

第 1 讲 LINGO 软件入门

司守奎

烟台市, 海军航空大学

Email: sishoukui@163.com

1.1 初识 LINGO 程序

LINGO 程序书写实际上特别简捷, 数学模型怎样描述, LINGO 语言就对应地怎样表达。首先介绍两个简单的 LINGO 程序。

例 1.1 求解如下的线性规划问题:

$$\begin{aligned} \max \quad & z = 72x_1 + 64x_2 \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 50, \\ 12x_1 + 8x_2 \leq 480, \\ 3x_1 \leq 100, \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

LINGO 求解程序如下

```
max=72*x1+64*x2;
```

```
x1+x2<=50;
```

```
12*x1+8*x2<=480;
```

```
3*x1<=100;
```

说明: LINGO 中默认所有的变量都是非负的, 在 LINGO 中就不需写出对应的约束。

例 1.2 抛物面 $z = x^2 + y^2$ 被平面 $x + y + z = 1$ 截成一椭圆, 求原点到这椭圆的最短距离。

该问题可以用拉格朗日乘子法求解。下面我们把问题归结为数学规划模型, 用 LINGO 软件求解。

设原点到椭圆上点 (x, y, z) 的距离最短, 建立如下的数学规划模型:

$$\begin{aligned} \min \quad & \sqrt{x^2 + y^2 + z^2} \\ \text{s.t.} \quad & \begin{cases} x + y + z = 1, \\ z = x^2 + y^2. \end{cases} \end{aligned}$$

LINGO 求解程序如下:

```
min=(x^2+y^2+z^2)^(1/2);
```

```
x+y+z=1;
```

```
z=x^2+y^2;
```

```
@free(x); @free(y);
```

说明: LINGO 中默认所有变量都是非负的, 这里 x, y 的取值是可正可负的, 所以使用 LINGO 函数 free。

例 1.3 求解如下的数学规划模型:

$$\begin{aligned} \min & \sqrt{\sum_{i=1}^{100} x_i^2} \\ \text{s.t.} & \begin{cases} \sum_{i=1}^{100} x_i = 1, \\ x_{100} = \sum_{i=1}^{99} x_i^2. \end{cases} \end{aligned}$$

用 LINGO 求解上述数学规划问题，使用集合和函数比较方便，使用集合的目的是为了定义向量，集合使用前，必须先定义；LINGO 程序中的标量不需要定义，直接使用即可。

sets:

var/1..100/:x;

endsets

min=@sqrt(@sum(var(i):x(i)^2));

@sum(var(i):x(i))=1;

x(100)=@sum(var(i)|i#le#99:x(i)^2);

@for(var(i)|i#le#99:@free(x(i)));

说明：如果不使用集合和函数，全部使用标量 x_1, x_2, \dots, x_{100} ，最后一个约束就要写 99 遍，@free(x1); ...; @free(x99)。

1.2 LINGO 软件的界面介绍

1.2.1 LINGO 的模型窗口

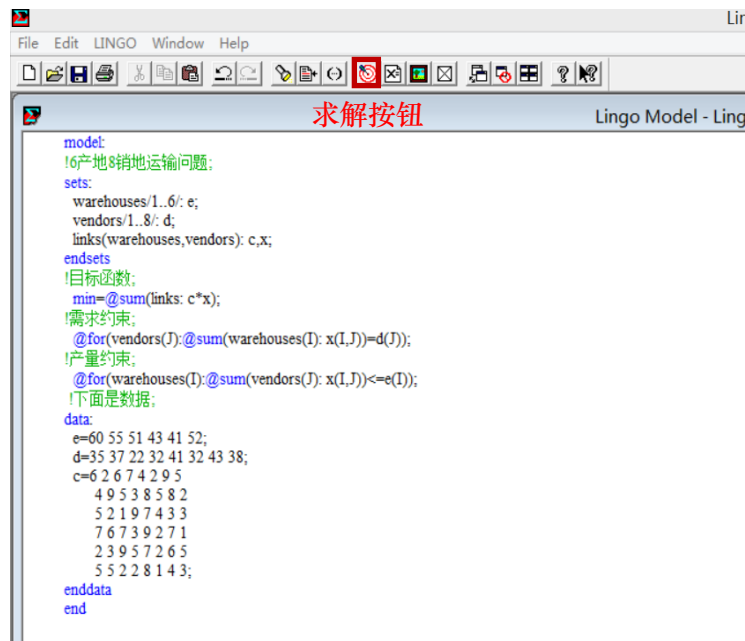


图 1.1 LINGO 的模型窗口

模型窗口输入格式要求：

- (1) LINGO 的数学规划模型包含目标函数、决策变量、约束条件三个要素。
- (2) 在 LINGO 程序中，每一个语句都必须要用一个英文状态下的分号结束，一个语句可以分几行输入。
- (3) LINGO 的注释以英文状态的 ! 开始，必须以英文状态下的分号结束；
- (4) LINGO 的变量不区分字母的大小写，必须以字母开头，可以包含数字和下划线，

不超过 32 个字符。

(5) LINGO 程序中，只要定义好集合后，其他语句的顺序是任意的。

(6) LINGO 中的函数以“@”开头。

(7) LINGO 程序默认所有的变量都是非负的。

(8) LINGO 程序中“>或<”号与“≥或≤”号功能相同。

2.LINGO 建模时需要注意的几个基本问题

(1) 尽量使用实数变量，减少整数约束和整数变量。

(2) 模型中使用的参数数量级要适当，否则会给出警告信息，可以选择适当的单位改变相对尺度。

(3) 尽量使用线性模型，减少非线性约束和非线性变量的个数，同时尽量少使用绝对值、符号函数、多变量求最大最小值、取整函数等非线性函数。

(4) 合理设定变量上下界，尽可能给出初始值。

2.2.2 LINGO 的求解器运行状态窗口

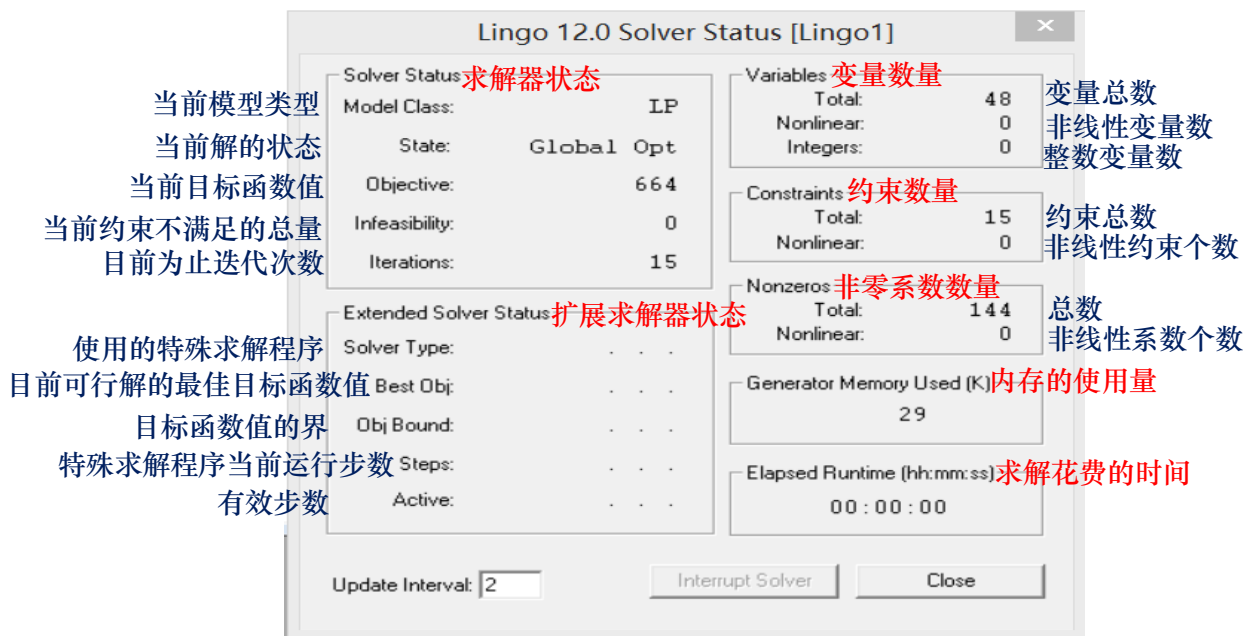


图 1.2 LINGO 的求解器状态窗口

1.求解器状态框

“当前解的状态”有如下几种：

Global Optimum 全局最优解

Local Optimum 局部最优解

Feasible 可行解

Infeasible 不可行解

Unbounded 无界解

Interrupted 中断

Undetermined 未确定

2.扩展求解器状态

“使用的特殊求解程序”有如下几种：

B-and-B 分支定界算法

Global 全局最优求解程序

Multistart 用多个初始点求解的程序

3.LINGO 求解的参数设置

LINGO10 软件管理的内存最大为 2G，如果你的计算机内存是 4G 的话，LINGO 的内存就设置为 2G，你的计算机内存是 8G 的话，也要设置成 2G。

LINGO 内存的设置是依次选择菜单 LINGO(第 3 个主菜单)→Options...→Model Generator。如图 1.3 所示红色标注的部分。

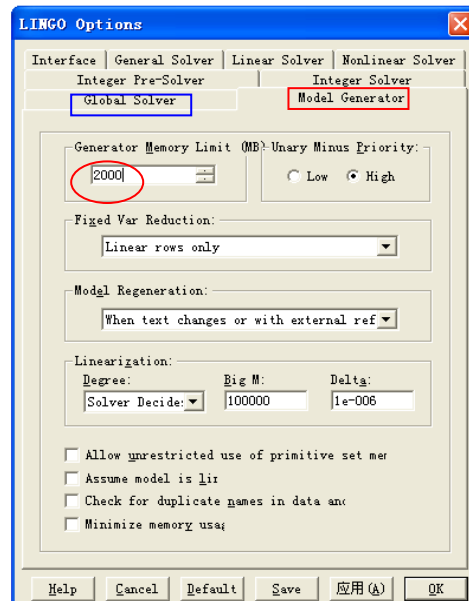


图 1.3 求解器 Options 的一些设置

如果模型是非线性模型，且欲求全局最优解，要把求解器设置成“Global”。进入图 1.3 中蓝色“Global Solver”后，在“Use Global Solver”前面打上“√”，设置完成后，要关闭 LINGO 软件，再重新启动 LINGO 软件。

1.2.3 一般线性规划问题的影子价格与灵敏度分析

如果没有学过运筹学，本部分就不要看了。

以例 1.1 的线性规划模型为例。

$$\begin{aligned} \max \quad & z = 72x_1 + 64x_2 \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 50, \\ 12x_1 + 8x_2 \leq 480, \\ 3x_1 \leq 100, \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

1. 影子价格

要进行灵敏度分析，必须选择如图 1.4 所示的画圈的选项，依次选择下列菜单 LINGO →Options...→General Solver 下 Dual Computations 选择 Prices。

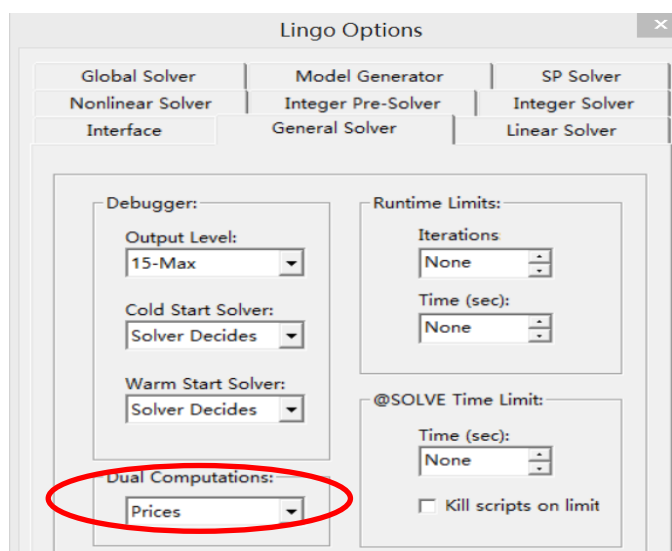


图 1.4 LINGO Options 设置

勾选了 Prices 选项后，运行 LINGO 程序，输出结果窗口中包含灵敏度分析，如图 1.5 所示。

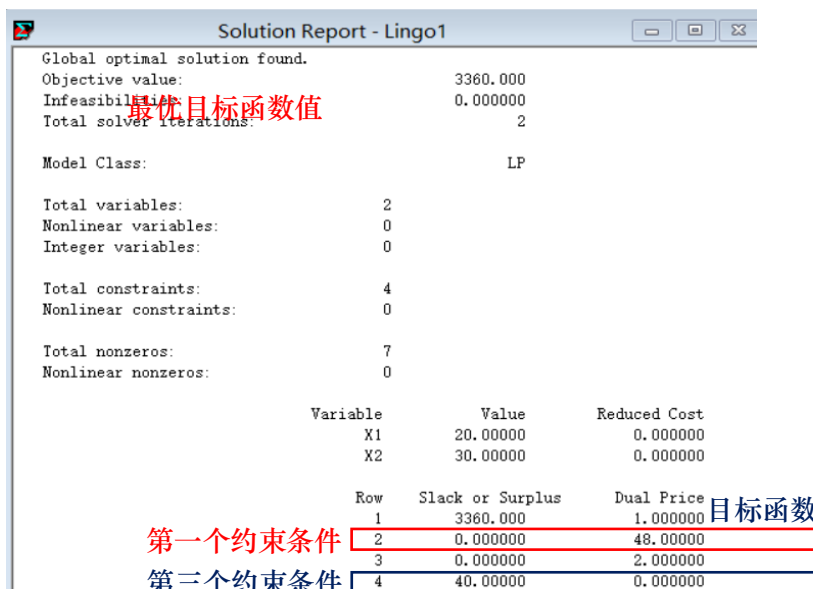


图 1.5 灵敏度分析

从结果可知，目标函数的最优值为 3360，决策变量 $x_1 = 20, x_2 = 30$ 。

(1) reduced cost 值对应于单纯形法计算过程中各变量的检验数。

(2) 图 1.5 中红色方框表示第二个约束条件，Slack or Surplus 值为 0 表示该约束松弛变量为 0，约束等号成立，为紧约束或有效约束。蓝色方框表示第三个约束松弛变量为 40，不等号成立，有剩余。

(3) Dual Price 对应影子价格，红色方框表示当第一个约束条件右端常数项增加 1 个单位，即由 50 变为 51 时，目标函数值增加 48，即约束条件 1 所代表的资源的影子价格。蓝色方框表示，第三个约束条件右端常数项增加 1 个单位时，目标函数值不变。

2. 确保最优基不变的系数变化范围

如果想要研究目标函数的系数和约束右端常数项系数在什么范围变化(假定其他系数保持不变)时，最优基保持不变。此时需要首先勾选图 1.6 所示的选项。

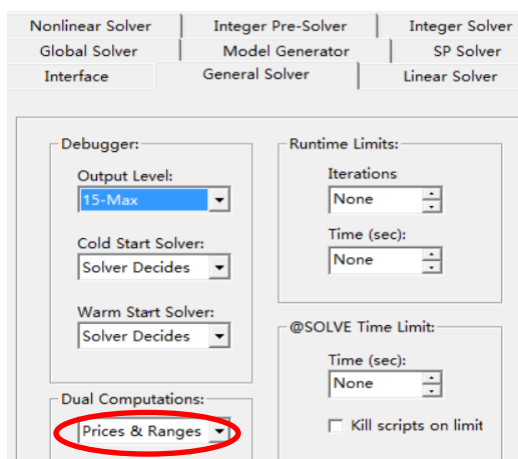


图 1.6 LINGO Options 对话框

此时不需要重新运行程序，关闭输出窗口，从菜单命令 LINGO 中选中“Range”，即可得到如下输出窗口，如图 1.7。

Objective Coefficient Ranges:			
Variable	Current Coefficient	Allowable Increase	Allowable Decrease
X1	72.00000	24.00000	8.000000
X2	64.00000	8.000000	16.00000
Righthand Side Ranges:			
Row	Current RHS	Allowable Increase	Allowable Decrease
2	50.00000	10.00000	6.666667
3	480.0000	53.33333	80.00000
4	100.0000	INFINITY	40.00000

图 1.7 灵敏度分析范围变化输出窗口

(1)Objective Coefficient Ranges 一栏反映了目标函数中决策变量的价值系数,可以看到 x_1 的系数是 72, x_2 的系数是 64, 说明 x_1 要想确保当前最优基不变, 在其他系数不变的情况下, x_1 系数的变化范围为 (64,96), 当 x_1 的系数在这个范围内变化时, 最优解不变, 但是最优目标函数值发生变化, 同样, x_2 系数的变化范围为 (48,72)。

(2) Righthand Side Ranges 一栏反映了约束条件右端代表资源系数的常数项, 可见第一个约束右端常数项在 (43.333333,60)变化时, 最优基不变, 但是最优解发生变化, 目标函数值也相应地发生变化。由于第三个约束松弛变量为 40, 有剩余, 可见无论再如何增加该资源, 只会使剩得更多, 对解没有影响, 但是如果减少量超过 40, 就会产生影响。

1.3 LINGO 软件的基本语法

1.3.1 集合

集合部分的语法为

sets:

集合名称 1/成员列表 1/: 属性 1_1, 属性 1_2, ..., 属性 1_n1;

集合名称 2/成员列表 2/: 属性 2_1, 属性 2_2, ..., 属性 2_n2;

派生集合名称 (集合名称 1, 集合名称 2): 属性 3_1, ..., 属性 3_n3;

endsets

例 1.4

sets:

product/A B/;

- ```

machine/M N/;
week/1..2/;
allowed(product,machine,week):x;
endsets

```
- 1.3.2 数据
- 数据部分的语法为
- ```

data:
属性 1=数据列表;
属性 2=数据列表;
enddata

```
- 在数据段中，=右侧不能出现运算，故不能出现分数，如 3/8，这里的/表示除法运算。
- 1.3.3 计算段
- 计算段部分不能含有变量，必须是已知数据的运算。
- ```

calc:
b=0;
a=a+1;
endcalc

```
- 1.3.4 目标函数、约束条件
- 目标函数和约束条件这里就不具体给出了，下面通过具体例子给出。

## 1.4 LINGO 函数

### 1.4.1 算术运算符

^ 乘方  
\* 乘  
/ 除  
+ 加  
- 减

### 1.4.2 逻辑运算符

在 LINGO 中，逻辑运算符主要用于集循环函数的条件表达式中，来控制在函数中哪些集成员被包含，哪些被排斥。在创建稀疏集时用在成员资格过滤器中。

LINGO 具有 9 种逻辑运算符

#not# 否定该操作数的逻辑值，#not# 是一个一元运算符。  
#eq# 若两个运算数相等，则为 true；否则为 false。  
#ne# 若两个运算符不相等，则为 true；否则为 false。  
#gt# 若左边的运算符严格大于右边的运算符，则为 true；否则为 false。  
#ge# 若左边的运算符大于或等于右边的运算符，则为 true；否则为 false。  
#lt# 若左边的运算符严格小于右边的运算符，则为 true；否则为 false。  
#le# 若左边的运算符小于或等于右边的运算符，则为 true；否则为 false。  
#and# 仅当两个参数都为 true 时，结果为 true；否则为 false。  
#or# 仅当两个参数都为 false 时，结果为 false；否则为 true。

### 1.4.3 关系运算符

在 LINGO 中，关系运算符主要是被用在模型中来指定一个表达式的左边是否等于、小于等于、或者大于等于右边，形成模型的一个约束条件。关系运算符与逻辑运算符#eq#、#le#、#ge#截然不同，逻辑运算符仅仅判断一个关系是否被满足，满足为真，不满足为假。

LINGO 有三种关系运算符：“=”、“<=”和“>=”。LINGO 中还能用“<”表示小于等于关系，“>”表示大于等于关系。LINGO 并不支持严格小于和严格大于关系运算符。

### 1.4.4 数学函数

LINGO 提供了大量的标准数学函数

@abs(x)返回 x 的绝对值。  
@sin(x)返回 x 的正弦值，x 采用弧度制。  
@cos(x)返回 x 的余弦值。  
@tan(x)返回 x 的正切值。

@exp(x)返回常数 e 的 x 次方。

@log(x)返回 x 的自然对数。

@lgm(x)返回 x 的 gamma 函数的自然对数。

@mod(x,y)返回 x 除以 y 的余数。

@sign(x)如果  $x < 0$  返回 -1, 当  $x > 0$  时返回 1, 当  $x = 0$  时返回 0。

@floor(x)返回 x 的整数部分。当  $x \geq 0$  时, 返回不超过 x 的最大整数; 当  $x < 0$  时, 返回不低于 x 的最大整数。

@smax(x1,x2,...,xn)返回  $x_1, x_2, \dots, x_n$  中的最大值。

@smin(x1,x2,...,xn)返回  $x_1, x_2, \dots, x_n$  中的最小值。

#### 1.4.5 变量界定函数

变量界定函数实现对变量取值范围的附加限制, 共 4 种

@bin(x)限制 x 为 0 或 1;

@bnd(L,x,U)限制  $L \leq x \leq U$ ;

@free(x)取消对变量 x 的默认下界为 0 的限制, 即 x 可以取任意实数;

@gin(x)限制 x 为整数。

在默认情况下, LINGO 规定变量是非负的, 也就是说下界为 0, 上界为  $+\infty$ 。@free 取消了默认的下界为 0 的限制, 使变量也可以取负值。@bnd 用于设定一个变量的上下界, 它也可以取消默认下界为 0 的约束。

#### 1.4.6 集循环函数

@for: 该函数用来产生对集成员的约束。

@sum: 该函数返回遍历指定的集成员的一个表达式的和。

@min 和 @max: 返回指定的集成员的一个表达式的最小值或最大值。

#### 1.4.7 其他函数

##### (1) 函数@TABLE

该函数以表格形式输出与集合和集合的属性相关的数据, 并且只能在数据段 (DATA) 中使用。目前该函数仅用于将数据输出到结果报告窗口或文本文件中, 而不能输出到数据库或电子表格 (EXCEL) 文件中。也就是说, 只能输出到 @TEXT 函数, 而不能输出到 @OLE 和 @ODBC 函数。

##### (2) 函数@WRITE和@WRITEFOR

在LINGO10.0中, 函数@WRITE和@WRITEFOR这两个函数是为了方便用户控制输出格式, 但它们还可以出现在计算段(CALC)随时输出中间结果, 并且不需要使用@TEXT 函数, 输出的结果也是被定向到缺省的输出设备 (通常就是标准的报告窗口)。如果希望改变缺省的输出设备, 可以采用@DIVERT函数。

注: LINGO 中矩阵数据是逐行存储的, Matlab 中数据是逐列存储的。

### 1.5 LINGO 的数据传递

下面我们以 LINGO 帮助中运输问题的 3 个例程来说明 LINGO 软件的数据传递。

例 1.5 使用 LINGO 软件计算 6 个产地 8 个销地的最小费用运输问题。单位商品运价如表 1.1 所示。

表 1.1 单位商品运价表

| 单位运价<br>产地 \ 销地 |                |                |                |                |                |                |                |                | 产量 |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----|
|                 | B <sub>1</sub> | B <sub>2</sub> | B <sub>3</sub> | B <sub>4</sub> | B <sub>5</sub> | B <sub>6</sub> | B <sub>7</sub> | B <sub>8</sub> |    |
| A <sub>1</sub>  | 6              | 2              | 6              | 7              | 4              | 2              | 5              | 9              | 60 |
| A <sub>2</sub>  | 4              | 9              | 5              | 3              | 8              | 5              | 8              | 2              | 55 |
| A <sub>3</sub>  | 5              | 2              | 1              | 9              | 7              | 4              | 3              | 3              | 51 |
| A <sub>4</sub>  | 7              | 6              | 7              | 3              | 9              | 2              | 7              | 1              | 43 |
| A <sub>5</sub>  | 2              | 3              | 9              | 5              | 7              | 2              | 6              | 5              | 41 |
| A <sub>6</sub>  | 5              | 5              | 2              | 2              | 8              | 1              | 4              | 3              | 52 |



|    |    |    |    |    |    |    |    |    |  |
|----|----|----|----|----|----|----|----|----|--|
| 销量 | 35 | 37 | 22 | 32 | 41 | 32 | 43 | 38 |  |
|----|----|----|----|----|----|----|----|----|--|

### 1.数学模型

设  $x_{ij}$  ( $i=1,2,\dots,6; j=1,2,\dots,8$ ) 表示产地  $A_i$  运到销地  $B_j$  的量,  $c_{ij}$  表示产地  $A_i$  到销地  $B_j$  的单位运价,  $d_j$  表示销地  $B_j$  的需求量,  $e_i$  表示产地  $A_i$  的产量, 建立如下线性规划模型

$$\begin{aligned} \min \quad & \sum_{i=1}^6 \sum_{j=1}^8 c_{ij} x_{ij}, \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^6 x_{ij} = d_j, & j=1,2,\dots,8, \\ \sum_{j=1}^8 x_{ij} \leq e_i, & i=1,2,\dots,6, \\ x_{ij} \geq 0, & i=1,2,\dots,6; j=1,2,\dots,8. \end{cases} \end{aligned}$$

### 2.LINGO 程序

例 1.5 中的目标函数表达式  $\min \sum_{i=1}^6 \sum_{j=1}^8 c_{ij} x_{ij}$  用 LINGO 语句表示为:

`min=@sum(links(i,j): c(i,j)*x(i,j));`

式中, @sum 是 LINGO 提供的内部函数, 其作用是对某个集合的所有成员, 求指定表示式的和, 该函数需要两个参数, 第一个参数是集合名称, 指定对该集合的所有成员求和; 第二个参数是一个表达式, 表示求和运算对该表达式进行, 两个参数之间用冒号“:”分隔。此处 @sum 的第一个参数是 links(i,j), 表示求和运算对派生集合 links 进行, 该集合的维数是 2, 共有 48 个成员, 运算规则是: 先对 48 个成员分别求表达式  $c(i,j)*x(i,j)$  的值, 然后求和, 相当于求  $\sum_{i=1}^6 \sum_{j=1}^8 c_{ij} x_{ij}$ , 表达式中的 c 和 x 是集合 links 的两个属性, 它们各有 48 个分量。

**注 1.1** 如果表达式中参与运算的属性属于同一个集合, 则 @sum 语句中索引 (相当于矩阵或数组的下标) 可以省略, 假如表达式中参与运算的属性属于不同的集合, 则不能省略属性的索引。例 1.5 的目标函数中的属性 c 和 x 属于同一个集合, 因而可以表示成:

`min=@sum(links: c*x);`

例 1.5 中的约束条件  $\sum_{j=1}^8 x_{ij} \leq e_i, i=1,2,\dots,6$  实际上表示了 6 个不等式, 用 LINGO 语言表示该约束条件, 语句为:

`@for(warehouses(i):@sum(vendors(j): x(i,j))<=e(i));`

语句中的 @for 是 LINGO 提供的内部函数, 它的作用是对某个集合的所有成员分别生成一个约束表达式, 它有两个参数, 两个参数之间用冒号“:”分隔; 第一个参数是集合名, 表示对该集合的所有成员生成对应的约束表达式, 上述 @for 的第一个参数为 warehouses, 它表示产地, 共有 6 个成员, 故应生成 6 个约束表达式; @for 的第二个参数是约束表达式的具体内容, 此处再调用 @sum 函数, 表示约束表达式的左边是求和, 是对集合 vendors 的 8 个成员求和, 即对表达式  $x(i,j)$  中的第二维 j 求和, 亦即  $\sum_{j=1}^8 x_{ij}$ , 约束表达式的右边是集合 warehouses 的属性 e, 它有 6 个分量, 与 6 个约束表达式一一对应。本语句中的属性分别属于不同的集合, 所以不能省略索引 i, j。

同样地，约束条件  $\sum_{i=1}^6 x_{ij} = d_j, j = 1, 2, \dots, 8$  用 LINGO 语句表示为：

```
@for(vendors(j):@sum(warehouses(i): x(i,j))=d(j));
```

(1) 程序和数据放在同一个文件

```
model:
!6 产地 8 销地运输问题;
sets:
 warehouses/1..6/: e;
 vendors/1..8/: d;
 links(warehouses,vendors): c,x;
endsets
!目标函数;
 min=@sum(links: c*x);
!需求约束;
 @for(vendors(J):@sum(warehouses(I): x(I,J))=d(J));
!产量约束;
 @for(warehouses(I):@sum(vendors(J): x(I,J))<=e(I));
!下面是数据;
data:
 e=60 55 51 43 41 52;
 d=35 37 22 32 41 32 43 38;
 c=6 2 6 7 4 2 9 5
 4 9 5 3 8 5 8 2
 5 2 1 9 7 4 3 3
 7 6 7 3 9 2 7 1
 2 3 9 5 7 2 6 5
 5 5 2 2 8 1 4 3;
enddata
end
```

注 1.2 在例 1.5 数学模型中  $C = (c_{ij})_{6 \times 8}$  是一个矩阵，即二维向量，在上述 LINGO 程序中，

属性 c 数据（LINGO 中通过 c(i,j) 引用属性的值）的排列方式为

$c(1,1), c(1,2), \dots, c(1,8), c(2,1), c(2,2), \dots, c(2,8), \dots, c(6,1), c(6,2), \dots, c(6,8)$ ;

即把矩阵的元素逐行排成一个行向量的格式，上面 LINGO 语句中为了容易看清属性 c 的数据个数，把数据排成了 6 行 8 列，中间用空格分隔（数据元素之间也可以用逗号“,”分隔），实际上可以把这 48 个元素的数据分成多少行表示都可以的，但中间不要有分号“;”，数据的末尾才能加分号“;”。

(2) 通过纯文本文件传递数据

使用 LINGO 函数 @file 从外部纯文本文件读入数据，执行一次 @file，输入 1 个记录，记录之间的分隔符为 ~。

使用 LINGO 函数 @text 把计算结果输出到外部纯文本文件中。

计算的 LINGO 程序如下：

```
model:
sets:
 warehouses/1..6/: e;
 vendors/1..8/: d;
 links(warehouses,vendors): c, x;
endsets
min=@sum(links: c*x);
@for(vendors(J):@sum(warehouses(I): x(I,J))=d(J));
@for(warehouses(I):@sum(vendors(J): x(I,J))<=e(I));
```

```

data:
 e=@file(sdata11.txt);
 d=@file(sdata11.txt);
 c=@file(sdata11.txt);
 @text(sdata12.txt)=@table(x); !把计算结果以表格形式输出到外部纯文本文件;
enddata
end

```

其中纯文本数据文件 sdata1.txt 中的数据格式如下

```

60 55 51 43 41 52~ !~是记录分割符，该第一个记录是产量;
35 37 22 32 41 32 43 38~ !该第二个记录是需求量;
6 2 6 7 4 2 9 5
4 9 5 3 8 5 8 2
5 2 1 9 7 4 3 3
7 6 7 3 9 2 7 1
2 3 9 5 7 2 6 5
5 5 2 2 8 1 4 3 !最后一个记录是单位运价;

```

### (3) 通过 Excel 文件传递数据

LINGO 通过 @OLE 函数实现与 Excel 文件传递数据，使用 @OLE 函数既可以从 Excel 文件中导入数据，也能把计算结果写入 Excel 文件。

从 Excel 文件中导入数据的调用格式如下：

属性名 1=@OLE(‘Excel 文件名’, ‘数据域名称 1’);

使用 @OLE 函数也能把计算结果写入 Excel 文件，使用格式如下

@OLE(‘Excel 文件名’, ‘数据域名称 2’)=属性名 2;

如数据域名称与属性名相同时，可以省略数据块名称。

计算的 LINGO 程序如下：

```

model:
sets:
 warehouses/1..6/: e;
 vendors/1..8/: d;
 links(warehouses,vendors): c, x;
endsets
min=@sum(links: c*x);
@for(vendors(J):@sum(warehouses(I): x(I,J))=d(J));
@for(warehouses(I):@sum(vendors(J): x(I,J))<=e(I));
data:
 e=@ole(sdata13.xls);
 d=@ole(sdata13.xls);
 c=@ole(sdata13.xls,cc); !Excel 中不允许使用域名“c”，对应的数据块定义成“cc”；
 @ole(sdata13.xls)=x;
enddata
end

```

同样的，域名列表 range\_name\_list 中的域名也可以替换为 Excel 的引用地址。

注 1.3 要注意 @ole 函数用于输出和输入之间的差异，记住

@ole( ... ) = object\_list; ↔ 输出，

object\_list = @ole( ... ); ↔ 输入。

例 1.6（续例 1.5）在例 1.5 的运输问题中，使用 Excel 文件的引用地址输入和输出数据。

model:

sets:

```

warehouses/1..6/: e;
vendors/1..8/: d;
links(warehouses,vendors): c,x;

endsets

data: !数据部分;
e=@ole(sdata14.xlsx,I1:J6);
d=@ole(sdata14.xlsx,A7:H7);
c=@ole(sdata14.xlsx,A1:H6);
@ole(sdata14.xlsx,A9:H14)=x; !把求解结果输出到Excel文件;

enddata

min=@sum(links(i,j): c(i,j)*x(i,j)); !目标函数;
@for(warehouses(i): @sum(vendors(j): x(i,j))<=e(i)); !约束条件;
@for(vendors(j): @sum(warehouses(i): x(i,j))=d(j));

end

```

其中 Excel 文件 sdata14.xlsx 中的数据内容如图 1.8 所示。求解结果的输出内容如图 1.9 所示。

|   | A  | B  | C  | D  | E  | F  | G  | H  | I  |
|---|----|----|----|----|----|----|----|----|----|
| 1 | 6  | 2  | 6  | 7  | 4  | 2  | 5  | 9  | 60 |
| 2 | 4  | 9  | 5  | 3  | 8  | 5  | 8  | 2  | 55 |
| 3 | 5  | 2  | 1  | 9  | 7  | 4  | 3  | 3  | 51 |
| 4 | 7  | 6  | 7  | 3  | 9  | 2  | 7  | 1  | 43 |
| 5 | 2  | 3  | 9  | 5  | 7  | 2  | 6  | 5  | 41 |
| 6 | 5  | 5  | 2  | 2  | 8  | 1  | 4  | 3  | 52 |
| 7 | 35 | 37 | 22 | 32 | 41 | 32 | 43 | 38 |    |

图 1.8 运输问题的已知数据

|    | A  | B  | C  | D  | E  | F  | G  | H  | I  |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 6  | 2  | 6  | 7  | 4  | 2  | 5  | 9  | 60 |
| 2  | 4  | 9  | 5  | 3  | 8  | 5  | 8  | 2  | 55 |
| 3  | 5  | 2  | 1  | 9  | 7  | 4  | 3  | 3  | 51 |
| 4  | 7  | 6  | 7  | 3  | 9  | 2  | 7  | 1  | 43 |
| 5  | 2  | 3  | 9  | 5  | 7  | 2  | 6  | 5  | 41 |
| 6  | 5  | 5  | 2  | 2  | 8  | 1  | 4  | 3  | 52 |
| 7  | 35 | 37 | 22 | 32 | 41 | 32 | 43 | 38 |    |
| 8  |    |    |    |    |    |    |    |    |    |
| 9  | 0  | 19 | 0  | 0  | 41 | 0  | 0  | 0  |    |
| 10 | 1  | 0  | 0  | 32 | 0  | 0  | 0  | 0  |    |
| 11 | 0  | 11 | 0  | 0  | 0  | 0  | 40 | 0  |    |
| 12 | 0  | 0  | 0  | 0  | 0  | 5  | 0  | 38 |    |
| 13 | 34 | 7  | 0  | 0  | 0  | 0  | 0  | 0  |    |
| 14 | 0  | 0  | 22 | 0  | 0  | 27 | 3  | 0  |    |

图 1.9 求解结果的输出数据

## 1.6 LINGO 子模型

子模型必须包含在主模型之内，即必须位于以“MODEL:”开头、以“END”结束的模块内。同一个主模型中，允许定义多个子模型，所以每个子模型本身必须命名，其基本语法是：

```

SUBMODEL mymodel:
可执行语句（约束+目标函数）；
ENDSUBMODEL

```

其中 *mymodel* 是该子模型的名字，可执行语句一般是一些约束语句，也可能包含目标函数，但不可以有自身单独的集合段、数据段、初始段和计算段。也就是说，同一个主模型内的变量都是全局变量，这些变量对主模型和所有子模型同样有效。

如果已经定义了子模型 *mymodel*，则在计算段中可以用语句“@SOLVE(*mymodel*);”求

解这个子模型

**例 1.7** 当参数  $a=0, 1, 2, 3, 4$  ;  $b=2, 4, 6, 7$  时, 分别求下列的非线性规划问题。

$$\begin{aligned} \min \quad & 4x_1^3 - ax_1 - 2x_2, \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 4, \\ 2x_1 + x_2 \leq 5, \\ -x_1 + bx_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned}$$

解  $a$  的取值有 5 种可能,  $b$  的取值有 4 种可能,  $(a,b)$  的取值总共有 20 种组合, 这就需要求解 20 个非线性规划问题。利用 LINGO 的子模型功能, 只要编写一个 LINGO 程序就可以解决问题。

计算的 LINGO 程序如下:

```
model:
sets:
var1/1..5/:a0; !a0用于存放a的取值;
var2/1..4/:b0; !b0用于存放b的取值;
var3/1 2/:x;
endsets
data:
a0=0 1 2 3 4;
b0=2 4 6 7;
enddata
submodel sub_obj: !定义目标函数子模型;
[obj] min=4*x(1)^3-a*x(1)-2*x(2); !为了下面引用目标函数的值, 这里对目标函数进行了
标号;
endsubmodel
submodel sub_con: !定义约束条件子模型;
x(1)+x(2)<4;
2*x(1)+x(2)<5;
-x(1)+b*x(2)>2;
endsubmodel
calc:
@for(var1(i):@for(var2(j):a=a0(i);b=b0(j);@solve(sub_obj,sub_con); !调用目标函数和约
束条件子模型, 即求解数学规划;
@write('a=',a0(i),'b=',b0(j),' 时 , 最 优 解 x1=',x(1),'x2=',x(2),' , 最 优 值 为 ',obj,' 。
',@newline(2))))); !输出最优解和最优值;
endcalc
end
```

## 习题 1

注 1.4 所有 LINGO 程序必须使用集合。

### 1.1 求解线性规划问题

$$\begin{aligned} \max \quad & z = 72x_1 + 64x_2 \\ \text{s.t.} \quad & \begin{cases} x_1 + x_2 \leq 50, \\ 12x_1 + 8x_2 \leq 480, \\ 3x_1 \leq 100, \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

## 1.2 求解线性规划问题

$$\min Z_1 = 20x_1 + 90x_2 + 80x_3 + 70x_4 + 30x_5,$$

$$\text{s.t.} \quad \begin{cases} x_1 + x_2 + x_5 \geq 30, \\ x_3 + x_4 \geq 30, \\ 3x_1 + 2x_3 \leq 120, \\ 3x_2 + 2x_4 + x_5 \leq 48, \\ x_j \geq 0, \quad j=1, \dots, 5. \end{cases}$$

1.3 美佳公司计划制造 I、II 两种家电产品。已知各制造一件时分别占用的设备 A、设备 B 的台时、调试工序时间及每天可用于这两种家电的能力、各销售一件时的获利情况，如表 1.2 所示。问该公司应制造两种家电各多少件，使获取的利润为最大。

表 1.2

| 项目     | I | II | 每天可用能力 |
|--------|---|----|--------|
| 设备 A/h | 0 | 5  | 15     |
| 设备 B/h | 6 | 2  | 24     |
| 调试工序/h | 1 | 1  | 5      |
| 利润/元   | 2 | 1  |        |

## 1.4 求解标准的指派问题，其中指派矩阵

$$C = \begin{bmatrix} 6 & 7 & 5 & 8 & 9 & 10 \\ 6 & 3 & 7 & 9 & 3 & 8 \\ 8 & 11 & 12 & 6 & 7 & 9 \\ 9 & 7 & 5 & 4 & 7 & 6 \\ 5 & 8 & 9 & 6 & 10 & 7 \\ 9 & 8 & 7 & 6 & 5 & 9 \end{bmatrix}.$$

1.5 已知矩阵  $A = \begin{bmatrix} 1 & 4 & 5 \\ 4 & 2 & 6 \\ 5 & 6 & 3 \end{bmatrix}$ ,  $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ , 求二次型  $f(x_1, x_2, x_3) = x^T A x$  在单位球面

$x_1^2 + x_2^2 + x_3^2 = 1$  上的最小值。

1.6 一个塑料大筐里装满了鸡蛋，两个两个地数，余 1 个鸡蛋；三个三个地数，正好数完；四个四个地数，余 1 个鸡蛋；五个五个地数，余 4 个鸡蛋；六个六个地数，余 3 个鸡蛋；七个七个地数，余 4 个鸡蛋；八个八个地数，余 1 个鸡蛋；九个九个地数，正好数完。要求建立数学规划模型求解如下问题：

- (1) 求大筐中鸡蛋个数的最小值是多少。
- (2) 求大筐中鸡蛋个数的另外两个较小的值。

1.7 已知某物资有 8 个配送中心可以供货，有 15 个部队用户需要该物资，配送中心和部

队用户之间单位物资的运费，15个部队用户的物资需求量和8个配送中心的物资储备量数据见表1。

表 1.3 配送中心和部队用户之间单位物资的运费和物资需求量、储备量数据

| 部队<br>用户 | 单位物资的运费 |        |       |       |        |        |        |       | 需求量  |
|----------|---------|--------|-------|-------|--------|--------|--------|-------|------|
|          | 1       | 2      | 3     | 4     | 5      | 6      | 7      | 8     |      |
| 1        | 390.6   | 618.5  | 553   | 442   | 113.1  | 5.2    | 1217.7 | 1011  | 3000 |
| 2        | 370.8   | 636    | 440   | 401.8 | 25.6   | 113.1  | 1172.4 | 894.5 | 3100 |
| 3        | 876.3   | 1098.6 | 497.6 | 779.8 | 903    | 1003.3 | 907.2  | 40.1  | 2900 |
| 4        | 745.4   | 1037   | 305.9 | 725.7 | 445.7  | 531.4  | 1376.4 | 768.1 | 3100 |
| 5        | 144.5   | 354.6  | 624.7 | 238   | 290.7  | 269.4  | 993.2  | 974   | 3100 |
| 6        | 200.2   | 242    | 691.5 | 173.4 | 560    | 589.7  | 661.8  | 855.7 | 3400 |
| 7        | 235     | 205.5  | 801.5 | 326.6 | 477    | 433.6  | 966.4  | 1112  | 3500 |
| 8        | 517     | 541.5  | 338.4 | 219   | 249.5  | 335    | 937.3  | 701.8 | 3200 |
| 9        | 542     | 321    | 1104  | 576   | 896.8  | 878.4  | 728.3  | 1243  | 3000 |
| 10       | 665     | 827    | 427   | 523.2 | 725.2  | 813.8  | 692.2  | 284   | 3100 |
| 11       | 799     | 855.1  | 916.5 | 709.3 | 1057   | 1115.5 | 300    | 617   | 3300 |
| 12       | 852.2   | 798    | 1083  | 714.6 | 1177.4 | 1216.8 | 40.8   | 898.2 | 3200 |
| 13       | 602     | 614    | 820   | 517.7 | 899.6  | 952.7  | 272.4  | 727   | 3300 |
| 14       | 903     | 1092.5 | 612.5 | 790   | 932.4  | 1034.9 | 777    | 152.3 | 2900 |
| 15       | 600.7   | 710    | 522   | 448   | 726.6  | 811.8  | 563    | 426.8 | 3100 |
| 储备量      | 18600   | 19600  | 17100 | 18900 | 17000  | 19100  | 20500  | 17200 |      |

(1) 根据题目给定的数据，求最小运费调用计划。

(2) 若每个配送中心，可以对用户配送物资，也可以不对用户配送物资；若配送物资的话，配送量要大于等于 1000 且小于等于 2000，求此时的费用最小调用计划。

1.8 求解下列的线性方程组

$$\begin{cases} 4x_1 + x_2 & = 1, \\ x_1 + 4x_2 + x_3 & = 2, \\ x_2 + 4x_3 + x_4 & = 3, \\ \vdots & \\ x_{998} + 4x_{999} + x_{1000} & = 999, \\ x_{999} + 4x_{1000} & = 1000. \end{cases}$$