路线问题

图 7-1 表示某城市的局部道路分布图。一货运汽车从 S 出发,最终到达目的地 E 。其中 $A_i (i = 1, 2, 3)$, $B_j (j = 1, 2)$ 和 $C_k (k = 1, 2)$ 是可供汽车选择的途经站点,各点连线上的数字表示两个站点间的距离。问:汽车应走哪条路线所经过的路线距离最短?

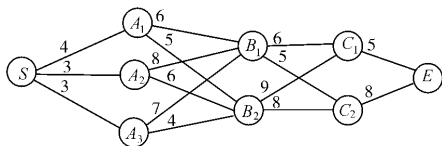


图 7-1

此例中,我们可以把从 S 到 E 这一过程划分成 4 个阶段:(1) $S \rightarrow A$, (2) $A \rightarrow B$, (3) $B \rightarrow C$, (4) $C \rightarrow E$ 。在每个阶段作出一个恰当

的决策,使由这些决策组成的策略所决定的路线距离最短。

【例 1.2】 资源分配问题

某公司订购 4 箱海鲜品,分配给下属的 I、II、III 三个门市部,各门市部获得海鲜品后的盈利如表 7-1 所示。试问:该公司如何分配海鲜品,才能获得最大利润?

表 7-1

门 市 部 \ 箱 数	0	1	2	3	4
I	0	4	7	8	10
II	0	2	4	8	9
III	0	5	7	9	11

【例 1.3】 投资决策问题

某公司决定在今后三年的每一年开始将资金投入 A 或 B 工程,年末的收入及其概率如表 7-2 所示。每年至多作一项投资,每次只能投入 1000 万元。求出使第三年末所拥有的期望金额达到最大的投资方案。

表 7-2

投资	收入(万元)	概率
A	0	0.4
	2000	0.6
B	1000	0.9
	2000	0.1

多阶段决策问题种类很多,除上述问题外,还有生产与贮存问题,系统可靠性问题,机器负

荷分配问题,定价问题,采购问题,设备更新问题,最优控制问题,等等。

7.2 动态规划的基本原理

7.2.1 动态规划的基本概念

1. 阶段

为了便于求解,常常把一个问题的整个活动过程根据时间、空间等自然因素划分成相互联系的若干阶段。通过逐步分析求解这几个阶段,最终得到最优解。最常遇到的阶段变量是离散的,因此,

一般用 k 来表示阶段序号 $1, 2, \dots$, 称 k 为阶段变量。

在例 1.1 中, 我们把它划分成四个阶段: (1) $S \rightarrow A$, (2) $A \rightarrow B$, (3) $B \rightarrow C$, (4) $C \rightarrow E$ 。

2. 状态

一个阶段的过程在开始时所面临的状况, 称为这个阶段过程的状态。例 1.1 中某阶段的状态就是它的始点, 如, 第一阶段只有一个状态 S , 第三阶段有两个状态 B_1 和 B_2 。描述过程状态的变量称为状态变量, 常用 x_k 表示第 k 阶段的某个状态。状态可用数字、字母等表示, 例 1.1 中, 就用 S, A_2, B_1, C_2, E 等字母表示各阶段的某些状态。状态变量可以是离散的, 或者是连续的。在实际问题中, 动态规划应用的成败, 通常取决于适当的规定状态变量。用 X_k 表示第 k 阶段的状态集合。例 1.1 中, $X_3 = \{B_1, B_2\}$, X_k 就是第 k 阶段所有始点的集合。

3. 决策

所谓决策是指一个阶段的状态确定以后, 从该状态演变到下一阶段的某一状态的一种选择。用来描述这种选择的变量称为决策变量。每一阶段的决策都依赖于该阶段的状态, 假设用 u_k 来表示第 k 阶段的决策变量, 则 $u_k = u_k(x_k)$ 。决策变量允许选择的范围, 称为允许决策集合, 记为 $U_k(x_k)$, $k = 1, 2, \dots$ 。显然, 每个决策变量都应在允许决策集合中选取, 即 $u_k(x_k) \in U_k(x_k)$ 。例 1.1 中, 第三阶段的状态集合 $X_3 = \{B_1, B_2\}$, 而确定了 B_1 以后, 其允许决策集合为 $U_3(B_1) = \{C_1, C_2\}$, 若选取该阶段的终点为 C_2 , 则 $u_3(B_1) = C_2$, 显然, $u_3(B_1) \in U_3(B_1)$ 。例 1.1 中, 在第一阶段, 若我们做出了走至 A_2 的决策, 则 A_2 就是第一阶段在我们的决策之下的结果, 它既是第一阶段的终点, 又是第二阶段的始点。从图 7-1 中还可以看出, 当某阶段的始点给定时, 它直接影响着后面各阶段的路线及整个路线的长短。

4. 策略

由各个阶段的决策 $u_k(x_k), k=1, 2, \dots, n$ 组成的决策序列, 称为一个全过程策略, 简称策略。可以表示为 $P_{1n}(x_1)$, 即: $P_{1n}(x_1) = \{u_1(x_1), u_2(x_2), \dots, u_n(x_n)\}$, 其中, $u_i(x_i) \in U_i(x_i)$ 。一般可供选择的策略有一定范围, 我们称这个范围为允许策略集合, 用 P_1 表示。从第 k 阶段到最终阶段各阶段决策所构成的决策序列, 称为 k 子过程策略, 简称子策略, 记为 $P_{kn}(x_k)$, 即: $P_{kn}(x_k) = \{u_k(x_k), u_{k+1}(x_{k+1}), \dots, u_n(x_n)\}$, 用 P_k 表示 k 子过程允许策略集合。在允许(子)策略集合中, 取得最优效果的(子)策略称为最优(子)策略。

5. 状态转移方程

作为在第 k 阶段某状态 x_k 下所做出的决策 $u_k(x_k)$ 的结果, 过程由第 k 阶段的状态 x_k 转移到第 $k+1$ 阶段的状态 x_{k+1} 。描述这种变化关系的方程称为状态转移方程。对于第 k 阶段, 状态转移方程依赖于第 k 阶段的状态及其决策, 即由第 k 阶段的状态及其决策可以确定第 $k+1$ 阶段的状态, 亦即 $x_{k+1} = T_k(x_k, u_k(x_k))$, 这就是状态转移方程, 它确定了由一个阶段的状态到下一个阶段的状态的演变过程。例 1.1 中, 对第二阶段的状态 A_1 , 若 $u_1(A_1) = B_1$, 则第三阶段的状态 $x_3 = B_1 = T_2(A_1, u_1(A_1))$ 。

6. 指标函数

用以衡量、评价所选取的策略或子策略或决策的优劣程度或效果的数量函数, 称为指标函数。指标函数是定义在全过程、后部子过程或各阶段上的确定性数量函数。这里的后部子过程是指从某阶段开始的后继过程。

(1) 阶段指标函数

用 $V_k(x_k, u_k(x_k))$ 表示第 k 阶段处于状态 x_k 且所做决策为 $u_k(x_k)$ 时的指标, 简记为 V_k 。例 1.1 的 V_k 值就是从状态 x_k 到状态

x_{k+1} 间的距离。如, $V_2(A_1, B_1) = 6$ 表示 A_1 到 B_1 的距离为 6。

(2) 过程指标函数

用 V_{kn} 表示从第 k 阶段开始的后部子过程上的过程指标函数, 可表示为: $V_{kn} = V_{kn}(x_k, u_k, x_{k+1}, u_{k+1}, \dots, u_n, x_{n+1})$, 它是从第 k 阶段开始到最后阶段为止的状态和决策的函数。

指标函数在不同阶段上可以是不一样的。

7. 最优解

用 $f_k(x_k)$ 表示 V_{kn} 的最优值, 它是状态 x_k 的函数, 称其为最优指标函数。即 $f_k(x_k) = \underset{P_{kn}(x_k) \in P_k}{\text{opt}} V_{kn}, k = 1, 2, \dots, n$ 。式中, opt 指“最优”, 可以是 min 或 max。相应的(子)策略称为最优(子)策略, 记为 $P_{kn}^*(x_k) = \{u_k^*, u_{k+1}^*, \dots, u_n^*\}$ 。

当 $k=1$, 且该阶段只有惟一状态时, $f_1(x_1)$ 就是问题的最优值, 相应的策略 $P_{1n}^*(x_1) = \{u_1^*, u_2^*, \dots, u_n^*\}$ 就是最优策略。

当 $k=1$, 该阶段的状态不惟一时, 最优值记为 f_0 , 有 $f_0 = \underset{x_1 \in X_1}{\text{opt}} \{f_1(x_1)\} = f_1(x_1^*)$, 此时的最优策略为 $P_{1n}^*(x_1 = x_1^*) = \{u_1(x_1^*), u_2^*, u_3^*, \dots, u_n^*\}$ 。最优策略和最优值统称为问题的最优解。

7.2.2 动态规划的最优化原理

在例 1.1 中, 共有 12 条全程路线可供选择, 用穷举法(或称完全枚举法)便可算出各条路线的长度, 从中找出距离最短的路线: $S \rightarrow A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow E$, $S \rightarrow A_3 \rightarrow B_1 \rightarrow C_1 \rightarrow E$ 和 $S \rightarrow A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow E$ 。但当阶段数很多, 且各阶段可供选择的决策也很多时, 穷举法的计算量将变得特别大, 因而是不足取的, 也是不可行的。

下面我们介绍求解例 1.1 的另一种方法——标号法。

标号法是在网络图中直观表示的一种方法。通常标号法采取“逆序求解”的方式来寻找问题的最优解, 即从最后阶段开始, 逐

次向阶段数小的方向推算,最终求得全局最优解。

下面给出标号法的一般步骤:

1° 从最后一段标起,该段各状态(即各始点)到终点的距离用数字分别标在各点上方的方格内,并用粗箭线连接各点和终点。

2° 向前递推,即给前一阶段的各个状态标号。每个状态上方方格内的数字表示该状态到终点的最短距离,即为该状态到该阶段已标号的各终点的段长,再分别加上对应终点上方方格内的数字,而取其最小者。将刚标号的点沿着最短距离所对应的已标号的点用粗箭线连接起来,表示出各刚标号的点到终点的最短路线。

3° 逐次向前递推,直到将第一阶段的状态(即起点)也标号,起点方格内的数字就是起点到终点的最短距离,从起点开始连接终点的粗箭线就是最短路线。

用标号法来求解例 1.1,结果如图 7-2 所示。

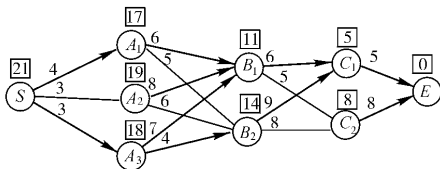


图 7-2

每个状态上方的方格内的数字表示该状态到 E 的最短距离,首尾相连的粗箭线构成每一状态到 E 的最短路线。因此,标号法不但给出起点到终点的最短路线和最短距离,同时也给出每一状态到终点的最短路线及其最短距离。如, A_1 到 E 的最短路线是 $A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow E$,最短距离是 17。

1. 最优化原理

假设某最短路线问题有一条最短路线为: $S = R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_k \rightarrow R_{k+1} \rightarrow \cdots \rightarrow R_n \rightarrow R_{n+1} = E$, 则这条路线上的任一状态 R_k 到 E 的路线必是 R_k 到 E 的最短路线之一。

用反证法不难证明上述结论。若 $R_k \rightarrow R_{k+1} \rightarrow \cdots \rightarrow R_n \rightarrow R_{n+1} = E$ 不是 R_k 到 E 的一条最短路线, 则必存在比路线 $R_k \rightarrow R_{k+1} \rightarrow \cdots \rightarrow R_n \rightarrow R_{n+1} = E$ 短的路线, 不妨记为 $R_k \rightarrow R'_{k+1} \rightarrow \cdots \rightarrow R'_n \rightarrow R_{n+1} = E$, 那么, 全程路线 $S = R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_k \rightarrow R'_{k+1} \rightarrow R'_{k+2} \rightarrow \cdots \rightarrow R'_n \rightarrow R_{n+1} = E$ 要比 $S = R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_k \rightarrow R_{k+1} \rightarrow \cdots \rightarrow R_n \rightarrow R_{n+1} = E$ 短, 这与假设矛盾, 故上述结论成立。

最短路线问题的这一特性推广到一般多阶段决策问题时, 可以表述成下列最优化原理(贝尔曼最优化原理):

整个过程的最优策略具有这样的性质: 对于最优策略过程中的任一状态而言, 无论其过去的状态和决策如何, 其余下的诸决策必构成一个最优子策略。

例 1.1 中, $S \rightarrow A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow E$ 是一条最短路线, 那么, 对 B_1 而言, 不论前面如何走, $B_1 \rightarrow C_1 \rightarrow E$ 必构成一条 B_1 到 E 的最短路线。

2. 函数基本方程(或称泛函方程)

例 1.1 中, 用标号法求最短路线的计算公式可以概括写成:

$$\begin{cases} f_5(x_5) = 0 \\ f_k(x_k) = \min_{u_k \in U_k(x_k)} \{d_k(x_k, u_k(x_k)) + f_{k+1}(u_k(x_k))\} \\ k = 4, 3, 2, 1 \end{cases}$$

其中, $d_k(x_k, u_k(x_k))$ 表示从状态 x_k 到由决策 $u_k(x_k)$ 所决定的状态 x_{k+1} 之间的距离。 $f_k(x_5) = 0$ 是边界条件, 表示全过程到第四阶段终点结束。

一般地, 对于 n 个阶段的决策过程, 假设只考虑指标函数是“和”与“积”的形式, 第 k 阶段和第 $k+1$ 阶段间的递推公式可表示如下:

(1) 当过程指标函数为下列“和”的形式时,

$$V_{kn} = \sum_{i=k}^n V_i(x_i, u_i(x_i))$$

则函数基本方程为

$$\begin{cases} f_{n+1}(x_{n+1}) = 0 \\ f_k(x_k) = \operatorname{opt}_{u_k \in U_k(x_k)} \{V_k(x_k, u_k(x_k)) + f_{k+1}(T_k(x_k, u_k(x_k)))\} \end{cases} \quad (7-1)$$

$$k = n, n-1, \dots, 2, 1$$

(2) 当过程指标函数为下列“积”的形式时,

$$V_{kn} = \prod_{i=k}^n V_i(x_i, u_i(x_i))$$

则函数基本方程为

$$\begin{cases} f_{n+1}(x_{n+1}) = 1 \\ f_k(x_k) = \operatorname{opt}_{u_k \in U_k(x_k)} \{V_k(x_k, u_k(x_k)) \cdot f_{k+1}(T_k(x_k, u_k(x_k)))\} \end{cases} \quad (7-2)$$

$$k = n, n-1, \dots, 2, 1$$

注意, 以上两种情况所对应的边界条件是不同的。

3. 动态规划的基本方法

标号法仅适用于求解像最短路线问题那样可以用网络图表示的多阶段决策问题。但不少多阶段决策问题不能用网络图表示。此时, 应该用函数基本方程来递推求解。

一般来说, 要用函数基本方程逆推求解, 首先要有效地建立动态规划模型, 然后再递推求解, 最后得出结论。

建立动态规划模型一般可按下列步骤进行:

1° 根据时间和空间的自然特征, 把实际问题恰当地划分为若干个阶段, 使问题能够转化成是一个多阶段决策问题。

2° 正确地选择状态变量 x_k , 确定状态集合 X_k , 使状态变量既能描述过程的演变特征, 又能满足无后效性及具备可知性。

这里所说的可知性, 是指所规定的各阶段的状态变量的值可以直接或间接知道。而无后效性, 也称马尔科夫性, 是指若某个阶段的状态给定, 则从该阶段以后, 过程的发展只与当前的状态有关, 而与过去的历史无关。也就是说, 在一个过程中, 过去的历史只能通过当前的状态去影响过程未来的发展, 当前的状态就是未

来过程的初始状态。无后效性的要求是为了保证下一阶段的状态及其指标函数最优值能惟一确定。

3° 确定决策变量 u_k 及每个阶段的允许决策集合 $U_k(x_k) = \{u_k\}$ 。

4° 写出状态转移方程,即给出从第 k 阶段到第 $k+1$ 阶段的状态转移规律:

$$x_{k+1} = T_k(x_k, u_k)$$

5° 正确地写出指标函数 V_{kn} , 正确的指标函数一般应满足三个条件:①它是定义在过程上的数量函数;②满足递推关系,或称具有可分性,即 $V_{k,n}(x_k, u_k, \dots, x_{n+1}) = \phi[x_k, u_k, V_{k+1,n}(x_{k+1}, u_{k+1}, \dots, x_{n+1})]$; ③ $\phi[x_k, u_k, V_{k+1,n}(x_{k+1}, u_{k+1}, \dots, x_{n+1})]$ 对其变元 $V_{k+1,n}$ 严格单调。

6° 写出动态规划函数基本方程

下面用上述方法具体求解例 1.1。

第一步:先考虑第四阶段,即 $k=4$,该阶段共有两个状态: C_1, C_2 ,设 $f_4(C_1)$ 和 $f_4(C_2)$ 分别表示 C_1, C_2 到 E 的最短距离,显然有 $f_4(C_1) = 5, f_4(C_2) = 8$,边界条件 $f_5(E) = 0$ 。

第二步:即 $k=3$,该阶段共有两个状态: B_1, B_2 。

(1) 从 B_1 出发有两种决策: $B_1 \rightarrow C_1, B_1 \rightarrow C_2$ 。记 $d_3(B_1, C_1)$ 表示 B_1 到 C_1 的距离,即,这里的每一种决策的阶段指标函数就是距离,所以, $B_1 \rightarrow C_1$ 的阶段指标函数为 $d_3(B_1, C_1) = 6, B_1 \rightarrow C_2$ 的阶段指标函数为 $d_3(B_1, C_2) = 5$ 。因此,有

$$f_3(B_1) = \min\{d_3(B_1, C_1) + f_4(C_1)$$

$$d_3(B_1, C_2) + f_4(C_2)\} = \min\{6 + 5, 5 + 8\} = 11。$$

那么,从 B_1 出发到 E 的最短路线是 $B_1 \rightarrow C_1 \rightarrow E$, 对应的决策 $u_3(B_1) = C_1$ 。

(2) 从 B_2 出发也有两种决策: $B_2 \rightarrow C_1, B_2 \rightarrow C_2$ 。同理,有

$$f_3(B_2) = \min\{d_3(B_2, C_1) + f_4(C_1)$$

$$d_3(B_2, C_2) + f_4(C_2) \} = \min \{ \underbrace{9+5}_{\dots\dots\dots}, 8+8 \} = 14$$

那么,从 B_2 出发到 E 的最短路线是 $B_2 \rightarrow C_1 \rightarrow E$, 且 $u_3(B_2) = C_1$ 。

第三步:即 $k=2$, 该阶段共有三个状态: A_1, A_2, A_3 。

(1) 从 A_1 出发有两种决策: $A_1 \rightarrow B_1, A_1 \rightarrow B_2$ 。则

$$f_2(A_1) = \min \{ d_2(A_1, B_1) + f_3(B_1) \}$$

$$d_2(A_1, B_2) + f_3(B_2) \} = \min \{ \underbrace{6+11}_{\dots\dots\dots}, 5+14 \} = 17$$

即 A_1 到 E 的最短路线为 $A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow E$, 且 $u_2(A_1) = B_1$ 。

(2) 从 A_2 出发也有两种决策: $A_2 \rightarrow B_1, A_2 \rightarrow B_2$ 。此时,

$$f_2(A_2) = \min \{ d_2(A_2, B_1) + f_3(B_1) \}$$

$$d_2(A_2, B_2) + f_3(B_2) \} = \min \{ \underbrace{8+11}_{\dots\dots\dots}, 6+14 \} = 19$$

即, A_2 到 E 的最短路线为 $A_2 \rightarrow B_1 \rightarrow C_1 \rightarrow E$, 且 $u_2(A_2) = B_1$ 。

(3) 从 A_3 出发也有两种决策: $A_3 \rightarrow B_1, A_3 \rightarrow B_2$ 。此时,

$$f_2(A_3) = \min \{ d_2(A_3, B_1) + f_3(B_1) \}$$

$$d_2(A_3, B_2) + f_3(B_2) \} = \min \{ \underbrace{7+11}_{\dots\dots\dots}, \underbrace{4+14}_{\dots\dots\dots} \} = 18$$

那么, A_3 到 E 有两条最短路线: $A_3 \rightarrow B_1 \rightarrow C_1 \rightarrow E$, 对应的 $u_2(A_3) = B_1$ 和 $A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow E$, 对应的 $u_2(A_3) = B_2$ 。

第四步:即 $k=1$, 该阶段只有一个状态 S , 从 S 出发有三种决策: $S \rightarrow A_1, S \rightarrow A_2, S \rightarrow A_3$ 。那么,

$$f_1(S) = \min \{ d_1(S, A_1) + f_2(A_1), d_1(S, A_2) + f_2(A_2) \}$$

$$d_1(S, A_3) + f_2(A_3) \} = \min \{ \underbrace{4+17}_{\dots\dots\dots}, \underbrace{3+19}_{\dots\dots\dots}, \underbrace{3+18}_{\dots\dots\dots} \} = 21$$

那么,从 S 到 E 共有三条最短路线: $S \rightarrow A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow E$, 此时, $u_1(S) = A_1$ 。 $S \rightarrow A_3 \rightarrow B_1 \rightarrow C_1 \rightarrow E$ 和 $S \rightarrow A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow E$, 此时, $u_1(S) = A_3$, 最短距离为 21。

7.3 函数空间迭代法和策略空间迭代法

例 1.1 中,我们介绍的最短路线问题,其阶段数是固定的,称

为定期多阶段决策问题。现在让我们来讨论这样的最短路线问题，其阶段数是不定的，在从一点到另一点的过程中，路线途经多少个其他点并无限制，其阶段数是由问题的条件和最优指标函数确定的待求未知数。

问题：设有 N 个点： $1, 2, \dots, N$ 。用 C_{ij} 表示任意两点 i, j 间的长度（可代表距离，费用等），当 i, j 间有一条从 i 到 j 的弧使 i, j 直接连结时，或 i, j 间有一条直接相连无向弧时， $0 \leq C_{ij} < +\infty$ ，否则， $C_{ij} = +\infty$ 。今设 N 为固定点，求任一点 i 到 N 的最短距离。

若 $f(i)$ 表示点 i 到 N 的最短距离，则由最优化原理，可得

$$\begin{cases} f(i) = \min_{j \neq i} \{C_{ij} + f(j)\}, & i = 1, 2, \dots, N-1 \\ f(N) = 0 & (\text{因 } C_{NN} = 0) \end{cases} \quad (7-3)$$

这是最短路线问题中的 Bellman 方程，它不是递推方程。从方程表达式可知，要求得某点 i 到 N 的最短距离，需要知道其他各点到 N 的最短距离，而这些最短距离也是待求的，所以不能简单地依赖方程用递推方法来求解。

函数空间迭代法和策略空间迭代法有效地解决了这类问题。

7.3.1 函数空间迭代法

设 $f_k(i)$ 表示第 k 次迭代中 i 点到 N 的最短距离。

函数空间迭代法求解步骤如下：

$$1^\circ \begin{cases} f_1(i) = C_{iN}, & i = 1, 2, \dots, N-1 \\ f_1(N) = 0, & i = N \end{cases} \quad (7-4)$$

$$2^\circ \begin{cases} f_k(i) = \min_{j \neq i} \{C_{ij} + f_{k-1}(j)\}, & i = 1, 2, \dots, N-1 \\ f_k(N) = 0, & i = N \end{cases} \quad (7-5)$$

$$k = 2, 3, \dots$$

3° 反复迭代 2°，直到 $f_k(i) = f_{k+1}(i) = \dots = f(i)$ 为止， $i = 1, 2, \dots, N$ 。

容易证明,当 k 相当大时, $f_k(i) = f_{k+1}(i) = \cdots = f(i)$ 成立; 当 k 增大时, $f_k(i)$ 逐渐逼近问题的最优函数 $f(i)$ 。故此法称为函数空间迭代法。

下面不加证明地给出下列结论:

(1) 函数空间迭代法确定的函数序列 $\{f_k(i)\}$ 单调下降, 且一致收敛于 $f(i)$, 其中 $f(i)$ 是(7-3)的解。

(2) 若 $0 \leq C_{ij} \leq +\infty (i, j = 1, 2, \cdots, N)$, 则收敛步数 P 有下列估计:

$$\frac{\lg(N-1)}{\lg 2} < P < \frac{\lg(N-1)}{\lg 2} + 1$$

(3) 对于无向图, 从各点到某固定点的最短距离与从此固定点到各点的最短距离一致; 但对于有向图该结论往往不成立。

另外, 当网络图中存在“长度”为负的弧, 但不存在“总长度”为负的闭回路时, $\{f_k(i)\}$ 不超过 $N-1$ 步收敛于 $f(i)$ 。

为便于计算, 下面用矩阵的方式来表示函数空间迭代法的计算步骤。

记 $C = (C_{ij})_{N \times N}$, 即由网络中任意两点间的弧长度组成的矩阵, 称为距离矩阵。

定义 3.1 算子 \otimes

任给矩阵 $A = (a_{ik})_{m \times n}$ 和 $B = (b_{kj})_{n \times l}$, 算子 \otimes 的意义是: 若 $P = (P_{ij})_{m \times l} = A \otimes B$, 则 $P_{ij} = \min_{1 \leq k \leq n} \{a_{ik} + b_{kj}\}$ 。

特别地, (1) 当 $l = 1$ 时, $B = (b_k)_{n \times 1}$, $P = (P_i)_{m \times 1}$, 分别是 n 维和 m 维列向量, 且 $P_i = \min_{1 \leq k \leq n} \{a_{ik} + b_k\}$ 。

(2) 当 $m = 1$ 时, $A = (a_k)_{1 \times n}$, $P = (P_j)_{1 \times l}$ 分别是 n 维和 l 维行向量, 且 $P_j = \min_{1 \leq k \leq n} \{a_k + b_{kj}\}$ 。那么, 舍去 $j \neq i$ 条件, (7-5) 式等价于:

$$\begin{bmatrix} f_k(1) \\ f_k(2) \\ \vdots \\ f_k(N) \end{bmatrix} = C \otimes \begin{bmatrix} f_{k-1}(1) \\ f_{k-1}(2) \\ \vdots \\ f_{k-1}(N) \end{bmatrix}, \quad k = 2, 3, \dots \quad (7-6)$$

考察(7-5)式知,舍去 $j \neq i$ 条件并不影响计算过程和结果,只要在计算结束时,在寻找任意点 i 到 N 的最短路线过程中补上这个条件即可。

【例 3.1】 在下列网络图(图 7-3)中,求:(1) 各点到 t 的最短路线;(2) t 到各点的最短路线。

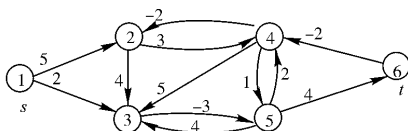


图 7-3

解 (1) 首先给出距离矩阵 C 和 $f_1(i)$ 的值, $i = 1, 2, \dots, 6$ 。然后用这些数据 and (7-6) 式中的公式求出 $f_2(i), f_3(i), \dots$, 直到 $f_k(i)$ 不再变化为止, $i = 1, 2, \dots, 6$ 。下面将计算结果列于下表。

C_{ij}	j	1	2	3	4	5	6	$f_1(i) \Rightarrow f_2(i) \Rightarrow f_3(i) \Rightarrow f_4(i) \Rightarrow f_5(i) \Rightarrow u^*(i)$					
i													
1		0	5	2	∞	∞	∞	∞	∞	3	3	3	3
2		∞	0	4	3	∞	∞	∞	∞	5	5	5	3
3		∞	∞	0	∞	-3	∞	∞	1	1	1	1	5
4		∞	-2	5	0	1	∞	∞	5	5	3	3	2
5		∞	∞	4	2	0	4	4	4	4	4	4	6
6		∞	∞	∞	-2	∞	0	0	0	0	0	0	6

其中, $u^*(i)$ 表示 i 点的最优决策,于是得到各点到 t 的最短路线和最短距离。

i 点	最短路线	最短距离
1	$\rightarrow 3 \rightarrow 5 \rightarrow 6$	3
2	$\rightarrow 3 \rightarrow 5 \rightarrow 6$	5
3	$\rightarrow 5 \rightarrow 6$	1
4	$\rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$	3
5	$\rightarrow 6$	4

(2) 由于图 7-3 是有向图, 因此, 各点到 t 的最短路线与 t 到各点的最短路线可能不同。

设有 N 个点: $1, 2, \dots, N$, 求 N 到任意一点 i 的最短路线和最短距离。与式(7-4)和式(7-5)类似, 我们有

$$1^\circ \begin{cases} f_1(i) = C_{Ni}, & i = 1, 2, \dots, N-1 \\ f_1(N) = 0, & i = N \end{cases} \quad (7-7)$$

$$2^\circ \begin{cases} f_k(i) = \min_{j \neq i} \{f_{k-1}(j) + C_{ji}\}, & i = 1, 2, \dots, N-1 \\ f_k(N) = 0, & i = N \end{cases} \quad (7-8)$$

$k = 2, 3, \dots$

其中, $f_k(i)$ 表示第 k 次迭代中 N 到点 i 的最短距离。式(7-8)舍去 $j \neq i$ 条件等价于

$$\begin{aligned} & [f_k(1), f_k(2), \dots, f_k(N)] \\ &= [f_{k-1}(1), f_{k-1}(2), \dots, f_{k-1}(N)] \otimes C \end{aligned} \quad (7-9)$$

由式(7-9), 本例中的(2)的计算结果列表如下。

$f_k(i)$ \ i	1	2	3	4	5	6	1	2	3	4	5	6	j \ C_{ij} \ i
$f_1(i)$	∞	∞	∞	-2	∞	0	0	5	2	∞	∞	∞	1
$f_2(i)$	∞	-4	3	-2	-1	0	∞	0	4	3	∞	∞	2
$f_3(i)$	∞	-4	0	-2	-1	0	∞	∞	0	∞	-3	∞	3
$f_4(i)$	∞	-4	0	-2	-3	0	∞	-2	5	0	1	∞	4
$f_5(i)$	∞	-4	0	-2	-3	0	∞	∞	4	2	0	4	5
$u^*(i)$	-	4	2	6	3	6	∞	∞	∞	-2	∞	0	6

注: “-”表示不存在

由上表中 $f_5(i)$ 和 $u^*(i), i = 1, 2, \dots, 6$, 可得

t 到各点的最短路线	点 i	最短距离
—	1	—
$6 \rightarrow 4 \rightarrow$	2	-4
$6 \rightarrow 4 \rightarrow 2 \rightarrow$	3	0
$6 \rightarrow$	4	-2
$6 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow$	5	-3

比较(1)和(2)的结果, 不难发现, 结果是不同的。

7.3.2 策略空间迭代法

策略空间迭代法的基本思路是: 首先给出初始策略 $\{u_1(i)\}$, 然后按某种迭代方式求得新策略 $\{u_2(i)\}, \{u_3(i)\}, \dots$, 直到最终求出最优策略。

步骤: 1° 选一不含回路的初始策略 $\{u_1(i), i = 1, 2, \dots, N - 1\}$ 。 $u_1(i)$ 表示在此策略下由 i 点到达的下个点, 令 $k = 1$ 。

2° 由策略 $u_k(i)$ 求指标函数 $f_k(i)$ 。即由方程组

$$\begin{cases} f_k(i) = C_{i, u_k(i)} + f_k[u_k(i)], i = 1, 2, \dots, N - 1 \\ f_k(N) = 0, \quad i = N \end{cases}$$

解出 $f_k(i)$ 。其中 $C_{i, u_k(i)}$ 已知。

3° 由指标函数值 $f_k(i)$ 求策略 $\{u_{k+1}(i)\}$, 其中 $u_{k+1}(i)$ 是 $\min_u \{C_{i, u} + f_k(u)\}$ 的解, 令 $k = k + 1$ 。

4° 按 2°、3° 反复迭代, 可逐次求得 $\{u_k(i)\}$ 和 $\{f_k(i)\}$ 。直到找到某一 k , 使 $u_k(i) = u_{k-1}(i)$ 对所有 i 成立。则 $\{u_k(i)\}$ 就是最优策略, 其相应的 $\{f_k(i)\}$ 为最优值。

不加证明, 给出下列结论:

若初始策略 $\{u_1(i)\}$ 不构成回路, 则以后迭代所得的策略 $\{u_k(i)\}$ 也不构成回路(即解是惟一的)。

由矩阵的表达方式, 步骤 3° 可以写成:

由

$$(u_{k+1}(1), \dots, u_{k+1}(N)) = C \otimes \begin{pmatrix} f_k(1) \\ f_k(2) \\ \vdots \\ f_k(N) \end{pmatrix} \quad (7-10)$$

来求 $u_{k+1}(i), i = 1, 2, \dots, N - 1$

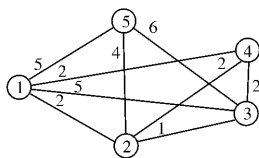
【例 3.2】 设有 1, 2, 3, 4, 5 五个城市, 相互距离如图 7-4 所示, 试用策略空间迭代法求各城到 5 城的最短路线和最短距离。

解 先选取一初始策略

$\{u_1(i)\}$:

$$u_1(1) = 5, \quad u_1(2) = 5$$

$$u_1(3) = 2, \quad u_1(4) = 1$$



然后, 反复采用由策略求指标函数和由

最优指标函数求策略两步迭代, 直至求得

$u_k(i) = u_{k-1}(i), i = 1, 2, \dots, N - 1$

时为止。计算结果列于下表中:

图 7-4

$\begin{matrix} j \\ \backslash i \\ C_{ij} \end{matrix}$	1	2	3	4	5	$f_1(i) \Rightarrow u_2(i) \Rightarrow f_2(i) \Rightarrow u_3(i)$			
1	0	2	5	2	5	5	5	5	5
2	2	0	1	2	4	4	5	4	5
3	5	1	0	2	6	5	2	5	2
4	2	2	2	0	$+\infty$	7	2	6	2
5	5	4	6	$+\infty$	0	0	6	0	6

上述表中的结果简单说明如下:

因初始策略中 1, 2 两城是直接到达 5 城的, 故应先计算。因此, 有

$$f_1(1) = C_{1, u_1(1)} + f_1[u_1(1)] = C_{15} + f_1(5) = C_{15} = 5$$

$$f_1(2) = C_{2,u_1(2)} + f_1[u_1(2)] = C_{25} + f_1(5) = C_{25} = 4$$

$$f_1(3) = C_{3,u_1(3)} + f_1[u_1(3)] = C_{32} + f_1(2) = 1 + 4 = 5$$

$$f_1(4) = C_{4,u_1(4)} + f_1[u_1(4)] = C_{41} + f_1(1) = 2 + 5 = 7$$

然后由 $C \otimes [f_1(1), f_1(2), \dots, f_1(5)]^T$ 确定出 $u_2(i), i = 1, 2, \dots, 5$, 同时得到 $[f_2(1), f_2(2), \dots, f_2(5)]^T$, 再由 $C \otimes [f_2(1), f_2(2), \dots, f_2(5)]^T$ 确定出 $u_3(i), i = 1, 2, \dots, 5$ 。由于 $u_3(i) = u_2(i), i = 1, 2, 3, 4, 5$, 所以 $u^*(i) = u_3(i), i = 1, 2, \dots, 5$ 。因此, 我们得到任意一点到 5 的最短路线和最短距离, 如表所示。

i 点	最短路线	最短距离
1	$\rightarrow 5$	5
2	$\rightarrow 5$	4
3	$\rightarrow 2 \rightarrow 5$	5
4	$\rightarrow 2 \rightarrow 5$	6

注记: (1) 我们常用逆序递推法求解多阶段决策问题, 也可以采用顺序递推法, 这种方法与逆序递推法本质上并无多大差异, 这里不再介绍。

(2) 动态规划的基本类型

根据状态变量取值是离散的还是连续的, 分为离散型和连续型。根据决策过程的演变是确定的还是随机的, 分为确定型和随机型。根据阶段变量的取值特征, 分为定期型, 即阶段变量取值个数固定; 不定期型, 即阶段变量取值个数不定, 但为有限个; 无期型, 即阶段变量取值个数无限多。等等。

(3) 动态规划的优缺点

优点, 主要有两个方面: ①易于求得全局最优解。动态规划把较复杂的问题划分成若干个相互联系阶段, 每个阶段的求解问题相对简单, 而通过逐段求解这一递推过程便可得到原问题的全局最优解。②可以得到有价值的相关信息。在最短路线的例子中, 我们已经知道, 我们在得到起点到终点最短路线的同时, 还可以得到其他各点到终点的最短路线。同样, 用动态规划的方法来求解其他问题, 也会产生类似的效果, 这些相关信息对进一步分析结果无

疑是有益的。

缺点,主要有两点:① 没有统一的标准模型可供采用。不同的实际问题,其动态规划模型也随之不同,因而,实际问题的动态规划模型的建立往往需要丰富的想像力和灵活的技巧性。② 存在所谓的“维数障碍”。即当问题的变量个数(维数)太大时,受计算机存储器容量和计算速度的限制,常常无法解决。

7.4 应用举例

1. 投资决策问题

现在求解例 1.3。

解 (1) 阶段数 $n = 3$, 阶段变量用 k 表示。

(2) 状态变量 x_k 表示第 k 年开始时手中钱的总额(万元)。

(3) 决策变量 $u_k(x_k)$ 表示第 k 年开始手中有 x_k 万元时的投资选择:

$$u_k(x_k) = \begin{cases} 0 (\text{不投资}), & x_k < 1000 \\ A, & x_k \geq 1000 \\ B, & x_k \geq 1000 \end{cases}$$

(4) $V_{kn}(x_k, u_k)$ 表示在第 k 年开始时状态为 x_k , 做出投资决策 u_k 后, 第 n 年末的累积期望利润。而 $f_k(x_k) = \max_{u_k \in \{0, A, B\}} V_{kn}(x_k, u_k)$ 。

(5) 第 k 年的期望利润

$$V_k(x_k, u_k) = \begin{cases} 0, & u_k(x_k) = 0 \text{ 时} \\ 0 \times 0.4 + 2000 \times 0.6 - 1000 = 200, & u_k(x_k) = A \text{ 时} \\ 1000 \times 0.9 + 2000 \times 0.1 - 1000 = 100, & u_k(x_k) = B \text{ 时} \end{cases}$$

(6) 函数基本方程

对于 $x_k \geq 1000$, 有

$$V_{kn}(x_k, u_k) = \begin{cases} f_{k+1}(x_k), & u_k(x_k) = 0 \text{ 时} \\ 200 + 0.4f_{k+1}(x_k - 1000) + 0.6f_{k+1}(x_k + 1000), & u_k(x_k) = A \text{ 时} \\ 100 + 0.9f_{k+1}(x_k) + 0.1f_{k+1}(x_k + 1000), & u_k(x_k) = B \text{ 时} \end{cases}$$

对于 $x_k < 1000$, 有 $V_{kn}(x_k, u_k) = 0$

$$f_k(x_k) = \max_{u_k \in \{0, A, B\}} V_{kn}(x_k, u_k)$$

计算如下:

$k = 3$ 时

x_3	$f_3(x_3)$	u_3^*
$0 \leq x_3 < 1000$	0	0
$1000 \leq x_3 < 3000$	200	A

$k = 2$ 时

$x_2 \backslash u_2$	$V_{23}(x_2, u_2)$			$f_2(x_2)$	u_2^*
	0	A	B		
$0 \leq x_2 < 1000$	0	—	—	0	0
$1000 \leq x_2 < 2000$	200	320	300	320	A
$2000 = x_2$	200	400	300	400	A

$k = 1$ 时

$x_1 \backslash u_1$	$V_{13}(x_1, u_1)$			$f_1(x_1)$	u_1^*
	0	A	B		
$x_1 = 1000$	320	440	428	440	A

因此,最优方案是每年均投资于 A 项目,三年末的最大利润为 440 万元。

2. 多阶段生产安排问题

多阶段生产安排问题也称为多阶段配置问题。

问题:设有某种原料,其数量为 \bar{x} (吨),用于生产两种不同类型的产品,记为类型 A 与类型 B。已知该原料投入生产后,还可以回收部分原料用于下一阶段的再生产。假设

$g(x)$ = 投入数量为 x 的原料, 生产 A 型产品的收益值
并且, $g(0) = 0$;

$h(x)$ = 投入数量为 x 的原料, 生产 B 型产品的收益值
并且, $h(0) = 0$ 。

a = 生产 A 型产品原料的回收率 ($0 \leq a < 1$);

b = 生产 B 型产品原料的回收率 ($0 \leq b < 1$)。

若对总数量为 \bar{x} 的原料进行 n 个阶段的生产, 问: 对每个阶段应如何分配原料用于生产 A 型及 B 型产品, 使得经过 n 个阶段的生产之后总的收益最大?

下面建立该问题的动态规划模型:

(1) 由问题本身, 分为 n 个阶段, 阶段变量为 k 。

(2) x_k 为第 k 阶段的状态变量, 表示第 k 阶段生产开始时, 原料数量为 x_k (吨)。

(3) u_k 为决策变量, 表示从第 k 阶段原料的数量 x_k 中分配给 A 型产品的数量 (吨), 那么, 允许决策集合为

$$U_k(x_k) = \{u_k | 0 \leq u_k \leq x_k\}$$

(4) 状态转移方程

$$x_{k+1} = a \cdot u_k + b(x_k - u_k) = bx_k + (a - b)u_k$$

$$k = 1, 2, \dots, n$$

(5) $f_k(x_k)$ 表示在状态为 x_k (吨) 时, 从第 k 阶段生产到最后阶段的最大收益。则函数基本方程为

$$\begin{cases} f_k(x_k) = \max_{0 \leq u_k \leq x_k} \{g(u_k) + h(x_k - u_k) + f_{k+1}(bx_k + (a - b)u_k)\} \\ f_n(x_n) = \max_{0 \leq u_n \leq x_n} \{g(u_n) + h(x_n - u_n)\} \end{cases} \quad (7-11)$$

$$k = n-1, n-2, \dots, 1$$

【例 4.1】 设 $n = 3, \bar{x} = 100$ (吨), A 型产品与 B 型产品的收益函数及原料回收率分别为

A 型产品: $g(x) = 0.6x, \quad a = 0.2$;

B 型产品: $h(x) = 0.4x$, $b = 0.5$ 。

此时, 状态转移方程为: $x_{k+1} = 0.5x_k - 0.3u_k$ 。

$k = 3$ 时

$$\begin{aligned} f_3(x_3) &= \max_{0 \leq u_3 \leq x_3} \{0.6u_3 + 0.4(x_3 - u_3)\} \\ &= \max_{0 \leq u_3 \leq x_3} \{0.4x_3 + 0.2u_3\} \\ &= 0.6x_3 \end{aligned}$$

此时, $u_3^* = x_3$, 即第 3 生产阶段, 原料分配给 A 型产品。

$k = 2$ 时

$$\begin{aligned} f_2(x_2) &= \max_{0 \leq u_2 \leq x_2} \{0.6u_2 + 0.4(x_2 - u_2) + f_3(x_3)\} \\ &= \max_{0 \leq u_2 \leq x_2} \{0.2u_2 + 0.4x_2 + 0.6(0.5x_2 - 0.3u_2)\} \\ &= \max_{0 \leq u_2 \leq x_2} \{0.02u_2 + 0.7x_2\} \\ &= 0.72x_2 \end{aligned}$$

此时, $u_2^* = x_2$, 即第 2 生产阶段, 原料分配给 A 型产品。

$k = 1$ 时

$$\begin{aligned} f_1(x_1) &= \max_{0 \leq u_1 \leq x_1} \{0.6u_1 + 0.4(x_1 - u_1) + f_2(x_2)\} \\ &= \max_{0 \leq u_1 \leq x_1} \{0.2u_1 + 0.4x_1 + 0.72(0.5x_1 - 0.3u_1)\} \\ &= \max_{0 \leq u_1 \leq x_1} \{-0.016u_1 + 0.76x_1\} \\ &= 0.76x_1 \end{aligned}$$

此时, $u_1^* = 0$, 即第 1 生产阶段, 原料分配给 B 型产品。

最终可得: $f_1(\bar{x}) = f_1(100) = 0.76 \times 100 = 76$, 即按上述方式生产, 经过 3 个阶段生产之后, 总的收益最大, 为 76。

3. 资源分配问题

所谓“资源分配问题”就是把一定数量的若干资源合理地分配给若干个使用者, 使指标函数达到最优。

这里仅讨论一种资源的分配问题。其一般提法是, 设某种资源

的总量为 a , 拟用于 n 项经营活动, 若给第 j 项活动分配 x_j 个单位, 其收益为 $g_j(x_j)$, 问应如何分配, 才能使这 n 项经营活动总的收益值最大?

这个问题可表示为

$$\begin{cases} \max & z = \sum_{i=1}^n g_i(x_i) \\ \text{s. t.} & x_1 + x_2 + \cdots + x_n = a \\ & x_i \geq 0, \quad i = 1, 2, \dots, n \end{cases}$$

因此, 可用非线性规划方法求解。但是利用这类问题的特性, 也可以把它看做一个多阶段决策问题, 建立如下的动态规划模型:

以阶段变量 k 表示资源分配给第 k 项经营活动的过程;

以状态变量 x_k 表示在开始给第 k 项经营活动分配资源时尚剩余的资源数量;

以决策变量 u_k 表示分配给第 k 项经营活动的资源数量, 则允许决策集合为 $U_k(x_k) = \{u_k | 0 \leq u_k \leq x_k\}$, 状态转移方程为 $x_{k+1} = x_k - u_k$;

以 $V_k(x_k, u_k)$ 表示从现有 x_k 个单位资源中分配给第 k 项经营活动 u_k 个单位资源后的预计收益。

以 $f_k(x_k)$ 表示从现有 x_k 个单位资源分配给第 k 项到第 n 项经营活动后, 所得的最大收益, 则函数基本方程为

$$\begin{cases} f_k(x_k) = \max_{u_k \in U_k(x_k)} \{V_k(x_k, u_k) + f_{k+1}(x_{k+1})\} \\ f_{n+1}(x_{n+1}) = 0 \\ k = n, n-1, \dots, 1 \end{cases}$$

由上式可求出资源的最优分配策略及其最大收益值。

现在求解例 1.2

解 (1) 以 $k=1, 2, 3$ 表示给三个门市部分配的过程;

(2) 状态变量 x_k 表示在开始给第 k 个门市部分配时尚未分

配的箱数；

(3) 决策变量 u_k 表示分配给第 k 门市部的箱数；

(4) 以 $V_k(x_k, u_k)$ 表示从现有 x_k 箱中分配给第 k 门市部 u_k 箱后的预计收益。以 $f_k(x_k)$ 表示从现有 x_k 箱中分配给第 k 门市部到第 3 门市部后所得的最大收益,则状态转移方程为 $x_{k+1} = x_k - u_k$,函数基本方程为

$$\begin{cases} f_k(x_k) = \max_{u_k \in U_k(x_k)} \{V_k(x_k, u_k) + f_{k+1}(x_{k+1})\} \\ f_4(x_4) = 0 \end{cases} \quad k = 3, 2, 1$$

计算如下：

$k = 3$ 时

$x_3 \backslash u_3$	$V_3(x_3, u_3) + 0$					$f_3(x_3)$	u_3^*
	0	1	2	3	4		
0	0					0	0
1	0	5				5	1
2	0	5	7			7	2
3	0	5	7	9		9	3
4	0	5	7	9	11	11	4

$k = 2$ 时, $x_3 = x_2 - u_2$

$x_2 \backslash u_2$	$V_2(x_2, u_2) + f_3(x_3)$					$f_2(x_2)$	u_2^*
	0	1	2	3	4		
0	0+0=0					0	0
1	0+5=5 2+0=2					5	0
2	0+7=7 2+5=7 4+0=4					7	0, 1
3	0+9=9 2+7=9 4+5=9 8+0=8					9	0, 1, 2
4	0+11=11 2+9=11 4+7=11 8+5=13 9+0=9					13	3

$k = 1$ 时, $x_2 = x_1 - u_1 = 4 - u_1$

$\begin{matrix} u_1 \\ x_1 \end{matrix}$	$V_1(x_1, u_1) + f_2(x_2)$					$f_1(x_1)$	u_1^*
	0	1	2	3	4		
4	0+13=13	4+9=13	7+7=14	8+5=13	10+0=10	14	2

按 $k = 1, 2, 3$ 的顺序查表, 方法如下:

$$u_1^* = 2 \frac{\text{查 } k=2 \text{ 的表}}{x_2 = x_1 - u_1^* = 2} \begin{cases} u_2^* = 0 \frac{\text{查 } k=1 \text{ 的表}}{x_3 = x_2 - u_2^* = 2} & u_3^* = 2 \\ u_2^* = 1 \frac{\text{查 } k=1 \text{ 的表}}{x_3 = x_2 - u_2^* = 1} & u_3^* = 1 \end{cases}$$

得到最优分配方案为: 分别给 1, 2, 3 门市部分配 2 箱, 0 箱, 2 箱或 2 箱, 1 箱, 1 箱。最大收益值为 14。

4. 设备更新问题

【例 4.2】 某厂拟于明年初购置某种设备一台, 以后每年初都要决定是继续使用还是更新。若继续使用旧设备, 则需要支付一定的维修费用, 若更新设备, 就要支付一定的购置费用。已知今后五年内每年初该设备的购置费用及使用不同年数的设备在各年内的维修费如下表所示, 问该厂在今后五年内应如何更新设备, 使总费用最少?

年度 i	1	2	3	4	5
年初购置费 P_i	10	13	17	21	26
第 i 年维修费 C_i	6	8	11	15	19

解 设 $V_i (i = 1, 2, 3, 4, 5)$ 表示“第 i 年初购进一台新设备”这种状态, 以 V_6 表示“第 5 年末”这一状态, 则该问题可用网络图 7-5 表示, 其中, 弧 (V_i, V_j) 上的数字表示第 i 年初购置的一台设备一直使用到第 j 年初的费用之和。那么, 原问题归结为找出一条从 V_1 到 V_6 的最短路线问题。

设 $f_k(i)$ 表示第 k 步迭代中 V_1 到 V_i 的最短距离。

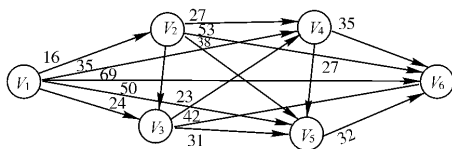


图 7-5

用函数空间迭代法求解, 计算结果列表如下:

$i \backslash j$	1	2	3	4	5	6	1	2	3	4	5	6	$j \backslash i$
$f_k(i)$													C_{ij}
$f_1(i)$	0	16	24	35	50	69	0	16	24	35	50	69	1
$f_2(i)$	0	16	24	35	50	66	∞	0	19	27	38	53	2
							∞	∞	0	23	31	42	3
$f_3(i)$	0	16	24	35	50	66	∞	∞	∞	0	27	35	4
							∞	∞	∞	∞	0	32	5
$u^*(i)$	1	1	1	1	1	3	∞	∞	∞	∞	∞	0	6

所以, 由 V_1 到 V_6 的最短路线是 $V_1 \rightarrow V_3 \rightarrow V_6$, 即明年初购置的设备使用到第 3 年初予以更新, 此后不再更新设备。最小总费用为 66。

习 题 七

1. 石油输送管道铺设最优方案的选择问题: 考察如下网络图, 设 A 为出发地, E 为目的地, B, C, D 分别为三个必须建立油泵加压站的地区, 其中的 $B_1, B_2, B_3; C_1, C_2, C_3; D_1, D_2$ 分别为可供选择的各站站位。图 7-6 中的线段表示管道可铺设的位置, 线段旁的数字表示铺设这些管线所需的费用。问如何铺设管道才能使总费用最小?

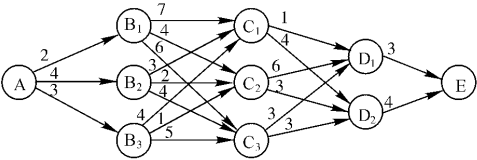


图 7-6

2. 某厂估计某一种新产品未来四年内每年在不同价格下的期望利润(万元)如下表所示。如果相邻两年间价格调整幅度分别为:(1)不超过 2 元,(2)不超过 4 元,(3)无任何限制,试就这三种情况分别确定各年最优价格。

利润 年 价格(元)				
	1	2	3	4
16	3	9	3	7
18	2	1	2	2
20	7	4	8	1
22	9	2	6	4
24	5	5	3	1

3. 设某县从事区农村扫盲工作的人员有 6 名,当派往各区工作人员人数不同时,扫盲人数所增加的数目也不同,它们的关系如下表所示。试求应派往各区多少工作人员,才能使扫盲的人数所增加的数目最大?

工作人员 人数	区名			
	1	2	3	4
0	0	0	0	0
1	20	25	18	28
2	42	45	39	47
3	60	57	61	65
4	75	65	78	74
5	85	70	90	80
6	90	73	95	85

4. 某工厂有三种产品准备参加展销,各产品重量与利润关系如下表所示。展销品运输车运送总重量不能超过 6 吨,问怎样安排参展,才能使总利润

最大?

产品种类	1	2	3
重量(吨)	2	3	4
利润(元)	80	130	180

5. 某科研项目由三个小组用不同手段去研究,它们失败的概率各为 0.4,0.6 和 0.8。为了减少三个小组都失败的可能性,现决定给三个小组中增派两名高级科学家,到各小组后,各小组科研项目失败概率如下表所示;问如何分派科学家才能使三个小组都失败概率(即科研项目最终失败的概率)最小?

高 级 科学家数	小 组		
	1	2	3
0	0.40	0.60	0.80
1	0.20	0.40	0.50
2	0.15	0.20	0.30

* 6. 假定有一台机器,它处于运行或损坏两种状态之一。如果运行一周,可得总利润 100 美元。如果一周中发生故障,总利润为零。如果在一周开始处于运行状态,并采取预防性维护措施,则一周内出现故障的概率是 0.4,如不进行维护,故障的概率是 0.7。然而,维护要花去 20 美元。当在一周开始时机器损坏,或以 40 美元的费用去修理,在此情况下出现故障的概率是 0.4;或用新机器取代,费用为 150 美元,它保证机器在第一周正常运行。假定第一周开始时为新机器,寻求一个最优的修理,取代和维护策略,使四周的总利润最大。

7. 某牧场有 15 千元资金,它可以用来买 A、B 两种牛。每花 1 千元买 A、B 牛后,当年及以后每年各可出生 2、3 头小牛,且当年各获利 500 元、200 元;以后每年用按此比例的资金继续买牛。问今后四年应如何买牛才能使第四年末的牛群最大?

8. 求“多阶段生产安排问题”的最优策略,其中,原料数量为 $\bar{x} = 500$,阶段数 $n = 4$,并且生产 A 型产品:收益函数 $g(x) = 0.8x$,回收率 $a = 0.3$;生产 B 型产品:收益函数 $h(x) = 0.6x$,回收率 $b = 0.6$ 。

9. 用动态规划解下列问题

$$\begin{cases} \max & z = 8x_1 + 7x_2 \\ \text{s. t.} & 2x_1 + x_2 \leq 8 \\ & 5x_1 + 2x_2 \leq 15 \\ & x_1, x_2 \text{ 为非负整数} \end{cases}$$

10. 用动态规划方法解下列非线性规划问题

$$\begin{cases} \max & z = 7x_1^2 + 6x_1 + 5x_2^2 \\ \text{s. t.} & x_1 + 2x_2 \leq 10 \\ & x_1 - 3x_2 \leq 9 \\ & x_1, x_2 \geq 0 \end{cases}$$

11. 某工厂拟将 5 台先进设备分配给下属的甲、乙、丙三个分厂,分厂获得该设备后每年为总厂提供的盈利如表所示,问如何分配这些设备,才能使总厂的盈利最大?

盈利(万元) \ 设备(台) 分厂	0	1	2	3	4	5
甲	0	3	7	9	12	13
乙	0	5	10	11	11	11
丙	0	4	6	11	12	12

12. 某厂生产一种机械设备,据市场调查,今后四个时期该产品需求量依次为 2,3,2,4 台,该厂每期最大生产能力为 6 台,每期生产固定费用 3 万元(若不生产则为 0),单台成本 1 万元,每期贮存保养费为每台 0.5 万元,若第 1 期初和第 4 期末无库存,试确定各期产量,使总费用最少。

第 8 章 进化算法

近十多年来,一类基于生物界的自然选择和自然遗传机制的优化计算方法——进化算法引起了人们的广泛关注,获得了迅速的发展和广泛的应用。它主要包括进化策略、进化规划和遗传算法这三类方法,本章将对这三类优化方法和模拟退火方法做简要的介绍。

8.1 引言 进化算法概述

进化算法是最近十多年来备受重视的一类优化算法,它与传统的以梯度为基础的优化算法相比,不需要使用梯度,计算过程对函数性态的依赖性较小,具有适应范围广、鲁棒性好、适于并行计算等优点。传统的优化算法虽具有较高的计算效率、较强的可靠性、比较成熟等优点,是一类最重要的、应用最广泛的优化算法,但由于其机理是建立在局部下降的基础上的,因而大多不太适合于全局性优化问题的求解。其次,对于不可微和高度病态的优化问题,以梯度为基础的优化算法往往也无能为力。因此,进化算法具有广阔的应用前景。

8.1.1 进化算法的起源和发展

遗传算法(GA, Genetic Algorithms)是最早广为人知的一类进化算法。20 世纪的 60 年代,美国 Michigan 大学的 John Holland 教授在研究自然和人工系统自适应行为的过程中,认识到生物的自然遗传现象与人工自适应系统行为的相似性,并提出在研究和

设计人工自适应系统时,可以借鉴生物自然遗传的基本原理,模仿生物自然遗传的基本方法。1967年,他的学生 J. D. Bagley 在博士论文中首次提出“遗传算法”一词。1969年, Holland 提出了一种借鉴生物进化机制的自适应机器学习方法,70年代初,提出了奠定遗传算法研究理论基础的基本定理——模式定理 (Schema Theorem), 1975年,出版了名著《自然系统和人工系统的自适应性》,这是第一本系统论述遗传算法的专著,它标志着遗传算法的正式诞生^[66,68]。1975年, De Jong 首先将遗传算法用于函数优化。80年代,遗传算法被广泛应用于各种复杂系统的控制和优化问题中。1985年,在美国召开了第一届遗传算法国际会议,并成立了国际遗传算法学会 (ISGA, International Society of Genetic Algorithms)。学会以后每两年举行一次。1989年, Holland 的学生 D. E. Goldberg 出版了专著《搜索、优化和机器学习中的遗传算法》,系统地总结了遗传算法研究的主要成果,标志着遗传算法已从古典阶段发展到了现代阶段。

进化规划 (EP, Evolutionary Programming) 是由美国学者 L. J. Fogel 等在 1966 年为求解预测问题首先提出来的。进化规划的基本思想是基于生物界的自然遗传和自然选择的生物进化原则,利用多点迭代算法来代替普通的单点迭代算法,并根据被正确预测的符号数来度量适应值,通过变异,父辈群体中的每个个体产生一个子代,父辈和子代中最好的那一半被选择生存下来。1992年, D. B. Fogel 基于正态分布变异,将进化规划扩展到可求解实值问题。

20 世纪 90 年代以前,由于学术界对在人工智能领域采用进化规划持有怀疑,因此,进化规划的思想和方法并未引起人们的足够重视。直至 90 年代初,才逐步引起学术界的重视。1992年,在美国举行了进化规划的第一届年会,以后每年举行一次,使进化规划得到了迅速的发展和推广、应用。1994年, D. B. Fogel 给出了进化

规划算法的一个收敛性证明,他把进化规划看做有限维的马尔可夫链,把种群中所有向量的集合对应于一种状态。

进化策略(ES, Evolution Strategies)的思想与进化规划有很多相似之处,但它是在欧洲独立于遗传算法和进化规划而发展起来的。1963年,德国柏林工业大学的两名学生 I. Rechenberg 和 H. P. Schwefel 利用流体工程研究所的风洞进行实验,以便确定气流中物体的最佳外形。由于当时的优化方法不适于解决这类问题,Rechenberg 提出按照生物进化思想,对物体的外形参数进行随机变化并试验其效果,进化策略的思想便由此诞生。当时,人们对这种随机策略无法接受,但他们仍坚持实验。1970年,Rechenberg 完成了关于进化策略研究的博士论文;1974年,Schwefel 把有关进化策略研究的成果做了系统的归纳整理。1990年,在欧洲召开了第一届“基于自然思想的并行问题求解”(PPSN, Parallel Problem Solving from Nature)国际会议,以后该会每两年举行一次,成为在欧洲召开的有关进化算法的主要国际会议。

1999年,我们对一类进化策略的收敛性进行了研究,对于一类水平集有界的连续函数,证明了该算法依概率收敛于问题的全局极小点^[69]。

由于遗传算法、进化规划与进化策略是不同研究领域的学者分别独立提出来的,在较长的一段时间内,相互之间没有正式沟通。直到1990年,从事遗传算法研究的学者才开始与从事进化规划和进化策略研究的学者有所交流;1992年,进化规划和进化策略这两个领域的研究人员首次接触到对方的工作,通过交流,发现彼此在研究中所依据的基本思想具有惊人的相似之处,都是基于生物界的自然遗传和自然选择等生物进化思想。因此,人们将这类方法统称为进化计算(EC, Evolutionary Computation),而将相应的算法统称为进化算法(EA, Evolutionary Algorithms)。

1993年,这一领域的第一份国际性杂志《进化计算》在美国正

式出版。1994 年,IEEE 神经网络委员会主持召开了第一届进化计算国际会议,以后每年举行一次。1996 年 5 月 20 日至 22 日在日本名古屋召开了第三届进化计算年会,有 100 多位学者参加,发表论文 154 篇。

8.1.2 进化算法的一般框架

进化算法也是一种迭代算法,即从给定的初始解出发,通过逐步迭代,不断改进,直到获得满意的近似最优解为止。在进化计算中,每一次迭代被视为一代生物个体的繁殖,因此称为“代”。但是,进化算法与普通的搜索算法(例如梯度型算法)有所不同:①普通的搜索算法一般是从一个解出发改进到另一个较好的解;而进化算法一般是从一组解出发改进到另一组较好的解(称为解群,而每一个解称为一个“个体”)。②在普通的搜索算法中,解的表示可采用任意的形式,不需进行特殊的处理;但在进化算法中,每一个解被视为一个生物个体,因此,一般要求用一条染色体来表示,即用一组有序排列的基因来表示。这就要求对每一个解进行编码。③普通的搜索算法一般都采用确定性的搜索策略,而进化计算在搜索过程中一般采用随机性的搜索策略。

下面对基因和染色体作简要说明。

基因是英文 gene 的音译。基因是决定生物和人类生、老、病、死和一切生命现象的物质基础,20 世纪上半叶才阐明了其化学本质。基因是由数以千计的碱基对组成的信息功能单位,它按顺序排列在细胞核的染色体上。据估计,人体内约有 3 万个基因。遗传学家认为:基因是染色体上的一个功能单位,它是脱氧核糖核酸,即 DNA 的一个片段。

一切有机体均由细胞构成(病毒除外),单细胞生物仅由一个细胞构成,多细胞生物体一般由数以万计的细胞组成。在分裂间期的人类细胞核内,可以看到被染成深紫色的短棒状小体,这就是由

蛋白质和 DNA 分子构成的染色体。

在自然界中,物种的性质是由染色体决定的,而染色体则是由蛋白质和脱氧核糖核酸组成的。

下面给出进化算法的一般步骤。

(1) 选定一组初始解;

(2) 对当前这组解的性能进行评价;

(3) 根据(2)中的评价结果,从当前这组解中选择一定数量的解作为迭代后的解的基础;

(4) 对(3)中所得到的解进行操作(如重组和变异),产生迭代后的解群;

(5) 检验(4)中得到的解群是否满足终止进化条件,若满足,则终止;否则,将这组解作为当前解,返回(2)。

8.2 进化策略

考虑如下的优化问题:

$$\min f(x), x \in \mathbf{R}^n \quad (2-1)$$

求解优化问题(2-1)的进化策略本质上是一类建立在模拟生物进化过程基础上的随机搜索方法。其基本思想可概括为:

(1) 将优化问题的目标函数 $f(x)$ 理解为(或转换到)某种生物种群对环境的适应性。

(2) 将决策变量 $x = (x_1, x_2, \dots, x_n)^T$ 对应为生物种群的一个个体,这样的一些个体的集合构成一个种群,种群中的个体的数目称为该种群的规模。

(3) 将求解优化问题的算法与生物种群的进化过程类比。每一次迭代被看成是一代生物个体的繁殖,称之为“代”。

8.2.1 标准的进化策略 —— $(\mu + \lambda) - ES$

早期的进化策略(简记为 $(1 + 1) - ES$)是由德国学者 Rechenberg 于 1973 年提出来的,采用的是一种简单的变异选择机制,每一代通过 Gauss 变异作用在一个个体上来产生一个子代。后来,他又提出了 $(\mu + 1) - ES$,这个策略将 μ 个($\mu > 1$)个体重组形成的一个子代,替换掉变异最差的父辈个体。1981 年, Schwefel 在上述工作的基础上提出了两种通用的进化策略: $(\mu + \lambda) - ES$ 和 $(\mu, \lambda) - ES$ 。这里 μ 表示当前种群的规模, λ 表示由当前种群通过杂交和变异而产生的中间种群的规模。 $(\mu + \lambda) - ES$ 是从两个种群的并集中选择 μ 个最好的个体作为下一代种群,而 $(\mu, \lambda) - ES$ 是从包含 λ 个个体的中间种群中选择 μ ($1 \leq \mu \leq \lambda$) 个最好的个体作为下一代种群。

下面给出 $(\mu + \lambda) - ES$ (简称算法 1)的具体计算步骤。

第一步 初始化: 在 R^n 中随机产生 μ 个个体 $x_1(0), x_2(0), \dots, x_\mu(0)$, 由它们构成初始种群 $X(0) = \{x_1(0), x_2(0), \dots, x_\mu(0)\}$ 。选取适应值函数 $E = f(x)$, 令 $x^*(0) = \arg \min_{1 \leq j \leq \mu} \{f(x_j(0))\}$, $k = 0$;

第二步 产生中间种群。取 $\lambda \geq \mu$, 令 $i = 1$ 。

2.1 以等概率从 $X(k)$ 中选取两个个体 $x_{i_1}(k), x_{i_2}(k)$;

2.2 将杂交算子作用于 $x_{i_1}(k), x_{i_2}(k)$, 以产生中间个体 $x'_{\mu+i}(k)$ (例如, 令 $x'_{\mu+1}(k) = [x_{i_1}(k) + x_{i_2}(k)]/2$);

2.3 进行变异: 令 $x_{\mu+i}(k) = x'_{\mu+i}(k) + \xi$, 其中 ξ 为随机向量, 其分量服从均值为 0, 均方差为 σ_i 的正态分布(又叫 Gauss 分布), 且各个分量之间相互独立;

2.4 若 $i = \lambda$, 转第三步; 否则, 令 $i = i + 1$, 转 2.1。

第三步 选择。从 $\{x_1(k), x_2(k), \dots, x_{\mu+\lambda}(k)\}$ 中选取 μ 个函数值最小的个体组成新一代种群 $X(k+1) = \{x_1(k+1), \dots,$

$x_\mu(k+1)\}$, 令 $x^*(k+1) = \arg \min_{1 \leq i \leq \mu} \{f(x_i(k+1))\}$ 。

第四步 终止检验: 判别 $X(k+1)$ 是否满足终止进化条件, 若满足, 则计算结束, 输出最优解 $x^*(k+1)$; 否则, 令 $k = k+1$, 返回第二步。

对于另一种进化策略 $(\mu, \lambda) - ES$, 只需修改上述算法中的第三步即可。

8.2.2 基于一致分布的进化策略

上述的进化策略是以正态分布 (Gauss 分布) 为基础的, 即变异过程是对杂交产生的 λ 个个体实施 Gauss 随机摄动实现的, 这样做可使新一代种群能够兼顾目标函数 (适应性函数) 局部下降和整体寻优两个方面。但是, 当目标函数 $f(x)$ 是单峰函数时, 问题 (2-1) 的局部最优解也是它的全局最优解, 对于这类问题应充分利用这一特点, 来构造更有效的算法。构造适合于求解这类问题的算法需考虑以下几点:

(1) 种群无须在大范围内分布。这意味着变异算子可取为在某区间上服从一致分布的随机向量。把变异算子改用在适当区间上服从一致分布的随机向量, 将大大减少随机数的调用次数, 而产生的新种群的质量不会有显著的差异, 数值试验证实了这一分析。

(2) 变异母体的选取。在这种情况下, 由于无须过多考虑种群中个体的多样性, 采用每代种群中的最优个体作为母体进行变异是合适的。假设第 k 次进化得到的最优点 (最优个体) 为 $x^*(k)$, 第 $k+1$ 次进化的种群只通过对 $x^*(k)$ 进行随机摄动产生, 将有更多机会使新生的个体落在水平集 $L(x^*(k)) = \{x | f(x) \leq f(x^*(k))\}$ 中, 从而较快地得到新的下降点, 同时, 也省去了算法一中排序花费的时间。

(3) 步长的修正策略。Gauss 分布中的均方差 σ , 在进化策略中起类似于一般优化算法中的搜索步长的作用, 当变异算子采用一

致分布的随机向量时,步长即相当于随机变量分布区间的长度。随着迭代的进行,步长也将随之减小,这时不能简单地根据相邻两代的最优点的距离决定步长的大小。可采用如下的步长修正策略:如果若干代内找不到下降点,说明步长太大,应适当缩小,以便较快地找到下降点。

下面给出求解单峰函数优化问题(2-1)的进化算法——基于一致分布的进化策略: $(\mu + \lambda) - ES$ (简称为算法 2) 的计算步骤。

选定初始点 x_0 , 初始步长 $r_0 > 0$, 计算精度 $\varepsilon > 0$, 步长压缩系数 $\alpha \in (0, 1)$, 内循环次数 T , 变异种群规模 m_1 , 杂交种群规模 m_2 。令 $r = r_0, i = 0$ 。

(1) 令 $k = 1$ 。

(2) 变异: $x_i^{(j)} = x_i + ru, j = 1, 2, \dots, m_1$, 其中 u 是一个随机向量, 其分量在 $\{-0.5, 0.5\}$ 上服从一致分布。

(3) 杂交: 用变异种群 $\{x_i^{(1)}, \dots, x_i^{(m_1)}\}$ 杂交产生中间种群 $\{x_i^{(m_1+1)}, \dots, x_i^{(m_1+m_2)}\}$ 。

(4) 寻优: 求变异种群和杂交种群的最优个体, 令 $x_{i+1}^* = \arg \min_{1 \leq j \leq m_1+m_2} \{f(x_i^{(j)})\}$ 。

(5) 比较。若 $f(x_{i+1}^*) < f(x_i^*)$, 则令 $i = i + 1, r = r_0$, 返回(1); 否则, 转(6)。

(6) 若 $k \leq T$, 则令 $k = k + 1$, 返回(2); 否则, 转(7)。

(7) 若 $r < \varepsilon$, 则计算结束, 输出最优解 x_i^* ; 否则, 令 $r = \alpha r$, 返回(1)。

8.2.3 求解约束优化问题的进化策略

考虑一般的约束优化问题:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & c_i(x) = 0, i \in E = \{1, 2, \dots, m\} \\ & c_i(x) \geq 0, i \in I = \{m+1, \dots, p\} \end{cases} \quad (2-2)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 。用 Ω 表示问题(2-2)的可行集。

引入 L_1 不可微精确罚函数:

$$P(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i \in E} |c_i(\mathbf{x})| - r \sum_{i \in I} \min(0, c_i(\mathbf{x})) \quad (r > \bar{r} > 0)$$

其中 $r > 0$ 为足够大的参数。则求解问题(2-2)可转化为求解如下的无约束优化问题^[36]:

$$\min P(\mathbf{x}, r) \quad (2-3)$$

因此,应用前述的两种进化策略均可通过求解问题(2-3)来获取问题(2-2)的最优解。彭宏等在文献 70, 71 中给出一种求解如下的约束优化问题(2-4)的进化策略实施方案,下面做一简介。

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & c_i(\mathbf{x}) = 0, i = 1, 2, \dots, m_e \\ & L_i \leq c_i(\mathbf{x}) \leq H_i, i = m_e + 1, \dots, m \end{cases} \quad (2-4)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T, L_i, H_i \in R, \Omega$ 为(2-4)的可行集。

定义 L_1 不可微精确罚函数为

$$P(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i=1}^m |g_i(\mathbf{x})| \quad (2-5)$$

其中

$$g_i(\mathbf{x}) = \begin{cases} 0, \mathbf{x} \in \Omega \\ c_i(\mathbf{x}), \mathbf{x} \notin \Omega, i = 1, 2, \dots, m_e \\ \max(L_i - c_i(\mathbf{x}), c_i(\mathbf{x}) - H_i), \mathbf{x} \notin \Omega, i = m_e + 1, \dots, m \end{cases} \quad (2-6)$$

$r > 0$ 为罚因子。则求解(2-4)等价求解问题

$$\min P(\mathbf{x}, r) \quad (2-3)$$

这里的 $P(\mathbf{x}, r)$ 由(2-5)和(2-6)式定义。

在进化计算过程中,方差 σ 的选取是重要的,它直接影响进化策略的收敛速度。文献 70, 71 给出的方差调整方案如下:

(1) 在第 k 代中, 适应性最好的个体记为 $x_{0,k}$ 。

(2) 若从第 k 代到第 $k+m$ 代, $x_{0,k+m}$ 的第 i 个分量没有改进, 则令其相应的方差 σ_i 适当减小, 即令 $\sigma_i = \delta \sigma_i, \delta \in [0.7, 0.9]$ 。

(3) 若某一次进化后, x_0 的第 i 个分量改变了 n_i , 则这一改变可能最优, 期望下一次进化能以概率 p 保持这种速度。令 $\sigma_i = \varphi \eta_i$, 其中 φ 满足

$$\frac{1}{\sqrt{2\pi}\varphi\eta_i} \int_{-\eta_i}^{\eta_i} \exp\left(-\frac{x^2}{2(\varphi\eta_i)^2}\right) dx = 1 - p, p \in [0.4, 0.5]$$

在进化过程中, 引入杂交机制。设杂交概率为 p_c , 每一代进化时产生一个随机数 t , 当 $t < p_c$ 时, 执行 N 步产生 N 个新个体 (N 为种群规模)。

下面给出求解约束优化问题(2-4)的进化策略(简称算法3)的计算步骤:

(1) 输入种群规模 N , 进化代数 K_M , 初始罚因子 $r > 0$ (在文献 70, 71 中取 $K_M \in [200, 300], N \in [200, 400], r = 100$), 令 $k = 0$ 。

(2) 随机产生 N 个初始父本 x_1, \dots, x_N , 选取方差向量 σ 的初值为 $(1, 1, \dots, 1)^T \in \mathbf{R}^n$ 。

(3) 构造个体适应性函数 $E(x)$ (常取 $E(x) = P(x, r)$)。

(4) 根据上述的方差调整方案设置进化过程的方差向量, 对第 k 次父本进行变异, 选择适应性最好的个体进入第 $k+1$ 次。

(5) 令 $k = k + 1$, 若 $k > K_M$, 则计算结束; 若连续 50 代 x_0 没有改进, 则令 $r = 10r$, 转(3); 否则转(4)。

8.2.4 数值例子

下面的数值结果取自王云诚的博士论文^[72]及文献 70, 71。

【例 2.1】 (Polak(1992)) 求解

$$\min f(\mathbf{x}) \tag{2-7}$$

其中 $\mathbf{x} = (x_1, x_2)^T, f(\mathbf{x}) = \max(f_1(\mathbf{x}), f_2(\mathbf{x}))$

$$f_1(\mathbf{x}) = (x_1 - \sqrt{x_1^2 + x_2^2} \cos(\sqrt{x_1^2 + x_2^2}))^2 + 0.005(x_1^2 + x_2^2)$$

$$f_2(\mathbf{x}) = (x_1 - \sqrt{x_1^2 + x_2^2} \sin(\sqrt{x_1^2 + x_2^2}))^2 + 0.005(x_1^2 + x_2^2)$$

问题(2-7)的最优解为 $\mathbf{x}^* = (0, 0)^T$ 。

【例 2.2】(彭宏(1998)) 求解

$$\begin{cases} \min & f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{s. t.} & x_1 - 2x_2 + 1 = 0 \\ & 1 - \frac{x_1^2}{4} - x_2^2 \geq 0 \end{cases} \quad (2-8)$$

对例 2.1,采用算法 1 与算法 2 的计算结果如下面的表 8-1 所示。对例 2.2,采用算法 2 与算法 3 的计算结果见表 8-2。

表 8-1 例 1 的计算结果

项 目		计算时间(s)	调用目标函数次数	误差 $\ \hat{\mathbf{x}} - \mathbf{x}^*\ _\infty$
算 法				
杂 交	算法 1	0.71	57 100	0.154 344 0E - 21
	算法 2	0.05	2878	0.585 687 1E - 9
不 杂 交	算法 1	0.83	57 121	0.105 879 1E - 21
	算法 2	2.48	18 3121	0.351 868 6E - 9

表 8-2 例 2 的计算结果

项 目	初始种群	近似最优解	近似最优值	调用目标函数次数
算 法				
算法 2	随机产生	0.822 875 7 0.911 437 8	1.393 465	53 011
算法 3	随机产生	0.822 9 0.911 4	1.393 47	1 520 000

8.2.5 进化策略的收敛性

关于进化策略的数学基础,包括收敛性、收敛速度、计算复杂性、稳健性等的结果还不多。下面介绍的是我们在 1999 年得到的关于标准的进化策略—— $(\mu + \lambda) - ES$ 的收敛性的一个结果^[69]。

考虑全局优化问题

$$\min f(\mathbf{x}) \quad (2-1)$$

其中 $f: \mathbf{R}^n \rightarrow \mathbf{R}$ 为连续函数。

假设 $S = \{\mathbf{x} | \arg \min_{x \in \mathbf{R}^n} f(\mathbf{x})\} \neq \emptyset$, 记 $f^* = \min_{x \in \mathbf{R}^n} f(\mathbf{x})$ 。

设 $\{\mathbf{x}_1(0), \dots, \mathbf{x}_\mu(0)\}$ 为由算法 1 产生的初始种群, 令 $\alpha = \max_{1 \leq i \leq \mu} \{f(\mathbf{x}_i(0))\}$ 。

在上述条件下, 可以证明如下结果^[69]。

定理 2.1 设 $\{X(k)\}$ 是由 $(\mu + \lambda) - ES$ 产生的种群序列, $\mathbf{x}^*(k) \in X(k)$ 为第 k 代种群的最优个体, 即 $\mathbf{x}^*(k) = \arg \min_{1 \leq i \leq \mu} \{f(\mathbf{x}_i(k))\}$, 如果算法的初始种群对应的水平集 $L_\alpha = \{\mathbf{x} | f(\mathbf{x}) \leq \alpha\}$ 有界, 则 $\forall \varepsilon > 0$, 都有

$$\lim_{k \rightarrow \infty} P(|f(\mathbf{x}^*(k)) - f^*| < \varepsilon) = 1$$

这就是说: $(\mu + \lambda) - ES$, 对于一类水平集有界的连续函数 $f(\mathbf{x})$, 它依概率收敛于问题(2.1)的全局极小点。

定理 2.1 的证明比较复杂, 有兴趣的读者可参看文献 69。进一步的结果也已经得到, 参见文献 73。

8.3 遗传算法

考虑如下的全局优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in S \end{cases} \quad (3-1)$$

其中 $x = (x_1, \dots, x_n)^T$, S 为问题(3-1)的可行集。

在遗传算法中,常需对问题的解进行编码,即通过变换 F 将 S 映射为 S_g 。要求变换 F 可逆,逆变换 F^{-1} 称为解码变换。通常, S_g 中的点是字符串的形式,假设 $S_g = B^l$, 即 S_g 是长度为 l 的二进制串的全体,一个长度为 l 的二进制串称为一个染色体,染色体的每一位称为基因。

8.3.1 遗传算法的基本结构

遗传算法的主要计算过程是:从随机产生的一个初始种群开始,通过一些算子的作用,产生下一代种群,再以新产生的种群为出发点,重复上述过程,直到满足结束准则为止。

遗传算法主要采用三个算子:选择、交叉、变异。不同的编码方案,选择策略和遗传算子相结合构成了不同的遗传算法,但其基本结构相同,可描述如下:

- (1) 随机产生初始种群 $X(0) = \{x_1(0), x_2(0), \dots, x_\mu(0)\}$ 作为父代种群,令 $k = 0$ 。
- (2) 计算种群 $X(k)$ 中每一个个体的适应值,并对个体进行编码。
- (3) 从种群 $X(k)$ 中选取 $\mu/2$ 对个体进入交配池。
- (4) 对交配池中的每对个体进行交叉,产生两个新个体。
- (5) 对每个新个体依变异概率 p_m 进行变异,并把变异后的个体作为下一代种群 $X(k+1)$ 的个体,令 $k = k + 1$ 。
- (6) 判断是否满足结束准则,若满足,则计算结束;否则转(2)。

8.3.2 遗传算法实施中一些问题的处理

1. 适应值函数的选取

在传统优化方法中,判断一个解点的好坏是根据目标函数值

的大小,而在遗传算法中,则是根据适应值的大小。因此,存在目标函数到适应值函数的对应问题。适应值函数应满足:

- (1) 非负性;
- (2) 目标函数的优化方向对应适应值增大方向。

对于无约束优化问题

$$\min f(\mathbf{x}) \quad (3-2)$$

可取适应值函数 $F(\mathbf{x}) = C_M - f(\mathbf{x})$, 其中 C_M 为常数, 应保证 $F(\mathbf{x}) \geq 0$ 。

对于

$$\max f(\mathbf{x}) \quad (3-3)$$

可取适应值函数 $F(\mathbf{x}) = f(\mathbf{x}) + C_m$ 。其中 C_m 为常数, 应保证 $F(\mathbf{x}) \geq 0$ 。

上面所述是适应值函数最简单的一种选取方法, 有时根据需要, 还需对适应值进行调整, 以保证适应值之间差距既不是很大, 又要适当拉开差距, 以便强化竞争, 又能避免过早收敛。

2. 编码

编码的目的是为实现交叉、变异等类似于生物界的遗传操作。编码方式有许多种, 它与求解速度、计算精度有直接关系, 对算法具有重要影响。常用的编码方式是二进制编码。因为每个实数都与一个二进制数相对应, 可取这个二进制字符串为该实数的编码。

3. 选择策略

在遗传算法中, 每一个个体都有相应的适应值, 用以表示该个体对环境的适应能力, 它是遗传算法中用到的惟一信息。选择就是按一定方式从旧个体中选出若干个个体作为父本来产生下一代的过程。选择应保证高适应值的个体被选择机会高。它实际是对“自然选择, 适者生存”的模拟。

最简单一种选择策略是根据 $[0, 1]$ 区间内的均匀分布的随机变量的试验值进行选择。具体方法是: 将 $[0, 1]$ 区间按种群中个体

的适应值的百分比划分为若干个小区间,按均匀分布产生随机数,选择随机数所在的区间对应的个体(即随机数落在哪个小区间,则相应的个体被选中)进入交配池。以按比例选择为例,每个个体所占的百分比 r 为 $r = 100f_i / \sum_i f_i (\%)$, f_i 表示个体 i 的适应值。下面举例说明。

【例 3.1】 设由 4 个个体组成的一个种群,每个个体适应值的比例关系如下表所示。

个 体	1	2	3	4	合 计
适应值	169	576	64	361	1 170
$r(\%)$	14.4	49.2	5.5	30.9	100

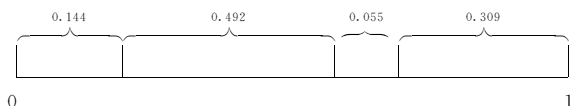


图 8-1

若按均匀分布产生的随机数为 $y_1 = 0.34$,则它位于第二个区间内,对应的个体 2 被选中进入交配池。若随机数为 $y_2 = 0.86$,则个体 4 被选中进入交配池。如此产生 4 个随机数,确定 4 个个体进入交配池。然后对交配池中的个体进行交叉操作。

4. 交叉算子

交叉算子是使种群内个体中的性状进行重新组合,产生新型后代,以实现高效搜索的重要算子。可分为一点交叉算子、两点交叉算子和多点交叉算子。

一点交叉操作就是随机地从交配池中选取一对待交叉的个体,并随机选择一个交叉位置,将其中一个个体串从交叉位置到右端的子串与另一个进行交叉的个体串对应位置的子串交换。

例如:两个待交叉的个体串为 $A = 101101$ 和 $B = 100010$,交叉位置在第 4 位后,则经过一点交叉操作后产生的两个新个体为

$A_1 = 101110$ 和 $B_1 = 100001$ 。

两点交叉算子等价于连续使用两次一点交叉算子。例如： $A = 110010, B = 001101$ ，欲由 A, B 产生新个体串 $C = 101100$ ，可采用两点交叉操作来实现。

$$\begin{array}{ccccccc}
 A & - & - & - & 1 & | & 1 & 0 & 0 & 1 & | & 0 & & 1 & | & 1 & 0 & 0 & 1 & 1 & & 0 & 1 & 0 & 0 & 1 & 1 \\
 & & & & & & & & & & & \Rightarrow & & & & & & & & & \Rightarrow & & & & & & \\
 B & - & - & - & 0 & | & 0 & 1 & 1 & 0 & | & 1 & & 0 & | & 0 & 1 & 1 & 0 & 0 & & 1 & 0 & 1 & 1 & 0 & 0 & - & - & - & C
 \end{array}$$

5. 变异算子

在标准的遗传算法中，变异算子一般是作为辅助算子使用的。它作用在个体的二进制位串上，以较小的概率 p_m 随机地改变个体串上的每一位（即将位上的 0 变为 1，1 变为 0）。变异概率 p_m 一般取为 $0.0001 \sim 0.01$ 。设个体 $S = (S_1, S_2, \dots, S_L)^T$ 经变异算子作用后变为 $S' = (S'_1, S'_2, \dots, S'_L)^T$ ，其中

$$S'_i = \begin{cases} S_i, & \theta_i > p_m \\ 1 - S_i, & \theta_i \leq p_m \end{cases}, i = 1, 2, \dots, L$$

这里 θ_i 是 0 与 1 之间的由均匀分布产生的随机数。

6. 结束准则和最优解的确定

自然界的进化过程是无终止的，遗传程序设计也是如此。但作为一次实际的运行，必须给出计算结束的条件或准则。经典的遗传算法是固定遗传代数 G ，达到后即结束。改进的方法是利用某种判别准则，判定种群已经成熟，并不再有进化趋势后作为计算结束的条件。常用的方法是：根据连续几代个体平均适应度变化很小（其差小于 $\epsilon > 0$ ）作为结束准则。

近似最优解的确定方法有两种：

- ① 当计算结束时，以当前代中的最佳个体作为最优解；
- ② 在进化过程中，保存最佳个体，当计算结束时，以保存的最佳个体作为最优解。

8.3.3 数值例子

【例 3.2】 求如下优化问题的最优解：

$$\begin{cases} \min & f(x) = x^2 \\ \text{s. t.} & 0 \leq x \leq 31 \end{cases} \quad (3-4)$$

解 (1) 在区间 $[0, 31]$ 上随机选取 4 个初始点组成初始种群。

(2) 对初始点进行编码。计算种群中每个个体的目标函数值和适应值。本题取适应值等于目标函数值。计算结果见表 8-3。

表 8-3 例 3.2 的计算结果

个体	决策变量值(x)	编码后的位串	适应值($f(x)$)	占总数的百分比(%)
A_1	13	01101	169	14.4
A_2	24	11000	576	49.2
A_3	8	01000	64	5.5
A_4	19	10011	361	30.9

(3) 按 $100f_i / \sum_i f_i(\%)$ 计算每个个体所占的百分比，列入表 8-3 中。根据个体所占的比例随机选取 4 个个体进入交配池，不妨设这 4 个个体是 A_1, A_2, A_3, A_4 。

(4) 交叉

任选两对个体进行交叉，并随机选取交叉位置，产生两对新个体，如下表所示。

表 8-4 例 3.2 的交叉操作

交配池中的个体	(x)	交配位置	新的个体	(x)	适应值($f(x)$)
0110 1	13	4	01100	12	144
1100 0	24	4	11001	25	625
11 000	24	2	11011	27	729
10 011	19	2	10000	16	256

(5) 变异

取变异概率 $p_m = 0.0001$, 对每个位串进行变异。在本例中没有发生变异。

由表 8-3, 8-4 可以看出: 经过一次迭代, 种群中个体的目标函数值的平均值增大, 由 292.5 增加到 438.5, 最优个体由 24 变为 27。这说明种群正向优化方向前进。

上面我们简要地介绍了遗传算法的基本框架, 由于时间和篇幅的限制, 还有许多重要内容未能写入书中。例如遗传算法中一个很棘手但又必须解决的问题是它的过早收敛问题。事实上也就是大范围粗糙搜索与小范围精细搜索之间的平衡问题, 或者说种群内的多样性与算法运行速度之间的协调问题。若不能保证种群的多样性, 致使种群过于集中, 将无法搜索其他区域, 陷入局部极小点。对于这个问题, 国内外已有许多学者进行过研究, 可参看文献 75 ~ 77。

8.4 进化规划

进化规划(EP) 与进化策略有许多相似之处, 但它是由美国学者 L. J. Fogel 等于 1966 年首先提出来的。早期主要用于离散优化问题, 1992 年, D. B. Fogel 将进化规划扩展到可求解连续函数的优化问题。1994 年, D. B. Fogel 给出了 EP 算法的一个收敛性证明。

在一般的 EP 算法中, 有时会遇到最优值几十代甚至几百代无改进的情况, 白白浪费了很多的计算时间。我们对 EP 算法进行了研究与改进。数值试验表明: 经过改进的 EP 算法不仅可行、有效, 而且简单易行, 容易编程计算。

8.4.1 一种改进的 EP 算法

考虑如下的约束优化问题：

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in \Omega \end{cases} \quad (4-1)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $f \in C$, Ω 为有界闭集。

求解问题(4-1)的改进的 EP 算法(简记为算法 A) 计算步骤为：

1. 初始化

在 Ω 中随机选取 m 个点 $x_1(0), x_2(0), \dots, x_m(0)$, 构成初始种群 $X(0) = \{x_1(0), x_2(0), \dots, x_m(0)\}$, 计算初始种群中每个个体 $x_i(0)$, $i = 1, 2, \dots, m$ 处的目标函数值 $f(x_i(0))$, $i = 1, 2, \dots, m$, 令 $t = 0$, $x^*(0) = \arg \min_{1 \leq j \leq m} \{f(x_j(0))\}$ 。

2. 进行变异: 令 $x_{i+m}(t) = x_i(t) + \alpha \xi$, $i = 1, 2, \dots, m$, 其中 $\xi \sim N(0, \sigma) = (N(0, \sigma_1), N(0, \sigma_2), \dots, N(0, \sigma_n))^T$, $N(0, \sigma_i)$ 表示均值为 0, 方差为 σ_i^2 的正态分布, 且 ξ 的 n 个分量相互独立。 $\alpha \in (0, 1]$ 为压缩因子。

若 $x_{i+m}(t) \in \Omega$, 则转第 3 步; 否则重复第 2 步。

3. 计算新个体 $x_{i+m}(t)$, ($i = 1, 2, \dots, m$) 处的目标函数值 $f(x_{i+m}(t))$, $i = 1, 2, \dots, m$ 。

4. 进行选择: 从 $\{x_1(t), \dots, x_{2m}(t)\}$ 中选取 m 个函数值最小的个体构成新一代种群 $X(t+1) = \{x_1(t+1), \dots, x_m(t+1)\}$, 令 $x^*(t+1) = \arg \min_{1 \leq j \leq m} \{f(x_j(t+1))\}$ 。

5. 终止检验: 判别 $X(t+1)$ 是否满足终止条件, 若满足, 则计算结束, 输出最优解 $x^*(t+1)$; 否则, 令 $t = t+1$, 返回第 2 步。

8.4.2 数值例子

为了检验改进的进化规划算法的可行性和有效性,选取了三个典型的优化算例进行测试,其中两个是多峰函数优化问题,另一个是非光滑优化问题。数值试验都是在方正 133 微机上进行的。每个问题的求解精度均为 $\varepsilon = 10^{-5}$,为了提高算法的执行效率,当迭代次数超过 10 次以后,取压缩因子 $\alpha = 0.02$,10 次以内取 $\alpha = 1$ 。

【例 4.1】

$$f(x) = -2\pi x - \sum_{i=0}^{10} \sin(2^{i+1}\pi x), \quad x \in [0,1] \quad (4-2)$$

该函数在区间 $[0,1]$ 内有多多个局部极值点。其全局最优值为 $f(0.566713) = -8.817893$ 。

$$\begin{aligned} \text{【例 4.2】} \quad f(x,y) = & (\cos(2\pi x) + \cos(2.5\pi x) - 2.1) \times \\ & (2.1 - \cos(3\pi y) - \cos(3.5\pi y)) \end{aligned}$$

其中 $(x,y) \in [0,3] \times [0,1.5]$ 。该函数在区域 $[0,3] \times [0,1.5]$ 内有多多个局部极值点和一个全局最优点。全局最优值为 $f(0.438974, 0.305734) = -16.09172$ 。

$$\text{【例 4.3】} \quad f(x,y) = \max(f_1(x,y), f_2(x,y))$$

其中

$$f_1(x,y) = (x - \sqrt{x^2 + y^2} \cos \sqrt{x^2 + y^2})^2 + 0.005(x^2 + y^2)$$

$$f_2(x,y) = (x - \sqrt{x^2 + y^2} \sin \sqrt{x^2 + y^2})^2 + 0.005(x^2 + y^2)$$

该函数虽然是一个单峰函数,但却是非光滑的。其全局最优值为 $f(0,0) = 0$ 。

下表给出了算法 A 的计算结果。

表 8-5 三个算例的数值结果

算例	方差	种群规模	迭代次数	CPU 时间(秒)	最优解	最优值
1	$\sigma = 0.5$	40	64	0.39s	0.5667138	- 8.817891
	$\sigma = 0.9$	40	36	0.27s	0.5667129	- 8.817884
2	$\sigma_1, \sigma_2 = 0.5$	50	104	0.98s	(0.4385967, 0.3058772)	- 16.091710
	$\sigma_1, \sigma_2 = 0.25$	50	38	0.33s	(0.43987949, 0.3058355)	- 16.091720
3	$\sigma_1, \sigma_2 = 1$	60	25	0.285	(2.070863E - 03, 4.478674E - 03)	8.320674E - 06
	$\sigma_1, \sigma_2 = 2$	60	12	0.11s	(1.219543E - 04, 2.346760E - 03)	4.991443E - 06

典型实例的优化计算表明,改进的进化规划算法具有良好的全局搜索能力和较快的收敛速度。在实际计算中,参数 σ, m 和 α 的取值因题而异,主要应考虑以下几点:

(1) 种群规模 m 的选取对进化算法的性能影响很大。种群规模选择过小,算法的全局搜索能力将受到影响,可能会过早地收敛于局部极小点;种群规模选择过大,将降低算法的计算速度,造成计算资源上的浪费。

(2) 变异算子是改进的进化规划算法的主要进化手段,它体现了算法的全局搜索能力。其主要作用在于不断开拓解的新空间,并使算法达到局部最优时能够使解逃离出局部极小陷阱,因为在理论上变异后的个体可以是整个解空间内的任何一个点。正态分布中的均方差 σ ,在算法中起类似于古典优化算法中的搜索步长的作用,在这里也称之为步长。随着迭代的进行,为了提高计算精度和收敛速度,步长应该随之减小。改进的进化规划算法采用的步长修正策略为:如果若干代内找不到下降点,说明步长太大,此时将步长适当压缩,以便较快地找到下降点。对于可行域范围较大的优化问题,初始步长 σ 也应较大,以便增强算法的空间搜索能力。

反之,则可令 σ 相对小些,压缩因子 α 应该与初始步长相互协调,其主要功能在于控制算法的计算精度和收敛速度。

8.4.3 改进 EP 算法的收敛性

下面介绍的是我们在 2002 年发表的一个结果^[96]。

首先,对全局优化问题(4-1)作如下假设:

假设 4.1 (i) 问题(4-1)的可行域 Ω 为 \mathbf{R}^n 中的紧集。

(ii) 目标函数 $f(x)$ 是区域 Ω 上的连续函数。

由假设 4.1 易知 $S = \{x | \arg \min_{x \in \Omega} f(x)\} \neq \emptyset$ 。任给 $\varepsilon > 0$,若记

$$D_0 = \{x \in \Omega | |f(x) - f^*| < \varepsilon\}; \quad D_1 = \Omega \setminus D_0 \quad (4-3)$$

其中 $f^* = \min_{x \in \Omega} f(x)$,则上述算法中产生的 m 个点可以分成两种状态:

(1) 至少有一个点属于 D_0 ,记为状态 S_0 ;

(2) m 个点均属于 D_1 ,记为状态 S_1 。

在上述条件下,可以证明如下结果^[96];

定理 4.2 假设 $\{X(t)\}$ 是由改进的进化规划算法产生的种群序列,其中 $x^*(t) \in \{X(t)\}$ 是第 t 代种群中的最优个体,即 $x^*(t) = \arg \min_{1 \leq i \leq m} f(x_i(t))$ 。如果假设 4.1 成立,则有

$$P\{\lim_{t \rightarrow \infty} f(x^*(t)) = f^*\} = 1 \quad (4-4)$$

即种群序列以概率 1 渐近收敛于优化问题(4-1)的全局极小点。

定理 4.2 的证明需要用到较多的预备知识,故从略,有兴趣的读者,可参看文献 96。

8.5 模拟退火方法

20 世纪 80 年代中期以后,随着高性能计算机的飞速发展,一系列新的优化方法,如模拟退火方法(SA, Simulated Annealing)、

遗传算法(GA)、进化规划(EP)、进化策略(ES)等获得了极其迅速的发展和广泛的应用。它们对解决大型复杂系统中出现的许多困难优化问题表现出很好的适应性。这些方法往往不仅具有简单、通用、自组织、自适应、自学习、适于并行处理等优点,而且有望成为将数值计算与语义表达、形象思维等高级智能行为联系的桥梁。因此,被一些学者称为智能优化方法。上世纪 90 年代以来,对这类算法的研究日渐成为计算机科学、信息科学和运筹优化领域研究的一个热点,并被广泛应用于机器学习、模式识别、蛋白质结构预测、图像处理、人工生命、经济预测等许多领域。本章前四节已对进化策略、进化规划和遗传算法作了简要介绍。本节将对模拟退火方法(SA)及其在蛋白质结构预测中的应用作一介绍。

8.5.1 模拟退火方法的基本思想和步骤

模拟退火方法是一种随机性的全局优化方法,其基本思想来源于固体的退火过程。其思想最早是由 Metropolis 等于 1953 年提出的,但未受到科学界的足够重视。直到 1983 年, Kirkpatrick 等首先认识到固体退火过程与组合优化问题之间的相似性,给出了一种模拟固体退火过程的迭代算法——模拟退火算法,这一方法才逐渐引起人们的重视,并在解决大规模组合优化问题中获得了成功的应用。

考虑如下的优化问题:

$$\min f(x), x \in \mathbf{R}^n \quad (5-1)$$

模拟退火方法的基本思想是将优化问题比拟成一个物理系统,将优化问题的目标函数 $f(x)$ 比拟为物理系统的能量 $E(x)$,模拟退火方法从某一较高的初始温度 $T_0 > 0$ 开始,通过模拟物理系统逐步降温以达到最低能量状态的退火过程来获得优化问题的全局最优解。模拟退火算法的基本步骤可简述如下:

1. 任给初始点 $x^{(0)}$, 足够大的初始温度 $T_0 > 0$, 令 $x^{(i)} = x^{(0)}, k$

$= 0$, 计算能量值 $E(\mathbf{x}^{(0)})$;

2. 若在该温度下达到内循环停止条件, 则转第 3 步; 否则从邻域 $N(\mathbf{x}^{(i)})$ 中随机选取一个解 $\mathbf{x}^{(j)}$, 计算能量差 $\Delta E_{ij} = E(\mathbf{x}^{(j)}) - E(\mathbf{x}^{(i)})$, 若 $\Delta E_{ij} \leq 0$, 则令 $\mathbf{x}^{(i)} = \mathbf{x}^{(j)}$; 否则若 $p = \exp(-\Delta E_{ij}/T_k) > \eta$ (η 为 $(0, 1)$ 上均匀分布的随机数), 则令 $\mathbf{x}^{(i)} = \mathbf{x}^{(j)}$, 重复第 2 步;

3. 置 $T_{k+1} = d(T_k)$, $k = k + 1$; 若满足结束准则, 则计算结束; 否则返回第 2 步。

其中 $d(T_k)$ 为降温方式。

上述算法常称为经典模拟退火方法(CSA)。

8.5.2 连续变量的模拟退火方法

早期的 SA 是针对组合优化问题的^[88,89], 但实践中还经常会遇到连续变量的优化问题, 1996 ~ 1997 年, 清华大学胡山鹰、陈丙珍等和中科院系统所杨若黎、顾基发对连续变量优化问题的模拟退火方法进行了有价值的探讨^[90~92]。

对模拟退火算法的研究主要分为两个方面, 一是基于马尔可夫链的有关理论, 研究算法的渐近收敛性; 二是研究算法及其在各类问题中的应用。文献^[88,89]指出, 模拟退火算法收敛到全局最小点要求满足如下条件: (1) 初始温度足够高, 使该温度下所有状态以相同的概率出现; (2) 降温速度足够慢, 每一温度下达到准平衡状态; (3) 终止温度趋近于零。但在实现过程中这些条件都难以完全满足, 所以模拟退火算法只能以一定的概率找到近似的全局最优解。冷却进度表、邻域结构和新解产生器、接受准则和随机数产生器被称为模拟退火算法的三大支柱, 它们对算法的收敛性起决定性作用, 将模拟退火算法引入连续变量函数的优化问题, 关键是邻域结构和新解产生器的构造, 不合适的邻域结构可能导致算法不收敛。关于邻域结构的一个重要结论是: 算法收敛性要求任意

两个可行解或互为邻近解,或互不为邻近解。在某些情况下要求邻近解具有相同的产生概率。

数值试验表明,以往提出的模拟退火算法在变量较少时能得到很好的求解效果,但是随着变量数目的增加,就难以收敛到全局最优解,针对蛋白质结构预测模型等自然科学与工程实际问题多变量多极值的特点,我们给出了一种邻域结构的构造方式和新解产生方法,使算法能较快收敛到全局最优解,并且在解决多维的全局优化问题时显示出良好的数值稳定性。下面就来介绍这一改进的SA,简称算法A(简记为SAA)。

考虑如下的简单约束优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & \mathbf{x} \in [\mathbf{x}^D, \mathbf{x}^U] \end{cases} \quad (5-2)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T, f \in c$ 。

首先给出一种新解产生方法:假设某一状态下决策变量 \mathbf{x} 取值 $(x_1, x_2, \dots, x_n)^T$;从当前解 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 中随机选取一个分量 x_r 产生随机扰动: $x'_r = x_r + \text{rand. scale. } (x_r^U - x_r^D)$, 然后进行边界处理得到新解:

$$x_r^N = \begin{cases} x_r^U - (x_r^D - x'_r), & x'_r < x_r^D \\ x'_r, & x_r^D \leq x'_r \leq x_r^U \\ x_r^D + (x'_r - x_r^U), & x'_r > x_r^U \end{cases} \quad (5-3)$$

这样处理之后,任意两个可行解满足了或互为邻近解或互不为邻近解的条件:(1) 每一点(包括边界点)的邻域规模相同,新解的产生概率相同,保证了算法要求的收敛性条件。(2) 产生随机扰动的分量随机选取。

算法A的计算步骤为:

step1 设定初始温度 T_{\max} , 每一温度下迭代次数 L_{\max} , 邻域规模因子 scale , 温度下降因子 dt ;

step2 随机产生初始解 $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T \in [\mathbf{x}^D, \mathbf{x}^U]$, 计

算目标函数值 $fun^0 = f(x^0)$;

step3 判断是否满足终止条件,若满足,则结束;否则,令 $T = T * dt, L = 1$,转 step4;

step4 从 $\{1, 2, \dots, n\}$ 中随机选取一个数 r ,使 x^0 中第 r 个变量产生随机扰动: $x_r = x_r^0 + \text{rand} * \text{scale} * (x_r^u - x_r^d)$, (rand 为 -1 到 1 之间一个随机数)。如果 x_r 超出上下边界,则按(5-3)式进行边界处理,从而得到一个新解 $x(x_1, x_2, \dots, x_n)$,计算新解的目标函数值 $fun = f(x)$;

step5 计算目标函数差 $Df = fun - fun^0$,根据 Metropolis 法则,判断是否接受新解。若接受,则令 $x^0 = x, fun^0 = fun$,转 step6;否则转 step6;

step6 若满足 $L > L_{\max}$,转 Step3,否则令 $L = L + 1$,转 step4。

该算法有以下几个可调参数:

1. 初始温度 T_{\max} : 主要根据目标函数差来确定,保证初始接受率足够高即可;

2. 终止条件: 在最优值未知的情况下,终止条件比较难确定,一般采取两种准则:一是给定终止温度;二是连续多次降温,能量函数值不再下降;

3. 温度下降因子 dt : 一般取 $0.95 - 0.98$;

4. 同一温度迭代次数(Markov 链长度) L_{\max} : L_{\max} 选取与问题规模和解空间大小有关;

5. 邻域规模因子 scale : 邻域规模因子与解空间的规模有直接关系,下面的算例取 $0.2 - 0.5$ 。

在第 5 步中,根据 Metropolis 法则,判断是否接受新解是指:若 $Df \leq 0$,则接受新解;否则,若 $p = \exp(-Df/T) > \eta$ (η 为 $(0, 1)$ 上均匀分布的随机数),则接受新解。

8.5.3 数值例子

为了说明上述算法 A(SAA) 的有效性,下面给出两个数值例子的结果。

$$\text{【例 5.1】} \quad \begin{cases} \min & f(\mathbf{x}) = \frac{1}{100} \sum_{i=1}^{100} (x_i^4 - 16x_i^2 + 5x_i) \\ \text{s. t.} & -10 \leq x_i \leq 10, i = 1, 2, \dots, 100 \end{cases} \quad (5-4)$$

引自文献[92],最优值为 $f^* = -78.3323$ 。

$$\text{【例 5.2】} \quad \begin{cases} \min & f(\mathbf{x}) = n + \sum_{i=1}^n [x_i^2 - \cos(2\pi x_i)] \\ \text{s. t.} & -5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, n \end{cases} \quad (5-5)$$

最优值为 $f(0) = 0$ 。

下表给出了例 5.1、例 5.2 的数值试验结果,从中可以看出: SAA 的计算效率要比文 92 的算法高。

表 8-6 数值试验结果

		目标数调用次数			精度
		最小	最大	平均	
例 5.1	SAA	16741	17581	17052	$1e-3$
	文献[92]	19980	27899	23664	$1e-3$
例 5.2		最小	最大	平均	精度
	$n = 100$	44691	45391	45171	$1e-5$
	$n = 1000$	192501	194101	193389	$1e-3$
	$n = 3000$	1192601	1193401	1192867	$1e-3$

8.5.4 蛋白质结构预测的一个优化模型

为了说明模拟退火方法在蛋白质结构预测中的应用,我们先来介绍蛋白质结构预测的一个优化模型。

联合残基力场是一种平均力场,它是通过对蛋白质数据库 PDB(Protein Data Bank) 中 195 种高分辨率的非同源蛋白质晶体

结构进行统计分析,结合全原子力场平均化建立起来的。

在联合残基力场中,氨基酸序列的主链被简化为一系列用虚键 $C^\alpha - C^\alpha$ 连接的 α 碳原子(C^α),每个 α 碳原子接一个联合侧链(SC),两个 α 碳原子之间的 $-NH-CO-$ 基团用一个联合肽基(P) 表示。其中只有 SC 和 P 为作用点, C^α 仅辅助确定作用点的几何位置;虚键 $C^\alpha - C^\alpha$ 链长固定不变(3.8 埃),侧链键 $C^\alpha - SC$ 链长仅跟残基类型有关,自由变量为:虚键二面角 γ 、虚键键角 θ 和侧链键角 α_{SC} 和 β_{SC} 。在文献[93] 中模型一和模型二仅考虑了联合残基力场的一个和三个能量项,这里的模型包括五个能量项,我们称为模型三:

$$\begin{aligned} \min f(X) = U = & \sum_{i < j} U_{SC_i SC_j}(\gamma, \theta, \alpha_{SC}, \beta_{SC}) + \omega_{tor} \sum_i U_{tor}(\gamma) + \\ & \omega_{loc} \sum_i U_b(\theta) + \sum_{i \neq j} U_{SC_i P_j}(\gamma, \theta, \alpha_{SC}, \beta_{SC}) + \\ & \omega_{el} \sum_{i < j-1} U_{P_i P_j}(\gamma, \theta) \end{aligned} \quad (5-6)$$

$$X \in [X^D, X^U]$$

$$X = (\gamma_1, \dots, \gamma_{N-1}, \theta_1, \dots, \theta_{N-2}, \alpha_{SC1}, \dots, \alpha_{SCN}, \beta_{SC1}, \dots, \beta_{SCN})^T$$

其中, $U_{SC_i SC_j}$ 表示联合侧链 SC_i 与 SC_j 相互作用,包含了侧链间疏水 / 亲水作用的平均自由能; $U_{SC_i P_j}$ 为联合侧链 SC_i 与联合肽基 P_j 相互作用; $U_{P_i P_j}$ 为联合肽基 P_i 与 P_j 相互作用,主要指它们之间的静电作用; U_{tor} 和 U_b 两项说明了局部性质, U_{tor} 为虚键二面角扭转能, U_b 为虚键键角变形能。其中, $U_{SC_i SC_j}$, U_{tor} 和 U_b 的描述可参考文献[93],下面列出另外两项。

联合侧链 SC_i 与联合肽基 P_j 之间的相互作用 $U_{SC_i P_j}$ [94,95]

该项是为了防止一个残基的侧链与另一个残基的主链靠得太近 而造成的不合理结构加入的惩罚项。对于相邻的两个残基, $U_{SC_i P_j}$ 可忽略不计。当第 i 个残基和第 j 个残基不相邻时, $U_{SC_i P_j}$ 通过下式(5-7) 计算。

$$U_{SCP_{ij}} = \epsilon_{SCP} \left(\frac{r_{SCP}^0}{r_{ij}} \right)^6 \quad (5-7)$$

其中, $\epsilon_{SCP} = 0.3 \text{ kcal/mol}$, $r_{SCP}^0 = 4.0$ 埃, r_{ij} 表示残基 i 和残基 j 之间的距离。

联合肽基 P_i 和 P_j 之间的相互作用 $U_{P_i P_j}$ 由下式确定

$$U_{P_i P_j} = \frac{A_{P_i P_j}}{r_{ij}^3} (\cos \alpha_{ij} - 3 \cos \beta_{ij} \cos \gamma_{ij}) - \frac{B_{P_i P_j}}{r_{ij}^6} [4 + (\cos \alpha_{ij} - 3 \cos \beta_{ij} \cos \gamma_{ij})^2 - 3(\cos^2 \beta_{ij} + \cos^2 \gamma_{ij})] + \epsilon_{P_i P_j} \left[\left(\frac{r_{P_i P_j}^0}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{P_i P_j}^0}{r_{ij}} \right)^6 \right] \quad (5-8)$$

其中, $\alpha_{ij}, \beta_{ij}, \gamma_{ij}$ 是定义肽基 P_i, P_j 相对位置的角度, r_{ij} 表示 P_i 和 P_j 之间的距离。常数项 $A_{P_i P_j}$ 和 $B_{P_i P_j}$ 可参考文献[94]。

8.5.5 模拟退火方法(SAA)在蛋白质结构预测中应用举例

利用算法 A 对脑啡肽和牛胰岛素的 B(D) 链的空间结构进行了预测。

脑啡肽 (Met-enkephalin) 是一条五残基的肽链, 它的氨基酸顺序是: H-Tyr-Gly-Gly-Phe-Met-OH。它的空间结构为典型的 $\text{II}'\beta$ 转角。在全原子力场中, 脑啡肽有 24 个可变的主链和侧链二面角和键角, 据估计能量表面的局部极小点超过 $3^{24}(10^{11})$ 个。Cornell 大学 Beker 化学实验室用 Monte Carlo 能量极小化方法和构象空间退火 (Conformation Space Annealing) 方法计算了脑啡肽的全原子力场 ECEPP 能量函数值, 两种方法得到的最低能量值分别为 -12.90 kcal/mol 和 -11.707 kcal/mol 。在本书所描述的联合残基力场模型中, 可变二面角和键角的数量减少到 17, 而由于甘氨酸 Gly 在该模型中虚键键长为 0, 相应键角 α_{sc} 和 β_{sc} 的改变对能量变化不产生影响, 所以脑啡肽的联合残基力场模型实质上

只有 13 个变量。利用算法 A 对脑啡肽的联合残基模型进行了 20 次运算,其中 13 次运算收敛到同一个极小点,能量极小值为 -13.857kcal/mol ,其他 7 次运算结果高于此能量值。初始温度分别设为 10,20 和 50;终止条件为温度降至 10^{-7} 。在 P III 733 微机上运行时间为 900 秒 \sim 1460 秒。结果见表 8-7。

表 8-7 脑啡肽能量极小化结果

能量, kcal/mol	γ_1	γ_2	γ_3	γ_4	θ_1	θ_2	θ_3	α_{SC1}	α_{SC4}	α_{SC5}	β_{SC1}	β_{SC4}	β_{SC5}
-13.857	127.12	-165.00	105.26	-127.86	113.91	89.44	84.14	94.55	102.46	155.51	39.04	72.81	-116.44

注:由于第 2,3 个残基为 Gly,其侧链虚键长度为零,所以 $\alpha_{SC2}, \alpha_{SC3}, \beta_{SC2}, \beta_{SC3}$ 的变化对能量的计算不产生影响。

分析模型一、模型二^[93]和模型三的优化结果可以得到以下结论:联合侧链相互作用在联合残基力场中起主导作用;模型三的优化结果比模型一和模型二的结果更合理些,说明其他能量项在蛋白质的结构预测中同样是不可忽略的;蛋白质的能量表面存在极多的局部极小点,而且它们的极小值差别微小,在得到的极小值非常相近的情况下,解的情况可能完全不同,这就进一步增加了蛋白质结构预测的难度。

牛胰岛素(Bovine Despentapeptide Insulin)含有 4 条肽链,空间结构呈典型的全 α 螺旋结构,其中 B(D) 链为一个 α 螺旋结构域,由 25 个氨基酸残基组成,其氨基酸序列为: H-PHE-VAL-ASN-GLN-HIS-LEU-CYS-GLY-SER-HIS-LEU-VAL-GLU-ALA-LEU-TYR-LEU-VAL-CYS-GLY-GLU-ARG-GLY-PHE-PHE-OH。从第 8 个氨基酸残基 GLY 到第 22 个氨基酸残基 ARG 为一个完整的 α 螺旋结构。在联合残基力场模型中,25 个残基的蛋白质共有 97 个变量,注意到 3 个甘氨酸 GLY 的存在,对能量变化起作用的变量有 91 个,能量表面的局部极小点超过 $3^{91}(10^{44})$ 个。我们设定初始温度为 10,降温因子为 0.98,终止条件为温度低于 10^{-7} ,10 次运算有 3 次收敛到同一个极小点,极小值为

— 126.313kcal/mol, 其他 7 次运算结果都比该极小值高 10kcal/mol 以上。在 P III 733 微机上运行时间为 4.5 — 5.0 小时。

习 题 八

1. 用进化策略(算法 1) 求解如下的优化问题:

$$(1) \min f(x) = x^4 - 16x^2 + 5x$$

$$(2) \min f(x) = 100(x_1^2 - x_2^2)^2 + (1 - x_1)^2$$

$$(3) \min f(x) = |x_1| + 4 - \cos x_2 - \cos x_3 - \cos(2x_3) + (|x_2| + |x_3|)/3$$

2. 画出用进化策略(算法 1) 求解无约束优化问题 $\min f(x), x \in \mathbf{R}^n$ 的程序框图, 编写计算程序, 并求解 1. (2)(3)。

3. 画出求解单峰函数优化问题的进化策略(算法 2) 的程序框图, 编写计算程序, 并求解如下问题:

$$(1) \min f(x) = \sum_{j=1}^n x_j^2;$$

$$(2) \text{ 8.2 节例 2.1;}$$

$$(3) \text{ 8.2 节例 2.2。}$$

4. 用遗传算法求解习题 1(1), (2), (3) 及

$$(4) \min f(x) = \frac{1}{100} \sum_{i=1}^{100} (x_i^4 - 16x_i^2 + 5x_i)$$

其中 $x_i \in [-10, 10], i = 1, 2, \dots, 100$ 。

5. 画出遗传算法的程序框图, 编写计算程序, 并求解 1. (2), (3) 及 4. (4)。

6. 画出改进的进化规划算法(算法 A) 的程序框图, 编写计算程序, 并求解

$$(1) \begin{cases} \min & f(\mathbf{x}) = \sum_{i=1}^{100} \frac{x_i^2}{200} - \prod_{i=1}^{100} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\ \text{s. t.} & -100 \leq x_i \leq 100, i = 1, 2, \dots, 100 \end{cases}$$

$$(2) \begin{cases} \min f(\mathbf{x}) = n + \sum_{i=1}^n [x_i^2 - \cos(2\pi x_i)] \\ \text{s. t.} & -5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, n \end{cases}$$

7. 画出改进的模拟退火方法(SAA)的程序框图,编写计算程序,并求解7(1),(2)。

8. 求解如下的优化问题

$$(1) \begin{cases} \min & f(\mathbf{x}) = \sum_{i=1}^{n-1} [(x_i^2 - x_{i+1})^2 + (1 - x_i)^2] \\ \text{s. t.} & |x_i| \leq 30, i = 1, 2, \dots, n \end{cases}$$

$$(2) \begin{cases} \min & f(\mathbf{x}) = - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) \\ \text{s. t.} & |x_i| \leq 500, i = 1, 2, \dots, n \end{cases}$$

$$(3) \begin{cases} \min & f(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10) \\ \text{s. t.} & |x_i| \leq 5.12, i = 1, 2, \dots, n \end{cases}$$

* 第 9 章 优化方法应用举例

优化方法在工程和管理中有广泛的应用。本章从我们的科研实践中挑选了五个应用例子,向读者做一介绍。这些例子都来自实践,且大多已在工程和管理中实际应用,取得了较好的效果,希望对读者有所启发。

9.1 经济预测中的人工神经网络方法

9.1.1 人工神经网络简介

人工神经网络是用大量的简单单元构成的非线性系统,它模仿人脑进行信息处理。具有学习、记忆、计算及各种智能处理功能。它可以在不同程度和层次上模仿人脑神经系统的信息处理、存储及检索功能。

目前应用较广,其基本思想最直观、最容易理解的人工神经网络是误差反向传播神经网络(Error-back-Propagation),简称 BP 网络。它是一种具有三层或三层以上的阶层型神经网络。上、下层之间各神经元实现全连接,即下层的每一个单元都实现全连接,而每层神经元之间无连接,网络按有教师示教的方式进行学习,当一对学习模式提供给网络之后,神经元的激活值从输入层经各中间层向输出层传播,在输出层的各神经元获得网络的输入响应。然后按减小期望输出与实际输出误差的方向,从输出层经各中间层逐

层修正各连接权,最后回到输入层,随着这种误差逆传播修正的不断进行,网络对输入模式响应的正确率也不断上升。

典型的 BP 网络是三层前馈阶层网络,网络的结构如图 9-1 所示。网络由输入层、隐含层、输出层(记为 i, j, k) 组成。各层之间实现全连接。BP 网络的学习,由四个过程组成:输入模式由输入层经隐含层向输出层的“模式顺传播”过程;网络的希望输出与网络实际输出之差的误差信号由输出层经隐含层逐层修正连接权的“误差逆传播”过程;由“模式顺传播”与“误差逆传播”的反复进行的网络“记忆训练”过程;网络趋向收敛即网络的全局误差趋向极小值的“学习收敛”过程。

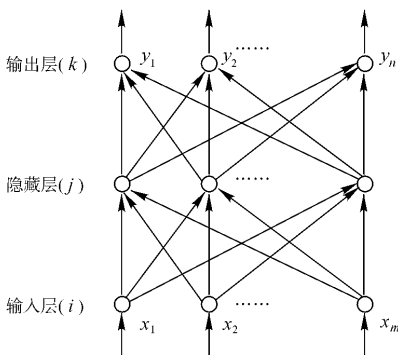


图 9-1 BP 网络的示意图

激励函数采用 S 型函数。函数 $f(x) = \frac{1}{1 + \exp[-(x + \theta)]}$, 其图像如图 9-2 所示,这个函数的优点是:它是一个连续可微的函数,因此可以用梯度法来求其最小值,这也就是这种神经网络学习算法的原理。BP 网络的学习算法称为反向传输算法

(Back-Propagation), 简称 BP 算法。

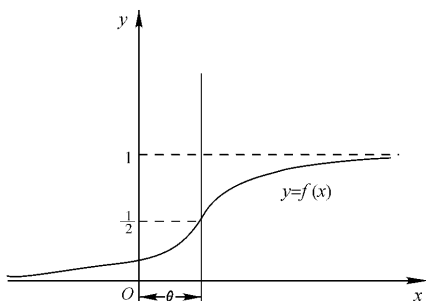


图 9-2 激励函数 $f(x)$ 的图像

给定系统的一组期望输出值 $\{t_{pk}\}$, 一般来说, 系统的输出值 $\{o_{pk}\}$ 与 $\{t_{pk}\}$ 不尽相同, 其平方误差 E 为

$$E = \frac{1}{2} \sum_p \sum_k (t_{pk} - o_{pk})^2 \quad (1-1)$$

网络学习的目标就是调整各连接权, 使整个系统的误差尽可能小。利用著名的 δ 学习律, 可以推导出权的学习公式为:

$$\Delta_p w_{kj} = \eta \delta_{pk} o_{pj} \quad (1-2)$$

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi} \quad (1-3)$$

其中, η 表示步长, δ_{pk} 和 δ_{pj} 分别为

$$\delta_{pk} = (t_{pk} - o_{pk}) o_{pk} (1 - o_{pk}) \quad (1-4)$$

$$\delta_{pj} = o_{pj} (1 - o_{pj}) \sum_k (\delta_{pk} w_{kj}) \quad (1-5)$$

对这类层次神经网络, 存在一个非常重要的连续函数影射定理——Kolmogorov 定理:

给定任一连续函数 $\Phi: E^m \rightarrow R^n, \Phi(X) = Y, E^m$ 是 m 维单位立方体, Φ 可以精确地由一个三层神经网络实现, 此神经网络的第一

层有 m 个处理单元,中间层有 $2(m+1)$ 个处理单元,输出层有 n 个处理单元。进一步的研究工作证明:含有一个隐含层的 BP 网络可以任意精度逼近定义在一个紧集上的任意非线性函数。

同时,研究表明,四层网络的结果比三层网络更易陷入局部极小点。此外,过多的网络节点会使网络的泛化能力减弱,网络预测能力下降。因此,在下面的预测模型中采用三层 BP 网络,其隐节点数略大于输入节点数目。

9.1.2 经济预测的神经网络模型及结果

1987 年, Lapedes 和 Farber 首次应用人工神经网络技术来进行经济预测,给经济预测科学又打开了一个新的天地。在我国,从 90 年代开始也陆续有人利用神经网络进行预测研究。在人工神经网络预测方面,较多的是采用 BP 网络。本节以某地区第一、二、三产业增加值和国内生产总值为预测目标,以 BP 网络为基本构架,设计出一串由 BP 网络组成的“地区主要经济指标的人工神经网络预测系统”。下面具体说明如何建立这个人工神经网络预测模型。

假设已知某地区第一、二、三产业的数据,记第 t 年的第一、二、三产业的增加值分别为 x_t, y_t, z_t , 国内生产总值 $G_t = x_t + y_t + z_t$ 。系统设计的目标是预测某地区今后 M 年第一、二、三产业的增加值和国内生产总值。设 T 是当前年份, M 为最大预测年数。从 $T+1$ 年开始,每一个 $x_t, y_t, z_t (t \in \{T+1, \dots, T+M\})$ 都是由一个单独的 BP 网络子系统进行预测,这样整个预测系统共有 $3M$ 个网络子系统组成。各子系统的输入亦有差别,但其类型是一样的,除包括预测年份以前的第一、二、三产业增加值(或预测值)外,还有与 x_t, y_t, z_t 有关的其他经济指标或相关因素,例如总人口,各产业的固定资产投资等,各产业预测系统采用的相关因素也各不相同,分别记为 $h1_t^i, h2_t^i, h3_t^i, (i = 1, 2, \dots, n)$ 。

预测第 $T+1$ 年第一产业和第 $T+K+1$ 年第一产业的子系

统结构如图 9-3 和图 9-4 所示。对第二、三产业的预测子系统与其类似,只要改变相应参数即可。这样,模型就可以利用输入的数据逐年预测该地区今后各年第一、二、三产业的增加值,该地区的 GDP 由三个产业加和即可。

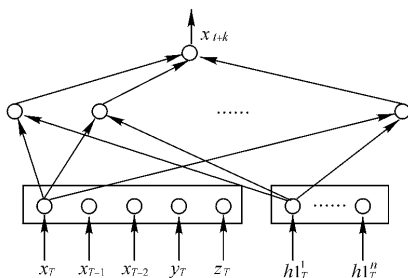


图 9-3

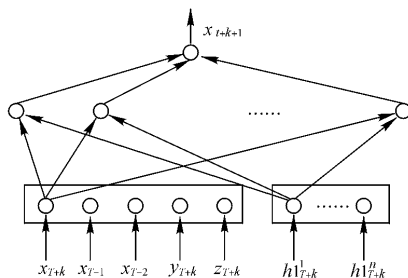


图 9-4

应用以上模型对某地区进行了预测,获得了较好的预测效果。以第二产业为例,如图 9-5 所示,1994 年以前为实际值,1994 年以

后为预测值。从图中可以看出,对于该地区,其经济发展比较平稳,每年第二产业的增长速度保持了基本相同的速度,只是在 1991 年前后略有迟缓,以后又继续保持比较良好的发展趋势。这项结果从 1994 年开始预测,该地区经过近三年的发展,事实证明模型预测的结果与该地区实际发展状况符合良好。这也从另一个侧面说明神经网络预测模型具有抗干扰性,当数据在某一点出现奇性时,模型仍能较好地处理它。

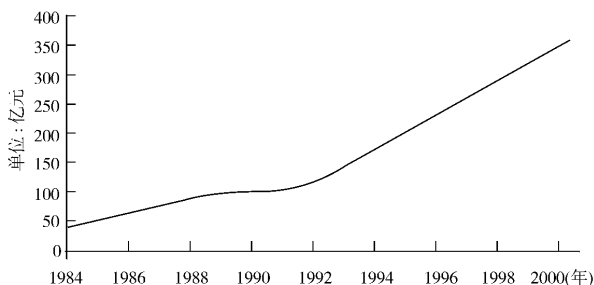


图 9-5 实际值及人工神经网络模型预测值

9.1.3 神经网络预测系统的特点

人工神经网络预测模型之所以取得了较好的预测结果,是因为具有以下特点。

(1) 预测模型采用误差反向传播模型(BP 网络)作为神经网络

络预测模型的基本结构,这是因为BP网络在学习算法、函数形式、收敛速度、网络结构等方面的理论已经比较成熟。由于三层网络可以实现任意非线性映射,而四层网络的结果比三层网络更容易陷入局部极小点,同时,过多的网络节点会使网络的预测能力下降,因此,本模型选用三层BP网络,其隐节点的数目略大于输入节点数。在预测中,每个BP网络必须“真正学习到地区经济的发展规律”后才能用于预测,这就要求网络学习后误差必须足够小。另外,过度学习会导致模型学习效果很好,但预测效果却变得很差。因采用了比较好的学习算法并对算法进行了改进,网络学习的次数不宜太多或太少,一般在10000~30000次之间比较理想。

(2) 由于神经网络模型既可以把所有可能相关的因素都引入模型,同时还可以对模型中输入指标的相关性作出分析,从而剔除那些相关性不好的因素,保留所有真正起作用的指标。因此,在模型中可以方便地引入各种影响预测对象的因素,例如第一产业的人口、土地面积、投资等因素以及第一产业的三组历史数据;第二产业的从业人口数、固定资产投资等因素及第二产业的三组历史数据;第三产业的从业人口数、投资、人均消费水平等因素及第三产业的三组历史数据。此外,正是因为神经网络模型在选择输入指标上具有较大的随意性,所以模型吸收了时间序列分析和回归分析的思想,在输入数据中加入了要预测指标的多组历史数据和各种主要的影响因素。总之,这种模型包括了两种常用的预测方法——时间序列分析和回归分析的共同优点。

(3) 预测模型为了尽可能地减少累计的预测误差,同时也充分地考虑到经济发展的波动性,在预测时采用一系列的网络模型对各个产业未来的增加值进行逐年预测,每一年的每个指标的预测分别通过一个独立的网络子系统来实现。而每个独立的网络子系统的输入同时包含了预测值和该地区当年或从前的真实数据。也就是说,在模型中,对某些指标网络学习的是逐年的规律,这些

指标可能是模型中逐年产生的,例如要预测的第一、二、三产业前一年的增加值;也可能是利用其他专用的模型事先预测出来的,例如用宋健的人口模型计算出的该地区今后的总人口数或产业人口数;还可能是今后该地区可能的指标值,例如各产业的投资。而另外一些指标网络学习的是几年滞后的规律,这些指标均是实际的值,包括两组第一、二、三产业的实际值及其他一些相关因素。

(4) 网络中引入了某些预测时可根据情况适当变动的因素,这些因素是今后该地区可能变动的受一些非经济因素影响很大的指标,例如各产业的投资。需要解释的是,这样做并不会使预测的结果没有意义。相反,这样做的好处是很明显的。事实上,在现实生活中,当我们对某个事物做出预测时,我们总会有各种各样的前提或假设。例如:如果今后几年政策条件较好,本地区经济比较稳定,那么……。这里的神经网络模型恰好实现了这种有条件的预测。具体地说,对应今后几年经济政策及各产业发展形式,将可能出现的情况分为三类:发展情况好、发展得一般、发展得不很理想。对这些可能的情况分别列出投资相应的数据,模型就会根据不同的情况做出不同的预测。

(5) BP 网络预测的另一优点是它可以通过检验输入变量的相关性,方便地实现对输入变量的选择,消除最初的模型中不相关的数据。其步骤如下:

- ① 将所有可能相关的数据作为网络输入;
- ② 完成网络的强制训练;
- ③ 逐一对各变量做强制为 0 的检验,从而判断出相关与不相关数据。

9.2 稠油油层热物性参数的计算模型及算法

油层热物性参数是稠油油藏热采工程计算和储层性质、油藏

工程研究等不可缺少的参数。用油层岩心测定热物性参数虽是常用的基本方法,但却受到取心、岩性和测试技术的限制,通常取得的参数数据很少,耗资甚大。在没有取心或无法利用岩心(如松散)测得热物性参数的情况下,利用蒸气吞吐生产数据和油藏地质特征建立计算油层和盖、底层热物性参数的数学模型,利用数值计算和优化方法,可求得稠油层、盖、底层的任一饱和态下受热区域的平均热物性参数。这为确定稠油层岩石热物性参数提供了一种新的途径。所给出的计算模型和软件已在三个油田开发方案中应用,计算结果规律性强,准确度较高,可以满足地质和工程应用的需要。

9.2.1 优化模型

本模型以单机蒸气吞吐采油的生产数据为基础,将描述油藏热采过程的非线性偏微分方程组初边值问题加以简化,对 T. C. Boberg 等人提出的数学模型及其近似解法进行推导、验证和改进。在此基础上,建立用计算产量来拟合实际产量的优化模型;并将求解无约束优化问题的 Hooke — Jeeves 方法加以改进和推广(简称 CHJ 方法),使之能适用于求解这类不可微的优化问题。

1. 基本假设

本模型适用油藏蒸气吞吐采油第一周期的油层和盖、底层的岩石热物性参数计算。假设条件为:

- (1) 油层和盖、底层是均质的各向同性的砂岩和泥岩;
- (2) 在加热半径内,油层受热均匀,在受热半径外,油层温度等于原始地层温度;
- (3) 地层为厚层,注气前油层温度为原始地层温度,热采时井底压力高于饱和蒸气压力;
- (4) 油层的盖层和底层泥岩的热物性相同;
- (5) 蒸气吞吐阶段的注气量、时间、干度、压力、温度和日产

油、水量以及油藏地质和岩石、原油物性等资料均为已知。

2. 数学模型

在上述假设条件下,通过建立描述蒸气吞吐热采过程中油藏区域内油、水、气三相的质量守恒方程,描述油藏饱和流体岩石的能量守恒方程及油藏区域内水与蒸气共存时的相平衡方程等组成的非线性偏微分方程组,并求解该方程组相应的初边值问题,即可求得油藏区域内温度、压力和饱和度等参数的分布,进而可计算出每天的采油量和采水量。由于我国几个油区已有足够多的蒸气吞吐采油井和生产数据资料,对于一口具体的井,油藏结构参数和原油、岩石的常规物性参数及热物性参数等是不难确定的。假定这些参数和我们给出的计算方法都是合乎实际的,则每天的油、水产量的计算值与生产的实际产量应当很接近。因此,可通过调整油层和盖、底层的热物性参数,使之产量计算值与实际值的偏差最小,即将计算油层和盖、底层岩石的热物性参数问题归结为如下的一个特殊的约束优化问题:

$$\begin{cases} \min & f(\mathbf{x}) \\ \text{s. t.} & \mathbf{a} \leq \mathbf{x} \leq \mathbf{b} \\ & x_1 = \rho_{rs}x_5 \\ & x_2 = \rho_{rc}x_6 \end{cases} \quad (2-1)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_6)^T$, $\mathbf{a} = (a_1, a_2, \dots, a_6)^T$, $\mathbf{b} = (b_1, b_2, \dots, b_6)^T$, $x_1 = \lambda_{rs}$, $x_2 = \lambda_{rc}$, $x_3 = \alpha_{rs}$, $x_4 = \alpha_{rc}$, $x_5 = C_{p,rs}$, $x_6 = C_{p,rc}$, 即分别为油层和盖(底)层的热导率($\text{W/m} \cdot ^\circ\text{C}$), 热扩散率(m^2/h), 比热容($\text{J/g} \cdot ^\circ\text{C}$)。 ρ_{rs} 、 ρ_{rc} 分别为油层和盖层的密度(g/cm^3)。下标 rs、rc 分别表示油层、盖层。 a_1, a_2, \dots, a_6 和 b_1, b_2, \dots, b_6 为常数, 可根据油藏的实际情况确定。

$$\text{而} \quad f(\mathbf{x}) = \beta_1 \sum_{t \in T} q_1(t, \mathbf{x}) + \beta_2 \left| \sum_{t \in T} q_2(t, \mathbf{x}) \right|$$

$$\text{其中} \quad 0 \leq \beta_i \leq 1, i = 1, 2, \beta_1 + \beta_2 = 1$$

$$q_1(t, \mathbf{x}) = |q_{\text{oh}}(t, \mathbf{x}) - q_o(t)|, \quad q_2(t, \mathbf{x}) = q_{\text{oh}}(t, \mathbf{x}) - q_o(t)$$

T 为计算的采油时段,可取 30 天、45 天、60 天、75 天等。 $q_{\text{oh}}(t, \mathbf{x})$ 是第 t 天的采油量计算值, $q_o(t)$ 是第 t 天的采油量实际值。据假设条件(油层的盖层和底层泥岩的热物性相同),故盖、底层的热物性参数视为相同。为求解方便,利用(2-1)式中两个等式约束条件可消去两个变量。

用优化方法求解(2-1)式,其最优解即为所要求的油层和盖、底层岩石的热物性参数值。

9.2.2 计算方法

用优化方法求解(2-1)式的困难在于:目标函数 $f(\mathbf{x})$ 不能用 \mathbf{x} 的显式表达,且 $f(\mathbf{x})$ 是非光滑的。因此,数值计算方法分为两部分。

(1) 目标函数 $f(\mathbf{x})$ 的计算

由计算模型(2-1)式知: $f(\mathbf{x})$ 的计算关键在于计算 $q_{\text{oh}}(t, \mathbf{x})$, 而 $q_{\text{oh}}(t, \mathbf{x})$ 需要通过求解描述蒸气吞吐采油过程的非线性偏微分方程组的初边值问题及有关的迭代公式求得。如所熟知:求解这样的初边值问题是困难的,一般是用差分法或有限元法求解,且计算量很大。另一方法是对上述初边值问题加以简化,设法给出一种近似解析解与数值计算相结合的方法。对上述两种思路,我们都进行了探索,取得了很有成效的结果。基于前一种方法,则采用部分参数项隐式的有限差分法和牛顿法求解上述问题;基于后一种方法,则给出了与计算 $q_{\text{oh}}(t, \mathbf{x})$ 有关的一系列迭代计算公式。计算了采油指数比、几何因数、油层加热半径、受热区平均温度、油水比、产出液携带能量、径向及垂向温度校正函数、有效和泄油半径处的压力及蒸气温度下相对渗透率、原油粘度等。因上述计算公式的导出和说明均较复杂,很占篇幅,在此从略。

(2) 优化计算方法

为论述方便,将计算模型(2-1)式表达为更一般的形式:

$$\begin{cases} \min & f(x) \\ \text{s. t.} & a \leq x \leq b \\ & x \in \mathbf{R}^n \end{cases} \quad (2-2)$$

其中 $a, b \in \mathbf{R}^n$ 为常向量,令

$$S = \{x \in \mathbf{R}^n | a \leq x \leq b\}$$

由于目标函数 $f(x)$ 是非光滑的,且需要经过积分和一系列迭代才能求得其值。因此一般的用导数的优化方法不便于应用,所以求解问题以采用直接搜索方法为宜。则将求解无约束优化问题的直接方法——Hooke-Jeeves 方法加以推广,使之能适用于求这类特殊的约束优化问题,称为 CHJ 方法,其计算步骤如下:

(0) 任取初始点 $x^{(0)} \in S$, 计算精度 $\varepsilon > 0, \alpha \in (0, 1), \beta > 0$, 令 $k = 0, j = 1$;

(1) 令 $x^{(2)} = x^{(0)}, k = k + 1$, 计算 $f(x^{(2)})$, 若 $k \in (4, 10)$, 转(8); 否则令 $c = x^{(0)} - a, d = b - x^{(0)}$, 转(2);

(2) 令

$$x_i^{(1)} = \begin{cases} x_i^{(0)} + \alpha d_i, & \text{当 } i = j \text{ 时,} \\ x_i^{(0)}, & \text{当 } i \neq j \text{ 时,} \end{cases} \quad i = 1, 2, \dots, n$$

计算 $f(x^{(1)})$, 若 $f(x^{(1)}) < f(x^{(2)})$, 令 $x^{(0)} = x^{(1)}$, 转(3); 否则, 令

$$x_i^{(1)} = \begin{cases} x_i^{(0)} - \alpha c_i, & \text{当 } i = j \text{ 时,} \\ x_i^{(0)}, & \text{当 } i \neq j \text{ 时,} \end{cases} \quad i = 1, 2, \dots, n$$

若 $f(x^{(1)}) \geq f(x^{(2)})$, 转(3);

(3) 令 $j = j + 1$, 若 $j \leq n$, 返回(2); 否则转(4);

(4) 若 $f(x^{(0)}) \geq f(x^{(2)})$, 转(5); 否则转(6);

(5) 令 $x^{(0)} = x^{(2)}, c = c/2, d = d/2, j = 1$, 返回(2);

(6) 若 $\|x^{(2)} - x^{(0)}\| < \varepsilon$, 取 $x^* = \frac{1}{2}(x^{(0)} + x^{(2)})$, 计算结束;

否则转(7);

(7) 进行模式搜索;

令 $E = x^{(0)} - x^{(2)}, I = 0$

a. 令 $x^{(3)} = x^{(0)} + \beta E$, 搜索一个步长 β , 使 $x^{(3)} \in S$, 令 $I = I + 1$, 转 b;

b. 若 $f(x^{(3)}) < f(x^{(0)})$, 则缩小搜索区域, 规则为:

若 $x_i^{(3)} > x_i^{(2)}$, 则令 $a_i = x_i^{(2)} - 0.25ac_i$,

若 $x_i^{(3)} < x_i^{(2)}$, 则令 $b_i = x_i^{(2)} + 0.25ad_i, i = 1, 2, \dots, n$

若 $x_i^{(3)} = x_i^{(2)}$, 则令 $\begin{cases} a_i = x_i^{(2)} - 0.25ac_i, \\ b_i = x_i^{(2)} + 0.25ad_i, \end{cases} i = 1, 2, \dots, n$

令 $x^{(0)} = x^{(3)}$, 返回(1); 否则, 若 $f(x^{(3)}) \geq f(x^{(0)})$, 且 $I = 1$, 则令 $E = -E$, 返回 a; 若 $f(x^{(3)}) \geq f(x^{(0)})$, 且 $I > 1$, 返回(1);

(8) 令 $c = (1 - k/15)(x^{(0)} - a), d = (1 - k/15)(b - x^{(0)})$, 返回(2)。

在上述算法中, 参数 α, β 可根据情况选定(如可取 $\alpha = 0.618, \beta = 0.5$)。(7) 中的常数 0.25 也可修改为 $r \in (0, 1)$ 。(8) 中的常数 15 也可酌情增大或缩小。计算精度 ε 可根据实际要求确定, 一般可取 $10^{-2} \sim 10^{-4}$ 。

9.2.3 输入数据

在建立数学模型, 给出计算方法, 研制出性能良好的计算软件之后, 还必须有齐全准确的输入数据。因为输入数据的代表性和准确性是计算结果可靠和实用程度的关键。根据计算模型需要, 系统地收集整理了国内外的有关资料, 特别是对研究区内的蒸气吞吐采油生产数据和油藏地质静态资料等更要认真筛选。

1. 动态资料

(1) 计算井的选定

选择确有代表性的蒸气吞吐生产井做计算井是选准输入数据的基础, 对建立模型、计算结果均有重要意义, 为此特定以下原则:

- ① 选择生产阶段完整(有冷采期),地质、工艺资料齐全;
- ② 井况良好,吞吐周期长,生产正常,即无长期关井、修井;
- ③ 注气、生产数据齐全准确,尽可能有岩心和原油分析资料;
- ④ 被选用井所处的位置有代表性。

按上述原则,在三个油田六套稠油层组,近千口蒸气吞吐井中选择 18 口井作为计算井。

(2) 生产数据选择

根据数学模型计算需要,将全部要计算的井层的井身结构、冷热采阶段产油量、油气比、注气质量以及经研究确定的泄油半径、表皮系数、蒸气温度下地层油粘度和相对渗透率等 49 项基础数据编制成两张综合数据表。对一些尚不能直接整理、测得的参数须经过综合分析研究确定,诸如:

① 表皮(污阻)系数 由于稠油层的特殊性,当时我们不能用测压和岩心分析法求得。因此,根据油藏地质特征、油井污染程度,并参考国内外文献资料,用下述方法给出各计算层的表皮(污阻)系数(s)。即首先综合研究计算井的钻采工艺及热注前后产量增长幅度等资料,把计算井投产前污染程度分为:未污染 $s = 0$,轻度污染 $s = 1 \sim 5$,中度污染 $s = 5 \sim 20$,较重污染 $s = 20 \sim 50$ 及严重污染 $s > 50$;其次是给出热采前即冷采时污阻系数 s_e ;再确定各计算井蒸气吞吐(激励)后消除污染程度(部分消除污染系数 $x = 0.04$,大部分消除污染系数 $x = 0.06$)。经上述步骤后按下式计算热采时的污阻系数 s_h 而 $s_h = s_e - x \cdot s_e$ 。这样给出的污阻系数较为符合油层实际。因为本研究区注气后油井产油量增加数十倍乃至百倍,主要是原油粘度降低,极大地增加流动性能,其次是因为注气而改善了油层污染状况。研究区的稠油沥青含量低,井底附近温度不太高,因此原油中的沥青不至于产生沉淀堵塞油层而造成新的污染。

② 蒸气温度下的相对渗透率 在不能实测相对渗透率的情

况下,采用德士古和雪弗龙东方石油公司合作编制的《高升油田热采可行性研究》报告中给出的 $60^{\circ}\text{C} \sim 260^{\circ}\text{C}$ 的相对渗透率特征值,再利用二次插值方法求出计算层任一温度下的油、水相对渗透率特征值。

③ 蒸气温度下的地层原油粘度 采用计算区脱气原油物性参数,用稠油和特稠油粘度计算公式试算对比并与高升、曙光等油田稠油和特稠油粘度曲线作参照,最后选用计算公式计算各层不同温度下的地层原油粘度。

2. 静态数据

收集整理计算区内全部岩心的物性、岩矿分析和电测解释资料以及分析泥岩的孔隙度、渗透率和饱和度资料计 57 口井近万个数据。再从中选择取心井段长、选样数量多等具有代表性强的井、层资料,经统计选用 20 项所需参数,以供数值计算需要。

9.2.4 计算软件及结果分析

按照前面给出的计算模型和算法,研制了专用的计算软件包 CHJM,它采用模块结构,菜单选择及某些智能化处理,包括一个主控程序和 19 个子程序及一些辅助程序。全部程序用 FORTRAN—77 语言编写,具有计算、绘图、灵敏度分析及某些参数自调等功能,使用灵活方便。可同时求出热采期间油层砂岩和盖、底层泥岩的热物性参数及拟合误差。

应用上述软件计算了辽河油田高升、曙光和欢喜岭三个油田五个稠油区六套油层 18 口井的油层和盖、底层的比热容、热导率和热扩散率。这些参数符合热物性参数变化规律和油藏地质特征。与实测数据比较符合,且计算精度较高,一般比热容的偏差在 $\pm 1.5\% \sim \pm 2.3\%$,热导率在 $\pm 2.5\% \sim \pm 5\%$,完全可以满足热采工艺设计、油藏开发方案和储层研究工作的需要,已在辽河油田应用。

最后需要指出:稠油油层热物性参数的计算问题实质上是一个非线性偏微分方程组初边值问题的系数反演问题。这是一个难度很大、十分繁杂的问题,它在计算和理论方面还有许多问题尚待解决。本节所给的是针对辽河油田的实际情况,计算稠油层热物性参数的一种可行的方案。进一步结果将在以后的论文中给出。

9.3 一类水火电系统的最优开发模型

本节研究一类水火电系统的最优开发规划问题,以黄河上游龙-青段梯级水电站群开发规划研究为背景,建立了这类问题的一个非线性混合整数规划模型,根据西北地区电力系统的实际资料,求解该问题,可优选出今后二十年内,该电力系统中计划建设的新电站的顺序,各电站装机容量的大小及与水电站相应的水库的特征参数。

9.3.1 假设和问题

电力是能源的重要组成部分,在国家建设中起着重要的作用。因此,电力系统中长期规划的研究一直为国内外许多学者所注意。

黄河上游水能资源丰富,是我国水电能源的富矿。梯级水电站的开发利用,对我国的四化建设和国计民生关系重大。将系统工程、运筹优化方法用于梯级水电站开发规划的研究,不仅可以提高工程效率,节约人力和时间,为工程人员和各级领导提供优化的规划(设计)方案,并且有可能产生巨大的经济效益。本节以黄河上游龙羊峡-青铜峡段梯级水电站群的开发规划研究为背景,考虑到西北地区的电力及其发展情况,从总体最优性出发,研究、建立了这一类问题的数学模型,即文中的模型(P)。

为了使模型比较符合实际,又便于求解,我们作如下假设。

1. 所研究的电力系统的电源仅由水电站和火电厂组成。所考

虑的规划期为 T 年。将此规划期分为 TE 段。每一段称为一个投资时段,每个时段长为 PE 年。规定待建的电站只能在某一时段初开工,且在这时段内建成,即每座电站的工期都不超过一个时段。因为可把一个电站(或水库)分成几个子项目,它们可分期建设。每个子项目的工期自然较短。这里,限于说明模型的原理和结构,把电站视为仅由一个子项目所组成。

2. 不同的工程项目其经济运行期限各不相同。对于中长期规划,在此期间所建电站,其正常经济寿命终结年一般都在规划期之外。因此,假定无论哪个电站在规划期的哪个时段开始建设,其经济寿命的终结年都在这一规划期之外。暂不考虑规划中待建电站投入运行后的折旧因素。对于在规划期之前已建成或正建的电站,除考虑它们在电力、电量和经济上与系统的联系外,还可考虑其折旧因素,其输出结果作为本模型输入数据的一部分。

3. 水电站的建设顺序和规模在很大程度上受相应水库的规模和各种用途的影响。因此,本文模型将包括水库特征参数的选择。在模型中将水电站与其相应水库在建设顺序上视为同期的。

4. 系统中已有 NR 座电站,在未来规划中欲建 NW 座水电站和 NF 座火电站,水电站编号从 1 到 NW ,火电站编号从 $NW + 1$ 到 $NW + NF$ 。

所以我们的问题可概述为:某电力系统计划在未来一段时间内,兴建一批电站和水库来满足该时段内系统对电力和综合利用的需求。已知:计划期内系统的电力、电量需要增长的预测资料;系统中已建成和正建电站的特征资料,包括各电站装机容量大小、功率因素和折旧率等;计划待建电站和水库的各种有关的技术经济资料,如水库坝高-保证出力关系曲线、目标函数中的各种费用关系等。

我们的任务是:制订某(西北)地区电力系统的长期(1986 ~ 2005 年)水电电源的发展规划。从系统总体最优性出发,考虑电力

系统中水火电的合理结构,考虑到可能的联网(对华北送电)对电力需要的影响,优选水火电电源,确定规划期内电站的装机顺序及各电站的装机规模,以及与水电站相应的水库的特征参数。

9.3.2 数学模型

1. 目标函数

衡量方案最优性的标准有:① 该方案各工程项目在计算期内总折算费用最小;② 该方案各工程项目在计算期内折算总收益最大;③ 该方案各工程项目的可靠性最高。其他的一些最优性标准多是以上三种标准的变形。对于一个系统来说,其规划方案的最优性标准也是如此。工程项目的计算期是指各可行方案从有支出的第一年算起到转入正常运行后任一年为止的一段时间间隔。本模型选为前述的规划期。计算期内方案的总折算费用是指满足系统发展需要的可行方案中各工程项目在计算期内的投资和运行费等都折算成等效作用于某一选定的基准年的一次性总投资。据文献 83,我们选计算期内方案的总折算费用最小为最优性准则。

总费用包括:各水库、各电站的建设投资和相应的运行费,附属工程的投资和运行费等。

(1) 水库的建设投资,它取决于其工程量的大小,而工程量的大小又主要依赖于坝高,因此,各水库的建设投资可视为其坝高的函数,据文献 82 的分析,比较简单又基本上符合实际的是采用如下的二次关系:

$$\begin{aligned} CRC_{jt} &= \alpha R_j^{(1)} H_{jt}^2 + \alpha R_j^{(2)} H_{jt} \\ j &= 1, 2, \dots, NW; \quad t = 1, 2, \dots, TE \end{aligned} \quad (3-1)$$

(2) 水电站建设投资,它主要包括厂房建设投资和设备安装,可由其装机容量大小来估算。

$$\begin{aligned} CPC_{jt} &= \alpha P_j^{(1)} x_{jt}^2 + \alpha P_j^{(2)} x_{jt} \\ j &= 1, 2, \dots, NW; \quad t = 1, 2, \dots, TE \end{aligned} \quad (3-2)$$

(3) 水库淹没损失,可以用水库的搬迁高程来估算,通常取20%,坝高为搬迁高程,因此,可将水库淹没损失表示为坝高的二次函数:

$$CRF_{jt} = \alpha F_j^{(1)} HH_{jt}^2 + \alpha F_j^{(2)} HH_{jt} \\ j = 1, 2, \dots, NW; \quad t = 1, 2, \dots, TE \quad (3-3)$$

其中

$$HH_{jt} = \sum_{\tau=1}^t H_{j\tau}$$

(4) 火电站建设投资与水电站类似,有

$$CPC_{jt} = CP_j^{(1)} x_{jt}^2 + CP_j^{(2)} x_{jt} \\ j = NW + 1, \dots, NW + NF; \quad t = 1, 2, \dots, TE \quad (3-4)$$

(5) 水库运行费,它受许多因素影响,而且组成复杂。包括:大修费,经常维修费,工资,材料费和库用电费等。在规划阶段,据文献82,83,可采用:水库第 d 年的运行费与到第 d 年该水库的总投资成正比:

$$CRP_{jd} = f \sum_{t=1}^M CRC_{jt} \quad (3-5)$$

其中: M 为正整数; f 为运行费率,可参照文献83确定;将 d 表示为 $d = M'PE + \gamma, 0 \leq \gamma < PE$,则当 $\gamma = 0$ 时, $M = M'$,当 $\gamma \neq 0$ 时, $M = M' + 1$ 。

(6) 水电站运行费,与上述类似,有

$$CPP_{jd} = \alpha W \sum_{t=1}^M CPC_{jt} \\ j = 1, 2, \dots, NW; d = 1, 2, \dots, T \quad (3-6)$$

其中 αW 为运行费率, M 可用上述方法确定。

(7) 火电站运行费,其计算比较复杂,在规划阶段,可用该电站到第 d 年的建设投资和它的年发电量来估算:

$$CPP_{jd} = \alpha F \sum_{t=1}^M CPC_{jt} + GF \cdot W_{jd},$$

$$j = NW + 1, \dots, NW + NF; d = 1, 2, \dots, T \quad (3-7)$$

其中: αF 为运行费率; GF 为与电能 — 燃料转换因子和燃料价格有关的系数。

上面列出了各种费用的计算公式, 但没有明显地提到输、变电费用, 因为它可以包括在电站的建设投资和运行费中。

将上述七项费用求和即为所求的目标函数。

$$C = \sum_{j=1}^{NW} \sum_{t=1}^{TE} (CRC_{jt} + CRF_{jt} \cdot PE) + \sum_{j=1}^{NW} \sum_{d=1}^T CRP_{jd} + \sum_{j=1}^{NW+NF} \left(\sum_{t=1}^{TE} CPC_{jt} + \sum_{d=1}^T CPP_{jd} \right) \quad (3-8)$$

将(3-1) ~ (3-7) 式代入(3-8) 式, 经整理后可得

$$C = \sum_{j=1}^{NW+NF} \sum_{t=2}^{TE} (A_j^{(1)} x_{jt} + A_j^{(2)} x_{jt}^2) + \sum_{j=1}^{NW} \sum_{t=1}^{TE} (B_j^{(1)} H_{jt} + B_j^{(2)} H_{jt}^2) + \sum_{j=1}^{NW} \sum_{t=1}^{TE} (D_j^{(1)} HH_{jt} + D_j^{(2)} HH_{jt}^2) + \sum_{j=NW+1}^{NW+NF} \sum_{d=1}^T C_j W_{jd} \quad (3-9)$$

其中: $A_j^{(1)}, A_j^{(2)}, B_j^{(1)}, B_j^{(2)}, C_j, D_j^{(1)}, D_j^{(2)}$ 是经过整理后形成的系数。由于(3-1) ~ (3-7) 式均为折算费用, 所以(3-9) 中的 C 为总折算费用, 基准年选为规划期的第一年。

2. 约束条件

(1) 装机容量的技术限制。根据各种技术条件的分析可给出每个电站装机容量的上限为 \bar{x}_j 和下限 \underline{x}_j 。为了刻画各子项目只能在一个投资时段内建造及建造顺序, 引入一组 0—1 变量 y_{jt} 。所以有

$$y_{jt} \underline{x}_j \leq x_{jt} \leq y_{jt} \bar{x}_j$$

$$j = 1, 2, \dots, NW + NF; \quad t = 1, 2, \dots, TE \quad (3-10)$$

(2) 电力系统的总容量约束。为了满足电力系统的要求,每个投资时段内所投入的装机容量应使得得到该时段为止,所投入的装机容量的总量与原有电站所提供的容量 $OLDX_t$ 之和不小于该时段系统所需容量的最大值 QN_t ,即

$$OLDX_t \sum_{j=1}^{NW+NF} \sum_{\tau=1}^t x_{j\tau} \geq QN_t, t = 1, 2, \dots, TE \quad (3-11)$$

(3) 电力、电量的相容性约束。设第 d 年相应的投资时段为 t 。若第 t 时段前没有建第 j 座电站,则第 t 时段内第 j 座电站的年发电量为零;若已建,则该电站的年发电量应当与其装机容量的大小相耦合。为此,引入一组 $0-1$ 变量 z_{jd} ,且有

$$\begin{aligned} \beta_j y_{ej} z_{jd} \sum_{\tau=1}^M x_{j\tau} &\leq W_{jd} \leq y_{ej} z_{jd} \sum_{\tau=1}^M x_{j\tau} \\ j &= 1, 2, \dots, NW + NF; d = 1, 2, \dots, T \end{aligned} \quad (3-12)$$

其中, y_{ej} 为第 j 座电站的年利用小时数; β_j 为该电站的利用率系数; M 为正整数,可用与目标函数中所讲的方法确定。

(4) 系统的电量约束,每一年各电站的发电量的总和应不小于该年系统所需要的总电量:

$$\sum_{j=1}^{NW+NF} W_{jd} + OLDW_d \geq QE_d, d = 1, 2, \dots, T \quad (3-13)$$

其中, $OLDW_d$ 为原有电站所能提供的总年发电量。

(5) 水电站装机容量与其相应水库坝高的限制条件。水电站装机容量不能超过其相应水库所能提供的最大保证出力 EW_{jt} ,即应有

$$\begin{aligned} \epsilon_j EW_{jt} &\leq x_{jt} \leq EW_{jt} \\ j &= 1, 2, \dots, NW; \quad t = 1, 2, \dots, TE \end{aligned} \quad (3-14)$$

而

$$EW_{jt} = EW_j(HH_{jt}) \quad (3-15)$$

(3-15) 式一般是非线性的。由于它具有变量可分离性,可通过折线逼近化为线性关系,所以这里可取为

$$EW_{jt} = \delta_j \cdot HH_{jt} \quad (3-16)$$

又

$$\underline{H}_j \cdot y_{jt} \leq H_{jt} \leq \bar{H}_j \cdot y_{jt} \quad (3-17)$$

其中 $\underline{H}_j, \bar{H}_j$ 可通过对该水库的地形、地质情况的分析、径流调节计算和水库的综合利用要求等确定。

(6) 顺序相容性约束

$$\sum_{t=1}^{TE} y_{jt} = 1 \quad (3-18)$$

(每座电站仅能在某一个投资时段内建设)

$$\sum_{j=1}^{NW+NF} y_{jt} \leq 2 \quad (3-19)$$

每一个投资时段 t 内同时建设的电站数目至多为 M 座, 此处, 取 $M = 2$ 。

综合上述可得如下的数学模型(P):

$$\begin{aligned} \min \quad C = & \sum_{j=1}^{NW+NF} \sum_{t=1}^{TE} (A_j^{(1)} x_{jt} + A_j^{(2)} x_{jt}^2) + \\ & \sum_{j=1}^{NW} \sum_{t=1}^{TE} (B_j^{(1)} H_{jt} + B_j^{(2)} H_{jt}^2) + \\ & \sum_{j=1}^{NW} \sum_{t=1}^{TE} (D_j^{(1)} HH_{jt} + D_j^{(2)} HH_{jt}^2) + \\ & \sum_{j=NW+1}^{NW+NF} \sum_{d=1}^T C_j W_{jd} \end{aligned}$$

$$\begin{aligned} \text{s. t.} \quad & y_{jt} \cdot \underline{x}_j \leq x_{jt} \leq \bar{x}_j \cdot y_{jt} \\ & j = 1, 2, \dots, TE; t = 1, 2, \dots, TE \end{aligned}$$

$$\sum_{j=1}^{NW+NF} \sum_{\tau=1}^t x_{j\tau} + OLDX_t \geq QN_t, t = 1, 2, \dots, TE$$

$$\beta_j y_{ej} z_{jd} \sum_{\tau=1}^M x_{j\tau} \leq W_{jd} \leq y_{ej} z_{jd} \sum_{\tau=1}^M x_{j\tau}$$

$$j = 1, \dots, NW + NF; d = 1, \dots, T$$

$$\sum_{j=1}^{NW+NF} W_{jd} + OLDW_d \geq QE_d; d = 1, 2, \dots, T$$

$$y_{jt} \cdot \underline{H}_j \leq H_{jt} \leq y_{jt} \cdot \bar{H}_j$$

$$j = 1, 2, \dots, NW; t = 1, 2, \dots, TE$$

$$\varepsilon_j EW_{jt} \leq x_{jt} \leq EW_{jt}$$

$$j = 1, \dots, NW; t = 1, \dots, TE$$

$$EW_{jt} = \delta_j \cdot HH_{jt}$$

$$j = 1, \dots, NW; t = 1, \dots, TE$$

$$\sum_{t=1}^{TE} y_{jt} = 1, j = 1, 2, \dots, NW + NF$$

$$\sum_{j=1}^{NW+NF} y_{jt} \leq 2, t = 1, 2, \dots, TE$$

$$H_{jt}, x_{jt}, W_{jd} \geq 0$$

$$j = 1, 2, \dots, NW + NF; t = 1, 2, \dots, TE;$$

y_{jt}, z_{jd} 为 0-1 变量; $d = 1, 2, \dots, T$

9.3.3 几点说明

1. 符号说明

模型中所用符号分为三大类, 分别代表决策变量、参数变量和下标变量。

决策变量有: x_{jt} 为 j 电站 t 投资时段投入的装机容量; y_{jt} 为装机顺序变量, 仅取 0—1 值; H_{jt} 为反映第 j 个水库坝高及建设顺序的变量; W_{jd} 为 j 电站第 d 年的年发电量。

参数变量有: QN_t, QE_d 分别为该时段(年)系统所需容量的最大值和系统所需总电量; $\alpha R_j^{(1)}, \alpha R_j^{(2)}$ 为 j 水库建设投资 CRC_{jt} 表示式中的回归系数; $\alpha P_j^{(1)}, \alpha P_j^{(2)}$ 为 j 电站建设投资 CPC_{jt} 表示式中的回归系数; $\alpha F_j^{(1)}, \alpha F_j^{(2)}$ 为 j 水库淹没损失 CRF_{jt} 表达式中的回归系数; CRP_{jd} — j 水库第 d 年的运行费; CPP_{jd} — j 电站第 d 年的运行费。

下标变量有： j 为电站的编号； i 为投资时段的编号； d 为运行年度的编号。

2. 关于模型的补充说明

(1) 关于装机容量。模型中没有明显地考虑各电站在系统负荷图上的历时工作位置。因为我们实际上考虑的是一个动态的时间过程，在这一过程中，负荷情况的变化，新电站的不断投入运行，使得电站在负荷图上的工作位置随着这些因素的变化而变化；另一方面，水电站装机容量选择的常规方法与电站的工作位置有密切的关系。如果直接考虑电站的工作位置，将使模型的非线性程度增高，使问题的求解变得非常困难。通过分析，我们发现：常规方法的实质在于充分利用电站的保证电量，因此我们采用如下的处理方法，即用工程人员提供的装机容量的上、下限来代替有关电站工作位置的约束。作为规划阶段的研究，可先用此模型求得装机顺序及装机规模，如有必要，再通过另一模型来选择各电站的最佳工作位置。

(2) 关于负荷预测。本文采用确定性的未来负荷资料作为已知资料输入模型。实际上，也可采用另一预测模型，通过计算来得到所需的资料，这也是一个具有现实意义的研究课题。

(3) 关于最后两组约束的说明。模型中的约束 (3-18) 和 (3-19) 是反映(水)电站的相容性的。(3-18) 式中的等号是指每个电站都必须建设，且仅能在某一投资时段内建设，也可改为 \leq 号，这适用于电站的选择问题。(3-19) 式右端的常数，根据实际情况可以调整。

3. 模型的一些特性

本文的模型具有参数优选和近似模拟双重用途。参数优选，根据实际系统的情况，可优选出以后一段时间(比如 20 年)内电力系统中计划建设的新电站的顺序，各电站装机容量的大小，以及与水电站相应的水库的特征参数；模拟方面可得到以后一段时间内系

统中新建电站的年发电量和年运行费用的近似结果,为生产安排、运行调度提供依据。

从模型(P)可见,这是一个二次目标函数,具有线性约束的混合整数规划问题。对于这类问题,还缺乏有效的求解方法及较实用的计算机软件。

9.4 干线公路网等级结构优化的目标规划模型及算法

9.4.1 引言

公路网等级结构是指公路网中不同技术等级公路的构成。干线公路网不同技术等级公路的各种组合所形成不同的公路网等级结构,对地区的经济发展及路网交通状况将产生不同的影响。干线公路网总里程预测是干线公路网等级结构优化的前提条件。干线公路网等级结构优化是在干线公路网总里程确定之后,按照国民经济发展计划和公路建设发展战略,根据公路网交通需求,建设资金筹措及服务水平的要求等,运用数学规划理论,对各种等级结构方案进行优化,寻求一种近似最优的干线公路网等级结构方案,以此来指导干线公路建设,使干线公路网满足交通需求,推动地区国民经济发展。

本节依据制定中长期公路网规划的需要,建立了一个干线公路网等级结构优化的目标规划模型,同时给出了算法,并将模型与算法应用于某省干线公路网规划之中,取得了较好的结果。

9.4.2 预备知识

1. 多目标线性规划

由于公路网规划涉及因素众多,故需用多目标线性规划方法

来解决公路网等级结构方案优选问题。该问题的多目标线性规划形式为

$$(VP1) \begin{cases} \text{V-min} & F(L) = (f_1(L), f_2(L), \dots, f_p(L))^T \\ & g_i(L) \geq 0 \quad (i = 1, 2, \dots, m) \\ & h_j(L) = 0 \quad (j = 1, 2, \dots, l) \end{cases}$$

其中 $L = (l_1, l_2, \dots, l_n)^T \in \mathbf{R}^n$, $p \geq 2$, f, g, h 均为线性函数; L 为高速公路(l_0)、一级公路(l_1)、二级公路(l_2)、三级公路(l_3)、四级公路(l_4)、等外路(l_5)组成的各等级公路里程指标集; $f_i(L)$ 为第 i 个目标级关于各等级公路里程指标集的目标函数; $g_i(L), h_j(L)$ 为各等级公路里程指标的约束函数。

对于不同的公路网发展需求,目标函数将有不同的表现形式。一般以路网通行能力、建设资金、发展战略、行程时间等构成多级目标。目标优先级的顺序由实际路网建设发展需要确定。

迄今为止,解多目标规划问题的计算方法已有多种,其中不少已取得了富有成效的应用。按照 C. L. Hwang 和 A. S. Masud 根据决策者偏爱信息的表示方式来划分求解多目标最优化方法的原则,可将多目标最优化的基本求解方法分成三大类,即:① 事先评价法(主要有评价函数法和目标规划法);② 事后评价法(有效解方法);③ 事中评价法(交互规划算法)。

2. 目标规划模型

目标规划模型是一类在实际问题中有着广泛应用的特殊的多目标优化模型。与一般多目标优化模型不同的是:这类模型不是对各个目标进行极小化或极大化,而是希望在约束条件的限制下,每一个目标都尽可能地接近于事先给定的各自对应的理想目标值。

在下面的讨论中,将用到一些术语和概念,在此作简要说明。

理想目标(Objective) 用语言的或变量的观点来看,理想目标是反映决策者愿望的一个比较笼统的提法。如:“通行能力最大”,“建设资金最省”等等。

期望值 (Aspiration Level) 期望值是达到理想目标的满意的, 或可接受的一个特定值, 它可以用来度量理想目标达到的程度。

现实目标 (Goal) 配上期望值的理想目标称之为现实目标。

目标规划是比较常用的多目标规划方法。其基本思想是: 对每一个目标函数引进一个期望值(理想值)。但由于种种条件的限制, 这些期望值往往并不能达到, 从而对每一个目标函数再引进正、负偏差变量, 然后对所有的目标函数建立约束方程, 与原来的约束条件一起组成新的约束条件。在这组新的约束条件下, 来寻找偏差最小的方案。由于各个目标的重要程度不同, 还要引入目标的优先等级和权系数。

在此, 记目标函数为 $f_i(x)$ 。这里, $f_i(x)$ 为线性函数, 并规定 b_i 是对应于第 i 个理想目标的期望值。

有三种现实目标函数表达方式: ① $f_i(x) \leq b_i$; ② $f_i(x) \geq b_i$; ③ $f_i(x) = b_i$ 。

这样, 目标函数与约束条件有了同样的形式。不管是哪种目标表达式, 都应加一个负偏差变量, 并减去一个正偏差变量, 使它们转换成目标规划的格式。表 9-1 归纳了这种转换规则。

当每个目标和约束都按上表转换后, 将建立能反映和衡量各种解的目标达成程度关系式。这种关系式或函数称为“达成函数”。另外, 为了满足目标(或约束), 要求有关偏差变量取极小, 在达成函数中应该反映这种要求。

表 9-1

目标或约束类型	目标规划格式	需要极小化的偏差变量
$f_i(x) \leq b_i$	$f_i(x) + \eta_i - \rho_i = b_i$	ρ_i
$f_i(x) \geq b_i$	$f_i(x) + \eta_i - \rho_i = b_i$	η_i
$f_i(x) = b_i$	$f_i(x) + \eta_i - \rho_i = b_i$	$\rho_i + \eta_i$

9.4.3 干线公路网等级结构优化的目标规划模型

1. 模型的结构

I. 目标函数

(1) 公路网建设资金最省

资金是实现公路网规划目标的关键。公路建设耗资巨大,为顺利完成干线公路网建设,需对各规划期的建设资金进行准确估算和统筹安排。公路建设资金主要用于公路新建和公路改扩建。公路新建主要是指新建高速公路、一级路、二级路等高等级公路;公路改扩建主要是指将三级路、四级路和等外路等低等级公路改扩建成二级路。

(2) 最大满足公路网远景交通需求且公路网利用效率最高

公路交通运输是国民经济、社会发展和综合交通运输大系统中的一个子系统。公路网的规模和技术等级结构取决于区域经济的发展和社会对公路交通的需求。公路网流量是描述社会经济的发展对公路网的需求程度的主要量化指标。公路网交通饱和度是反映公路网利用效率的一个指标,它描述了交通容量对交通需求的适应程度。

(3) 公路网通行能力最大

公路网通行能力或称公路网容量,是指公路网中各路段交通容量的加权(里程权)平均值。公路网通行能力是公路网的一项重要指标,是度量公路网疏导车辆能力的物理量。

(4) 公路网技术等级最高

公路网技术等级是综合反映公路网整体技术等级水平的物理量,它是公路网中各技术等级公路的加权(里程权)平均值。

II. 约束条件

(1) 计划期不同技术等级公路里程之和等于预测的总里程;

(2) 计划期不同技术等级公路的里程折算成标准一般二级公路的相应里程之和等于预测的等效总里程;

公路网的通车总里程能从宏观上反映公路网的发展规模,在一定程度上反映公路网的发展水平以及公路网的连通情况。但是由于不同技术等级公路的通行能力不同,所以同一里程长度但不同等级公路所能满足的交通需求不同。为了研究问题方便,且能够对不同区域公路网的发展水平进行比较,需将各种不同技术等级公路的里程按通行能力,折算成标准一般二级公路的相应里程,即等效里程。

(3) 完成国家、省重点工程建设。

2. 模型的数学描述

1. 目标函数

(1) 公路网建设资金最省

$$\min \sum_{j=0}^2 p_j(t)(l_j(t) - l_j(t-1)) + \sum_{j=3}^5 q_j(t)(l_j(t-1) - l_j(t))$$

其中, $l_j(t)$ 为 t 年第 j 等级公路的里程数; $p_j(t)$ 为 t 年新建第 j 等级公路的单位里程造价; $q_j(t)$ 为 t 年第 j 等级公路改建成二级路的单位里程造价。

(2) 最大满足公路网远景交通需求且公路网利用效率最高

$$\min |S_N(t) - 1|$$

其中, $S_N(t) = Q_N(t)/C_N(t)$ 为 t 年的交通饱和度; $Q_N(t)$ 为 t 年的公路网流量预测值; $C_N(t)$ 为 t 年的公路网容量。

(3) 公路网通行能力最大

$$\max C_N(t) = \sum_j C_j(t)P_j(t)$$

其中, $C_j(t)$ 为 t 年公路网中 j 技术等级公路的交通容量(辆/d); $P_j(t)$ 为 t 年公路网中 j 等级公路的里程权, 即 $P_j(t) = l_j(t)/L(t)$ 。

(4) 公路网技术等级最高

$$\max J_N(t) = \sum_j J_j(t) P_j(t)$$

其中, $J_j(t)$ 为 t 年公路段 j 的技术等级。

II. 约束条件

(1) 计划期不同技术等级公路里程之和等于预测的总里程:

$$\sum_j l_j(t) = L(t)$$

(2) 计划期不同技术等级公路的里程折算成标准一般二级公路的相应里程之和等于预测的等效总里程:

$$\sum_j l_j(t) \mu_j = L_{eq}(t)$$

其中, μ_j 为换算系数 $\mu_j = C_j/C_2$, 这里的 C_j, C_2 分别为 j 级公路与二级公路通行能力(辆/日)。

(3) 完成国家、省重点工程建设

$$\alpha_j(t) \leq l_j(t) \leq \beta_j(t), \quad j = 0, 1, 2, \dots, 5$$

其中, $\alpha_j(t), \beta_j(t)$ 分别表示 t 年上级交通部门下达的完成 j 等级公路建设的下限和上限指标。

综合以上目标函数、约束条件以及目标函数优先级顺序, 得到干线公路网等级结构优化的目标规模模型(GP):

$$\min [R_1 \rho_1 + R_2(\eta_2 + \rho_2) + R_3 \eta_3 + R_4 \eta_4 + R_5(\eta_5 + \rho_5)]$$

$$\text{s. t.} \quad \sum_{j=0}^2 p_j(t)(l_j(t) - l_j(t-1)) +$$

$$\sum_{j=3}^5 q_j(t)(l_j(t-1) - l_j(t)) + \eta_1 - \rho_1 = e_1$$

$$S_N(t) + \eta_2 - \rho_2 = e_2 = 1$$

$$\sum_j C_j(t) P_j(t) + \eta_3 - \rho_3 = e_3$$

$$\sum_j J_j(t) P_j(t) + \eta_4 - \rho_4 = e_4$$

$$\sum_j l_j(t) = L(t)$$

$$\sum_j l_j(t) \mu_j + \eta_5 - \rho_5 = L_{\text{eq}}(t)$$

$$0 \leq \alpha_j(t) \leq l_j(t) \leq \beta_j(t) \quad (j = 0, 1, 2, \dots, 5)$$

$$\rho_i, \eta_i \geq 0 \quad (i = 1, 2, \dots, 5)$$

其中, R_i 为第 i 个目标的优先级别; e_i 为第 i 个目标的期望值; ρ_i, η_i 分别为第 i 个目标的正、负偏差变量。

9.4.4 算法及算例

下面给出目标规划模型的一个建模过程及交互式求解算法框架。

Step1. 选择目标函数、约束条件、建立基础的多目标规划模型。

Step2. 在 Step1 基础上生成目标规划模型, 具体步骤为:

- 为每一个理想目标确定期望值, 即转换成现实目标;
- 对每一个现实目标和约束都加上正、负偏差变量;
- 将目标按其重要性划分优先级;
- 建立达成函数。

Step3. 用线性目标规划的多阶段算法求解。

Step4. 判断现行最优解, 若满意, 则停止; 否则, 转 Step5。

Step5. 提出改进方案, 若对原多目标规划模型进行改进, 转 Step1; 若对目标规划模型进行改进(如输入新的目标期望值、重新确定优先等级等), 转 Step2。

算例 由以上目标函数、约束条件以及目标函数优先级顺序, 用目标规划方法对某省干线公路网 2000 年、2005 年、2010 年和 2015 年各规划期的等级结构规划进行优化求解, 得到未来某省干线公路网等级结构规模(见表 9-2, 9-3), 这一结果已被实际部门采用。

表 9-2 某省干线公路网等级结构优化结果(方案一) 单位:km

类别 年份	高速	一级	二级	三级	四级	等外	合 计
2000	801	108	6584	3959	576	186	12214
2005	1939	432	7874	3300	285	0	13830
2010	2276	1032	10023	3216	0	0	16547
2015	2432	1439	14630	2846	0	0	21347

表 9-3 某省干线公路网等级结构优化结果(方案二) 单位:km

类别 年份	高速	一级	二级	三级	四级	等外	合 计
2000	886	104	6814	3800	590	182	12376
2005	2048	215	9202	3273	238	87	15063
2010	2308	1032	12394	3216	0	0	18950
2015	2510	1645	14408	3067	0	0	21630

9.4.5 结束语

由于使用了目标规划法来求解公路网等级结构的多目标优化模型(VP1),得到了较好的优化结果。通过实践,使我们体会到目标规划方法确有明显的优点:它能把决策者的期望与客观可能较好地结合起来,把“硬约束”转化为“弹性约束”,这显示了目标规划模型的灵活性和可塑性。另外,目标规划模型比较适合于人机交互决策。这样可以将用户或决策者的经验,或有关定性分析结论与数学模型结合起来,得到比较实用的优化决策。

9.5 一个多目标动态投入产出优化模型及算法

9.5.1 引言

自美国经济学家 W. Leontief 于本世纪 30 年代首次提出投入产出经济模型以来,该方法得到了广泛应用。其中投入产出技术与最优化技术的结合是重要的研究与应用方向之一。但过去所研究的投入产出优化模型大多局限于单目标模型。

社会经济系统是一个复杂的动态大系统。在可持续发展战略指导下,一个国家和地区中长期发展规划追求的应是自然、人、社会、经济的协调发展,即多目标协调发展。单纯追求某一个目标最大(最小)而不顾及其他目标,将可能导致社会经济的畸形发展。基于这种认识,本节以动态投入产出方程为核心约束,建立了一个多目标优化模型,较好地反映了实际的社会经济系统的发展规律。

与以往一般投入产出模型不同的是,在模型中,我们不是将各部门的产出 $x_i(t)$ ($i = 1, 2, \dots, n$) 作为决策变量,而是将产出的增量 $\Delta x_i(t) = x_i(t+1) - x_i(t)$ 作为决策变量。这一变化使得所建立模型的约束数目减少了 $n \times (T+1)$ 个,其中 n 为部门数, T 为规划期长(年)。从而使问题规模减少。

9.5.2 多目标动态投入产出优化模型

Leontief 离散型动态投入产出模型的基本形式是

$$X(t) = A(t)X(t) + B(t)(X(t+1) - X(t) + Y_c(t)) \quad (5-1)$$

其中, $X(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$ 为第 t 年总产量向量; $Y_c(t) = (y_{c1}(t), y_{c2}(t), \dots, y_{cn}(t))^T$ 为第 t 年的最终净需求向量; $A(t) = (a_{ij}(t))_{n \times n}$ 为直接消耗系数阵; $B(t) = (b_{ij}(t))_{n \times n}$ 为投资系数

阵。

1. 模型的结构

1. 目标函数

(1) 规划期内国内生产总值(GDP) 累计最大

国内生产总值是衡量一个地区经济总量的重要综合指标之一。比较各国或地区间的经济实力常考察这一指标。因此,选择 GDP 最大为目标函数之一。采用规划期内 GDP 累计最大是希望得到规划期内 GDP 均衡平稳增长的解。

(2) 规划期末年第三产业增加值比重尽可能达到预期目标

第三产业水平是衡量现代社会经济发展程度的重要标志。我国由于改革开放前片面推行工业化政策,尤其是盲目发展重制造业,忽视非物质生产部门特别是服务业的发展,导致第三产业发展滞后,产业结构失调。目前,与世界同等水平国家相比,我国第三产业比重偏低。因此调整产业结构的一项重要内容就是要大力发展第三产业。

(3) 尽可能达到综合经济平衡

综合经济平衡是经济系统良性发展的基础。它要求各部门协调发展,供需平衡。模型中,通过在动态投入产出方程中引入正、负偏差变量来描述各部门的失衡状况,并由此控制以实现综合经济平衡这一目标。

(4) 规划期内某时间段 GDP 平均速度尽可能达到预期目标

这一目标可以使我们着重研究某一个时间段的问题。例如,如果以 1992 年的投入产出表为基础开始计算,我们可能更关注的是“九五”计划期,即 1996 年 ~ 2000 年的 GDP 平均发展速度。可通过选定不同的 GDP 平均速度期望进行多方案发展分析。

(5) 能耗最低

(6) 污染最小

能源紧缺和环境恶化是各地区经济发展普遍面临的问题,而

且随着经济的发展,这些矛盾将日益突出,成为经济发展和人民生活 质量提高的主要障碍之一。地区发展规划应在可持续发展战略的 指导下,进行有效的资源配置,控制污染,为这些困扰人类的全球问题的解决做出地区贡献。

II. 约束条件

(1) 动态投入产出平衡约束

该约束类是模型的核心约束。

(2) 积累、消费约束

积累和消费是经济系统中的重要活动。积累率过高,将导致经济过热增长;积累率过低,又会影响扩大再生产。经济发展应保持适当的积累、消费比例。总积累、总消费和净流入之和应不超过当年的国内生产总值。

(3) 资源约束

资源约束不仅包括水、电、煤等自然资源、能源约束,一般还应考虑劳动力、资金等方面的限制。

(4) 非负约束

这一约束是模型本身对决策变量的要求,但后面将会看到,通过适当的变换,可以去掉这类约束。

2. 模型的数学描述

前面给出了多目标动态投入产出优化模型的基本框架,下面用数学语言给予详细描述。

I. 目标函数

(1) 规划期内国内生产总值(GDP) 累计最大

$$\max \sum_{t=1}^T f(t) \quad (5-2)$$

其中, $f(t) = \sum_{i=1}^n x_i(t) - \sum_{i=1}^n \sum_{j=1}^n a_{ij}(t)x_j(t)$ 为第 t 年的 GDP, T 为规划期长; $x_i(t)$ 和 $a_{ij}(t)$ 的含义见(5-1)式的说明。

(2) 规划期末年第三产业增加值比重尽可能达到预期目标

$$\min \left| \sum_{i \in N_3} c_i(T) x_i(T) - r f(T) \right| \quad (5-3)$$

其中, N_3 为第三产业部门指标集; $c_i(t)$ 为第 t 年 i 部门的增加值率, $c_i(t) = 1 - \sum_{j=1}^n a_{ij}(t)$; r 为规划期末年第三产业增加值占 GDP 比重期望。

(3) 尽可能达到综合经济平衡

$$\min \sum_{t=1}^T \sum_{i=1}^n (\eta_{-i}(t) + \eta_{+i}(t)) \quad (5-4)$$

其中, $\eta_{-i}(t)$ 和 $\eta_{+i}(t)$ 分别代表第 t 年 i 部门动态投入产出平衡方程的负、正偏差变量(参见(5-8)(5-9)式)。

(4) 规划期内某时间段 GDP 平均速度尽可能达到预期目标

设这段规划期为 $T_1 \sim T_2$, $T_1 \geq 1$, $T_2 \leq T$, 期望平均速度为 R 。则这一目标可表述为

$$\min |f(T_1 - 1)(1 + R)^{(T_2 - T_1 + 1)} - f(T_2)| \quad (5-5)$$

(5) 能耗最低

$$\min \sum_{t=1}^T \mathbf{E}(t)^T \mathbf{X}(t) \quad (5-6)$$

其中, $\mathbf{E}(t) = (e_1(t), e_2(t), \dots, e_n(t))^T$ 为第 t 年的部门单位产出能耗向量, 如水、电、煤等能耗。

(6) 污染最小

$$\min \sum_{t=1}^T \mathbf{P}(t)^T \mathbf{X}(t) \quad (5-7)$$

其中, $\mathbf{P}(t) = (p_1(t), p_2(t), \dots, p_n(t))^T$ 为第 t 年的部门污染指数向量, 它可以通过万元产出的三废排放量来衡量。

II. 约束条件

(1) 动态投入产出平衡约束

$$\begin{aligned} \mathbf{X}(0) = & \mathbf{A}(0)\mathbf{X}(0) + \mathbf{B}(0)(\mathbf{X}(1) - \mathbf{X}(0)) + \mathbf{Y}_c(0) + \\ & \boldsymbol{\eta}_-(0) - \boldsymbol{\eta}_+(0) \end{aligned} \quad (5-8)$$

$$\mathbf{X}(t) = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}(t)(\mathbf{X}(t+1) - \mathbf{X}(t)) + \mathbf{Y}_c(t) +$$

$$\eta_{-}(t) - \eta_{+}(t) \quad t = 1, \dots, T \quad (5-9)$$

(2) 积累、消费约束

$$\sum_{i=1}^n \sum_{j=1}^n b_{ij}(t)(x_j(t+1) - x_j(t)) \leq f(t) - \sum_{i=1}^n y c_i(t) \quad t = 1, 2, \dots, T \quad (5-10)$$

(3) 资源约束

$$K_i(t)^T X(t) \leq s k_i(t) \quad t = 1, 2, \dots, T+1; i = 1, 2, \dots, m \quad (5-11)$$

其中, $K_i(t) = (k_{i1}(t), k_{i2}(t), \dots, k_{in}(t))^T$ 是第 i 种资源单位产出资源消耗向量, 如水、电、煤、劳动力等资源; $s k_i(t)$ 为该种资源第 t 年的最大可提供量。

(4) 非负约束

$$X(t+1) - X(t) \geq 0, \quad t = 0, 1, \dots, T \quad (5-12)$$

9.5.3 一种线性变换后的模型

上述模型中把各部门的产出 $x_i(t), i = 1, 2, \dots, n; t = 1, 2, \dots, T+1$ 作为决策变量。这时必须有第 II (4) 类约束(5-12) 式。当 n 和 T 较大时, 这一约束数目 $n \times (T+1)$ 很大, 但可以通过引入另一个决策变量来代替 $X(t)$, 即令

$$\Delta X(t) = X(t+1) - X(t), \quad t = 1, 2, \dots, T \quad (5-12')$$

将 $\Delta X(t)$ 视为决策变量。由于线性目标规划的求解自然要求其决策变量大于等于 0, 因此, 可省去约束类(5-12)。这将减小问题规模, 显著提高运算效率。

由(5-12') 式知

$$X(t+1) = X(0) + \Delta X(0) + \Delta X(1) + \dots + \Delta X(t) \quad t = 1, 2, \dots, T \quad (5-3)$$

将 9.5.2 节模型中的 $X(t)$ 用(5-13) 式替代, 并将决策变量移到等号(或不等号) 左面, 已知量移到右面, 则整个模型可变换如下。

I. 目标函数

$$(1) \quad \max \sum_{t=0}^T f_1(t)$$

其中, $f_1(t) = \sum_{t_1=1}^t \mathbf{C}(t_1)^T \Delta \mathbf{X}(t_1 - 1)$ 代表第 t 年较之基年 ($t = 0$) 的增量, 即有 $f_1(t) = f(t) - f(0)$ 。因此上式与 (5-2) 式相比相差一个常数项 $Tf(0)$ 。后面的目标有的也会与先前的相差某个常数项, 不再一一指出。这里 $\mathbf{C}(t) = (c_1(t), c_2(t), \dots, c_n(t))^T$ 为第 t 年的增加值率向量。

$$(2) \quad \min \left| \sum_{i \in N_3} c_i(0)x_i(0) + \sum_{t=1}^T \sum_{i \in N_3} c_i(t)\Delta x_i(t-1) - r(f(0) + f_1(T)) \right|$$

$$(3) \quad \min \sum_{t=0}^T \sum_{i=1}^n (\eta_{-i}(t) + \eta_{+i}(t))$$

$$(4) \quad \min \left| (f_1(T_1 - 1) + f(0))(1 + R)^{(T_2 - T_1 + 1)} - (f_1(T_2) + f(0)) \right|$$

$$(5) \quad \min \sum_{t=1}^T (\mathbf{E}(t)^T \Delta \mathbf{X}(t-1))$$

$$(6) \quad \min \sum_{t=1}^T (\mathbf{P}(t)^T \Delta \mathbf{X}(t-1))$$

II. 约束条件

$$(1) \quad \mathbf{B}(0)\Delta \mathbf{X}(0) + \eta_{-}(0) - \eta_{+}(0)$$

$$= (\mathbf{I} - \mathbf{A}(0))\mathbf{X}(0) - \mathbf{Y}_c(0)$$

$$(\mathbf{A}(t) - \mathbf{I})(\Delta \mathbf{X}(0) + \Delta \mathbf{X}(1) + \dots + \Delta \mathbf{X}(t-1) +$$

$$\mathbf{B}(t)\Delta \mathbf{X}(t) + \eta_{-}(t) - \eta_{+}(t)$$

$$= (\mathbf{I} - \mathbf{A}(t))\mathbf{X}(0) - \mathbf{Y}_c(t)$$

$$t = 1, 2, \dots, T$$

$$(2) \quad - \sum_{t_1=1}^{t-1} \mathbf{C}(t_1)^T \Delta \mathbf{X}(t_1 - 1) + \sum_{i=1}^n \sum_{j=1}^n b_{ij}(t) \Delta x_j(t)$$

$$\leq \mathbf{C}(0)^T \mathbf{X}(0) - \sum_{i=1}^n y c_i(t) \quad t = 1, 2, \dots, T$$

$$\begin{aligned}
 (3) \quad & K_i(t)^T(\Delta X(0) + \Delta X(1) + \cdots + \Delta X(t-1)) \\
 & \leq sk_i(t) - K_i(0)^T X(0) \\
 & t = 1, 2, \dots, T+1; i = 1, 2, \dots, m
 \end{aligned}$$

至此,我们建立了地区多目标投入产出优化模型。

9.5.4 多目标动态投入产出优化模型的求解

上述多目标动态投入产出优化模型可用(广义)目标规划法求解,此时需要给出目标函数的权系数和各目标的期望值。另外,如何给定合理的外生变量 $Y_c(t)$,以及直接消耗系数阵 $A(t)$ 和投资系数阵 $B(t)$ 的变化规律也是该模型能否提供较好决策选择的关键。

综上所述,我们以动态投入产出方程组为核心约束建立了多目标动态投入产出优化模型。较之单目标模型,它更加准确、多方位地描述了经济系统的运行规律,体现了自然、人、社会、经济协调发展的原则。

动态投入产出模型常出现计算的不稳定性。采用目标规划的多目标优化模型使动态投入产出优化模型的解的不稳定性得到很大改善,并且得到的是综合经济平衡的解;多目标优化模型更适合于人机交互决策的实现。

下面给出多目标动态投入产出的优化模型的一个交互式求解算法的框架。

第一步 将上述模型转换为目标规划模型,其步骤为:

- (1) 为每个理想目标确定期望值;
- (2) 对每个现实目标添加正负偏差变量;
- (3) 根据实际情况和需要,对约束添加正负偏差变量;
- (4) 将各个目标按其重要性划分优先级,确定权系数;
- (5) 建立目标达成函数。

第二步 用线性目标规划的多阶段算法求解上面得到的目标规划模型。

第三步 判断现行解是否为最优解,若满意,则输出计算结果,计算结束;否则转第四步。

第四步 对目标规划模型进行改进,例如修改目标期望值,调查各个目标的优先等级及权系数等,然后返回第二步,继续进行迭代计算。若经过多次修改,仍得不到令人满意的解,可考虑对原来建立的多目标动态投入产出优化模型进行修改,例如对直接消耗系数和投资系数及其变化规律进行调整。然后再返回第一步。

下面给出一个算例。

算例 我们以1992年为基年,建立了某地区7个部门的一个多目标动态投入产出优化模型,并进行了实际预测。下面给出其中的一种方案的预测结果(1993年、1994年的结果与实际经济发展情况基本相同,略)(表9-4)。

表 9-4 某地区 7 个部门发展预测 单位:亿元

年份	指标	农业	轻工业	重工业	建筑业	运邮业	商业	非物质部门
1992	增加值	32.15	30.85	78.62	14.81	28.52	23.27	62.31
1995	增加值 增长率	40.66 15.00%	56.08 16.38%	137.12 10.07%	17.27 10.59%	45.58 20.60%	41.83 11.87%	100.11 10.87%
1996	增加值 增长率	43.51 7.00%	67.76 20.83%	145.39 6.03%	19.12 10.72%	57.93 27.10%	46.29 10.66%	110.71 10.59%
1997	增加值 增长率	46.55 7.00%	75.19 10.97%	154.83 6.49%	21.21 10.93%	69.85 20.58%	51.18 10.57%	121.13 9.42%
1998	增加值 增长率	49.81 7.00%	80.56 7.14%	165.52 6.91%	23.55 11.03%	81.14 16.17%	57.59 12.51%	132.29 9.21%
1999	增加值 增长率	53.30 7.00%	86.48 7.36%	176.91 6.88%	26.13 10.93%	88.46 9.02%	69.14 20.06%	145.80 10.21%
2000	增加值 增长率	55.89 4.86%	92.90 7.42%	189.56 7.15%	29.00 11.00%	96.55 9.14%	81.91 18.48%	160.64 10.18%

从预测结果来看,各部门发展比较均衡稳定,一、二、三次产业结构变化合理,由1992年的11.9:45.9:42.2变到2000年的7.9:44.1:48.0。这说明本节建立的模型和给出的方法的合理性和有效性。

部分习题答案或提示

习 题 一

1. 用 5.5m 长的圆钢下料,有如下几种方案:

件数 方 案	零 件	A	B	C	余料(m)
1		1	1	0	0.3
2		1	0	2	0
3		0	1	2	1.0
4		0	2	1	0.1
5		0	0	4	0.7

设按第 i 种方案下料所需圆钢数量为 x_i 根, $i = 1, 2, \dots, 5$; 用 w 表示所用圆钢总根数, 则数学模型为

$$\left\{ \begin{array}{l} \min \quad w = \sum_{i=1}^5 x_i \\ \text{s. t.} \quad x_1 + x_2 \geq 100 \\ \quad \quad x_1 + x_3 + 2x_4 \geq 200 \\ \quad \quad 2x_2 + 2x_3 + x_4 + 4x_5 \geq 400 \\ \quad \quad x_i \geq 0, x_i \in I, i = 1, 2, \dots, 5 \\ \quad \quad I = \{0, 1, 2, \dots\} \end{array} \right.$$

2. 设 x_{ij} 表示第 i 煤矿运往第 j 城市煤的吨数, $i = 1$, 表示甲煤矿, $i = 2$, 表示乙煤矿, $j = 1, 2, 3$ 分别代表 A 、 B 、 C 三个城市, 运输煤的总吨公里

数为 z , 则数学模型为

$$\left\{ \begin{array}{l} \min \quad z = 90x_{11} + 70x_{12} + 100x_{13} + 80x_{21} + 65x_{22} + 80x_{23} \\ \text{s. t.} \quad x_{11} + x_{12} + x_{13} \leq 200 \\ \quad \quad x_{21} + x_{22} + x_{23} \leq 250 \\ \quad \quad x_{11} + x_{21} \geq 100 \\ \quad \quad x_{12} + x_{22} \geq 150 \\ \quad \quad x_{13} + x_{23} \geq 200 \\ \quad \quad x_{ij} \geq 0, i = 1, 2; j = 1, 2, 3 \end{array} \right.$$

$$3. \quad \left\{ \begin{array}{l} \min \quad z = \sum_{i=1}^{20} (c_0 x_i + c_1 y_i) \\ \text{s. t.} \quad y_0 = 0 \\ \quad \quad y_i = y_{i-1} + x_i - b_i \\ \quad \quad 0 \leq y_i \leq V \\ \quad \quad 0 \leq x_i \leq a_i \end{array} \quad i = 1, 2, \dots, 20 \right.$$

6. 设 m 个用户的位置分别为 (x_i, y_i) , $i = 1, 2, \dots, m$, 欲选定的供应中心位置为 (x, y) , 则

$$\min w = \max \{ \sqrt{(x - x_i)^2 + (y - y_i)^2}, i = 1, 2, \dots, m \}$$

$$13. \quad (1) \quad \nabla^2 f(x) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad f(x) \text{ 为严格凸函数};$$

$$(2) \quad \nabla^2 f(x) = \begin{pmatrix} 2 & 2 & 0 \\ 2 & 4 & 4 \\ 0 & 4 & 8 \end{pmatrix}, \quad f(x) \text{ 是凸函数};$$

$$(3) \quad \nabla^2 f(x) = \begin{pmatrix} 2 & -2 & 2 \\ -2 & -6 & -6 \\ 2 & -6 & 0 \end{pmatrix}, \quad f(x) \text{ 不是凸函数}.$$

14. 提示 用数学归纳法证明。

22. (1) 是, (2) 不是。

习 题 二

$$1. (1) \begin{cases} \max & f(\mathbf{x}) = 10x_1 + 4x_2 \\ \text{s. t.} & 3x_1 + 4x_2 + x_3 = 10 \\ & 5x_1 + 2x_2 + x_4 = 8 \\ & x_1 - 2x_2 + x_5 = 3 \\ & x_i \geq 0, i = 1, 2, \dots, 5 \end{cases}$$

$$9. \mathbf{x}^* = (5/3, 0, 2/3)^T, \quad w^* = 7/3$$

$$10. (1) \mathbf{x}^* = (4/3, 11/3)^T, \quad w^* = -29/3$$

$$(2) \mathbf{x}^* = (1, 1.5)^T, \quad z^* = -26.5$$

$$(3) \mathbf{x}^* = (0.5, 3, 0)^T, \quad w^* = -5.5$$

$$(4) \mathbf{x}^* = (3, 0)^T, \quad w^* = 9$$

$$(5) \mathbf{x}^* = (3.5, 0, 0, 0.25, 0.75)^T, \quad z^* = -36.5$$

$$11. (1) \mathbf{x}^* = (0, 2/5, 9/5)^T, \quad w^* = 11/5$$

$$(2) \mathbf{x}^* = (3, 0, 1, 3)^T, \quad w^* = 2$$

$$12. (1) \mathbf{x}^* = (2.5, 2.5, 2.5, 0)^T, \quad z^* = -15$$

$$(2) \mathbf{x}^* = (0.25, 0, 0.75, 0.25)^T, \quad z^* = -4.75$$

$$(3) \mathbf{x}^* = (15, 0, 0, 20, 0, 0, 12)^T, \quad z^* = 60$$

15. 施甲肥 300 公斤, 乙肥 230/3 公斤, 丙肥 45 公斤, 不施丁肥, 最低成本为 280 元。

$$23. (1) \mathbf{x}^* = (4, 0)^T, \quad z^* = 4$$

$$(2) \mathbf{x}^* = (3/5, 6/5)^T, \quad z^* = 12/5$$

习 题 三

$$2. \mathbf{x}^* = 4, \quad f(\mathbf{x}^*) = -156$$

$$3. \mathbf{x}^* = 2, \quad f(\mathbf{x}^*) = 0$$

$$4. (1) \mathbf{x}^* = (5/3, -2/3)^T, \quad Q(\mathbf{x}^*) = -7/3$$

$$6. \mathbf{x}^* = 2, \quad f(\mathbf{x}^*) = -2/3$$

$$7. \mathbf{x}^* = 0.69315, \quad f(\mathbf{x}^*) = -0.38629$$

$$16. \mathbf{x}^* = (2/3, 3/2)^T, f(\mathbf{x}^*) = -5.8333$$

$$18. \mathbf{x}^* = (1, 2)^T, f(\mathbf{x}^*) = -8$$

$$20. \mathbf{x}^* = (1, 1)^T, f(\mathbf{x}^*) = 0$$

$$21. \mathbf{x}^* = (2, 1)^T, f(\mathbf{x}^*) = 0$$

$$23. (1) \mathbf{x}^* = (1, 2)^T, f(\mathbf{x}^*) = -12;$$

$$(3) \mathbf{x}^* = (-4, -3, -1)^T$$

$$27. \mathbf{x}^* = (-0.41936, 0)^T, f^* = 0.77291$$

$$28. \mathbf{x}^* = (1, -1/2)^T, f(\mathbf{x}^*) = -3/4$$

$$29. \mathbf{x}^* = (0, 0)^T, f(\mathbf{x}^*) = 0$$

$$35. (1) \mathbf{x}^* = (0, 0)^T, f(\mathbf{x}^*) = 0;$$

$$(2) \mathbf{x}^* = (5, 6)^T, f(\mathbf{x}^*) = 0;$$

$$(3) \mathbf{x}^* = (1, 1)^T, f(\mathbf{x}^*) = 0.$$

$$39. (1) \mathbf{x}^* = (4, 2)^T, f(\mathbf{x}^*) = -8.$$

习 题 四

$$2. \mathbf{x}^* = (0, -3)^T, u^* = \left(\frac{1}{6}, 0\right)^T$$

$$4. (1) \mathbf{x}^* = \left(\frac{5}{3}, -\frac{2}{3}\right)^T, Q(\mathbf{x}^*) = -\frac{7}{3}$$

$$(2) \mathbf{x}^* = (0.9, 0.1)^T, Q(\mathbf{x})^* = -\frac{21}{20}$$

$$7. (1) \mathbf{x}^* = (1, 0)^T, f(\mathbf{x}^*) = 1$$

$$(2) \mathbf{x}^* = (2/3, \sqrt{3}/3)^T, f(\mathbf{x}^*) = -2\sqrt{3}/9$$

$$8. (1) \mathbf{x}^* = (0, 0)^T, f(\mathbf{x}^*) = 0; (3) \mathbf{x}^* = (1, 0)^T, f(\mathbf{x}^*) = 8/3$$

$$11. \mathbf{x}^* = (0.626, 26.000)^T, A^* = 101.12$$

$$14. \mathbf{x}^* = (0.823, 0.911)^T, f(\mathbf{x}^*) = 1.393$$

$$31. \mathbf{x}^* = (1/2, 1/4, 0, 1/4)^T, f(\mathbf{x}^*) = 1/2$$

$$32. \mathbf{x}^* = (78.000, 33.000, 29.995, 45.000, 36.776)^T$$

$$f(\mathbf{x}^*) = -30665.5$$

$$33. (1) \mathbf{x}^* = \left(\frac{1}{4}, \frac{3}{4}\right)^T \quad (2) \mathbf{x}^* = (1, 2)^T, f(\mathbf{x}^*) = 20$$

$$34. (2) \mathbf{x}^* = (5.27, 3.68)^T, f(\mathbf{x}^*) = -79.9$$

$$35. (1) \mathbf{x}^* = (-1.71719, 1.59575, 1.82721, -0.763621, -0.763622)^T$$

$$f(\mathbf{x}^*) = -2.91963$$

$$(2) \mathbf{x}^* = (3.24190, 0, 1.63415, 0.123980, 0.889636, 1.24020, 2.$$

$$87021)^T$$

$$f(\mathbf{x}^*) = -44.4688$$

习 题 五

1. (1) 设 A, B 两种产品各生产 x_1, x_2 吨, 则数学模型为

$$\begin{cases} \text{V-min} & (4x_1 + x_2, 2x_1 + 5x_2) \\ \text{s. t.} & x_1 + x_2 \geq 7 \\ & x_1 \leq 5, x_2 \leq 6, \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(2) \mathbf{x}^* = (1, 6)^T$$

2. 设 A, B 两种型号的摩托车每周正常生产各为 x_1, x_2 辆, 加班生产各为 y_1, y_2 辆, 则数学模型为

$$\begin{cases} \text{V-min} & [-(100x_1 + 80x_2 + 90y_1 + 70y_2), 3y_1 + 2y_2] \\ \text{s. t.} & 3x_1 + 2x_2 \leq 120 \\ & 3y_1 + 2y_2 \leq 48 \\ & x_1 + y_1 \geq 30 \\ & x_2 + y_2 \geq 30 \\ & x_1, x_2 \geq 0, y_1, y_2 \geq 0, x_i, y_i \in \{0, 1, 2, \dots\}, i = 1, 2 \end{cases}$$

$$10. R_1^* = \{x^* | x^* = 1\}, R_2^* = \{x^* | 3 \leq x^* \leq 4\}, R^* = \emptyset, R_{pa}^* = \{x | 1 \leq x \leq 3\}, R_{wp}^* = \{x | 1 \leq x \leq 4\}$$

$$13. \mathbf{x}^* = (3/2, 15/16)^T$$

习 题 六

$$5. (1) \mathbf{x}^* = (2, 1)^T, z^* = 8$$

$$(2) \mathbf{x}^* = (2, 2)^T, z^* = 4$$

$$(4) \mathbf{x}^* = (3, 1)^T \text{ 和 } (2, 2)^T, z^* = 4$$

$$(5) \mathbf{x}^* = (3, 2.5)^T, z^* = -13$$

$$6. (1) \mathbf{x}^* = (4, 1)^T, z^* = -14$$

- * 9. 提示 (1) 用匈牙利方法求解
(3) 虚拟两个目的地 D 、 E

习 题 七

1. 最短路线 $A \rightarrow B_2 \rightarrow C_1 \rightarrow D_1 \rightarrow E$ 或 $A \rightarrow B_3 \rightarrow C_1 \rightarrow D_1 \rightarrow E$ 或 $A \rightarrow B_3 \rightarrow C_2 \rightarrow D_2 \rightarrow E$
5. 1 组 1 名, 2 组 0 名, 3 组 1 名, 三个小组都失败的概率为 0.06。
7. 所用资金用于购买 B 种牛, 四年末拥有 70 头牛。
9. $x^* = (0, 7)^T$, $z^* = 49$
10. $x^* = (9.6, 0.2)^T$, $z^* = 702.92$
11. 第一方案: (甲, 乙, 丙) = (0, 2, 3),
第二方案: (甲, 乙, 丙) = (2, 2, 1), 总盈利 21 万元。
12. $p^* = \{5, 0, 6, 0\}$ 台, $f^* = 20.5$ 万元

习 题 八

1. (2) $x^* = (1, 1)^T$, $f^* = 0$
3. (1) $x^* = (0, 0, \dots, 0)^T$, $f^* = 0$
6. (1) $x^* = (0, 0, \dots, 0)^T$, $f^* = 0$
(2) $x^* = (0, 0, \dots, 0)^T$, $f^* = 0$
8. (1) $x^* = (1, 1, \dots, 1)^T$, $f^* = 0$
(3) $x^* = (0, 0, \dots, 0)^T$, $f^* = 0$

参 考 文 献

- 1 L. Cooper, U. N. Bhat, L. J. Leblanc. Introduction to operations research models, 1977
- 2 H. P. Williams. Model building in mathematical programming, 1978
- 3 唐焕文等. 数学模型引论. 第二版, 北京: 高等教育出版社, 2001
- 4 P. E. Gill, W. Murray, M. H. Wright. Practical optimization, 1981
- 5 许国志, 马仲蕃. 整数规划初步. 沈阳: 辽宁教育出版社, 1985
- 6 施光燕. 整数规划. 大连理工大学教材, 1979
- 7 A. Kaufmann, et al. Integer and mixed programming. Theory and applications, 1977
- 8 唐焕文. 实用数学规划导论. 大连: 大连工学院出版社, 1986
- 9 马仲蕃, 魏权龄, 赖炎连. 数学规划讲义. 北京: 中国人民大学出版社, 1981
- 10 V. V. Kolbin. Stochastic programming, 1977
- 11 M. S. Bazaraa, C. M. Shetty. Nonlinear programming. Theory and algorithms, 1979
- 12 Демьянов В Х, Васильев Л В. 不可微最优化, 金洪臻等译. 大连: 大连理工大学出版社, 1991
- 13 F. H. Clarke. Optimization and nonsmooth analysis, 1983
- 14 S. I. Gass. Linear programming: Methods and applications. New York: McGraw-Hill, 1985
- 15 G. B. Dantzig. Linear programming and extensions, 1963
- 16 D. G. Luenberger 线性与非线性规划引论. 夏尊铨等译. 北京: 科学出版社, 1980
- 17 张建中等. 线性规划. 北京: 科学出版社, 1990
- 18 管梅谷等. 线性规划. 济南: 山东科技出版社, 1983
- 19 方述诚等. 线性优化及扩展: 理论与算法. 北京: 科学出版社, 1994

- 20 薛嘉庆. 线性规划. 北京: 高等教育出版社, 1989
- 21 杨林锡, 邓成梁, 甘应爱. 数学规划的原理和方法. 武汉: 华中工学院出版社, 1985
- 22 S. P. Bradley, A. C. Hax, T. L. Magnanti. Applied mathematical programming, 1977
- 23 席少霖, 赵凤治. 最优化计算方法. 上海: 上海科技出版社, 1983
- 24 薛嘉庆. 最优化原理与方法. 北京: 冶金工业出版社, 1983
- 25 邓乃扬等著. 无约束最优化计算方法. 北京: 科学出版社, 1982
- 26 R. Fletcher. Practical Methods of Optimization, Vol. 1, 1980, Vol. 2, 1981
- 27 陈开周. 最优化计算方法. 西安: 西北电讯工程学院出版社, 1985
- 28 邓乃扬, 诸梅芳. 最优化方法. 沈阳: 辽宁教育出版社, 1987
- 29 M. Avriel. 非线性规划——分析与方法. 李元熏, 陈开明, 魏国华等译. 上海: 上海科学技术出版社, 上册, 1979, 下册, 1980
- 30 P. E. Gill, W. Murray. Numerical methods for constrained optimization, 1974
- 31 D. F. Bertsekas. Constrained optimization and Lagrange multiplier Methods, 1982
- 32 袁亚湘, 孙文瑜. 最优化理论与方法. 北京: 科学出版社, 1997
- 33 陈开明. 非线性规划. 上海: 复旦大学出版社, 1991
- 34 胡毓达. 非线性规划. 上海: 高等教育出版社, 1991
- 35 席少霖. 非线性最优化方法. 北京: 高等教育出版社, 1992
- 36 赵瑞安, 吴方. 非线性最优化理论和方法. 杭州: 浙江科技出版社, 1992
- 37 袁亚湘. 非线性规划数值方法. 上海: 上海科技出版社, 1993
- 38 赵凤治等. 约束最优化计算方法. 北京: 科学出版社, 1991
- 39 朱道立. 大系统优化理论与应用. 上海: 上海交通大学出版社, 1987
- 40 钱令希. 工程结构优化设计. 北京: 水利电力出版社, 1983
- 41 J. L. Ignizio. 单目标和多目标系统线性规划. 闵仲求等译. 上海: 同济大学出版社, 1986
- 42 胡毓达. 实用多目标最优化. 上海: 上海科技出版社, 1990

- 43 魏权龄. 多目标规划讲义. 中国人民大学印, 1982
- 44 林铨云, 董加礼. 多目标优化的方法与理论. 长春: 吉林教育出版社, 1992
- 45 B. E. Gillett. 运筹学导论——计算机算法. 机械工业出版社, 1982
- 46 D. M. Himmelbau. 实用非线性规划. 张义新等译. 北京: 科学出版社, 1981
- 47 王众托. 系统工程引论. 北京: 电子工业出版社, 1991
- 48 D. P. Bertsekas. Dynamic programming: Deterministic and Stochastic models, 1987, 李人厚, 韩崇昭译. 西安: 西安交通大学出版社, 1990
- 49 卢向华, 侯定丕, 魏权龄. 运筹学教程. 北京: 高等教育出版社, 1991
- 50 李维铮等. 运筹学. 北京: 清华大学出版社, 1982
- 51 邵济煦等. 管理运筹学简明教程. 西安: 西安电子科技大学出版社, 1988
- 52 魏国华等. 实用运筹学. 上海: 复旦大学出版社, 1987
- 53 R. T. Rockafellar. Convex analysis, 1972
- 54 唐焕文等, 数学学报, 1990, 33(4): 521~527
- 55 晏晓焰等. 线性规划问题初始基础可行解的一种求法. 数值计算与计算机应用, 1984, 5(3): 167~177
- 56 何岳山. 线性规划问题初始可行基的一种方法. 应用数学, 1990, 3(3): 88~90
- 57 R. G. Bland. New finite pivoting rules of the simplex method, Math. Oper. Res. 1977, 2: 103~170
- 58 A. Charnes. Optimality and degeneracy in linear programming. Econometrica 1952, 20: 160~170
- 59 越民义. 椭圆法介绍. 运筹学杂志, 1983, 1(1): 1~9
- 60 N. Karmarkar. A new polynomial-time algorithm for linear programming. Combinatorica 1984, 4: 373~395
- 61 吴方. 拟牛顿法. 中科院应用数学所, 1983
- 62 H. Y. Huang. Unified approach to quadratically convergent algorithms for function minimization. J. Optimization theory and applications, 1970, 5: 405~423

- 63 吴方,桂湘云. 一类具有 $n+1$ 个参数的变测度算法. 数学学报, 1981, 24(6): 921~930
- 64 唐焕文, 施光燕. 一个通用的优化软件包 YOUHUA-1A. 计算机应用与软件, 1987, 4(4): 24~30
- 65 D. P. Bertsekas. Multiplier Method: A Survey. Automatica, 1976, 12(2): 133~145
- 66 R. J. Benayoun, et al. Linear programming with multiple objective functions: STFP Method. Math. Programming, 1975, 1(3)
- 67 孙艳丰, 王众托. 遗传算法在优化问题中的应用研究进展. 控制与决策, 1996, 11(4): 425~431
- 68 谢金星. 进化计算简要综述. 控制与决策, 1997, 12(1): 1~7
- 69 李宏, 唐焕文, 郭崇慧. 一类进化策略的收敛性分析. 运筹学学报, 1999, 3(4): 79~83
- 70 彭宏等. 计算工程优化问题的进化策略. 华南理工大学学报, 1997, 25(12): 17~21
- 71 彭宏等. 解约束优化问题的进化策略与混合进化策略的比较. 数值计算与计算机应用, 1998, 19(1): 35~40
- 72 王云诚. 系统优化的若干方法研究: [博士论文]. 大连理工大学, 1999
- 73 郭崇慧, 唐焕文. 演化策略的全局收敛性, 计算数学, 2001, 23(1): 105~110
- 74 席裕庚等. 遗传算法综述. 控制理论与应用, 1996, 13(4): 697~708
- 75 徐宗本, 高勇. 遗传算法过早收敛现象的特征分析及预防. 中国科学(E辑), 1996, 26(4): 365~375
- 76 恽为民. 遗传算法的运行机理分析. 控制理论与应用, 1996, 13(3): 289~297
- 77 刘勇, 康立山, 陈毓屏. 遗传算法, 北京: 科学出版社, 1995
- 78 唐焕文等. 经济指标预测的人工神经网络系统. 武汉: 武汉出版社, 1997
- 79 唐焕文, 秦学志. 计算稠油层岩石热物性参数的数学模型, 模糊系统与数学增刊(运筹学及其应用专辑) 1994, 351~354
- 80 T. C. Boberg 著. 热力采油工程方法. 辽河油田科技处译. 北京: 石油工业出版社, 1980

- 81 赵可培. 目标规划及其应用. 上海: 同济大学出版社, 1989
- 82 D. T. Olayghine, D. M. Himmelblau. Optimal expansion of a water resources system. New York: Academic Press, 1974
- 83 水电部水利水电建设总局, 水力发电工程评价暂行规定, 1983
- 84 刘宝碇, 赵瑞清. 随机规划与模糊规划. 北京: 清华大学出版社, 1998
- 85 那日萨, 唐焕文. 一个多目标动态投入产出优化模型及算法 (I), (II), 系统工程理论与实践, 1998, 18(8): 49~52, 18(9): 91~95
- 86 唐焕文, 张立卫. 凸规划的极大熵方法. 科学通报, 1994, 39(8): 682~684
- 87 王云诚, 张立卫, 唐焕文. 一般约束凸规划极大熵方法的收敛性. 大连理工大学学报, 1995, 35(6): 764~769
- 88 刑文训, 谢金星. 现代优化计算方法[M]. 北京: 清华大学出版社, 1999
- 89 康立山, 谢云, 矢尤勇, 罗祖华, 非数值并行算法——模拟退火算法[M]., 北京: 科学出版社, 1994
- 90 胡山鹰, 陈丙珍, 何小荣. 模拟退火法用于连续变量问题全局优化初探[J]. 计算机与应用化学, 1996, 13(1): 7-13
- 91 胡山鹰, 陈丙珍, 何小荣, 沈静珠. 非线性规划问题全局优化的模拟退火法[J], 清华大学学报, 1997, 37(6): 5-9
- 92 杨若黎, 顾基发. 一种高效的模拟退火全局优化算法[J]. 系统工程理论与实践, 1997, 17(5): 29-35
- 93 靳利霞, 唐焕文. 蛋白质空间结构预测的一种优化模型及算法[J]. 应用数学与计算数学学报, 2000, 14(2): 33-41
- 94 A. Liwo, M. R Pincus, R. J. Wawak, S. Rackovsky, H. A. Scheraga. Prediction of protein conformation on the basis of a search for compact structures: thest on avian pancreatic polypeptide[J]. Protein Science, 1993, 2: 1715-1731
- 95 M. R Liwo, M. R. Pincus, R. J. Wawak, S. Rackovsky, H. A. Scheraga. Calculation of protein backbone geometry from α carbon coordinates based on peptide-group dipole alignment[J]. Protein Science, 1993, 2: 1697-1714
- 96 郭崇慧, 唐焕文, 一种改进的进化规划算法及其收敛性, 高校计算数学学报, 2002, 24(1): 51~56

附录 最优化方法中常用的 数学基础知识汇编

一、矢量和矩阵

矢 量

n 个数 x_1, x_2, \dots, x_n 所组成的有序数组, 称为一个 n 维矢量 \boldsymbol{x} , x_i 称为矢量 \boldsymbol{x} 的第 i 个分量(或元素)。记号 \boldsymbol{x} 代表列矢量, 记号 \boldsymbol{x}^T 代表行矢量, 即 $\boldsymbol{x}^T = (x_1, x_2, \dots, x_n)$ 。矢量亦称为向量, 用小写粗体字母表示, 如 $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}$ 等。所有 n 维矢量构成的集合用 E^n 表示。

零矢量 所有分量均为零的矢量, 称为零矢量, 用 $\mathbf{0}$ 表示。

坐标矢量 第 i 个分量为 1, 其余分量均为零的矢量, 称为第 i 个坐标矢量, 或第 i 个单位矢量, 用 \boldsymbol{e}_i 表示, 即 $\boldsymbol{e}_i = (0, 0, \dots, 0, 1, 0, \dots, 0)^T$ 。

矢量加法与数乘矢量

设 $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^T$, $\boldsymbol{y} = (y_1, y_2, \dots, y_n)^T$, 则

$$\boldsymbol{x} + \boldsymbol{y} = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n)^T$$

$$a\boldsymbol{x} = (ax_1, ax_2, \dots, ax_n)^T$$

其中 a 为数。

线性相关与线性无关

设矢量 $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(k)} \in E^n$, 如果存在一组不全为零的常

数 $\lambda_1, \lambda_2, \dots, \lambda_k$, 使 $\sum_{j=1}^k \lambda_j \mathbf{x}^{(j)} = \mathbf{0}$, 则称矢量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 线性相关。否则称它们为线性无关, 或线性独立。

线性组合 若 $\mathbf{y} = \sum_{j=1}^k \lambda_j \mathbf{x}^{(j)}$, 则称矢量 \mathbf{y} 为矢量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 的线性组合。

基 用 E^n 表示 n 维线性空间, 如果 E^n 中的任何矢量都可以表示为 E^n 中的矢量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 的线性组合, 则称矢量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 生成 E^n 。

设矢量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)} \in E^n$, 如果它们生成 E^n , 并且去掉其中任何矢量, 便不能生成 E^n , 则称 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 为 E^n 的一组基。矢量组 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 构成 E^n 的一组基的充要条件是 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$ 线性无关, 并且 $k = n$ 。

矢量的范数 设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 则矢量 \mathbf{x} 的欧氏范数定义为

$$\|\mathbf{x}\| = (\mathbf{x}^T \mathbf{x})^{1/2} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2} \quad (1-1)$$

范数具有下列性质:

- 1° 非负性 $\|\mathbf{x}\| \geq 0$, 且 $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$;
- 2° 齐次性 $\|\alpha \mathbf{x}\| = |\alpha| \|\mathbf{x}\|$, α 为实数;
- 3° 三角不等式 $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ 成立。

内 积

设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$, 则矢量 \mathbf{x} 和 \mathbf{y} 的内积定义为

$$(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} = \sum_{j=1}^n x_j y_j \quad (1-2)$$

若 $(\mathbf{x}, \mathbf{y}) = 0$, 则称两矢量 \mathbf{x}, \mathbf{y} 正交, 记为 $\mathbf{x} \perp \mathbf{y}$ 。

Schwartz 不等式

$$|(x, y)| = |x^T y| \leq \|x\| \|y\| \quad (1-3)$$

平行四边形法则

设 $a, b \in E^n$, 则有

$$\|a + b\|^2 + \|a - b\|^2 = 2(\|a\|^2 + \|b\|^2) \quad (1-4)$$

定理 1.1 设 $x^{(1)}, x^{(2)}, \dots, x^{(k)}$ 为线性空间 S 的一组基, $y \in$

S , 且 $y = \sum_{i=1}^k \lambda_i x^{(i)}$, 若对某个 j , 有 $\lambda_j \neq 0 (1 \leq j \leq k)$, 则用 y 代替 $x^{(j)}$ 所得到的矢量组 $x^{(1)}, x^{(2)}, \dots, x^{(j-1)}, y, x^{(j+1)}, \dots, x^{(k)}$ 仍为 S 的一组基。

矩 阵

$m \times n$ 个数 a_{ij} , 按一定次序排成 m 行、 n 列的表, 称为 $m \times n$ 矩阵, 记为

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

也可简记为 $A = (a_{ij})_{m \times n}$, 其中 a_{ij} 是矩阵 A 中位于第 i 行第 j 列的元素。若 $m = n$, 则 A 称为 n 阶方阵。

特殊矩阵

所有元素均为零的矩阵, 称为零矩阵, 用 O 表示。若 n 阶方阵的元素 $a_{ii} = 1 (i = 1, 2, \dots, n)$, 且当 $i \neq j$ 时, 所有的 $a_{ij} = 0$, 则称此 n 阶方阵为 n 阶单位矩阵, 用 I_n (或 E_n) 表示, 有时简记为 I 。若 n 阶方阵 A 的元素满足 $a_{ij} = a_{ji} (i = 1, 2, \dots, n; j = 1, 2, \dots, n)$, 则称 A 为对称矩阵。若 A 为对称矩阵, 则 $A^T = A$, 其中 A^T 表示 A 的转置矩阵。

矩阵的加减法与数乘矩阵

令 $A = (a_{ij})_{m \times n}$, $B = (b_{ij})_{m \times n}$, 定义

$$A \pm B = (a_{ij} \pm b_{ij})_{m \times n} \quad (1-5)$$

$$\alpha A = (\alpha a_{ij})_{m \times n} \quad (1-6)$$

矩阵的乘法

令 $A = (a_{ij})_{m \times n}$, $B = (b_{ij})_{n \times p}$, 积 AB 定义为 $m \times p$ 矩阵 $C = (c_{ij})_{m \times p}$, 其中

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, p \quad (1-7)$$

分块矩阵

矩阵可划分为子矩阵, 例如 $m \times n$ 矩阵 A 可分块如下:

$$A = (a_{ij})_{m \times n} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

其中 A_{11} 为 $m_1 \times n_1$ 矩阵, A_{12} 为 $m_1 \times n_2$ 矩阵, A_{21} 为 $m_2 \times n_1$ 矩阵, A_{22} 为 $m_2 \times n_2$ 矩阵, 且 $m = m_1 + m_2$, $n = n_1 + n_2$ 。

矩阵的逆阵

设 A 为 n 阶方阵, 用 $\det(A)$ 或 $|A|$ 表示 A 的行列式。若 $|A| \neq 0$, 则称 A 为 n 阶非奇异方阵。如果有 n 阶方阵 B 存在, 使得 $AB = BA = I_n$, 则称 B 为 A 的逆矩阵, 记为 $B = A^{-1}$, 即 $AA^{-1} = A^{-1}A = I_n$ 。若方阵 A 的逆阵存在, 则它是唯一的。

定理 1.2 方阵 A 有逆阵存在的充要条件是它的行列式 $|A| \neq 0$ 。

矩阵的秩

设 A 为 $m \times n$ 矩阵, A 的秩用 $\text{rank}(A)$ 表示, 定义为 A 中最大的线性无关的行数, 或等价地定义为 A 中最大的线性无关的列数。若 $\text{rank}(A) = \min\{m, n\}$, 则称 A 为满秩矩阵。易见, n 阶非奇异方阵 A 的秩 $\text{rank}(A) = n$ 。

矩阵的特征值与特征矢量

设 A 为 n 阶方阵, 若有复数 λ 和非零矢量 x , 使 $Ax = \lambda x$ 成立, 则称 λ 为 A 的一个特征值(根), 称 x 为 A 的与 λ 相应的特征矢量或特征向量。 $|A - \lambda I_n|$ 是 λ 的 n 次多项式, 称为 A 的特征多项式, 当

$x \neq 0$ 时, 因为

$$Ax = \lambda x \Leftrightarrow (A - \lambda I_n)x = 0 \Leftrightarrow |A - \lambda I_n| = 0$$

所以方阵 A 的特征值也就是特征多项式 $|A - \lambda I_n|$ 的根。

正定和半正定矩阵

设 A 为 n 阶对称矩阵。

1. 如果对任何实的 n 维非零矢量 x , 都有 $x^T Ax > 0$, 则称 A 为正定矩阵, 记为 $A > 0$ 。

2. 如果对任何实的 n 维矢量 x , 都有 $x^T Ax \geq 0$, 则称 A 为半正定(或正半定)矩阵, 记为 $A \geq 0$ 。

3. 如果 $(-A) > 0$, 则称 A 为负定矩阵, 记为 $A < 0$ 。

4. 如果 $(-A) \geq 0$, 则称 A 为半负定(或负半定)矩阵, 记为 $A \leq 0$ 。

定理 1.3 对称矩阵 A 为正定(半正定)矩阵的充要条件是 A 的顺序主子式都大于零(都非负)。

定理 1.4 对称矩阵 A 为正定(半正定)矩阵的充要条件是 A 的特征值都是正的(非负的)。

定理 1.5 若矩阵 A 非奇异, 则 $A^T A$ 为对称正定矩阵, 其特征值均为正数。

矩阵 A 的范数 $\|A\|$ 定义为

$$\|A\| = \max_{\|x\|=1} \|Ax\| \quad (1-8)$$

矩阵的条件数

设 A 为 n 阶非奇异方阵, 定义 A 的条件数为

$$P(A) = \|A^{-1}\| \|A\| \quad (1-9)$$

二、集合和序列

集 合

具有某种性质的元素(或对象)的全体, 称为集合。规定一个集合可以列举出它的所有元素, 或指明它的元素必须满足的性质,

例如集合 $S = \{1, 2, 3, 4\}$, 也可表示为 $S = \{x | 1 \leq x \leq 4, x \text{ 为整数}\}$ 。若 x 是 S 的一个元素, 则写成 $x \in S$, 若 x 不是 S 的元素, 则写成 $x \notin S$ 或 $x \notin S$ 。集合用大写字母如 A, B, S, P 等表示。空集用 \emptyset 表示, 它不含有任何元素。

集合的并、交和子集

两集合 S_1 和 S_2 的并集定义为

$$S_1 \cup S_2 = \{x | x \in S_1 \text{ 或 } x \in S_2\} \quad (2-1)$$

两集合 S_1 和 S_2 的交集定义为

$$S_1 \cap S_2 = \{x | x \in S_1 \text{ 且 } x \in S_2\} \quad (2-2)$$

若集 A 的每个元素都是集 B 的元素, 则称 A 为 B 的子集, 记为 $A \subset B$, 或 $B \supset A$ 。若 $A \subset B$ 和 $B \subset A$ 同时成立, 则称集 A 与集 B 相等, 记为 $A = B$ 。

实数集合

设 a, b 为实数, $[a, b]$ 表示所有满足条件 $a \leq x \leq b$ 的实数 x 的集合, (a, b) 表示所有满足条件 $a < x < b$ 的实数 x 的集合, 其余类推。

设 S 为实数集, 满足条件 $y \leq x, \forall x \in S$ 的实数 y 称为 S 的下界, 显然, 若 $z < y$, 而 y 为 S 的下界, 则 z 亦为 S 下界, S 所有下界中的最大者称为 S 的最大下界或下确界, 记为

$$\inf S = \inf\{x | x \in S\} = \max\{y | y \leq x, \forall x \in S\} \quad (2-3)$$

类似地可定义 S 的最小上界或上确界, 记为

$$\sup S = \sup\{x | x \in S\} = \min\{y | y \geq x, \forall x \in S\} \quad (2-4)$$

邻 域

点 $x \in \mathbf{R}^n$ 的 $\epsilon (\epsilon > 0)$ 邻域, 定义为

$$N_\epsilon(x) = \{y | \|y - x\| < \epsilon, y \in \mathbf{R}^n\} \quad (2-5)$$

内点和开集

设集 $S \subset \mathbf{R}^n$, $S \neq \emptyset$, $x \in S$ 。若存在 $\varepsilon > 0$, 使得 $N_\varepsilon(x) \subset S$, 则称 x 为 S 的一个内点。 S 的所有内点组成的集合称为 S 的内部, 记为 $\text{int } S$ 。

$$\text{int } S = \{x | x \in S, \text{ 且有 } \varepsilon > 0, \text{ 使 } N_\varepsilon(x) \subset S\}$$

如果 $\text{int } S = S$, 则称 S 为开集。

接触点、极限点和闭集

设集 $S \subset \mathbf{R}^n$, $x \in \mathbf{R}^n$, 如果对任意 $\varepsilon > 0$, 都有

$$S \cap N_\varepsilon(x) \neq \emptyset \quad (2-6)$$

则称 x 为 S 的接触点, S 与其所有的接触点的并集称为 S 的闭包, 记为 $\text{cl } S$ 。如果对任意 $\varepsilon > 0$, 都有

$$(S - \{x\}) \cap N_\varepsilon(x) \neq \emptyset \quad (2-7)$$

则称 x 为 S 的极限点。如果 $S = \text{cl } S$, 则称 S 为闭集。

边界点和边界

设集 $S \subset \mathbf{R}^n$, $x \in S$, 若对任意 $\varepsilon > 0$, $N_\varepsilon(x)$ 既含有 S 的点, 又含有不属于 S 的点, 则称 x 为 S 的边界点, S 的所有边界点组成的集合, 称为 S 的边界, 记为 ∂S 。

紧 集

如果存在 $M > 0$, 使得对任意 $x \in S$, 都有 $\|x\| \leq M$, 则称 S 为有界集, 否则称 S 为无界集, 如果集 S 是有界集, 又是闭集, 则称它为紧集。

收敛序列和子序列

设 $\{x^{(k)}\}$ 为 \mathbf{R}^n 中矢量组成的序列。如果对于 \mathbf{R}^n 中的确定矢量 x^* , 当 $k \rightarrow \infty$ 时, 有 $\|x^{(k)} - x^*\| \rightarrow 0$, 即对任给的 $\varepsilon > 0$, 存在一个正整数 N , 使得对所有的 $k \geq N$, 总有 $\|x^{(k)} - x^*\| < \varepsilon$ 成立, 则称序列 $\{x^{(k)}\}$ 是收敛的, 并以 x^* 为其极限, 记为 $\lim_{k \rightarrow \infty} x^{(k)} = x^*$ 。易见, 任何收敛序列的极限是唯一的。

对给定的序列 $\{x^{(k)}\}$, 考虑满足条件 $k_1 < k_2 < k_3 < \dots$ 的正整

数序列 $\{k_i\}$, 则称序列 $\{x^{(k_i)}\}$ 为 $\{x^{(k)}\}$ 的一个子序列。

$\{x^{(k)}\}$ 收敛于 x^* 当且仅当 $\{x^{(k)}\}$ 的每个子序列都收敛于 x^* 。

柯西(Cauchy) 序列

如果对任给的 $\varepsilon > 0$, 存在一个正整数 N , 使得当 $n, m > N$ 时, 都有 $\|x^{(n)} - x^{(m)}\| < \varepsilon$ 成立, 则称 $\{x^{(k)}\}$ 为柯西序列。 \mathbf{R}^n 中的序列 $\{x^{(k)}\}$ 收敛的充要条件为 $\{x^{(k)}\}$ 是一个柯西序列。

定理 2.1 设 S 为紧集, $\{x^{(k)}\}$ 为 S 中的任一无穷序列, 则 $\{x^{(k)}\}$ 中必有收敛的子序列, 且其极限点属于 S 。

三、函 数

连续函数

设函数 $f: S \subset \mathbf{R}^n \rightarrow \mathbf{R}$, $x_0 \in S$, 如果对任给的 $\varepsilon > 0$, 存在 $\delta > 0$, 使得当 $x \in S \cap N_\delta(x_0)$ 时, 恒有 $|f(x) - f(x_0)| < \varepsilon$ 成立, 则称函数 $f(x)$ 在 x_0 点连续。如果 $f(x)$ 在 S 中的每一点都连续, 则称 $f(x)$ 在 S 上连续。简记为 $f(x) \in C$ 。

对矢量函数 $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$, $x_0 \in S$, 如果它的每个分量函数 f_1, f_2, \dots, f_p 都在 x_0 点连续, 则说 $F(x)$ 在 x_0 点连续。

可微函数

设 $S \subset \mathbf{R}^n$ 非空, $f: S \rightarrow \mathbf{R}$, 若在点 $x_0 \in \text{int } S$, 存在矢量 $\nabla f(x_0)$ (称为 $f(x)$ 在 x_0 点的梯度) 和函数 $\beta: \mathbf{R}^n \rightarrow \mathbf{R}$, 使得

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \|x - x_0\| \beta(x_0; x - x_0), \quad \forall x \in S \quad (3-1)$$

(其中 $\lim_{x \rightarrow x_0} \beta(x_0; x - x_0) = 0$), 则称 $f(x)$ 在 x_0 点(一次)可微, 若 $f(x)$ 在开集 $S_0 \subset S$ 中的每一点都可微, 则称 $f(x)$ 在 S_0 上可微。

定理 3.1 若 $f(x)$ 在 x_0 点可微, 则 $f(x)$ 在 x_0 点连续, 并且只有唯一的一个梯度矢量, 即

$$\nabla f(\mathbf{x}_0) = \left(\frac{\partial f(\mathbf{x}_0)}{\partial x_1}, \frac{\partial f(\mathbf{x}_0)}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x}_0)}{\partial x_n} \right)^T \quad (3-2)$$

如果在点 $\mathbf{x}_0 \in \text{int } S$, 存在矢量 $\nabla f(\mathbf{x}_0)$ 和 n 阶实对称方阵 $H(\mathbf{x}_0)$ (称为 $f(\mathbf{x})$ 在 \mathbf{x}_0 点的海赛矩阵), 以及函数 $\beta: \mathbf{R}^n \rightarrow \mathbf{R}$, 使得

$$\begin{aligned} f(\mathbf{x}) = & f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0) + \\ & (1/2) (\mathbf{x} - \mathbf{x}_0)^T H(\mathbf{x}_0) (\mathbf{x} - \mathbf{x}_0) + \\ & \|\mathbf{x} - \mathbf{x}_0\|^2 \beta(\mathbf{x}_0; \mathbf{x} - \mathbf{x}_0), \quad \forall \mathbf{x} \in S \end{aligned} \quad (3-3)$$

(其中 $\lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \beta(\mathbf{x}_0; \mathbf{x} - \mathbf{x}_0) = 0$), 则称函数 $f(\mathbf{x})$ 在点 \mathbf{x}_0 二次可微。

定理 3.2 若 $f(\mathbf{x})$ 在点 \mathbf{x}_0 二次可微, 则

$$H(\mathbf{x}_0) = \left(\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} \right)_{n \times n} \quad (3-4)$$

其中

$$\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_j \partial x_i}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n$$

多元函数的中值定理

设 $S \subset \mathbf{R}^n$ 为非空开凸集, $f: S \rightarrow \mathbf{R}$ 一次可微, 则对 S 中的每一个 \mathbf{x}_1 和 \mathbf{x}_2 , 有

$$f(\mathbf{x}_2) = f(\mathbf{x}_1) + \nabla f(\mathbf{x})^T (\mathbf{x}_2 - \mathbf{x}_1) \quad (3-5)$$

成立, 其中 $\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$, 对某个 $\lambda \in (0, 1)$ 。

多元函数的 Taylor 定理

设 $S \subset \mathbf{R}^n$ 为非空开凸集, $f: S \rightarrow \mathbf{R}$ 二次可微, 则二阶形式的 Taylor 定理成立, 即对 S 中的每一个 \mathbf{x}_1 和 \mathbf{x}_2 , 有

$$\begin{aligned} f(\mathbf{x}_2) = & f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1) + \\ & (1/2) (\mathbf{x}_2 - \mathbf{x}_1)^T H(\mathbf{x}) (\mathbf{x}_2 - \mathbf{x}_1) \end{aligned} \quad (3-6)$$

其中 $H(\mathbf{x})$ 是 $f(\mathbf{x})$ 在 \mathbf{x} 点的海赛矩阵, 而 $\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2$, 对某个 $\lambda \in (0, 1)$ 。

公式 (3-5) 称为 n 元函数的中值公式或一阶 Taylor 公式, 公式 (3-6) 称为 n 元函数二阶 Taylor 公式。