# 第一章 最优化问题总论

无论做任何一件事，人们总希望以最少的代价取得最大的效益，也就是力求最好，这就是优化问题．最优化就是在一切可能的方案中选择一个最好的方案以达到最优目标的学科．例如，从甲地到乙地有公路、水路、铁路、航空四种走法，如果我们追求的目标是省钱，那么只要比较一下这四种走法的票价，从中选择最便宜的那一种走法就达到目标．这是最简单的最优化问题，实际优化问题一般都比较复杂．

概括地说，凡是追求最优目标的数学问题都属于最优化问题．作为最优化问题，一般要有三个要素：第一是目标；第二是方案；第三是限制条件．而且目标应是方案的“函数”．如果方案与时间无关，则该问题属于静态最优化问题；否则称为动态最优化问题．

## §1.1 最优化问题数学模型

最简单的最优化问题实际上在高等数学中已遇到，这就是所谓函数极值，我们习惯上又称之为经典极值问题．

例1.1 对边长为*a*的正方形铁板，在四个角处剪去相等的正方形以制成方形无盖水槽，问如何剪法使水槽的容积最大？

解 设剪去的正方形边长为*x*，由题意易知，与此相应的水槽容积为

．

令 ，

得两个驻点：．

第一个驻点不合实际，这是因为剪去4个边长为的正方形相当于将铁板全部剪去．现在来判断第二个驻点是否为极大点．

∵ ， ，

∴  是极大点．

因此，每个角剪去边长为的正方形可使所制成的水槽容积最大．

例1.2 求侧面积为常数，体积最大的长方体体积．

解 设长方体的长、宽、高分别为体积为，则依题意知体积为

，

条件为

．

由拉格朗日乘数法，考虑函数

，



由题意可知应是正数，由此，将上面三个等式分别乘以并利用条件，得到



比较以上三式可得

，

从而．又侧面积固定的长方体的最大体积客观存在，因此侧面积固定的长方体中以正方体体积最大，其最大值为．

例1.3 某单位拟建一排四间的停车房，平面位置如图1.1所示．由于资金及材料的限制，围墙和隔墙的总长度不能超过40m，为使车房面积最大，应如何选择长、宽尺寸？

解 设四间停车房长为，宽为．由题意可知面积为

，

且变量，应满足

，

．

即求 ，

|  |  |
| --- | --- |
| |  | | --- | | *x*2 | |

|  |  |
| --- | --- |
| |  | | --- | | *x*1 | |

|  |  |
| --- | --- |
| |  | | --- | | 图1.1 | |



以上三个例子，虽然简单，但是它代表了三种类型的最优化问题．

第一个例子代表无约束极值问题：

一般可表示为或．这里，是定义在上的可微函数．

求极值的方法是从如下含有个未知数的非线性方程组



中解出驻点，然后判定或验证这些驻点是不是极值点．

第二个例子代表具有等式约束的极值问题：

一般可表示为



该问题的求解通常采用拉格朗日乘数法，即把这个问题转化为求



的无约束极值问题．

第三个例子代表具有不等式约束的极值问题．

下面具体分析上述三种类型的最优化问题中按经典极值问题解法可能出现不能解决的情况：

（1）当变量个数增加且方程组又是非线性，求解此方程组只有在相当特殊情况下才能人工解出．正因为如此，通常高等数学中的求极值问题的变量个数一般不超过三个．

（2）当限制条件出现不等式，无论变量数多少，按经典极值方法求解根本无法解决．

直到本世纪50年代，最优化理论的建立以及电子计算机的迅速发展，才为求解各种最优化问题提供了雄厚的基础和有效手段．不过最优化方法作为一门崭新的应用学科，有关理论和方法还没有完善，有许多问题还有待解决，目前正处于迅速发展之中．

最优化问题的数学模型包含三个要素：变量（又称设计变量）、目标函数、约束条件．

### 一、变量

一个优化设计方案是用一组设计参数的最优组合来表示的．这些设计参数可概括地划分为两类：一类是可以根据客观规律、具体条件、已有数据等预先给定的参数，统称为常量；另一类是在优化过程中经过逐步调整，最后达到最优值的独立参数，称为变量．优化问题的目的就是使各变量达到最优组合．变量的个数称为优化问题的维数．例如有个变量的优化问题就是在维空间中寻优．维空间中的点就代表优化问题中的一个方案．当变量为连续量时，称为连续变量；若变量只能在离散量中取值，称为离散变量．

### 二、目标函数

反映变量间相互关系的数学表达式称为目标函数．其值的大小可以用来评价优化方案的好坏．按照规范化的形式，一般把优化问题归结为求目标函数的极小化问题，换句话说，目标函数值越小，优化方案越好．对于某些追求目标函数极大的问题，可以转化成求其负值最小的问题，即

．

因此在本书的叙述中，一律把优化问题描述为目标函数的极小化问题，其一般形式为

．

如果优化问题只有一个目标函数，称为单目标函数，如果优化问题同时追求多个目标，则该问题的目标函数称为多目标函数．

多目标优化问题的目标函数通常表示为

，

其中．这时的目标函数在目标空间中是一个维矢量，所以又称之为矢量优化问题（一般用min加一个前缀“”来表示矢量极小化）．

### 三、约束条件

变量间本身应该遵循的限制条件的数学表达式称为约束条件或约束函数．

约束条件按其表达式可分为不等式约束和等式约束两种，即



上式中“*s*. *t*.”为*Subject to*的缩写，意即“满足于”或“受限于”．按约束条件的作用还可将约束条件划分为起作用的约束（紧约束、有效约束）和不起作用的约束（松约束、消极约束）．等式约束相当于空间里一条曲线（曲面或超曲面），点*X*必须为该曲线（曲面或超曲面）上的一点，因而总是紧约束．有一个独立的等式约束，就可用代入法消去一个变量，使优化问题降低一维．因此，数学模型中独立的等式约束个数应小于变量个数；如果相等，就不是一个待定优化系统，而是一个没有优化余地的既定系统．不等式约束通常是以其边界表现出约束作用的，它只限制点***X***必须落在允许的区域内（包括边界上），因而不等式约束的个数与变量个数无关．不带约束条件的优化问题称为无约束最优化问题；带约束条件的优化问题称为约束最优化问题．

### 四、带约束条件的优化问题数学模型表示形式

综上所述，全书所要讨论的问题是如下的（静态）最优化问题，其表示形式有三种：

第一种最优化问题表示形式为



第二种最优化问题表示形式为



第三种最优化问题表示形式为

 （1.1）

其中．

上述三种表示形式中，称为集约束．在所讨论的最优化问题中，集约束是无关紧要的．这是因为一般，不然的话，通常也可用不等式约束表达出来．因此今后一般不再考虑集约束．

满足所有约束的点称为容许点或可行点．容许点的集合称为容许集或可行域．可用



表示．

一般地，对于最优化问题（1.1）的求解，是指在可行域内找一点，使得目标函数在该点取得极小值，即



这样的称为问题（1.1）的最优点，也称极小点，而相应的目标函数值称为最优值；合起来，称为最优解，但习惯上常把本身称为最优解．最优点的各分量和最优值必须是有限数．

## §1.2 最优化问题的算法

### 一、二维最优化问题的图解法

二维最优化问题具有鲜明的几何解释，这对于理解有关理论和掌握所述的方法是很有益处的．下面讨论的二维最优化问题为



#### （一）约束集合

当约束函数为线性时，等式约束在的坐标平面上为一条直线；不等式约束在的坐标平面上为一半平面．当约束函数（例如）为非线性时，则等式约束条件在的坐标平面上为一条曲线（如图1.2所示）．当约束函数（例如）为非线性时，则不等式约束在的坐标平面上为曲线把坐标平面分成两部分当中的一部分（如图1.3所示）．

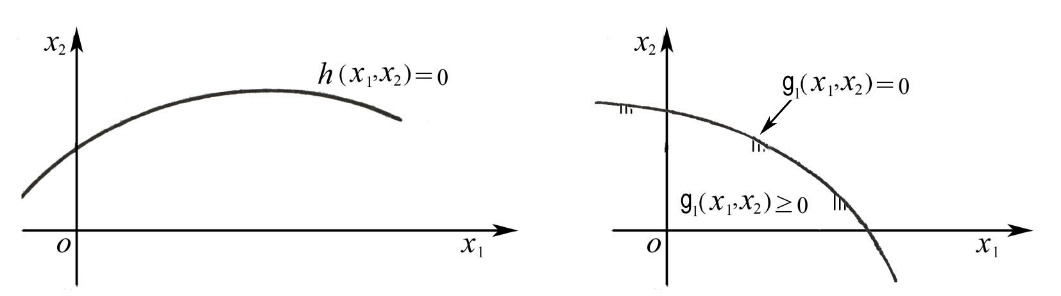


图1.2 图1.3

综上所述，当把约束条件中的每一个等式所确定的曲线，以及每一个不等式所确定的部分在坐标平面上画出之后，它们相交的公共部分即为约束集合*D*．

例1.4 在坐标平面上画出约束集合

．

解 满足的区域为以原点为圆心，半径为1的圆盘；满足的区域为第一象限中的扇形（如图1.4所示）．

#### （二）等高线

我们知道



在三维空间中表示一张曲面．（其中为常数）在三维空间中表示平行于平面的一个平面，这个平面上任何一点的高度都等于常数（如图1.5所示）．

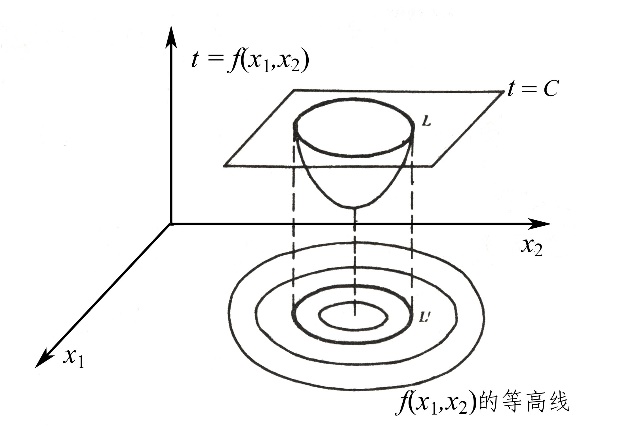
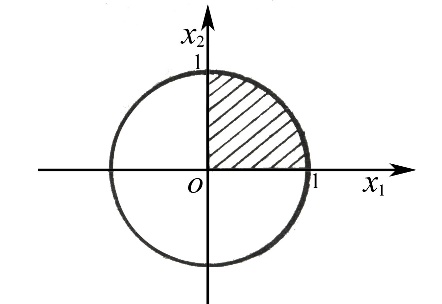


图1.4 图1.5

现在，在三维空间中曲面与平面有一条交线．交线在平面上的投影曲线是，可见曲线上的点到平面的高度都等于常数，也即曲线上的的函数值都具有相同的值．当常数取不同的值时，重复上面的讨论，在平面上得到一簇曲线——等高线．不难看出，等高线的形状完全由曲面的形状所决定；反之，由等高线的形状也可以推测出曲面的形状．在以后的讨论中，不必具体画出曲面的图形，只须在平面上变动常数画出曲线簇．

例1.5 在坐标平面上画出目标函数的等高线．

解 因为当取时，曲线表示是以原点为圆心，半径为的圆．因此等高线是一簇以原点为圆心的同心圆（如图1.6所示）．

|  |
| --- |
| 图1.6 |

#### （三）几何意义及图解法

当在坐标平面上分别画出约束集合*D*以及目标函数的等高线后，不难求出二维最优化问题的最优解．下面举例说明．

例1.6 用图解法求解二维最优化问题



解 由例1.4得到约束集合*D*（如图1.7所示）．目标函数的等高线是以为圆心的同心圆，并且这簇同心圆的外圈比内圈的目标函数值大．因此，这一问题成为在约束集合中找一点，使其落在半径最小的那个同心圆上．不难看出，问题的最优解．

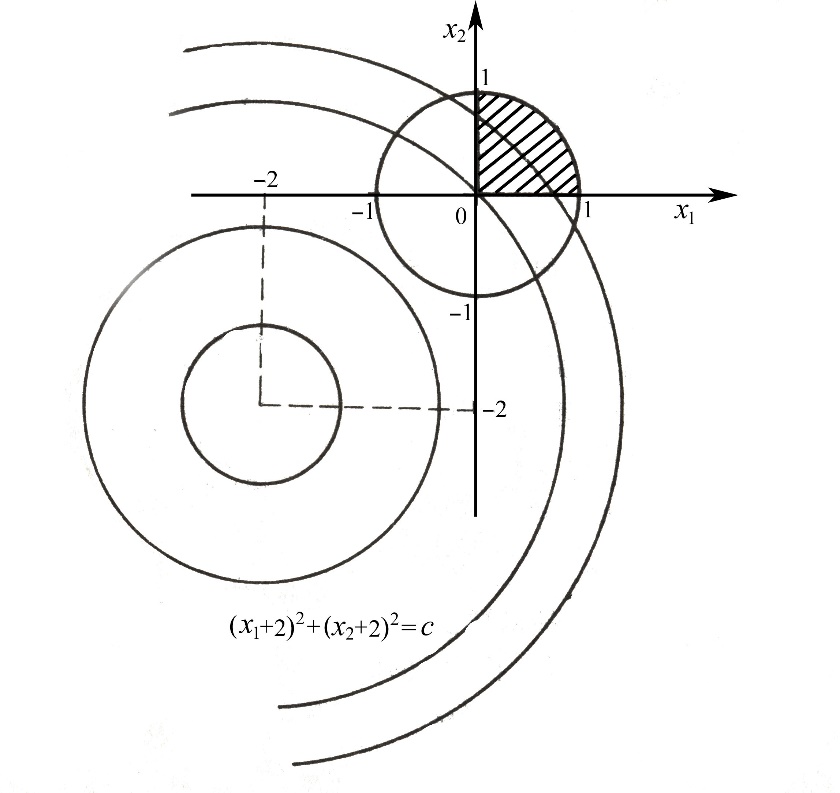


图1.7

### 二、最优化问题的迭代解法

#### （一）迭代方法

在经典极值问题中，解析法虽然具有概念简明，计算精确等优点，但因只能适用于简单或特殊问题的寻优，对于复杂的工程实际问题通常无能为力，所以极少使用．

最优化问题的迭代算法是指：从某一选定的初始点出发，根据目标函数、约束函数在该点的某些信息，确定本次迭代的一个搜索方向和适当的步长，从而到达一个新点，用式子表示即为

 （1.2）

式中代表前一次已取得的迭代点，在开始计算时为迭代初始点，代表新的迭代点，代表第次迭代计算的搜索方向，代表第次迭代计算的步长因子．

按照式（1.2）进行一系列迭代计算所根据的思想是所谓的“爬山法”，就是将寻求函数极小点（无约束或约束极小点）的过程比喻为向“山”的顶峰攀登的过程，始终保持向“高”的方向前进，直至达到“山顶”．当然“山顶”可以理解为目标函数的极大值，也可以理解为极小值，前者称为上升算法，后者称为下降算法．这两种算法都有一个共同的特点，就是每前进一步都应该使目标函数有所改善，同时还要为下一步移动的搜索方向提供有用的信息．如果是下降算法，则序列迭代点的目标函数值必须满足下列关系

．

如果是求一个约束的极小点，则每一次迭代的新点都应该在约束可行域内，即



图1.8为迭代过程示意图．

由上面的迭代过程可知，在迭代过程中有两个规则需要确定：一个是搜索方向的选取；一个是步长因子的选取．一旦和的选取方法确定，则一种迭代算法就确定，即不同的规则就对应不同的最优化方法．

#### （二）收敛速度与计算终止准则

##### （1）收敛速度

作为一个算法，能够收敛于问题的最优解当然是必要的，但仅收敛还不够，还必须能以较快的速度收敛，这才是好的算法．

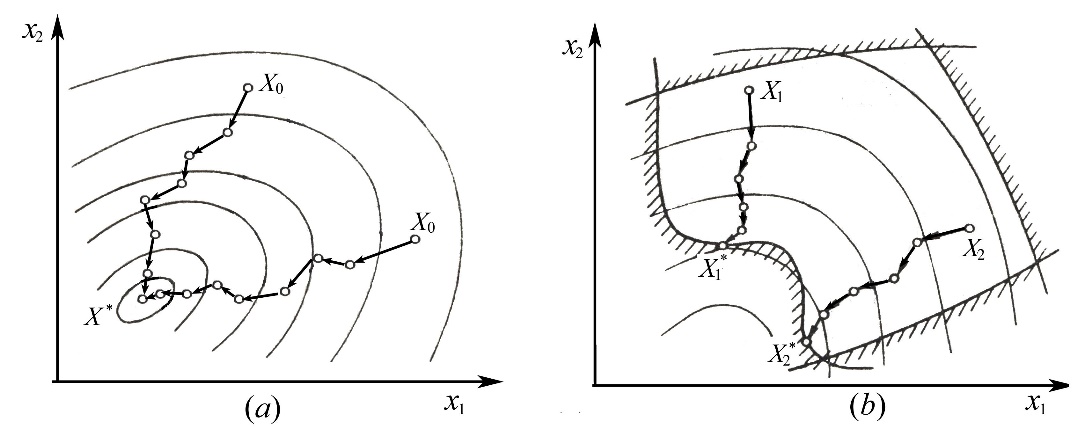


图1.8

定义1.1 设由算法*A*产生的迭代点列在某种“||·||”的意义下收敛

于点，即，若存在实数及一个与迭代次数无关

的常数，使得



则称算法*A*产生的迭代点列具有阶收敛速度，或称算法*A*为阶收敛的．特别地：

1. 当时，称迭代点列具有线性收敛速度或称算法*A*为线性收敛(最慢的)的．
2. 当时，或时，称迭代点列具有超线性收敛速度或称算法*A*是超线性收敛．
3. 当时，迭代点列叫做具有二阶收敛速度或算法*A*是二阶收敛(最快的)的．

一般认为，具有超线性收敛或二阶收敛的算法是较快速的算法．

例1.7 设一算法*A*产生迭代点列，它收敛于点，试判定算法*A*的收敛速度．

解 因为 ，

即取 ．

所以算法*A*具有线性收敛速度．

例1.8 设一算法*A*产生迭代点列，它收敛于，试确定*A*的收敛速度．

解 因为



，

即取．

所以*A*是超线性收敛的．

##### （2）计算终止准则

用迭代方法寻优时，其迭代过程不能无限制地进行下去，那么什么时候截断这种迭代呢？这就是迭代什么时候终止的问题．

从理论上说，当然希望最终迭代点到达理论极小点，或者使最终迭代点与理论极小点之间的距离足够小时才终止迭代．但是这在实际上是办不到的．因为对于一个待求的优化问题，其理论极小点在哪里并不知道．所知道的只是通过迭代计算获得的迭代点列，因此只能从点列所提供的信息来判断是否应该终止迭代．

对于无约束优化问题通常采用的迭代终止准则有以下几种：

①点距准则

相邻两迭代点之间的距离已达到充分小，即

，

上式中是一个充分小的正数，代表计算精度．

②函数下降量准则

相邻两迭代点的函数值下降量已达到充分小．当时，可用函数绝对下降量准则

．

当时，可用函数相对下降量准则

．

③梯度准则

目标函数在迭代点的梯度已达到充分小，即

．

这一准则对于定义域上的凸函数是完全正确的．若是非凸函数，有可能导致误把驻点作为最优点．（凸函数的定义请参见第二章2.6节）

对于约束优化问题，不同的优化方法有各自的终止准则．

综上所述，优化算法的基本迭代过程如下：

① 选定初始点，置．

② 按照某种规则确定搜索方向．

③ 按某种规则确定使得

．

④ 计算

．

⑤ 判定是否满足终止准则．若满足，则打印和，停机；否则置，转②．

上述算法用框图表达如图1.9．

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | *X*是否满足终止准则 | |

|  |  |
| --- | --- |
| |  | | --- | | 输出*X*， *f*(*X*) | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 结束 | |

|  |  |
| --- | --- |
| |  | | --- | | 选定*X*0 | |

|  |  |
| --- | --- |
| |  | | --- | | 确定*P* | |

|  |  |
| --- | --- |
| |  | | --- | | 确定*t*，使得*f* (X0+*t P*)< *f* (X0) | |

|  |  |
| --- | --- |
| |  | | --- | | *X*=*X*0+*t P* | |

|  |  |
| --- | --- |
| |  | | --- | | *X*0=*X* | |

## §1.3 最优化算法分类

所谓优化算法，其实就是一种搜索过程或规则，它是基于某种思想和机制，通过一定的途径或规则来得到满足用户要求的问题的解．

就优化机制与行为而分，目前工程中常用的优化算法主要可分为：经典算法、构造型算法、改进型算法、基于系统动态演化的算法和混合型算法等．

（1）经典算法．包括线性规划、动态规划、整数规划和分枝定界等运筹学中的传统算法，其算法计算复杂性一般很大，只适于求解小规模问题，在工程中往往不实用．

（2）构造型算法．用构造的方法快速建立问题的解，通常算法的优化质量差，难以满足工程需要．譬如，调度问题中的典型构造型方法有：Johnson法、Palmer法、Gupta法、CDS法、Daunenbring的快速接近法、NEH法等．

（3）改进型算法，或称邻域搜索算法．从任一解出发，对其邻域的不断搜索和当前解的替换来实现优化．根据搜索行为，它又可分为局部搜索法和指导性搜索法．

①局部搜索法．以局部优化策略在当前解的邻域中贪婪搜索，如只接受优于当前解的状态作为下一当前解的爬山法；接受当前解邻域中的最好解作为下一当前解的最陡下降法等．

②指导性搜索法．利用一些指导规则来指导整个解空间中优良解的探索，如SA、GA、TS等．

（4）基于系统动态演化的算法．将优化过程转化为系统动态的演化过程，基于系统动态的演化来实现优化，如神经网络和混沌搜索等．

（5）混合型算法．指上述各算法从结构或操作上相混合而产生的各类算法．

优化算法当然还可以从别的角度进行分类，如确定性算法和不确定性算法，局部优化算法和全局优化算法等．

## §1.4 组合优化问题简介

### 一、组合优化问题建模

优化问题涉及的工程领域很广，问题种类与性质繁多，归纳而言，最优化问题可分为函数优化问题和组合优化问题两大类，上一节介绍的最优化数学模型属于函数优化问题，该函数优化的对象是一定区间内的连续变量，而组合优化的对象则是解空间中的离散状态．本节重点介绍组合优化问题．

组合优化问题是通过对数学方法的研究去寻找离散事件的最优编排、分组、次序或筛选等，所研究的问题涉及信息技术、经济管理、工业工程、交通运输、通信网络等诸多领域，该问题数学模型可表示为



其中为目标函数，和为约束函数，*X*为决策变量，表示有限个点组成的集合．

一个组合优化问题可用3个参数表示，其中表示决策变量的定义域，*D*表示可行解区域，*D*中的任何一个元素称为该问题的可行解，表示目标函数，满足的可行解称为该问题的最优解．组合最优化问题的特点是可行解集合为有限集．由直观可知，只要将中有限个点逐一判别是否满足约束条件和比较目标函数值的大小，该问题的最优解一定存在并可以求得，下面是三个典型的组合优化问题．

例1.9 0-1背包问题（knapsack problem）

设有一个容积为的背包，个体积分别为，价值分别为的物品，如何以最大的价值装包?这个问题称为0-1背包问题．用数学模型表示为

， （1.3）



其中目标（1.3）欲使包内所装物品的价值最大，式（1.4）为包的能力限制，式（1.5）表示为二进制变量，表示装第*i*个物品，则表示不装．

例1.10 旅行商问题（TSP，traveling salesman problem）

一个商人欲到个城市推销商品，每两个城市*i*和之间的距离为，如何选择一条道路使得商人每个城市走一遍后回到起点且所走路径最短．TSP还可以细分为对称和非对称距离两大类问题．当对任意的时都有 ，则称该TSP为对称距离TSP，否则称为非对称距离TSP．对一般的TSP，它的一种数学模型描述为

， （1.6）



以上是基于图论的数学模型．其中式（1.10）中的决策变量=1表示商人行走的路线包含从城市到城市的路径，表示商人没有选择走这条路．的约束可以减少变量的个数，使得共有个决策变量．目标（1.6）要求距离之和最小．式（1.7）要求商人从城市出来一次，式（1.8）要求商人走入城市只有一次．式（1.7）和式（1.8）表示每个城市经过一次．仅有式（1.7）和式（1.8）的约束无法避免回路的产生，一条回路是由个城市和条弧组成，因此，式（1.9）约束旅行商在任何一个城市子集中不形成回路，其中表示集合中元素个数．

例1.11 聚类问题

*m*维空间上的个模式要求聚类成类，使得各类本身内的点最相近，即，其中为第类的中心，，，为第类中的点数．

### 二、算法复杂性

前面给大家介绍的三个组合优化问题例子，模型建立都比较简单，但要求它们的最优解却很困难，而解模型的困难主要原因是所谓的“组合爆炸”，如聚类问题的可能划分方式有个，TSP问题有个．显然状态数量随问题规模呈超指数增长，若计算机每秒处理1亿种排列，则列举20个城市问题的20！种排列约需几百年．如此巨大的计算量是无法承受的，更不用谈更大规模问题的求解，因此解决这些问题的关键在于寻求有效的优化算法，也正是由于组合优化问题算法的复杂性，激起了人们对它的理论与算法研究的兴趣．

算法的复杂性是指算法对时间复杂性和对空间复杂性．按照算法复杂性求解的难易程度，可把组合优化问题分为P类，NP类和NP完全类．

算法或问题的复杂性一般表示为问题规模（如TSP问题中的城市数）的函数，时间的复杂性记为，空间的复杂性记为．在算法分析和设计中，沿用实用性的复杂性概念，即把求解问题的关键操作，如加、减、乘，比较等运算指定为基本操作，算法执行基本操作的次数则定义的算法的时间复杂性，算法执行期间占用的存储单元则定义为算法的空间复杂性．P类问题指具有多项式时间求解算法的问题类．许多优化问题仍没有找到求得最优解的多项式时间算法，称这种比P类问题更广泛的问题为非确定型多项式算法的问题类，即NP问题．

### 三、NP完全问题

离散问题的求解常常要从有限个方案中选出一个满意的结果来 ，也许有人认为，从有限个方案中挑选一个，总是比较容易的．然而，事实并非如此，关键在于问题的规模．由于计算机的出现，人们对问题的求解在观念上发生了改变，一个在理论上可解的问题如果在求解时需要花费相当多，以至于不合理的时间（如几百年甚至更长时间），我们不能认为它已解决，而应当努力寻找更好的算法．

如何比较算法的好坏呢？从不同的角度出发可以有不同的回答．这里，仅就算法的计算速度作一个十分粗略的比较．

设有一台每小时能进行*M*次运算的计算机．并设问题已有两种不同的算法，算法A对规模为的问题约需作次运算，算法*B*则约需作次运算．运用算法*A*在一个小时内大约可解一个规模为的问题，而算法*B*则大约可解一个规模为的问题．

现在假设计算机有了改进，例如计算速度提高了100倍．此时，利用算法*A*能求解的问题规模增大了10倍，利用算法*B*可解的问题规模只增加了．前者得到了成倍的增加，而后者则几乎没有什么改变，今天无法求解的问题，将来也很少有希望解决．由于这一原因，对算法作如下分类．

定义1.2（多项式算法） 设*A*是求解某类问题*D*的一个算法，为问题*D*的规模，用表示用算法*A*在计算机上求解这一问题时需作的初等运算的次数．若存在一个多项式和正整数*N*，当时，总有（不论求解的*D*是怎样的具体实例），则称算法*A*是求解问题*D*的一个多项式算法．

定义1.3（指数算法） 设算法*B*是求解某类问题*D*的一个算法，若存在一个常数，对任意，总可以找到问题*D*的一个规模为的实例，用算法*B*求解时，所需的计算量约为，则称*B*为求解问题*D*的一个指数算法．

多项式算法被称为是“好”算法（或有效算法），而指数算法则一般认为是“坏”算法，因为它只能用来求解规模很小的问题．

这样看来，对一个问题只有在找到求解它的多项式算法后才能较为放心．然而十分可惜的是，对于许多具有广泛应用价值的离散模型，人们至今仍未找到多项式算法．现在的任何算法在最坏的情况下计算量均可达到或接近．

1971年和1972年，S. Cook和R. Karp分别发表了相关论文，奠定了NP完全理论基础．Cook指出，NP完全类问题，具有两个性质：

（1）这类问题中的任何一个问题至今均未发现有多项式算法．

（2）只要其中任一个问题找到了多项式算法，那么其他所有问题也就有了多项式算法．

现在，NP完全类中的问题已被扩充到了四千多个，其中包括前面讨论的旅行商问题．对它们的研究使人们越来越相信这样一个猜测：对这类问题也许根本不存在多项式算法．

# 第二章 最优化问题数学基础

为了便于学习最优化方法，本章将对与优化方法密切相关的数学知识作一简要介绍，而有些数学知识将在讲解各种算法时，随之介绍．

## §2.1 二次型与正定矩阵

### 一、二次型与实对称矩阵

二次型理论在最优化设计中应用十分广泛．应用矩阵的乘法运算，二次型与实对称矩阵紧密地联系在一起了，从而二次型的基本问题又可转化成实对称矩阵问题．

二次型理论问题起源于化二次曲线和二次曲面的方程为标准形式的问题．推广到*n*维空间中，二次超曲面的一般方程为



用矩阵表示可简记为



其中矩阵*A*的元素正是二次型的项的系数的一半，是二次型的项的系数．因此，二次型和它的矩阵*A*是相互唯一决定的，且．

### 二、正定矩阵

定义2.1 如果二次型



对于任何一组不全为零的数恒有

，

则称正定，且二次型矩阵*A*也称为正定．

简言之，一个对称矩阵*A*如果是正定的，则二次型对于所有非零向量*X*其值总为正．类似可以给出定义，若二次型，则*A*为半正定矩阵；若，则*A*为半负定矩阵；若二次型既不是半正定又不是半负定，就称矩阵*A*为不定的．

矩阵*A*为正定的充要条件是它的行列式的顺序主子式全部大于零，即

．

由此可见，正定矩阵必然是非奇异的．

例2.1 判断矩阵是否正定．

，

§2.2 方向导数与梯度

一、方向导数

所谓方向导数的概念是作为偏导数的一个推广而引入的，它主要研究函数沿任一给定方向的变化率．

定义2.2 设在点处可微，*P*是固定不变的非零向量，是方向*P*上的单位向量，则称极限

 （2.1）

为函数在点处沿*P*方向的方向导数，式中是它的记号．

定义2.3 设是连续函数，，且，若存在，当时都有，则称*P*为在点处的下降方向．若，则称*P*为在点处的上升方向．

由以上两个定义可立刻得到如下的结论：

若，则从出发在附近沿*P*方向是下降；若，则从出发在附近沿*P*方向是上升．

事实上，若，则当充分小时，根据式（2.1）必有

，

即 ，

其中是从出发在*P*方向上的点，说明是下降的．同理可以说明，若，则是上升的．

二、梯度

定义2.4 以的*n*个偏导数为分量的向量称为在*X*处的梯度，记为



梯度也可以称为函数关于向量的一阶导数．

以下几个特殊类型函数的梯度公式是常用的：

（1）若（常数），则，即；

（2）．

证 设，则

．

于是的第个分量是



所以．

（3）．

（4）若是对称矩阵，则

．

三、梯度与方向导数之间的关系

定理2.1 设在点处可微，则

，

其中是方向上的单位向量．

证明在相应的数学分析中可找到（故略去）．

由这个定理容易得到下列结论：

(1) 若，则*P*的方向是函数在点处的下降方向；

(2) 若，则****的方向是函数在点处的上升方向．

方向导数的正负决定了函数值的升降，而升降的快慢就由它的绝对值大小决定．绝对值越大，升降的速度就越快，即



=，

上式中的等号，当且仅当的方向与的方向相同时才成立．

由此可得如下重要结论(如图2.1所示）：

(1)梯度方向是函数值的最速上升方向；

(2)函数在与其梯度正交的方向上变化率为零；

(3)函数在与其梯度成锐角的方向上是上升的，而在与其梯度成钝角的方向上是下降的；

(4)梯度反方向是函数值的最速下降方向．

对于一个最优化问题，为了尽快得到最优解，在每一步迭代过程中所选取的搜索方向****总是希望它等于或者是靠近于目标函数的负梯度（即－）

|  |
| --- |
| 图2.1 |

的方向，这样才能使函数值下降的最快．

例2.2 试求目标函数在点处的最速下降方向，并求沿这个方向移动一个单位长后新点的目标函数值．

**解** 因为

，

所以最速下降方向是．

这个方向上的单位向量是

．

故新点是

，

对应目标函数值为．

## §2.3 海色矩阵及泰勒展式

### 一、海色（Hesse）矩阵

前面说过，梯度是关于的一阶导数，现在要问关于的二阶导数是什么？

定义 2.5 设：，，如果在点处对于自变量的各分量的二阶偏导数



都存在，则称函数在点处二阶可导，并且称矩阵



是在点处的Hesse矩阵．

在数学分析中已经知道，当在点处的所有二阶偏导数为连续时有



因此，在这种情况下Hesse矩阵是对称的．

例2.3 求目标函数的梯度和Hesse矩阵．

解 因为

，

所以 ．

又因为



所以 ．

例2.4 设，求线性函数



在任意点*X*处的梯度和Hesse矩阵．

解 设，则

，

 （2.2）

∴ ．

由式（2.2）进而知



∴ （阶零矩阵）．

例2.5 设是对称矩阵，，求二次函数

在任意点处的梯度和Hesse矩阵．

解 设，，则



将它对各变量求偏导数，得



∴ ．

在上式中显然



再对它们求偏导数得



∴ ．

以上例子说明，元函数求导与一元函数的求导在形式上是一致的，即线性函数的一阶导数为常向量，其二阶导数为零矩阵；而二次函数的一阶导数为线性向量函数，二阶导数为常矩阵．

最后介绍在今后的计算中要用到的向量函数的导数．

定义2.6 设，记，如果在点处对于自变量的各分量的偏导数都存在，则称向量函数在点处是一阶可导的，并且称矩阵



是在点处的一阶导数或Jacobi矩阵，简记为

．

由于元函数的梯度是向量函数



所以的一阶导数或Jacobi矩阵为





即 ．

据此，从上式得知，函数梯度的Jacobi矩阵即为此函数的Hesse矩阵．

下面给出今后要用到的几个公式：

（1），其中是分量全为常数的维向量，是阶零矩阵．

（2），其中是维向量，是阶单位矩阵．

（3），其中是阶矩阵．

（4）设，其中，，则

．

### 二、泰勒展开式

多元函数的泰勒展开在最优化方法中十分重要，许多方法及其收敛性的证明是从它出发的，这里给出泰勒展开定理及其证明．

定理2.2 设具有二阶连续偏导数，则

， （2.3）

其中，而．

证 设，于是

，．

对按一元函数在点展开，得到

，

其中．令，于是

． （2.4）

又因为，，代入式（2.4）中，所以

．

## §2.4 极小点的判定条件

函数在局部极小点应满足什么条件？反之，满足什么条件的是局部极小点？这是我们关心的基本问题．

下面针对多元函数的情形给出各类极小点的定义．

定义2.7 对于任意给定的实数，满足不等式的的集合称为点的邻域，记为

．

定义2.8 设，若存在点和数，都有，则称为的局部极小点（非严格）．

定义2.9 设，若存在点和数，但，都有，则称为的严格局部极小点．

定义2.10 设，若存在点，都有，则称为在*D*上的全局极小点（非严格）．

定义2.11 设，若存在点，但，都有，则称为在*D*上的严格全局极小点．

由以上定义看到，是局部极小点，是指在以为中心的一个邻域中在点处取得最小的值；而是全局极小点，是指在定义域*D*中在点处取得最小的值．全局极小点可能在某个局部极小点处取得，也可能在*D*的边界上取得．实际问题通常是求全局极小点，但是直到目前为止，最优化中绝大多数方法都是求局部极小点的，解决这一矛盾的一种方法是先求出所有的局部极小点，再求全局极小点．

定理2.3 设具有连续的一阶偏导数．若是的局部极小点并且是*D*的内点，则

． （2.5）

证 设是任意单位向量，因为是的局部极小点，所以存在，当或时总有

． （2.6）

引入辅助一元函数，此时，由式（2.6）得．又因是*D*的内点，所以与它对应的是的局部极小点．又根据一元函数极小点的必要条件，得到，即．再由单位向量的任意性得．

这里条件（2.5）仅仅是必要的，而不是充分的．例如在点处的梯度是，但是双曲面的鞍点，而不是极小点（如图2.2所示）．

定义2.12 设是*D*的内点．若，则称为的驻点．

定理2.4 设具有连续二阶偏导数，是*D*的一个内点．若，并且是正定的，则是的严格局部极小点．

证 因为是正定矩阵，则必存在，使得对于所有的都有



（参看高等代数二次型理论）．现在将在点处按泰勒公式展开，并注意到，于是可得



当充分接近（但）时，上式左端的符号取决于右端第一项，因此．

一般说来，这个定理仅具有理论意义．因为对于复杂的目标函数，Hesse矩阵不易求得，它的正定性就更难判定了．

定理2.5 若多元函数在其极小点处的Hesse矩阵是正定的，则它在这个极小点附近的等值面近似地呈现为同心椭球面簇．

证 设是多元函数的极小点，并设是充分靠近极小点的一个等值面，即充分小．把在点展成泰勒公式，即



右端第二项因是极小点有而消失．如果略去第4项，那么

，

又因为，所以



（2.7）

按假设正定，由二次型理论知式（2.7）是以为中心的椭球面方程．

## §2.5 锥、凸集、凸锥

在本节中，给出维Euclid空间中的锥、凸集和凸锥的定义，以及与其相关的一些概念和性质．

### 一、定义与简单性质

定义2.13 集合．若及任意的数，均有，则称*C*为锥．

定义2.14 设是中的个已知点．若对于某点存在常数且使得，则称是的凸组合．若且，则称是的严格凸组合．

定义2.15 集合．若和，以及任­意的数，均有



则称*C*为凸集．

定义2.16 设且，，则集合称为中的半空间．

特别地，规定空集是凸集．容易验证，空间、半空间、超平面、直线、点、球都是凸集．

定理2.6 任意一组凸集的交仍然是凸集．

证 设，其中*I*是的下标集，都是凸集．任取，则对于任意都是．任取且，因是凸集，有．于是，即*C*是凸集．

若集合*C*为锥，*C*又为凸集，则称*C*为凸锥．若*C*为凸集，也为闭集，则称*C*为闭凸集．若*C*为凸锥，也为闭集，则称*C*为闭凸锥．

由数学归纳法不难证明如下的定理2.7和2.8．

定理2.7 集合*C*为凸集的充分必要条件是，及任意数（），，有．

定理2.8 集合*C*为凸锥的充分必要条件是，及任意数（），均有

．

定义 2.17 有限个半空间的交称为多面集，其中为矩阵，为维向量．

例2.6 集合



为多面集，其几何表示如图2.3画斜线部分．

|  |
| --- |
| 图2.3 |

在多面集的表达式中，若，则多面集也是凸锥，称为多面锥．

在有关凸集的理论及应用中，极点和极方向的概念有着重要作用．

定义 2.18 设为非空凸集，，若不能表示成中两个不同点的凸组合；换言之，若假设，，必推得，则称是凸集的极点．

|  |
| --- |
| 图2.4 |

按此定义，图2.4中多边形的顶点和是极点，而和不是极点．图2.4中圆周上的点均为极点．

由图2.4可以看出，在给定的两个凸集中，任何一点都能表示成极点的凸组合．

定义 2.19 设为中的闭凸集，为非零向量，如果对中的每一个，都有射线，则称向量为的方向．又设和是的两个方向，若对任何正数，有，则称和是两个不同的方向．若的方向不能表示成该集合的两个不同方向的正的线性组合，则称为的极方向．

|  |
| --- |
| 图2.5 |

例2.7 如图2.5所示，对于集合

，

凡是与向量夹角小于或等于的向量，都是它的方向．和是的两个极方向．的其它方向都能表示成这两个极方向的正线性组合．

例2.8 设为非空集合，是非零向量．证明为的方向的充要条件是且．

证 按照定义2.19，为的方向的充要条件是对每一个，有

． （2.8）

根据集合的定义，由式（2.8）可得

， （2.9）

． （2.10）

由于，及可取任意非负数，因此由式（2.9）和式（2.10）知及．

对于有界多面集，它的每一个点可用极点的凸组合来表示，反之，由极点的凸组合表示的点一定属于这个多面集．对此可简要说明如下：

设有多面集，如图2.6(a)所示．若，则



其中均为非负数，且

，

即为极点，和的凸组合．

反之，设

，，，

则，即．

如果多面集是无界的，那么对于该多面集中的任一点（极点除外），只用极点来表示是不行的，还需要用极方向．如图2.6(b)所示，这时有

，

其中是中某个数，是某一个非负数．

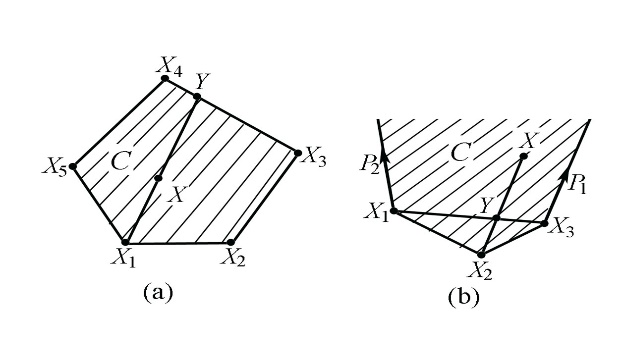


图2.6

概括起来，有下列定理：

定理2.9 设为非空多面集，则有

（1）极点集非空，且存在有限个极点

（2）极方向集合为空集的充要条件是有界．若无界，则存在有限个极方向

（3）的充要条件是

 ，

其中，，．

关于上述定理的证明参见参考文献[6]．

### 二、凸集分离定理

凸集分离定理是凸分析中最重要的定理之一，它在最优化理论和模型当中具有重要的应用．

所谓集合的分离是指对于两个集合*C*1和*C*2存在一个超平面*H*，使得*C*1在*H*的一边，而*C*2在*H*的另一边．如果超平面方程为，那么对位于*H*某一边的点必有，而对位于*H*另一边的必有．

定义2.20 设*C*1和*C*2是中的两个非空集合，是超平面，若对于每一个都有，对于每一个都有（或情况恰好相反），则称超平面*H*分离集合*C*1和*C*2．

定理2.10 若*C*为闭凸集，，则存在，，以及数，对，有，并且存在，使得．

定理2.11 设*C*为凸集，，则存在，，使得，有

．

定理2.12 设*C*为闭凸集，则*C*可表为所有包含*C*的半空间的交，即

，

其中

．

上述定理的证明过程参见参考文献[6]．

## §2.6 凸函数

### 一、各类凸函数定义及性质

设函数定义在凸集*C*上，其中．

定义2.21 若存在常数，使得，以及，若有

，

则称为一致凸函数；若有

，

则称为严格凸函数；若有

，

则称为凸函数．

定义2.22 设为可微函数．若，当



都有

，

则称为伪凸函数．

定义2.23 对，且，以及，若

，

则称为严格拟凸函数；

定义2.24 对，以及，若

，

则称为拟凸函数；

定义2.25 对，且，以及，若

，

则称为强拟凸函数．

定理2.13 若为一致凸函数，则为严格凸函数．

证：设为一致凸函数，则，，及，有



，

即为严格凸函数．

定理2.14 若为严格凸函数，则为凸函数．

定理2.15 设为可微函数．若为凸函数，则为伪凸函数．

定理2.16 设为伪凸函数，则为严格拟凸函数．

定理2.17 设为下半连续的严格拟凸函数，则为拟凸函数．

定理2.18 若为严格凸函数，则为强拟凸函数．

定理2.19 设为强拟凸函数，则为严格拟凸函数．

下半连续的定义及定理2.14—定理2.19的证明过程参见参考文献[6]．上述几种凸函数有图2.7所示的关系．

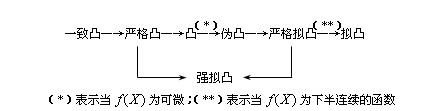


图2.7

凸函数与凸集之间有如下关系：

定理2.20 设，其中*C*为非空凸集．若是凸函数，则对于任意实数，水平集是凸集．

证 若是空集，则是凸集．以下设非空，任取，则．设且，由是凸函数知

，

即，所以是凸集．

判定一个函数是否为凸函数，一般说来是比较困难，但当函数可微时，有如下几个定理可供使用．

定理2.21 设是可微函数，其中*C*为凸集．则

（1）为凸函数的充要条件是，都有

； （2.11）

（2）为严格凸函数的充要条件是，且，都有

．

证 （1）先证必要性 ．已知是*C*上的凸函数，要证式（2.11）．由凸函数定义知，对满足的任意数都有

，

令，则．代入上式中，经移项可得

． （2.12）

令****，由的可微性，利用一阶泰勒展式、方向导数定义及式（2.12），可得

．

这就证明了式（2.11）．

再证充分性．任取一对数且．考虑点，根据充分性假设，应有

，

．

两式分别乘以和后相加，得到



．

由凸函数定义知，是*C*上的凸函数．

（2）充分性可依照（1）的充分性证得，下证必要性．

因为严格凸函数本身是凸函数，所以，且，都有



以下证明式中只能取“>”号．假设存在，且，使得

． （2.12）

取，由的严格凸性，有

． （2.13）

把式（2.12）代入式（2.13）中，经整理得

．

根据本定理（1）部分结论得知，此式与是凸函数相矛盾．

定理2.22 设是二次可微函数，*C*为非空开凸集，则为*C*上凸函数的充要条件是Hesse矩阵在*C*上任意点均半正定．

证明略．

定理2.23 设是二次可微函数，*C*为非空凸集．若Hesse矩阵在*C*上任意点均正定，则在*C*上为严格凸函数．

证明略，需要注意，该定理的逆命题不真．

例如在上为严格凸函数，但是它的Hesse矩阵在点处是半正定的．

### 二、凸规划

定义2.26 设，其中是非空凸集，是凸函数，则形式为的问题称为凸规划问题．

更进一步，设

若都是上的凸函数，都是上的线性函数，则容易验证*C*是凸集．事实上，因为都是凸函数，根据定理2.20集合也都是凸集．

此外，超平面，也都是凸集．显然，*C*是的交集，根据定理2.6，*C*是凸集．于是，在这种情况下凸规划问题又可表示成如下形式：



如下定理指明凸规划的一个重要性质．

定理2.24 设是凸规划问题的局部极小点，

（1）若是凸函数，则是凸规划问题全局极小点；

（2）若是严格凸函数，则是凸规划问题的唯一全局极小点．

证（1）使用反证法．假设不是全局极小点，则必存在使得．对于与的任意凸组合，其中且，根据的凸性，有





由此看到，当充分小时，充分接近，注意到此时也有，而这与是局部极小点相矛盾．因此必是全局极小点．

（2）假设不是唯一全局极小点．必存在但使得．考虑中点．由的严格凸性，有．此式与为全局极小点相矛盾．这就证明了唯一性．

定义2.27 形式为

 （2.14）

的函数称为元二次函数，其中

，

这里的*A*是对称矩阵，即．

若*A*为正定，则称（2.14）为正定二次函数．注意到，由定理2.23知，正定二次函数是严格凸函数，在最优化算法构造中它起着特殊的作用．

定义2.28 形式为

 （2.15）

的问题称为二次规划问题，其中是矩阵，是矩阵．

## §2.7 约束问题的最优性条件

所谓最优性条件就是最优化问题的目标函数与约束函数在最优点处满足的充要条件．这种条件对于最优化算法的终止判定和最优化理论推证都是至关重要的．最优性必要条件是指在最优点处满足哪些条件；充分条件是指满足哪些条件的点是最优点．本节仅讲述最基本的结论．

### 一、约束最优解

对约束优化问题的求解，其目的是在由约束条件所规定的可行域内，寻求一个目标函数值最小的点及其函数值．这样的解称为约束最优解．约束最优点除了可能落在可行域内的情况外，更常常是在约束边界上或等式约束曲面上，因此它的定义及它的一阶必要条件与无约束优化问题不同．

#### （一）约束优化问题的类型

约束优化问题根据约束条件类型的不同分为三种，其数学模型如下：

（1）不等式约束优化问题（IP型）

 （2.16）

（2）等式约束优化问题（EP型）



（3）一般约束优化问题（GP型）



#### （二）约束优化问题的局部解与全局解

按一般约束优化问题，其可行域为

．

若对某可行点存在，当与它邻域的点之距离时，总有则称为该约束优化问题的一个局部最优解．

下面以一个简单例子说明．设有



该问题的几何图形如图2.8所示．从图上的目标函数等值线和不等式约束与等式约束的函数曲线可写出它的两个局部最优解．这是因为在点邻域的任一满足约束的点，都有；同理，亦然．

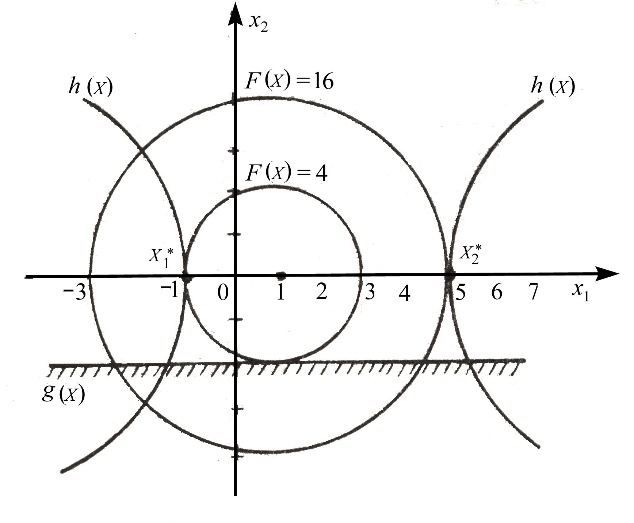
１

图2.8

对某些约束优化问题，局部解可能有多个．在所有的局部最优解中，目标函数值最小的那个解称为全局最优解．

在上例中，由于，所以全局最优解为．

由此可知，约束优化问题全局解一定是局部解，而局部解不一定是全局解．这与无约束优化问题是相同的．

### 二、约束优化问题局部解的一阶必要条件

对于约束，现在进一步阐明起作用约束与不起作用约束的概念．一般的约束优化问题，其约束包含不等式约束和等式约束．在可行点处，如果有，则该约束称可行点的起作用约束；而如果有，则该约束称可行点的不起作用约束．对于等式约束，显然在任意可行点处的等式约束都是起作用约束．

在某个可行点处，起作用约束在的邻域内起到限制可行域范围的作用，而不起作用约束在处的邻域内就不产生影响．因此，应把注意力集中在起作用约束上．

#### （一）IP型约束问题的一阶必要条件

图2.9所示为具有三个不等式约束的二维最优化问题．

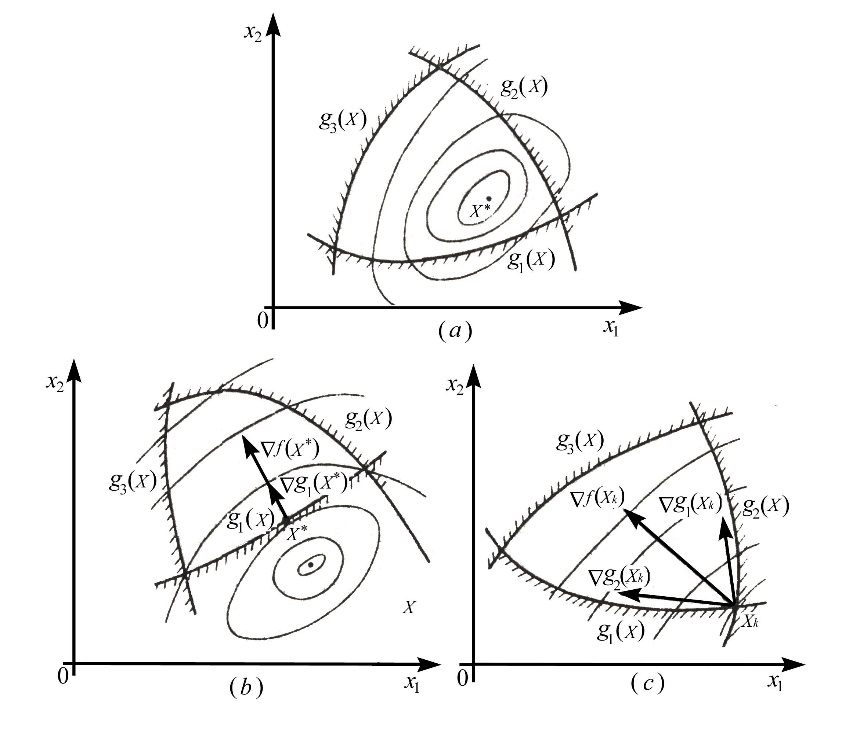


图2.9

图2.9（a）是最优点在可行域内部的一种情况．在此种情形下，点的全部约束函数值均大于零，所以这组约束条件对其最优点都不起作用．换句话说，如果除掉全部约束，其最优点也仍是同一个点．因此这种约束优化问题与无约束优化问题是等价的．图2.9（b）所示的约束最优点在的边界曲线与目标函数等值线的切点处．此时，，所以是起作用约束，而其余的两个是不起作用约束．

既然约束最优点是目标函数等值线与边界的切点，则在点处目标函数的梯度与约束函数梯度矢量必共线，而且方向一致．若取非负乘子，则在处存在如下关系

．

另一种情况如图2.9（c）所示．当前迭代点在两约束交点上，该点目标函数的梯度矢量夹于两约束函数的梯度矢量之间．显然，在点邻近的可行域内部不存在目标函数值比更小的可行点．因此，点就是约束最优点，记作．由图可知，此时点目标函数的梯度可表达为约束函数梯度和的线性组合．若用代替即有



成立，且式中的乘子和必为非负．

总结以上各种情况，最优解的一阶必要条件为



对于（2.16）IP型约束问题的一阶必要条件讨论如下：

设最优点位于个约束边界的汇交处，则这个约束条件组成一个起作用的约束集．按上面的分析，对于点必有下式成立

 （2.17）

但是在实际求解过程中，并不能预先知道最优点位于哪一个或哪几个约束边界的汇交处．为此，把个约束全部考虑进去，并取不起作用约束的相应乘子为零，则最优解的一阶必要条件应把式（2.17）修改为

 （2.18）

式（2.18）为IP型问题约束最优解的一阶必要条件，它与式（2.17）等价．因为在下，对于起作用约束，必有使式（2.18）中的第四式成立；对于不起作用约束，虽然而必有，可见式（2.18）与式（2.17）等价．

#### （二）EP型约束问题的一阶必要条件

图2.10所示为具有一个等式约束条件的二维化问题，其数学模型为



在该问题中，等式约束曲线是它的可行域，而且目标函数等值线与约束曲线的切点是该约束问题的最优解．

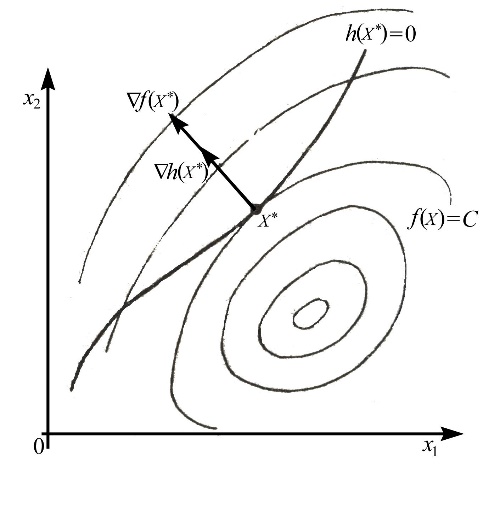


图2.10

在点处，目标函数的梯度与约束函数的梯度共线．因此，在最优点处一定存在一个乘子，使得



成立．

对于一般的维等式约束优化问题，其数学模型为



则为其解的一阶必要条件为



#### （三）GP型约束问题解的一阶必要条件

由上述不等式约束优化与等式约束优化问题的一阶必要条件，可以推出一般约束优化问题的条件．

设维一般约束优化问题的数学模型为

 （2.19）

则为其解的一阶必要条件应为

 （2.20）

函数



称为关于问题（2.19）的广义拉格朗日函数，式中

，



为拉格朗日乘子．由于引入拉格朗日函数，条件（2.20）中的第一式可写为

．

#### （四）Kuhn—Tｕcker条件（简称K—T条件）

在优化实用计算中，常常需要判断某可行迭代点是否可作为约束最优点输出而结束迭代，或者对此输出的可行结果进行检查，观察它是否已满足约束最优解的必要条件，这种判断或检验通常借助于条件进行的．对于IP型问题，条件可叙述如下：

如果是一个局部极小点 ，且各梯度矢量组成线性无关的矢量系，那么必存在一组非负乘子，使得



成立．

必须指出，在一般情形下，条件是判别约束极小点的一阶必要条件，但并非充分条件．只是对于凸规划问题，即对于目标函数为凸函数，可行域为凸集的最优化问题，条件才是约束最优化问题的充分条件．而且，在这种情况下的局部最优解也必为全局最优解．

应用条件检验某迭代点是否为约束最优点的具体作法可按下述步骤进行：

（1）检验是否为可行点．为此需要计算处的诸约束函数值，若是可行点，则．

（2）选出可行点处的起作用约束．前面已求得个值，其中等于零或相当接近零的约束就是起作用约束．把这些起作用约束重新编排成序列．

（3）计算点目标函数的梯度和*I*个起作用约束函数的梯度．

（4）按条件，点应满足

. (2.22）

将式（2.22）中的各梯度矢量用其分量表示，则可得到为变量的线性方程组



由于矢量系是线性无关的，所以该方程组存在唯一解．通过解此线性方程组，求得一组乘子，若所有乘子均为非负，即，则即为约束最优解．否则，点就不是约束最优点．

例2.9 设约束优化问题



它的当前迭代点为，试用条件判别它是否为约束最优点．

解：（1）计算点的诸约束函数值



是可行点．

（2）点起作用约束是

．

（3）求点梯度



（4）求拉格朗日乘子

按条件应有



写成线性方程组



解得．乘子均为非负，故满足约束最优解的一阶必要条件．

如图2.11所示，点确为该约束优化问题的局部最优解，由于可行域是凸集，所以点也是该问题的全局最优解．

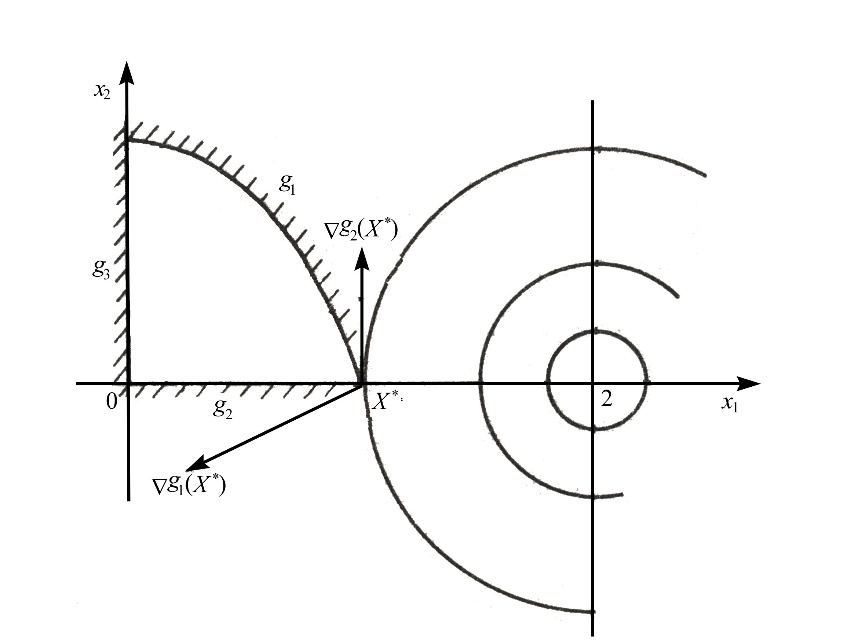


图2.11

GP型的约束最优化问题的条件类似于IP型约束最优化问题的条件：

如果是一个局部极小点 ，且各梯度矢量和组成线性无关的矢量系，那么必存在两组乘子和，使得

 （2.22）

成立．

# 第三章 线性规划及其对偶问题

线性规划是最优化问题的一种特殊情形，也是运筹学的一个重要分支，它的实质是从多个变量中选取一组适当的变量作为解，使这组变量满足一组确定的线性式，而且使一个线性目标函数达到最优（最大或最小）．

线性规划的应用极为广泛，自1949年美国数学家G. B. Dantzing提出一般线性规划问题求解的方法——单纯形法之后，线性规划无论在理论上、计算方法和开拓新的应用领域中，都获得了长足的进步，线性规划从解决技术问题的最优化设计到工业、农业、商业、交通运输业、军事、经济计划和管理决策等领域都有广泛的发展和应用．本章主要从线性规划的基本概念、数学模型、单纯形法、对偶理论、灵敏度分析等方面进行介绍．

## §3.1 线性规划数学模型基本原理

### 一、线性规划的数学模型

满足以下三个条件的数学模型称为线性规划的数学模型：

（1）每一个问题都用一组决策变量表示某一方案；每一组值就代表一个具体方案．

（2）有一个目标函数，可用决策变量的线性函数来表示，按问题的不同，要求目标函数实现最大化或最小化．

（3）有一组约束条件，可用一组线性等式或不等式来表示．

线性规划问题的一般形式为



这里，目标函数中的系数叫做目标函数系数或价值系数，约束条件中的常数叫做资源系数，约束条件中的系数

叫做约束系数或技术系数．

### 二、线性规划问题的标准形式

所谓线性规划问题的标准形式，是指目标函数要求，所有约束条件都是等式约束，且所有决策定量都是非负的，即



或简写为



可以规定各约束条件中的资源系数，否则等式两端乘以“”．线性规划问题的矩阵表示为



其中，，

，．

任意的线性规划模型都可以转化为标准形式：

（1）若目标函数是求最大值的问题，这时只需将所有目标函数系数乘以“－1”，求最大值的问题就变成了求最小值的问题，即

．

求其最优解后，把最优目标函数值反号即得原问题的目标函数值．

（2）若约束条件为不等式，这里有两种情况：一种是“≤”不等式，则可在“≤”不等式的左端加入一个非负的新变量（叫松驰变量），把不等式变为等式；另一种是“≥”不等式，则可在“≥”不等式的左端减去一个非负松驰变量（也叫剩余变量），把不等式变为等式．松驰变量在目标函数中对应的系数为零．

（3）若存在取值无约束的变量，可令，其中，．

例3.1 将下列线性规划问题化为标准形式



解 将目标函数变为，令，其中，在第一个约束不等式中加入松驰变量，在第二个约束不等式中减去剩余变量，则可得标准形式



### 三、线性规划的解的概念和基本定理

考虑线性规划标准形式的约束条件

，

其中为矩阵，，是维向量．假定增广矩阵的秩=矩阵的秩，把矩阵的列进行可能的重新排列，使．这里*B*为矩阵，且有逆矩阵存在，即，称*B*为该线性规划问题的一个基．不失一般性，设

，

称为基向量，与基向量对应的变量称为基变量，记为，其余的变量称为非基变量，记为．令个非基变量均为0，并用高斯消元法，可得一个解，称为该约束方程组的基解，其中．

满足非负约束条件（基解的非零分量都）的基解称为基可行解．对应于基可行解的基称为可行基．基可行解的非零分量个数小于时，称为退化解．

线性规划的解的基本定理：

引理3.1 线性规划问题的可行解为基可行解的充要条件是*X*的正分量所对应的系数列向量是线性无关的．

证 必要性由基可行解的定义可知．下证充分性

若向量组线性无关，则必有；当时，它们恰构成一个基，从而为相应的基可行解．当时，则一定可以从其余的列向量中取出个与构成最大的线性无关向量组，其对应的解恰为，所以它是基可行解．

定理3.1 线性规划问题的基可行解*X*对应于可行域*D*的顶点．

证 不失一般性，假设基可行解*X*的前个分量为正，故

． （3.1）

现在分两步来讨论，分别用反证法．

（1）若*X*不是基可行解，则它一定不是可行域*D*的顶点．

根据引理3.1，若*X*不是基可行解，则其正分量所对应的系数列向量线性相关，即存在一组不全为零的数，使得

 （3.2）

用一个的数乘式（3.2），再分别与式（3.1）相加和相减，得到

，．

现取

，

，

由可得，即是连线的中点．

另一方面，当充分小时，可保证

，

即是可行解，这证明了不是可行域的顶点．

（2）若不是可行域的顶点，则它一定不是基可行解．

因为不是可行域的顶点，故在可行域中可找到不同的两点，

，

，

使 ．

设是基可行解，对应向量组线性无关，当时，有，由于是可行域的两点，应满足

．

将这两式相减，即得

．

因，所以上式系数不全为零，故向量组线性相关，与假设矛盾，即*X*不是基可行解．

定理3.2 若可行域有界，线性规划问题的目标函数一定可以在其可行域的顶点上达到最优．

证 设是可行域的顶点，若不是顶点，且目标函数在处达到最优（标准形式是）．

因不是顶点，所以它可以用*D*的顶点线性表示为

．

因此

． （3.3）

在所有的顶点中必然能找到某一个顶点，使是所有中最小者，并且将代替式（3.3）中的所有，得到

，

由此得到

．

根据假设，是最小值，所以只能有

，

即目标函数在顶点处也达到最小值．

§3.2 线性规划迭代算法

单纯形法是求解线性规划问题的迭代算法．

一、单纯形法的计算步骤

单纯形法的基本思路是：从可行域中某个基可行解（一个顶点）开始，转换到另一个基可行解（顶点），直到目标函数达到最优时，基可行解即为最优解．单纯形法的基本过程如图3.1所示．

|  |  |
| --- | --- |
| |  | | --- | | 是否最优解或无有限最优解 | |

|  |  |
| --- | --- |
| |  | | --- | | 沿边界找新的基可行解解 | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 结束开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 初始基可行解 | |

图3.1

为计算方便，通常借助于单纯形表来计算，从初始单纯形表3.1开始，每迭代一步构造一个新单纯形表．

单纯型表中列中填入基变量；列中填入基变量的价值系数；列中填入约束方程组右端的常数；列的数字是在确定换入变量后，按规则计算填入；最后一行称为检验数行，对应各非基变量的检验数是，（这里令）．

表3.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | … |  |  | … |  |  |
|  | … |  |  | … |  |
| *c*1 | *x*1 | *b*1 | 1 | … | 0 | *a*1,*m*+1 | … | *a*1*n* |  |
| *c*2 | *x*2 | *b*2 | 0 | … | 0 | *a*2,*m*+1 | … | *a*2*n* |  |
| ┇ | ┇ | ┇ | ┇ | … | ┇ | ┇ | … | ┇ | ┇ |
| *cm* | *xm* | *b*m | 0 | … | 0 | *am,m*+1 | … | *amn* | *θm* |
|  | －*f*(*x*) |  | 0 | … | 0 |  | … |  |  |

单纯形法的计算步骤：

（1）找出初始可行基，确定初始基可行解，建立初始单纯形表．

（2）检验各非基变量的检验数（）．若所有，则已得到最优解，停止计算．否则转入下一步．

（3）在中，若所有，则此问题无最优解，停止计算．否则转入下一步．

（4）根据，确定为换入变量．按规则计算

，

可确定为换出变量，转入下一步．

（5）以为主元素进行迭代（用高斯消元法），把所对应的列向量

，

将列中的换为，得到新的单纯形表，重复步骤（2）—步骤（5），直到终止．

单纯形法的流程图如图3.2所示．

|  |  |
| --- | --- |
| |  | | --- | | 初始可行基 | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 以为主元素进行迭代 | |

|  |  |
| --- | --- |
| |  | | --- | | 得到最优解 | |

|  |  |
| --- | --- |
| |  | | --- | | 得到最优解 | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | NY | |

|  |  |
| --- | --- |
| |  | | --- | | NY | |

|  |  |
| --- | --- |
| |  | | --- | | 所有 | |

|  |  |
| --- | --- |
| |  | | --- | | 所有? | |

|  |  |
| --- | --- |
| |  | | --- | | 计算 | |

|  |  |
| --- | --- |
| |  | | --- | | 计算 | |

图3.2

若目标函数要求实现最大化，一方面可将最大化转换为最小化，另一方面也可在上述计算步骤中将判定最优解的改为，将换入变量的条件改为．

### 二、初始可行基的确定

1. 若线性规划问题是



则从中一般能直接观察到存在一个初始可行基

．

（2）对所有约束条件是“≤”形式的不等式，可以利用化标准形式的方法，在每个约束条件的左端加入一个松驰变量，经过整理重新对及进行编号，可得下列方程组



显然得到一个单位矩阵B可作为初始可行基

．

（3）对所有约束条件是“≥”形式的不等式及等式约束情况，若不存在单位矩阵时，可采用人工变量，即对不等式约束减去一个非负的剩余变量后，再加入一个非负的人工变量；对等式约束再加入一个非负的人工变量，总可得到一个单位矩阵作为初始可行基．

例3.2 求解线性规划问题



解：将线性规划问题化为标准形式



作初始单纯形表，按单纯形法计算步骤进行迭代，结果如下（表3.2）．

表3.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | －2 | －3 | 0 | 0 | 0 |  |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| 0 | *x*3 | 8 | 1 | 2 | 1 | 0 | 0 | 4 |
| 0 | *x*4 | 16 | 4 | 0 | 0 | 1 | 0 | － |
| 0 | *x*5 | 12 | 0 | [4] | 0 | 0 | 1 | 3 |
|  |  | 0 | －2 | －3 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *x*3 | 2 | [1] | 0 | 1 | 0 | －1/2 | 2 |
| 0 | *x*4 | 16 | 4 | 0 | 0 | 1 | 0 | 4 |
| －3 | *x*2 | 3 | 0 | 1 | 0 | 0 | 1/4 | － |
|  |  | 9 | －2 | 0 | 0 | 0 | 3/4 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| －2 | *x*1 | 2 | 1 | 0 | 1 | 0 | －1/2 | － |
| 0 | *x*4 | 8 | 0 | 0 | －4 | 1 | [2] | 4 |
| －3 | *x*2 | 3 | 0 | 1 | 0 | 0 | 1/4 | 12 |
|  |  | 13 | 0 | 0 | 2 | 0 | －1/4 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| －2 | *x*1 | 4 | 1 | 0 | 1 | 1/4 | 0 |  |
| 0 | *x*５ | 4 | 0 | 0 | －2 | 1/2 | 1 |  |
| －3 | *x*2 | 2 | 0 | 1 | 1/2 | －1/8 | 0 |  |
|  |  | 14 | 0 | 0 | 3/2 | 1/8 | 0 |  |

表3.2最后一行的检验数均为正，这表示目标函数值已不可能再减小，于是得到最优解

，

目标函数值．

### 三、单纯形法的有关说明

对线性规划问题

 （3.5）

若系数矩阵中不含单位矩阵，没有明显的基可行解时，常采用引入非负人工变量的方法来求初始基可行解．下面分别介绍常用的“大*M*法”和“两阶段法”．

#### （一）大*M*法

在约束条件式（3.5）中加入人工变量，人工变量在目标函数中的价值系数为*M*，*M*为一个很大的正数．在迭代过程中，将人工变量从基变量中逐个换出，如果在最终表中当所有检验数时，基变量中不再含有非零的人工变量，这表示原问题有解，否则无可行解．

例3.3 求解线性规划问题



解：将原问题化为标准形式并引入人工变量，得



用单纯形法计算，得表3.3．

根据表3.3的最后一行的检验数均，得最优解，最优值，由于人工变量的值均为零，故得原问题的最优解，最优值为．

表3.3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | －3 | 1 | 1 | 0 | 0 | *M* | *M* |  |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 | *x*6 | *x*7 |
| 0 | *x*4 | 11 | 1 | －2 | 1 | 1 | 0 | 0 | 0 | 11 |
| *M* | *x*6 | 3 | －4 | 1 | 2 | 0 | －1 | 1 | 0 | 3/2 |
| *M* | *x*7 | 1 | －2 | 0 | [1] | 0 | 0 | 0 | 1 | 1 |
|  |  | －4*M* | －3+6*M* | 1－*M* | 1－3*M* | 0 | M | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *x*4 | 10 | 3 | －2 | 0 | 1 | 0 | 0 | －1 | － |
| *M* | *x*6 | 1 | 0 | [1] | 0 | 0 | －1 | 1 | －2 | 1 |
| 1 | *x*3 | 1 | －2 | 0 | 1 | 0 | 0 | 0 | 1 | － |
|  |  | －*M*－1 | －1 | 1－*M* | 0 | 0 | M | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *x*4 | 12 | [3] | 0 | 0 | 1 | －2 | 2 | －5 | 4 |
| 1 | *x*2 | 1 | 0 | 1 | 0 | 0 | －1 | 1 | －2 | － |
| 1 | *x*3 | 1 | －2 | 0 | 1 | 0 | 0 | 0 | 1 | － |
|  |  | －2 | －1 | 0 | 0 | 0 | *1* | *M*－1 | *M*+1 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| －3 | *x*1 | 4 | 1 | 0 | 0 | 1/3 | －2/3 | 2/3 | －5/3 |  |
| 1 | *x*2 | 1 | 0 | 1 | 0 | 0 | －1 | 1 | －2/3 |  |
| 1 | *x*3 | 9 | 0 | 0 | 1 | 2/3 | －4/3 | 4/3 | －7/3 |  |
|  |  | 2 | 0 | 0 | 0 | 1/3 | *1/3* | *M*－1/3 | *M*－2/3 |  |

#### （二）两阶段法

两阶段法是把线性规划问题的求解过程分为两个阶段：

第一阶段，给原问题加入人工变量，构造仅含价值系数为1的人工变量的目标函数且要求实现最小化，其约束条件与原问题相同，即



然后用单纯形法求解上述问题，若得到，这说明原问题存在基可行解，可进入第二阶段计算，否则原问题无可行解，停止计算．

第二阶段，将第一阶段计算得到的最终表，除去人工变量，将目标函数行的系数换为原问题的目标函数系数，作为第二阶段计算的初始单纯形表进行计算．

例3.4 用两阶段法求解线性规划问题



解 第一阶段，标准化并引入人工变量，得如下的线性规划

，



用单纯形法计算该线性规划（见表3.4），最优解为，最优值．

表3.4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 | *x*6 | *x*7 |
| 0 | *x*4 | 11 | 1 | －2 | 1 | 1 | 0 | 0 | 0 | 11 |
| 1 | *x*6 | 3 | －4 | 1 | 2 | 0 | －1 | 1 | 0 | 3/2 |
| 1 | *x*7 | 1 | －2 | 0 | [1] | 0 | 0 | 0 | 1 | 1 |
|  |  | －4 | 6 | －1 | －3 | 0 | *1* | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *x*4 | 10 | 3 | －2 | 0 | 1 | 0 | 0 | －1 | － |
| 1 | *x*6 | 1 | 0 | [1] | 0 | 0 | －1 | 1 | －2 | 1 |
| 0 | *x*3 | 1 | －2 | 0 | 1 | 0 | 0 | 0 | 1 | － |
|  |  | －1 | 0 | －1 | 0 | 0 | *1* | 0 | 3 |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *x*4 | 12 | [3] | 0 | 0 | 1 | －2 | 2 | －5 | 4 |
| 0 | *x*2 | 1 | 0 | 1 | 0 | 0 | －1 | 1 | －2 | － |
| 0 | *x*3 | 1 | －2 | 0 | 1 | 0 | 0 | 0 | 1 | － |
|  |  | 0 | 0 | 0 | 0 | 0 | *0* | 1 | 1 |  |

由于人工变量，所以得原问题的基可行解为．于是进入第二阶段计算（见表3.5），最优解为，最优值，于是原问题的最优解为，最优值为．

表3.5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | －3 | 1 | 1 | 0 | 0 |  |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| 0 | *x*4 | 12 | [3] | 0 | 0 | 1 | －2 | 4 |
| 1 | *x*2 | 1 | 0 | 1 | 0 | 0 | －1 | － |
| 1 | *x*3 | 1 | －2 | 0 | 1 | 0 | 0 | － |
|  |  | －2 | －1 | 0 | 0 | 0 | 1 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| －3 | *x*1 | 4 | 1 | 0 | 0 | 1/3 | －2/3 |  |
| 1 | *x*2 | 1 | 0 | 1 | 0 | 0 | －1 |  |
| 1 | *x*3 | 9 | 0 | 0 | 1 | 2/3 | －4/3 |  |
|  |  | 2 | 0 | 0 | 0 | 1/3 | 1/3 |  |

## §3.3 对偶问题的基本原理

### 一、对偶问题的提出

对偶性是线性规划的重要内容之一，每一个线性规划问题必然有与之相伴而生的另一个线性规划问题，我们称一个叫原问题，另一个叫对偶问题，这两个问题有着非常密切的关系，让我们先分析一个实际的线性规划模型与其对偶线性规划问题的经济意义．

例3.5 某工厂计划在下一生产周期生产3种产品，，，这些产品都要在甲、乙、丙、丁4种设备上加工，根据设备性能和以往的生产情况知道单位产品的加工工时，各种设备的最大加工工时限制，以及每种产品的单位利润（单位：千元），如表3.6所示，问如何安排生产计划，才能使工厂得到最大利润？

表3.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 产 品  设 备 |  |  |  | 总工时限制/h |
| 甲/h | 2 | 1 | 3 | 70 |
| 乙/h | 4 | 2 | 2 | 80 |
| 丙/h | 3 | 0 | 1 | 15 |
| 丁/h | 2 | 2 | 0 | 50 |
| 单位利润/千元 | 8 | 10 | 2 |  |

解 设分别为产品的产量，构造此问题的线性规划模型为



现在从另一个角度来讨论该问题．

假设工厂考虑不安排生产，而准备将所有设备出租，收取租费．于是，需要为每种设备的台时进行估价．

设分别表示甲、乙、丙、丁4种设备的台时估价．由表3.6可知，生产一件产品需用各设备台时分别为，如果将不用于生产产品，而是用于出租，那么将得到租费

．

当然，工厂为了不至于蚀本，在为设备定价时，保证用于生产产品的各设备台时得到的租费，不能低于产品的单位利润8千元，即

．

按照同样分析，用于生产一件产品的各设备台时，，，所得的租费，不能低于产品的单位利润10千元，即

．

同理，还有

．

另外，价格显然不能为负值，所以．

企业现在设备的总以时数为70*h*，80*h*，15*h*，50*h*，如果将这些台时都用于出租，企业的总收入为

．

企业为了能够得到租用设备的用户，使出租设备的计划成交，在价格满足上述约束的条件下，应将设备价值定得尽可能低，因此取的最小值，综合上述分析，可得到一个与例3.5相对应的线性规划，即



称后一个规划问题为前一个规划问题的对偶问题，反之，也称前一个规划问题是后一个规划问题的对偶问题．

### 二、原问题与对偶问题的表达形式和关系

在线性规划的对偶理论中，把如下线性规划形式称为原问题的标准形式



而把如下线性规划形式称为对偶问题的标准形式



若用矩阵形式表示，则原问题和对偶问题分别可写成如下形式：

原问题

 （3.6）

对偶问题

 （3.7）

原问题与对偶问题的关系见表3.7．

表3.7

|  |  |
| --- | --- |
| 原问题（对偶问题） | 对偶问题（原问题） |
| min | max |
| 目标函数系数  右边常数  约束条件系数矩阵 | 右边常数  目标函数系数  系数矩阵的转置 |
| 第个约束条件为“≥”型  第个约束条件为“=”型  第个约束条件为“≤”型 | 第个变量≥0  第个变量无限制  第个变量≤0 |
| 第个变量≥0  第个变量无限制  第个变量≤0 | 第个约束条件为“≤”型  第个约束条件为“=”型  第个约束条件为“≥”型 |

例3.6 求下面线性规划问题的对偶问题



解：根据表3.7可直接写出上述问题的对偶问题



### 三、对偶理论

定理3.3（弱对偶定理） 对偶问题（max）的任何可行解，其目标函数值总是不大于原问题（min）任何可行解的目标函数值．

证 由定理所设及问题（3.6）和问题（3.7）容易看出

．

定理3.4（对偶定理） 假如原问题或对偶问题之一具有有限的最优解，则另一问题也具有有限的最优解，且两者相应的目标函数值相等．假如一个问题的目标函数值是无界的，则另一问题没有可行解．

证明从略．

定理3.5（互补松驰定理） 假如和分别是原问题（3.6）和对偶问题（3.7）的可行解，是原问题剩余变量的值，是对偶问题松驰变量的值，则、分别是原问题和对偶问题最优解的充要条件是

．

证 由定理所设，可知有

 （3.8）

 （3.9）

分别以左乘式（3.8），以右乘式（3.9），两式相减，得

．

若，根据弱对偶定理知

．

这说明，分别是原问题和对偶问题最优解，反之亦然．

根据互补松驰定理和决策变量满足非负条件可知，在最优解时，和同时等于0，所以有， ．

于是，互补松驰定理也可以这样叙述：最优化时，假如一个问题的某个变量取正数，则相应的另一个问题的约束条件必取等式；或者一个问题中的约束条件不取等式，则相应于另一问题中的变量必为零．

例3.7 已知线性规划问题



已知其对偶问题的最优解为，试用对偶理论找出原问题的最优解．

解：先写出它的对偶问题



将的值代入约束条件，得（2），（3），（4）为严格不等式，由互补松驰定理得，因，原问题的两个约束条件应取等式，故有

，

．

求解后得到，故原问题的最优解为

．

### 四、对偶问题的迭代算法

对偶单纯形法是对偶问题的迭代算法，其基本思想是：从原问题的一个基本解出发，此基本解不一定是可行解，但它对应着对偶问题的一个可行解；然后检验原问题的基本解是否可行，即是否有负的分量．如果有小于零的分量，则进行迭代，求另一个基本解，此基本解对应着另一个对偶可行解．如果得到的基本解的分量皆非负，则该基本解为最优解．也就是说，对偶单纯形法在迭代过程中始终保持对偶解的可行解，使原问题的基本解由不可行逐步变为可行．当同时得到对偶问题与原问题的可行解时，便得到原问题的最优解．

对线性规划问题的标准形式



对偶单纯形法的计算步骤如下：

(1)找出原问题的一个基，构成初始对偶基可行解，使所有检验数，构成初始对偶单纯形表．

(2)若所有，则当前的解是最优解，停止计算，否则计算，则行为主行，该行对应的基变量为换出变量．

(3)若所有，则对偶问题无界，原问题无解，停止计算，否则计算

，

则列为主列，该列对应的基变量为换入变量．

(4)以为主元素进行迭代，然后转回步骤(2)．

对偶单纯形法的流程图如图3.3所示．

例3.8 用对偶单纯形法求解下述线性规划问题



解：首先将“≥”约束条件两边反号，再加入松驰变量，可得原问题的一个基



建立初始单纯形表，见表3.8．

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | 2 | 3 | 4 | 0 | 0 |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| 0 | *x*4 | －3 | －1 | －2 | －1 | 1 | 0 |
| 0 | *x*5 | －4 | [－2] | 1 | －3 | 0 | 1 |
|  |  |  | 2 | 3 | 4 | 0 | 0 |

从表3.8看出，所有检验数，则对应对偶问题的解是可行的，因列数字为负，需进行迭代，计算

．

所以为换出变量．又因为

，

|  |  |
| --- | --- |
| |  | | --- | | 原问题的基本解的检验数 | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 以为主元素进行迭代 | |

|  |  |
| --- | --- |
| |  | | --- | | 得到最优解 | |

|  |  |
| --- | --- |
| |  | | --- | | 得到最优解 | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | NY | |

|  |  |
| --- | --- |
| |  | | --- | | NY | |

|  |  |
| --- | --- |
| |  | | --- | | 所有 | |

|  |  |
| --- | --- |
| |  | | --- | | 所有? | |

|  |  |
| --- | --- |
| |  | | --- | | 计算 | |

|  |  |
| --- | --- |
| |  | | --- | | 计算 | |

图3.3

所以为换入变量，以换入、换出变量所在行列交叉处元素“－2”为主元素，按单纯形法计算步骤进行迭代，得表3.9．

表3.9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | 2 | 3 | 4 | 0 | 0 |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| 0 | *x*4 | －1 | 0 | [－5/2] | 1/2 | 1 | －1/2 |
| 2 | *x*1 | 2 | 1 | －1/2 | 3/2 | 0 | －1/2 |
|  |  |  | 0 | 4 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | *x*2 | 2/5 | 0 | 1 | －1/5 | －2/5 | 1/5 |
| 2 | *x*1 | 11/5 | 1 | 0 | 7/5 | －1/5 | －2/5 |
|  |  |  | 0 | 0 | 3/5 | 8/5 | 1/5 |

由表3.9的最后一行看出，所有检验数，故原问题的最优解为

．

若对应两个约束条件对偶变量为，，则可得对偶问题的最优解为

．

## §3.4 线性规划问题灵敏度

在建立实际的线性规划模型时，所收集到的数据不是很精确；另一方面在实际应用中，各种信息瞬息万变，已形成的数学模型中的某些数据需要随之而变．因此，对于一个线性规划问题，研究当数据发生变动时解的变化情况是很重要的．下面仅介绍两种数据变化而导致解的变化的情况，这就是灵敏度分析问题．

### 一、价值系数的变化

假设只有一个系数变化，其它系数保持不变 ，的变化只影响检验解而不影响解的非负定性，下面分别就是非基变量系数和基变量系数两种情况进行讨论．

（1）是非基变量的系数

由于不变，因而对任何都不变．这时非基变量的系数的变化只影响与有关的一个检验数的变化，而对其它没有影响，设系数从变化到，这时检验数被所代替，在当前解是原问题的最优解时，有，假如，则必须引进基，单纯形法继续进行，否则原解仍是变化后的新问题的最优解，最优解不变相当于变化的界限为

．

2）为基变量的系数

当被所代替时，变成，可计算为

． （3.10）

特别是当时，，且，因此，仍为零．由式（3.10）知，基变量的价值系数的变化会引起整个价值系数行的变化，变化值为乘以最终表相应该基变量所在的行的数值．列本身则调整为．

由式（3.10）可看出，当对某个非基变量，式（3.10）为负时会引起基的变化，若要保持最优解不变，分析变化值且大于或小于零以及值是正或负的情况，得出会保持最优解不变的的变化界限为

．

例3.8 以例3.2的最终表为例，设基变量的系数变化，在原最优解不变条件下，确定的变化范围．

解 此时例3.2的最终表便成为表3.10

表3.10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | －2 | －3+ | 0 | 0 | 0 |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| －2 | *x*1 | 4 | 1 | 0 | 0 | 1/4 | 0 |
| 0 | *x*5 | 4 | 0 | 0 | －2 | 1/2 | 1 |
| －3 | *x*2 | 2 | 0 | 1 | 1/2 | －1/8 | 0 |
|  |  |  | 0 |  | 3/2 | 1/8 | 0 |

为了保持原最优解不变，则的检验数应当为零，进行行初等变换，得表3.11．

表3.11

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *CB* | *XB* | *b* | －2 | －3+ | 0 | 0 | 0 |
| *x*1 | *x*2 | *x*3 | *x*4 | *x*5 |
| －2 | *x*1 | 4 | 1 | 0 | 0 | 1/4 | 0 |
| 0 | *x*5 | 4 | 0 | 0 | －2 | 1/2 | 1 |
| －3+ | *x*2 | 2 | 0 | 1 | 1/2 | －1/8 | 0 |
|  |  |  | 0 | 0 | 3/2－ | 1/8+ | 0 |

从表（3.11）可得

且．

由此可得的变化范围为

，

即的价值系数可以在[0，4]之间变化，而不影响原最优解．

### 二、资源系数的变化

假设资源系数变化为，的变化将会影响解的可行性，但不会引起检验数的符号变化．根据基可行解的矩阵表示可知，，所以只要变化必定会导致最优解的数值发生变化，最优解的变化分为两类：一类是保持，最优基*B*不变；另一类是中出现负分量，这将使最优基*B*变化，若最优基不变，则只需将变化后的代入的表达式重新计算即可；若中出现负分量，则要通过迭代求解新的最优基和最优解．

设系数变化到，而其它系数都不变，这样使最终表中原问题的解相应变化为

，

其中为原最优解，为的第个分量，为的第行第列元素，为了保持最优基不变，应使，即

．

由此可得到保持最优基不变时，资源系数的变化界限为

．

例3.9 若例3.2的第二个约束条件中变化为，在最优解不变的条件下，求的变化范围．

解 计算



可得



所以的变化范围是（－8，16）．显然的变化范围是（8，32）．

**第四章 一维搜索法**

由第一章关于求解最优化问题概述中我们知道，从已知迭代点出发按照基本迭代公式来求解最优化问题，其关键在于如何构造一个搜索方向和确定一个步长，使下一迭代点处的目标函数值下降，即．现在我们来讨论，当搜索方向已经确定的情况下，如何来确定步长？步长因子的选取有多种方法，如取步长为常数，但这样选取的步长并不最好，如何选取最好步长呢？实际计算通常采用一维搜索来确定最优步长．

对无约束最优化问题

，

当已知迭代点和下降方向时，要确定适当的步长使

比有所下降，即相当于对于参变量的函数



要在区间上选取使，即

．

由于这种从已知点出发，沿某一下降的探索方向来确定步长的问题，实质上是单变量函数关于变量的一维搜索选取问题，故通常叫做一维搜索．按这种方法确定的步长又称为最优步长，这种方法的优点是，它使目标函数值在搜索方向上下降得最多．

§4.1 搜索区间及其确定方法

一、搜索区间

设一维最优化问题为

．　 （4.3）

为了求解问题（4.3），我们引入如下的搜索区间概念．

定义4.1 设，并且

，

若存在闭区间使，则称是问题（4.3）的搜索区间．

简言之，一个一维最优化问题的搜索区间，就是包含该问题最优解的一个闭区间．通常，在进行一维搜索时，一般要先确定出问题的一个搜索区间，然后在此区间中进行搜索求解．

二、加步探索法

下面，介绍一个确定问题(4.3)的搜索区间的简单方法．这个方法的思想是：先选定一个初始点和初始步长．然后，沿着轴的正方向探索前进一个步长，得到新点．若目标函数在新点处的值是下降了，即

，

则下一步就从新点出发加大步长，再向前探索．若目标函数在新点处的 函数值上升，即

，

则下一步仍以为出发点以原步长开始向轴的负方向同样探索．当达到目标函数上升的点时，就停止探索，这时便得到问题（4.3）的一个搜索区间．这种以加大步长进行探索来寻找探索区间的方法叫做加步探索法．

加步探索法算法的计算步骤：

(1) 选取初始数据．选取初始点，计算．给出初始步长，加步系数，令．

(2) 比较目标函数值．令，计算，若，转(3)．否则转(4)．

(3)加大探索步长．令，同时，令，，，，转(2)．

（4）反向探索．若，转换探索方向，令，转（2）．否则，停止迭代，令



输出．

加步探索法算法的流程图如图4.2所示。

在加步探索法中，一般建议．若能估计问题（4.3）的最优解的大体位置的话，初始点要尽量取接近于问题（4.3）的最优解．在具体运用上述加步探索法时，有时还要考虑一些细节问题．例如，当探索得到新点处的目标函数值和出发点处相同时，以及初始步长应如何选取等，都需作适当处理．

三、单谷区间与单谷函数

由于以后要介绍的一些维搜索方法，主要适用于问题（4.3）在搜索区间中只有唯一的最优解的情况，为此，我们再给出下面单谷区间与单谷函数概念．

定义4.2 设，闭区间．若存在点，使得在上严格递减，在上严格递增，则称是函数的单谷区间，是上单谷函数．

由定义4.2易知，一个区间是某函数的单谷区间意味着，在该区间中函数只有一个“凹谷”（极小值）．例如，图4.3中的是的单谷区间，也即是上的单谷函数．图4.4中的不是的单谷区间，即不是上的单谷函数．

另外，从定义4.2还可知，某区间上的单谷函数在该区间上不一定是连续函数，而凸函数在所给区间上必然是单谷函数（如图4.3所示）．由定义4.1和定义4.2知，函数的单谷区间总是相应问题（4.3）的一个搜索区间（如图4.3所示），但反之不然（如图4.4所示）．

间总是相应问题（4.3）的一个搜索区间（如图4.3所示），但反之不然（如图4.4所示）．

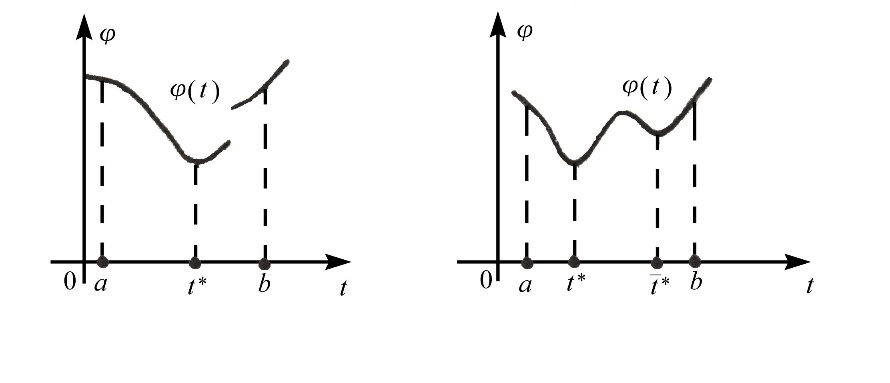


图4.3 图4.4

单谷区间和单谷函数有如下有用的性质：

定理4.1 设是的单谷区间，任取并且．

（1）若有，则是的单谷区间．

（2）若有，则是的单谷区间．

证明略．

定理4.1说明，经过函数值的比较可以把单谷区间缩短为一个较小的单谷区间．换句话说，利用这个定理可以把搜索区间无限缩小，从而求出极小点．以下介绍的几种，一维搜索方法都是利用这个定理通过不断地缩短搜索区间的长度，来求得一维最优化问题的近似最优解．

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 选取初始点*t*0，初始步长*h*0*>*0，加步系数*α>*1*，*令*k=*0 | |

|  |  |
| --- | --- |
| |  | | --- | | *φ*0*=φ*(*t*0)，比较目标函数值*tk+*1*=tk+hk*, *φk+*1*=φ*(*tk+*1*)* | |

|  |  |
| --- | --- |
| |  | | --- | | *a=*min{*t*，*tk+*1}  *b*=max{*t*，*tk+*1} | |

|  |  |
| --- | --- |
| |  | | --- | | 结束 | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | *φk+*1*<φk* | |

|  |  |
| --- | --- |
| |  | | --- | | *hk+*1*=hk,t=tk ,tk=tk+*1 *,φk=φk+*1*,k=k+*1 | |

|  |  |
| --- | --- |
| |  | | --- | | *k=*0 | |

|  |  |
| --- | --- |
| |  | | --- | | *hk =－ hk ,k=k+*1 | |

图4.2

§4.2 对分法

一、对分法基本原理

求解一维最优化问题一般可先确定它的一个有限搜索区间，把问题化为求解问题，然后通过不断缩短区间的长度，最后求得最优解．

设在已获得的搜索区间内具有连续的一阶导数．因为在上可微，故在上连续，由此知在上有最小值．

令，总可求得极小点．不妨设在上是单减函数；在上是单增函数．所以时，，故；当时，，亦即．对分法的原理如图4.5所示．

|  |
| --- |
| 图4.5 |

二、对分法迭代步骤

已知，表达式，终止限．

(1) 确定初始搜索区间，要求．

(2) 计算的中点．

(3) 若，则，转(4)；若，则，转(5)；若，则，转(4)．

(4) 若，则，转(5)；否则转(2)．

(5) 打印，结束．

对分法的计算流程如图4.6所示．

二、对分法迭代步骤

已知，表达式，终止限．

(1) 确定初始搜索区间，要求．

(2) 计算的中点．

(3) 若，则，转(4)；若，则，转(5)；若，则，转(4)．

(4) 若，则，转(5)；否则转(2)．

(5) 打印，结束．

对分法的计算流程如图4.6所示．

三、对分法有关说明

对分法每次迭代都取区间的中点．若这点的导数值小于零，说明的根位于右半区间中（如图4.5所示），因此去掉左半区间；若中点导数值大于零，则去掉右半区间；若中点导数值正好等于零，则该点就是极小点．因为每次迭代都使原区间缩短一半，所以称为对分法或二分法．

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 确定[*a*，*b*]，要求 | |

|  |  |
| --- | --- |
| |  | | --- | | *c=(a+b)/2* | |

|  |  |
| --- | --- |
| |  | | --- | | *b=c* | |

|  |  |
| --- | --- |
| |  | | --- | | *t\*=*(*a+b*)*/2* | |

|  |  |
| --- | --- |
| |  | | --- | | 输出*t\** | |

|  |  |
| --- | --- |
| |  | | --- | | 结束 | |

|  |  |
| --- | --- |
| |  | | --- | | *t\*=c* | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | | *a=c* | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

图4.6

§4.3 Newton切线法

一、Newton切线法基本原理

设在已获得的搜索区间内具有连续二阶导数，求

．

因为在上可微，故在上有最小值，令．

下面不妨设在区间中经过次迭代已求得方程的一个近似根．过作曲线的切线，其方程是

． （4.4）

然后用这条切线与横轴交点的横坐标作为根的新的近似（如图4.7所示）．它可由方程（4.4）在令的解出来，即

．

这就是Newton切线法迭代公式

二、Newton切线法迭代步骤

已知，表达式，终止限．

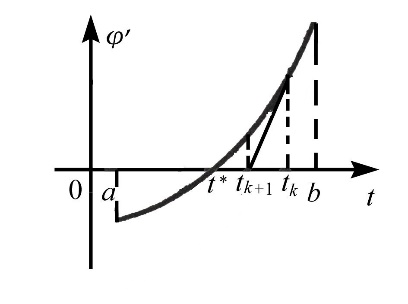


图4.7

(1) 确定初始搜索区间，要求．

(2) 选定．

(3) 计算．

(4) 若，则，转（3）；否则转（5）．

(5) 打印，结束．

Newton切线法的计算流程如图4.8所示．

三、Newton切线法有关说明

这种方法如果初始点选得适当，收敛速度很快，通常经过几次迭代就可以得到满足一般精度要求的结果，但是它也有缺点．第一，需要求二阶导数．如果在多维最优化问题的一维搜索中使用这种方法，就要涉及Hesse矩阵，一般是难于求出的．第二，当曲线在上有较复杂的弯曲时，这种方法也往往失效．如图4.9所示的迭代：，结果跳出．迭代或

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | 输出 | |

|  |  |
| --- | --- |
| |  | | --- | | 开始 | |

|  |  |
| --- | --- |
| |  | | --- | | 结束 | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | Y | |

|  |  |
| --- | --- |
| |  | | --- | | N | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | | 选定t0,确定[a b],要求 | |

者发散，或者找到的根并不是我们想要的结果．第三，即使曲线比较正常，在中或者上凹或者下凹，初始点的选取也必须适当．在图4.10（*a*）的情况下，曲线上凹，应选点b作为初始点；而在图4.10（*b*）的情况下，曲线下凹，应选点a为初始点．否则都可能失败．

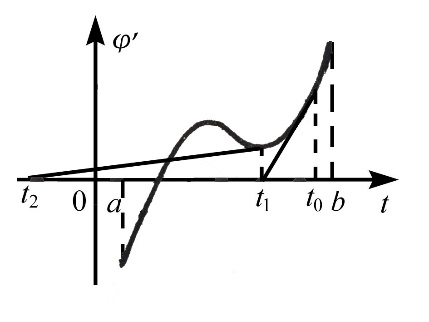
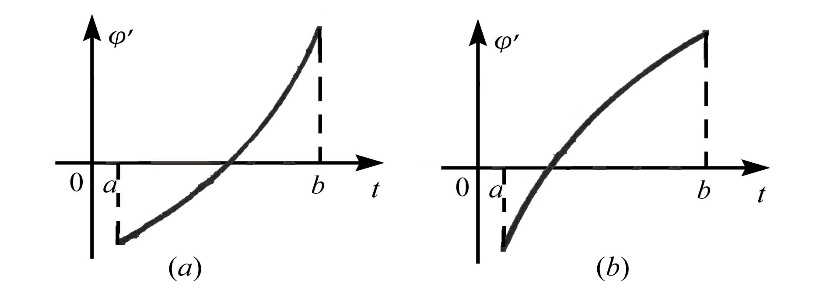


图4.9

图4.8



§4.4 黄金分割法

一、黄金分割法基本原理

要介绍黄金分割法有必要回顾一下古老的黄金分割问题．所谓黄金分割就是将一线段分为二段时，要求整段长*L*与较长段*x*的比值正好等于较长段*x*与较短段的比值（如图4.11所示），即

．

于是有，解出其正根

．

由此可见长段的长度应为全长的0.618倍，而短段的长度应为全长的0.382倍．因为古代的人们认为按0.618的比率来分割线段是最协调，胜似黄金，故称之为黄金分割．

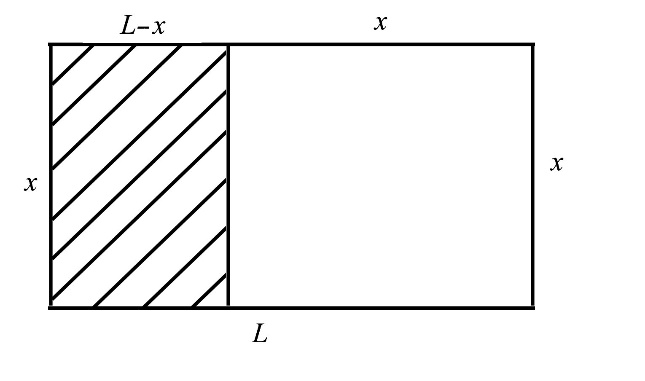


图4.11

用黄金分割法进行一维搜索，其基本思想是在单谷区间内适当插入两点，由此把区间分为三段，然后再通过比较这两点函数值大小，就可以确定是删去最左段还是最右段，或者同时删去左右两段保留中间段．如此继续下去可将单谷区间无限缩小．

二、黄金分割法迭代步骤

现在提出一个问题，就在上如何选取二点使得迭代次数最小而区间缩短最快？要解决这个问题，人们想到对区间选二点等价于将区间长度进行黄金分割，也就是将第一个搜索点取在的0.618处，第二个搜索点取成的对称点即的0.382处（如图4.12所示），亦即

，

．

计算与的值，并根据与的值的大小关系分情况讨论：

（1） 若，说明是好点，于是把区间划掉，保留，则内有一保留点，置新的区间；

（2）若，说明是好点，于是应将划掉，保留，则内有保留点，置新的区间．(2)计算．

(3)计算．

(4) 若，则打印，结束；否则转(5)

|  |
| --- |
| 图4.12 |

(5) 判别是否满足：若满足，则置

，

然后转(3)；否则，置

，

然后转(4)．

§4.5 抛物线插值法

一、抛物线插值法基本原理

考虑一维搜索问题

，

假设其中是定义在区间上的单谷函数．首先用试探法在上找一点，使之满足

．

通过目标函数曲线上的三个点，作它的二次拟合曲线（如图4.14所示）

．

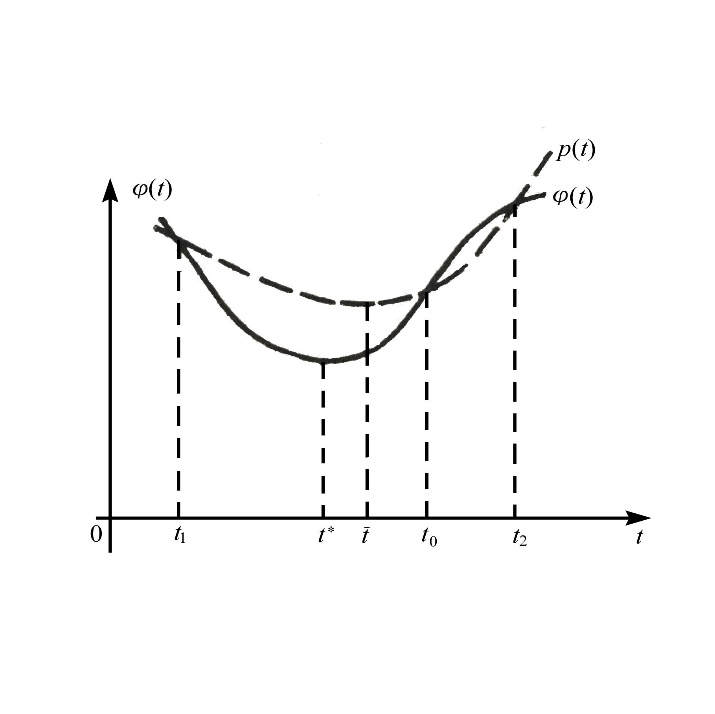


图4.14

由于上述三个点既是目标函数曲线上的点，又是二次拟合曲线上的点，故有方程组

 （4.5）

将方程组（4.5）中的消去，得

 （4.6）

从方程组（4.6）可解出待定系数

， （4.7）

． （4.8）

对于二次拟合函数，我们很容易求得它的极小值点．令，即，从中解出

 （4.9）

即为二次拟合函数的极小值点．

将式（4.7）与式（4.8）代入式（4.9）得

 （4.10）

用区间上二次拟合函数的这个极小值点作为目标函数在该区间极小值点的一个估计值．若和已充分接近，即对给定的允许误差使

 （4.11）

成立时，就可被看作是在区间内近似最优解；否则应缩短区间，按照值保持两头大、中间小的原则构成新的三点，继续上述过程，直至不等式（4.11）成立为止．

二、抛物线插值法迭代步骤

下面具体介绍一下缩短区间，构成新三点的方法．

由式（4.15）得到的点，在区间内既可能在点的左侧（即），又可能在的右侧（即），分别对应这两种情形比较和的大小，又有，及等三种情形，故共有如下六种情况（如图4.15与图4.16所示）：

（1）对于图4.15（*a*）的情况：因，所以相对来说是好点，故划掉区间，保留为新区间，故置，，保持不变；

（2）对于图4.15（*b*）的情况：因，所以相对来说是好点，故划掉，保留为新区间，故置，与保持不变；

（3）对于图4.15（*c*）的情况：因，所以相对来说是好点，

|  |  |
| --- | --- |
| |  | | --- | | 图4.15 图4.16 | |

故划掉，保留为新区间，故置，，保持不变；

（4）对于图4.15（*d*）的情况：因，所以相对来说是好点，故划掉保留，故置，与保持不变．

（5）对于图4.16（*a*）的情况：一般同时划掉及仅留中间的，故置，，，，， ；

（6）对于图4.16（*b*）的情况：一般同时划掉及，仅留中间，故置，，，，  ，．

通过上述讨论，我们可直接给抛物线插值法的迭代流程图（图4.17）．

三、抛物线插值法有关说明

抛物线插值法是多项式逼近法的一种．所谓多项式逼近，是利用目标函数在若干点的函数值或导数值等信息，构成一个与目标函数相接近的低次插值多项式，用该多项式的最优解作为目标函数的近似最优解．

第五章 常用无约束最优化方法

本章开始讨论多维无约束最优化问题

 （5.1）

其中．这个问题的求解是指，在中找一点，使得对于任意的都有

 （5.2）

成立．点就是问题（5.1）的全局最优点．但是，大多数最优化方法只能求到局部最优点，即在中找到一点，使得式（5.2）在的某个领域中成立．这个矛盾对于实际问题来讲一般容易解决，因为根据问题的实际意义多半可以直接判定用优化方法求出的局部最优解是否为全局最优解．但在理论上这是个比较复杂的问题，本书不涉及．

无约束优化方法是优化技术中极为重要和基本的内容之一．它不仅可以直接用来求解无约束优化问题，而且很多约束优化问题也常将其转化为无约束优化问题，然后用无约束优化方法来求解．另外，有些无约束优化方法只需略加处理，即可用于求解约束优化问题．

无约束优化理论发展较早，比较成熟，方法也很多，新的方法还在陆续出现．把这些方法归纳起来可以分成两大类：一类是仅用计算函数值所得到的信息来确定搜索方向，通常称它为直接搜索法，简称为直接法，另一类需要计算函数的一阶或二阶导数值所得到的信息来确定搜索方向，这一类方法称为间接法（或解析法）．直接法不涉及导数和Hesse矩阵，适应性强，但收敛速度一般较慢；间接法收敛速度一般较快，但需计算梯度，甚至需要计算Hesse矩阵．一般的经验是，在可能求得目标函数导数的情况下还是尽可能使用间接方法；相反，在不可能求得目标函数的导数或根本不存在导数的情况下，当然就应该使用直接法．

# 第五章 常用无约束最优化方法

§5.1 最速下降法

对于问题（5.1）为了求其最优解，按最优化算法的基本思想是从一个给定的初始点出发，通过基本迭代公式，按照特定的算法产生一串点列，如果点列收敛，则该点列的极限点为问题（5.1）的最优解．

一、最速下降法基本原理

在基本迭代公式中，每次迭代搜索方向取为目标函数的负梯度方向，即，而每次迭代的步长取为最优步长，由此所确定的算法称为最速下降法．

为了求解问题（5.1），如图5.1所示，假定我们已经迭代了次，获得了第个迭代点．现在从出发，可选择的下降方向很多，一个非常自然的想

|  |
| --- |
| 图5.1 |

法是沿最速下降方向（即负梯度方向）进行搜索应该是有利的，至少在邻近的范围内是这样．因此，取搜索方向为

．

为了使目标函数在搜索方向上获得最多的下降，沿进行一维搜索，由此得到第个迭代点，即

，

其中步长因子按下式确定

，

也可记为

． （5.3）

显然，令就可以得到一个点列，其中是初始点，由计算者任意选定．当满足一定的条件时，由式（5.3）所产生的点列必收敛于的极小点．

以后为书写方便，记．因此，．在不发生混淆时，再记．

### 二、最速下降法迭代步骤

已知目标函数及其梯度，终止限、和．

（1）选定初始点，计算，；置．

（2）作直线搜索：；计算

，．

（3）用终止准则检测是否满足：若满足，则打印最优解，，结束；否则，置，转（2）．

将最速下降法应用于正定二次函数

 （5.4）

可以推出显式迭代公式．

设第次迭代点为，我们来求的表达式．对式（5.4）关于求梯度，有

，（5.5）

因此，

． （5.6）

现在从出发沿作直线搜索以确定，于是

， 5.7）

其中是最优步长因子．

又因式（4.2），有，再利用式（5.5），式（5.6）和式（5.7）可得



或

．

由此解出

，

代入式（5.7）中得到

． （5.8）

这就是最速下降法用于二次函数的显式迭代公式．

例5.1 试用最速下降法求函数的极小点．迭代两次，计算各迭代点的函数值，梯度及其模，并验证相邻两个搜索方向是正交的．设初始点为．

解 与式（5.4）比较，得

，

梯度表达式是

，

由，计算出

，

，

．

因为目标函数是二次的，可以使用式（5.8），所以有

，

计算





，



因为



由此说明相邻两个搜索方向与、与是正交的．

三、最速下降法有关说明

最速下降法的优点是算法简单，每次迭代计算量小，占用内存量小，即使从一个不好的初始点出发，往往也能收敛到局部极小点，但它有一个严重缺点就是收敛速度慢．

|  |
| --- |
| 图5.3 |

沿负梯度方向函数值下降很快的说法，容易使人们产生一种错觉，认为这一定是最理想的搜索方向，沿该方向搜索时收敛速度应该很快，然而事实证明，梯度法的收敛速度并不快．特别是对于等值线（面）具有狭长深谷形状的函数，收敛速度更慢．其原因是由于每次迭代后下一次搜索方向总是与前一次搜索方向相互垂直，如此继续下去就产生所谓的锯齿现象（如图5.3所示）．即从直观上看，在远离极小点的地方每次迭代可能使目标函数有较大的下降，但是在接近极小点的地方，由于锯齿现象，从而导致每次迭代行进距离缩短，因而收敛速度不快．

§5.2 Newton法

如果目标函数在上具有连续的二阶偏导数，其Hesse矩阵正定并且可以表达成为显式（今后为了简便起见，记，那么可以使用下述的Newton法．这种方法一旦好用，收敛速度是很快的．它是一维搜索Newton切线法的推广．

一、Newton法基本原理

为寻求收敛速度快的算法，我们考虑在应用基本迭代公式中，每轮迭代在迭代的起始点处用一个适当的二次函数来近似该点处的目标函数，由此用点指向近似二次函数极小点的方向来构造搜索方向

（如

|  |
| --- |
| 图5.4 |

图5.4所示）．

下面具体讨论Newton法．

设最优化问题为，其中二阶可导，Hesse矩阵正定．

不妨设经过次迭代已获点，现将在处展成二阶泰勒公式，于是有



显然是二次函数，特别由假设知还是正定二次函数，所以是凸函数且存在唯一全局极小点．为求此极小点，令



即可解得

，

亦即

． （5.9）

对照基本迭代公式



易知，式（5.9）中的搜索方向

，

步长因子．换句话说从点出发沿搜索方向



并取步长即可得的极小点．因此，



是直指点处近似二次函数的极小点的方向．此时称



为从点出发的Newton方向．从初始点开始，每一轮从当前迭代点出发，沿Newton方向并取步长的算法称为Newton法．

### 二、Newton法迭代步骤

已知目标函数及其梯度，Hesse矩阵，终止限．

（1）选定初始点；计算；置．

（2）计算．

（3）由方程解出．

（4）计算．

（5）判别终止准则是否满足：若满足，则打印最优解，结束；否则，置，转（2）．

Newton法的流程如图5.5所示．

例5.2 试用Newton法求的极小点，初始点取为．

解 梯度为，

Hesse矩阵为

，

其逆矩阵为

．

由迭代公式（5.13）得第1 迭代点为



由于此时，停止迭代得，因此，它就是极小点．

从上述例5.2我们看到，用Newton法求解，只经一轮迭代就得到最优解．这一结果并不是偶然的，因为从Newton方向的构造我们知道，对于正定二次函数，Newon方向就是指向其极小点的方向．因此，用Newton法解目标函数为正定二次函数的无约束最优化问题，只需一次迭代就可得最优解．

对于目标函数是非二次函数的非约束最优化问题，一般地说，用Newton法通过有限轮迭代并不能保证可求得最优解．但由于目标函数在最优解附近能近似于二次函数，因此当先取接近于最优解的初始点使用Newton法求解时，其收敛速度一般是较快的．事实上，可以证明在初始点离最优解不远的条件下，Newton法是二次收敛的．但是当初始点选得离最优解太远时，Newton法并不一定是收敛的方法，甚至连其下降性也很难保证．

### 三、Newton法有关说明

Newton法收敛速度非常快具有二次收敛的优点，但它存在下面四个严重的缺点：

（1）尽管每次迭代不会使目标函数上升，但仍不能保证下降．当Hesse矩阵非正定时，Newton法的搜索将会失败．

（2）对初始点要求严格．一般要求比较接近或有利于接近极值点，而这在实际计算中是比较难办的．

（3）在进行某次迭代时可能求不出搜索方向．由于搜索方向为

，

若目标函数在点Hesse矩阵是奇异的，则不存在，因而不能构成newton方向，从而使迭代无法进行．

（4）newton方向构造困难，计算相当复杂，除了求梯度以外还需计算Hesse矩阵及其逆矩阵，占用机器内存相当大

## §5.3 修正Newton法

### 一、修正Newton法基本原理

为了克服Newton法的缺点，人们保留选取Newton方向作为搜索方向，摒弃其步长恒取1，而用一维搜索确定最优步长，由此产生的算法称为修正Newton法（或阻力Newton法）．

### 二、修正Newton法迭代步骤

已知目标函数及其梯度，Hesse矩阵，终止限．

（1）选取初始点，令．

（2）求梯度向量，计算．若，停止迭代输出，否则转（3）．

（3）构造Newton方向．计算

，

取

．

（4）进行一维搜索．求，使得

．

令，转（2）．

修正Newton法的流程如图5.6所示．

### 三、修正Newton法有关说明

修正Newton法克服了Newton法的缺点．特别是，当迭代点接近于最优解时，此法具有收敛速度快的优点，对初始点的选择要求不严．但是，修正Newton法仍需要计算目标函数的Hesse矩阵和逆矩阵，所以计算量和存贮量均很大．另外，当目标函数的Hesse矩阵在某点处出现奇异时，迭代将无法进行，因此修正Newton法仍有相当的局限性．

## §5.4 共轭方向法

构成各种不同最优化方法，往往取决于如何从基本迭代公式中确定搜索方向．在最速下降法中，由于取，从而导致搜索路线出现锯齿状，收敛速度降低；而在Newton法中，由于选取，收敛速度虽很快，但计算量很大，特别是还要求Hesse矩阵正定，从而导致该算法对初始点选择要求过严．

下面我们将给大家介绍一种新的最优化方法，它的计算量小，收敛速度没有Newton法快，但比最速下降法快得多的新算法，称为共轭方向法．

### 一、共轭方向法基本原理

首先介绍共轭方向概念及其些性质．

定义5.1 设是对称正定矩阵．对于非零向量，若有

 （5.10）

则称与相互共轭或正交的．

对于非零向量组，若有

 （5.11）

则称是共轭方向组或正交方向组，也称它们是一组的共轭方向．

在上述定义中若（阶单位矩阵），则式（5.10）和式（5.11）成为和．由此可知，共轭概念是正交概念的推广，共轭方向是正交方向的推广．

定理5.1 设是正定矩阵，是非零向量．若是一组共轭方向，则它们是线性无关的．

证明 设有一组实数，使得

．

依次以左乘上式得到

， （5.12）

因为是一组的共轭方向，故有

．

又由于*A*是正定矩阵，所以对于，有

．

把以上两式用于式（5.12），便可推知

．

由此证明是线性无关．

考虑以二次函数为目标函数的无约束极小化问题

， （5.13）

其中是对称正定矩阵，，有如下定理：

定理5.2 设是对称正定矩阵，是一组的共轭方向．对于问题（5.13），若从任意点出发依次沿进行一维搜索，则至多经过轮迭代，可得问题（5.13）的最优解．

证明 设从点出发依次按方向进行一维搜索产生的迭代点是

，， （5.14）

其中最优步长是通过下式

，（5.15）

来确定．

又由和式（5.14）可推知

，

即

. （5.16）

利用式（5.16）有



对上式右乘可得



因为是共轭方向，故得



或

． （5.17）

另外，由式（5.15）可知

 （5.18）

因为

，

所以由式（5.18）有

．

由式（5.17）和上式，对于，我们得到

．

特别地，当时有

 ． （5.19）

由于是一组的共轭方向，由定理5.1知它们是线性无关的，因而向量可表示为这些向量的线性组合

为这些向量的线性组合

，

其中是实数，．

所以由式（5.19）有

，

即有．于是，得

，

即是的驻点．又因为是对称正定，故是凸函数，因而是问题（5.13）的最优解．这说明，至多经过次迭代必可求得（5.13）的最优解．

通常，我们把从任意点出发，依次沿某组共轭方向进行一维搜索的求解最优化问题的方法，叫做共轭方向法．

为了直观起见，首先考虑二维情况．现在我们把下降法用于形式为（5.13）的二元二次函数．

任选初始点，从出发沿某个下降方向作一维搜索得到（如图5.7所示）．

|  |
| --- |
| 图5.7 |

由式（4.2）知

， （5.20）

而且向量所在的直线必与某条等值线（椭圆）相切于点．下一次迭代，如果按最速下降法选择负梯度方向为搜索方向，那末将要发生锯齿现象，为了克服这种现象，我们设想，下一次迭代的搜索方向将直指极小点．如果能够选定这样的搜索方向，那么对于二元二次函数只须顺次进行两次直线搜索就可以求到极小点了．下面来讨论这样的方向应该满足什么条件，怎样确定．

因为直接指极小点，所以可写成

， （5.21）

其中是最优步长因子，显然，当时，．

对（5.13）中的函数求导数得

， （5.22）

因为是极小点，所以有

．

将式（5.21）代入上式，并由式（5.22）可得

．

等式两边同时左乘，并注意到式（5.20）和，于是有

． （5.23）

这就是为使直指极小点所必须满足的条件．显然式（5.23）中和是的共轭向量．由式（5.20），不难给出的表达式．设

， （5.24）

两边同时左乘，有

．

由此解出

，

代回到式（5.24），得

．

一般地，在维空间中可以找出个互相共轭的方向，对于元正定二次函数，从任意初始点出发，顺次沿这个共轭方向最多作次直线搜索就可以求得目标函数的极小点．这就是共轭方向法的算法形成的基本思想．

对于元正定二次目标函数，从任意初始点出发，如果经过有限次迭代就能够求得极小点，那末这种算法称为具有二次终止性．例如Newton法对于二次函数只须经过一次迭代就可以求得极小点，因此是二次终止的；而最速下降法不具有二次终止性．共轭方向法（包括共轭梯度法，变尺度法等）是二次终止的．一般来说，具有二次终止性的算法，在用于一般函数时，收敛速度较快．

### 二、共轭方向法迭代步骤

已知具有正定矩阵的二次目标函数和终止限．

（1）选定初始点和具有下降的方向，置．

（2）作直线搜索．

（3）判别是否满足：若满足，则打印，结束；否则转（4）．

（4）提供共轭方向使得

．

（5），转（2）．

### 三、共轭方向法有关说明

上述算法针对二次目标函数，但也可用于一般的非二次函数．在寻优过程中因舍入误差不能满足时，可取为新的初始点，再重复前面的过程．

## §5.5 共轭梯度法

如果在共轭方向法中初始的共轭向量恰好取为初始点处的负梯度，而以下各共轭方向由第迭代点处的负梯度与已经得到的共轭向量的线性组合来确定，那么就构成了一种具体的共轭方向法．因为每一个共轭向量都是依赖于迭代点处的负梯度而构造出来的，所以称为共轭梯度法．

### 一、共轭梯度法基本原理

设从任意点出发，第一个搜索方向取为处的负梯度方向

．

当搜索得到点后，设以下按



来产生搜索方向．为了使选择使所产生的和是共轭，以右乘上式的两边，于是有

．

因为要使和是共轭，应有，故由上式得

．

综上所述，可以生成个方向

 （5.25）

式（5.25）中含有问题（5.13）的目标函数系数矩阵，这对于目标函数是非二次函数的问题是不方便的．通过简化（详见参考文献[26]，一般可以利用目标函数的梯度信息，来产生个共轭方向



由此得共轭梯度法．

例5.3 用共轭梯度法求

，

其中，选初始点为．

解 显然

，

．

令 ，

则 ，

∴，

∴，，

．

所以新的搜索方向



由此，有

，

并且可推知

，

∴．

因而得下一迭代点

，

由于停止迭代输出所求得

．

例5.3的迭代的路径如图5.10所示

三、共轭梯度法有关说明

|  |
| --- |
| 图5.10 |

实际上，可以把共轭梯度法看作是最速下降法的一种改进．当令时，就变为最速下降法．共轭梯度法由于不涉及矩阵，仅仅存储向量，因而存储量小，适合于维数较高的优化问题．另外，共轭梯度法不要求精确的直线搜索．但是，不精确的直线搜索可能导致迭代出来的向量不再共轭，从而降低方法的效能．克服的办法是，重设初始点，即把经过次迭代得到的作为初始点重新迭代．

## §5.6 变尺度法

我们知道Newton法最突出的优点是收敛速度快，在这一点上其它算法无法比拟的．因此，建议凡是Hesse矩阵比较容易求出的问题尽可能使用Newton法求解．然而Newton法还有一个严重缺陷，就是每次迭代都要计算目标函数的Hesse矩阵和它的逆矩阵，当问题的维数较大时，计算量迅速增加，从而就抵消了Newton法的优点．为此，人们开始寻找一种算法既可以保持Newton法收敛速度快的优点，又可以摆脱关于Hesse矩阵的计算，这就是本节要给大家介绍的变尺度算法．

变尺度法是一种非常好的方法．其中DFP算法和BFGS算法，可以说直到目前为止是在不用Hesse矩阵的方法中最好的算法．

### 一、变尺度法基本原理

在Newton法中，基本迭代公式

，

其中，，．令

，

于是有

 （5.26）

其中是初始点，和分别是目标函数在点的梯度和Hesse矩阵．为了消除这个迭代公式中的Hesse逆矩阵可用某种近似矩阵来替换它，即构造一个矩阵序列去逼近Hesse逆矩阵序列．此时式（5.26）变为

．

事实上，式中无非是确定了第次迭代的搜索方向．为了取得更大的灵活性，我们考虑更一般的迭代公式

， （5.27）

其中步长因子通过从出发沿作直线搜索来确定．式（5.27）是代表很广的一类迭代公式．例如，当（单位矩阵）时，它变为最速下降法的迭代公式．为使确实与近似并且有容易计算的特点，必须对附加某些条件

第一，为保证迭代公式具有下降性质，要求中的每一个矩阵都是对称正定的．

理由是，为使搜索方向是下降方向，只要



成立即可，即



成立．当对称正定时，此式必然成立，从而保证式（5.27）具有下降性质．

第二，要求之间的迭代具有简单形式．显然，

 （5.28）

是最简单的形式了．其中称为校正矩阵，式（5.28）称为校正公式．

第三，必须满足拟Newton条件．所谓拟Newton 条件由下面的推导给出．

设迭代过程已进行到步，和均已求出，现在推导所必须满足的条件．

设目标函数具有连续的二阶偏导数．现在将在处展成二阶泰勒公式



．

令，于是有

，

即

．

当正定时，

．

当用近似时，由此看出也必须满足

． （5.29）

换句话说，式（5.29）就是称为近似Newton条件．为了今后书写方便，记

，

，

于是拟Newton条件可写为

 ． （5.30）

### 二、变尺度法迭代步骤（拟Newton法）

已知目标函数及其梯度，终止限．

（1）选定初始点；计算；选定初始矩阵，要求对称正定（例如，）；置．

（2）计算搜索方向．

（3）作直线搜索计算



（4）判别终止准则是否满足：若满足，则就是所求的极小点，打印，结束；否则转（5）．

（5）计算

（6）转（2）．

其中校正矩阵可由确定的公式来计算．不同的公式对应不同的拟Newton算法．拟Newton算法的流程如图5.11所示．

以下几段将要讨论各种公式的构成以及相应算法．

但是不论哪个公式都必须满足拟Newton条件，由式（5.30）和式（5.28）知，必须满足



或

 ． （5.31）

由此可见，与，和有关．

满足式（5.31）的有无穷多个，因此上述拟Newton算法构成一簇算法．下面分别介绍两个常用的公式．

#### （一）DFP算法

DFP算法首先是由Davidon（1959年）提出来的，后来，Fletcher和Powell（1963年）对Davidon的方法作了改进，最后才形成DFP算法．D、F、P是这三位学者名字的字头．这种算法是无约束最优化方法最有效的方法之一．

##### 1．DFP算法基本原理

考虑如下形式的校正公式

 ． （5.32）

其中，是待定维向量，，是待定常数．这时，校正矩阵是

．

现在来确定．

根据拟Newton条件，必须满足（5.31），于是有



或

．

满足这个方程的待定向量和有无穷多种取法，下面是其中的一种：

，

．

注意到和都是数量，不妨取

，

同时定出

．

将这两式代回式（5.32）得

． （5.33）

这就是DFP校正公式．

##### 2．DFP算法迭代步骤

在拟Newton算法中，只要把第（5）步改为计算式（5.33）而其它不变，该算法就为DFP算法了．但是由于计算中舍去误差的影响，特别是直线搜索不精确的影响，可能要破坏迭代矩阵的正定性，从而导致算法失效．为保证的正定性，采取以下重置措施：迭代次后，重置初始点和迭代矩阵，即以后重新迭代．

已知目标函数及其梯度，问题的维数，终止限．

（1）选定初始点．计算．

（2）置．

（3）作直线搜索；计算．

（4）判别终止准则是否满足：若满足，则打印，结束；否则转（5）．

（5）若，则置，转（2）；否则转（6）．

（6）计算

，

，

，

，

置，转（3）

DFP算法的流程留给读者自己绘制．

例5.4 用DFP算法求，取．

解 当我们取时，DFP法与最速下降法具有相同的第1迭代点，在例5.1中已作了计算

，

，

．

以下用DFP法作第二次迭代

，

，

按DFP算法中的第（6）步计算

．

因为

，

，

，

，

所以



搜索方向为

．

从出发沿进行直线搜索，即



由知，所以

．

由于，所以是极小点．

#### （二）BFGS算法

我们再介绍另一个有效和著名的变尺度算法．由于它是Broyden， Fletcher（1970年），Goldfarb（1969年）和Shanno（1970年）共同研究的结果，因而叫做BFGS法．

##### 1．BFGS算法基本原理

考虑如下形式的校正公式

， （5.34）

式中，

．

这时校正矩阵为

．

式（5.34）中有一个参数，它可以取任何实数，每取一个实数就对应一种拟Newton算法．

容易验证，当取时就是DFP校正公式．

令



就转变为著名的BFGS校正公式

．

##### 2．BFGS算法迭代步骤

已知目标函数及其梯度，问题的维数，终止限．

（1）选取初始点，初始矩阵，给定终止限．

（2）求初始梯度向量，计算，若，停止迭代输出，否则转（3）．

（3）构造初始BFGS方向，取，令转（4）．

（4）进行一维搜索，求，使得，令，转（5）．

（5）求梯度向量，计算，若，停止迭代输出；否则转（6）．

（6）检验迭代次数，若，令转（3）；否则转（7）．

（7）构造BFGS方向，用BFGS公式



计算，取，令，转（4）．

BFGS算法的流程如图5.12所示．

### 三、变尺度法有关说明

变尺度法中的两个重要算法DFP算法和BFGS算法，它们的迭代过程相同，区别仅在于校正矩阵选取不同，对于DFP法，由于一维搜索的不精确和计算误差的积累可能导致某一轮的奇异，而BFGS法对一维搜索的精度要求不高，并且由它产生的不易变为奇异矩阵．BFGS法比DFP法更具有好的数值稳定性，它比DFP法更具有实用性．有关DFP法及BFGS算法理论推导见参考文献[4]

## **§5.7 坐标轮换法**

坐标轮换法属于直接法．它既可用于无约束最优化问题的求解，又可经过适当的处理用于约束最优化问题的求解．

### 一、坐标轮换法基本原理

坐标轮换法的基本思想是把含有个变量的优化问题轮换地转化为单变量（其它变量视为常量）的优化问题．所谓单变量优化问题就是沿某个坐标轴方向进行一维搜索的问题．

坐标轮换法的寻优思路是：先选定一个初始点作为第一轮搜索的始点，依次沿个坐标轴方向进行一维搜索，每次只在一个坐标轴方向上改变相应变量的值，其它个变量均保持不变．在沿第一个坐标轴方向进行一维搜索得到目标函数值的最小点（或近似最小点）后，再以此点作为始点转到沿第二个坐标轴方向进行一维搜索得到，直到沿第个坐标轴方向搜索结束得到为一个循环．如果不满足收敛准则，则以作为初始点转入下一轮循环，直到经过次循环，获得满足收敛准则的点，即作为最优点．对于二维最优化问题，其搜索过程如图5.13所示．

在坐标轮换法中，沿各个坐标轴方向进行一维搜索时，常选用最优步长法或加速步长法．加速步长法从初始点出发，沿搜索（坐标轴）方向先取一个较小的步长，作前进（或后退）试探．如试探成功（目标函数值有所减小），则按步长序列，加大步长（注意每次加大步长都是由初始点算起），直至试探失败（目标函数值比前一次的有所增加）时，则取其前一次的步长作为沿这个坐标轴方向搜索的最优步长，并计算出该方向上的终止点，而后以这个终止点为始点再进行下一坐标轴方向的搜索，并重复上述步骤．如此迭代下去，直到找到最优点．

本节只用一维搜索法来确定最优步长．

### 二、坐标轮换法迭代步骤

取初始点，置坐标轴搜索方向：

，

，

……

．

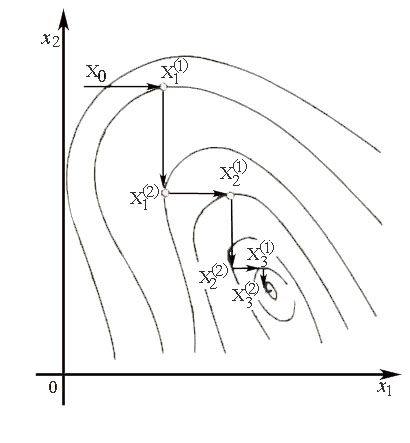


图5.13

首先沿方向进行一维搜索，求出该方向上目标函数的极值点；再以为初始点沿方向进行一维搜索，得到极值点；仿此依次沿进行一维搜索，最终得到极值点．这就完成了第一轮循环的搜索．

如果能够满足收敛准则，即可停止搜索，以作为输出．否则，继续以为初始点，进行第二轮循环，依次沿进行一维搜索，得到第二循环的极值点．如此进行下去，直至最终找到满足收敛准则（终止准则）的点，即求得了最优解，再求出目标函数值．具体迭代过程如下：

已知目标函数，终止限．

（1）任选取始点作为第一轮循环的初始点，．

（2）置搜索方向依次为

，

，

……

．

（3）按下式求最优步长并进行迭代计算

，

，

式中为循环次数，；为该循环中一维搜索的序号，；为利用一维搜索求出的最优步长．

（4）如果，即转（5）；如果，则转（3）．

（5）收敛性准则，若满足判别式，即停止迭代，输出最优解及；若不满足，则令转（3）．

坐标轮换法的计算流程如图5.14所示．

例5.5 用坐标轮换法求，初始点．

解 从初始点出发，依次沿方向搜索，以第一步为例，从出发，沿出发，沿

方向搜索，求得点



求

，

即

，

得，即取，于是得

．

再从出发，沿方向搜索，求得点



求



可得，即取，于是有

．

终止判别

，

因终止条件不满足，需继续迭代，取，进行第二轮循环迭代，各轮迭代计算数据见表5.1，最优解为．

表5.1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 循环  迭代  序号 |  |  | | |  | | |  |  | 是否满足收敛准则 |
|  | |  |  | |  |
| 1 | 0.00，3.00 | 3.13 | 3.13，1.56 | | 1.44 | 3.13，1.56 | | 1.63 | 3.45 | 否 |
| 2 | 3.13，1.56 | －0.50 | 2.63，1.56 | | －0.25 | 2.63，1.31 | | 0.16 | 0.56 | 否 |
| 3 | 2.63，1.31 | －0.19 | 2.44，1.31 | | －0.09 | 2.44，1.22 | | 0.04 | 0.21 | 否 |
| 4 | 2.44，1.22 | －0.09 | 2.35，1.22 | | －0.05 | 2.35，1.17 | | 0.015 | 0.10 | 否 |
| 5 | 2.35，1.17 | －0.06 | 2.29，1.17 | | －0.03 | 2.29，1.14 | | 0.007 | 0.06 | 否 |
| 6 | 2.29，1.14 | －0.04 | 2.25，1.14 | | －0.02 | 2.25，1.12 | | 0.004 | 0.045 | 否 |
| 7 | 2.25，1.12 | －0.03 | 2.22，1.12 | | －0.01 | 2.22，1.11 | | 0.002 | 0.03 | 是 |
|  |  |  |  |  |  |  |  |  |  |  |

三、坐标轮换法有关说明

坐标轮换法的优点是算法简单，计算量小，其缺点是计算效率低，对高维问题尤为突出．因此，坐标轮换法通常用于维数较低的优化问题（一般）．

## §5.8 单纯形法

单纯形法是利用对简单几何图形各顶点的目标函数值作相互比较，在连续改变几何图形的过程中，逐步以目标函数值较小的顶点取代目标函数值最大的顶点，从而进行求优的一种方法，属于直接法之一．

### 一、单纯形法基本原理

现以求二元函数的极小点为例，说明单纯形法形成原理．

设二元函数在平面上取不在同一条直线上的三个点，和，并以它们为顶点构成一单纯形——三角形．算出各顶点的函数值，，，比较其大小，现假定比较后有

 ．

这说明点最差，点最好，点次差．为了寻找极小点，一般来说应向最差点的反对称方向进行搜索．以记为的中点（如图5.15所示），在的延长线上取点，使



称为关于的反射点．

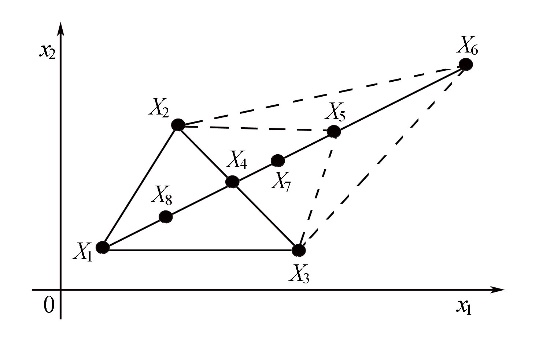


图5.15

算出的函数值，可能出现以下几种情形：

(1) 

这说明搜索方向正确，可进一步扩大效果，继续沿向前搜索，也就是向前扩张．这时取

 ，

其中为扩张因子，一般取．

如果，说明扩张有利，就可以点代替点构成新的单纯形．如果，说明扩张不利，舍序，仍以代替构成新的单纯形．

(2) 

这说明搜索方向正确，但无须扩张，以代替构成新的单纯形．

(3) 

这表示点走得太远，应缩回一些．若以表示压缩因子，则有

， （5.35）

常取为0.5．以代替构成新的单纯形．

(4) 

这时应压缩更多一些，将新点压缩至至之间，令

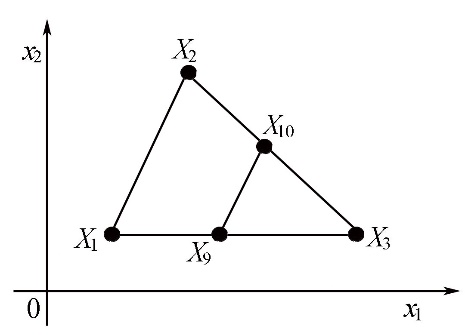
． （5.36）

注意，将式（5.35）中的代之以，即可得式（5.36）．如果，则以代替构成新的单纯形，否则认为方向上所有点的函数值都大于，不能沿此方向搜索．这时，可以以为中心进行缩边．若使顶点和向移近一半距离（如图5.16所示），得新单纯形．以此单纯形为基础再进行寻优．

以上说明，不管哪种情况，我们都可以得到一个新的单纯形，其中至少有一顶点的函数值比原单纯形为小．如此继续，直至满足收敛终止准则．

在维情况下，一个单纯形含有个顶点，计算工作量较大，但原理和上述二维情况相同．

|  |  |
| --- | --- |
| |  | | --- | | 图5.16 | |



### 二、单纯形法迭代步骤

已知设为维变量，目标函数为，终止限为．

(1) 构成初始单纯形

在维空间中选初始点（离最优点越近越好），从出发，沿各坐标方向以步长得个顶点，．这样选择顶点可保证向量组

这样选择顶点可保证向量组



线性无关．否则，就会使搜索范围局限在较低维的空间内，有可能找不到极小点．当然，在各坐标方向可以走不同的距离．

步长的范围可取0.5~15.0，开始时常取，接近最优点时要减小，例如取0.5~1.0．

(2) 计算各顶点的函数值

，

比较各函数值的大小，确定最好点、最差点和次差点，即

，，

，，

．

(3) 计算之外各点的“重心”

，

求出反射点

．

(4) 当时，需要扩张．令

．

如果，则以代替形成一新单纯形；否则，以代替构成新单纯形．然后转(8)．

(5) 当时，以代替构成新单纯形，然后转(8)．

(6) 当时，则需要收缩．即令

． （5.37）

以代替得新单纯形，并转(8)．

(7)　当时，令

．

如果，则将单纯形缩边，可将向量的长度缩小一半，即



这样可得一新单纯形．否则就以代替得新单纯形．然后转(8)．

(8) 收敛性检验

每次迭代得到新单纯形后，即应进行收敛性检验，如满足收敛指标，则迭代停止，即为所求的近似解．否则，继续进行迭代计算． 通常所用的收敛准则是



或

，

式中和为预先给定的允许误差．

单纯形法的流程图留给读者自己完成．

例5.6 试用单纯形法求的极小值．

解 选，并取和．这三点不在一条直线上，用它们作为初始单纯形的顶点（如图5.17所示）．然后计算各顶点的函数值：，可知为最差点，为最好点．以表示和的重心，则

，

反射点

，

．

由于，故需扩张．取，则

，

．

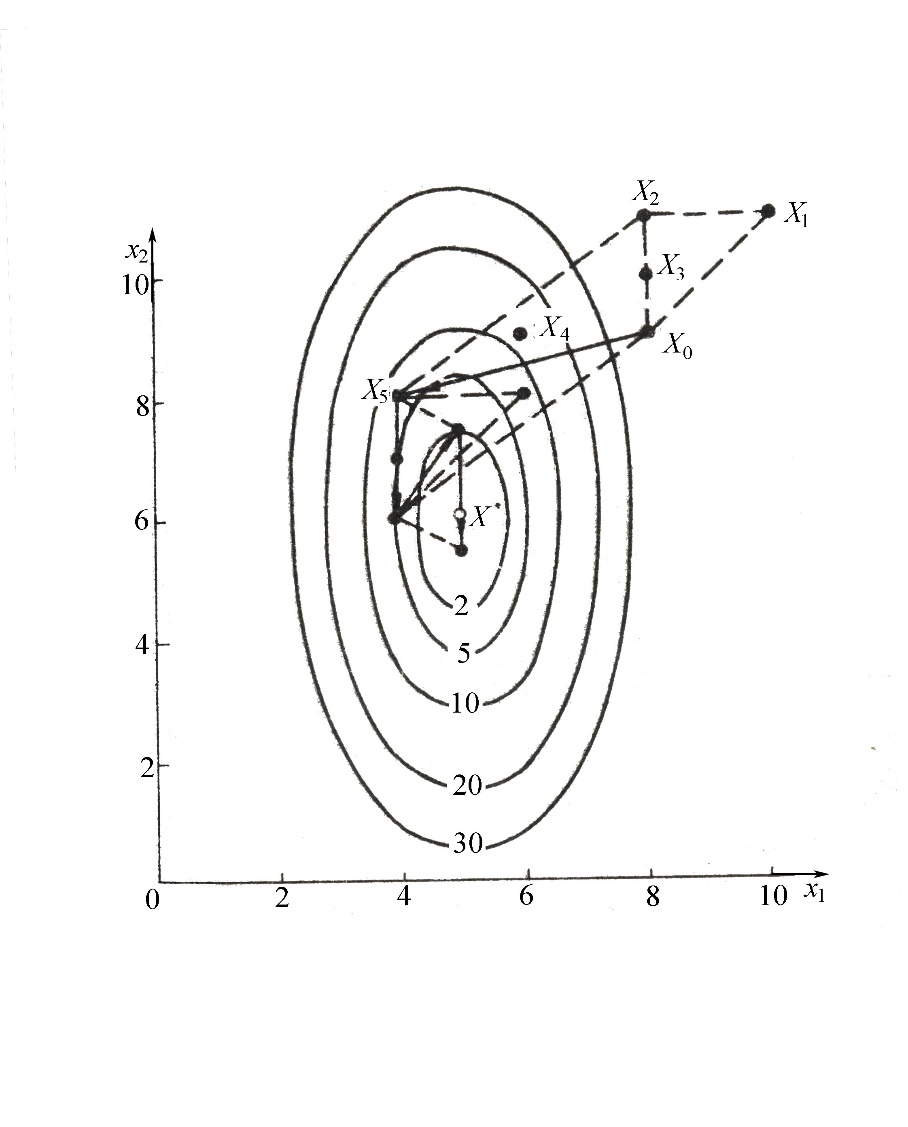
由于，故需扩张．取，则

，

．

因为，故以代替，由，和构成新单纯形，然后进行下一个循环．该问题的最优解为，=0．经32次循环，可把目标函数减小到．在图5.17中给出了前几次迭代的情形．

|  |  |
| --- | --- |
| |  | | --- | | 图5.17 | |



三、单纯形法有关说明

本算法上机占用内存很少，对变量不多且精度要求不高的问题此法很方便，但当变量个数多于10以上，此法就显得不十分有效．

# 第六章 最优化问题程序设计方法

最优化问题程序设计方法是二种规格化的设计方法，它首先要求将工程设计问题按优化设计所规定的格式建立数学模型，然后选择合适的最优化方法编写出计算机程序，最后通过计算机计算自动获得最优方案．

## §6.1 最优化问题建模一般步骤

### 一、建立最优化问题的数学模型

工程优化问题的数学模型，是要把工程设计中的问题用数学关系式准确表达出来．为达到这些要求，所建立起来的数学模型往往都是很复杂的．由于工程设计问题各有其特点，所以数学模型也是多种多样的．因此，在工程设计中正确地建立数学模型，不仅是一项艰巨复杂的工作，而且也是解决优化设计问题的关键与前提．在很多情况下，建立优化问题的数学模型工作一直是一项重要的研究课题．

优化数学模型包括三个内容：变量、目标函数及约束条件．它们的基本概念和意义已在第一章做了介绍．

### 二、选择合适的优化方法

各种优化方法都有各自的特点和一定的适用场合．根据具体的最优化问题，适当地选择优化方法才会有较好的效果．

选择优化方法时，主要考虑的因素是：目标函数的维数与连续性；它的一阶、二阶偏导数是否存在，是否易于求得；约束条件是等式约束，还是不等式约束或两者兼有等不同情况．

一般地，对于维数较低的问题应选用结构简单、易于编程的方法．对于维数较高的问题，效率就显得十分重要，应选择收敛速度较快的方法．对于求导困难或导数不存在的优化问题应选用直接法．

### 三、制订流程图和编写源程序

为了使编写源程序有正确的思路，必须先根据具体最优化问题制定一个较详细的流程图．该图应反映优化计算的步骤及各种运算之间的逻辑关系．流程图既便于程序的编制，又便于使用者对程序的阅读．

编写源程序是一种技巧性较高而且很细致的工作．即使是一个较为简单的最优化问题，也需要考虑许多方面的因素．若某些优化方法已有比较成熟的源程序，应尽量优先采用，以期缩短编程时间和提高计算的可靠性与有效性．

一个新编制的程序，即使在编写过程中已经作过周密的考虑，也很难在计算机上一次通过，总会发生这样或那样的障碍，可能是语法规则方面的错误，也可能是运行错误等等．因此，新编程序必须经过调试和试算后才能确认它的正确性．试算是必要的一环．所谓试算，是用一个比较简单的、已经作好标准答案的题目用编好的源程序运算，观察结果是否正确，以期检查程序的正确性，试算通过后再作正式计算，其结果就比较可信了．

分析优化结果的目的在于考证优化结果的正确性与实用性．尽管最优化方法本身是一种科学方法，是可以信赖的．但由于实际工程问题的复杂性和某些算法在研究上的不完善性，或由于设计者在建模中失误与疏忽，都会导致计算结果与实际情况不相符，甚至有时是荒谬的．所以对优化结果要进行分析．如果经分析，发现计算结果存在问题，则需寻查原因，进行调整，修改，直至获得完全符合实际情况为止．

最后还需指出，一般情况下通过优化计算所得的最优解只能保证是一个局部最优解．只有凸规划问题的局部最优解才是全局最优解．为了得到全局最优解，只要多选几个分布在不同位置的初始点进行优化计算．若所得各解都归于同一解上去，可认为所得解为全局最优解，否则应从这些解中择其目标函数最小者做为全局最优解．

§6.2常用最优化方法的特点及选用标准

前几章给大家介绍了各种常用最优化方法，它们在实际使用中都有一定效果，各有其特点，看法也不太一致．到底哪种算法较好?哪种较差?为了比较它们的特性，我们有必要建立合理的评价准则．

最优化方法评价准则主要有以下几方面：

（1）可靠性

所谓可靠性是指算法在合理的精度要求下，在一定允许时间内能解出各种不同类型的最优化问题的成功率．能够解出的问题越多，则算法的可靠性越好．简言之，如果算法对于不同的问题有时好，有时差，随机性太大，实际使用时没有把握，则可靠性就差．

（2）有效性

这是指算法的解题效率而言的．有效性常用两种衡量标准，其一是用同一题目，在相同的精度要求和初始条件下，比较占用机时数多少；其二是在相同精度要求下，计算同一题目获得最优解时所需要的计算目标函数值次数及导数值次数．

（3）简便性

简便性包括两个方面的含意．一方面是指实现这种算法人们所需要的准备工作量的大小．例如，编制程序的复杂程度，程序调试出错率的高低，算法中所用调整参数的多少等等．另一方面是指算法所占用存贮单元的数量，如果某些算法占用单元数很大，就会对机型提出特殊要求，显然这对使用者是不方便的．

由上面的三个评价准则可以看出，要断然地肯定某算法最好或最坏是不可能的．因为各种算法就上面三个准则作评价时一般是各有长短，而且由于目标函数的多样性，各种算法对不同目标函数所体现出来的准则衡量结果也有差异．因此算法的评价实际上是一个比较复杂的问题．

下面分别就常用无约束方法和常用约束方法作一概略评论，以供参考．

一、常用无约束最优化方法评价准则

表10.1是几种常用无约束方法的比较．

表10.1

|  |  |  |
| --- | --- | --- |
| 方法 | 算法特点 | 适用条件 |
| 最速下降法 | 属于间接法之一．方法简便，但要计算一阶偏导数，可靠性较好，能稳定地使函数下降，但收敛速度较慢，尤其在极值点附近更为严重． | 适用于精度要求不高或用于对复杂函数寻找一个好的初始点． |
| Newton法 | 属于间接法之一．需计算一、二阶偏导数和Hesse矩阵的逆阵，准备工作量大，算法复杂，占用内存量大．此法具有二次收敛性，在一定条件下其收敛速度快，要求迭代点的Hesse矩阵必须非奇异且定型（正定或负定）．对初始点要求较高，可靠性较差． | 目标函数存在一阶＼二阶偏导数，且维数不宜太高． |
| 共轭方向法 | 属于间接法之一．具有可靠性较好，占用内存少，收敛速度较快． | 适用于维数较高的目标函数 |
| 变尺度法 | 属于间接法之一．具有二次收敛性，收敛速度快．可靠性较好，只需计算一阶偏导数．对初始点要求不高，优于Newton法．因此，目前认为此法是最有效的方法之一，但需内存量大．对维数太高的问题不太适宜． | 适用维数较高的目标函数（*n*=10～50）且具有一阶偏导数． |
| 坐标轮换法 | 最简单的直接法之一．只需计算函数值，无需求导，使用时准备工作量少．占用内存少．但计算效率低，可靠性差． | 用于维数较低（*n*<5）或目标函数不易求导的情况． |
| 单纯形法 | 此法简单，直观，属直接法之一．上机计算过程中占用内存少，规则单纯形法终止条件简单，而不规则单纯形法终止条件复杂，应注意选择，才可能保证计算的可靠性． | 可用于维数较高的目标函数． |

二、常用约束最优化方法评价标准

表10.2列出了几种常用约束方法的比较．

表10.2

|  |  |  |
| --- | --- | --- |
| 方法 | 方法特点 | 适用场合 |
| 外点法 | 将约束优化问题转化为一系列无约束优化问题．初始点可以任选，罚因子应取为单调递增数列．初始罚因子及递增系数应取适当较大值． | 可用于求解含有等式约束或不等式约束的中等维数的约束最优化问题． |
| 内点法 | 将约束优化问题转化为一系列无约束优化问题．初始点应取为严格满足各个不等式约束的内点，障碍因子应取为单调递减的正数序列．初始障碍因子选择恰当与否对收敛速度和求解成败有较大影响． | 可用于求解只含有不等式约束的中等维数约束优化问题． |
| 混合罚函数法 | 将约束优化问题转化为一系列无约束优化问题，用内点形式的混合罚函数法时．初始点及障碍因子的取法同上；用外点形式的混合罚函数时．初始点可任选，罚因子取法同外点法相同． | 可用于求解既有等式约束又有不等式约束的中等维数的约束优化问题． |
| 约束坐标轮换法 | 由可行点出发，分别沿各坐标轴方向以加步探索法进行搜索，使每个搜索点在可行域内，且使目标函数值下降． | 可用于求解只含有不等式约束，且维数较低（*n*<5），目标函数的二次性较强的优化问题． |
| 复合形法 | 在可行域内构造一个具有个顶点的复合形，然后对复合形进行映射变化，逐次去掉目标函数值最大的顶点． | 可用于求解含不等式约束和边界约束的低维优化问题． |

## §6.3 最优化问题编程的一般过程

本节将讨论有关最优化问题编程的一般过程问题，它是优化工作的一个重要方面．我们所关心的是如何针对不同的优化问题编写供使用者调用方便、界面友好的优化子程序或实用程序库，而不是编写最优化方法程序本身．

最优化问题的编程一般分为以下几个阶段：

### 一、准备阶段

确定任务，选用计算机及所用语言．准备原始资料，包括最优化问题的数学模型与原始数据等．

### 二、编制程序

(1) 先画出代表整个计算流程的总框图．如有必要再画出有关局部过程的详细框图．对较大的最优化问题，它给出了整个构思及各分支的相互联系，是必不可少的一步，但有些人忽视这点．当然框图也不宜过于详细，否则，反而主次不分，起不到应有作用．

(2) 根据框图，按算法语言规定逐条编写设计程序．特别要强调的是对一个程序必须多次反复核对，每一个符号都要认真对待．

程序结构有两种：整体式和模块式（或积木式），一般简单的程序采用只有一个主程序的整体式结构比较简便明了．对于大型复杂的最优化问题编程一般采用模块式程序设计方法．这样主控程序就可以较短，基本体现出总框图的思路，主控程序中需要用到有关的子程序时，只要调用它就行了．模块式程序设计具有各模块子程序便于分头编制，分段上机，调试，子程序调用方便，易于检查维护等优点．

### 三、调试程序

在计算机上编译检查输入的程序有无符号或语法错误．有的是编程对了，但输入时打错了，有的则是编写时就错了，可按相应的语法错误表逐个加以改正．要注意计算机检查错误是一气完成的，前面一句有错误常会引起后面一大串语法错误，有时只要把第一个错误改正后就解决了，当计算机打印出“有错”时首先要认真检查程序的错误，不要以为是计算机误动作．

只有认真地把计算机指出的每一个错误全部改正后才叫程序的“语法通过”或“编译通过”．要完成这一步应做到认真仔细、有耐心，以及逐渐建立起来的上机经验．

### 四、上机试算

程序的语法通过只表示语法上没有问题了，但程序中的公式在数学上有无意义（如个别公式中分母出现零值，负数开平方根或结果太大而超界等）？公式本身是否有错？执行路线对否？输入数据对否？这些都需要进行检查．因此必须先把一个已知其结果的（靠手算或借用别人已算出的）一套原始数据作为输入，然后将通过程序计算出输出结果与已知的结果对照逐一检查，直到每一项都与原来的已知结果相符才算合格，否则计算结果的可靠性是无法保证的．

如果一个程序具有多种功能，则必须对所有功能逐一进行试算，直到符合为止．

查出运行错误需有一定经验，行之有效的方法是在程序中插入必要的抽查中间结果的输出语句，以便迅速判断出错位置，缩小搜索范围．完成试算这一步后，所编程序方可以初步投入使用．

### 五、改进程序

在使用过程中还可能会碰到一些事先估计不到的意外情况，如原来准备的数组大小超界，曲线不够光滑等．同时也会发现一些值得改进的地方．这就要改编原来的程序，并可进一步明确这个程序的使用范围．经过一段使用检验才能认为程序可以正式交付使用．修改程序时要特别注意程序中上下的联系，不要只看局部不顾整体而任意改动，否则会越改越乱．一般而言，程序通用性越强，结构越复杂，出错机会也越多．如果编制程序时贪快，考虑不周到和不仔细，那么调试工作量会大出好几倍．一定要使一个错误也没有，计算机才能算出正确的答案．

### 六、在计算机上求解最优化问题的一些要点

计算机在求解优化问题时，第一次运算就不收敛，这时怎么办?优化计算失败的原因可能是下述情况中的一种：

(1) 起点（初始点）在非可行域内，因而一开始时就不运算；

(2) 在移动了一段距离后，停止在非可行域内；

(3) 是在可行域内移动，但停止在一个显然是非优化的位置上．

这些困境可能是由下述的一个或几个因素造成的：

(1) 起点在非可行域内，问题本身一开始就处于病态；

(2) 起点在非可行域内，所用的优化方法不能处理这种情况；

(3) 对所用特定的优化方法，问题处于病态；．

(4) 问题处于过约束状态，不可能有可行域或约束过紧，可行域过小．

为摆脱这种困境，要求程序能打印出目标函数、约束函数和变量的终值．这样就可发现失败是属于三种情况中的哪一种．

如果属第一、二种情况，运算停止在非可行域内，这可能是上述四种原因中的任一种造成的．建议首先从第四条着手．从有关技术规范的约束条件中，挑出一个或几个怀疑的对象，将它们放宽．也可根据打印出来的终值，找出满足的约束条件，将它们放宽．这可能要反复多次，直至运算移入可行域内．如果仍不成功，应更换所用的优化方法和初值再进行试算．

如果属第一种情况，可先用一种简单的最优化方法找出一个可行点，然后再用另一种方法求出最优解．

如果发生第三种情况，运算是在可行域内进行的，但后来停顿了．这种情况可能是第三条原因造成的．求解的问题对所用的算法处于病态，至少在一定范围内是这样的．应该用其它的方法．用调整程序参数的方法，如调整步长等，有时会起点作用，但一般是不会奏效的．

在进行迭代计算时，如果数据给得不合适，无论怎样迭代也达不到精度要求．遇到此种情况，计算机会算个没完，也不打印出任何结果，如不及时结束，永不休止，这就是称为“死循环”．

在最优化问题求解时迭代出现“死循环”的可能性最大．最简单的解决方法是在程序中专设一些计算开关变量，如*W*．每迭代一次，它自动加1，即．超过规定次数时，如次，则自动停止迭代，转向下一步或结束，同时打印出故障信息，以便查对．

## §6.4 优化问题设计实例

本节主要举出几个优化设计实例，阐明优化设计的一般步骤，包括设计问题的分析，数学模型的建立，优化方法的选用，优化结果分析，可供初学者参考．

### 实例一 圆柱螺旋弹簧的优化设计

弹簧的优化设计就是在保证满足工作能力要求的条件下，确定一组设计参数，使某个（或某几个）指标达到最优．弹簧的类型很多，这里主要介绍常用的压缩圆柱螺旋弹簧的优化设计．对其它类型的弹簧，亦可仿此进行．

### 一、优化设计数学模型的建立

（一）设计变量

在压缩圆柱螺旋弹簧的优化设计中，涉及的参数较多，如弹簧钢丝直径*d、*弹簧中径、弹簧总圈数、弹簧工作圈数、弹簧旋、细长、自由高度、节距等等．但是，在这些参数中，有些并不是独立的，它们可以表示成其它参数的函数．多数情况是把弹簧钢丝直径，中径及工作圈数作为设计变量，当然，也可根据具体要求的不同，选取其它参数作为设计变量．

（二）目标函数

在圆柱螺旋弹簧的优化设计中，可以作为优化目标的项目较多，如要求弹簧在满足工作能力条件下，质量最小或外廓尺寸最小 ；或在一定空间的限制下能贮存的能量最大，或要求弹簧的支态性能最好等．如果同时要求个项目达到某种指标，那就要建立几个目标函数，这就属于多目标优化设计问题了．

（三）约束条件

弹簧优化设计的约束条件，一般有强度约束、刚度约束、振动稳定性约束及弹簧尺寸约束等．

### 二、设计实例

试设计一压缩圆柱螺旋弹簧，要求其质量最小．弹簧材料为65Mn，最大工作载荷=40N，最小工作载荷为0，载荷变化频率=25Hz，弹簧寿命为104h，弹簧钢丝直径的取值范围为1~4mm，中径的取值范围为10~30mm，工作圈数不应小于4.5圈，弹簧旋绕比不应小于4，弹簧一端固定，一端自由，工作温度为50℃，弹簧变形量不小于10mm．

（一）确定优化设计的目标函数

取弹簧钢丝的直径、工作圈数及中径为变量，即

． （10.1）

本题的优化目标是使弹簧质量最小，圆柱螺旋弹簧的质量可表示为

， （10.2）

其中表示弹簧材料的密度，对于钢材；表示工作圈数；表示死圈数，常取，现取； 表示弹簧中径(单位：mm)；表示弹簧钢丝直径(单位：mm)．

将已知参数值代入式(10.1)和式(10.2)，经整理后得目标函数为

．

（二）确定约束条件

根据弹簧性能和结构上的要求，可写出如下的约束条件：

（1）强度条件

① 静强度条件．静强度条件是指在最大工作载荷作用下，弹簧不应产生永久变形，即

 （10.3）

其中*K*表示弹簧曲度系数，

；

表示弹簧的许用剪应力，取．

经代入式(10.3)整理后，得强度约束条件为

．

② 疲劳强度条件，载荷循环总次数，因，属于承受Ⅰ类载荷的弹簧，应进行疲劳强度计算，要求

， (10.4)

其中为计算安全系数，

， (10.5)

表示钢丝的脉动循环疲劳极限，取；表示弹簧中的最小剪应力，因，故Mpa；表示设计安全系数，取＝1.2．

将，值代入式（10.4）和式（10.5）整理后得

．

与静强度条件比较，该条件已包含了静强度条件，故略去静强度约束条件，改用

．

（2）刚度条件

弹簧刚度应满足变形要求即

，

式中为弹簧刚度变形量(单位：mm)，

，

其中*G*表示弹簧钢丝的剪切弹性模量，其值与温度有关，对于工作温度在50℃左右时可取；表示弹簧最小变形量，且由题可知．故得刚度约束条件为

．

（3）稳定性条件

避免压缩弹簧产生失稳现象，对于一端固定、另一端自由的弹簧，要求

，

式中为弹簧的自由高度，．

代入整理后，得稳定性约束条件为

．

（4）不发生撞圈现象

．

由的计算公式可知，此条件自然得到满足．

（5）不发生共振现象

，

其中表示弹簧固有频率，；表示外载荷变化频率，．故得约束条件

．

（6）弹簧旋绕比的限制

，

故得

．

（7）对的取值限制

显然，，且应取标准值，即1.0，1.2，1.6，2.0，2.5，3.0，3.5，4.0mm等；（式中50为估取的偏大值）；．

由上可知，此压缩圆柱螺旋弹簧的优化设计是一个三维的约束优化问题，其数学模型为

由上可知，此压缩圆柱螺旋弹簧的优化设计是一个三维的约束优化问题，其数学模型为

，



（三）优化方法与优化结果

因为维数等于3，是个小型约束优化问题，故可用直接优化方法求解．在此选用约束坐标轮换法求解．取初始设计参数为，初始质量为．

优化后的参数为，对应的质量为．

### 实例二 烟叶叶组配方的优化设计

在烟草工业中，叶组配方是指将各种不同类型、香型、产地、等级或不同性质、因素的烟叶，按照卷烟产品类型，以不同比例进行合理适宜的混合．叶组配方的配比是卷烟质量的重要保障．叶组配方的基本要求是质量、风格及成本稳定．叶组配方的优化设计就是在满足一定约束条件下，实现一定的优化目标．

### 一、优化设计数学模型的建立

(一)设计变量

在叶组配方中，设组成配方的单料烟叶有种，某种单料烟叶在叶组配方中所占的百分比为(=1,2,…,)．

(二)目标函数

目标函数可以是成本最低，也可以是叶组配方中某种化学成分的含量最低，也可以是感官质量中的某项指标或多项指标最优，也可以是多目标的组合(此时应考虑各目标之间的权重)．若某项约束指标不是寻优目标，则应建立此指标的约束方程；若某项约束指标是寻优目标，则建立相应的目标函数．如考虑成本最低，设叶组配方中第种单料烟叶的单价为(=1,2,…,)，则目标函数为

．

(三)约束条件

(1) 叶组配方物理属性的约束方程

①叶组配方含梗率的约束方程

设组成叶组配方的某种单料烟叶的含梗率为(=1,2,…,)，含梗率的下限和上限分别为和，则叶组配方含梗率的约束方程可表示为

．

② 叶组配方阴燃速率、平衡水分、填充性的约束方程

设组成叶组配方的某单料烟叶的阴燃速率、平衡水分、填充性为，(=1,2,3；=1,2,…,)，阴燃速率、平衡水分和填充性的下限和上限分别为和，则叶组配方的阴燃速率、平衡水分和填充性的约束方程可表示为

．

(2) 叶组配方化学成分的约束方程

①叶组配方总糖、总氮、烟碱、PH值的约束方程

设组成叶组配方的某单料烟叶的总糖、总氮、烟碱、PH值为(=1,2,3,4；=1,2,…,),总糖、总氮、烟碱、PH值的下限和上限分别为和，则叶组配方总糖、总氮、烟碱、PH值的约束方程可表示为

．

②叶组配方的糖/氮，糖/碱

按照上面的假设，若叶组配方的糖/氮的下限和上限分别为和，糖/碱的下限和上限分别为和，则叶组配方的糖/氮，糖/碱的约束方程可表示为

， ．

(3) 叶组配方烟气化学成分的约束方程

设叶组配方的焦油量、烟碱量和CO量的下限和上限分别为和(=1,2,3)，则叶组配方烟气化学成分的约束方程可表示为

．

(4) 叶组配方感官质量的约束方程

设叶组配方的光泽、香气、谐调性、刺激性、杂气、余味的下限和上限分别为和(=1,2,…,6)，则叶组配方感官质量的约束方程为