最优化篇

开篇有益

优化模型

实际问题中，人们经常遇到一类决策问题：在一系列客观或主观限制条件下，寻求使所关注的某个或多个指标达到最大（或最小）的决策。这种决策问题通常称为优化问题。解决这类问题的方法称为最优化方法，又称数学规划，它是运筹学里一个十分重要的分支。

最优化问题的数学模型的一般形式为：

$$opt \quad z = f(x) \qquad (1)$$

$$\begin{aligned} s.t. \quad & h_i(x) = 0, i = 1,\Lambda\ ,l \\ & g_j(x) \le 0, j = 1,\Lambda\ ,m \\ & t_k(x) \ge 0, k = 1,\Lambda\ ,n \\ & x \in D \subseteq R^s \end{aligned} \qquad (2)$$

opt（optimize）是最优化的意思，可以使求最小min（minimize）或求最大max（maximize），s.t.（subject to）是“受约束于”。

模型包含三个要素：决策变量decision bariable，目标函数objective function，约束条件constraints。

（2）所确定的x的范围称为可行域feasible region，满足（2）的解x称为可行解feasible solution，同时满足（1）（2）的解x^*称为最优解Optimal solution，整个可行域上的最优解称为全局最优解global optimal solution，可行域中某个领域上的最优解称为局部最优解local optimal solution。最优解所对应的目标函数值称为最优值optimum。

不同优化模型的求解方法以及求解难度有很大的不同，可按如下方法对模型进行分类：

（一）按有无约束条件（2）可分为：

1. 无约束优化unconstrained optimization。

这类问题蕴含了重要的寻优计算方法。

2. 约束优化constrained optimization。

大部分实际问题都是约束优化问题。

（二）按决策变量取值是否连续可分为：

1. 数学规划mathematical programming或连续优化continuous optmization。

可继续划分为线性规划(LP)Linear programming和非线性规划(NLP) Nonlinear programming。在非线性规划中有一种规划叫做二次规划(QP)Quadratic programming，二次规划问题的目标为二次函数，约束为线性函数。

2. 离散优化d-iscrete optimization或组合优化combinatorial optimization。

这类优化问题中包含一种常用的优化：整数规划(IP)Integer programming，整数规划中又包含很重要的一类规划：0-1（整数）规划 Zero-one programming，这类规划问题的决策变量只取 0 或者 1。

在求解组合优化问题中，出现了很多现代优化计算方法。

（三）按目标的多少可分为：
1.单目标规划。
2.多目标规划。
（四）按模型中参数和变量是否具有不确定性可分为：
1.确定性规划。
2.不确定性规划。
（五）按问题求解的特性可分为：
1.目标规划。
2.动态规划。
3.多层规划。
4.网络优化。
5.……等等。

求解软件

对优化问题的求解常用的是 LINGO 软件和 MATLAB 软件，本篇的程序编写基本都是用这两个软件完成的。

对于 LINGO 软件，线性优化求解程序通常使用单纯形法 simplex method，单纯形法虽然在实际应用中是最好最有效的方法，但对某些问题具有指数阶的复杂性，为了能解大规模问题，也提供了内点算法 interior point method 备选（LINGO 中一般称为障碍法，即 barrier)，非线性优化求解程序采用的是顺序线性规划法，也可用顺序二次规划法，广义既约梯度法，另外可以使用多初始点（LINGO 中称 multistart）找多个局部最优解增加找全局最优解的可能，还具有全局求解程序—分解原问题成一系列的凸规划。关于软件的使用方法可以参考 ppt 课件《LINGO 软件武功秘籍》以及实验书籍《数学软件与数学实验》。

对于 MATLAB 软件，有 MATLAB 优化工具箱，线性规划大型问题使用内点算法（也是默认算法），单纯形法和积极集法根据实际情况来解中小型问题。对于非线性规划问题，基本函数用信赖域等方法的结合来求解不同规模的问题。

本篇导读

第一章 无约束优化

寻优经典计算算法,matlab 实现

第二章 线性规划

完备的线性规划求解与应用

第三章 非线性规划

非线性模型的建立与求解

第四章 多目标规划

多目标决策的理论与方法

第五章　随机规划

随机规划的理论与方法

第六章 目标规划

目标规划的理论与方法

第七章 动态规划

动态规划的理论与方法

第八章 多层规划

？？

第九章 网络优化

图论方法及其他网络方法

第十章 组合优化算法

禁忌搜索算法，模拟退火算法，遗传算法，蚁群优化算法，人工神经网络

第二章 线性规划

理论印象

一般形式：

$$\min z(\text{或}\max z) = \sum_{j=1}^{n} c_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} a_{ij} x_j = (\text{或} \le \text{或} \ge) b_i, i = 1,\Lambda\ ,m$$

$$x_j \ge 0, j = 1,\Lambda\ ,n$$

决策变量为 x_j，其它都为常数。一般线性规划问题都可以通过引入非负的松弛变量 slack variable 与非负的剩余变量 surplus v-ariable 的方法化为标准形式（约束全是等约束）。

线性规划问题的可行域 feasible region 是一个凸集 convex set（任意两点的连线上的点都在区域内部，可以看作是没有凹坑的凸多面体），所以最优解 Optimal solution/point 在凸多面体的某个顶点上达到.

求解方法：单纯形算法 simplex method。

实质上就是在保证可行 feasible（用最小比值法则）的前提下，先在可行解 feasible solution/point 中取一个顶点，判断有没有达到最优，如果没有，就按照一定的规则（让正检验数 testing number 对应的变量入基 base），旋转 pivot 到比这个顶点更优的一个点，再判断有没有达到最优，这样迭代下去，直到达到最优 Optimal（或判断为不可行 infeasible，或判断为无界 unbounded）为止. 所以单纯形法实质上是一种具有某种规则的搜索算法. 而 LINGO 软件就是采用单纯形法对线性规划问题进行求解的.

2.1 连续性线性规划

如果实际问题具有如下性质，我们可以考虑建立线性规划模型：

1. 比例性：每个决策变量对目标函数以及右端项的贡献与该决策变量的取值成正比。

2. 可加性：每个决策变量对目标函数以及右端项的贡献与其他决策变量的取值无关。

3. 连续性：每个决策变量的取值都是连续的。

比例性与可加性保证了目标函数和约束条件对于决策变量的线性性，连续性则允许得到决策变量的实数最优解. 如果决策变量不连续，考虑建立离散的优化模型.

理论印象

敏感性分析与影子价格

讨论目标函数的系数的变化对最优解的影响以及约束右端项的变化对最优基的影响称为敏感性分析Sensitivity analysis。敏感性分析决定了当前最优生产计

划是否改变，影子价格是否还具有意义。

原问题对偶问题dual problem的解称为影子价格shadow price或对偶价格Dual Price。影子价格经济含义是：在资源得到最优配置，使总效益最大时，该资源投入量每增加一个单位所带来总收益的增加量. 影子价格只是一种静态的资源最优配置价格，是以线性规划为计算方法的计算价格，为资源的合理配置及有效利用提供了正确的价格信号和计量尺度.

在最优情况下我们可以把影子价格与资源的市场价格作比较，如果影子价格大于市场价格，考虑出售部分资源以获得更大利润；如果影子价格小于市场价格，则从市场购买资源以获得更大利润.

但是影子价格有意义是有范围的，如果约束的改变使得最优基（规划理论里用$c_B B^{-1}$）发生改变，当前的影子价格就不再具有意义.

例 1　阶段生产问题　某公司生产某产品，最大生产能力为 10000 单位, 每单位存储费 2 元，预定的销售量与单位成本如下：

表 8-2　每个月份的单位成本与销量

月份	单位成本	销售量
1	70	6000
2	71	7000
3	80	12000
4	76	6000

求一生产计划, 使

1) 满足需求；

2) 不超过生产能力；

3) 成本(生产成本与存储费之和)最低.

问题分析

这是一个多阶段生产计划问题，涉及多阶段存储，因为只需要制定这 4 个月的生产计划，所以不妨假定 1 月初无库存, 4 月底买完, 当月生产的不作为当月的库存, 也就是只有 1 月份到 2 月份，2 月份到 3 月份，3 月份到 4 月份的库存，库存量无限制.

模型建立 1

决策变量：

所谓生产计划，就是每月生产多少，差什么就设什么，所以设 x_i 为第 i 月产量。为了建立模型的方便，将其它已知的量设为字母，设 d_i 为销售量，e_i 为存储费，c_i 为单位成本.

目标：

生产成本：为 $\sum_{j=1}^{4} c_j x_j$ ，

第 j 月到 $j+1$ 月的库存量（记作第 $j+1$ 月的库存量）应该是 1 月到 j 月的总产量减去 1 月到 j 月的总销售量，即：$\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i$.

总库存费：$\sum_{j=1}^{3}(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i)e_{j+1}$.

总成本：$\sum_{j=1}^{4} c_j x_j + \sum_{j=1}^{3}(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i)e_{j+1}$ ，求其最小值.

约束：

满足需求：如果每个月末都会有非负的存储量，显然是满足需求的，因而可用约束 $\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i \geq 0$ ， $j=1,2,3$ ；

四个月的计划：所以四个月的总产量等于总需求量，$\sum_{i=1}^{4} x_i = \sum_{i=1}^{4} d_i$ ；

产量限制：$0 \leq x_i \leq 100$ ，因为产量比较大，不限制变量的取整数约束，看成一个连续变化的量.

综上所述，建立数学模型如下：

$$\min f = \sum_{j=1}^{4} c_j x_j + \sum_{j=1}^{3}(\sum_{i=1}^{j} x_i - \sum_{i=1}^{j} d_i)e_{j+1}$$

$$s.t.\begin{cases} \sum_{i=1}^{j} x_i \geq \sum_{i=1}^{j} d_i & j=1,2,3 \\ \sum_{i=1}^{4} x_i = \sum_{i=1}^{4} d_i & \\ 0 \leq x_i \leq 10000 & i=1,2,3,4 \end{cases}$$

程序编写

```
model:
title 生产计划程序1;
  Sets:
    yuefen/1..4/:c,x,e,d;
  endsets
  data:
    c=70 71 80 76;
    d=6000 7000 12000 6000;
    e=2 2 2 2 ;
```

```
   a=10000;
  enddata
   min=@sum(yuefen:c*x)+
          @sum(yuefen(j)|j#lt#4:
               @sum(yuefen(i)|i#le#j:x-d)*e(j+1));
   @for(yuefen(j)|j#lt#4:
       @sum(yuefen(i)|i#le#j:x)> @sum(yuefen(i)|i#le#j:d));
   @sum(yuefen:x)= @sum(yuefen:d);
   @for(yuefen:x<a);
end
```

运行结果

Model Title: :生产计划程序1

Variable	Value	Reduced Cost
A	10000.00	0.000000
C(1)	70.00000	0.000000
C(2)	71.00000	0.000000
C(3)	80.00000	0.000000
C(4)	76.00000	0.000000
X(1)	10000.00	0.000000
X(2)	10000.00	0.000000
X(3)	5000.000	0.000000
X(4)	6000.000	0.000000
E(1)	2.000000	0.000000
E(2)	2.000000	0.000000
E(3)	2.000000	0.000000
E(4)	2.000000	0.000000
D(1)	6000.000	0.000000
D(2)	7000.000	0.000000
D(3)	12000.00	0.000000
D(4)	6000.000	0.000000

由阴影部分X的值可知，1月份生产10000件，2月份生产10000件，3月份生产5000件，4月份生产6000件.

模型改进

上面这个模型中用到了存储量，而这个量都是用生产量与销售量来表示后嵌套在表达式中，由此我们想，如果引入库存量这个变量，再利用库存平衡方程，模型将更为流畅简洁.

模型建立 2

设 x_i 为第 i 月产量， d_i 为销售量， e_i 为存储费，c_i 为单位成本，设第 i 月初的库存

量为 s_i .

$$\min f = \sum_{i=1}^{4} c_i x_i + \sum_{i=1}^{4} e_i s_i$$

$$s.t. \quad s_{i+1} = s_i + x_i - d_{i,}$$

$$s_1 = 0$$
$$s_5 = 0$$

$$0 \le x_i \le 100\text{且为整数}，\quad i = 1,2,3,4$$
$$s_i \ge 0\text{且为整数} \qquad i = 1,2,3,4$$

程序编写

```
Model:
Title 生产计划程序2;
Sets:
yuefen/1..4/:c,x,e,d,s;
endsets
data:
c=70 71 80 76;
d=6000 7000 12000 6000;
e=2 2 2 2 ;
a=10000;
enddata
min=@sum(yuefen:c*x+e*s);
@for(yuefen(i)|i#lt#4:s(i+1)=s(i)+x(i)-d(i));
s(4)+x(4)-d(4)=0;
s(1)=0;
@for(yuefen:x<a);
End
```

运行结果

```
Model Title: :生产计划程序2

            Variable           Value        Reduced Cost

             …………(省略)
             X( 1)        10000.00            0.000000
             X( 2)        10000.00            0.000000
             X( 3)        5000.000            0.000000
```

```
X( 4)        6000.000           0.000000
…………(省略)
S( 1)        0.000000           0.000000
S( 2)        4000.000           0.000000
S( 3)        7000.000           0.000000
S( 4)        0.000000           6.000000
```

由阴影部分知，1月到2月没有库存，2月到3月库存4000件，3月到4月库存7000件。

经验介绍

通过这个模型知道，有时候适当的多引入一些变量也是必要的，可以将模型简单化，只要引入的变量数不至于大到严重的影响计算速度，多引入一些变量，帮助我们理解模型，分析结果都有好处.

考虑到这个问题有产有销，与运输问题十分类似，不妨转换为运输问题.

模型建立3

设 x_{ij} 表示第 i 月生产的产品在第 j 月卖出去的数量，c_{ij} 表示第 i 月生产的产品在第 j 月卖出去时的生产成本与存储成本之和，d_j 表示第 j 月的销售量. 从而可得下表：

表8-3　生产月生产的产品在需求月卖出时单位总成本

	需求月 1	需求月 2	需求月 3	需求月 4	产量
生产月 1	70	72	74	76	10000
生产月 2	-	71	73	75	10000
生产月 3	-	-	80	82	10000
生产月 4	-	-	-	76	10000
销量	6000	7000	12000	6000	

建立模型如下：

$$\min f = \sum_{j=1}^{4}\sum_{i=1}^{j} c_{ij}x_{ij}$$

$$s.t.\ \sum_{i=1}^{j} x_{ij} = d_j,\quad i,j=1,2,3,4$$

$$\sum_{j=i}^{4} x_{ij} \le 10000, i,j=1,2,3,4$$

$$x_{ij} \ge 0,\qquad i,j=1,2,3,4$$

程序编写

```
model:
title 生产计划程序3;
```

```
sets:
yuefen/1..4/:a,d,xx;
!定义上三角矩阵;
link(yuefen,yuefen)|&2#ge#&1:c,x;
endsets
data:
c=70 72 74 76
     71 73 75
        80 82
           76;
d=6000 7000 12000 6000;
a=10000 10000 10000 10000;
enddata
min=@sum(link:c*x);
@for(yuefen(i):@sum(yuefen(j)|j#ge#i:x(i,j))<a(i););
@for(yuefen(j):@sum(yuefen(i)|j#ge#i:x(i,j))>d(j););
!得到每个月的生产量;
@for(yuefen(i):xx=@sum(yuefen(j):x(i,j)));
End
```

例 2　生产决策问题　某工厂可以用 A、B 两种原料生产Ⅰ、Ⅱ、Ⅲ三种产品（每种产品都同时需要用两种原料），有关数据如下表：

表 8－6　　单位消耗与资源限制

	产品Ⅰ	产品Ⅱ	产品Ⅲ	现有原料（吨）
原料 A	2	1	1	7
原料 B	1	3	2	11
单位产品利润（万元）	2	3	1	

问:

1）求最优的生产计划。
2）若产品Ⅰ的单位利润上涨 1 万元，要不要改变生产计划?
3）若目前市场上原料 A 的实际价格为 0.5 万元/吨，工厂应如何决策?
4）若目前市场上原料 A 的实际价格为 0.8 万元/吨，工厂应如何决策?

模型建立

设 x_1, x_2, x_3 分别表示Ⅰ,Ⅱ,Ⅲ的生产量，问题的模型建立如下：

$$\max f = 2x_1 + 3x_2 + x_3$$

$$\begin{cases} 2x_1 + x_2 + 2x_3 \le 7 \\ x_1 + 3x_2 + 2x_3 \le 11 \\ x_1, x_2, x_3 \ge 0 \end{cases}$$

程序编写

```
model:
title 生产决策问题;
[maxf]max=2*x1+3*x2+x3;
[A]2*x1+x2+2*x3<7;
[B]x1+3*x2+2*x3<11;
END
```

运行结果

```
Model Title: 生产决策问题

                   Variable           Value        Reduced Cost
                         X1        2.000000            0.000000
                         X2        3.000000            0.000000
                         X3        0.000000            1.800000

                        Row    Slack or Surplus      Dual Price
                       MAXF        13.00000            1.000000
                          A        0.000000           0.6000000
                          B        0.000000           0.8000000
```

打开敏感性分析开关，即打开LINGO | Options | General Solver（通用求解程序）选项卡，在dual Computations（对偶计算）下选择Prices and Ranges保存设置，求解后再在程序窗口下点击LINGO | Range，调出敏感性分析报告：

```
Ranges in which the basis is unchanged:

                              Objective Coefficient Ranges
                           Current        Allowable        Allowable
             Variable   Coefficient        Increase         Decrease
                   X1      2.000000        4.000000         1.000000
                   X2      3.000000        3.000000         2.000000
                   X3      1.000000        1.800000         INFINITY

                                  Righthand Side Ranges
                  Row       Current        Allowable        Allowable
                                RHS         Increase         Decrease
                    A      7.000000         15.00000         3.333333
                    B      11.00000         10.00000         7.500000
```

对问题1，最优生产计划是生产2吨Ⅰ，3吨Ⅱ，最大利润13万元.

对问题2，由敏感性分析报告可看到，在不改变最优解的前提下，产品Ⅰ的

单位利润可以上涨4万元，所以不用改变生产计划，最大利润增加2万元。

对于问题3，如果目前市场上的原料A的实际价格是0.5<0.6(A的影子价格)，因此应该购进原料A，扩大生产能力，最大购进数量是15吨，利润将会增加（0.6-0.5）×15=1.5万元.

对于问题4，如果目前市场上的原料A的实际价格是0.8>0.6(A的影子价格)，因此应该出售部分原料将更为赚钱，最大出售量为3.33吨，利润将会增加（0.8-0.6）×3.33=0.66万元.

这里需要说明的是，上面说的最大购买量或者出售量是在保证当前的最优基不变的前提下的最大量，因为最优基不变才能保证当前的影子价格有意义，比如最大购进量15吨，究竟能不能再多购进一些呢？这个要具体问题具体分析，可能再购进会剩余，也可能超过15吨的量在一个新的最优基对应的新的影子价格下增长利润．可以自己动手更改约束的右端项运行程序观察.

例 3　基金使用计划（2001 年全国大学生数学建模竞赛 C 题）

某校基金会有一笔数额为 M 元的基金, 打算将其存入银行或购买国库券. 当前银行存款及各期国库券的利率见表 3-17. 假设国库券每年至少发行一次, 发行时间不定. 取款政策参考银行的现行政策.

校基金会计划在 n 年内每年用部分本息奖励优秀师生, 要求每年的奖金额大致相同, 且在 n 年末仍保留原基金数额. 校基金会希望获得最佳的基金使用计划, 以提高每年的奖金额. 请你帮助校基金会在如下情况下设计基金使用方案, 并对 $M = 5000$ 万元, $n = 10$ 年给出具体结果：

① 只存款不购国库券；

② 可存款也可购国库券；

③ 学校在基金到位后的第 3 年要举行百年校庆, 基金会希望这一年的奖金比其它年度多 20%.

表 3-17

	银行存款税后年利率（%）	国库券年利率（%）
活期	0.792	
半年期	1.664	
一年期	1.800	
二年期	1.944	2.55
三年期	2.160	2.89
五年期	2.304	3.14

我们来解决问题 2，详细过程略。

符号说明

M 表示基金数；

A 表示每年发放的奖金额；

x_{i0} 表示第 i 年用于活期存款的资金；

r_0 表示活期存款的税后年利率；

x_{i1} 表示第 i 年用于半年期存款的资金；

r_1 表示半年期存款的税后年利率；

x_{i2} 表示第 i 年用于一年期存款的资金；

r_2 表示一年期存款的税后年利率；

x_{i3} 表示第 i 年用于二年期存款的资金；

x_{i4} 表示第 i 年用于三年期存款的资金；

x_{i5} 表示第 i 年用于五年期存款的资金；

y_{i1} 表示第 i 年用于购买二年期国库券的资金；

y_{i2} 表示第 i 年用于购买三年期国库券的资金；

y_{i3} 表示第 i 年用于购买五年期国库券的资金；

r_3 表示二年期存款的税后年利率；

r_4 表示三年期存款的税后年利率；

r_5 表示五年期存款的税后年利率；

R_1 表示二年期国库券的年利率；

R_2 表示三年期国库券的年利率；

R_3 表示五年期国库券的年利率；

问题分析

根据基本原则，我们应优先考虑买国库券．（对于国库券每年发行时间都在年初的特殊情况，其求解模型类似模型 I，在这里我们不作讨论．）由于每年发行国库券的时间和发行的次数不定（每年至少发行一次），为了不使用于购买国库券的那部分资金闲置，我们设立如下的解决方案：

以二年期国库券为例：由于在年初投放资金时不能购买国库券，我们先将购买国库券的资金全部用于半年期存款，如果在该半年内发行了国库券，我们就将资金全部取出购买国库券，在国库券到期的那年将本息全部用于半年期存款，到期后转入活期存款；如果在该半年内没有发行国库券，我们将半年到期的自己全部用于活期存款，用于购买下半年一定会发行的国库券，国库券到期之后再全部转入活期存款．因此，我们将其运转周期定为三年，在这三年里，不管国库券什么时候发行，该部分资金一定有两年是用于存国库券，有半年用于存半年期，还有半年是存活期．即采用活期、半年期、国库券的“组合式”投资．同理，三年期国库券，五年期国库券的周期分别为四年，六年．那么，该部分资金在这几年里的收益为：

收益 = 本金 × (1 + 年数×国库券年利率)×(1 + 半年年利率÷2) × (1 + 活期年利率÷2)

记

第 i 年用于银行存款的钱：

$$X(i) = x_{i2} + x_{i3} + x_{i4} + x_{i5},\ i = 1, 2, \cdots, n.$$

第 i 年可用于投资的钱：

$$W(i) = (1 + r_2)\, x_{(i-1),2} + (1 + 2r_3)\, x_{(i-2),3} + (1 + 3r_4)\, x_{(i-3),4} + (1 + 5r_5)\, x_{(i-5),5} - A.$$

“组合式”购买的收益率：

$$p_1 = (1 + 2R_1)(1 + r_0/2)(1 + r_1/2) = 1.06394,$$
$$p_2 = (1 + 3R_2)(1 + r_0/2)(1 + r_1/2) = 1.10008,$$
$$p_3 = (1 + 5R_5)(1 + r_0/2)(1 + r_1/2) = 1.17125.$$

模型建立（错误，待修改）

$\max A,$

$$\begin{cases}
X(1) + y_{11} + y_{12} + y_{13} = M, \\
X(2) + y_{21} + y_{22} + y_{23} = (1 + r_2)\, x_{12} - A, \\
X(3) + y_{31} + y_{32} + y_{33} = (1 + r_2)\, x_{22} + (1 + 2r_3)\, x_{13} - A, \\
X(4) + y_{41} + y_{42} + y_{43} = (1 + r_2)\, x_{32} + (1 + 2r_3)\, x_{23} + (1 + 3r_4)\, x_{14} + p_1 y_{11} - A, \\
X(5) + y_{51} + y_{52} + y_{53} = (1 + r_2)\, x_{42} + (1 + 2r_3)\, x_{33} + (1 + 3r_4)\, x_{24} + p_1 y_{21} + p_2 y_{12} - A, \\
X(6) + y_{61} + y_{62} = W(6) + p_1 y_{31} + p_2 y_{22}, \\
x_{72} + x_{73} + x_{74} + y_{71} + y_{72} = W(7) + p_1 y_{41} + p_2 y_{32} + p_3 y_{13}, \\
x_{82} + x_{83} + x_{84} + y_{81} = W(8) + p_1 y_{51} + p_2 y_{42} + p_3 y_{23}, \\
x_{92} + x_{93} = W(9) + p_1 y_{61} + p_2 y_{52} + p_3 y_{33}, \\
x_{10,2} = W(10) + p_1 y_{71} + p_2 y_{62} + p_3 y_{43},
\end{cases}$$

$$W(11)+p_1 y_{81}+p_2 y_{72}+p_3 y_{53}=M.$$

$$x_{ij}\geqslant 0,\ y_{ij}\geqslant 0,\ A\geqslant 0.$$

模型求解

```
model:
title 基金使用计划;
sets:
year1/1..10/:S; !S代表题目中的X;
year2/1..11/:W;
year3/1..6/:r;
year4/1..3/:RR; !RR代表题目中的大R;
link(year1,year3):x;
link1(year1,year4):y;
endsets

data:
M=5000;
r=0.0792 0.01664 0.01800 0.01944 0.02610 0.02304;
RR=0.0255 0.0289 0.0314;
enddata

max=A;
@for(year1(i):S(i)=x(i,3)+x(i,4)+x(i,5)+x(i,6));
@for(year2(i)|i#ge#6#and#i#le#10:W(i)=(1+r(3))*x(i-1,3)+(1+2*r(4))*x(
i-2,4)+(1+3*r(5))*x(i-3,5)+(1+5*r(6))*x(i-5,6)-A);
(1+2*RR(1))*(1+r(1)/2)*(1+r(2)/2)=p1;
(1+3*RR(2))*(1+r(1)/2)*(1+r(2)/2)=p2;
(1+5*RR(3))*(1+r(1)/2)*(1+r(2)/2)=p3;
S(1)+@sum(year4(j):y(1,j))=M;
S(2)+@sum(year4(j):y(2,j))=(1+r(3))*x(1,3)-A;
S(3)+@sum(year4(j):y(3,j))=(1+r(3))*x(2,3)+(1+2*r(4))*x(1,4)-A;
S(4)+@sum(year4(j):y(4,j))=(1+r(3))*x(3,3)+(1+2*r(4))*x(2,4)+(1+3*R(5
))*x(1,5)+p1*y(1,1)-A;
S(5)+@sum(year4(j):y(5,j))=(1+r(3))*x(4,3)+(1+2*r(4))*x(3,4)+(1+3*R(5
))*x(2,5)+p1*y(2,1)+p2*y(1,2)-A;
S(6)+y(6,1)+y(6,2)=W(6)+p1*y(3,1)+p2*y(2,2);
x(7,3)+x(7,4)+x(7,5)+y(7,1)+y(7,2)=W(7)+p1*y(4,1)+p2*y(3,2)+p3*y(1,3)
;
x(8,3)+x(8,4)+x(8,5)+y(8,1)=W(8)+p1*y(5,1)+p2*y(4,2)+p3*y(2,3);
x(9,3)+x(9,4)=W(9)+p1*y(6,1)+p2*y(5,2)+p3*y(3,3);
x(10,3)=W(10)+p1*y(7,1)+p2*y(6,2)+p3*y(4,3);
W(11)+p1*y(8,1)+p2*y(7,2)+p3*y(5,3)=M;
end
```

解得：每年奖金额 A 为：127.521 万元．投资方案如表 3-19.

表 3-19（单位：万元）

i	1	2	3	4	5	6	7	8	9	10
x_{i2}	232.216	0	0	0	125.267	0	0	0	0	0
x_{i3}	227.549	0	0	0	0	0	0	0	0	—
x_{i4}	222.010	0	0	0	0	0	0	0	—	—
x_{i5}	0	0	0	0	0	0	—	—	—	—
y_{i1}	0	0	0	0	0	0	0	0	—	—
y_{i2}	229.789	0	0	0	0	0	4661.043	—	—	—
y_{i3}	4088.435	108.876	108.876	108.876	0	—	—	—	—	—

2.2 整数线性规划与 0-1 规划

理论印象

在线性规划问题中，如果所有的决策变量都取整数，则称为整数线性规划，如果决策变量只取 0 或 1 则称为 0-1 线性规划。

整数规划的求解方法是分枝定界法，其思想是“分而治之”的策略，即逐次对解空间进行划分。所谓分枝，就是不断去掉非整数最优解的区域对可行域进行划分，所谓定界，如果在某区域得到整数最优解，可以作为原问题最优解的下界（对极小化问题），其它区域上的最优解（无整数限制）如果比它大，就可以剪枝，从而避免完全枚举。

0-1 规划的求解方法有隐枚举法（穷举法），分枝定界法。

例 1　下料问题 1　现要做 100 套钢架,用长为 2.9m、2.1m 和 1.5m 的元钢各一根,已知原料长 7.4m,问如何下料,使用的原材料最省?

问题分析

之所以有这样的问题，是因为下料的方式有很多种，如何下料，就是每一种下料方式下了多少根钢材，显然这就是我们的决策变量.

合理的下料方式是剩余料头的长度不会超过最短原料需求（本题为1.5m），下料的所有的合理方式可用LINGO编写搜索算法全部搜索出来.

LINGO程序编写

```
model:
title 搜索合理的下料方式;
sets:
ren/1..5/:; !用一根原料可下各需求长度的最多根数定义元素个数，最多为4，这里要定义5;
long(ren,ren,ren):; !有三种需求长度，定义三维数组;
endsets
data:!搜索所有满足过滤条件的i,j,k;
@text('d:xialiao.txt')=@writefor(long(i,j,k)|
                !一种下料方式下料长度和不超过总长度;
                (7.4-2.9*(i-1)-2.1*(j-1)-1.5*(k-1))#ge#0
                    #and#
```

```
            !合理模式的余料小于最短需求;
            (7.4-2.9*(i-1)-2.1*(j-1)-1.5*(k-1))#lt#@smin(2.9,2.1,1.5)
      !输出下料方式到文本文件renxinglong.txt，我们需要的数是0--4;
      :i-1,4*' ',j-1,4*' ',k-1, 4*' ',7.4-2.9*(i-1)-2.1*(j-1)-1.5*(k-1)
@newline(1));
enddata

end
```

求解得到8种下料方式，程序自动在D盘建立renxinglong.txt 文件并写入结果，对每一种方式再简单计算一下余料的得到下面表格：

表 8－7　　所有下料方式

	I	II	III	IV	V	VI	VII	VIII
2.9m	2	1	1	1	0	0	0	0
2.1m	0	2	1	0	3	2	1	0
1.5m	1	0	1	3	0	2	3	4
剩余料头	0.1	0.3	0.9	0	1.1	0.2	0.8	1.4

模型建立

设 x_i 表示按第 i 种方式下料的根数，i=1…8，则问题的线性规划模型为：

$$\min f = 0.1x_1 + 0.3x_2 + 0.9x_3 + 0x_4 + 1.1x_5 + 0.2x_6 + 0.8x_7 + 1.4x_8$$

$$s.t.\begin{cases} 2x_1 + x_2 + x_3 + x_4 \ge 100 \\ 2x_2 + 3x_3 + 3x_5 + 2x_6 + x_7 \ge 100 \\ x_1 + x_3 + 3x_4 + 2x_6 + 3x_7 + 4x_8 \ge 100 \\ x_i \ge 0,\ j = 1,2,3,4,5,6,7,8;\ x_i\text{取整} \end{cases}$$

模型说明

关于目标函数的选取有两种，一种是剩余的余料最少，上面选取的就是这个目标，另一种就是所用的原料的根数最少（$\min f = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$），这两个目标是有区别的，余料最少时所用的原料不一定最少，因为有截出多余的满足需求的元刚并没有算作余料，可以根据实际情况选择这两种目标.

决策变量要限制取整数，是一个纯整数线性规划问题.

这种全方式设变量的模型只适合小型下料问题，大型下料问题或者对下料方式有限制的问题将不适合.

程序编写

```
MODEL:
TITLE 下料问题;
!定义集合段;
SETS:
HANG/1..3/:B;!定义矩阵的行;
LIE/1..8/:C,X;!定义矩阵的列以及变量;
XISHU(HANG,LIE):A;!定义约束的系数矩阵;
ENDSETS
DATA:
! 数据可以直接粘贴表格;
A=
```

2	1	1	1	0	0	0	0
0	2	1	0	3	2	1	0
1	0	1	3	0	2	3	4

```
;
C=
```

0.1	0.3	0.9	0	1.1	0.2	0.8	1.4

```
;!目标函数的系数;
B=100 100 100;!约束的右端项;
ENDDATA
[OBJ]MIN=@SUM(LIE:C*X);
@FOR(HANG(I):[YUESHU]
     @SUM(LIE(J):A(I,J)*X(J))>B(I));
@FOR(LIE:@GIN(X)); !限制变量取整数;
END
```

运行结果

```
Global optimal solution found.
  Objective value:                          10.00000
  Total solver iterations:                         7


  Model Title: 下料问题

                    Variable          Value        Reduced Cost
```

```
………………(省略)
  X( 1)       0.000000          0.1000000
  X( 2)       0.000000          0.1000000
  X( 3)       0.000000          0.8000000
  X( 4)       100.0000           0.000000
  X( 5)       0.000000          0.8000000
  X( 6)       50.00000           0.000000
  X( 7)       0.000000          0.7000000
  X( 8)       0.000000           1.400000
……………（省略）
```

所以最优方案是：用第4种方式下料100根，第6种下料方式下料50根．最少剩余料头10m．（最优解不唯一）

例2 露天矿生产的车辆安排（2003全国大学生数学建模竞赛B题）

露天矿里有若干个爆破生成的石料堆，每堆称为一个铲位，每个铲位已预先根据铁含量将石料分成矿石和岩石。一般来说，平均铁含量不低于 25%的为矿石，否则为岩石。每个铲位的矿石、岩石数量，以及矿石的平均铁含量（称为品位）都是已知的。每个铲位至多能安置一台电铲，电铲的平均装车时间为 5 分钟。

卸货地点（以下简称卸点）有卸矿石的矿石漏、2 个铁路倒装场（以下简称倒装场）和卸岩石的岩石漏、岩场等，每个卸点都有各自的产量要求。从保护国家资源的角度及矿山的经济效益考虑，应该尽量把矿石按矿石卸点需要的铁含量（假设要求都为 29.5%±1%，称为品位限制）搭配起来送到卸点，搭配的量在一个班次（8 小时）内满足品位限制即可。从长远看，卸点可以移动，但一个班次内不变。卡车的平均卸车时间为 3 分钟。

所用卡车载重量为 154 吨，平均时速 28 km/h 。卡车的耗油量很大，每个班次每台车消耗近 1 吨柴油。发动机点火时需要消耗相当多的电瓶能量，故一个班次中只在开始工作时点火一次。卡车在等待时所耗费的能量也是相当可观的，原则上在安排时不应发生卡车等待的情况。电铲和卸点都不能同时为两辆及两辆以上卡车服务。卡车每次都是满载运输。

每个铲位到每个卸点的道路都是专用的宽 60 m 的双向车道，不会出现堵车现象，每段道路的里程都是已知的。

一个班次的生产计划应该包含以下内容：出动几台电铲，分别在哪些铲位上；

出动几辆卡车，分别在哪些路线上各运输多少次。一个合格的计划要在卡车不等待条件下满足产量和质量（品位）要求，而一个好的计划还应该考虑下面两条原则之一：

1.总运量（吨公里）最小，同时出动最少的卡车，从而运输成本最小；

2.利用现有车辆运输，获得最大的产量（岩石产量优先；在产量相同的情况下，取总运量最小的解）。

请你就两条原则分别建立数学模型，并给出一个班次生产计划的快速算法。针对下面的实例，给出具体的生产计划、相应的总运量及岩石和矿石产量。

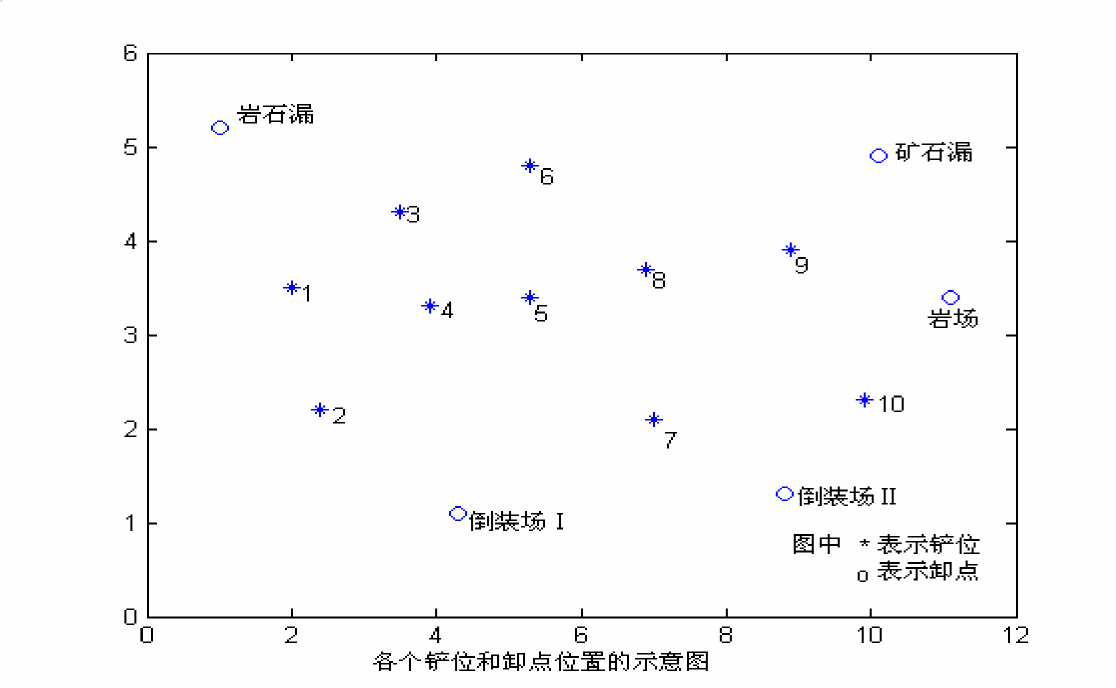
某露天矿有铲位 10 个，卸点 5 个，现有铲车 7 台，卡车 20 辆。各卸点一个班次的产量要求：矿石漏 1.2 万吨、倒装场Ⅰ1.3 万吨、倒装场Ⅱ1.3 万吨、岩石漏 1.9 万吨、岩场 1.3 万吨。

铲位和卸点位置的二维示意图如下，各铲位和各卸点之间的距离（公里）如下表：

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石漏	5.26	5.19	4.21	4.00	2.95	2.74	2.46	1.90	0.64	1.27
倒装场Ⅰ	1.90	0.99	1.90	1.13	1.27	2.25	1.48	2.04	3.09	3.51
岩场	5.89	5.61	5.61	4.56	3.51	3.65	2.46	2.46	1.06	0.57
岩石漏	0.64	1.76	1.27	1.83	2.74	2.60	4.21	3.72	5.05	6.10
倒装场Ⅱ	4.42	3.86	3.72	3.16	2.25	2.81	0.78	1.62	1.27	0.50

各铲位矿石、岩石数量(万吨)和矿石的平均铁含量如下表：

	铲位 1	铲位 2	铲位 3	铲位 4	铲位 5	铲位 6	铲位 7	铲位 8	铲位 9	铲位 10
矿石量	0. 95	1. 05	1. 00	1. 05	1. 10	1. 25	1. 05	1. 30	1. 35	1. 25
岩石量	1. 25	1. 10	1. 35	1. 05	1. 15	1. 35	1. 05	1. 15	1. 35	1. 25
铁含量	30%	28%	29%	32%	31%	33%	32%	31%	33%	31%



问题分析

本题看起来象运输问题，但与典型的运输问题明显有以下不同：

1. 这是运输矿石与岩石两种物资的问题；
2. 属于产量大于销量的不平衡运输问题；
3. 为了完成品位约束，矿石要搭配运输；
4. 产地、销地均有单位时间的流量限制；
5. 运输车辆只有一种，每次满载运输，154吨/车次；
6. 铲位数多于铲车数意味着要最优的选择不多于7个产地作为最后结果中的产地；
7. 最后求出各条路线上的派出车辆数及安排。

近似处理：

因而我们可以先求出产位、卸点每条线路上的运输量。然后求出各条路线上的派出车辆数及安排。

模型假设

1.卡车在一个班次中不应发生等待或熄火后再启动的情况；
2.在铲位或卸点处由两条路线以上造成的冲突问题面前，我们认为只要平均时间能完成任务，就认为不冲突。我们不排时地进行讨论；
3.空载与重载的速度都是28km/h，耗油相差很大；
4.卡车可提前退出系统；
5.铲车一个班次内不移动铲位。

符号说明

x_{ij} ： 从i铲位到j号卸点的石料运量 （车）；
c_{ij} ： 从i号铲位到j号卸点的距离(公里)；
T_{ij} :从i号铲位到号j卸点路线上运行一个周期平均时间(分)；
A_{ij} ： 从号铲位到号卸点最多能同时运行的卡车数(辆)；

B_{ij} ：从号铲位到号卸点路线上一辆车最多可运行的次数(次)；
p_i：i号铲位的矿石铁含量 p=(30,28,29,32,31,33,32,31,33,31) %;
q_j : j号卸点任务需求，q=(1.2,1.3,1.3,1.9,1.3)*10000(吨);
c_{ki} ：i号铲位的铁矿石储量(万吨);
c_{yi} ：i号铲位的岩石储量(万吨);
f_i :描述第i号铲位是否使用的0-1变量，取1为使用；0为关闭;

其中：

$$T_{ij}=\frac{i\text{到}j\text{距离}\times 2}{\text{平均速度}}+3+5,\ A_{ij}=\left\lfloor\frac{T_{ij}}{5}\right\rfloor,B_{ij}=\left\lfloor\frac{8\times 60-(A_{ij}-1)\times 5}{T_{ij}}\right\rfloor,$$

模型建立

建立整数规划程序如下：

$$\min\sum_{i=1}^{10}\sum_{j=1}^{5}x_{ij}\times c_{ij}$$

s.t. $x_{ij}\le A_{ij}\times B_{ij}, i=1,\Lambda,10, j=1,\Lambda,5$

$$\sum_{j=1}^{5}x_{ij}\le f_i\times 8\times 60/5, i=1,\Lambda,10$$

$$\sum_{i=1}^{10}x_{ij}\le 8\times 20, j=1,\Lambda,5$$

$$\left.\begin{array}{l}x_{i1}+x_{i2}+x_{i5}\le ck_i\times 10000/154\\ x_{i3}+x_{i4}\le cy_i\times 10000/154\end{array}\right., i=1,\Lambda,10$$

$$\sum_{i=1}^{10}x_{ij}\ge q_j/154, j=1,\Lambda,5$$

$$\left.\begin{array}{l}\sum_{i=1}^{10}x_{ij}\times(p_i-30.5)\le 0\\ \sum_{i=1}^{10}x_{ij}\times(p_i-28.5)\ge 0\end{array}\right\}, j=1,2,5$$

$$\sum_{i=1}^{10}f_i\le 7$$

x_{ij}为非负整数

f_i 为0-1整数

约束从上至下分别是：

（1）道路能力(卡车数)约束

（2）电铲能力约束

（3）卸点能力约束

（4）铲位储量约束
（5）产量任务约束
（6）铁含量约束
（7）电铲数量约束
（8）整数约束

程序编写

```
model:
title CUMCM-2003B-01;
sets:
cai / 1..10 /:crate,cnum,cy,ck,flag;
xie / 1 .. 5 /:xsubject,xnum;
link( xie,cai ):distance,lsubject,number,che,b;
endsets
data:
crate=30 28 29 32 31 33 32 31 33 31;
xsubject= 1.2 1.3 1.3 1.9 1.3 ;
distance= 5.26 5.19 4.21 4.00 2.95 2.74 2.46 1.90 0.64 1.27
        1.90 0.99 1.90 1.13 1.27 2.25 1.48 2.04 3.09 3.51
        5.89 5.61 5.61 4.56 3.51 3.65 2.46 2.46 1.06 0.57
        0.64 1.76 1.27 1.83 2.74 2.60 4.21 3.72 5.05 6.10
        4.42 3.86 3.72 3.16 2.25 2.81 0.78 1.62 1.27 0.50;
cy = 1.25 1.10 1.35 1.05 1.15 1.35 1.05 1.15 1.35 1.25;
ck = 0.95 1.05 1.00 1.05 1.10 1.25 1.05 1.30 1.35 1.25;
enddata
!目标函数;
min=@sum( cai (i):
       @sum ( xie (j):
        number (j,i)*154*distance (j,i)));
!卡车每一条路线上最多可以运行的次数;
@for (link (i,j):
b(i,j)=@floor((8*60-(@floor((distance(i,j)/28*60*2+3+5)/5)-1)*
5)/(distance(i,j)/28*60*2+3+5)));
!每一条路线上的最大总车次的计算;
@for( link (i,j):
lsubject(i,j)=(@floor((distance(i,j)/28*60*2+3+5)/5))*b(i,j));
!计算各个铲位的总产量;
@for (cai(j):
     cnum(j)=@sum(xie(i):number(i,j)));
!计算各个卸点的总产量;
@for (xie(i):
     xnum(i)=@sum(cai(j):number(i,j)));
!道路能力约束;
```

```
@for (link (i,j):
     number(i,j)<=lsubject(i,j));
!电铲能力约束;
@for (cai (j) :
     cnum(j) <= flag(j)*8*60/5 );
!电铲数量约束 ---- added by Xie Jinxing, 2003-09-07;
@sum(cai(j): flag(j) ) <=7;
!卸点能力约束;
@for (xie (i):
     xnum (i)<=8*20);
!铲位产量约束;
@for (cai (i):
number(1,i)+number(2,i)+number(5,i)<=ck(i)*10000/154);
@for (cai (i):     number(3,i)+number(4,i)<=cy(i)*10000/154);
!产量任务约束;
@for (xie (i):
     xnum (i)>= xsubject (i)*10000/154);
!铁含量约束;
@sum(cai (j):
    number(1,j)*(crate(j)-30.5) )<=0;
@sum(cai (j):
    number(2,j)*(crate(j)-30.5) )<=0;
@sum(cai (j):
    number(5,j)*(crate(j)-30.5) )<=0;
@sum(cai (j):
    number(1,j)*(crate(j)-28.5) )>=0;
@sum(cai (j):
    number(2,j)*(crate(j)-28.5) )>=0;
@sum(cai (j):
    number(5,j)*(crate(j)-28.5) )>=0;
!关于车辆的具体分配;
@for (link (i,j):
      che (i,j)=number (i,j)/b(i,j));
!各个路线所需卡车数简单加和;
hehe=@sum (link (i,j): che (i,j));
!整数约束;
@for (link (i,j): @gin(number (i,j)));
@for (cai (j): @bin(flag (j)));
!车辆能力约束;
hehe<=20;
ccnum=@sum(cai (j): cnum(j) );
end
```

	铲位1	铲位2	铲位3	铲位4	铲位5	铲位6	铲位7	铲位8	铲位9	铲位10

矿漏		13						54		11
倒Ⅰ		42		43						
岩场									70	15
岩漏	81		43							
倒Ⅱ		13	2							70

	铲位1	铲位2	铲位3	铲位4	铲位5	铲位6	铲位7	铲位8	铲位9	铲位10
矿石漏		0.867						1.862		0.314
倒场Ⅰ		1.077		1.162						
岩场									1.892	0.326
岩石漏	1.841		1.229							
倒场Ⅱ		0.684	0.1							1.489

	铲位1	铲位2	铲位3	铲位4	铲位5	铲位6	铲位7	铲位8	铲位9	铲位10
矿石漏								1 (29)		
倒场Ⅰ		1 (39)		1 (37)						
岩场									1 (37)	
岩石漏	1(44)		1 (35)							
倒场Ⅱ										1 (47)

因而在本思想下的最优结果就是：铲位1、2、3、4、8、9、10处各放置一台电铲。一共使用了13辆卡车；总运量为85628.62吨公里；岩石产量为32186吨；矿石产量为38192吨。

例3 选址问题 某公司计划在A，B，C三个区建立销售部，确定了7个位置$M_1 \cdots M_7$可供选择，并且规定：

（1）在A区，从M_1,M_2,M_3三个点中至多选两个；

（2）在B区，从M_4,M_5两个点中至少选一个；

（3）在C区，从M_6,M_7两个点中至少选一个；

已知：如果选择$M_1 \cdots M_7$的分别投资为200，300，350，250，350，200，400万元，预计每年可获利50，80，120，70，100，60，120万元，现在公司可用于投资的资金共有1200万元，问应如何建立销售部？

模型建立

设选择$M_1 \cdots M_7$的投资分别是b_i万元，每年获利c_i万元，总资金为b万元.

设0-1变量 $x_i (i=1 \cdots 7)$ 如下：

$$x_i = \begin{cases} 1, & \text{选择} M_i \text{投资} \\ 0, & \text{否则} \end{cases}$$

则可建立模型如下：

$$\max f = \sum_{i=1}^{7} c_i x_i$$

$$s.t. \begin{cases} \sum_{i=1}^{7} b_i x_i \le b \\ x_1 + x_2 + x_3 \le 2 \\ x_4 + x_5 \ge 1 \\ x_6 + x_7 \ge 1 \\ x_i = 0 \text{或} 1 \qquad i = 1, 2, \ldots, 7 \end{cases}$$

模型说明

这是一个0-1规划问题，当我们面对的对象只有两种结果时，通常引入0-1变量来考虑.

0-1变量和一个量相乘可以表示是否选择这个量，所以模型中 $\sum_{i=1}^{7} c_i x_i$ 并不是代表把7个量加起来，而是在约束条件下限定了取“1”的0-1变量所对应的项加起来，所以程序实际上是寻找满足约束的所有的0，1组合，经过比较判断来得到最优.

程序编写

```
model:
title 选址问题;
sets:
place/1..7/:c,b,x;
endsets
data:
c=50 80 120 70 100 60 120;
b=200 300 350 250 350 200 400;
enddata
max=@sum(place:c*x);
@sum(place:b*x)<1200;
@sum(place(i)|i#le#3:x(i))<2;
@sum(place(i)|i#ge#4#and#i#le#5:x(i))>1;
@sum(place(i)|i#eq#6#or#i#eq#7:x(i))>1;
@for(place:@bin(x));
End
```

运行结果

为了让结果只显示非零解，选择LINGO|Solution, 弹出下面面板：

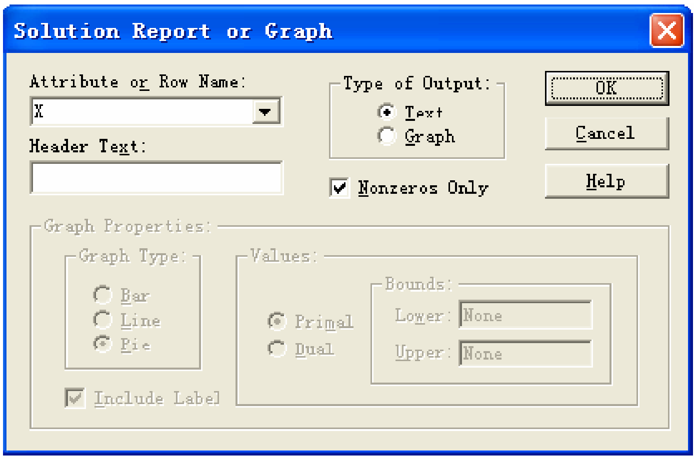


图 8 — 4　　改变约束的右端项

按上面面板设置后确认后，得到报告窗口如下：

```
Global optimal solution found.
 Objective value:                              370.0000
 Extended solver steps:                               0
 Total solver iterations:                             0


 Model Title: 选址问题

                       Variable           Value        Reduced Cost
                          X( 3)        1.000000           -120.0000
                          X( 4)        1.000000           -70.00000
                          X( 6)        1.000000           -60.00000
                          X( 7)        1.000000           -120.0000
```

所以在 M_3, M_4, M_6, M_7 建立销售部，每年盈利 370 万元.

例 4　DVD 在线租赁（2005 年全国大学生数学建模竞赛 B 题）

考虑如下的在线 DVD 租赁问题。顾客缴纳一定数量的月费成为会员，订购 DVD 租赁服务。会员对哪些 DVD 有兴趣，只要在线提交订单，网站就会通过快递的方式尽可能满足要求。会员提交的订单包括多张 DVD，这些 DVD 是基于其偏爱程度排序的。网站会根据手头现有的 DVD 数量和会员的订单进行分发。每个会员每个月租赁次数不得超过 2 次，每次获得 3 张 DVD。会员看完 3 张 DVD 之后，只需要将 DVD 放进网站提供的信封里寄回（邮费由网站承担），就可以继续下次租赁。请考虑以下问题：

1） 网站正准备购买一些新的 DVD，通过问卷调查 1000 个会员，得到了愿意观看这些 DVD 的人数（表 1 给出了其中 5 种 DVD 的数据）。此外，历史数据显示，60%的会员每月租

赁 DVD 两次，而另外的 40%只租一次。假设网站现有 10 万个会员，对表 1 中的每种 DVD 来说，应该至少准备多少张，才能保证希望看到该 DVD 的会员中至少 50%在一个月内能够看到该 DVD？如果要求保证在三个月内至少 95%的会员能够看到该 DVD 呢？

2） 表 2 中列出了网站手上 100 种 DVD 的现有张数和当前需要处理的 1000 位会员的在线订单（表 2 的数据格式示例如下表 2，具体数据请从 http://mcm.edu.cn/mcm05/problems2005c.asp 下载），如何对这些 DVD 进行分配，才能使会员获得最大的满意度？请具体列出前 30 位会员（即 C0001~C0030）分别获得哪些 DVD。

3） 继续考虑表 2，并假设表 2 中 DVD 的现有数量全部为 0。如果你是网站经营管理人员，你如何决定每种 DVD 的购买量，以及如何对这些 DVD 进行分配，才能使一个月内 95%的会员得到他想看的 DVD，并且满意度最大？

4） 如果你是网站经营管理人员，你觉得在 DVD 的需求预测、购买和分配中还有哪些重要问题值得研究？请明确提出你的问题，并尝试建立相应的数学模型。

表 1 对 1000 个会员调查的部分结果

DVD 名称	DVD1	DVD2	DVD3	DVD4	DVD5
愿意观看的人数	200	100	50	25	10

表 2 现有 DVD 张数和当前需要处理的会员的在线订单（表格格式示例）

DVD 编号		D001	D002	D003	D004	…
DVD 现有数量		10	40	15	20	…
会员在线订单	C0001	6	0	0	0	…
	C0002	0	0	0	0	…
	C0003	0	0	0	3	…
	C0004	0	0	0	0	…
	…	…	…	…	…	…

注：D001~D100 表示 100 种 DVD, C0001~C1000 表示 1000 个会员, 会员的在线订单用数字 1,2,…表示，数字越小表示会员的偏爱程度越高，数字 0 表示对应的 DVD 当前不在会员的在线订单中。

（注：表 2 数据位于文件 B2005Table2.xls 中，可从 http://mcm.edu.cn/mcm05/problems2005c.asp 下载）

考虑问题 2，

基本假设

每月租两次的会员所租 DVD 在将近半个月但不超过半个月时归还，每月租一次的会员在接近一个月但不超过一个月事归还。

问题的分析

观察 37 号 DVD 的库存量为 106，有愿望观看此 DVD 的人数只有 91 人，因而会剩余 15 张，总数 3007 张，减去 15 张，还有 2992 张，如果每人分 3 张，还欠缺 8 张，于是肯定有会员分不到自己想看的 3 张，于是分派方式出现多样化，不同的分派方式会员必然有不同的满意度。由于会员只是对 DVD 进行排序，所以有必要对满意度进行量化。

满意度的量化

不妨用比较简单的方式处理问题，设置下列三种满意度的计算方式：

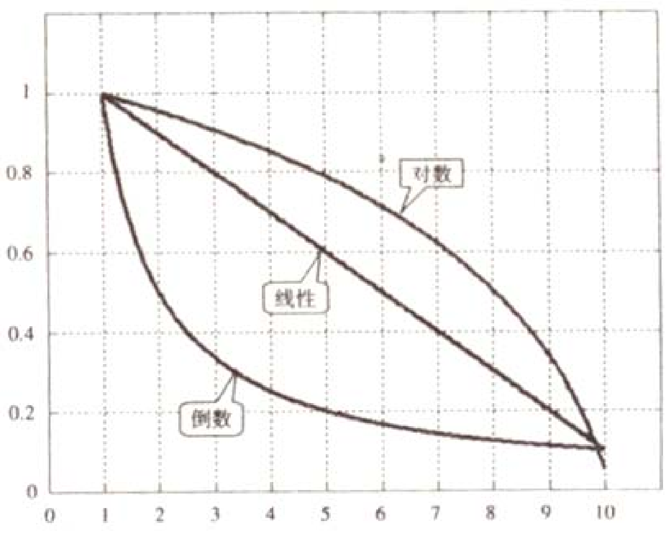
$f_1=\frac{11-x}{10}$，$f_2=\frac{1}{x}$，$f_3=0.413\ln(11-x)+0.05$，$x=1,2,\Lambda,10$

（其中 f_3 由隶属度函数 $f_3=a\ln(11-x)+b$，让 $f_3(1)=1, f_3(10)=0.05$ 得到）

由 f_3 计算满意度并归一化为：

DVD 排序	1	2	3	4	5	6	7	8	9	10
满意度	1	0.9565	0.9079	0.8528	0.7892	0.714	0.622	0.5033	0.336	0.05
归一化	0.1486	0.1421	0.1349	0.1267	0.1173	0.1061	0.09240	0.0748	0.0499	0.0074

比较满意度的优劣



从图像可看出，f_3 表示的满意度从 1 到 5 下降缓慢，从 6 到 10 下降比较快，比较符合实际情况，因而用 f_3 相对合理一些。

模型建立

假设会员至少分得两张 DVD。

c_{ij}：第 i 会员得到第 j 种 DVD 时的满意度

$x_{ij}=1$：给第 i 会员分配第 j 种 DVD，$x_{ij}=0$：不给第 i 会员分配第 j 种 DVD

b_j：第 j 种 DVD 的数量

建立 0-1 规划模型如下：

$$\max\ z=\sum_{i=1}^{1000}\sum_{j=1}^{100}c_{ij}x_{ij}$$

$$s.t.\begin{cases}2\le\sum_{j=1}^{100}x_{ij}\le 3, i=1,2,\Lambda\ ,1000\\ \sum_{i=1}^{1000}x_{ij}\le b_j, j=1,2,\Lambda\ ,100\\ x_{ij}=0\ or\ 1\end{cases}$$

程序编写：

```
MODEL:
SETS:
HY/1..1000/:A;
DVD/1..100/:D;
FP(HY,DVD):C,X;
ENDSETS
DATA:
D=@OLE('SHUJU.XLS','KUCUN');
C=@OLE('SHUJU.XLS','KUCUN');
@OLE(('SHUJU.XLS','FENPEI')=X;
ENDDATA
MAX=@SUM(FP:C*X);
@FOR(FP:@BIN(X));
@FOR(HY:A(I)=@SUM(DVD(J):X(I,J)));
@FOR(HY:@BND(2,A,3));
@FOR(DVD(J):@SUM(HY:X(I,J))<D(J));
N=@SUM(HY:A);
END
```

提示：先建立 SHUJU.XLS 才能运行程序，此处省略。

如果采用要么分 3 张要么不分的方法，将上述约束的第一行修改如下即可：

$$\sum_{j=1}^{100}x_{ij}=3y_i\ \ ,y_i=0\ or\ 1,\qquad i=1,2,\Lambda\ ,1000\ 。$$

例5 面试顺序问题 有4名同学到一家公司参加三个阶段的面试，公司要求每个同学都必须首先找公司秘书初试，然后到主管部门处复试，最后到经理处参

加免试，并且不允许插队，由于4名同学的专业背景不同，所以每人在三个阶段的面试时间也不同，如表所示，这4名同学约定他们全部面试完以后一起离开公司，假定现在时间是早上8：00，请问他们最早何时能离开公司？

表 8－7　面试时间要求

	秘书初试	主管复试	经理面试
同学甲	13	15	20
同学乙	10	20	18
同学丙	20	16	10
同学丁	8	10	15

模型分析与建立

记t_{ij}为第i名同学参加第j阶段面试需要的时间（已知），令x_{ij}表示第i名同学参加第j阶段面试的开始时刻（不妨记早上8：00面试开始为0时刻）（$i=1,2,3,4; j=1,2,3$），T为完成全部面试所花费的最少时间.

优化目标为：$\min T=\left\{\max_i\left\{x_{i3}+t_{i3}\right\}\right\}$

约束条件：

个人时间先后次序约束：　$x_{ij}+t_{ij}\le x_{i.j+1}$，　$i=1,2,3,4; j=1,2.$

同阶段不同同学时间不相容：（同阶段靠前同学的完成时间小于靠后同学的开始时间）

$x_{ij}+t_{ij}-x_{kj}\le My_{ik}$,

$x_{kj}+t_{kj}-x_{ij}\le M\left(1-y_{ik}\right)$,　M取大于T的常数，

$y_{ik}=0\ or\ 1$（取1表示k排在i的前面）

$i,k=1,2,3,4$

$j=1,2,3$

不妨给定i,k的大小关系：$i<k$

可将目标改为如下线性优化目标：

$$\begin{aligned}
&\min T\\
s.t.\quad &T\ge x_{13}+t_{13},\\
&T\ge x_{23}+t_{23},\\
&T\ge x_{33}+t_{33},\\
&T\ge x_{43}+t_{43}.
\end{aligned}$$

程序编写

```
model:
title:面试问题;
sets:
student/1..4/:;
office/1..3/:;
link1(student,office):x,t;
link2(student,student)|&1#lt#&2:y;
endsets
data:
t=13 15 20
  10 20 18
  20 16 10
   8 10 15 ;
enddata
min=time;
!time大于每名同学最后面试完毕时间;
@for(student(i):time>x(i,3)+t(i,3););
!面试先后次序约束;
@for(student(i):
    @for(office(j)|j#lt#3:x(i,j)+t(i,j)<x(i,j+1)););
!每个阶段只能面试一个同学,y(i,k)=1表示第k名同学排在第i名同学前面;取M=1000;
@for(student(i):
    @for(office(j):
        @for(student(k)|k#gt#i:
            x(i,j)+t(i,j)-x(k,j)<1000*y(i,k))));
@for(student(i):
    @for(office(j):
        @for(student(k)|k#gt#i:
            x(k,j)+t(k,j)-x(i,j)<1000*(1-y(i,k)))));
!定义0-1变量,最后通过0-1变量可以查看面试顺序;
@for(link2:@bin(y));
End
```

运行结果

```
Model Title: :面试问题

                Variable           Value        Reduced Cost
                    TIME        84.00000            0.000000
                 ……………………（省略）
               Y( 1, 2)         0.000000           -1000.000
               Y( 1, 3)         0.000000            0.000000
               Y( 1, 4)         1.000000            1000.000
```

```
Y( 2, 3)      0.000000        -1000.000
Y( 2, 4)      1.000000         0.000000
Y( 3, 4)      1.000000         0.000000
```

所以面试完成至少需要 84min，面试顺序为 4-1-2-3（丁-甲-乙-丙）.

2.3 一般线性规划与 0-1 规划的 matlab 求解

一、一般线性规划问题的 matlab 解

假设一般的线性规划问题的数学模型是

$$
\begin{aligned}
&\min_x f^T x\\
&\text{subject to}\quad Ax \le b\\
&\qquad\qquad\qquad Aeqx = beq\\
&\qquad\qquad\qquad lb \le x \le ub
\end{aligned}
$$

基本用法

函数调用格式 1：x=linprog(f,A,b)

功能：求具备第一个约束的模型。

函数调用格式 2：x=linprog(f,A,b,Aeq,beq)

功能：求具备前两个约束的模型。

函数调用格式 3：x=linprog(f,A,b,Aeq,beq,lb,ub)

功能：求具备三个约束的模型。

函数调用格式 4：x=linprog(f,A,b, Aeq,beq,lb,ub,x0)

功能：求具备三个约束的模型，设定初始值为 x0。

函数调用格式 5：x=linprog(f,A,b, Aeq,beq,lb,ub,x0,options)

功能：求解上述模型，并按 options 指定的内部优化参数进行优化。（比如指定精度，算法，步长，迭代次数等，可以用 options=optimset（）创建或编辑优化选项参数结构，有 30 多个，完全掌握需要较深的优化专业知识。）

函数调用格式 6：[x,fval]=linprog(f,A,b)

功能：求解并返回目标函数值。

函数调用格式 7：[x,fval,exitflag]=linprog(f,A,b)

功能：返回 exitflag 的值，描述函数计算退出的条件。比如可据此判断是否得到最优解。exitflag=0 表示优化结果已经超过了函数的估计值或者已声明的最大迭代次数；exitflag>0 表示优化过程中变量收敛于解 x，exitflag<0 表示计算不收敛。

函数调用格式 8：[x,fval,exitflag,output]=linprog(f,A,b)

功能:返回包含优化信息的输出变量 output。output 有 3 个分量，iterations 表示优化过程的迭代次数，cgiterations 表示 PCG 迭代次数（只在大规模算法有用），algorithm 表示优化所采用的算法。

函数调用格式 9：[x,fval, exitflag,output,lambda]=linprog(f,A,b)

功能：将解 x 处的拉格朗日乘子（线性规划可看做影子价格）返回到参数 lambda 中。Lambda 有四个量 Lambda.ineqlin, Lambda.eqlin, Lambda.upper, Lambda.lower，分别对应于四种约束。但是不能象 LINGO 那样得到其有意义的范围。

算法选择：

默认为内点算法。

Opt1=optimset（'largescale'，'off';）； %二次规划的有效集方法

Opt2=optimset（opt1，'simplex'，'on'）；   %单纯形算法

初始点只在有效集方法中有效。

提示：模型的目标函数是最小化问题，最大化需要转化，求解之前先进行初始数据矩阵的输入，如果某个约束不存在，比如第一个约束不存在，可令 A=[],b=[]，或者直接在函数中输入空矩阵。

例 1 比如 2.1 节例 2 的模型：

$$\max z = 2x_1 + 3x_2 + x_3$$

Subject to

$$\begin{cases} 2x_1 + x_2 + 2x_3 \le 7 \\ x_1 + 3x_2 + 2x_3 \le 11 \\ x_1, x_2, x_3 \ge 0 \end{cases}$$

求解输入如下：

```
>> f=[-2,-3,-1];
  A=[2 1 2;1 3 2];
  b=[7 11];
lb=zeros(3,1);
>> [x,fval, exitflag,output,lambda]=linprog(f,A,b,[],[],lb)
Optimization terminated.

x =

    2.0000
    3.0000
    0.0000



fval =

  -13.0000



exitflag =

     1
```

```
output =

      iterations: 7
       algorithm: 'large-scale: interior point'
    cgiterations: 0
         message: 'Optimization terminated.'


lambda =

    ineqlin: [2x1 double]
      eqlin: [0x1 double]
      upper: [3x1 double]
      lower: [3x1 double]
```

二、**0-1 规划的 matlab 求解**

在 MATLAB 自带的优化工具箱中，并没有求解整数规划的函数，但有求解 0-1 规划的函数。使用的算法是分枝定界法。

设 0-1 线性规划问题的数学模型是

$$\begin{aligned} &\min_x f^T x \\ &\text{subject to} \quad Ax \le b \\ &\qquad\qquad\quad Aeqx = beq \\ &\qquad\qquad\quad x_i = 0\, or\, 1 \end{aligned}$$

基本用法

函数调用格式 1：x=bintprog(f)
功能：求只有 0-1 约束的模型。
函数调用格式 2：x= bintprog (f,A,b)
函数调用格式 3：x= bintprog (f,A,b,Aeq,beq,lb,ub)
函数调用格式 4：x= bintprog (f,A,b, Aeq,beq,lb,ub,x0)
必须设定初始值为 x0 为 0 或 1。
函数调用格式 5：x= bintprog (f,A,b, Aeq,beq,lb,ub,x0,options)
函数调用格式 6：[x,fval]= bintprog (f,A,b)
函数调用格式 7：[x,fval,exitflag]= bintprog (f,A,b)
函数调用格式 8：[x,fval,exitflag,output]= bintprog (f,A,b)
功能解释与实例参考并仿照前面即可。

第三章　非线性规划

理论印象

将非线性规划模型可以更具体的表示为如下形式：

$$\min z = f(x)$$

$$\begin{aligned} s.t. \quad & h_i(x) = 0, i = 1,\Lambda\ ,l \\ & g_j(x) \le 0, j = 1,\Lambda\ ,m \\ & x \in R^n \end{aligned}$$

如果只有等约束 h_i，则可以用拉格朗日乘数法构造拉格朗日函数：

$L(x,\mu) = f(x) + \sum_{i=1}^{m} \mu_i h_i(x)$（$\mu_i$ 为参数），然后求解非线性方程组

$\begin{cases} \dfrac{\partial L}{\partial x_i} = 0 \\ \dfrac{\partial L}{\partial \mu_i} = 0 \end{cases}$ 即可。

对上述模型，通过讨论 x 的可行方向与下降方向，可以得到如下的 KKT 条件（Karush-Kuhn-Tucker 条件）（具体可微等条件略）：局部最优解 x 满足

$\begin{cases} \nabla f(x) + \sum_{i=1}^{m} \mu_i \nabla h_i(x) + \sum_{j=1}^{l} \lambda_i \nabla g_j(x) = 0 \\ \lambda_j g_j(x) = 0 \end{cases}$，$\mu_i, \lambda_i \ge 0$。

对等约束模型构造拉格朗日函数：

$$L(x,\mu,\lambda) = f(x) + \sum_{i=1}^{m} \mu_i h_i(x) + \sum_{j=1}^{l} \lambda_j g_j(x)$$

KKT 条件中的公式刚好就是函数 L 对 x 的导数（梯度）等于 0.(μ,λ)通常称为拉格朗日乘子。

3.1 二次规划

理论印象

目标函数为二次函数，约束为线性函数的优化问题称为二次规划。
二次规划模型的一般形式：

$$\begin{aligned} \min\ & f(x) = \frac{1}{2} x^T H x + c^T x \\ s.t.\ & Ax \le b \end{aligned}$$

$H \in R^{n\times n}$ 为对称矩阵。特别的，当 H 正定时，模型称为凸二次规划。凸二次规划局部最优解（KKT 点）就是全局最优解。

对等约束的凸二次规划，构造拉格朗日函数：

$$L(x,\lambda)=\frac{1}{2}x^T Hx+c^T x+\lambda^T(Ax-b)$$

求导得如下方程组：

$$\begin{cases} Hx+c+A^T\lambda=0 \\ Ax-b=0 \end{cases}$$

解方程组即得最优解。

有效集方法：对于存在不等式约束的二次规划，将等约束（对应有效集）与不等约束分开，将非有效约束去掉，通过解一系列等式约束的二次规划来实现不等式约束的优化。

例 1　投资组合问题　某三支股票在 12 年的价格如下：

年份	股票A	股票B	股票C	股票指数
1943	1.300	1.225	1.149	1.258997
1944	1.103	1.290	1.260	1.197526
1945	1.216	1.216	1.419	1.364361
1946	0.954	0.728	0.922	0.919287
1947	0.929	1.144	1.169	1.057080
1948	1.056	1.107	0.965	1.055012
1949	1.038	1.321	1.133	1.187925
1950	1.089	1.305	1.732	1.317130
1951	1.090	1.195	1.021	1.240164
1952	1.083	1.390	1.131	1.183675
1953	1.035	0.928	1.006	0.990108
1954	1.176	1.715	1.908	1.526236

解决如下问题：

（1）如果在 1955 年你有一笔资金投资这三种股票，并期望年收益率至少达到 15%，那么你应当如何投资？分析投资组合与回报率以及风险的关系。

（2）如果还可以投资国库券，年收益率为 5%，如果投资呢？

（3）如果幂目前持有的股票比例为：A 占 50%，B 占 35%，C 占 15%，买卖股票按交易额的 1%收取交易费，你会怎么办？

对问题 1

问题分析 1

投资股票的收益是不确定的，因而是一个随机变量，我们可以用期望值来表示。风险可以用方差来衡量，方差越大，风险越大。期望与协方差可由上表求得。

符号说明

R_1, R_2, R_3：A,B,C 三种股票的收益率,是随机变量。

x_1, x_2, x_3： A,B,C 三种股票的投资比例。

模型建立 1

目标： $\min D(x_1 R_1 + x_2 R_2 + x_3 R_3)$，即 $\min \sum_{j=1}^{3} \sum_{i=1}^{3} x_i x_j \operatorname{cov}(R_i, R_j)$

约束： $x_1 ER_1 + x_2 ER_2 + x_3 ER_3 \geq 0.15$

$$x_1 + x_2 + x_3 = 1,$$
$$x_1, x_2, x_3 \geq 0$$

这是一个二次规划问题。

LINGO 程序编写

```
MODEL:
Title 投资组合模型;
SETS:
   YEAR/1..12/;
   STOCKS/ A,  B,  C/: Mean,X;
   link(YEAR, STOCKS): R;
   STST(Stocks,stocks): COV;
ENDSETS
DATA:
   TARGET = 1.15;
! R是原始数据;
   R =
    1.300    1.225    1.149
    1.103    1.290    1.260
    1.216    1.216    1.419
    0.954    0.728    0.922
    0.929    1.144    1.169
    1.056    1.107    0.965
    1.038    1.321    1.133
    1.089    1.305    1.732
    1.090    1.195    1.021
    1.083    1.390    1.131
    1.035    0.928    1.006
    1.176    1.715    1.908;
ENDDATA
CALC:  !计算均值向量Mean与协方差矩阵COV;
@for(stocks(i): Mean(i) =
    @sum(year(j): R(j,i)) / @size(year) );
@for(stst(i,j): COV(i,j) = @sum(year(k):
     (R(k,i)-mean(i))*(R(k,j)-mean(j))) / (@size(year)-1) );
```

```
ENDCALC
[OBJ] MIN = @sum(STST(i,j): COV(i,j)*x(i)*x(j));
[ONE] @SUM(STOCKS: X) = 1;
[TWO] @SUM(stocks: mean*x) >= TARGET;
END
```

计算结果：

A 占 53%，B 占 36%，C 占 11% ，风险（方差）为 0.0224138。即标准差为 0.1497123.

在数据段，修改赋值语句：`TARGET =?;`可以输入不同的回报率来观察投资组合以及相应的风险。略。

问题分析 2

在实际股票市场上，由于股票种类庞大，对协方差的计算几乎是不可能进行的，因为有必要避开协方差的计算再来处理这个问题，我们可以利用股价指数。由于股票指数是整个市场的大势信息，对具体每只股票的涨跌有显著影响的，我们可简单假设影响是线性关系。从而可以通过线性回归的方法找这个线性关系。

预处理—线性回归

三支股票的收益 $R_i(i=1,2,3)$ 与股票指数 M 的线性关系为：

$$R_i = u_i + b_i M + e_i$$

用最小二乘法求回归系数，LINGO 程序如下：

```
MODEL:
Title 线性回归模型;
SETS:
  YEAR/1..12/:M,MM;
  STOCKS/A,  B,  C/: u, b, s2, s;
  link(YEAR, STOCKS): R, RR, e;
ENDSETS
DATA:
! RR和MM是原始数据;
RR=
```

1.300	1.225	1.149
1.103	1.290	1.260
1.216	1.216	1.419
0.954	0.728	0.922
0.929	1.144	1.169
1.056	1.107	0.965
1.038	1.321	1.133
1.089	1.305	1.732
1.090	1.195	1.021

1.083	1.390	1.131
1.035	0.928	1.006
1.176	1.715	1.908

```
;
MM =
```

1.258997
1.197526
1.364361
0.919287
1.057080
1.055012
1.187925
1.317130
1.240164
1.183675
0.990108
1.526236

```
;
num=?;
ENDDATA
CALC:
@for(year:m=mm-1);
@for(link:r=rr-1);
mean0=@sum(year: (M)/@size(year));!M的均值;
s20=@sum(year: (M-mean0)^2) / (@size(year)-1);!M的方差，标准差;
s0=@sqrt(s20);!s0=(s20)^(1/2);
ENDCALC
[OBJ] MIN = @sum(stocks(i)|i#eq#num: s2(i));
@for(link(k,i)|i#eq#num: [ERROR] e(k,i) = R(k,i)-u(i)-b(i)*M(k));
@for(stocks(i)|i#eq#num:
![VAR] s2(i)=(@sum(year(k): @sqr(e(k,i))) / (@size(year)-2)) ;
[VAR] s2(i)=(@sum(year(k): (e(k,i))^2) / (@size(year)-2)) ;
![STD] s(i)=@sqrt(s2(i)) );
[STD] s(i)=s2(i)^(1/2) );
@for(stocks: @free(u);@free(b) );
@for(link: @free(e) );
END
```

运行结果：

指数 M：均值 $m_0 = 1.191458$，方差 $s_0^2 = 0.02873661$，标准差 $s_0 = 0.1695188$。

A:u_1 =0.5639761， b_1 =0.4407264， s_1^2=0.005748320， s_1=0.07581767

B:u_2 = -0.2635059， b_2 = 1.239802， s_2^2= 0.01564263， s_2= 0.1250705

$C: u_3 = -0.5809590$， $b_3 = 1.523798$， $s_3^2 = 0.03025165$， $s_3 = 0.1739300$

模型建立 2

$$R = \sum_{i=1}^{3} x_i R_i = \sum_{i=1}^{3} x_i (u_i + b_i M + e_i)$$

可计算期望得： $ER = \sum_{i=1}^{3} x_i (u_i + b_i m_0)$

方差： $DR = \sum_{i=1}^{3} \left[(x_i b_i)^2 s_0^2 + x_i^2 s_i^2 \right]$

令 $y = \sum_{i=1}^{3} x_i b_i$ ，建立二次规划模型如下：

$$\min \quad \sum_{i=1}^{3} \left[y^2 s_0^2 + x_i^2 s_i^2 \right]$$

$$\text{s.t.} \quad y = \sum_{i=1}^{3} x_i b_i$$

$$\sum_{i=1}^{3} x_i (u_i + b_i m_0) \ge 0.15$$

$$x_1 + x_2 + x_3 = 1,$$

$$x_1, x_2, x_3 \ge 0$$

程序编写

```
MODEL:
Title 利用股票指数简化投资组合模型;
SETS:
  STOCKS/A, B, C/: u, b, s2, x;
ENDSETS
DATA:
! mean0,s20,u,b,s2是线性回归的结果数据;
mean0=1.191458;
s20 = 0.02873661;
s2  = 0.005748320,0.01564263,0.03025165;
u   = 0.5639761,  -0.2635059,-0.5809590;
b   = 0.4407264,  1.239802,  1.523798;
ENDDATA
[OBJ] MIN = s20*y*y + @sum(stocks: s2*x*x);
![OBJ] MIN = s20*@sqr(y) + @sum(stocks: s2*@sqr(x));
```

```
@sum(stocks: b*x)=y;
@sum(stocks: x)=1;
@sum(stocks: (u+b*mean0)*x)>1.15;
    END
```

运行结果

```
Local optimal solution found.
  Objective value:                          0.2465621E-01
  Extended solver steps:                            5
  Total solver iterations:                         55


  Model Title: 利用股票指数简化投资组合模型

                 Variable           Value        Reduced Cost
                  ……
                    X( A)        0.5266052          0.000000
                    X( B)        0.3806461          0.000000
                    X( C)        0.9274874E-01       0.000000
```

对问题 2，模型仿上建立，求解程序如下：

```
MODEL:
Title 含有国库券的投资组合模型;
SETS:
  STOCKS/ A,  B,  C, D/: Mean,X;
  STST(Stocks,stocks): COV;
ENDSETS
DATA:
  TARGET = 1.1; ! 1.15;
! Mean是收益均值,COV是协方差矩阵;
 mean=1.089083  1.213667  1.234583  1.05;
  COV=0.01080754 0.01240721 0.01307513 0
      0.01240721 0.05839170 0.05542639 0
      0.01307513 0.05542639 0.09422681 0
      0          0          0          0;
ENDDATA
[OBJ] MIN = @sum(STST(i,j): COV(i,j)*x(i)*x(j));
[ONE] @SUM(STOCKS: X) = 1;
[TWO] @SUM(stocks: mean*x) >= TARGET;
    END
```

运行结果摘取一部分如下：

```
                   X( A)       0.4343274E-01       0.000000
                    X( B)       0.2142643           0.000000
                    X( C)       0.7169959E-01        0.000000
```

```
X( D)      0.6706034          0.000000
```

与前面结果比较发现，风险资产之间的比例不变，变化的只是投资于风险资产与无风险资产之间的比例，这正是经济学上的“分离定理”，1981 年 Tobin 教授因发现这个定理获得诺贝尔奖。

对问题 3，

设购买股票 A,B,C 的比例为 y_1, y_2, y_3。

卖出股票 A,B,C 的比例为 z_1, z_2, z_3。

可建立模型如下：

$$\min \sum_{j=1}^{3}\sum_{i=1}^{3} x_i x_j \operatorname{cov}\left(R_i, R_j\right)$$

$$\text{s.t.}\quad x_1 ER_1 + x_2 ER_2 + x_3 ER_3 \ge 0.15$$

$$x_1 + x_2 + x_3 + 0.01\left(y_1 + y_2 + y_3 + z_1 + z_2 + z_3\right) = 1,$$
$$x_i = c_i + y_i - z_i, i = 1,2,3$$
$$x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3 \ge 0$$

程序编写

```
MODEL:
Title 考虑交易费的投资组合模型;
SETS:
   STOCKS/ A,  B,  C/: C,Mean,X,Y,Z;
   STST(Stocks,stocks): COV;
ENDSETS
DATA:
   TARGET = 1.15;
! 股票的初始份额;
 c=0.5  0.35  0.15;
! Mean是收益均值,COV是协方差矩阵;
 mean=1.089083  1.213667  1.234583;
  COV=0.01080754 0.01240721 0.01307513
      0.01240721 0.05839170 0.05542639
      0.01307513 0.05542639 0.09422681;
ENDDATA
[OBJ] MIN = @sum(STST(i,j): COV(i,j)*x(i)*x(j));
[ONE] @SUM(STOCKS: X+0.01*Y+0.01*Z) = 1;
[TWO] @SUM(stocks: mean*x) >= TARGET;
@FOR(stocks: [ADD] x = c - y + z);
    END
```

运行结果（部分）

```
X( A)      0.5264748          0.000000
 X( B)      0.3500000          0.000000
 X( C)      0.1229903          0.000000
```

例 2　钢管的订购和运输（2000 年全国大学生数学建模竞赛 B 题）

要铺设一条 $A_1 \to A_2 \to \Lambda \to A_{15}$ 的输送天然气的主管道，如图一所示(见下页)。经筛选后可以生产这种主管道钢管的钢厂有 $S_1, S_2, \Lambda\ S_7$。图中粗线表示铁路，单细线表示公路，双细线表示要铺设的管道(假设沿管道或者原来有公路，或者建有施工公路)，圆圈表示火车站，每段铁路、公路和管道旁的阿拉伯数字表示里程(单位 km)。

为方便计，1km 主管道钢管称为 1 单位钢管。

一个钢厂如果承担制造这种钢管，至少需要生产 500 个单位。钢厂 S_i 在指定期限内能生产该钢管的最大数量为 s_i 个单位，钢管出厂销价 1 单位钢管为 p_i 万元，如下表：

i	1	2	3	4	5	6	7
s_i	800	800	1000	2000	2000	2000	3000
p_i	160	155	155	160	155	150	160

1 单位钢管的铁路运价如下表：

里程(km)	≤300	301～350	351～400	401～450	451～500
运价(万元)	20	23	26	29	32

里程(km)	501～600	601～700	701～800	801～900	901～1000
运价(万元)	37	44	50	55	60

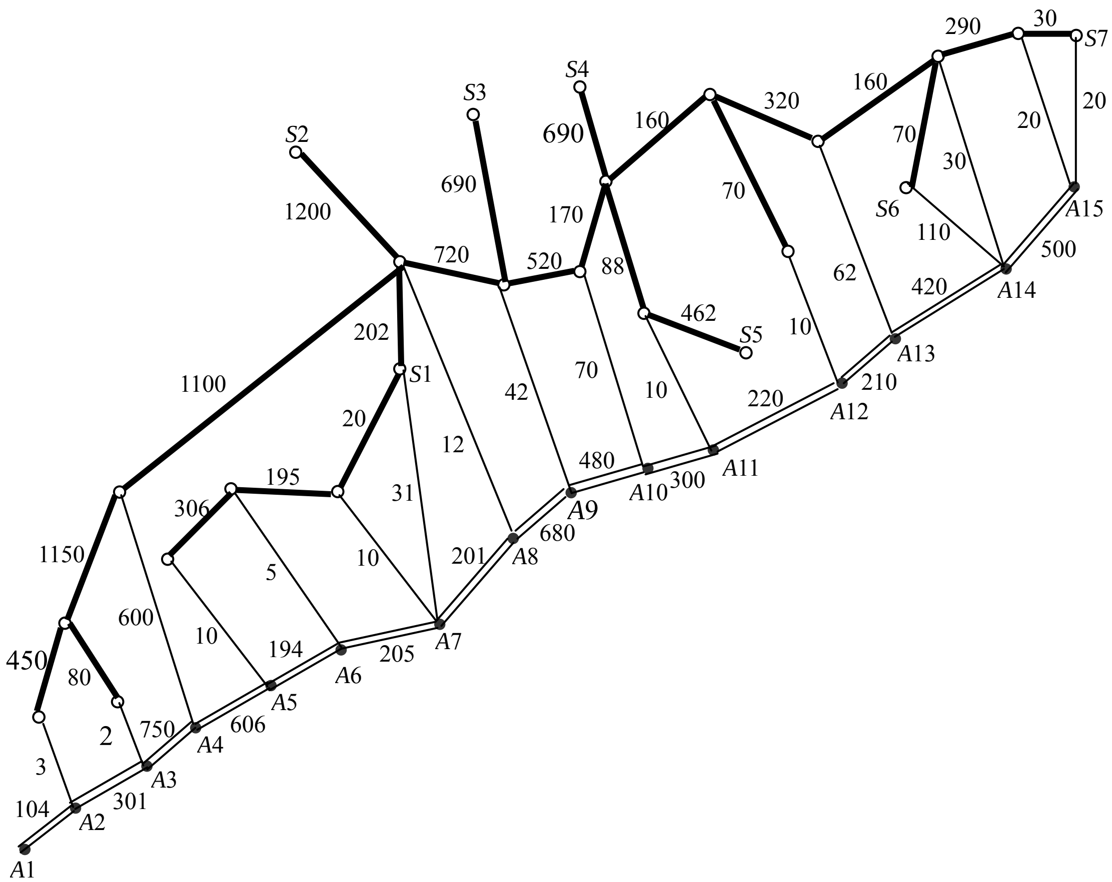
1000km 以上每增加 1 至 100km 运价增加 5 万元。

公路运输费用为 1 单位钢管每公里 0.1 万元（不足整公里部分按整公里计算）。

钢管可由铁路、公路运往铺设地点（不只是运到点 $A_1, A_2, \Lambda\ , A_{15}$，而是管道全线）。

（1）请制定一个主管道钢管的订购和运输计划，使总费用最小（给出总费用）。

（2）（3）问略



问题分析：从钢厂 s_i 向 A_j 运输钢管，应该走费用最短路相应的最小费用为 c_{ij}，包括销售价和运费两部分，可由图论方法得到（此处省略），我们认为沿管道路线每铺设一公里要卸下一单位钢管，因而在某一点可向左铺（设为 y_j），也可向右铺（设为 z_j），相邻两点对铺应该是合拢的。可建立如下二次规划模型：

模型建立

$$\min f = \sum_{i=1}^{7}\sum_{j=1}^{15} c_{ij}x_{ij} + \frac{0.1}{2}\sum_{j=1}^{15}(1+y_j)y_j + (1+z_j)z_j$$

s.t. $\sum_{j=1}^{15} x_{ij} \le s_i$（产量限制）

$\sum_{j=1}^{15} x_{ij} = 0$ 或 $\sum_{j=1}^{15} x_{ij} = 15$，（订货限制）

$z_j + y_{j+1} = a_j$， $j = 1\Lambda\ 14$（对铺合拢）

$$\sum_{i=1}^{7} x_{ij} = y_j + z_j, \quad j = 1\Lambda\ 15 \text{（定铺相等）}$$

$$y_1 = z_{15} = 0$$

$$x_{ij} \geq 0, y_j \geq 0, z_j \geq 0$$

模型建立

```
MODEL:
TITLE 钢管购运计划;
SETS:
    SUPPLY/S1..S7/:S,f;
    NEED/A1..A15/:a,y,z;
    LINK(Supply, need): C, X;
ENDSETS
DATA:
S=800 800 1000 2000 2000 2000 3000;
a=104, 301, 750, 606, 194, 205, 201, 680, 480, 300, 220, 210, 420, 500,;
c=@text(finalcost.txt);
@TEXT(FinalResult.txt)=@writefor(supply(i):
   @writefor(need(j):@format(x(i,j),'5.0f')), @newline(1) );
@TEXT(FinalResult.txt)=@writefor(need:@format(y,'5.0f') );
@TEXT(FinalResult.txt)=@write(@newline(1));
@TEXT(FinalResult.txt)=@writefor(need:@format(z,'5.0f') );
ENDDATA
[obj] MIN=@sum(link(i,j):c(i,j)*x(i,j))
          +0.05*@sum(need(j):y(j)^2+y(j)+z(j)^2+z(j));
! 约束;
@for(supply(i): [con1] @sum(need(j):x(i,j)) <= S(i)*f(i) );
@for(supply(i): [con2] @sum(need(j):x(i,j)) >= 500*f(i));
@for(need(j):   [con3] @sum(supply(i):x(i,j)) = y(j)+z(j));
@for(need(j)|j#NE#15:   [con4] z(j)+y(j+1)=a(j));
y(1)=0; z(15)=0;
@for(supply: @bin(f));
@for(need: @gin(y));
END
```

二次规划的 **Matlab** 求解

假设二次规划模型的形式为：

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

$$s.t. \qquad Ax \leq b$$

$$Aeqx = beq$$

$$lb \leq x \leq ub$$

调用函数为 quadprog（），调用形式如上，列举最简与最一般如下：

X=quadprog(H,c,A,b);

[x,fval,exitflag,output,lambda]=quadprog(H,c,A,b,Aeq,beq,lb,ub,x0,options)

求解方法在中等规模模式下，是有效集法，在大规模模式下，是一种置信域方法。

例 3 用 matlab 求解例 1 的问题一模型。

```
>> R=[1.300     1.225     1.149
      1.103     1.290     1.260
      1.216     1.216     1.419
      0.954     0.728     0.922
      0.929     1.144     1.169
      1.056     1.107     0.965
      1.038     1.321     1.133
      1.089     1.305     1.732
      1.090     1.195     1.021
      1.083     1.390     1.131
      1.035     0.928     1.006
      1.176     1.715     1.908];
H=2*cov(R);
A=mean(R);
b=1.15;
Aeq=[1 1 1];
beq=1;
x=quadprog(H,[0 0 0],-A,-b,Aeq,beq)
```

运行结果如下：

```
x =
    0.5301
    0.3564
    0.1135
```

为了分析回报率与风险之间的关系，引入偏好系数 β，建立如下模型：

目标： $\min \beta \sum_{j=1}^{3}\sum_{i=1}^{3} x_i x_j \operatorname{cov}(R_i, R_j) - (x_1 ER_1 + x_2 ER_2 + x_3 ER_3)$

约束： $x_1 + x_2 + x_3 = 1,$

$x_1, x_2, x_3 \ge 0$

可用下面程序画图观察分析回报率与风险的关系。需要调试

```
c=-A;
opt=optimset('large','off','Display','off');
for i=1:100,beta=0.01*i;H=beta*H;
x=quadprog(H,c,[],[],Aeq,beq,[0 0 0],[],[],opt);
```

```
REV(i)=-c*x;
STD(i)=sqrt(x'*H*x/2);
end
plot(REV,STD)
xlabel('预期收益率')
ylabel('风险（均方差）')
```

3.2 一般非线性规划

理论印象

约束优化的算法基本上可以分为两大类，直接法，和间接法，其中间接法事将约束优化问题转化为无约束问题求解，而直接发则不需要转化。直接法包括可行方向法，坐标轮换法等。

序列二次规划法(SQP)

思想：用一序列二次规划的解，去逼近原问题的解。在某点将目标函数二阶泰勒展开，约束一阶泰勒展开，略去高阶项即得到在这个点处的二次规划，称为二次规划子问题，求解得到新的解，在新的解（点）处重复这个过程。

算法：step1.选取初始点 x_1，正定矩阵 B_1，令 $k=1$；

Step2.求解二次规划子问题：

$$\min Q(d)=\frac{1}{2}d^T B_k d+\nabla f\left(x_k\right)^T d$$

$$\text{s.t.}\nabla h_i\left(x_k\right)^T d+h_i\left(x_k\right)=0, i=1\Lambda\ m$$

$$\nabla g_j\left(x_k\right)^T d+g_j\left(x_k\right)\le 0, j=1\Lambda\ l$$

求解得到 d_k 及对应拉格朗日乘子 λ_{k+1}；

Step3.从点 x_k，沿方向 d_k 进行一维搜索，确定步长 t_k，令 $x_{k+1}=x_k+t_k d_k$；

Step4.利用 λ_{k+1} 等信息，采用某种拟牛顿法对 B_k 进行修正得到 B_{k+1}，$k=k+1$ 转 Step2。

罚函数法：

思想：采用罚函数（可行点处罚函数值为原目标函数值，不可行点处罚函数值为一个很大的数）的思想，将约束最优化问题转化为无约束问题来求解，包括外点法，内点法，混合罚函数法。

算法可参考其它书籍。

例 1　供应与选址　建筑工地的位置(用平面坐标 *a*, *b* 表示，距离单位：公里)及水泥日用量 *d*(吨)下表给出。有两个临时料场位于 P (5,1), Q (2, 7),日储量各

有 20 吨。从 A, B 两料场分别向各工地运送多少吨水泥，使总的吨公里数最小。两个新的料场应建在何处，节省的吨公里数有多大？

	1	2	3	4	5	6
a	1.25	8.75	0.5	5.75	3	7.25
b	1.25	0.75	4.75	5	6.5	7.75
d	3	5	4	7	6	11

问题分析：和运输问题类似，约束涉及供应与需求，是线性的，目标是运量与距离的乘积。两问的模型都是一样的，第一问知道料场位置，距离是已知的，建立的模型为线性规划模型，第二问不知道料场的位置，距离是未知的，建立的模型为非线性规划模型。

符号说明：

(a_j, b_j)，d_j：工地的位置及需求量，$j = 1\Lambda\ 6$

(x_i, y_i)，e_i：料场的位置及日储量，$i = 1\Lambda\ 2$

c_{ij}：i 料场向 j 工地的运送量

模型建立：

$$\min f = \sum_{i=1}^{2}\sum_{j=1}^{6} c_{ij}\sqrt{(x_i - a_j)^2 + (x_i - b_j)^2}$$

$$\textbf{s.t.}\quad \sum_{j=1}^{6} c_{ij} \le e_i, i = 1\Lambda\ 2$$

$$\sum_{i=1}^{2} c_{ij} = d_j, j = 1\Lambda\ 6$$

$$c_{ij} \ge 0$$

程序编写

```
第二问：
MODEL:
Title Location Problem;
sets:
   demand/1..6/:a,b,d;
   supply/1..2/:x,y,e;
   link(supply,demand):c;
endsets
data:
a=1.25,8.75,0.5,5.75,3,7.25;
b=1.25,0.75,4.75,5,6.5,7.75;
d=3,5,4,7,6,11; e=20,20;
```

```
enddata
!初始段：对集合属性定义初值（迭代算法的迭代初值）;
init:
!初始点;
x,y=5,1,2,7;
endinit
min=@sum(link(i,j): c(i,j)*((x(i)-a(j))^2+(y(i)-b(j))^2)^(1/2) );
@for(demand(j):@sum(supply(i):c(i,j)) =d(j););
@for(supply(i): @sum(demand(j):c(i,j)) <=e(i); );
@for(supply: @bnd(0.5,X,8.75); @bnd(0.75,Y,7.75); );
END
```

直接运行可得到局部最优解，如果调用全局最优求解程序，耗时比较长，这也是求解非线性规划问题的特点。在程序中，如果将初始段的赋值放到数据段中，即是第一问的求解程序。

例 2　下料问题 2

原料钢管每根长 19 米，现有 4 种需求：4 米 50 根，5 米 10 根，6 米 20 根，8 米 15 根，由于采用不同切割模式太多，会增加生产和管理成本，规定切割模式不能超过 3 种，如何下料最节省？

预处理：

```
model:
title 搜索合理的下料方式;
sets:
dm:;
long(dm,dm,dm,dm):; !有三种需求长度，定义三维数组;
endsets
data:
dm=1..5;
@text('e:\数学建模\2009\rxl.txt')=@write(3*' ','下料方式',9*' ','余料长度',@newline(1));
@text('e:\数学建模\2009\rxl.txt')=@write(16*'-',4*' ',8*'-',@newline(1));
@text('e:\数学建模\2009\rxl.txt')=@writefor(long(i,j,k,m)|
            (19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#ge#0
        #and#(19-8*(i-1)-6*(j-1) -5*(k-1)-4*(m-1))#lt#a
      !输出下料方式到文本文件rxl.txt;
      :i-1,4*' ',j-1,4*' ',k-1,4*' ',m-1,8*' ',
19-8*(i-1)-6*(j-1) -5*(k-1)-4*(m-1),@newline(2));
!输出计算段计数过的下料方式总数;
@text('e:\数学建模\2009\rxl.txt')=@write('下料方式总数为：',2*' ',n,'种');
Enddata
calc:
a=@smin(8,6,5,4);
```

```
    b=@floor(19/a)+1;!用于确定dm;
    n=0;
    @for(long(i,j,k,m)|
                 (19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#ge#0
             #and#(19-8*(i-1)-6*(j-1)-5*(k-1)-4*(m-1))#lt#a
    :n=n+1);!下料方式计数;
    endcalc
end
```

运行程序后在 d:rxl.txt 中输出 16 中下料模式及余料长度，很显然，如果没有下料方式的限制，按照，就需要设置 16 个变量，如果需求种数更多，就会有更多的下料方式，因而先前一般下料问题的做法只适合处理小型问题。

决策变量

由于有下料方式种数的限制，由于不知道是哪三种下料方式，本着缺什么量就设什么量的原则，设置决策变量如下：

x_i ~按第 i 种模式切割的原料钢管根数(i=1,2,3)

$r_{1i}, r_{2i}, r_{3i}, r_{4i}$ ~ 第 i 种切割模式下，每根原料钢管生产 4 米、5 米、6 米和 8 米长的钢管的数量

模型建立

$$Min\ \ x_1 + x_2 + x_3$$

$$\text{s.t.}\ \ r_{11}x_1 + r_{12}x_2 + r_{13}x_3 \geq 50$$

$$r_{21}x_1 + r_{22}x_2 + r_{23}x_3 \geq 10$$

$$r_{31}x_1 + r_{32}x_2 + r_{33}x_3 \geq 20$$

$$r_{41}x_1 + r_{42}x_2 + r_{43}x_3 \geq 15$$

$$16 \leq 4r_{11} + 5r_{21} + 6r_{31} + 8r_{41} \leq 19$$

$$16 \leq 4r_{12} + 5r_{22} + 6r_{32} + 8r_{42} \leq 19$$

$$16 \leq 4r_{13} + 5r_{23} + 6r_{33} + 8r_{43} \leq 19$$

$$x_1 \geq x_2 \geq x_3$$

$$26 \leq x_1 + x_2 + x_3 \leq 31$$ %为了缩小可行域便于求解,可取特殊方式确定上下界。

$x_i, r_{1i}, r_{2i}, r_{3i}, r_{4i}$ (i =1,2,3) 为整数

程序编写

```
model:
```

```
Title 钢管下料 - 最小化钢管根数的LINGO模型;
SETS:
NEEDS/1..4/:LENGTH,NUM;
CUTS/1..3/:X;
PATTERNS(NEEDS,CUTS):R;
ENDSETS
DATA:
     LENGTH=4 5 6 8;
     NUM=50 10 20 15;
     CAPACITY=19;
ENDDATA
min=@SUM(CUTS(I): X(I) );
@FOR(NEEDS(I): @SUM(CUTS(J): X(J)*R(I,J) ) >NUM(I) ); !满足需求约束;
@FOR(CUTS(J): @SUM(NEEDS(I): LENGTH(I)*R(I,J) ) <CAPACITY ); !合理切割
     模式约束;
@FOR(CUTS(J): @SUM(NEEDS(I): LENGTH(I)*R(I,J) )
     >CAPACITY-@MIN(NEEDS(I):LENGTH(I)) );      !合理切割模式约束;
@SUM(CUTS(I): X(I) ) >26; @SUM(CUTS(I): X(I) ) <31;  !人为增加约束;
@FOR(CUTS(I)|I#LT#@SIZE(CUTS):X(I)>X(I+1) );  !人为增加约束;
@FOR(CUTS(J): @GIN(X(J)) ) ;
@FOR(PATTERNS(I,J): @GIN(R(I,J)) );
end
```

结果

模式 1：每根原料钢管切割成 3 根 4 米和 1 根 6 米钢管，共 10 根；

模式 2：每根原料钢管切割成 2 根 4 米、1 根 5 米和 1 根 6 米钢管，共 10 根；

模式 3：每根原料钢管切割成 2 根 8 米钢管，共 8 根。

用去原料钢管总根数为 28 根。

例 3　下料问题 3（2004 全国研究生数学建模竞赛 B 题）

一个好的下料方案首先应该使原材料的利用率最大，从而减少损失，降低成本，提高经济效益。其次要求所采用的不同的下料方式尽可能少，即希望用最少的下料方式来完成任务。因为在生产中转换下料方式需要费用和时间，既提高成本，又降低效率。此外，每种零件有各自的交货时间，每天下料的数量受到企业生产能力的限制。因此实用下料问题的目标是在生产能力容许的条件下，以最少数量的原材料，尽可能按时完成需求任务，同时下料方式数也尽量地小. 请你们为某企业考虑下面两个问题。

1. 建立一维单一原材料实用下料问题的数学模型，并用此模型求解下列问题，制定出在生产能力容许的条件下满足需求的下料方案，同时求出等额完成任务所需的原材料数，所采用的下料方式数和废料总长度. 单一原材料的长度为 3000mm，需要完成一项有53种不同长度零件的下料任务. 具体数据见表一，其中 l_i 为需求零件的长度，n_i 为需求零件的数量. 此外，在每个切割点处由于锯缝所产生的损耗为5mm. 据估计，该企业每天最大下料能力是100块，要求在4天内完成的零件标号（i）为：5, 7, 9, 12, 15, 18, 20, 25,

28,36,48；要求不迟于6天完成的零件标号(i)为:4,11,24,29,32,38,40,46,50.（提示：可分层建模。(1).先考虑用材料既少,下料方式又少的模型，或先仅考虑所用材料最少的模型及增加一种下料方式大致相当于使原材料总损耗增加0.08%情况下的最佳方案。(2).在解决具体问题时,先制定4天的下料方案,再制定6天的下料方案,最后制定53种零件的下料方案. 这一提示对第2题也部分适用.）

表一 需求材料的数据 单位：mm

i	1	2	3	4	5	6	7	8	9	10
l_i	1743	1680	1532	1477	1313	1285	1232	1217	1180	1177
n_i	4	216	104	38	4	60	4	8	6	10
i	11	12	13	14	15	16	17	18	19	20
l_i	1105	1055	1046	1032	1030	975	893	882	847	845
n_i	8	2	4	8	8	2	8	301	6	38
i	21	22	23	24	25	26	27	28	29	30
l_i	830	795	766	745	732	719	714	690	665	633
n_i	30	8	4	4	34	18	4	4	90	30
i	31	32	33	34	35	36	37	38	39	40
l_i	630	600	590	588	582	578	540	488	455	434
n_i	30	212	108	482	196	8	32	4	52	42
i	41	42	43	44	45	46	47	48	49	50
l_i	420	415	414	411	405	328	313	290	275	265
n_i	8	8	8	60	136	4	68	286	502	286
i	51	52	53							
l_i	255	184	155							
n_i	292	57	24							

2. 建立二维单一原材料实用下料问题的数学模型，并用此模型求解下列问题.制定出在企业生产能力容许的条件下满足需求的下料方案，同时求出等额完成任务所需的原材料块数和所需下料方式数.这个问题的单一原材料的长度为 3000mm,宽度为100mm，需要完成一项有43种不同长度和宽度零件的下料任务. 具体数据见表二，其中 l_i, w_i, n_i 分别为需求零件的长度、宽度和数量. 切割时的锯缝可以是直的也可以是弯的，切割所

引起的锯缝损耗忽略不计.据估计，该企业每天最大下料能力是20块 要求在4天内完成的零件标号(i)为: 3,7,9,12,15, 18, 20, 25, 28, 36.

表二 需求材料的数据 单位：mm

i	1	2	3	4	5	6	7	8	9	10
l_i	1105	1055	1046	1032	1030	975	893	882	847	845
w_i	30	20	50	30	20	50	30	20	30	30
n_i	24	6	12	24	24	6	24	1001	20	108
i	11	12	13	14	15	16	17	18	19	20
l_i	830	795	766	745	732	719	714	690	665	633
w_i	30	20	35	30	30	30	50	30	20	30
n_i	90	40	12	12	68	54	10	12	270	90
i	21	22	23	24	25	26	27	28	29	30
l_i	630	600	590	588	582	578	540	488	455	434
w_i	30	35	20	20	30	20	50	20	20	30
n_i	90	612	508	2082	496	24	62	20	162	92
i	31	32	33	34	35	36	37	38	39	40
l_i	420	415	414	411	405	328	313	290	275	265
w_i	20	30	20	30	20	30	50	30	20	30
n_i	40	24	40	180	536	12	128	686	2002	686
i	41	42	43							
l_i	255	184	155							
w_i	30	20	50							
n_i	692	357	52							

考虑第一问：

符号说明

L ——原材料的长度 （L=3000mm）;

W ——原材料的宽度（W=100mm）；

N ——所用的原材料总数量；

K ——所采用的下料方式总数量；

l_i ——第 i 号零件的长度 $l_i \le L$；（单位：mm，i=1, 2, 3…53）

w_i——第 i 号零件的宽度 $w_i \le W$；

n_i ——第 i 号零件的需求量；

a_{ij}——第 j 种下料方式中切割第 i 号零件的数量；

x_j——按第 j 种下料方式切割的原材料的数量；

c_j——按第 j 种下料方式切割的废料长度（mm）；

G_1——第一问中要求在 4 天内完成的零件号的集合

$$G_1=\{5,7,9,12,15,18,20,25, 28,36,48\}$$

G_2——第一问中要求在不迟于 6 天完成的零件号的集合

$$G_2=\{:4,11,24,29,32,38,40,46,50\}$$

G_3——第二问中要求在 4 天内完成的零件号的集合

$$G_3=\{3,7,9,12,15, 18, 20, 25, 28, 36.\}$$

模型建立

此问要求：在4天内完成的零件标号(i)为：5, 7, 9, 12, 15, 18, 20, 25, 28, 36, 48；不迟于6天完成的零件标号(i)为:4, 11, 24, 29, 32, 38, 40, 46, 50。而该企业每天最大下料能力是100块，我们要制定出在生产能力容许的条件下满足需求的下料方案，同时要求等额完成任务，我们的目标是要尽可能节省材料，尽可能用少的下料方式。

为此我们建立**多目标整数规划模型**：

$$MinN=\sum_j x_j\text{，}\quad MinK=\sum_j \frac{x_j}{x_j} \tag{1}$$

$$
S.T.\quad\begin{cases}
\sum_{i=1}^{53} a_{ij}(l_i+5)-5 \le L，\forall j \\
L-\sum_{i=1}^{53} a_{ij}(l_i+5) < \min_{1\le i\le 53}(l_i)，\forall j \\
\sum_j a_{ij}x_j = n_i，\ i=1,2,\Lambda\ 53 \\
\sum_{i\in G_1}(\sum_j \frac{a_{ij}x_j}{\sum_{i\in G_1} a_{ij}}) \le 100\times 4 \\
\sum_{i\in G_1 \text{Y} G_2}(\sum_j \frac{a_{ij}x_j}{\sum_{i\in G_1 \text{Y} G_2} a_{ij}}) \le 100\times 6 \\
a_{ij}\ge 0，\text{且为整数} \\
x_j \ge 0，\text{且为整数}
\end{cases}
$$

注：1. 目标函数：若采用了第 j 种下料方式，则 x_j 为大于 0 的整数，因此 x_i/x_j=1；若没有采用第 j 种下料方式，则 x_j 为 0，可得：x_i/x_j=0/0=0，这样 $K=\sum(x_j/x_j)$即表示了所用的下料方式数量；

2. 约束条件中第一条是：考虑了锯缝时，原材料长度 L 对下料方式的限制，即对于任意一种下料方式，所得到的零件总长度与锯缝总长度之和要小于等于 L；

3. 约束条件中第二条是：考虑了锯缝时，对于每一种下料方式的废料长度要小于零件的最小长度；

4. 约束条件中第三条是：为了满足题中要求的等额完成任务的限制条件；

5. 约束条件中第四条是：为了满足在企业每天生产能力是 100 块时，要求在 4 天内完成零件集合 G_1 的条件，其中 $a_{ij}x_j/\sum a_{ij}(i\in G_1)$第 j 种下料方式中所切割的第 i 种零件数占这种下料方式中所切割的零件集合 G_1 中零件数的权数，因此$\sum(\sum(a_{ij}x_j/\sum a_{ij}))\ (i\in G_1)$完成零件集合 G_1 所用的原材料数，又由于在 4 天内要完成零件集合 G_1，故上面算出的所用的原材料数要小于等于 100×4，注意若$\sum a_{ij}=0(i\in G_1)$，即表示第 j 种下料方式中没有切割到零件集合 G_1 中的零件，因此：$a_{ij}=0,\forall i\in G_1$；

6. 约束条件中第五条和第四条的解释类似；约束条件中第六条和第七条表示 a_{ij} 和 x_j 要取整数。

对于废料的度量：由于存在锯缝为 5mm，对任何一种可行的下料方式

$(a_{1j}, a_{2j}, \cdots, a_{53j})$，则其满足条件 $\sum_{i=1}^{53} a_{ij}(l_i+5)-5 \le L$，所以如果单纯的用 $L-\sum_{i=1}^{53} a_{ij}(l_i+5)$ 来度量此种下料方式的废料是不对的，这可能取到负值；实际上，又由于对问题一有假设 4，我们可以知道：对所有满足 $L-\sum_{i=1}^{53} a_{ij}(l_i+5) \le 0$ 的下料方式来说，废料都为 0；故而我们可以得到废料的度量方式：

$$c_j = \begin{cases} L-\sum_{i=1}^{53} a_{ij}(l_i+5), 当 L-\sum_{i=1}^{53} a_{ij}(l_i+5) > 0 时 \\ 0，当 L-\sum_{i=1}^{53} a_{ij}(l_i+5) \le 0 时 \end{cases}$$

经过数学处理，得到：

$$c_j = \left\{\left[L-\sum_{i=1}^{53} a_{ij}(l_i+5)\right] + \left|L-\sum_{i=1}^{53} a_{ij}(l_i+5)\right|\right\} \Big/ 2$$

因此废料总量为：$C=\sum c_j x_j$

废弃率定义为：$q=C/\sum 3000x_j=\sum c_j x_j/\sum 3000x_j$

利用率定义为：$p=1-q=1-C/\sum 3000x_j=1-\sum c_j x_j/\sum 3000x_j$

此模型变量较多，直接求解比较困难（几乎是不可能的），因而必须分层次处理这个问题。下面给出刘承平老师的解题思路（详细过程略）：

问题的分析：理论上解决此类问题最好的方法是先求出所有的下料方式（53种零件的所有下料方式共10307032种），然后建立整数线性规划模型求解，这种方法对于本问题是不实际的.

另一种方法像文献[2]一样建立整数非线性规划模型用Lingo软件求解，这种方法对于本问题也是不实际的.（ 所用的原材料的数量为N=809根，所用的下料方式为K=54，废弃总长度=46113mm,废弃率为q=1.9%,利用率为：$p=1-q$ =98.1%;）

然而实际问题总是要解决的，解决实际问题时，首先考虑前人是怎么解决的，如何得到更好的解决方案，在求解过程中寻求解决实际问题的最好方案. 本文从建立人工下料模型开始，利用现有的规划理论逐步优化得到一个本问题特别好的下料方案.

人工下料模型：人工下料模型是依次下料最长的零件，并尽量利用余料，即只要余料可用就充分利用余料，而不用新的原材料. 这种模型很容易编写一个MATLAB 程序求解，运算结果得到 64 种下料方式、下料总根数为 811 根、余料长度总和为 42217mm. 这一结果说明最好的下料方案至少需要用原材料数 811-42217/3005≈797(根).

逐根优化的下料模型：由于 53 种零件的所有下料方式数太大，逐根优化的下料模型每次从剩余零件的所有下料方式中挑选 1 种下料，直至完成所有的下料任务. 每次下料方式的选择模型如下：

$$\max \sum (l_i + 5) x_i$$

$$s.t.\begin{cases} \sum (l_i + 5) x_i \le 3005 \\ x_i \le N_i, \text{取整} \end{cases}$$

x_i 为该下料方式中第 i 号规格零件的下料数量）N_i 为第 i 号规格零件的剩余数量。

逐根优化的下料模型结果为：下料方式 71 种、下料总根数为 926 根、余料长度总和为 387792㎜.

分步优化的下料模型：首先，选用需求数量较多的零件开始，编写一个 MATLAB 程序搜索所有的下料方式建立线性规划模型求解，并取其解的整数部分. 本问题中选用需求零件的数量超过 55 件的 17 种规格零件，用 MATLAB 编程搜索所有的下料方式，一共有 61054 种下料方式，删除余料长度超过 20㎜ 后剩下 8187 种，利用线性规划模型求解得到 17 种下料方式(表 2 中 1-17)，下料时用掉原材料总根数为 660 根、余料长度总和为 2555㎜. 下料后，剩余需求零件的种类仍为 53 种.

其次，采用优化的人工下料模型，即在上述人工下料模型中删除余料长度超过 10㎜ 的下料方式后又得到 16 种下料方式(表 2 中 18-33)，下料时累计用掉原材料总根数为 725 根、余料长度累计总和为 2854㎜. 此时，剩余需求零件的种类减少到 39 种.

最后，采用逐根优化的下料模型，再得到 38 种下料方式(表 2 中 34-71)，下料时累计用掉原材料总根数为 799 根、下料方式 71 种、余料长度累计总和为 6157㎜. 此时，完成了全部零件的下料任务.

下料方案 1

编号	下料方式(零件编号,零件下料数量)	余料长度(mm)	原材料根数
1	(2,1) (18,1) (44,1)	17	42
2	(2,1) (32,1) (44,1) (48,1)	4	17
3	(2,1) (32,1) (45,1) (48,1)	10	10
4	(2,1) (33,1) (45,1) (48,1)	20	30
5	(2,1) (35,1) (50,2) (52,1)	4	12
6	(2,1) (48,2) (49,1) (51,1) (52,1)	1	42
7	(2,1) (48,2) (50,2) (52,1)	1	1
8	(2,1) (50,2) (51,3)	0	57
9	(3,1) (34,2) (49,1)	2	103
10	(6,1) (35,2) (50,2)	1	59
11	(18,2) (45,1) (49,2) (51,1)	1	6
12	(18,1) (29,1) (34,1) (48,1) (49,2)	0	66
13	(18,1) (29,1) (35,2) (50,1)	4	23
14	(18,1) (32,1) (34,2) (47,1)	9	67
15	(18,1) (34,1) (45,1) (48,1) (49,2) (51,1)	0	71
16	(18,1) (35,1) (45,1) (49,4)	1	16
17	(32,3) (33,2)	0	38
18	(9,2) (31,1)	0	3
19	(11,1) (12,1) (21,1)	0	2

20	(22,3) (32,1)	0	2
21	(37,5) (49,1)	0	6
22	(2,1) (5,1)	2	4
23	(30,1) (31,3) (39,1)	2	9
24	(10,2) (30,1)	3	5
25	(8,1) (11,1) (29,1)	3	1
26	(13,2) (17,1)	5	2
27	(24,4)	5	1
28	(11,1) (14,1) (19,1)	6	5
29	(39,5) (40,1) (51,1)	6	2
30	(17,1) (18,2) (46,1)	0	2
31	(21,3) (38,1)	7	4
32	(15,1) (16,2)	10	1
33	(20,2) (21,1) (39,1)	10	16
34	(7,1) (50,1) (52,2) (53,7)	0	1
35	(7,1) (47,1) (48,4) (50,1)	0	1
36	(7,1) (46,1) (48,1) (49,1) (50,2) (53,2)	0	1
37	(7,1) (45,2) (46,1) (48,1) (53,2)	0	1
38	(1,1) (43,3)	0	2
39	(43,1) (44,1) (45,1) (53,11)	0	1
40	(39,1) (41,5) (42,1)	0	1
41	(39,1) (41,3) (42,2) (50,1) (53,1)	0	1
42	(37,1) (39,5) (53,1)	0	1
43	(34,2) (39,3) (40,1)	0	2
44	(33,1) (35,1) (37,1) (40,1) (42,1) (43,1)	0	1
45	(8,1) (32,1) (33,1) (36,1)	0	1
46	(28,3) (39,2)	0	1
47	(27,1) (35,1) (40,1) (42,3)	0	1
48	(27,1) (30,1) (32,1) (36,1) (39,1)	0	1
49	(26,1) (28,1) (36,2) (42,1)	0	1
50	(25,1) (30,1) (35,1) (36,1) (39,1)	0	1
51	(17,1) (22,1) (26,1) (36,1)	0	2
52	(14,1) (25,1) (30,1) (34,1)	0	1
53	(6,1) (30,2) (40,1)	0	1
54	(2,1) (25,1) (36,1)	0	1
55	(20,1) (30,2) (40,2)	1	5
56	(23,2) (25,1) (26,1)	2	2
57	(3,1) (35,1) (40,2)	3	1
58	(8,1) (17,1) (40,2)	7	2
59	(4,1) (30,1) (40,2)	7	1
60	(8,1) (14,1) (25,1)	9	2
61	(8,1) (15,1) (25,1)	11	2
62	(4,2)	41	18
63	(4,1) (15,1) (40,1)	49	1

64	(25,4)	57	6
65	(15,2) (40,2)	57	2
66	(25,1) (40,5)	73	1
67	(26,1) (40,5)	86	1
68	(1,1) (26,1) (40,1)	94	2
69	(26,4)	109	2
70	(26,2) (27,2)	119	1
71	(19,1) (20,1)	1303	1

程序实现？？？？？
例

3.3 非线性规划问题的 MATLAB 求解

用 MATLAB 优化工具箱求解非线性规划时要求化为如下形式：

$$\min z = f(x)$$

$$\text{s.t.}\quad Ax \le b$$

$$Aeqx = beq$$

$$lb \le x \le ub$$

$$c(x) \le 0$$

$$ceq(x) = 0$$

求解程序名为 fmincon，最简调用为：
x=fmincon(f,x0,A,b)
最一般的调用格式为：
[x,fval,exitflag,output,lambda,grad,hessian]=fmincon(@f,x0,A,b,Aeq,beq,lb,ub,nonlcon,options,P1,P2,...)

@f 中的 f 为目标函数的 m 文件名（一般将目标函数用 m 文件给出），x0 为初值，A,b,Aeq,beq,lb,ub 和前面是一样的，是线性约束，nonlcon 是非线性约束条件，一般也用 m 文件给出（比如为 confun.m，调用时 nonlcon 的地方写入@confun），P1,P2 等为传给目标函数 f.m 或者非线性约束文件 confun.m 的参数.

对于中等规模算法，用的是 SQP 方法，对于大规模的算法（仅有线性约束或上下界约束）用的是置信域方法。

例 1 求解优化问题

$$\min\ f(s,t) = s^4 - 4s - 8t + 15$$

$$s.t.\ 9 - s^2 - t^2 \le 0$$

$$2s + 3t \le 2$$

$$t - s \le 5$$

Step1：建立目标函数文件 f(x).m：

```
function  y=f(x)
y=x(1)^4-4*x(1)-8*x(2)+15;
```

Step2 :建立非线性约束文件 confun.m ：

```
function  [c,ceq]=confun(x)
c=9-x(1)^2-x(2)^2 ;
ceq=[] ;
```

Step3 ：在命令窗口输入命令：

```
>>A=[2 3 ;1 -1] ;
    b=[2   5]’;
x=fmincon(@f(x),[1,2],A,b,[],[],[],[],@confun)
```

运行可求得最优解为（s，t）=（1.2420，-2.7308）。

例2　用 matlab 求解 3.1 节的例 1 供应与选址问题。

```
Function  f=f(x)
   a=[1.25,8.75,0.5,5.75,3,7.25];
   b=[1.25,0.75,4.75,5,6.5,7.75];
f=0;
for i=1:6
%( x(13),x(14))为第一个料场坐标，x(1)—x（6）为其到 6 个工地的运量;
%( x(15),x(16))为第二个料场坐标，x(7)—x（12）为其到 6 个工地的运量;
  d1=sqrt((x(13)-a(i))^2+(x(14)-b(i))^2);
d2=sqrt((x(15)-a(i))^2+(x(16)-b(i))^2);
f=d1*x(i)+d2*(i+6)+f;
end
```

计算程序如下：

```
format  short
   a=[1.25,8.75,0.5,5.75,3,7.25];
   b=[1.25,0.75,4.75,5,6.5,7.75];
d=[3,5,4,7,6,11]’; e=[20,20]’;
A1=[ones(1,6),zeros(1,10);zeros(1,6),ones(1,6),zeros(1,4));
B1=e;
A2=[eye(6),eye(6),zeros(6,4)];
B2=d;
X0=[zeros(1,12) 5 1 2 7];
v1=zeros(1,16);
v2=[d’,d’,[10,10,10,10]];
opt=optimset(‘LargeScale’,’off’,’MaxFunEbals’,4000,’Maxlter’,1000);
[x,f,exitflag,output]=fmincon(‘f(x)’,x0,A1,B1,A2,B2,v1,v2,[],opt);
```

计算得新料场的位置：（5.6959 4.9285），（7.2500，7.7500），总吨公里 89.88.

第四章 多目标规划

第一节　基本理论

线性规划只研究在满足一定条件下，单一目标函数取得最优解，而在企业管理中，经常遇到多目标决策问题，如拟订生产计划时，不仅考虑总产值，同时要考虑利润，产品质量和设备利用率等。这些指标之间的重要程度（即优先顺序）也不相同，有些目标之间往往相互发生矛盾。

线性规划致力于某个目标函数的最优解，这个最优解若是超过了实际的需要，很可能是以过分地消耗了约束条件中的某些资源作为代价。线性规划把各个约束条件的重要性都不分主次地等同看待，这也不符合实际情况。

求解线性规划问题，首先要求约束条件必须相容，如果约束条件中，由于人力，设备等资源条件的限制，使约束条件之间出现了矛盾，就得不到问题的可行解，但生产还得继续进行，这将给人们进一步应用线性规划方法带来困难。

为了弥补线性规划问题的局限性，解决有限资源和计划指标之间的矛盾，在线性规划基础上，建立多目标规划方法，从而使一些线性规划无法解决的问题得到满意的解答。

例 1　重新考虑 2.1 中的例 3 选址问题。

分析：当我希望得到最大收益的时候，我也希望我的投资是最少的。

模型：

$$\max f = \sum_{i=1}^{7} c_i x_i$$

$$\min g = \sum_{i=1}^{7} b_i x_i$$

$$s.t.\begin{cases} \sum_{i=1}^{7} b_i x_i \le b \\ x_1 + x_2 + x_3 \le 2 \\ x_4 + x_5 \ge 1 \\ x_6 + x_7 \ge 1 \\ x_i = 0\text{或}1 \qquad i = 1,2,\dots,7 \end{cases}$$

例 2　用直径为 1(单位长)的圆木制成截面为矩形的梁,为使重量最轻,而强度最大,问截面的高与宽应取何尺寸?

解：设高 x_1 宽 x_2。

$$\min \quad x_1 x_2$$

$$\max \quad \frac{1}{6} x_1 x_2^{\ 2}$$

$$s.t. \quad x_1^2 + x_2^2 = 1$$

$$x_1, x_2 \geq 0$$

例 3 重新考虑 3.1 中的例 1.

分析：希望风险最小而获利最大。

模型：

$$\min D(x_1R_1 + x_2R_2 + x_3R_3)$$

$$\max \ (x_1ER_1 + x_2ER_2 + x_3ER_3)$$

$$\text{s.t.} \ \ x_1ER_1 + x_2ER_2 + x_3ER_3 \geq 0.15$$

$$x_1 + x_2 + x_3 = 1,$$
$$x_1, x_2, x_3 \geq 0$$

多目标决策问题有许多共同的特点，其中最显著的是：目标的不可公度性，和目标间的矛盾性。因此不能简单的把多个目标归并为单个目标，并使用单目标决策问题的方法去求解多目标问题。

一般确定性多目标问题的数学模型为：

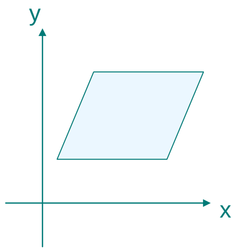
$$\min_{x \in R^n} f(x) = (f_1(x), f_2(x), \Lambda\ , f_p(x))$$

$$\text{s.t.} \begin{cases} g_j(x) \geq 0, j = 1,2.\Lambda\ , m \\ h_k(x) = 0, k = 1,2,\Lambda\ , l \end{cases}$$

f_i，g_j，h_k：$R^n \to R$．$p \geq 2$。

记可行域为 D.

绝对最优解 x^*：$\forall x \in D, f_i(x) \geq f_i(x^*)$. 通常是不存在的。如下图平行四边形区域.



有效解（Pareto 解）x^*：不存在 $x \in D$，使得 $f_i(x) \leq f_i(x^*)$，$j = 1\Lambda\ p$，存在某个 i_0，$f_{i_0}(x) < f_{i_0}(x^*)$。有效解通常是很多的。

弱有效解 x^*：不存在 $x\in D$，使得 $f_i(x) < f_i(x^*)$， $j=1\Lambda\ p$。有点目标达到最优。

多目标决策的本质问题是：如何跟户决策者的主观价值判断对有效解及你选哪个好坏的比较？由于可行域中的一个点，对应目标函数是一个向量，所以问题实际是：如何比较两个向量的大小？

5 个基本要素：

决策变量：$x=(x_1,x_2,\Lambda\ ,x_n)^T$

目标函数：$f(x)=(f_1(x),f_2(x),\Lambda\ ,f_p(x))$

可行解集：$X=\left\{x\in R^n \middle| g_j(x)\ge 0, j=1,2.\Lambda\ ,m h_k(x)=0,k=1,2,\Lambda\ ,l\right\}$

偏好关系：在像集 $f(X)$ 上有某个二元关系π（称为偏好序）反映决策者的偏好。

解得定义：在已知的偏好关系下定义 f 在 X 上的最好解。

第二节　多目标规划的常用解法

思想：转化为单目标问题

评价函数法

如果能够根据决策者提供的偏好信息构造一个实函数 $u(f(x))$（称为效用函数），使得求满意解等价于求以该实函数为新目标函数的单目标规划问题的最有解，则可用已有算法求解。

评价函数法的基本思想是：借助于几何或应用中的直观效果. 构造所谓的评价函数 $u(f(x))$. 从而将多目标优化问题转化为单目标优化问题. 然后利用单目标优化问题的求解方法求出最优解. 并把这种最优解当作多目标优化问题的最优解. 这里关键的问题是转化后的单目标优化问题的最优解必须是多目标问题的有效解和弱有效解. 否则是不能接受的.

由于以下两个原因限制了该方法的实际应用：
许多场合下，决策者提供的偏好信息不足于确定效用函数。
构造实际问题的效用函数是相当困难的。

（1）线性加权法：

先按目标函数 $f_1(x),f_2(x),\Lambda\ ,f_p(x)$ 的重要程度给一组权数 $\varpi_1,\varpi_2,\Lambda\ ,\varpi_p$，满足 $\varpi_i\ge 0,\sum_{i=1}^{p}\varpi_i=1$，然后定义评价函数：$u(f(x))=\sum_{i=1}^{p}\varpi_i f_i(x)$,最后求解非线性规划问

题：$\min\limits_{x\in D} u\left(f(x)\right)$。

如果每个函数的属性独立于其它函数的属性，则可以用这个方法。

（2）变权加权法：

在线性加权方法中，一旦多属性效用函数确定了，则其权值就确定了，而不依赖于各属性的效用，有些时候权值是随着其相应属性效用的变化而变化的，此时用变权加权形式：

$$\min_{x\in D} u\left(f(x)\right)=\min_{x\in D}\sum_{i=1}^{p}\varpi_i\left(f_i(x)\right)f_i(x)$$

（3）**指数加权法：**

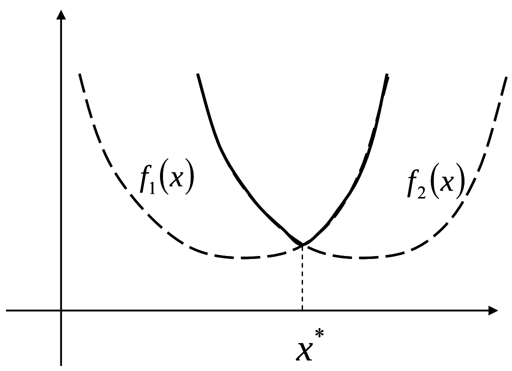
如果各个属性是串行结构，即只要有一个效用为 0，则总体效用为 0，可采用指数加权形式：

$$\min_{x\in D} u\left(f(x)\right)=\min_{x\in D}\prod_{i=1}^{p}\left(f_i(x)\right)^{\varpi_i}$$

（4）极小极大（min-max）法

在决策时，有时候采取保守策略是稳妥的。即在最坏的情况下，寻求最好的结果。即求解非线性规划问题：

$$\min_{x\in D} u\left(f(x)\right)=\min_{x\in D}\left(\max_{1\le i\le p} f_i(x)\right)$$



此非线性规划问题目标不可微，不能直接用于基于梯度的算法，可等价转化为：

$$\min_{x,t}\quad t$$

s.t. $f_i(x)\le t, i=1\Lambda\ p$

$x\in D$

（5）**理想点法**

先求解单目标的最优值确定理想点：$f_i^*=\min\limits_{x\in D} f_i(x), i=1\Lambda\ p$，

在找距离理想点最近的 x 点作为最优解：$\min\limits_{x\in D} u\left(f(x)\right)=\min\limits_{x\in D}\sqrt{\sum\limits_{i=1}^{p}\left(f_i(x)-f_i^*\right)^2}$

（6）**加权偏差函数法**

a．给定理想点：$\left(\bar{f}_1,\bar{f}_2,\Lambda\ ,\bar{f}_p\right)$，求非线性规划问题：

$$\min_{x\in D} u(f(x)) = \min_{x\in D}\left[\sum_{i=1}^{p}\varpi_i\left(f_i(x)-\bar{f}_i\right)^k\right]^{\frac{1}{k}}$$，（$\varpi_i \ge 0$， $\bar{f}_i \le f_i^*$）

b．采用几何平均函数作为评价函数

$$\min_{x\in D} u(f(x)) = \min_{x\in D}\prod_{i=1}^{p}\left(f_i(x)-\bar{f}_i\right)^{\varpi_i}$$，（$\varpi_i > 0$， $\bar{f}_i < f_i^*$）

c．采用极大模函数：

$$\min_{x\in D} u(f(x)) = \min_{x\in D}\max_{1\le i\le p}\varpi_i\left(f_i(x)-\bar{f}_i\right)$$

（7）费效比函数：

$$\min_{x\in D} u(f(x)) = \min_{x\in D}\prod_{i=1}^{k} f_i(x)\Big/\prod_{i=k+1}^{p} f_i(x)$$

特别的，如果目标为：$\min f_1(x)$，$\max f_2(x)$，则可求解$\max\frac{f_2(x)}{f_1(x)}$，若 f_1 为投资，f_2为收益，则计算的结果为单位投资的总收入最大。

（8）功效系数函数：

对不同的性质的目标函数统一量纲，再构造效用函数：

$f_{i,\min} = \min_{x\in D} f_i(x), i = 1\Lambda\ p$， $f_{i,\max} = \max_{x\in D} f_i(x), i = 1\Lambda\ p$

比如构造功效系数函数：$d_i(x) = \frac{f_{i,\max} - f_i(x)}{f_{i,\max} - f_{i,\min}} \in [0,1]$，

然后求解规划问题：$\max_{x\in D}\sum_{i=1}^{p} d_i(x)$或$\max_{x\in D}\prod_{i-1}^{p} d_i(x)$。

还可以对功效系数函数进行加权构造效用函数，如

$$u(x) = \left(\prod_{i-1}^{p}\left[d_i(f_i(x))\right]^{\varpi_i}\right)^{\frac{1}{\sum_{i=1}^{p}\varpi_i}}$$

（9）参考目标法

约束法：在多个目标中选定一个主要目标，而对其他目标设定一个期望值，在要求结果不比期望值坏的情况下，求主要目标的最优值。

比如第一个目标设为主要目标的话，设定期望值$\left(f_2^0,\Lambda\ f_p^0\right)$有：

$\min f_1(x)$

s.t. $f_i(x) \le f_i^0$， $i = 2\Lambda\ p$

$x \in D$

优点:适当的选择目标函数值，可以方便的得到原问题的每一个有效解。而且

重点保证了重要目标的效益，同时又照顾了其他目标，而且在当前点的 Kuhn-Tucker 乘子可用来确定置换率，帮助决策者寻找更合意的方案。

分层序列法：
把多个目标按照重要程度进行排序，先求第一个目标的最有解，在达到此目标的条件下求第二个目标的最优解，依次类推直到求得最后一个目标的最优解。
缺点：当先前的目标只有唯一的最优解时，后面的求解失去了意义。
改进—**宽容分层序列法**：给前面的最优值设置一定的宽容值，和最优值相差宽容值之内的都是可以接受的。

（10）逼近理想解法
在多目标问题的决策过程中，决策人总是希望找到所有属性指标都为最优的解，即希望尽可能的远离各属性指标都最劣的解。所有属性指标都处于最优的解称为正理想解，所有属性指标都处于最劣的解称为负理想解，从几何上看，若一个方案在某种测试下，最靠近理想解，而又最远离负理想解，则该方案就认为是决策问题的最优解。

正负理想解： $f_i^+ = \min_{x\in D} f_i(x)$， $f_i^- = \max_{x\in D} f_i(x)$， $i = 1,\Lambda, p$

计算距离，不妨取为欧式距离： $d^+(x) = \left(\sum_{i=1}^{p}\left(f_i^+ - f_i(x)\right)^2\right)^{\frac{1}{2}}$，

$$d^-(x) = \left(\sum_{i=1}^{p}\left(f_i(x) - f_i^-\right)^2\right)^{\frac{1}{2}}$$

计算测度： $c(x) = \dfrac{d^-(x)}{d^-(x) + d^+(x)}$

求最大测度： $\max_{x\in D} c(x)$。

并以此解作为多目标决策问题的最优解。

第三节　模糊线性规划
例　小点的题
例 公交线路问题（全国大学生数学　建模竞赛）

第四节　多目标规划的 matlab 求解

用目标达到法求解多目标规划的计算过程，可以通过调用 Matlab 软件系统优化工具箱中的 fgoalattain 函数实现。

在 Matlab 的优化工具箱中，fgoalattain 函数用于解决此类问题。其数学模型形式为：

$$\min \gamma$$

$$F(x)-weight \cdot \gamma \le goal$$

$$c(x) \le 0$$

$$ceq(x)=0$$

$$A x \le b$$

$$Aeq\ x=beq$$

$$lb \le x \le ub$$

其中，x，weight，goal，b，beq，lb 和 ub 为向量；A 和 Aeq 为矩阵；c(x)，ceq(x)和 F(x)为函数。

调用格式：

x=fgoalattain(F,x0,goal,weight)

x=fgoalattain(F,x0,goal,weight,A,b)

x=fgoalattain(F,x0,goal,weight,A,b,Aeq,beq)

x=fgoalattain(F,x0,goal,weight,A,b,Aeq,beq,lb,ub)

x=fgoalattain(F,x0,goal,weight,A,b,Aeq,beq,lb,ub,nonlcon)

x=fgoalattain(F,x0,goal,weight,A,b,Aeq,beq,lb,ub,nonlcon,options)

x=fgoalattain(F,x0,goal,weight,A,b,Aeq,beq,lb,ub,nonlcon,options,P1,P2)

[x,fval]=fgoalattain(…)

[x,fval,attainfactor]=fgoalattain(…)

[x,fval,attainfactor,exitflag,output]=fgoalattain(…)

[x,fval,attainfactor,exitflag,output,lambda]=fgoalattain(…)

说明：F 为目标函数；*x*0 为初值；goal 为 F 达到的指定目标；weight 为参数指定权重；A、b 为线性不等式约束的矩阵与向量；Aeq、beq 为等式约束的矩阵与向量；lb、ub 为变量 *x* 的上、下界向量；nonlcon 为定义非线性不等式约束函数 c(x)和等式约束函数 ceq(x)；options 中设置优化参数。

x 返回最优解；fval 返回解 *x* 处的目标函数值；attainfactor 返回解 *x* 处的目标达到因子；exitflag 描述计算的退出条件；output 返回包含优化信息的输出参数；lambda 返回包含拉格朗日乘子的参数。

某企业拟生产 A 和 B 两种产品，其生产投资费用分别为 2100 元/*t* 和 4800 元/*t*。A、B 两种产品的利润分别为 3600 元/*t* 和 6500 元/*t*。A、B 产品每月的最大生产能力分别为 5*t* 和 8*t*；市场对这两种产品总量的需求每月不少于 9*t*。试问该企业应该如何安排生产计划，才能既能满足市场需求，又节约投资，而且使生产利润达到最大最。

该问题是一个线性多目标规划问题。如果计划决策变量用 x_1 和 x_2 表示，它们分别代表 A、B 产品每月的生产量（单位：*t*）；$f_1(x_1,x_2)$ 表示生产 A、B 两种产品的总投资费用（单位：元）；$f_2(x_1,x_2)$ 表示生产 A、B 两种产品获得的总利润（单位：元）。那么，该多目标规划问题就是：求 x_1 和 x_2，使：

$$\min f_1(x_1,x_2)=2100x_1+4800x_2$$

$$\max f_2(x_1,x_2)=3600x_1+6500x_2$$

而且满足：

$$\begin{cases} x_1 \le 5 \\ x_2 \le 8 \\ x_1 + x_2 \ge 9 \\ x_1, x_2 \ge 0 \end{cases}$$

求解程序如下：

① 编辑目标函数 M 文件 ff12.m

```
function f=ff12(x)
f(1)=2100*x(1)+4800*x(2);
f(2)=-3600*x(1)–6500*x(2);
```

② 按给定目标取：

```
goal=[30000,-45000];
weight=[30000,-45000];
```

③ 给出：

```
x0=[2,2];
A=[1 0; 0 1;-1 -1];
b=[5,8,-9];
lb=zeros(2,1);
```

④ 调用 fgoalattain 函数：

```
[x,fval,attainfactor,exitflag]=fgoalattain(@ff12,x0,goal,weight,A,b,[],[],lb,[])
```

运行后，输出结果为：

```
x =
     5     4
fval =
        29700       -44000
attainfactor =
    -0.0100
exitflag =
     1
```

例 2：投资问题 某企业拟用 1000 万元投资于 A、B 两个项目的技术改造。设 x_1、 x_2 分别表示分配给 A、B 项目的投资（万元）。据估计，投资项目 A、B 的年收益分别为投资的 60%和 70%；但投资风险损失，与总投资和单项投资均有关系：$0.001x_1^2 + 0.002x_2^2 + 0.001x_1x_2$。据市场调查显示， A 项目的投资前景好于 B 项目，因此希望 A 项目的投资额不小 B 项目。试问应该如何在 A、B 两个项目之间分配投资，才能既使年利润最大，又使风险损失为最小？

该问题是一个非线性多目标规划问题，将它用数学语言描述出来，就是：求 x_1、x_2，使：

$$\max f_1(x_1,x_2)=0.60x_1+0.70x_2$$

$$\min f_2(x_1,x_2)=0.001x_1^2+0.002x_2^2+0.001x_1x_2$$

而且满足：

$$\begin{cases} x_1+x_2=1000 \\ -x_1+x_2\le 0 \\ x_1,x_2\ge 0 \end{cases}$$

求解程序如下：

① 首先编辑目标函数 M 文件 ff13.m

```
function f=ff13(x)
f(1)=-0.6*x(1)-0.7*x(2);
f(2)=0.001*x(1)^2+0.002*x(2)^2+0.001*x(1)*x(2);
```

② 按给定目标取：

```
goal=[-625,875];
weight=[-625,875];
```

③ 给出：

```
x0=[200,200];
A=[-1,1];
b=0;
Aeq=[1,1];
beq=1000;
lb=zeros(2,1);
```

④ 调用 fgoalattain 函数：

```
[x,fval,attainfactor,exitflag]=fgoalattain(@ff13,x0,goal,weight,A,b,Aeq,beq,lb,[])
```

运行后，输出结果为：

```
x =
  750.0000    250.0000

fval =
```

```
 -625.0000    875.0000

attainfactor =
 -5.4254e-016

exitflag =
       1
```

第六章　目标规划

在前面看到线性规划有着完整的理论与求解方法，也有很好的应用，但是线性规划也有很大的局限性：

1. 只能处理单目标问题，而实际中经常碰到多目标问题，而且目标与约束是可以转化的；
2. 约束都是刚性条件，需要严格满足；而实际情况中，并不是这样严格，而且过于严格可能导致无解，不利于我们做出决策；
3. 约束或者目标看成同等重要，而实际中各个目标可能有重要性的差别；
4. 给出的是最优解，而许多实际问题之需要找到满意解即可。

例如，在企业安排生产问题中，既希望利润高，又要消耗低，还要考虑市场上产品的销路等等。当然，这些目标之间往往是相互矛盾的，要追求利润最大，通常消耗便不可能最低。而且目标具有“柔性”，能否构造这样一个数学模型，其结果，即使利润尽量地大，同时使消耗尽量地低，销路尽量地好呢?

目标规划 goal　programming 便用来处理这样的问题。

第一节　目标规划的数学模型

先来看一个引例。

例 1 多目标生产计划问题　　某工厂计划用所拥有的三种资源生产代号为 A、B 的两种产品，原材料资源可供量为 90 吨，使用专用设备台时最多为 200 台时，劳动力 300 个；生产单位产品 A 需用原材料 2.5 吨，设备台时 4 个和劳动力 3 个，产品 B 则需用原料 1.5 吨，设备台时 5 个和劳动力 10 个。扣除成本，每单位产品 A、B 分别可获利 7(百元)和 12(百元)，求一个生产计划，使获利最大。

问题求解

据题设，设 x_1，x_2 依次表示产品 A 和 B 的生产量，容易得到其线性规划基础模型为

$$\max f = 7x_1 + 12x_2$$

$$\text{s.t.} \quad 2.5x_1 + 1.5x_2 \le 90$$

$$4x_1 + 5x_2 \le 200$$

$3x_1 + 10x_2 \le 300$

$x_1, x_2 \ge 0$

可求出其绝对最优解为 x_1=20，x_2=24，最大利润值为 428(百元)。

现增加以下考虑：

1. 上述结果并未考虑市场信息和资源的可塑性条件，仅仅根据现有生产能力和固定不变的产品价格求得的，因而是脱离实际的“理想化”方案。依据市场调查和生产能力，厂长认为上述利润指标不易达到，决定降低为 420(百元)，当然力求超过。

2. 根据市场调查和预测，产品 B 开始出现滞销现象，随着市场需求的改变，预测两种产品的需求量比例大致为 1:1，而目前的产品比例失调，有待调整。

3. 根据原材料市场信息，这种原材料的市场价格下跌，而所生产的产品价格基本稳定，故决策者希望尽量将原材料转化为产品，即希望原料要全部用掉。但按原生产计划看，原材料将有剩余(4 吨)。因此，尽可能将原材料全部转化为利润成为一个重要的生产规划指标。

为了叙述方便，先来考虑单利润指标情况。实现利润 420 是决策者的希望，但在计划具体实施后，由于各方面因素的制约，完全有可能达不到，也完全可能超过该指标，换句话说，可能实现的利润指标和规定的利润指标完全可能不一致而产生某一差距。我们称这个差距为**偏差变量**，记以 d 。规定 $d \ge 0$ 。从决策者的心理和要求来分析，使之绝对满意可以做不到，但他总希望将来得到的实际利润与规定的指标值之间偏差量愈小愈好，这就“等价地”表出了他希望利润值达到 420 的目标。当然，他所希望的是未达规定指标的实际值与规定值的偏差量越小越好。

由于可能达到并超过，也可能达不到，我们引入下述符号：d^+——表示超出指标的偏差变量，称为**正偏差变量**。d^-——表示未达指标的偏差变量，称为**负偏差变量**。自然规定 $d^+, d^- \ge 0$ 。显然，偏差变量 d^+， d^- 的取值有且仅有下述三种情形：

i)超额完成指标时， $d^+ > 0$， $d^- = 0$；

ii)未能完成指标时， $d^+ = 0$， $d^- < 0$；

iii)恰好完成指标时， $d^+ = 0$， $d^- = 0$。

有了偏差变量的概念，上述利润指标就可以比较灵活地进行表示了。事实上，决策者的目标是利润达到或超过 420。因此，他所希望的自然是 $d^+ > 0$。但实际中完全可能 $d^- < 0$，这是决策者所不希望出现的，而一但出现 $d^- > 0$，也希望 d^- 尽可能地小。因此，决策者最关心的是 d^- 达到最小，故此时的目标函数可表示

为

$\min d^-$

这样，我们把目标函数写成了偏差变量的函数。注意，例子中原来的目标函数显然不再成为目标规划的目标函数，由于它在目标规划中只是问题要达到的目标之一，因而也成了一个约束条件。事实上，作为目标之一的利润值已被限制(约束)在 420 百元，用偏差变量很容易将它表成为

$7x_1+12x_2+d^- - d^+ = 420$

它自然是约束条件，而且确切地表出了目标利润应为 420 百元这一约束。事实上，当达不到 420 百元时，由于 $d^+=0$，从而 $d^-=420-\left(7x_1+12x_2\right)>0$，即，$d^+=0$，$d^-<0$，恰好说明利润指标未达要求；当超过 420 百元时，由于 $7x_1+12x_2>420$，$d^-=0$，故 $d^+=\left(7x_1+12x_2\right)-420>0$；而当利润恰为 420 百元时，因 $d^- - d^+=0$，故有 $d^-=d^+=0$。

由于这一约束条件是目标规划的目标之一的约束要求，故又称为**目标约束**，其特点是带有偏差变量的等式约束。凡非目标约束的约束条件统称为**系统约束**或刚性约束。也相应地称目标约束为柔性约束，这主要是因为这种约束较刚性约束来的灵活。

至此，我们可把上述单利润指标的规划问题写成如下形式:

$\min d^-$

$7x_1+12x_2+d^- - d^+ = 420$

称这种规划模式为目标规划模式。其特点主要两条，其一是目标函数是各目标的偏差变量的函数，其二是约束条件中含有目标约束条件。

有了上面关于单指标目标规划的构模原理，我们来讨论具有三个指标情形的例 1 该如何构模。

将上述偏差变量给以下标 1 表示利润指标对应的偏差变量：

$\min d_1^-$

$7x_1+12x_2+d_1^- - d_1^+ = 420$

现考虑原材料要求的指标。记 d_2 表示原材料偏差变量，则 d_2^- 表示未用量，d_2^+ 表示超用量，由于希望 90 吨原料全部用完，既不希望有余也不希望超支，故目标函数应为

$\min\left(d_2^- + d_2^+\right)$

相应的目标约束完全类似利润指标情形，应为

$2.5x_1+1.5x_2+d_2^- - d_2^+ = 90$

又设d_3表示产品比例指标偏差变量。即以d_3表示产品 A 和 B 之产量差距，则由于要求两种产品的产量尽可能达到或接近 1:1，故有目标函数

$$\min\left(d_3^- + d_3^+\right)$$

其对应的约束条件可写为

$$x_1 - x_2 + d_3^- - d_3^+ = 0$$

这是一个新增加的目标约束，当两产品产量一致时，$x_1 - x_2 = 0$，从而$x_1 : x_2 = 1:1$。

我们的目的是求这三个目标的统一体的最优化方案，即从整体看，我们希望各个指标的偏差总和达最小，要把所涉及指标都考虑到，只能按各目标的轻重缓急分级考虑。事实上，各目标的重要程度是不同的，可以因人，因地，因时而异。比如产品的产量问题，对有的企业来讲是第一位的，而对别的企业来讲则是第二位的；同一企业此时此地产量第一而彼时彼地可能产量就放在第二位甚至于第三位，等等。这就需要决策者或决策集团根据各目标的重要程度，科学地予以排队。

我们规定p_1表示第一位重要，p_2表示第二位重要，即满足$p_1 >> p_2$。称p_k为**优先因子**。如第一位重要的目标是要求超额完成利润指标，则赋予它优先因子p_1，其在整个问题的目标函数中表为

$$p_1 d_1^-,$$

列为第一优先级；其次要目标是要求恰好用完原材料，赋予优先因子p_2，列为第二优先级表为

$$p_2\left(d_2^- + d_2^+\right),$$

最后是要求产品产量比例达 1:1 的目标，赋予优先因子p_3，列为第三优先级，在整体目标函数中表为

$$p_3\left(d_3^- + d_3^+\right)$$

于是整个问题的目标函数表为

$$\min z = p_1 d_1^- + p_2\left(d_2^- + d_2^+\right) + p_3\left(d_3^- + d_3^+\right)$$

从而三个目标的例 1 的目标规划模型便是：

$$\min z = p_1 d_1^- + p_2\left(d_2^- + d_2^+\right) + p_3\left(d_3^- + d_3^+\right)$$

$$\text{s.t.}\quad 4x_1 + 5x_2 \le 200$$

$$3x_1 + 10x_2 \le 300$$

$$7x_1 + 12x_2 + d_1^- - d_1^+ = 420$$

$$2.5x_1 + 1.5x_2 + d_2^- - d_2^+ = 90$$

$$x_1 - x_2 + d_3^- - d_3^+ = 0$$

$$x_1, x_2 \ge 0，\quad d_i^-, d_i^+ \ge 0, i = 1,2,3$$

目标规划的一般模型：

设$x_j(j = 1,\Lambda, n)$是目标规划的决策变量，共有m个约束是刚性约束，有l个柔性目标约束，其目标规划约束的偏差为$d_i^-, d_i^+(i = 1,\Lambda, l)$，设有$q$个优先级别，分别为$p_1,\Lambda, p_q$，在同一个优先级别$p_k$中，不同的目标有不同的权重（重要性的区别），分别记为$\varpi_{kj}^+, \varpi_{kj}^-$，目标规划模型的一般数学表达式为：

$$\min z = \sum_{k=1}^{q} p_k \sum_{j=1}^{l} \left(\varpi_{kj}^- d_j^- + \varpi_{kj}^+ d_j^+\right)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \le (=, \ge) b_i，\quad i = 1,\Lambda, m$$

$$\sum_{j=1}^{n} c_{ij} x_j + d_i^- - d_i^+ = g_i，\quad i = 1,\Lambda, l$$

$$x_j \ge 0，\quad j = 1,\Lambda, n$$

$$d_i^-, d_i^+ \ge 0，\quad i = 1,\Lambda, l$$

确定目标函数的一般原则是：

对于约束：都是固定的形式，在刚性约束中添加$d_i^- - d_i^+$后都取等号。

对于目标函数：

1） 如果要求恰好达到目标，即要求目标的正负偏差都尽可能的小，则取$\min z = f\left(d_k^- + d_k^+\right)$；

2） 如果要求超过指标值，即要求目标的正偏差不限，而负偏差越小越好，则取$\min z = f\left(d_k^-\right)$；

3） 如果要求超过指标值，即要求目标的负偏差不限，而正偏差越小越好，则取$\min z = f\left(d_k^+\right)$；

4） 再根据各目标的优先级赋予相应的优先因子和加权系数。

第二节 目标规划模型的求解

对目标规划模型（我们主要讨论线性模型）的求解方法有两种，一种是直接使用单纯形法，结合优先级别 p_k 来判断判别数的情况，我们主要介绍另一种方法—序贯式算法。

序贯式算法的思想是：将目标规划模型按照各目标的优先等级次序，依次分解为若干个单目标的规划问题，按级别高低依次求解，优先级高的最优解求出后，作为该目标偏差的上界添加到低级别问题中作为约束条件，再对低级规划问题求解。

我们可以采用 LINGO 软件实现这个算法。

第一节的问题 LINGO 程序编写如下：

```
model:
title 目标规划;
sets:
  Level/1..3/: P, z, Goal;
  Variable/1..2/: x;
  H_Con_Num/1..2/: b;
  S_Con_Num/1..3/: g, dplus, dminus;
  H_Cons(H_Con_Num, Variable): A;
  S_Cons(S_Con_Num, Variable): C;
  Obj(Level, S_Con_Num): Wplus, Wminus;
endsets
data:
  P= ? ? ?;
  Goal = ? ? 1000;
  b = 200 300;
  g=  420 90 0;
  A = 4 5
      3 10;
  C = 7   12
      2.5 1.5
       1  -1  ;
  Wplus =  0 0 0
           0 1 0
           0 0 1;
  Wminus = 1 0 0
           0 1 0
           0 0 1;
enddata

min=@sum(Level: P * z);
@for(Level(i):
```

```
  z(i)=@sum(S_Con_Num(j): Wplus(i,j)*dplus(j))
      +@sum(S_Con_Num(j): Wminus(i,j)*dminus(j)));
@for(H_Con_Num(i):
  @sum(Variable(j): A(i,j) * x(j)) <= b(i));
@for(S_Con_Num(i):
  @sum(Variable(j): C(i,j)*x(j))
    + dminus(i) - dplus(i) = g(i);
);
@for(Level(i) | i #lt# @size(Level):
  @bnd(0, z(i), Goal(i));
);
    End
```

需要求解三次才能得到最后解。

第一次求解：P(1),P(2),P(3)分别输入1，0，0，Goal（1），Goal（2）输入1000，1000（较大的值即可，让约束不起作用），得到最优偏差为0.

第二次求解：P(1),P(2),P(3)分别输入0，1，0，Goal（1），Goal（2）输入0（为第一次的最优偏差），1000，得到最优偏差为0.

第三次求解：P(1),P(2),P(3)分别输入0，0，1，Goal（1），Goal（2）输入0，0（为第二次的最优偏差），得到最优偏差为1.538462，进而求得最优利润。