



辽宁工程技术大学“十五”规划教材

最优化理论与方法

高雷阜 编著

Theory and Methods
of Optimization



NEUPRESS
东北大学出版社

辽宁工程技术大学“十五”规划教材

最优化理论与方法

Theory and Methods of Optimization

高雷阜 编著

东北大学出版社

· 沈 阳 ·

©高雷阜 2005

图书在版编目 (CIP) 数据

最优化理论与方法 / 高雷阜编著. — 沈阳 : 东北大学出版社, 2005.11

(辽宁工程技术大学“十五”规划教材)

ISBN 7-81102-212-5

I. 最… II. 高… III. 最优化、IV. O224

中国版本图书馆 CIP 数据核字 (2005) 第 135519 号

出 版 者：东北大学出版社

地址：沈阳市和平区文化路 3 号巷 11 号

邮编：110004

电话：024—83687331 (市场部) 83680267 (社务室)

传真：024—83680180 (市场部) 83680265 (社务室)

E-mail: neuph @ neupress.com

http: // www. neupress. com

印 刷 者：沈阳农业大学印刷厂

发 行 者：东北大学出版社

幅面尺寸：184mm×260mm

印 张：14.375

字 数：350 千字

出版时间：2005 年 11 月第 1 版

印刷时间：2005 年 11 月第 1 次印刷

责任编辑：孟 颖

责任校对：张 萍

封面设计：唐敏智

责任出版：杨华宁

定 价：29.50 元

前 言

最优化理论与方法讨论的是为找出对象问题的最优解决策略而采取的模型化及其方法。其过程是先将待解决问题用最优化形式描述为在给定的约束条件下找出使某个目标函数达到最大（小）解，然后再采用数学上严密的算法来求解。这一领域被认为是运筹学的一个重要的组成部分。

所谓问题解决方法，除了运筹学外，人们还提出了人工智能、专家系统、系统理论、模糊集合、神经网络和遗传算法等典型方法，并使它们实用化。这些方法各具特征，承担解决问题某个方面的任务，但难以替代数学规划（最优化的核心部分）的作用。这是因为把要解决的问题描述成最优化问题后，没有其他方法能像数学规划那样深刻解析对象问题，并给出精确解。

我们不仅从数学观点对这些最优化方法进行理论研究，还强调了它们作为实用工具在现实中被广泛应用的事实。大多数有代表性的最优化算法已有方便实用的应用数学软件（优化工具箱）。但是，有效地利用这些成果是以有待解决的问题已经被模型化成最优化问题的形式为前提的。

尽管仅靠最优化的方法不能解决所有问题，但这些方法对解决最优化问题中可以模型化的部分是必不可少的，今后也会继续起到解决方法的核心作用。

最优化有自己独特严谨的理论结构体系，实用有效的算法，色彩斑斓的应用。最优化算法构造巧妙，注重计算过程和收敛性分析。

计算机科学的发展为最优化方法提供了强大的计算支撑，算法的易于实现既是最优化理论的逻辑演绎，同时也促进了理论的螺旋式发展。成熟的应用数学软件既成为应用迅猛发展的有力工具，也使得各种不同算法的成功比较得以完成。

本书是根据作者多年来的讲义修改补充写成的。主要读者对象为数学与应用数学、信息与计算科学等理科专业和管理工程、系统工程等专业本科生和理工科院校的工科研究生，以及从事运筹优化应用的工程技术人员、管理人员。全书共分七章，比较系统地介绍了线性规划、非线性规划、整数规划、多目标规划、动态规划、蒙特卡罗方法，还简单地介绍了应用数学软件 Matlab。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典型的优化问题。本书第 3、4、5、6、7 章由高雷阜撰写，第 1、2 章由赵艳艳撰写，全书由高雷阜修改定稿。限于水平，书中疏漏难免，恳请读者批评指正。

高雷阜

2005 年 10 月

目 录

第 1 章 Matlab 简介	1
1.1 变 量	1
1.2 运算符及标点符号	1
1.3 数 组	2
1.3.1 数组的创建	2
1.3.2 数组元素的访问	2
1.3.3 数组的方向	3
1.3.4 向量函数	3
1.4 矩 阵	3
1.4.1 创建矩阵	3
1.4.2 矩阵的提取	4
1.4.3 删 除	5
1.4.4 矩阵函数	5
1.4.5 特征值和特征向量	5
1.5 图形绘制	6
1.5.1 二维图形绘制	6
1.5.2 图形标注	6
1.5.3 三维图形绘制	7
1.6 M 文件	9
1.7 优化工具箱	9
1.7.1 Matlab 求解优化问题的主要函数	10
1.7.2 优化函数的输入变量	10
1.7.3 优化函数的输出变量	11
1.7.4 控制参数 options 的设置	11
第 2 章 线性规划	12
2.1 线性规划模型	12
2.1.1 模型实例	12
2.1.2 线性规划的特征	13
2.1.3 标准化的方法	14

2.1.4	线性规划问题的解	16
2.2	线性规划的基本定理	17
2.2.1	凸集的概念	17
2.2.2	凸集分离定理	18
2.2.3	线性规划解的基本定理	22
2.3	单纯形法	26
2.3.1	单纯形法的基本思想	26
2.3.2	初始基可行解	26
2.3.3	最优性准则	27
2.3.4	基可行解的迭代与改进	29
2.3.5	单纯形表及其计算步骤	31
2.4	人工变量单纯形法	34
2.4.1	大 M 法(big- M method)	34
2.4.2	两阶段单纯形法(two-phase method)	37
2.5	改进单纯形法	41
2.5.1	B^{-1} 与 \bar{B}^{-1} 的关系	41
2.5.2	改进单纯形法的算法步骤	42
2.5.3	改进单纯形法的特点	43
2.6	对偶问题	44
2.6.1	对偶问题的表达	44
2.6.2	对偶问题的基本性质	48
2.6.3	对偶单纯形法	50
2.7	灵敏度分析	51
2.7.1	价格系数 c_j 的变化	52
2.7.2	资源约束向量 b 的变化	53
2.7.3	约束方程组的系数矩阵 A 的变化	54
2.7.4	增加一个变量 x_{n+1}	54
2.7.5	增加一个约束条件	55
2.8	多项式时间算法	56
2.9	线性规划程序求解	64
	习 题	68
第3章	非线性规划	72
3.1	非线性规划模型和基础理论	72
3.1.1	非线性规划模型	72
3.1.2	梯度与 Hessian 矩阵	73
3.1.3	多元函数的 Taylor 展式	75
3.2	凸函数	76
3.2.1	凸函数的定义	76

3.2.2	凸函数的判别	77
3.2.3	凸函数的极值	80
3.2.4	凸规划(convex programming)	81
3.3	最优性条件	81
3.3.1	无约束问题的最优性条件	81
3.3.2	约束问题的最优性条件	85
3.4	迭代下降算法	102
3.5	一维搜索	103
3.5.1	Fibonacci 法	105
3.5.2	0.618 法	107
3.5.3	0.618 法与 Fibonacci 法的关系	109
3.5.4	对分法	110
3.5.5	牛顿法	111
3.5.6	抛物线法	113
3.5.7	三次插值法	114
3.6	最速下降法	116
3.7	牛顿法	118
3.7.1	牛顿法	118
3.7.2	阻尼牛顿法	118
3.7.3	牛顿法的进一步修正	119
3.8	共轭方向法	119
3.9	拟牛顿法	126
3.9.1	对称秩 1 算法	127
3.9.2	DFP 算法	128
3.9.3	BFGS 公式及 Broyden 族	130
3.10	无约束最优化的直接方法	132
3.10.1	Hooke-Jeeves 方法	132
3.10.2	单纯形法	135
3.11	惩罚函数法和障碍函数法	140
3.11.1	惩罚函数法	140
3.11.2	障碍函数法	142
3.11.3	混合惩罚函数法	143
3.12	可行方向法	144
3.12.1	线性约束情形	144
3.12.2	非线性约束情形	151
3.13	Rosen 梯度投影法	152
3.13.1	投影矩阵	152
3.13.2	梯度投影法原理	153
3.14	既约梯度法	159

3.14.1	Wolfe 既约梯度法	159
3.14.2	广义既约梯度法	164
3.15	非线性规划程序求解	167
3.15.1	用 Matlab 解一元函数的无约束优化问题	167
3.15.2	用 Matlab 解多元函数的无约束优化问题	168
3.15.3	用 Matlab 解约束优化问题	170
习 题	172
第 4 章	整数规划	175
4.1	整数规划的数学模型	175
4.2	割平面法	177
4.2.1	纯整数线性规划的情形	177
4.2.2	混合整数线性规划的情形	180
4.3	分枝估界法	181
4.4	隐枚举法	183
习 题	187
第 5 章	多目标规划	189
5.1	多目标规划问题举例	189
5.2	多目标规划问题的解集和像集	191
5.2.1	多目标规划问题的解集	191
5.2.2	多目标规划问题的像集	192
5.3	处理多目标规划问题的一些方法	193
5.3.1	约束法	193
5.3.2	分层序列法	193
5.4	评价函数法	194
5.4.1	理想点法	195
5.4.2	平方和加权法	196
5.4.3	线性加权和法	197
5.4.4	极小-极大法(min-max 法)	198
5.4.5	乘法	198
5.5	逐步法(Step Method)	199
习 题	200
第 6 章	动态规划	202
6.1	多阶段决策问题与动态规划	202
6.1.1	多阶段决策问题	202
6.1.2	动态规划的基本概念	202
6.2	动态规划模型与求解	204

6.2.1 动态规划模型	204
6.2.2 动态规划的求解	205
6.3 动态规划应用举例	206
习 题	211
第7章 蒙特卡罗方法	213
7.1 基本原理和方法	213
7.1.1 蒙特卡罗方法的基本原理	213
7.1.2 随机数产生原理	213
7.2 基本方法和流程图	214
7.3 约束条件的处理	214
7.3.1 不等式约束情形	215
7.3.2 等式约束情形	215
7.3.3 整数变量的情形	216
7.4 非线性规划的优化解	217
习 题	218
参考文献	219

第 1 章 Matlab 简介

Matlab 的全称是 Matrix Laboratory, 由美国 Math Works 公司开发, 它是一种集数值计算、符号运算、可视化建模和发展以及图形处理等多种功能于一体的图形化语言, 主要包括两大内容: 核心函数和工具箱. 现将其基本功能和操作介绍如下.

1.1 变 量

Matlab 中变量的命名规则:

- (1) 变量名必须是不含空格的单个词;
- (2) 变量名区分大小写;
- (3) 变量名最多不超过 19 个字符;
- (4) 变量名必须以字母开始, 之后可以是任意字母、数字或下划线, 变量名中不允许使用标点符号.

除了上述命名规则, Matlab 还有几个特殊变量, 见表 1.1.

表 1.1

特殊变量	取 值
ans	用于结果的缺省变量名
pi	圆周率
eps	计算机的最小数, 当和 1 相加时就产生一个比 1 大的数
flops	浮点运算数
inf	无穷大
NaN	不定量
i, j	$i = j = \sqrt{-1}$
nargin	所用函数的输入变量数目
nargout	所用函数的输出变量数目
realmin	最小可用正实数
realmax	最大可用正实数

1.2 运算符及标点符号

运算符见表 1.2.

表 1.2

运算	算术运算符	数组运算符	矩阵运算符
加法	+	+	+
减法	-	-	-
乘法	*	.*	*
除法	./, \	./, \	./, \

在 Matlab 中:

(1) Matlab 的每条命令后, 若为逗号或无标点符号, 则显示命令的结果; 若命令后为分号, 则禁止显示结果.

(2) “%”后的文字为注释语句.

(3) “/...”表示续行.

1.3 数 组

1.3.1 数组的创建

简单数组的创建见表 1.3.

表 1.3

常用命令	说 明
$x = [1\ 2\ 3\ 4\ 5]$	创建包含指定元素的行向量
$x = m:n$	创建从 m 开始, 加 1 计数, 到 n 结束的行向量
$x = m:p:n$	创建从 m 开始, 加 p 计数, 到 n 结束的行向量
$x = \text{linspace}(m, n, p)$	创建从 m 开始, 到 n 结束, 有 p 个元素的行向量
$x = \text{logspace}(m, n, p)$	创建从 10^m 开始, 到 10^n 结束, 有 p 个元素的对数分隔行向量

例 1.1 >> $x = 1: 2: 11$

```
x =
    1         3         5         7         9        11
```

>> $y = \text{linspace}(0, 1, 5)$

```
y =
    0         0.2500    0.5000    0.7500    1.0000
```

>> $z = \text{logspace}(0, 1, 5)$

```
z =
    1.0000    1.7783    3.1623    5.6234   10.0000
```

1.3.2 数组元素的访问

(1) 访问一个元素: 数组元素可以用下标访问, 如 $x(i)$ 表示数组 x 的第 i 个元素.

(2) 访问一块元素: $x = (a: b: c)$ 表示访问数组 x 从第 a 个元素开始、以步长为 b 、到第 c 个元素(但不超过 c)的元素.

例 1.2 >> $m = x(3: 2: 6)$, $n = x(6: -4: 1)$

```

m =
    5    9
n =
   11    3

```

1.3.3 数组的方向

例 1.2 中的数组都是行向量，数组也可以是列向量。列向量的数组操作和运算与行向量是一样的，唯一的区别是结果以列的形式显示。产生列向量有两种方法：直接产生和转置产生。

```

例 1.3 >> c=[1; 2; 3], b=c'
c =
     1
     2
     3
b =
     1     2     3

```

1.3.4 向量函数

一般标量函数都可用于数组，此时函数作用于数组的每个元素。向量函数只有作用于向量才有意义，常用的向量函数见表 1.4。向量函数也可作用于矩阵，此时运算结果为一个行向量，行向量的每个元素是函数作用于矩阵的相应列向量所得的结果。

表 1.4

函 数	功 能	函 数	功 能
max	最大值	mean	平均值
min	最小值	median	中间值
sum	和	prod	乘积
length	长度	sort	从小到大排列

1.4 矩 阵

1.4.1 创建矩阵

(1) 直接输入创建矩阵。Matlab 中不需要设置矩阵的类型与维数，它们由输入的格式和内容来唯一决定。

小矩阵和没有任何规律的矩阵可以通过直接输入元素的方法来创建，数组、矩阵的元素由中括号括起来，同一行元素由逗号或者空格分开，行与行之间由回车或者分号分开。

```

例 1.4 >>[1, 2
           2, 3]
ans =
     1     2

```

```

2 3
>> [1, 2; 3, 4]
ans =
1 2
3 4

```

这种方法工作量大且容易出错, 因此 Matlab 提供了许多函数和方法可以方便快速地创建矩阵.

(2) 使用冒号创建矩阵. 方法与创建数组类似.

例 1.5 >> a = [-pi: pi; 2: 8]

```

a =
-3.1416 -2.1416 -1.1416 -0.1416 0.8584 1.8584 2.8584
2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000

```

(3) 使用函数创建矩阵(见表 1.5)

表 1.5

矩阵符号	说明
zeros(i, j)	创建 i 行 j 列的全零矩阵
ones(i, j)	创建 i 行 j 列的全 1 矩阵
eye(i, j)	创建 i 行 j 列的对角线为 1 的矩阵
rand(i, j)	创建 i 行 j 列的 (0, 1) 均步随机矩阵
[]	返回一个空矩阵

1.4.2 矩阵的提取

(1) 通过使用冒号提取矩阵.

例 1.6 >> x = [1 2 3; 4 5 6; 7 8 9];

```
>> x(:, 1)
```

```

ans =
1
4
7

```

(2) Matlab 提供了 diag, tril, triv 等函数, 可以实现对矩阵的提取. 现以 diag 为例.

例 1.7 >> x = (1: 3); A = diag(x)

```

A =
1 0 0
0 2 0
0 0 3
>> x = diag(A)
x =
1
2
3

```

```
>> A=diag(x, 1)
```

```
A =
```

```
0 1 0 0
```

```
0 0 2 0
```

```
0 0 0 3
```

```
0 0 0 0
```

```
>> x=diag(A, 1)
```

```
x =
```

```
1
```

```
2
```

```
3
```

1.4.3 删除

(1) $x(:, j) = []$ 将矩阵 x 的第 j 列删除;

(2) $x(i, :) = []$ 将矩阵的第 i 行删除.

例 1.8 >> $x = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9];$

```
>> x(:, 1) = []
```

```
x =
```

```
2 3
```

```
5 6
```

```
8 9
```

1.4.4 矩阵函数

内容见表 1.6.

表 1.6

函数	功能	函数	功能
expm	矩阵的指数	rank	矩阵的秩
logm	矩阵的对数	condeig	对应于特征值的条件数
cond	矩阵的条件数	norm	矩阵的范数
det	方阵的行列式值	null	矩阵的零空间

1.4.5 特征值和特征向量

(1) $d = \text{eig}(A)$ 返回矩阵 A 的特征值 d ;

(2) $[V, D] = \text{eig}(A)$ 返回两个矩阵 V 和 D , D 的主对角线由 A 的特征值组成, V 的列向量由 A 的特征向量组成.

例 1.9 >> $A = \text{magic}(3);$

```
>> [V, D] = eig(A)
```

```
V =
```

$$D = \begin{bmatrix} -0.5774 & -0.8131 & -0.3416 \\ -0.5774 & 0.4714 & -0.4714 \\ -0.5774 & 0.3416 & 0.8131 \\ 15.0000 & 0 & 0 \\ 0 & 4.8990 & 0 \\ 0 & 0 & -4.8990 \end{bmatrix}$$

1.5 图形绘制

1.5.1 二维图形绘制

Matlab 作图是通过描点、连线实现的, 故在画一个曲线图形之前, 必须先取得该图形上的一系列的点的坐标, 命令为

```
plot(x, y, s)
plot(x, y)
plot(x, y1, s1, x, y2, s2, ..., x, yn, sn)
```

其中, x, y 是向量, 分别表示点集的横坐标和纵坐标, 命令 $\text{plot}(x, y, s)$ 描绘该点集所表示的曲线, 其线型由 s 确定:

y 黄色	.	点	-	实线
r 红色	o	圈	--	虚线
g 绿色	+	加号	:	点线
b 蓝色	s	正方形	-.	点划线

命令 $\text{plot}(x, y1, s1, x, y2, s2, \dots, x, yn, sn)$ 将多条线画在一起, 参数同 $\text{plot}(x, y, s)$.

例 1.10 在 $[0, 2 * \pi]$ 中, 用红线画 $\sin(x)$, 用绿圈画 $\cos(x)$.

```
>> x=linspace(0, 2 * pi, 30); y=sin(x); z=cos(x);
>> plot(x, y, 'r', x, z, 'go')
```

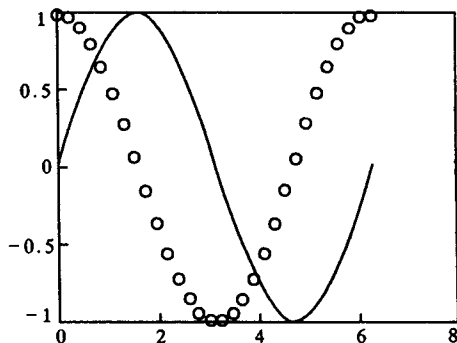


图 1.1

1.5.2 图形标注

(1) xlabel, ylabel, title 三个函数分别对 x 轴, y 轴以及全图进行标注, 可以使图形变

得完整.

```
例 1.11 fplot('sin(x)./x', [-20 20 -0.4 1.2]);  
title('Fplot of f(x) = sin(x)/x');  
xlabel('x'), ylabel('f(x)')
```

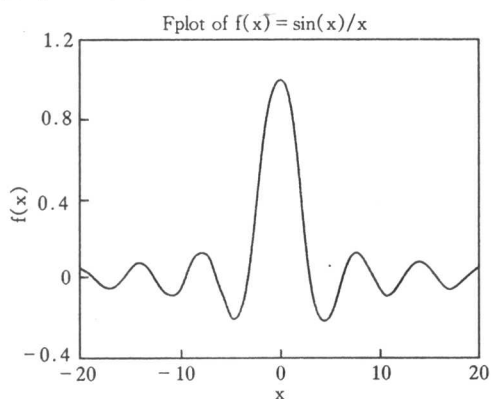


图 1.2

(2) 函数 `gtext` 可以将一个字符串放到图形中, 位置由鼠标确定.

(3) `grid on` 可以在当前坐标轴上添加网格线.

```
例 1.12 plot(peaks(80));  
gtext('a beautiful figure', 'fontsize', 16);  
grid on
```

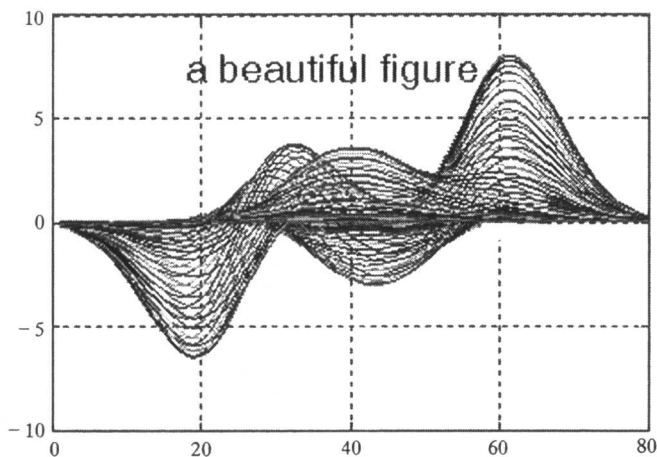


图 1.3

1.5.3 三维图形绘制

(1) `plot3` 函数可以用来绘制一个三维的曲线, 它的格式类似于 `plot`, 不过多了一个 z 方向的数据.

```
例 1.13 t=0: pi/50: 10 * pi;  
plot3(sin(t), cos(t), t);
```