

1、运筹学的内容，主要包括最优化理论，但是熊伟的那本书中主要是线性最优化，推荐的软件是 winQSB 教学。在研究生课程中的最优化理论和 matlab 内容主要是非线性的优化，用的是 matalab。其实，最优化理论中，有专门的软件 lingo，试了一试，还真不错。

2、通过例子，简单的把这个软件怎么应用的演示一下。首先，lingo 的软件的下载和安装，跳过。接着，求解一个线性的优化模型吧，看图：

$$\min Z = \sum_{j=1}^{10} x_j$$
$$\begin{cases} 2x_1 + 2x_2 + x_3 + x_4 + x_5 \geq 1000 \\ x_1 + 2x_3 + x_4 + 4x_6 + 3x_7 + 2x_8 + x_9 \geq 1000 \\ x_2 + 2x_4 + 3x_5 + x_7 + 2x_8 + 4x_9 + 5x_{10} \geq 1000 \\ x_j \geq 0, j = 1, 2, \dots, 10 \end{cases}$$

这个下料的问题，用 lingo 求解的时候，从几个方面考虑：变量的设置，目标函数的设置和约束的设置。

(1) 变量一般是用 set 和 endset 来完成的，如下：

```
sets:
! r: 5个措施;
row/1..5/:r;
! Q:10个路段;
col/1..10/:c;
! ;
links(row,col):x,co,e,v,s;
endsets
```

题目意思中可以看出有 10 个方案，因此应该这样设置变量

```
sets:
pro/1..10/:p; ! Q:10个方案;
endsets
```

(2) 目标函数是有内置的，如下：

```
!MIN=@sum( links(i,j):x(i,j)*co(i,j));
```

这里题目的意思是求最小值，因此可以用内置函数来完成

```
MIN=@sum(pro(i):p(i));
```

(3) 约束一般也是用循环函数来完成的，如下：

```
@for( xb2(i): @sum( xb1(j): a(i,j)*x(j) ) +c(i)<= b(i););
```

根据题目的意思，可以看出，应该有三个约束。有的人会问，为什么大于 0 的约束，不考虑进去，那是软件已经默认进去的，就是说默认是大于 0 的。

$$\begin{aligned} \min Z &= \sum_{j=1}^{10} x_j \\ \begin{cases} 2x_1 + 2x_2 + x_3 + x_4 + x_5 \geq 1000 \\ x_1 + 2x_3 + x_4 + 4x_6 + 3x_7 + 2x_8 + x_9 \geq 1000 \\ x_2 + 2x_4 + 3x_5 + x_7 + 2x_8 + 4x_9 + 5x_{10} \geq 1000 \\ x_j \geq 0, j = 1, 2, \dots, 10 \end{cases} \end{aligned}$$

那么，约束可以写成

```
2*x(1)+2*x(2)+x(3)+x(4)+x(5)>=1000
x(1)+2*x(3)+x(4)+4*x(6)+2*x(7)+2*x(8)+x(9)>=1000
x(2)+2*x(4)+3*x(5)+x(7)+2*x(8)+4*x(9)+5*x(10)>=1000
```

最后，根据 lingo 的格式，得到模型，然后求解，看代码：

```
model:

title ninekwlltest;

sets:
pro/1..10/:x; ! Q:10个方案;
endsets

@for( pro( i ) : @GIN( x( i ) ));
```

```
MIN=@sum(pro(i):x(i));
```

```
2*x(1)+2*x(2)+x(3)+x(4)+x(5)>=1000;
```

```
x(1)+2*x(3)+x(4)+4*x(6)+2*x(7)+2*x(8)+x(9)>=1000;
```

```
x(2)+2*x(4)+3*x(5)+x(7)+2*x(8)+4*x(9)+5*x(10)>=1000;
```

```
end
```

运行结果，如下图：

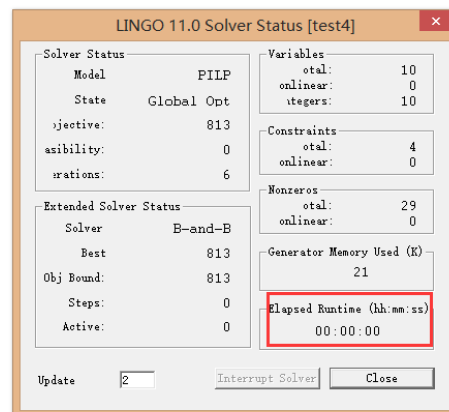
Global optimal solution found.
Objective value: 813.0000
Objective bound: 813.0000
Infeasibilities: 0.000000
Extended solver steps: 0
Total solver iterations: 6

813.0000

Model Title: ninekwlltest

Variable	Value	Reduced Cost
X(1)	500.0000	1.000000
X(2)	0.000000	1.000000
X(3)	0.000000	1.000000
X(4)	0.000000	1.000000
X(5)	0.000000	1.000000
X(6)	63.00000	1.000000
X(7)	0.000000	1.000000
X(8)	0.000000	1.000000
X(9)	250.0000	1.000000
X(10)	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	813.0000	-1.000000
2	0.000000	0.000000
3	2.000000	0.000000
4	0.000000	0.000000



我去，比 winqsb 快多了，这个死软件死机了，一直再运算。