§3 最佳方案的模糊决策

例5 某露天煤矿有五个边坡设计方案，其各项参数根据分析计算结果得到边坡设计方案的参数如下表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项 目 | 方案 | 方案 | 方案 | 方案 | 方案 |
| 可采矿量(万吨) | 4700 | 6700 | 5900 | 8800 | 7600 |
| 基建投资(万元) | 5000 | 5500 | 5300 | 6800 | 6000 |
| 采矿成本(元/吨) | 4.0 | 6.1 | 5.5 | 7.0 | 6.8 |
| 不稳定费用(万元) | 30 | 50 | 40 | 200 | 160 |
| 净现值(万元) | 1500 | 700 | 1000 | 50 | 100 |

据勘探该矿探明储量8800吨，开采总投资不超过8000万元，试作出各方案的优劣排序，选出最佳方案。

解 首先确定隶属函数：  
 （1）可采矿量的隶属函数

因为勘探的地质储量为8800吨，故可用资源的利用函数作为隶属函数



（2）投资约束是8000万元，所以。

（3）根据专家意见，采矿成本元/吨为低成本，元/吨为高成本，故



（4）不稳定费用的隶属函数。

（5）净现值的隶属函数

取上限15（百万元），下限0.5（百万元），采用线性隶属函数



根据各隶属函数计算出5个方案所对应的不同隶属度，见下表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项 目 | 方案 | 方案 | 方案 | 方案 | 方案 |
| 可采矿量 | 0.5341 | 0.7614 | 0.6705 | 1 | 0.8636 |
| 基建投资 | 0.3750 | 0.3125 | 0.3375 | 0.15 | 0.25 |
| 采矿成本 | 1 | 0.76 | 1 | 0.4 | 0.48 |
| 不稳定费用 | 0.85 | 0.75 | 0.8 | 0 | 0.2 |
| 净现值 | 1 | 0.4480 | 0.6552 | 0 | 0.0345 |

这样就决定了模糊关系矩阵



根据专家评价，诸项目在决策中占的权重为，于是得诸方案的综合评价为



由此可知：方案最佳，次之，最差。

程序计算如下：

（1）首先编写函数文件myfun.m如下：

function f=myfun(x);

f(1,:)=x(1,:)/8800;

f(2,:)=1-x(2,:)/8000;

f(3,:)=0;

f(3,find(x(3,:)<=5.5))=1;

flag=find(x(3,:)>5.5 & x(3,:)<=8);

f(3,flag)=(8-x(3,flag))/2.5;

f(4,:)=1-x(4,:)/200;

f(5,:)=(x(5,:)-50)/1450;

（2）编写程序文件如下：

x=[4700 6700 5900 8800 7600

5000 5500 5300 6800 6000

4.0 6.1 5.5 7.0 6.8

30 50 40 200 160

1500 700 1000 50 100];

r=myfun(x);

a=[0.25,0.20,0.20,0.10,0.25];

b=a\*r

**3.1.2 相互排斥的约束条件**

**有两个相互排斥的约束条件**

 或 。

为了统一在一个问题中，引入变量，则上述约束条件可改写为：



其中是充分大的数。

约束条件

 或 

可改写为



如果有个互相排斥的约束条件：



为了保证这个约束条件只有一个起作用，我们引入个变量和一个充分大的常数，而下面这一组个约束条件

 （1）

 （2）

就合于上述的要求。这是因为，由于（2），个中只有一个能取0值，设，代入（1），就只有的约束条件起作用，而别的式子都是多余的。

**3.1.3 关于固定费用的问题（Fixed Cost Problem）**

在讨论线性规划时，有些问题是要求使成本为最小。那时总设固定成本为常数，并在线性规划的模型中不必明显列出。但有些固定费用（固定成本）的问题不能用一般线性规划来描述，但可改变为混合整数规划来解决，见下例。

例5 某工厂为了生产某种产品，有几种不同的生产方式可供选择，如选定的生产方式投资高（选购自动化程度高的设备），由于产量大，因而分配到每件产品的变动成本就降低；反之，如选定的生产方式投资低，将来分配到每件产品的变动成本可能增加。所以必须全面考虑。今设有三种方式可供选择，令

表示采用第种方式时的产量；

表示采用第种方式时每件产品的变动成本；

表示采用第种方式时的固定成本。

为了说明成本的特点，暂不考虑其它约束条件。采用各种生产方式的总成本分别为

 .

在构成目标函数时，为了统一在一个问题中讨论，现引入变量，令

 （3）

于是目标函数



（3）式这个规定可表为下述3个线性约束条件：

 （4）

其中是一个充分小的正常数，是个充分大的正常数。（4）式说明，当时必须为1；当时只有为0时才有意义，所以（4）式完全可以代替（3）式。

**§4 蒙特卡洛法（随机取样法）**

前面介绍的常用的整数规划求解方法，主要是针对线性整数规划而言，而对于非线性整数规划目前尚未有一种成熟而准确的求解方法，因为非线性规划本身的通用有效解法尚未找到，更何况是非线性整数规划。

然而，尽管整数规划由于限制变量为整数而增加了难度；然而又由于整数解是有限个，于是为枚举法提供了方便。当然，当自变量维数很大和取值范围很宽情况下，企图用显枚举法（即穷举法）计算出最优值是不现实的，但是**应用概率理论**可以证明，在一定的计算量的情况下，完全可以得出一个满意解。

例7 已知非线性整数规划为：





对该题，目前尚无有效方法求出准确解。如果用显枚举法试探，共需计算个点，其计算量非常之大。然而应用蒙特卡洛去随机计算个点，便可找到满意解，那么这种方法的可信度究竟怎样呢？

下面就分析随机取样采集个点计算时，应用概率理论来估计一下可信度。

不失一般性，假定一个整数规划的最优点不是孤立的奇点。

假设目标函数落在高值区的概率分别为0.01，0.00001，则当计算个点后，有任一个点能落在高值区的概率分别为

，

。

解 （）首先编写M文件mente.m定义目标函数f 和约束向量函数g，程序如下：

function [f,g]=mengte(x);

f=x(1)^2+x(2)^2+3\*x(3)^2+4\*x(4)^2+2\*x(5)-8\*x(1)-2\*x(2)-3\*x(3)...

-x(4)-2\*x(5);

g(1)=sum(x)-400;

g(2)=x(1)+2\*x(2)+2\*x(3)+x(4)+6\*x(5)-800;

g(3)=2\*x(1)+x(2)+6\*x(3)-200;

g(4)=x(3)+x(4)+5\*x(5)-200;

（ii）编写如下程序求问题的解：

rand('state',sum(clock));

p0=0;

tic

for i=1:10^5

x=99\*rand(5,1);

x1=floor(x);x2=ceil(x);

[f,g]=mengte(x1);

if sum(g<=0)==4

if p0<=f

x0=x1;p0=f;

end

end

[f,g]=mengte(x2);

if sum(g<=0)==4

if p0<=f

x0=x2;p0=f;

end

end

end

x0,p0

toc

§5 整数规划的计算机解法

整数规划问题的求解可以使用Lingo等专用软件。对于一般的整数规划问题，无法直接利用Matlab的函数，必须利用Matlab编程实现分枝定界解法和割平面解法。但对于指派问题等整数规划问题，可以直接利用Matlab的函数bintprog进行求解。

例8 求解下列指派问题，已知指派矩阵为



解：编写Matlab程序如下：

c=[3 8 2 10 3;8 7 2 9 7;6 4 2 7 5

8 4 2 3 5;9 10 6 9 10];

c=c(:);

a=zeros(10,25);

for i=1:5

a(i,(i-1)\*5+1:5\*i)=1;

a(5+i,i:5:25)=1;

end

b=ones(10,1);

[x,y]=bintprog(c,[],[],a,b);

x=reshape(x,[5,5]),y

求得最优指派方案为，最优值为21。

非线性规划:

对于非线性规划模型(NP)，可以采用迭代方法求它的最优解。迭代方法的基本思想是：从一个选定的初始点出发，按照某一特定的迭代规则产生一个点列，使得当是有穷点列时，其最后一个点是(NP)的最优解；当是无穷点列时，它有极限点，并且其极限点是(NP)的最优解。

设是某迭代方法的第轮迭代点，是第轮迭代点，记

 （1）

这里，并且的方向是从点向着点的方向。式（1）就是求解非线性规划模型(NP)的基本迭代格式。

通常，我们把基本迭代格式（1）中的称为第轮搜索方向，为沿方向的步长，使用迭代方法求解(NP)的关键在于，如何构造每一轮的搜索方向和确定适当的步长。

设，若存在，使

，

称向量是在点处的下降方向。

设，若存在，使

，

称向量是点处关于的可行方向。

一个向量，若既是函数在点处的下降方向，又是该点关于区域的可行方向，称之为函数在点处关于的可行下降方向。

现在，我们给出用基本迭代格式（1）求解(NP)的一般步骤如下：

0° 选取初始点，令。

1° 构造搜索方向，依照一定规划，构造在点处关于的可行下降方向作为搜索方向。

2° 寻求搜索步长。以为起点沿搜索方向寻求适当的步长，使目标函数值有某种意义的下降。

3° 求出下一个迭代点。按迭代格式（1）求出

。

若已满足某种终止条件，停止迭代。

4° 以代替，回到1°步。

考虑一维极小化问题

 （2）

若是区间上的下单峰函数，我们介绍通过不断地缩短的长度，来搜索得（2）的近似最优解的两个方法。

为了缩短区间，逐步搜索得（2）的最优解的近似值，我们可以采用以下途径：在中任取两个关于是对称的点和（不妨设，并把它们叫做搜索点），计算和并比较它们的大小。对于单峰函数，若，则必有，因而是缩短了的单峰区间；若，则有，故是缩短了的单峰区间；若，则和都是缩短了的单峰。因此通过两个搜索点处目标函数值大小的比较，总可以获得缩短了的单峰区间。对于新的单峰区间重复上述做法，显然又可获得更短的单峰区间。如此进行，在单峰区间缩短到充分小时，我们可以取最后的搜索点作为（2）最优解的近似值。

应该按照怎样的规则来选取探索点，使给定的单峰区间的长度能尽快地缩短？

* + 1. Fibonacci法

如用表示计算个函数值能缩短为单位长区间的最大原区间长度，可推出满足关系





数列称为Fibonacci数列，称为第个Fibonacci数，相邻两个Fibonacci数之比称为Fibonacci分数。

当用斐波那契法以个探索点来缩短某一区间时，区间长度的第一次缩短率为，其后各次分别为。由此，若和是单峰区间中第1个和第2个探索点的话，那么应有比例关系

， 

从而

， （3）

它们关于确是对称的点。

如果要求经过一系列探索点搜索之后，使最后的探索点和最优解之间的距离不超过精度，这就要求最后区间的长度不超过，即

 （4）

据此，我们应按照预先给定的精度，确定使（4）成立的最小整数作为搜索次数，直到进行到第个探索点时停止。

用上述不断缩短函数的单峰区间的办法，来求得问题（2）的近似解，是Kiefer(1953年)提出的，叫做Finbonacci法，具体步骤如下：

1° 选取初始数据，确定单峰区间，给出搜索精度，由（4）确定搜索次数。

2° ，计算最初两个搜索点，按（3）计算和。

3° while 



if 



else



end



end

4° 当进行至时，



这就无法借比较函数值和的大小确定最终区间，为此，取



其中为任意小的数。在和这两点中，以函数值较小者为近似极小点，相应的函数值为近似极小值。并得最终区间或。

**由上述分析可知，斐波那契法使用对称搜索的方法，逐步缩短所考察的区间，它能以尽量少的函数求值次数，达到预定的某一缩短率。**

2.1.2 0.618法

若，满足比例关系



称之为黄金分割数，其值为。

**黄金分割数**和Fibonacci分数之间有着重要的关系，它们是

1° ，为偶数，

，为奇数。

2° 。

现用不变的区间缩短率0.618，代替斐波那契法每次不同的缩短率，就得到了黄金分割法（0.618法）。这个方法可以看成是斐波那契法的近似，实现起来比较容易，效果也相当好，因而易于为人们所接受。

用0.618法求解，从第2个探索点开始每增加一个探索点作一轮迭代以后，原单峰区间要缩短0.618倍。计算个探索点的函数值可以把原区间连续缩短次，因为每次的缩短率均为，故最后的区间长度为



这就是说，当已知缩短的相对精度为时，可用下式计算探索点个数：



当然，也可以不预先计算探索点的数目，而在计算过程中逐次加以判断，看是否已满足了提出的精度要求。

0.618法是一种等速对称进行试探的方法，每次的探索点均取在区间长度的0.618倍和0.382倍处。

**2.3 无约束极值问题的解法**

无约束极值问题可表述为

 （5）

求解问题（5）的迭代法大体上分为两点：一是用到函数的一阶导数或二阶导数，称为解析法。另一是仅用到函数值，称为直接法。

**2.3.1 解析法**

**2.3.1.1 梯度法（最速下降法）**

对基本迭代格式

 （6）

我们总是考虑从点出发沿哪一个方向，使目标函数下降得最快。微积分的知识告诉我们，点的负梯度方向

，

是从点出发使下降最快的方向。为此，称负梯度方向为在点处的最速下降方向。

按基本迭代格式（6），每一轮从点出发沿最速下降方向作一维搜索，来建立求解无约束极值问题的方法，称之为最速下降法。

这个方法的特点是，每轮的搜索方向都是目标函数在当前点下降最快的方向。同时，用或作为停止条件。其具体步骤如下：

1°选取初始数据。选取初始点，给定终止误差，令。

2°求梯度向量。计算， 若，停止迭代，输出。否则，进行3°。

3° 构造负梯度方向。取

.

4° 进行一维搜索。求，使得



令转2°。

例4 用最速下降法求解无约束非线性规划问题



其中，要求选取初始点。

解：（）

编写M文件detaf.m如下

function [f,df]=detaf(x);

f=x(1)^2+25\*x(2)^2;

df(1)=2\*x(1);

df(2)=50\*x(2);

（ii）编写M文件zuisu.m

clc

x=[2;2];

[f0,g]=detaf(x);

while norm(g)>0.000001

p=-g'/norm(g);

t=1.0;f=detaf(x+t\*p);

while f>f0

t=t/2;f=detaf(x+t\*p);

end

x=x+t\*p

[f0,g]=detaf(x)

end

2.3.1.2 Newton法

考虑目标函数在点处的二次逼近式



假定Hesse阵



正定。

由于正定，函数的驻点是的极小点。为求此极小点，令

，

即可解得

.

对照基本迭代格式（1），可知从点出发沿搜索方向。



并取步长即可得的最小点。通常，把方向叫做从点出发的Newton方向。从一初始点开始，每一轮从当前迭代点出发，沿Newton方向并取步长为1的求解方法，称之为Newton法。其具体步骤如下：

1°**选取初始数据**。选取初始点，给定终止误差，令。

2°**求梯度向量。**计算，若，停止迭代，输出。否则，进行3°。

3°**构造Newton方向**。计算，取

.

4° 求下一迭代点。令，转2°。

例5 用Newton法求解，



选取。

解：（）



编写M文件nwfun.m如下：

function [f,df,d2f]=nwfun(x);

f=x(1)^4+25\*x(2)^4+x(1)^2\*x(2)^2;

df=[4\*x(1)^3+2\*x(1)\*x(2)^2;100\*x(2)^3+2\*x(1)^2\*x(2)];

d2f=[2\*x(1)^2+2\*x(2)^2,4\*x(1)\*x(2)

4\*x(1)\*x(2),300\*x(2)^2+4\*x(1)^2];

（ii）编写M文件：

clc

x=[2;2];

[f0,g1,g2]=nwfun(x)

while norm(g1)>0.00001

p=-inv(g2)\*g1

x=x+p

[f0,g1,g2]=nwfun(x)

end

如果目标函数是非二次函数，一般地说，用Newton法通过有限轮迭代并不能保证可求得其最优解。

为了提高计算精度，我们在迭代时可以采用变步长计算上述问题，程序如下：

clc

x=[2;2];

[f0,g1,g2]=nwfun(x)

while norm(g1)>0.00001

p=-inv(g2)\*g1,p=p/norm(p)

t=1.0,f=nwfun(x+t\*p)

while f>f0

t=t/2,f=nwfun(x+t\*p),

end

x=x+t\*p

[f0,g1,g2]=nwfun(x)

end

**2.4 Matlab求无约束极值问题**

在Matlab工具箱中，用于求解无约束极值问题的函数有fminunc和fminsearch，用法介绍如下。

求函数的极小值



其中可以为标量或向量。

Matlab中fminunc的基本命令是

[X,FVAL,EXITFLAG,OUTPUT,GRAD,HESSIAN]=FMINUNC(FUN,X0,OPTIONS,P1,P2, ...)

其中的返回值X是所求得的极小点，FVAL是函数的极小值，其它返回值的含义参见相关的帮助。FUN 是一个M文件，当FUN只有一个返回值时，它的返回值是函数；当FUN有两个返回值时，它的第二个返回值是的梯度向量；当FUN有三个返回值时，它的第三个返回值是的二阶导数阵（Hessian阵）。**X0是向量的初始值**，**OPTIONS是优化参数**，可以使用确省参数。P1，P2是可以传递给FUN的一些参数。

例6 求函数的最小值。

解：编写M文件fun2.m如下：

function [f,g]=fun2(x);

f=100\*(x(2)-x(1)^2)^2+(1-x(1))^2;

g=[-400\*x(1)\*(x(2)-x(1)^2)-2\*(1-x(1));200\*(x(2)-x(1)^2)];[f’(x2),f’(x1)]

在Matlab命令窗口输入

options = optimset('GradObj','on');

fminunc('fun2',rand(1,2),options)

即可求得函数的极小值。

在求极值时，也可以利用上二阶导数，编写M文件fun3.m如下：

function [f,df,d2f]=fun3(x);

f=100\*(x(2)-x(1)^2)^2+(1-x(1))^2;

df=[-400\*x(1)\*(x(2)-x(1)^2)-2\*(1-x(1));200\*(x(2)-x(1)^2)];

d2f=[-400\*x(2)+1200\*x(1)^2+2,-400\*x(1)

-400\*x(1),200];

在Matlab命令窗口输入

options = optimset('GradObj','on','Hessian','on');

fminunc('fun3',rand(1,2),options)

即可求得函数的极小值。

求多元函数的极值也可以使用Matlab的fminsearch命令，其使用格式为：

[X,FVAL,EXITFLAG,OUTPUT]＝FMINSEARCH(FUN,X0,OPTIONS,P1,P2,...)

例7 求函数取最小值时的值。

解 编写的M文件fun4.m如下：

function f=fun4(x);

f=sin(x)+3;

在命令窗口输入

x0=2;

[x,y]=fminsearch(@fun4,x0)

即求得在初值2附近的极小点及极小值。

* 1. **二次规划**

若某非线性规划的目标函数为自变量的二次函数，约束条件又全是线性的，就称这种规划为二次规划。

Matlab中二次规划的数学模型可表述如下：



这里是实对称矩阵，是列向量，是相应维数的矩阵。

Matlab中求解二次规划的命令是

[X,FVAL]= QUADPROG(H,f,A,b,Aeq,beq,LB,UB,X0,OPTIONS)

X的返回值是向量，FVAL的返回值是目标函数在X处的值。（具体细节可以参看在Matlab指令中运行help quadprog后的帮助）。

例8 求解二次规划



解 编写如下程序：

h=[4,-4;-4,8];

f=[-6;-3];

a=[1,1;4,1];

b=[3;9];

[x,value]=quadprog(h,f,a,b,[],[],zeros(2,1))

求得

。

1. 求解下面问题





解： 按问题的变量个数划分阶段，把它看作为一个三阶段决策问题。设状态变量为，并记；取问题中的变量为决策变量；各阶段指标函数按乘积方式结合。令最优值函数表示第阶段的初始状态为，从阶段到3阶段所得到的最大值。

设， ， 

则有

， ， 

用逆推解法，从后向前依次有

 及最优解 



由，得和（舍去）

又，而，故为极大值点。

所以 及最优解 。



同样**利用微分法**易知 ，最优解。

由于已知，因而按计算的顺序反推算，可得各阶段的最优决策和最优值。即

，

由



所以

，

由



所以

，

因此得到最优解为：，，；

最大值为：。