

# LINGO软件简介

主讲人：薛 震



中北大学数学系

E-mail : [xuezhen@live.nuc.edu.cn](mailto:xuezhen@live.nuc.edu.cn)

# 主要内容:

LINGO入门

在LINGO中使用集合

LINGO与外部文件的接口

LINGO在数学建模中的应用

# 第一节

## LINGO入门



- 一、LINGO的基本特征
- 二、LINGO的窗口介绍

LINGO: Linear Interactive and General Optimizer

即“交互式的线性和通用优化求解器”，它是一种专门用于求解最优化问题的软件，由美国芝加哥大学的 Linus Schrage 教授于1980年开发，其公司(LINDO系统公司)的网址为：<http://www.lindo.com>

LINGO软件能求解的优化模型：

- 线性规划(LP)
- 非线性规划(NLP)
- 二次规划(QP)



# 一、LINGO的基本特征

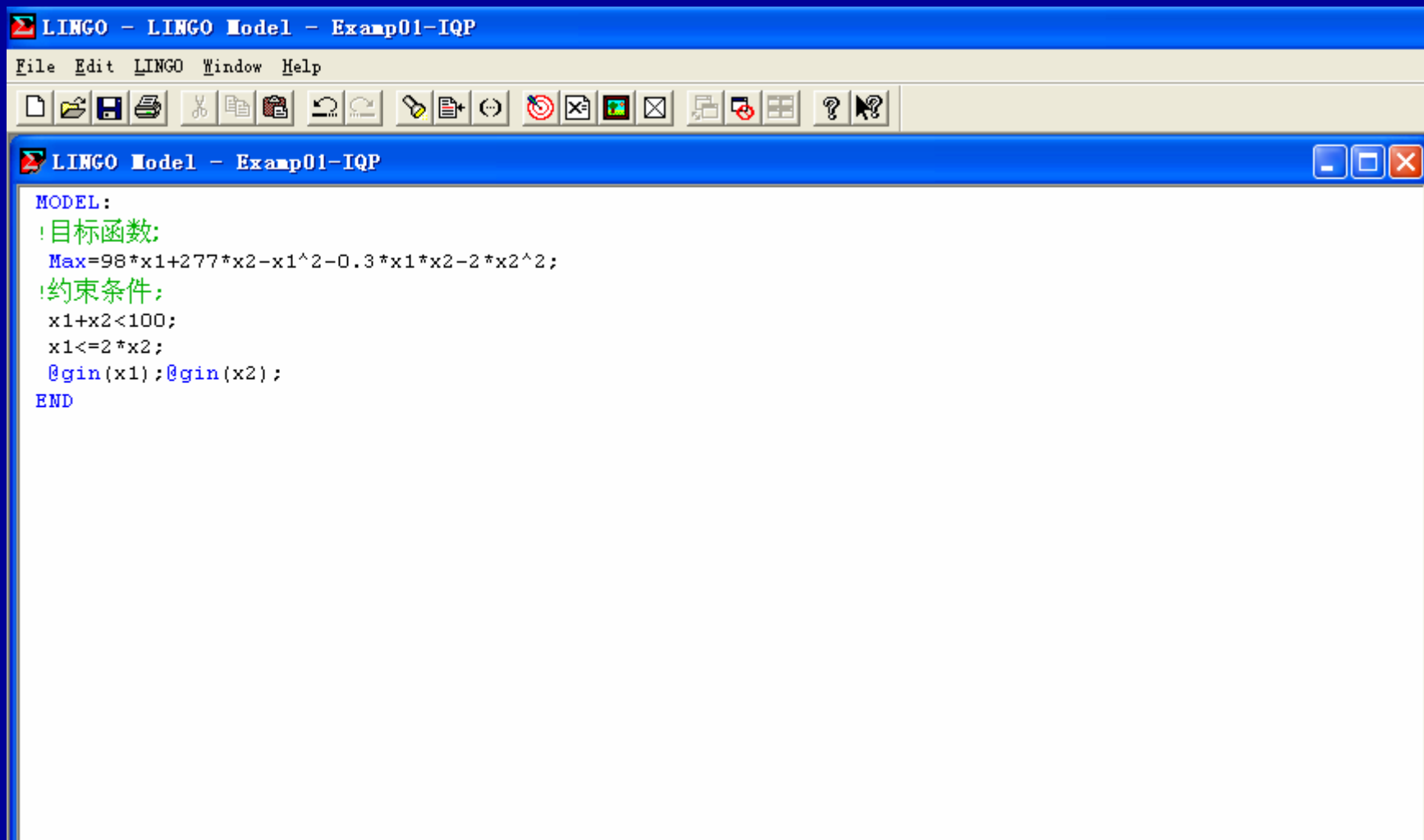
例1：用LINGO求解二次规划问题：

目标函数  $\max 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 2x_2^2$

约束条件 
$$\begin{cases} x_1 + x_2 \leq 100 \\ x_1 \leq 2x_2 \\ x_1, x_2 \geq 0 \text{ 为整数} \end{cases}$$



**解：** 直接在LINGO的模型窗口中输入程序：



The screenshot shows the LINGO software interface. The title bar reads "LINGO - LINGO Model - Examp01-IQP". The menu bar includes "File", "Edit", "LINGO", "Window", and "Help". The toolbar contains various icons for file operations, editing, and solving. The main text area contains the following LINGO model code:

```
MODEL:
!目标函数:
Max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
!约束条件:
x1+x2<100;
x1<=2*x2;
@gin(x1);@gin(x2);
END
```



# 1、程序语言说明

```
LINGO Model - Examp01-IQP

MODEL:
!目标函数:
Max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
!约束条件:
x1+x2<100;
x1<=2*x2;
@gin(x1);@gin(x2);
END
```

LINGO程序以“MODEL”开始,以“END”结束. 它们之间由语句组成, 且每个语句都以分号“;”结尾.

一行中感叹号“!”后面的文字将被认为是注释语句, 不参与模型的建立(内容为绿色字符).

LINGO中的语句顺序  
是不重要的,因为LINGO  
总是根据“Max=”或“Min=”

```
LINGO Model - Examp01-IQP

MODEL:
!目标函数:
Max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
!约束条件:
x1+x2<100;
x1<=2*x2;
@gin(x1);@gin(x2);
END
```

语句寻找目标函数,其它语句都是约束条件.

LINGO程序中不区分大小写字母.(实际上任何小写字  
符将被转换为大写字符)

LINGO中的变量必须以字母开头,且最多不能超过32个  
字符.





在LINGO中, 以@开头的  
都是函数的调用.

LINGO已假定所有变量  
非负, 可用限定变量取值  
范围的函数@BIN、@GIN、@FREE、@BND改变变  
量的非负假定.

```
LINGO Model - Examp01-IQP

MODEL:
!目标函数;
Max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
!约束条件;
x1+x2<100;
x1<=2*x2;
@gin(x1);@gin(x2);
END
```

## 2、关键字说明

主要关键字:  
(多为蓝色字符)

MODEL:	模型的开始
END	模型的结束
TITLE	对模型命名
MAX	目标最大化
SETS:	集合段开始
ENDSETS	集合段结束
DATA:	数据段开始
ENDDATA	数据段结束



### 3、LINGO函数说明

LINGO主要函数:

基本的数学函数

变量定界函数

文件输入输出函数

集合循环函数

集合操作函数

概率中的函数

结果报告函数



## 基本的数学函数

@ABS(X): 绝对值函数, 返回X的绝对值.

@COS(X): 余弦函数, 返回X的余弦值(X为弧度).

@EXP(X): 指数函数, 返回  $e^X$  的值.

@LOG(X): 自然对数函数, 返回X的自然对数值.

@FLOOR(X): 取整函数, 返回X的整数部分.

@MOD(X,Y): 模函数, 返回X对Y取模的结果, 即X除以Y的余数, 这里的X和Y为整数.



@POW(X,Y): 指数函数, 返回  $X^Y$  的值.

@SQR(X): 平方函数, 返回X的平方值.

@SQRT(X): 平方根函数, 返回X的正的平方根值.

@SIGN(X): 符号函数, 返回X的符号值(  $X < 0$  时  
返回-1,  $X \geq 0$  时返回+1 ).

@SMAX(list): 最大值函数, 返回一系列数(list)的最  
大值.



## 变量定界函数

@GIN(X): 限制X为整数.

@BIN(X): 限制X为0或1.

@FREE(X): 取消对X的符号限制.

@BND(L,X,U): 限制  $L \leq X \leq U$  .

**注:** 有关其它函数的介绍, 请参考LINGO的帮助文件.



## 4、运算符说明

### 运算符

算数运算符:  $+$ (加法),  $-$ (减法或负号),  $*$ (乘法),  
 $/$ (除法),  $^$ (求幂).

关系运算符:  $<$ (即 $\leq$ , 小于等于),  $=$ (等于),  
 $>$ (即 $\geq$ , 大于等于).

注: 优化模型中的约束一般没有严格小于、严格大于关系.



## 逻辑运算符:

#AND#(与),      #OR#(或),      #NOT#(非);

#EQ#(等于),      #NE#(不等于),      #GT#(大于).


#GE#(大于等于), #LT#(小于), #LE#(小于等于).

**注:** 逻辑运算的结果为“真”(TRUE)和“假”(FALSE), LINGO  
中用数字1代表TRUE, 其它值都是FALSE.





# 运算符的优先级

优先级	运算符
最高	#NOT# , -(负号)
	$\wedge$
	$*$ , $/$
	$+$ , $-$ (减法)
	#EQ# , #NE# , #GT# , #GE# , #LT# , #LE#
	#AND# , #OR#
最低	$<$ , $=$ , $>$



## 二、LINGO的窗口介绍

LINGO的窗口：

- LINGO的主窗口
- LINGO模型窗口
- LINGO状态窗口
- LINGO报告窗口



# 例1的运算结果：

主窗口

模型窗口

报告窗口

状态窗口

The screenshot displays the LINGO software interface with the following components:

- 主窗口 (Main Window):** The top window titled "LINGO - Solution Report - Examp01-IQP" showing the model and solution report.
- 模型窗口 (Model Window):** A window titled "LINGO Model - Examp01-IQP" showing the model code.
- 报告窗口 (Report Window):** A window titled "Solution Report - Examp01-IQP" showing the solution report.
- 状态窗口 (Status Window):** A window titled "LINGO Solver Status [Examp01-IQP]" showing solver status and extended solver status.

**Model Code:**

```
MODEL:
!目标函数:
Max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
!约束条:
x1+x2=200;
x1<=200;
@gin(x1);
END
```

**Solution Report:**

```
Local optimal solution found
Objective value:
Extended solver steps:
Total solver iterations:
```

**LINGO Solver Status [Examp01-IQP]:**

Solver Status	
Model	PINLP
State	Local Opt
Objective:	11077.5
Feasibility:	0
Iterations:	149

Variables	
total:	2
nonlinear:	2
integers:	2

Constraints	
total:	3
nonlinear:	1

Nonzeros	
total:	6
nonlinear:	2

Extended Solver Status	
Solver	B-and-B
Best	11077.5
Obj Bound:	11077.5
Steps:	3
Active:	0

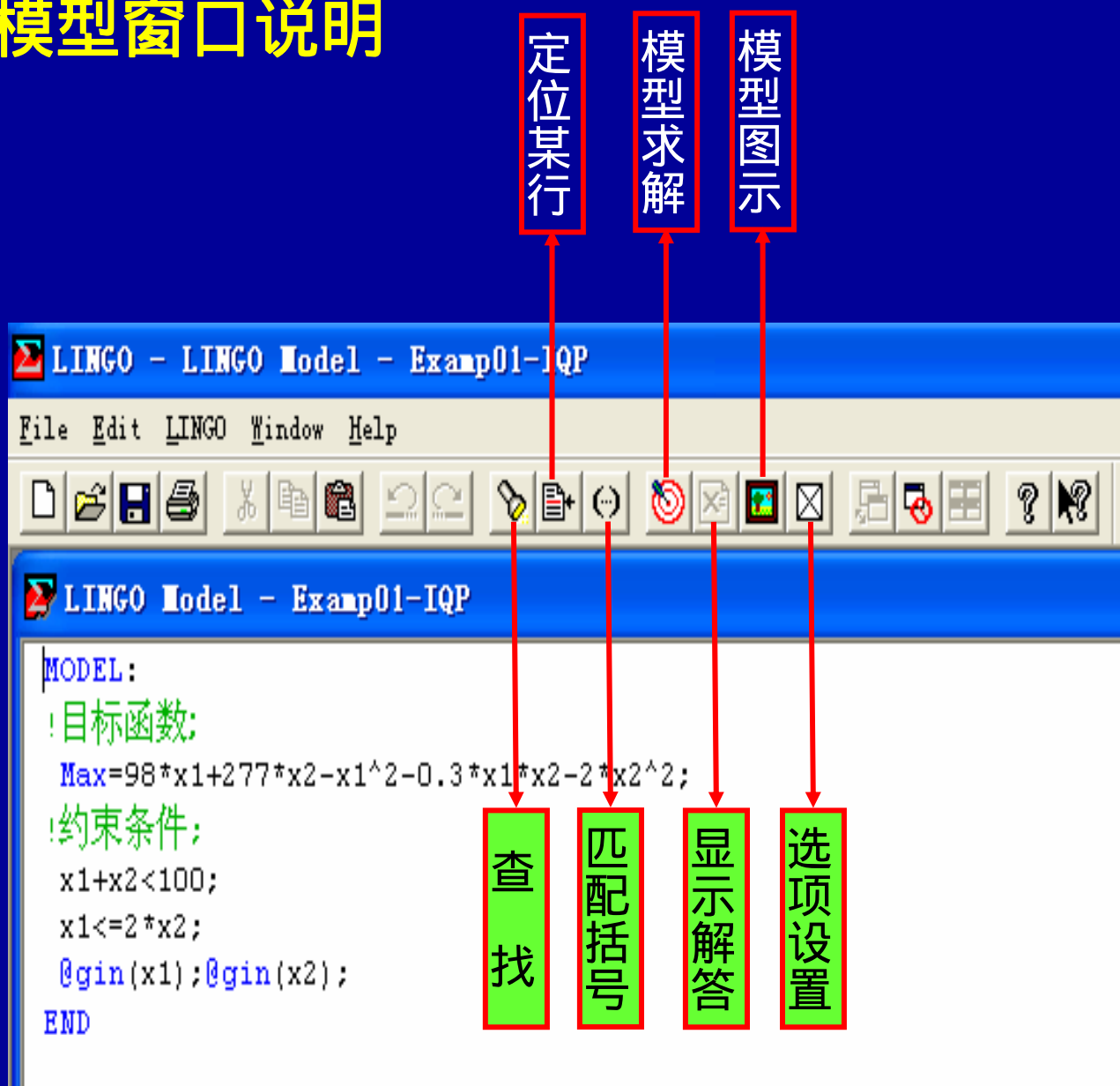
Generator Memory Used (K)	
total:	18

Elapsed Runtime (hh:mm:ss): 00:00:01

Update: 2 Interrupt Solver Close



# 1、主窗口与模型窗口说明



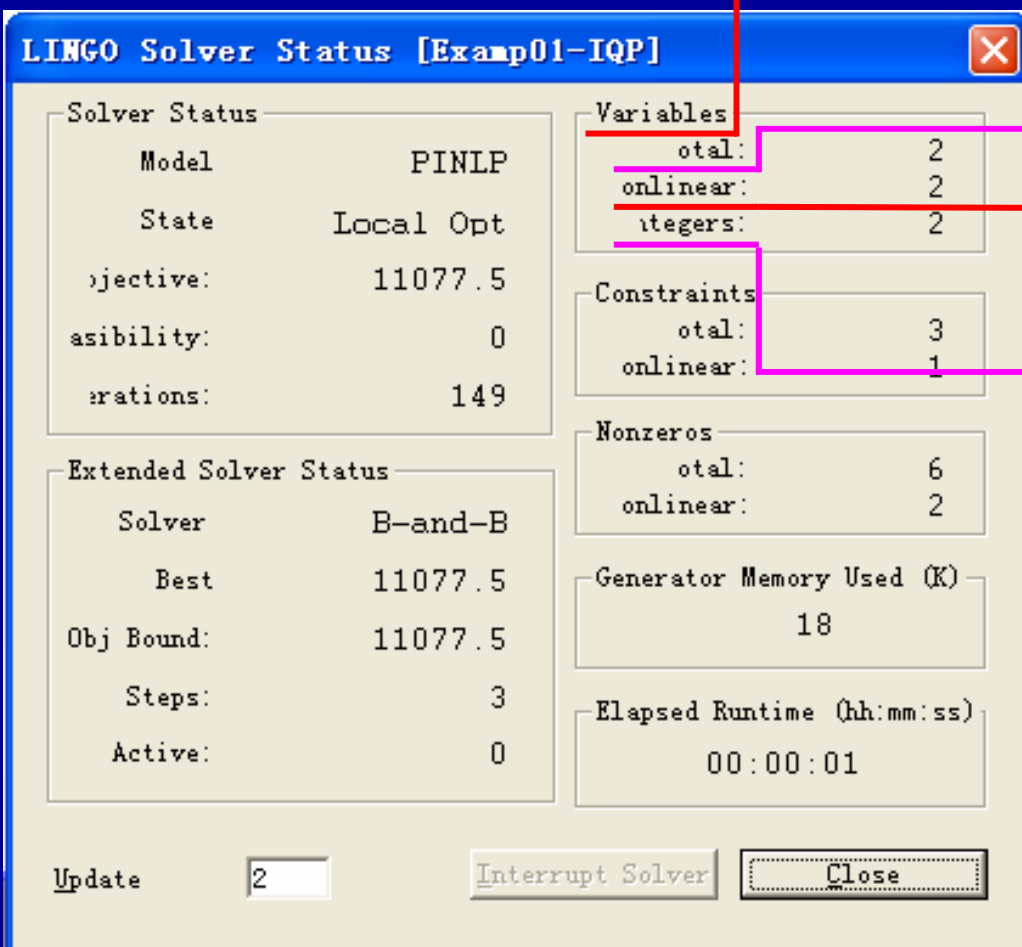


## 文件类型

- \*.lg4 : LINGO格式的模型文件
- \*.lng : 文本格式的模型文件
- \*.ldt : LINGO数据文件
- \*.ltf : LINGO命令脚本文件
- \*.lgr : LINGO报告文件
- \*.ltx : LINGO格式的文本文件
- \*.mps : 数学规划系统格式的模型文件



## 2、状态窗口说明（例1）



Variables(变量数量)

Total(变量总数)

Nonlinear(非线性变量)

Integer(整数数量)

注:由于LINGO对中文操作系统的兼容性不好,所以有些显示字符和单词被截掉了.



# 状态窗口说明

LINGO Solver Status [Examp01-IQP]

Solver Status	
Model	PINLP
State	Local Opt
Objective:	11077.5
Feasibility:	0
Iterations:	149

Extended Solver Status	
Solver	B-and-B
Best	11077.5
Obj Bound:	11077.5
Steps:	3
Active:	0

Variables	
total:	2
nonlinear:	2
integers:	2

Constraints	
total:	3
nonlinear:	1

Nonzeros	
total:	6
nonlinear:	2

Generator Memory Used (K)	
18	

Elapsed Runtime (hh:mm:ss)	
00:00:01	

Update

Constraints(约束数量)

Nonzeros(非零系数数量)

内存使用量

求解花费的时间



# 状态窗口说明

求解器状态框

模型的类型

解的状态

Objective(解的目标函数值)

Infeasibility(约束不满足总量)

到目前为止的迭代次数

扩展的求解器状态框

LINGO Solver Status [Examp01-IQP]

Solver Status	
Model	PINLP
State	Local Opt
Objective:	11077.5
Infeasibility:	0
Iterations:	149

Variables	
total:	2
nonlinear:	2
integers:	2

Constraints	
total:	3
nonlinear:	1

Nonzeros	
total:	6
nonlinear:	2

Generator Memory Used (K)	
18	

Elapsed Runtime (hh:mm:ss)	
00:00:01	

Extended Solver Status	
Solver	B-and-B
Best	11077.5
Obj Bound:	11077.5
Steps:	3
Active:	0

Update





### 3、报告窗口说明（例1）

得到局部最优解

变量的检验数

扩展求解步骤数

松弛或剩余变量的值

Variable	Value	Reduced Cost
X1	35.00000	-8.500002
X2	65.00000	-6.500004

Row	Slack or Surplus	Dual Price
1	11077.50	1.000000
2	0.000000	0.000000
3	95.00000	0.000000

对偶价格的值



## 第二节

# 在LINGO中使用集合

- 一、集合的基本用法
- 二、基本集合与派生集合
- 三、集合的使用小结





# 一、集合的基本用法

**例2:** 某帆船公司需要决定下四个季度的帆船生产量。下四个季度的帆船需求量分别是40条,60条,75条,25条,这些需求必须按时满足。每个季度正常的生产能力是40条帆船,每条船的生产费用为400美元。如果加班生产,每条船的生产费用为450美元。每个季度末,每条船的库存费用为20美元。假定生产提前期为0,初始库存为10条帆船。如何安排生产可使总费用最小?



**分析：**该问题考虑的是4个季度，求解过程中需要设16个变量(需求量、正常生产的产量、加班生产的产量、库存量等)，若考虑的是更多季度(如1000个)的时候，则问题比较麻烦。下面利用集合的概念，可简化问题的求解。

**解：**用DEM, RP, OP, INV分别表示需求量、正常生产的产量、加班生产的产量、库存量，则DEM, RP, OP, INV对每个季度都应该有一个对应的值，即它们都是由4个元素组成的数组。



目标函数是所有费用的和:

$$\min \sum_{I=1}^4 [400RP(I) + 450OP(I) + 20INV(I)]$$

约束条件有:

能力限制  $RP(I) \leq 40, I = 1, \dots, 4$

产品数量的平衡方程  $INV(0) = 10$

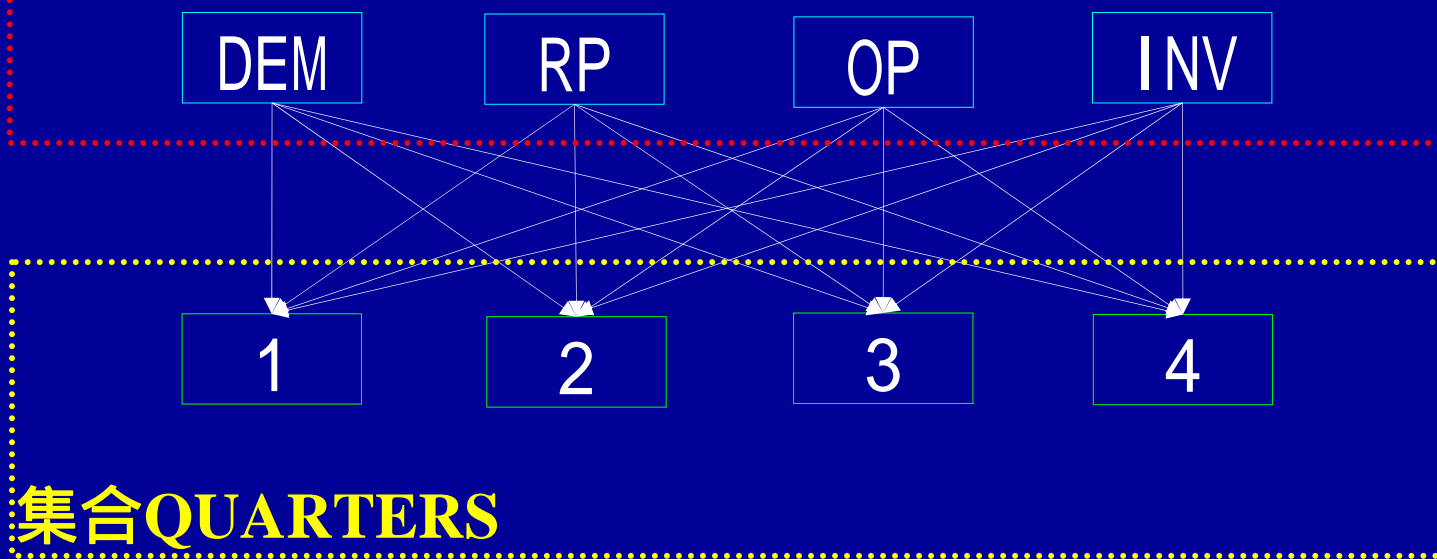
$$INV(I) = INV(I-1) + RP(I) + OP(I) - DEM(I)$$
$$I = 1, \dots, 4$$

非负约束  $DEM(I), RP(I), OP(I), INV(I) \geq 0, I = 1, \dots, 4$



记4个季度组成的集合为 $QUARTERS=\{1,2,3,4\}$ ，则数组DEM,RP,OP,INV对集合QUARTERS中的每个元素分别对应于一个值，把DEM,RP,OP,INV称为该集合的属性。

### 集合QUARTERS的属性



该模型的LINGO程序为：

集合段

数据段

目标与约束段

```
LINGO Model - Examp02

MODEL:

SETS:
    QUARTERS/1,2,3,4/:DEM,RP,OP,INV;
ENDSETS

DATA:
    DEM=40,60,75,25;
ENDDATA

MIN=@SUM(QUARTERS(I):400*RP(I)+450*OP(I)+20*INV(I));
@FOR(QUARTERS(I):RP(I)<40);
@FOR(QUARTERS(I)|I#GT#1:
    INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(I););
INV(1)=10+RP(1)+OP(1)-DEM(1);

END
```



## 集合段

SETS:

QUARTERS/1,2,3,4/:DEM,RP,OP,INV;

ENDSETS

语句“QUARTERS/1,2,3,4/:DEM,RP,OP,INV;”就是定义了集合QUARTERS={1,2,3,4},以及对应于该集合的属性DEM,RP,OP,INV,其结果正是定义了16个变量名DEM(1), DEM(2), DEM(3), .





## 数据段

```
DATA:
```

```
    DEM=40,60,75,25;
```

```
ENDDATA
```

语句“**DEM=40,60,75,25;**”给出了已知的常量DEM的值，  
即 $DEM(1)=40$ ， $DEM(2)=60$ ， $DEM(3)=75$ ， $DEM(4)=25$ 。



## 目标与约束段

```
MIN=@SUM(QUARTERS(I):400*RP(I)+450*OP(I)+20*INV(I));  
@FOR(QUARTERS(I):RP(I)<40);  
@FOR(QUARTERS(I)|I#GT#1:  
    INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(I););  
INV(1)=10+RP(1)+OP(1)-DEM(1);
```

目标函数是用求和函数“@SUM(集合(下标):关于集合属性的表达式)”的方式定义的。

约束是用循环函数“@FOR(集合(下标):关于集合属性的约束关系)”的方式定义的。语句“@FOR(QUARTERS(I):RP(I)<40);”的含义为“每个季度正常的生产能力是40条船”



## 目标与约束段

```
MIN=@SUM(QUARTERS(I):400*RP(I)+450*OP(I)+20*INV(I));  
@FOR(QUARTERS(I):RP(I)<40);  
@FOR(QUARTERS(I)|I#GT#1:  
    INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(I););  
INV(1)=10+RP(1)+OP(1)-DEM(1);
```

对于产品数量的平衡方程而言，由于下标 $I=1$ 时的约束关系与 $I=2,3,4$ 时有所区别(因为定义的变量 $INV$ 是不包含 $INV(0)$ )，因此把 $I=1$ 的约束关系单独写出“ $INV(1)=10+RP(1)+OP(1)-DEM(1);$ ”，而对 $I=2,3,4$ 对应的约束，增加了一个逻辑表达式来刻画：

$@FOR(QUARTERS(I)|I\#GT\#1:$

$INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(I););$



# 运行该LINGO程序得到解答报告：

全局最优解：

RP=(40,40,40,25)

OP=( 0,10,35,0 )

最小成本=78450

Solution Report - Examp02		
Global optimal solution found.		
Objective value:		78450.00
Total solver iterations:		8
Variable	Value	Reduced Cost
DEM( 1)	40.00000	0.000000
DEM( 2)	60.00000	0.000000
DEM( 3)	75.00000	0.000000
DEM( 4)	25.00000	0.000000
RP( 1)	40.00000	0.000000
RP( 2)	40.00000	0.000000
RP( 3)	40.00000	0.000000
RP( 4)	25.00000	0.000000
OP( 1)	0.000000	20.00000
OP( 2)	10.00000	0.000000
OP( 3)	35.00000	0.000000
OP( 4)	0.000000	50.00000
INV( 1)	10.00000	0.000000
INV( 2)	0.000000	20.00000
INV( 3)	0.000000	70.00000
INV( 4)	0.000000	420.0000
Row	Slack or Surplus	Dual Price
1	78450.00	-1.000000
2	0.000000	30.00000
3	0.000000	50.00000
4	0.000000	50.00000
5	15.00000	0.000000
6	0.000000	450.0000
7	0.000000	450.0000
8	0.000000	400.0000
9	0.000000	430.0000



## 二、基本集合与派生集合

**例3:** 建筑工地的位置(用平面坐标 $a, b$ 表示, 距离单位: km)及水泥日用量 $d$ (单位: t)由表2-1给出.

(1) 目前有两个临时料场位于 $P(5, 1), Q(2, 7)$ , 日储量各有20t. 求从 $P, Q$ 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.

(2) 为进一步减少吨公里数, 打算改建两个日储量仍各为20t的新料场, 两个新的料场应建在何处, 节省的吨公里数有多大?



工地位置坐标

表2-1 (选址问题)

工地	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

水泥日用量



**解：**记工地的位置为  $(a_i, b_i)$ ，水泥日用量为  $d_i$ ；料场位置为  $(x_j, y_j)$ ，日储量为  $e_j$ ；从料场  $j$  向工地  $i$  的运送量为  $c_{ij}$ ， $i = 1, \dots, 6$ ； $j = 1, 2$ 。

**目标函数：**总吨公里数最小

$$\min f = \sum_{j=1}^2 \sum_{i=1}^6 c_{ij} \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}$$

**约束条件：**各工地的日用量必需满足

$$\sum_{j=1}^2 c_{ij} = d_i, i = 1, \dots, 6$$



**约束条件：** 各料场的运送量不能超过日储量

$$\sum_{i=1}^6 c_{ij} \leq e_j, j=1,2$$

当使用临时料场时(问题(1)), 决策变量只有  $c_{ij}$ , 所以此时的优化模型为线性规划模型; 当为新建料场选址时(问题(2)), 决策变量为  $c_{ij}$  和  $x_j, y_j$ , 所以在新建料场时的优化模型是非线性规划模型(NLP).

下面先用集合的概念求解NLP模型(问题(2)), 而把现有临时料场的位置作为初始解告诉LINGO.





定义需求点demand和供应点supply两个集合，分别有6个和2个元素(下标)；利用集合demand和supply定义一个二维集合 $link=\{(s,t) | s \in demand, t \in supply\}$ ，然后将 $c_{ij}$  (运送量)定义成这个新集合的属性。

**注：**类似于demand和supply这种直接把元素列举出来的集合称为基本集合；而把link这种基于其它集合而派生出来的多维集合称为派生集合。

具体的输入程序如下：



## 集合段

## 数据段

## 初始段

## 目标与约束段

```
LINGO Model - Examp03

MODEL:
Title Location Problem;
sets:
    demand/1..6/:a,b,d;
    supply/1..2/:x,y,e;
    link(demand,supply):c;
endsets
data:
    !locations for the demand(需求点的位置);
    a=1.25,8.75,0.5,5.75,3,7.25;
    b=1.25,0.75,4.75,5,6.5,7.75;
    !quantities of the demand and supply (供需量);
    d=3,5,4,7,6,11; e=20,20;
enddata
init:
    !initial locations for the supply (初始点);
    x=5,2;
    y=1,7;
endinit

!Objective function (目标);
[OBJ] min=@sum(link(i,j): c(i,j)*((x(j)-a(i))^2+(y(j)-b(i))^2)^(1/2) );
!demand constraints (需求约束);
@for(demand(i):[DEMAND_CON] @sum(supply(j):c(i,j)) =d(i));
!supply constraints (供应约束);
@for(supply(i):[SUPPLY_CON] @sum(demand(j):c(j,i)) <=e(i); );
@for(supply: @bnd(0.5,x,8.75); @bnd(0.75,y,7.75); );
END
```

{1,2,3,4,5,6}

二维集合

$(a_6, b_6)$





## 目标与约束段说明

```
!Objective function (目标);  
[OBJ] min=@sum(link(i,j): c(i,j)*((x(j)-a(i))^2+(y(j)-b(i))^2)^(1/2) );  
!demand constraints (需求约束);  
@for(demand(i):[DEMAND_CON] @sum(supply(j):c(i,j)) =d(i));  
!supply constraints (供应约束);  
@for(supply(i):[SUPPLY_CON] @sum(demand(j):c(j,i)) <=e(i); );  
@for(supply: @bnd(0.5,X,8.75); @bnd(0.75,Y,7.75); );
```

此处[OBJ]和[DEMAND\_CON]、[SUPPLY\_CON]为对目标行和两类约束分别进行命名;

集合循环函数是指对集合上的元素(下标)进行循环操作的函数,其一般用法如下:

@function(setname[(set\_index\_list)[|condition]]:expression\_list)

其中function是五种集合函数(FOR, MAX, MIN, PROD, SUM)之一.



# 运行该LINGO程序得到解答报告：

局部最优解：

$$X(1)=3.254883$$

$$Y(1)=5.652332$$

$$X(2)=7.250000$$

$$Y(2)=7.750000$$

$$\text{最小运量}=85.26604$$

**Solution Report - Examp03**

Local optimal solution found.  
Objective value: 85.26604  
Total solver iterations: 61

Model Title: Location Problem

Variable	Value	Reduced Cost
A( 1)	1.250000	0.000000
A( 2)	8.750000	0.000000
A( 3)	0.500000	0.000000
A( 4)	5.750000	0.000000
A( 5)	3.000000	0.000000
A( 6)	7.250000	0.000000
B( 1)	1.250000	0.000000
B( 2)	0.750000	0.000000
B( 3)	4.750000	0.000000
B( 4)	5.000000	0.000000
B( 5)	6.500000	0.000000
B( 6)	7.750000	0.000000
D( 1)	3.000000	0.000000
D( 2)	5.000000	0.000000
D( 3)	4.000000	0.000000
D( 4)	7.000000	0.000000
D( 5)	6.000000	0.000000
D( 6)	11.000000	0.000000
X( 1)	3.254883	0.000000
X( 2)	7.250000	0.6335133E-06
Y( 1)	5.652332	0.000000
Y( 2)	7.750000	0.5438639E-06
E( 1)	20.000000	0.000000
E( 2)	20.000000	0.000000
C( 1, 1)	3.000000	0.000000
C( 1, 2)	0.000000	4.008540
C( 2, 1)	0.000000	0.2051358
C( 2, 2)	5.000000	0.000000
C( 3, 1)	4.000000	0.000000



**注：**如果要把料场 $P(5, 1)$ ,  $Q(2, 7)$ 的位置看成是已知并且固定的，这时是LP模型(即问题(1)).只要在上述程序中把初始段的“ $x=5, 2; y=1, 7$ ”语句移到数据段就可以了.

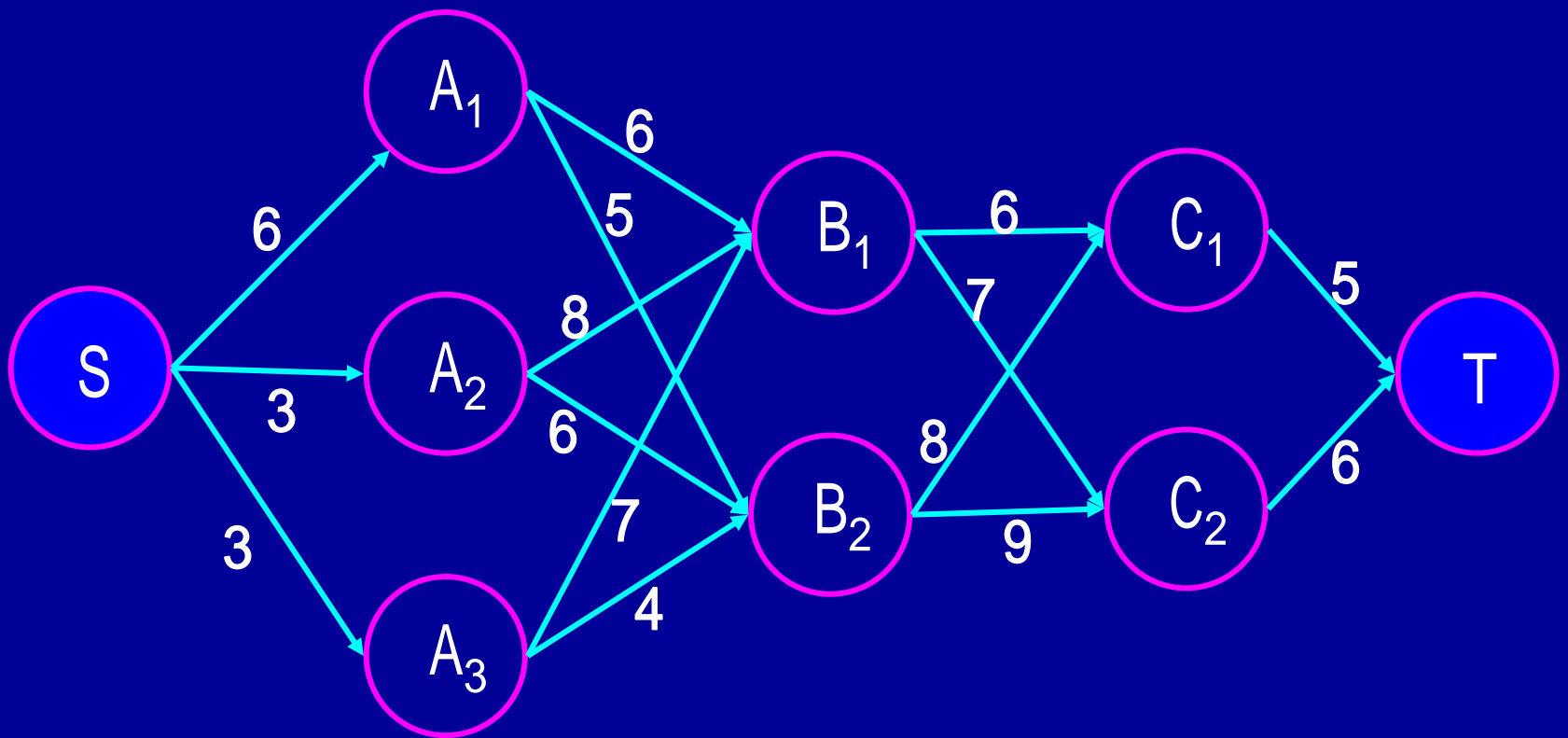
运行结果得到全局最优解，最小运量为136.2275.



**例4:**在纵横交错的公路网中,货车司机希望找到一条从一个城市到另一个城市的最短路. 假设图2-2表示的是该公路网,节点表示货车可以停靠的城市,弧上的权表示两个城市之间的距离(百公里). 那么, 货车从城市S出发到达城市T,如何选择行驶路线,使所经过的路程最短?



图2-2 ( 最短路问题 )



**解:** 假设从  $S$  到  $T$  的最优行驶路线  $P$  经过城市  $C_1$ , 则  $P$  中从  $S$  到  $C_1$  的子路也一定是从  $S$  到  $T$  的最优行驶路线. 因此, 为了得到从  $S$  到  $T$  的最优行驶路线, 只需先求出从  $S$  到  $C_k (k=1,2)$  的最优行驶路线. 同样, 为了求出从  $S$  到  $C_k (k=1,2)$  的最优行驶路线, 只需先求从  $S$  到  $B_j (j=1,2)$  的最优行驶路线, 依次类推. 这样, 把从  $S$  到  $T$  的行驶过程分成4个阶段:

$$S \longrightarrow A_i (i=1,2,3); A_i \longrightarrow B_j (j=1,2)$$
$$B_j \longrightarrow C_k (k=1,2); C_k \longrightarrow T.$$




记  $d(X,Y)$  为城市  $Y$  与城市  $X$  之间的直接距离,  $L(X)$  表示城市  $S$  到城市  $X$  的最优行驶路线的路长, 则有:

$$L(S) = 0;$$

$$L(X) = \min_{Y \neq X} \{L(Y) + d(Y, X)\}, \quad X \neq S$$

直接计算如下:  $L(A_1) = 6, \quad L(A_2) = 3, \quad L(A_3) = 3$

$$L(B_1) = \min\{L(A_1) + 6, L(A_2) + 8, L(A_3) + 7\} = 10 = L(A_3) + 7$$

$\vdots$

最优行驶路线为:  $S \longrightarrow A_3 \longrightarrow B_2 \longrightarrow C_1 \longrightarrow T$



上述方法称为动态规划, 下用LINGO求解该问题:

对L进行赋值  
(逗号不可少)

@index(S)=1  
S在集合中的位置

```
MODEL:
SETS:
  CITIES /S,A1,A2,A3,B1,B2,C1,C2,T/: L;
  ROADS(CITIES, CITIES)/
    S,A1 S,A2 S,A3
    A1,B1 A1,B2 A2,B1 A2,B2 A3,B1 A3,B2
    B1,C1 B1,C2 B2,C1 B2,C2
    C1,T C2,T/: D;
ENDSETS
DATA:
  D = 6 3 3
      6 5 8 6 7 4
      6 7 8 9
      5 6;
  L = 0, , , , , , , ;
ENDDATA
@FOR( CITIES( i) | #GT # @index(S):
  L( i) = @MIN( ROADS( j, i): L( j) + D( j, i));
END
```

一个元素

!属性L(i)表示城市s到城市i的最优行驶路线的路长;  
!派生集合ROADS表示的是网络中的道路(弧);  
!由于并非所有城市间都有道路直接连接, 所以将弧具体列出;

!属性D(i, j)是城市i到j的直接距离(已知);


!因为L(S)=0;

该程序没有目标函数,这在LINGO中是允许的,可以用来寻找可行解.

!这行中"@index(S)"可以直接写成"1";  
!这就是前面写出的最短路关系式;



# 运行该LINGO程序得到解答报告：

 Solution Report - Examp04

Feasible solution found.  
Total solver iterations: 0

Variable	Value
L( S)	0.000000
L( A1)	6.000000
L( A2)	3.000000
L( A3)	3.000000
L( B1)	10.000000
L( B2)	7.000000
L( C1)	15.000000
L( C2)	16.000000
<u>L( T)</u>	<u>20.000000</u>
D( S, A1)	6.000000
D( S, A2)	3.000000
D( S, A3)	3.000000
D( A1, B1)	6.000000
D( A1, B2)	5.000000
D( A2, B1)	8.000000
D( A2, B2)	6.000000
D( A3, B1)	7.000000
D( A3, B2)	4.000000
D( B1, C1)	6.000000
D( B1, C2)	7.000000
D( B2, C1)	8.000000

从  $S$  到  $T$  的最优行驶路线的路长为20.

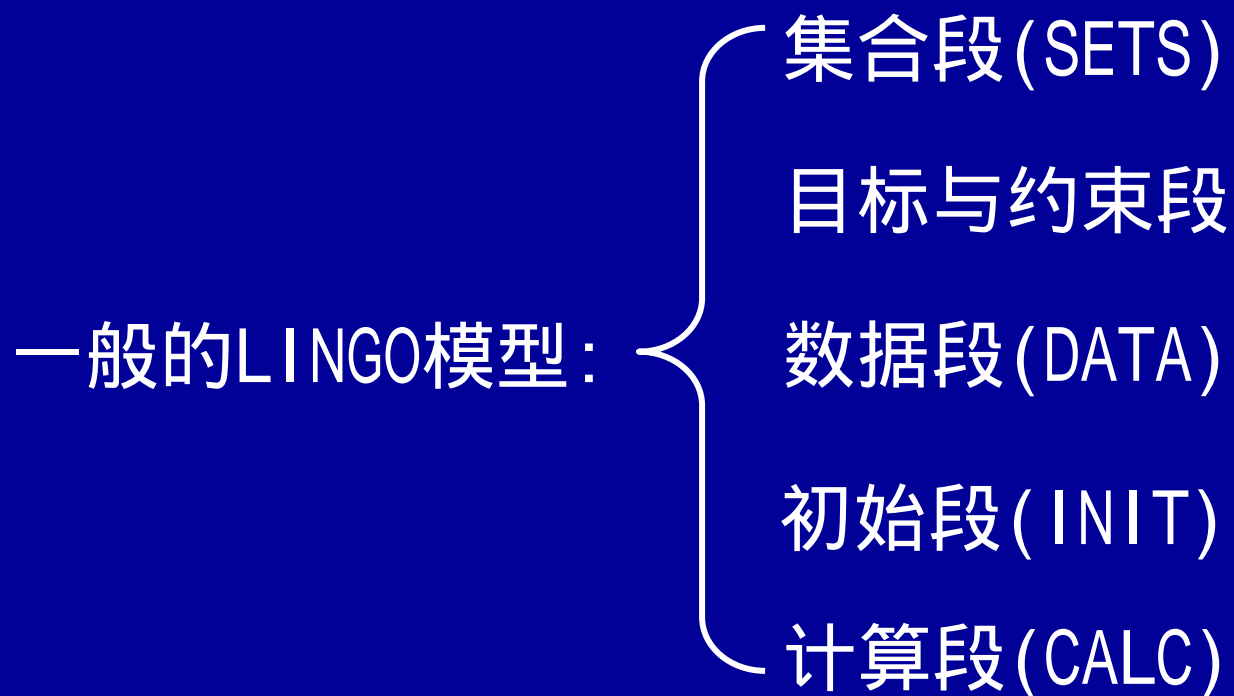
最优行驶路线为：

$S \rightarrow A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow T$

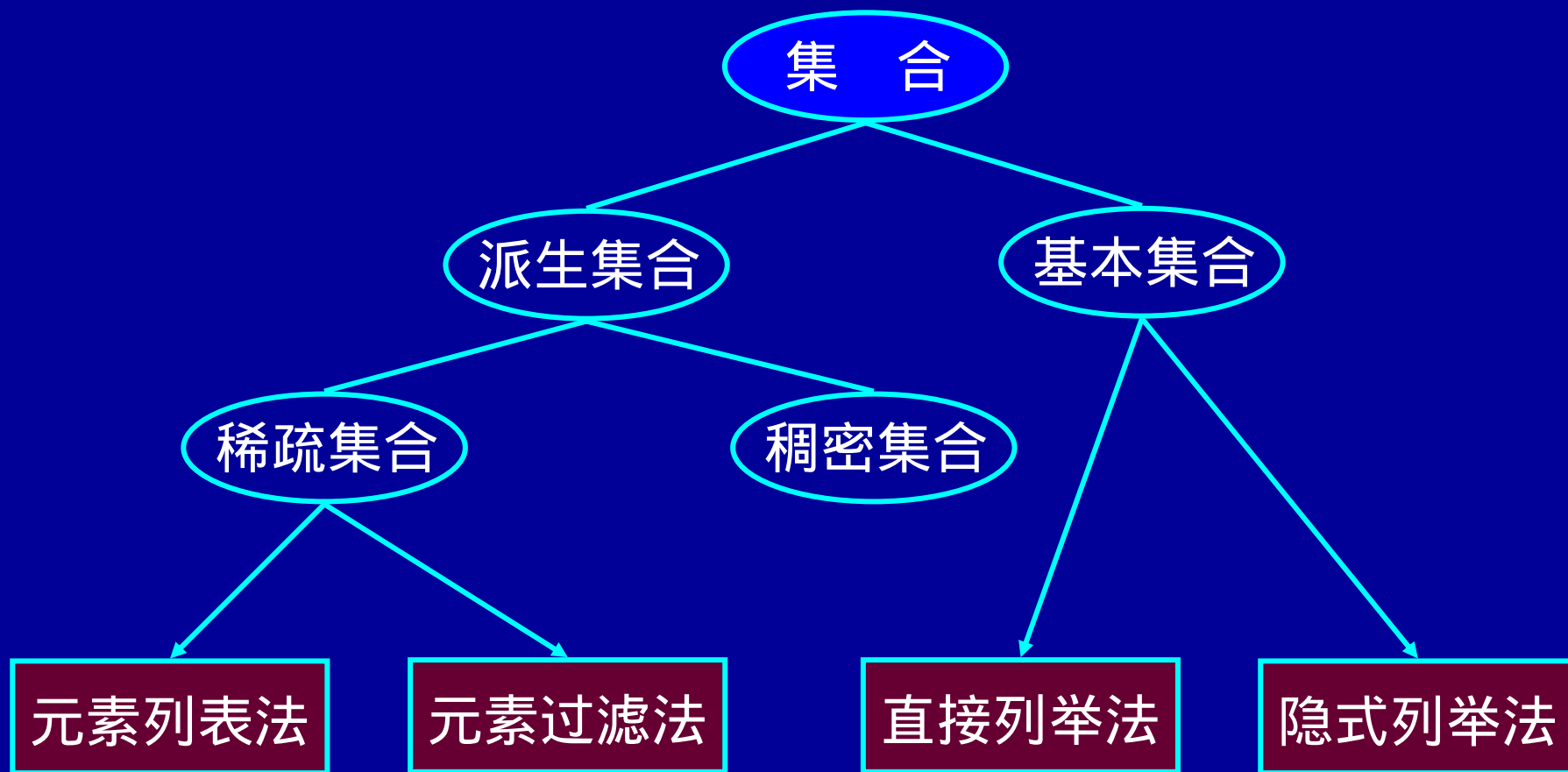


# 三、集合的使用小结

## 1、LINGO模型最基本的组成要素



## 2、集合的类型及其关系





## 基本集合的隐式列举法说明

类型	隐式列举格式	示例	示例集合的元素
数字型	1..n	1..5	1, 2, 3, 4, 5
字符-数字型	stringM..stringN	car101..car208	car101, car102, ... , car208
星期型	dayM..dayN	MON..FRI	MON, TUE, WED, THU, FRI
月份型	monthM..monthN	OCT..JAN	OCT, NOV, DEC, JAN
年份-月份型	monthYearM.. monthYearN	OCT2001..JAN2002	OCT2001, NOV2001, DEC2001, JAN2002



### 3、集合的定义语法

(1)基本集合的一般定义格式为：

`setname[/member_list/][:attribute_list];`

其中`setname`为定义的集合名，`member_list`为元素列表，`attribute_list`为属性列表。

**注：**语法中凡是在方括号“`[]`”中的内容，表示的是可选的项(即该项可以没有)。



(2) 派生集合的一般定义格式为：

`setname(parent_set_list)[/member_list/][:attribute_list];`

其中`parent_set_list`为父集合列表。





## 第三节

# LINGO与外部文件的接口



- 一、通过文本文件传递数据
- 二、通过Excel文件传递数据

# 一、通过文本文件传递数据

## 1、通过文本文件输入数据

@FILE函数通常可以在集合段和数据段使用,但不允许嵌套使用. 这个函数的一般用法是: @FILE(filename)

当前模型引用其它ASCII码文件中的数据或文本时可以采用该语句, 其中filename为存放数据的文件名(可以包含完整的路径名,没有指定路径时表示在当前目录下寻找这个文件), 该文件记录之间必须用“~”分开.



**例5:** 假想一个最简单的采购问题：有多个城市都需要采购一定的物品，但每个城市只允许在自身所在的城市采购，设城市I的最低需求量为 $NEED(I)$ ，最大的供应量是 $SUPPLY(I)$ ，单件采购成本是 $COST(I)$ 。问如何采购可使总成本最小？



**解:** 设采购量用ORDERED表示, 这个问题的优化

模型是一目了然的.

目标函数:  $\min \sum COST(I) \times ORDERED(I)$

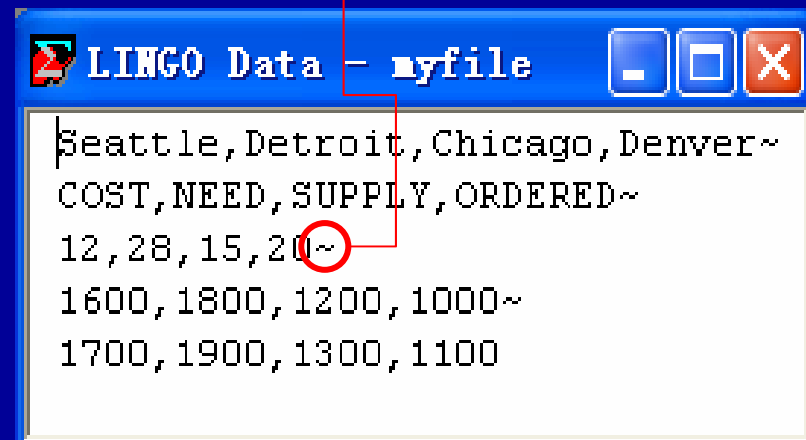
约束条件:  $NEED(I) \leq ORDERED(I) \leq SUPPLY(I)$

假设存放数据的文本文件

myfile.lbt(LINGO数据文件)的

内容如右图所示:

记录之间必须用“~”分开



```
Seattle, Detroit, Chicago, Denver~  
COST, NEED, SUPPLY, ORDERED~  
12, 28, 15, 200~  
1600, 1800, 1200, 1000~  
1700, 1900, 1300, 1100
```

在LINGO模型窗口中建立如下LINGO模型：

```
MODEL:
SETS:
    MYSET / @FILE(myfile.ldt) / : @FILE(myfile.ldt);
ENDSETS
MIN = @SUM( MYSET( I):ORDERED( I) * COST( I));
@FOR( MYSET( I):
    ORDERED( I) > NEED( I);
    ORDERED( I) < SUPPLY( I));
DATA:
    COST = @FILE( myfile.ldt);
    NEED = @FILE( myfile.ldt);
    SUPPLY = @FILE( myfile.ldt);
ENDDATA
END
```

```
LINGO Data - myfile
Seattle, Detroit, Chicago, Denver~
COST, NEED, SUPPLY, ORDERED~
12, 28, 15, 20~
1600, 1800, 1200, 1000~
1700, 1900, 1300, 1100
```

程序与数据分离



# 运行该LINGO程序得到解答报告：

Solution Report - Examp05			
Global optimal solution found.			
Objective value:		107600.0	
Total solver iterations:		0	
Variable		Value	Reduced Cost
COST( SEATTLE)		12.00000	0.000000
COST( DETROIT)		28.00000	0.000000
COST( CHICAGO)		15.00000	0.000000
COST( DENVER)		20.00000	0.000000
NEED( SEATTLE)		1600.000	0.000000
NEED( DETROIT)		1800.000	0.000000
NEED( CHICAGO)		1200.000	0.000000
NEED( DENVER)		1000.000	0.000000
SUPPLY( SEATTLE)		1700.000	0.000000
SUPPLY( DETROIT)		1900.000	0.000000
SUPPLY( CHICAGO)		1300.000	0.000000
SUPPLY( DENVER)		1100.000	0.000000
ORDERED( SEATTLE)		1600.000	0.000000
ORDERED( DETROIT)		1800.000	0.000000
ORDERED( CHICAGO)		1200.000	0.000000
ORDERED( DENVER)		1000.000	0.000000
Row	Slack or Surplus	Dual Price	
1	107600.0	-1.000000	
2	0.000000	-12.00000	

全局最优解：

最小成本=107600



## 2、通过文本文件输出数据

**@TEXT**函数用于文本文件输出数据。通常只在数据段使用此函数。其一般用法是：**@TEXT(['filename'])**

它用于将解答结果送到文本文件filename中，当省略filename时，结果送到标准的输出设备(通常就是屏幕)。filename中可以包含完整的路径名，没有指定时表示在当前目录下生成这个文件(如果文件已存在，将会被覆盖)。



**例6:** 在上面例5的例子中(模型Examp05.lg4), 假如

在数据段增加一些

@TEXT语句,则可

以输出更多的结果.

把上面的程序修改

为(Example06.lg4):

输出语句

```
LINGO Model - Examp06

MODEL:
SETS:
    MYSET / @FILE(myfile.lgt) / : @FILE(myfile.lgt);
ENDSETS
MIN = @SUM( MYSET( I):ORDERED( I) * COST( I));
@FOR( MYSET( I):
    [con1] ORDERED( I) > NEED( I);
    [con2] ORDERED( I) < SUPPLY( I));

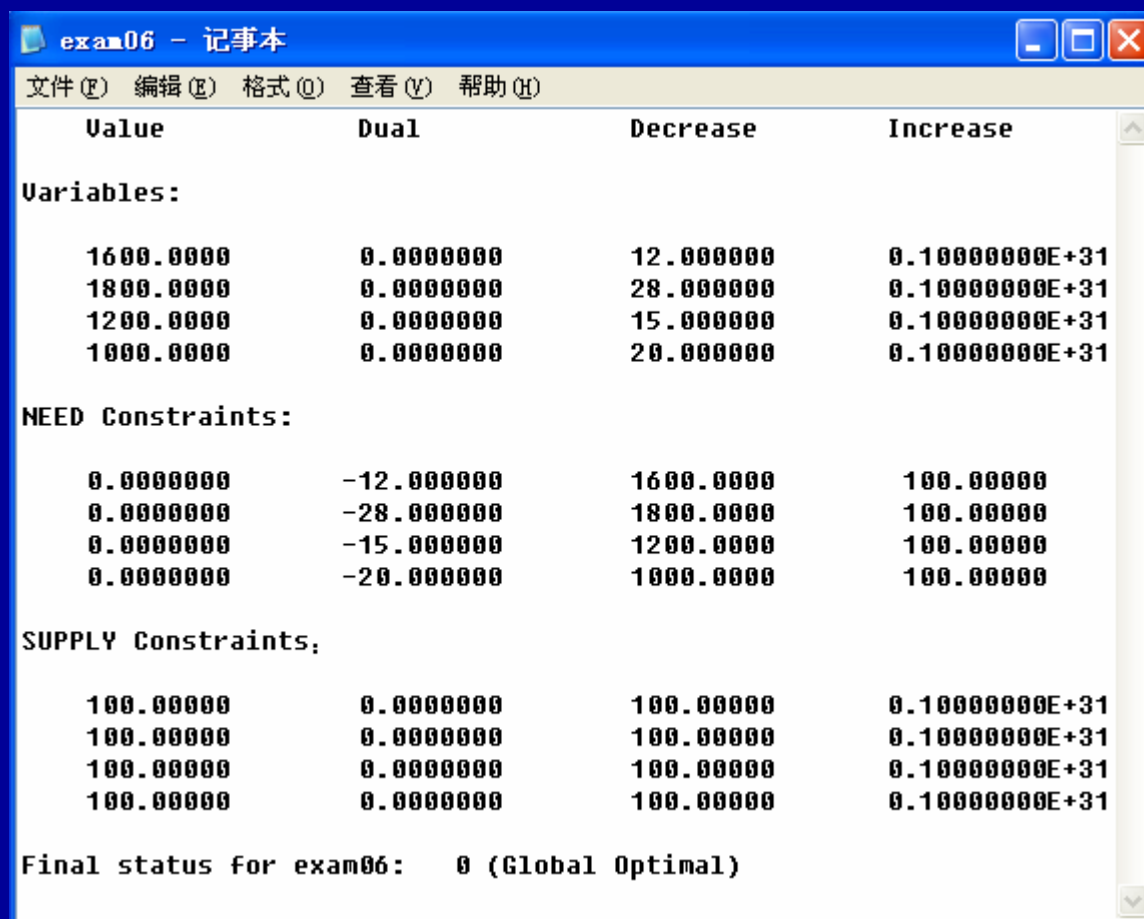
DATA:
    COST = @FILE( myfile.lgt);
    NEED = @FILE( myfile.lgt);
    SUPPLY = @FILE( myfile.lgt);
    @TEXT('exam06.txt')=@write(4*' ', 'Value', 12*' ', 'Dual', 13*' ',
        'Decrease', 8*' ', 'Increase', @newline(2));
    @TEXT('exam06.txt')=@write('Variables:', @newline(2));
    @TEXT('exam06.txt')=@write(ordered, @DUAL(ordered), @RANGED(ordered), @RANGEU(ordered);
    @TEXT('exam06.txt')=@write(@newline(1), 'NEED Constraints:', @newline(2));
    @TEXT('exam06.txt')=@write(@newline(1), 'SUPPLY Constraints:', @newline(2));
    @TEXT('exam06.txt')=@write(@newline(1), 'Final status for exam06: ', @status(),
        @if(@status(), ' (Maybe Not Global Optimal)', ' (Global
Optimal)'), @newline(1));
ENDDATA
END
```





当模型求解结束后，LINGO会将我们用@TEXT函数请求的值存入文件exam06.txt中(当前目录下将生成这个文件)。用文本编辑器打开该文件，发现其中显示的结果是：

**注：**运行程序时，需把选项卡General Solver中的选项Dual Computations|Prices改为Prices and Ranges.



Value	Dual	Decrease	Increase
Variables:			
1600.0000	0.0000000	12.000000	0.10000000E+31
1800.0000	0.0000000	28.000000	0.10000000E+31
1200.0000	0.0000000	15.000000	0.10000000E+31
1000.0000	0.0000000	20.000000	0.10000000E+31
NEED Constraints:			
0.0000000	-12.000000	1600.0000	100.00000
0.0000000	-28.000000	1800.0000	100.00000
0.0000000	-15.000000	1200.0000	100.00000
0.0000000	-20.000000	1000.0000	100.00000
SUPPLY Constraints:			
100.00000	0.0000000	100.00000	0.10000000E+31
100.00000	0.0000000	100.00000	0.10000000E+31
100.00000	0.0000000	100.00000	0.10000000E+31
100.00000	0.0000000	100.00000	0.10000000E+31
Final status for exam06: 0 (Global Optimal)			



## 二、通过Excel文件传递数据

@OLE函数只能在LINGO模型的集合段、数据段和初始段使用. 无论用于输入或输出数据, 该函数的使用格式都是: @OLE(spreadsheet\_file[,range\_name\_list])

其中spreadsheet\_file是电子表格文件的名称, 应当包括扩展名(如\*.xls等), 还可以包含完整的路径名, 只要字符数不超过64均可; range\_name\_list是指文件中包含数据的单元范围(格式与Excel中工作表的单元范围格式一致).

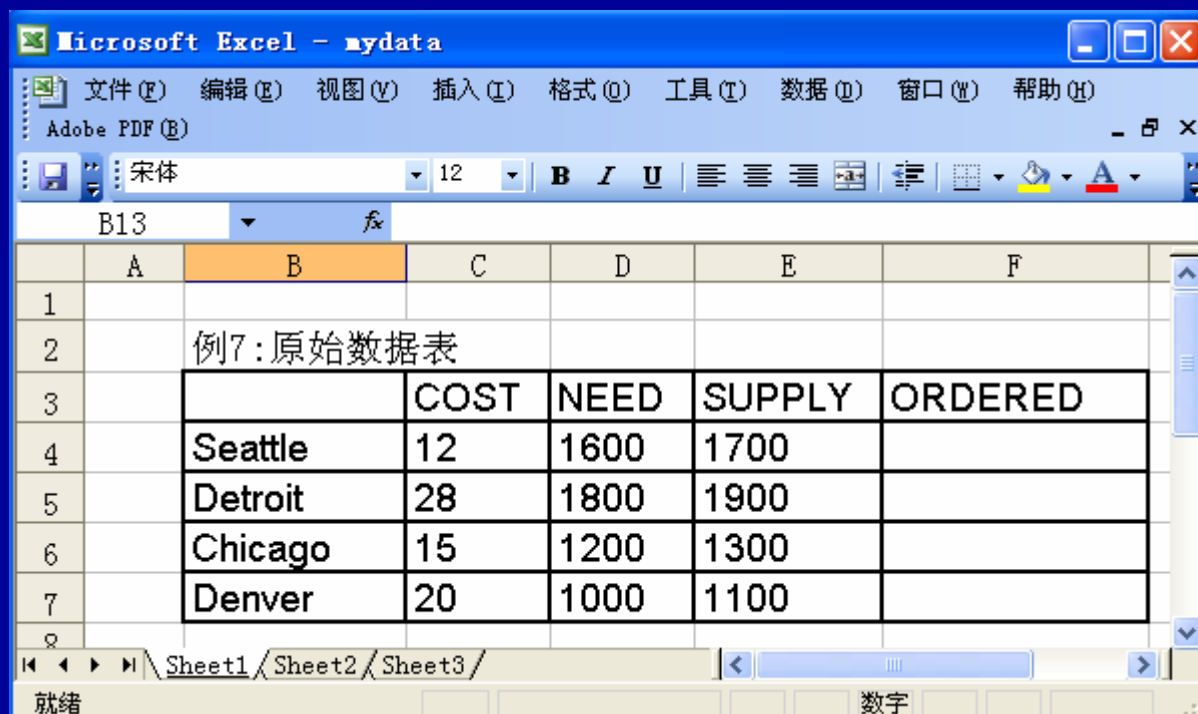


具体来说,当从Excel中向LINGO模型中输入数据时,在集合段可以直接采用“@OLE(...)”的形式,但在数据段和初始段应采用“属性(或变量)=@OLE(...)”的赋值形式;当从LINGO向Excel中输出数据时,应当采用“@OLE(...)=属性(或变量)”的赋值形式(自然,输出语句只能出现在数据段中)。



**例7:** 继续考虑上面例5的例子中(模型Examp05.lg4), 但通过@OLE语句输入输出数据.

首先, 用Excel建立一个名为mydata.xls的Excel数据文件, 如图所示:



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - mydata". The menu bar includes "文件(F)", "编辑(E)", "视图(V)", "插入(I)", "格式(O)", "工具(T)", "数据(D)", "窗口(W)", and "帮助(H)". The toolbar shows various icons for file operations and formatting. The active cell is B13. The data table is as follows:

	A	B	C	D	E	F
1						
2		例7:原始数据表				
3			COST	NEED	SUPPLY	ORDERED
4		Seattle	12	1600	1700	
5		Detroit	28	1800	1900	
6		Chicago	15	1200	1300	
7		Denver	20	1000	1100	

The status bar at the bottom shows "就绪" (Ready) and "数字" (Numbers).

为了能够通过@OLE函数与LINGO传递数据, 我们需要对这个文件中的数据进行命名. 具体做法是: 用鼠标选中表格的B4:B7单元, 然后选择命令“插入|名称|定义”, 这时将会弹出一个对话框, 请您输入名称, 例如可以将它命名为CITISES. 同理, 可以将C4:C7单元、 D4:D7单元、 E4:E7单元以及 F4:F7单元, 分别命名为COST、 NEED、 SUPPLY和SOLUTION.



把上面例5的程序修改为(存入文件Examp07.lg4):

从‘mydata.xls’的CITIES所指单元中取出数据作为集合MYSET的元素

从mydata.xls的COST指定的单元给COST赋值(NEED,SUPPLY类似)

将ORDERED的值输出赋给mydata.xls中由SOLUTION指定的单元格

```
LINGO Model - Examp07

MODEL:
SETS:
    MYSET / @OLE('E:\我的文档\LINGO软件简介\mydata.xls','CITIES') / :
        COST,NEED,SUPPLY,ORDERED;

ENDSETS

MIN = @SUM( MYSET( I): ORDERED( I) * COST( I));

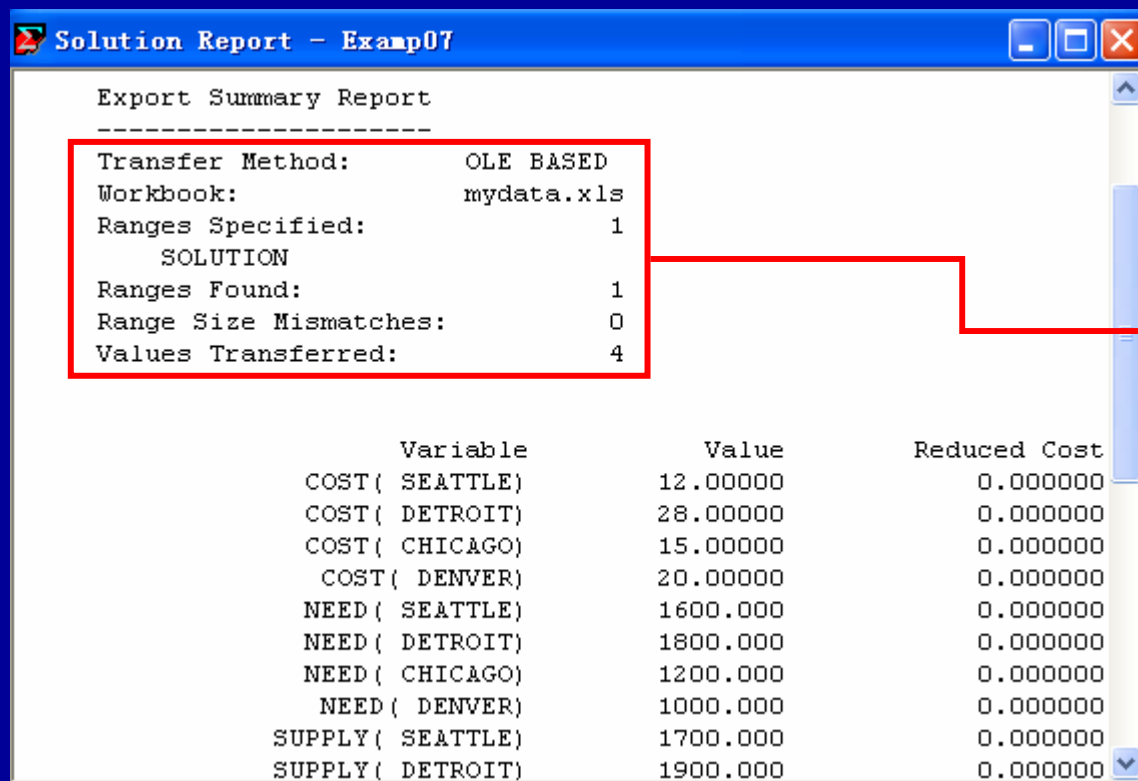
@FOR( MYSET( I):
    [CON1] ORDERED( I) > NEED( I);
    [CON2] ORDERED( I) < SUPPLY( I));

DATA:
    COST,NEED,SUPPLY = @OLE('mydata.xls');
    @OLE('mydata.xls','SOLUTION')=ORDERED;
ENDDATA

END
```

运行该LINGO程序得到解答报告：

若这时再打开mydata.xls查看，会发现ORDERED对应的一列也被自动写上了解答的结果。



Solution Report - Examp07

Export Summary Report

-----

Transfer Method:	OLE BASED
Workbook:	mydata.xls
Ranges Specified:	1
SOLUTION	
Ranges Found:	1
Range Size Mismatches:	0
Values Transferred:	4

Variable	Value	Reduced Cost
COST( SEATTLE)	12.00000	0.000000
COST( DETROIT)	28.00000	0.000000
COST( CHICAGO)	15.00000	0.000000
COST( DENVER)	20.00000	0.000000
NEED( SEATTLE)	1600.000	0.000000
NEED( DETROIT)	1800.000	0.000000
NEED( CHICAGO)	1200.000	0.000000
NEED( DENVER)	1000.000	0.000000
SUPPLY( SEATTLE)	1700.000	0.000000
SUPPLY( DETROIT)	1900.000	0.000000

输出总结报告



## 第四节

# LINGO在建模中的应用



- 一、露天矿生产的车辆安排
- 二、钢管下料问题



# 软件介绍

# 结束

