Lingo 编程应用

Lingo Programming Application

主讲人：刘瑞华

时　间：2023.06.08

目录

PART
ONE

软件入门

Lingo 软件

目标对象

- **规划问题：** 线性规划问题、非线性规划、二次规划、整数规划、动态规划、多目标规划 等；
- **规划问题特点：** 变量比较多，或者约束条件表达式比较复杂，编程计算虽然可行，但工作量很大，程序很长而繁琐，稍不小心就会出错；
- **Lingo 的特点：** 专门用来求解各种规划问题的软件包，其功能十分强大，是求解优化模型的最优选择。

Lingo 概况（1）

发展历史

- **母体**：Lingo 是美国 LINGO 系统公司 （Lindo System Inc）开发的求解数学规划系列软件的一个软件；
- **功能**：求解 大型线性、非线性和整数规划 问题；
- **版本**：Lingo 的不同版本对 模型的变量总数、非线性变量数目、整数变量数目和约束条件的数量 都分别做出了不同的限制；
- **各种版本**： Demo、Solve Suite、Super、Hyper、Industrial、Extended；
- **目前版本**： V18.0

Lingo 概况（3）

功能特点

（1）既能求解线性规划问题，也有较强的求解非线性规划问题的能力；

（2）输入模型简练直观；

（3）运行速度快、计算能力强；

（4）内置建模语言，提供几十个内部函数，从而能以较少语句，较直观的方式描述较大规模的优化模型；

（5）将集合的概念引入编程语言，很容易将实际问题转换为 Lingo 模型；

（6）能方便地与 Excel、数据库等其他软件交换数据；

Lingo 的基本用法（1）

优化模型

（1） 目标函数：最大值、最小值

（2） 决策变量

（3） 约束条件：等式，或不等式

数学模型 1.1

目标函数 $\max = 200x_1 + 300x_2$

约束条件
$$\begin{cases} x_1 \le 100, \\ x_2 \le 120, \\ x_1 + 2x_2 \le 160 \\ x_i \ge 0, i = 1,2. \end{cases}$$

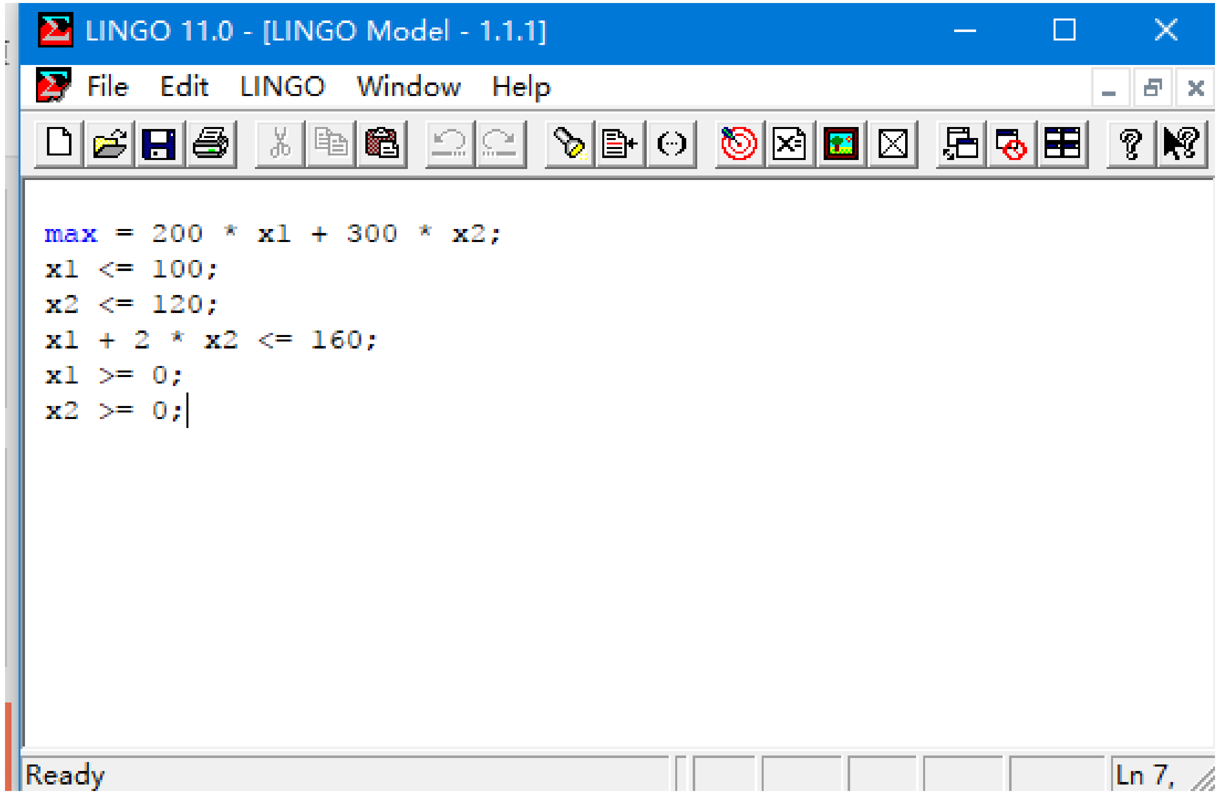
Lingo 的基本用法（2）

数学模型

目标函数 $\max = 200x_1 + 300x_2$

约束条件
$$\begin{cases} x_1 \le 100, \\ x_2 \le 120, \\ x_1 + 2x_2 \le 160 \\ x_i \ge 0, i = 1,2. \end{cases}$$

Lingo 代码 1.1



Lingo 的基本用法（3）

Lingo 的语法规定：

(1) 求目标函数的最大值或最小值分别用 max = , 或 min = 来表示;

(2) 每个语句必须以分号“; ”结束，每行可以有多个语句，语句可以跨行;

(3) 变量名称必须以字母、数字和下划线所组成，长度不超过32字符，不区分大小写;

(4) 可以给语句加上标号，例如 【OBJ】 max = 200 * x1 + 300 * x2;

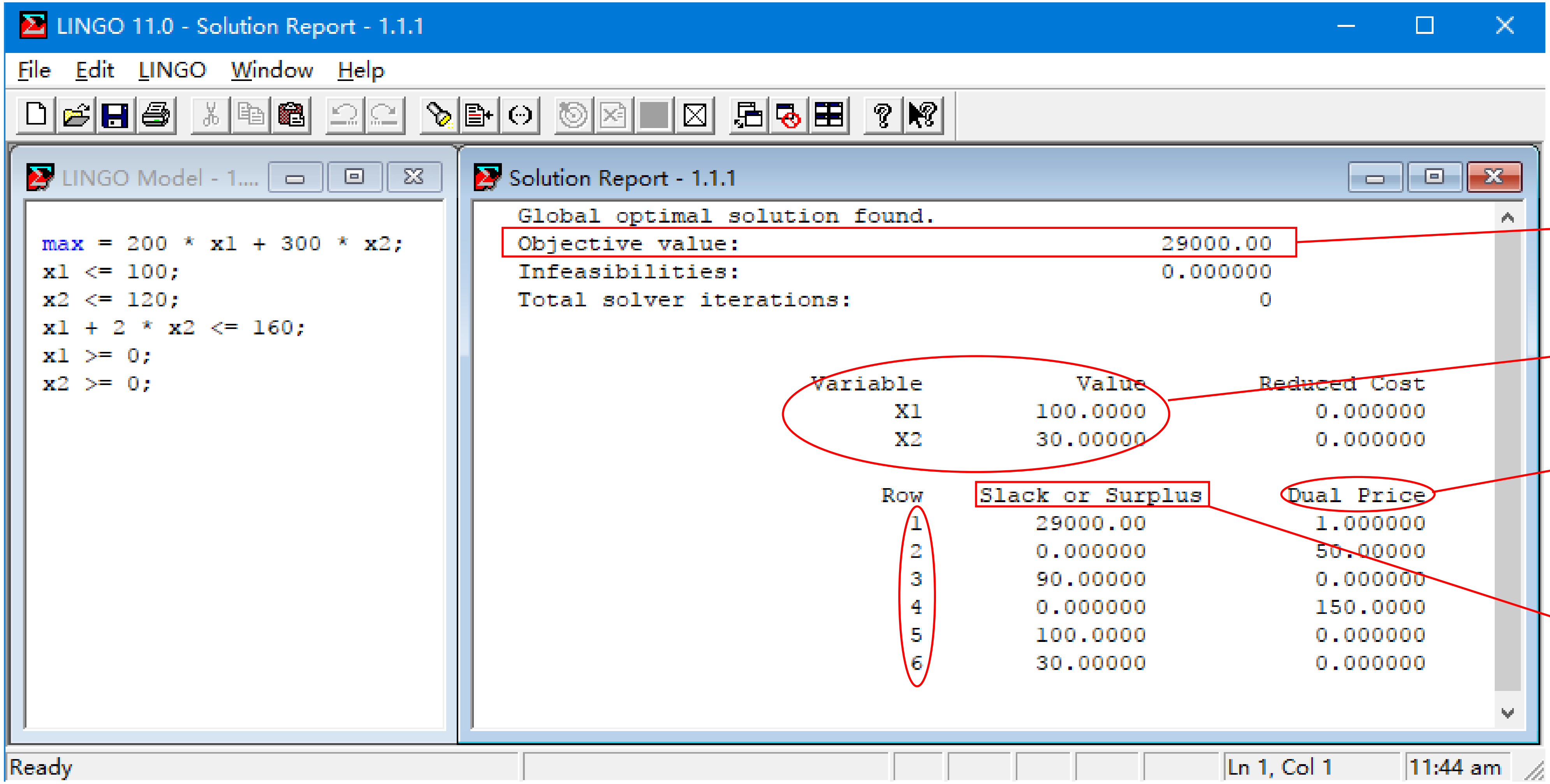
(5) 以 ！ 开头，以 “; ”号结束的语句是注释语句;

(6) 如果对变量的取值范围没有作特殊说明，则默认所有决策变量都非负;

(7) Lingo 模型以语句“Model”开头，以“End”结束，但也可以省略;

Lingo 的基本用法（4）

结果运行



目标函数值

变量值

影子价格

松弛或剩余

Lingo 的基本用法（5）

结果解析

（1）**松弛或剩余：即约束条件的左边与右边的差值**

- 对于“<=”不等式，右边减左边的差值称为松弛；
- 对于“>=”不等式，左边减右边的差值称为剩余；
- 当约束条件左右两边相等时，松弛或剩余的值为零；
- 如果约束条件无法满足时，即没有可行解；

（2）**影子价格：**上面 row2 的松弛值为 0，第一个变量已经达到饱和状态，影子价格为50，

- **即：当第一个变量增加 1 时，其能使目标函数值增加 50；**

Lingo 的基本用法（6）

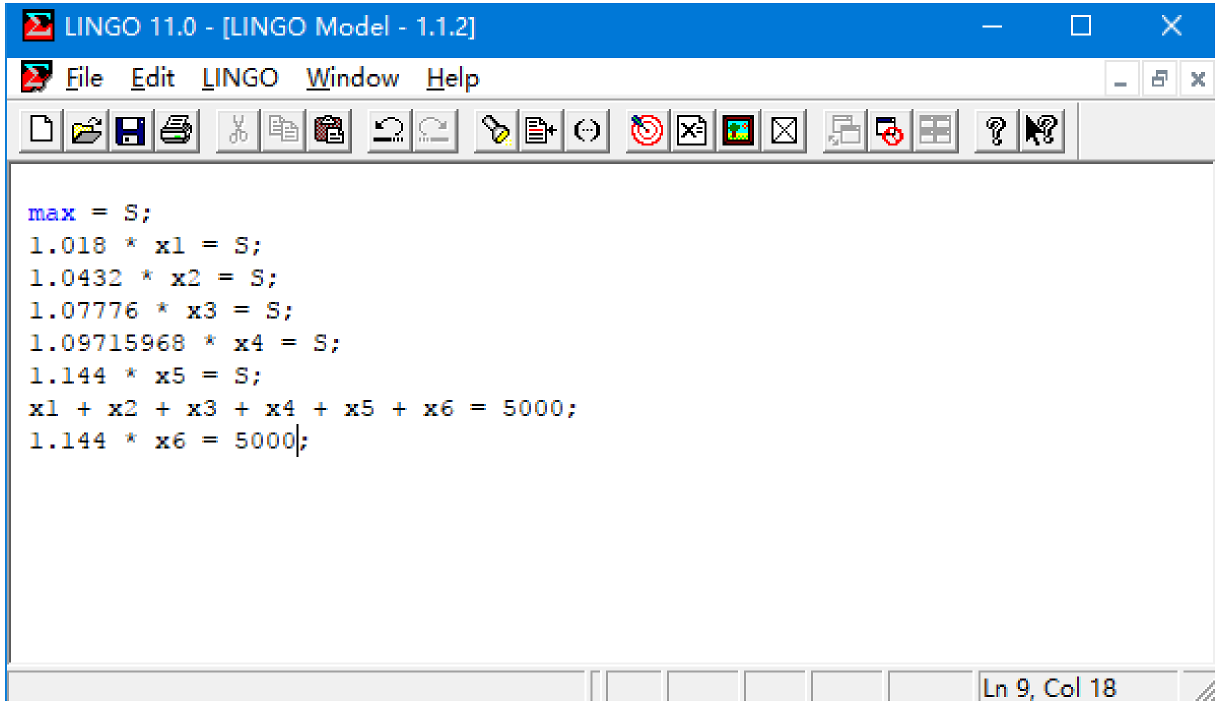
模型练习 1.2

$$\max \quad S$$

$$s.t. \quad \begin{cases} a_i x_i = S, i = 1, 2, \cdots, 5, \\ \sum_{i=1}^{6} x_i = 5000, \\ a_5 x_6 = 5000. \end{cases}$$

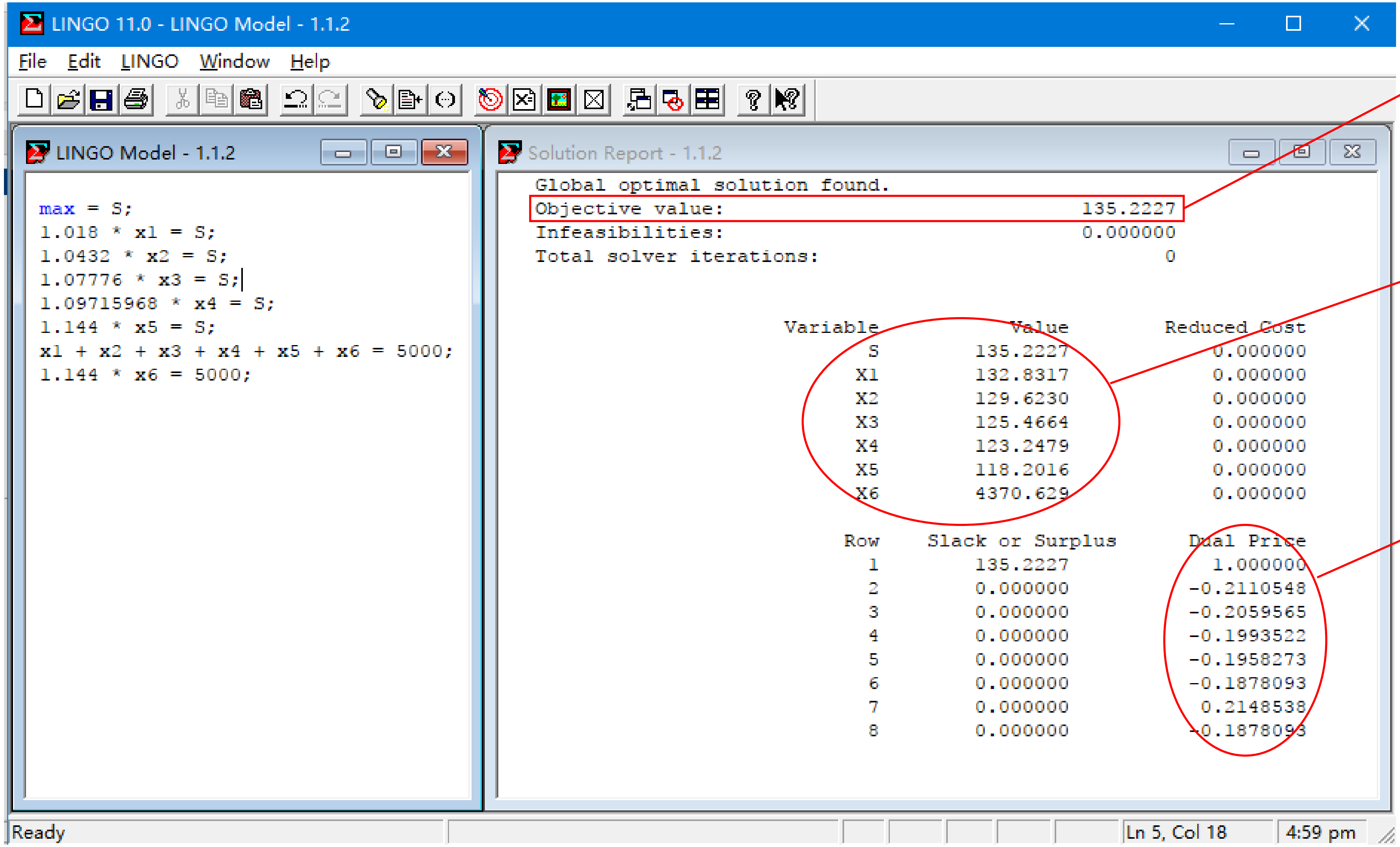
存期年限	1年	2年	3年	4年	5年
最有收益	1.018	1.0432	1.0776	1.09715968	1.144

Lingo 代码 1.2



Lingo 的基本用法（7）

运行结果



目标函数值

决策变量值

影子价格

PART TWO

编程语言

软件语言

(1)　Lingo 输入模型较直观，数学表达式可以直接输入；

(2)　针对变量多、约束条件个数也多的情况，Lingo 也提供了简单的有效的建模语言；

例题问题（1）

例 2.1：某公司有 6 个供货仓库，库存货物总数分别为 60,55,51,43,41,52，现有8个客户各要一批货，数量分别为 35,37,22,32,41,32,43,38，各货仓到 8 各客户处的单位货物运输价见下表。

表：供货栈到客户的单位货物运价（元/每单位）

	V1	V2	V3	V4	V5	V6	V7	V8
W1	6	2	6	7	4	2	5	9
W2	4	9	5	3	8	5	8	2
W3	5	2	1	9	7	4	3	3
W4	7	6	7	3	9	2	7	1
W5	2	3	9	5	7	2	6	5
W6	5	5	2	2	8	1	4	3

例题问题（2）

模型假设 2.1

解：引入符号：

x_{ij}：代表从第 i 个货栈到第 j 个客户的货物运量；

c_{ij}：表示从第 i 个货栈到第 j 个客户的单位货物运价；

a_i：表示第 i 个货栈的最大供货量；

d_j：表示第 j 个客户的订货量；

目标函数：总运输费用最少。

约束条件

（1） 各货栈运出的货物总量不超过其库存数；

（2） 各客户收到的货物总量等于其订货数量；

（3） 决策变量 x_{ij} 非负。

数学模型 2.1

$$\min \quad \sum_{i=1}^{6}\sum_{j=1}^{8} c_{ij}\cdot x_{ij}$$

$$s.t.\quad \begin{cases} \sum_{j=1}^{8} x_{ij} \le a_i, i=1,2,\cdots,6, \\ \sum_{i=1}^{6} x_{ij} = d_j, j=1,2,\cdots,8, \\ x_{ij} \ge 0, i=1,2,\cdots,6, j=1,2,\cdots,8. \end{cases}$$

集合定义部分（1）

集合（Set）

- **集合**：是一组相关对象构成的组合，代表模型中的实际事物，并与数学变量及常量联系起来。
  - **例**：6个仓库可以看成一个集合，8个客户可以看成另一个集合。

- **集合**：包括三方面内容
  - **集合名称**
  - **集合内的成员**：组成集合的个体、也称元素
  - **集合的属性**：可以看成是与该集合有关的变量或常量，相当于数组

集合定义部分（2）

本例先定义仓库集合：

WH / W1 .. W6 /: AI ;

其中：

- WH：集合的名称
- W1 .. W6 ：集合的成员，表明该集合有6个成员，分别对应于6个供货栈，“ .. ”是特定的
  略号，如果不用该省略号，也可以一一罗列出来，各成员之间用逗号，或空格分开
- AI：集合的属性，可以看成一个一维数组，有6个分量，分别表示各供货栈现有货物的总数

注：集合、成员、属性的命名规则与变量相同，可按自己的意愿，用有一定意义的字母、数字串来表示，式中“ / ”和“ /: ”是规定的语法规则。

集合定义部分（3）

本例还定义客户集合：

VD / V1 .. V8 /: DJ ;

该集合有8个成员，DJ 是集合的属性，有8个分量，表示各客户的需求量。

以上两个集合称为**初始集合**，或称基本集合，原始集合，初始集合的属性都相当于一维数组。

为了表示数学模型中从货栈到客户的运输关系以及与此相关的运输单价 c_{ij} 和运量 x_{ij} ，再定义一个表示运输关系路线的集合：

LINKS(WH, VD) : C, X;

该集合以初始集合 WH 和 VD 为基础，称为**衍生集合**，或称派生集合， C 和 X 是该衍生集合的两个属性。

集合定义部分（4）

衍生集合的定义语句由如下要素组成：

（1） 集合的名称；

（2） 对应的初始集合；

（3） 集合的成员，可以省略不写明；

（4） 集合的属性，可以没有；

注：

（1） 定义衍生集合时，可以用罗列的方式将衍生集合的成员一一罗列出来，如果省略不写，则默认衍生集合的成员取它所对应初始集合的所有可能的组合

（2） 衍生集合 LINKS 的定义中没有指明成员，则 LINKS 成员取 WH 和 VD 的所有可能的组合。

集合定义部分（5）

（3） 集合 LINKS 有48个成员，48个成员可以排列成一个矩阵，其行数与集合 WH 的成员个数相等，列数与集合 VD 的成员个数相等，其中WH 有6个成员，VD 有8个成员。

相应地，集合 LINKS 的属性 C 和 X 都相当于二维数组，各有 48 个分量，

C 表示货栈 W_i 到客户 V_j 的单位货物运价，

X 表示货栈 W_i 到客户 V_j 的货物运量。

集合定义部分（6）

本模型完整的集合定义为：

```
SETS:

    WH / W1 .. W6 / : AI;

    VD / V1 .. V8 / : DJ;

    LINKS( WH, VD ): C, X;

ENDSETS
```

注：

（1） 集合定义部分以语句 SETS： 开始，以语句 ENDSETS 结束，这两个语句必须单独成一行；

（2） ENDSETS 后面不加标点符号。

数据初始化——数据段（1）

上述集合中属性 X （有 48 个分量）是决策变量，是待求未知数，属性 AI， DJ， 和 C （分别有 6， 8， 48 个分量 ）都是已知数，Lingo 建模语言通过数据初始化部分来实现对已知属性赋予初始值，格式如右。

注：数据初始化部分以语句 DATA：开始，以语句 ENDDATA 结束，这两个语句必须单独成一行，数据之间用逗号，或空格。

```
DATA:
    AI = 60, 55, 51, 43, 41, 52;
    DJ = 35, 37, 22, 32, 41, 32, 43, 38;
     C = 6, 2, 6, 7, 4, 2, 5, 9
         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
```

目标函数（1）

目标函数表达式为： $\min \sum_{i=1}^{6}\sum_{j=1}^{8} c_{ij}\cdot x_{ij}$，用 Lingo 语句表达为：

```
min = @SUM( LINKS( i, j ) : C( i, j ) * X( i, j ) ) ;
```

- **@sum** 是 Lingo 提供的内部函数，其作用是对某个集合的所有成员，求指定表达式的和。

  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。

此处，@sum 的第一个参数是 Links(i, j), 表示求和运算对衍生集合 Links 进行，

- 运算规则：求表达式 C(i, j) * X(i, j) 的值，相当 $\sum_{i=1}^{6}\sum_{j=1}^{8} c_{ij}\cdot x_{ij}$

约束条件（1）

约束条件 $\sum_{j=1}^{8} x_{ij} \le a_i$，$(i=1, 2, \cdots, 6)$ 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(WH(i) : @SUM(VD(j) : X(i, j)) <= AI(i));

注释：

◆ @for：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**

➢ 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；

✓ @for 的第一个参数为 WH，它表示货栈，共有6个成员，故应生成6个约束表达式

约束条件（2）

- 第二个参数是是**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 VD 的8个成员，并且对表达式 X(i, j) 中的第二维 j 求和，即 $\sum_{j=1}^{8} x_{ij}$

注： @sum 和 @for 函数可以嵌套使用。

同样地，约束条件 $\sum_{i=1}^{6} x_{ij} \le d_j$，$(j = 1, 2, \cdots, 8)$，用 Lingo 语句可以写成：

```
@FOR( VD( j ) : @SUM( WH( i ) : X( i, j ) ) = DJ( j ) ) ;
```

完整的模型（1）

```
MODEL:

    SETS:
        WH / W1 .. W6 / : AI;
        VD / V1 .. V8 / : DJ;
        LINKS( WH, VD ): C, X;
    ENDSETS

    DATA:
        AI = 60, 55, 51, 43, 41, 52;
        DJ = 35, 37, 22, 32, 41, 32, 43, 38;
         C = 6, 2, 6, 7, 4, 2, 5, 9
             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

min =  @SUM( LINKS( i, j ) : C( i, j ) * X( i, j ) );
@FOR( WH( i ) : @SUM( VD( j ) : X( i, j ) ) <= AI( i ) );
@FOR( VD( j ) : @SUM( WH( i ) : X( i, j ) ) = DJ( j ) );

END
```

注:

（1）Lingo 模型以语句 Model: 开始，以语句 END 结束，这两个语句单独成一行;

（2）完整的模型由集合定义、数据段、目标函数和约束条件等组成，这几个部分的先后次序没关系;

（3）“！”开头的语句是注释语句（可有可无），若有必须要有“；”结束

完整的模型（2）

运行结果

Lingo 15.0 - [Solution Report - 1.2.1]

File Edit Solver Window Help

```
Global optimal solution found.
Objective value:                              664.0000
Infeasibilities:                              0.000000
Total solver iterations:                            15
Elapsed runtime seconds:                          0.44

Model Class:                                        LP

Total variables:                     48
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                   15
Nonlinear constraints:                0

Total nonzeros:                     144
Nonlinear nonzeros:                   0
```

For Help, press F1   Ln 1, Col 1

```
C( W6, V8)        3.000000        0.000000
X( W1, V1)        0.000000        5.000000
X( W1, V2)        19.00000        0.000000
X( W1, V3)        0.000000        5.000000
X( W1, V4)        0.000000        7.000000
X( W1, V5)        41.00000        0.000000
X( W1, V6)        0.000000        2.000000
X( W1, V7)        0.000000        2.000000
X( W1, V8)        0.000000        10.00000
X( W2, V1)        1.000000        0.000000
X( W2, V2)        0.000000
X( W2, V3)        0.000000
X( W2, V4)        32.00000
X( W2, V5)        0.000000
X( W2, V6)        0.000000
X( W2, V7)        0.000000
X( W2, V8)        0.000000
X( W3, V1)        0.000000
X( W3, V2)        11.00000
X( W3, V3)        0.000000
X( W3, V4)        0.000000
X( W3, V5)        0.000000
X( W3, V6)        0.000000
X( W3, V7)        40.00000
X( W3, V8)        0.000000
```

```
X( W3, V8)        0.000000        4.000000
X( W4, V1)        0.000000        4.000000
X( W4, V2)        0.000000        2.000000
X( W4, V3)        0.000000        4.000000
X( W4, V4)        0.000000        1.000000
X( W4, V5)        0.000000        3.000000
X( W4, V6)        5.000000        0.000000
X( W4, V7)        0.000000        2.000000
X( W4, V8)        38.00000        0.000000
X( W5, V1)        34.00000        0.000000
X( W5, V2)        7.000000        0.000000
X( W5, V3)        0.000000        7.000000
X( W5, V4)        0.000000        4.000000
X( W5, V5)        0.000000        2.000000
X( W5, V6)        0.000000        1.000000
X( W5, V7)        0.000000        2.000000
X( W5, V8)        0.000000        5.000000
X( W6, V1)        0.000000        3.000000
X( W6, V2)        0.000000        2.000000
X( W6, V3)        22.00000        0.000000
X( W6, V4)        0.000000        1.000000
X( W6, V5)        0.000000        3.000000
X( W6, V6)        27.00000        0.000000
X( W6, V7)        3.000000        0.000000
X( W6, V8)        0.000000        3.000000
```

完整的模型（3）

运行结果

最优运输方案

	V1	V2	V3	V4	V5	V6	V7	V8	合计
W1	0	19	0	0	41	0	0	0	60
W2	1	0	0	32	0	0	0	0	33
W3	0	11	0	0	0	0	40	0	51
W4	0	0	0	0	0	5	0	38	43
W5	34	7	0	0	0	0	0	0	41
W6	0	0	22	0	0	27	3	0	52
合计	35	37	22	32	41	32	43	38	

Lingo 语言的优点

（1）对大规模数学规划，Lingo 语言所建模较简洁，语句不多；

（2）模型易于扩展，因为@for @sum等语句并没有指定循环或求和的上下限，如果在集合定义部分增加集合成员的个数，则循环或求和自然扩展，不需要改动目标函数和约束条件；

（3）数据初始化部分与其他部分分开，对同一模型用不同数据来计算是，只需改动数据部分即可，其他语句不变；

（4）“集合”是Lingo很有特色的概念，集合的属性可以根据需要确定用多少个，可以用来代表已知量，也可以代表决策变量；

（5）使用了集合，以及@for，@sum等集合操作函数以后，可以用简洁的语句表达出常见的规划模型中的目标函数和约束条件，即使变量个数增加，组成模型的语句并不随之增加。

习题练习（2.1）

1. min $z = 13x_1 + 9x_2 + 10x_3 + 11x_4 + 12x_5 + 81x_6$,

$$
s.t.\begin{cases}
x_1 + x_4 = 400, \\
x_2 + x_5 = 600, \\
x_3 + x_6 = 500, \\
0.4x_1 + 1.1x_2 + x_3 \le 800, \\
0.5x_4 + 1.2x_5 + 1.3x_6 \le 900, \\
x_i \ge 0,\ i = 1, \cdots, 6.
\end{cases}
$$

习题解答（2.1）

```
min = 13 * x1 + 9 * x2 + 10 * x3 + 11 * x4
+ 12 * x5 + 81 * x6;

x1 + x4 = 400;
x2 + x5 = 600;
x3 + x6 = 500;
0.4 * x1 + 1.1 * x2 + x3 <= 800;
9.5 * x4 + 1.2 * x5 + 1.3 * x6 <= 900;
```

```
Global optimal solution found.
Objective value:                              16904.71
Infeasibilities:                              0.000000
Total solver iterations:                             2
Elapsed runtime seconds:                          0.50

Model Class:                                        LP

Total variables:                      6
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    6
Nonlinear constraints:                0

Total nonzeros:                      18
Nonlinear nonzeros:                   0


                        Variable           Value        Reduced Cost
                              X1        363.2899            0.000000
                              X2        140.6219            0.000000
                              X3        500.0000            0.000000
                              X4        36.71013            0.000000
                              X5        459.3781            0.000000
                              X6        0.000000            68.34403

                             Row    Slack or Surplus      Dual Price
                               1        16904.71           -1.000000
                               2        0.000000           -14.23972
                               3        0.000000           -12.40923
                               4        0.000000           -13.09930
                               5        0.000000            3.099298
                               6        0.000000           0.3410231
```

习题练习（2.2）

2. 某公司有3个供货栈，库存货物总数分别为21, 12，27，现有4个客户各需要一批货，数量分别为9， 18， 15， 18，各供货栈到4个客户处的单位货物运输价如表所示（元/每单位）。试确定各货栈到各客户的货物调运数量，使总的运输费用最小。

	V1	V2	V3	V4	库存
w1	6	22	6	20	21
w2	2	18	4	16	22
w3	14	8	20	10	27
需求	9	18	15	18	

习题解答（2.2-1）

模型假设

解：引入符号：

x_{ij}：表示从第 i 个货栈到第 j 个客户的货物运量；

c_{ij}：表示从第 i 个货栈到第 j 个客户的单位货物运价；

a_i：表示第 i 个货栈的最大供货量；

d_j：表示第 j 个客户的订货量。

目标函数：总运输费最少

约束条件：

（1） 各货栈运出的货物总量不超过其库存数；

（2） 各客户收到的货物总量等于其订货数量；

（3） 决策变量 x_{ij} 非负。

数学模型 2.2

$$\min \sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}\cdot x_{ij}$$

$$s.t.\begin{cases} \sum_{i=1}^{4} x_{ij} = d_j,\ j=1,\ 2,\ 3,\ 4, \\ \sum_{j=1}^{4} x_{ij} \le a_i,\ j=1,\ 2,\ 3, \\ x_{ij} \ge 0,\ i=1,\ 2,\ 3,\ j=1,\ 2,\ 3,\ 4. \end{cases}$$

习题解答（2.2-2）

```
Model :
	sets:
		WH / W1, W2, W3 /: AI;
		VD / V1, V2, V3, V4 /: DJ;
		Link( WH, VD ) : C, X;
	endsets  ! 定义集合;

	data:
		AI = 21, 22, 27;
		DJ = 9, 18, 15, 18;
		C = 6, 22, 6, 20
		     2, 18, 4, 16
		    14, 8, 20, 10;
	enddata  ! 已知集合赋初始值;

	min = @sum( link( i, j ) : C( i, j ) * X( i, j ) ); ! 目标函数;

	@for( VD( j ) : @sum( WH( i ) : X( i, j ) ) = DJ( j ) ); ! 约束条件;
	@for( WH( i ) : @sum( VD( j ) : X( i, j ) ) <= AI( i ) ); ! 约束条件;
End
```

习题解答（2-3）

```
Global optimal solution found.
Objective value:                              478.0000
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          7.89

Model Class:                                        LP

Total variables:                     12
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    8
Nonlinear constraints:                0

Total nonzeros:                      36
Nonlinear nonzeros:                   0
```

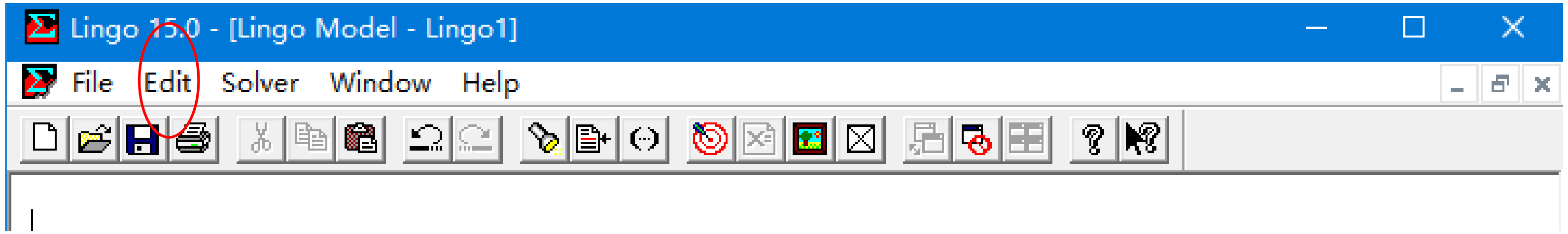
```
X( W1, V1)        0.000000        2.000000
X( W1, V2)        0.000000        6.000000
X( W1, V3)        11.00000        0.000000
X( W1, V4)        0.000000        2.000000
X( W2, V1)        9.000000        0.000000
X( W2, V2)        0.000000        4.000000
X( W2, V3)        4.000000        0.000000
X( W2, V4)        9.000000        0.000000
X( W3, V1)        0.000000        18.00000
X( W3, V2)        18.00000        0.000000
X( W3, V3)        0.000000        22.00000
X( W3, V4)        9.000000        0.000000

       Row    Slack or Surplus      Dual Price
         1        478.0000          -1.000000
         2        0.000000          -4.000000
         3        0.000000          -16.00000
         4        0.000000          -6.000000
         5        0.000000          -18.00000
         6        10.00000          0.000000
         7        0.000000          2.000000
         8        0.000000          8.000000
```

PART THREE

软件介绍

编辑菜单（1）



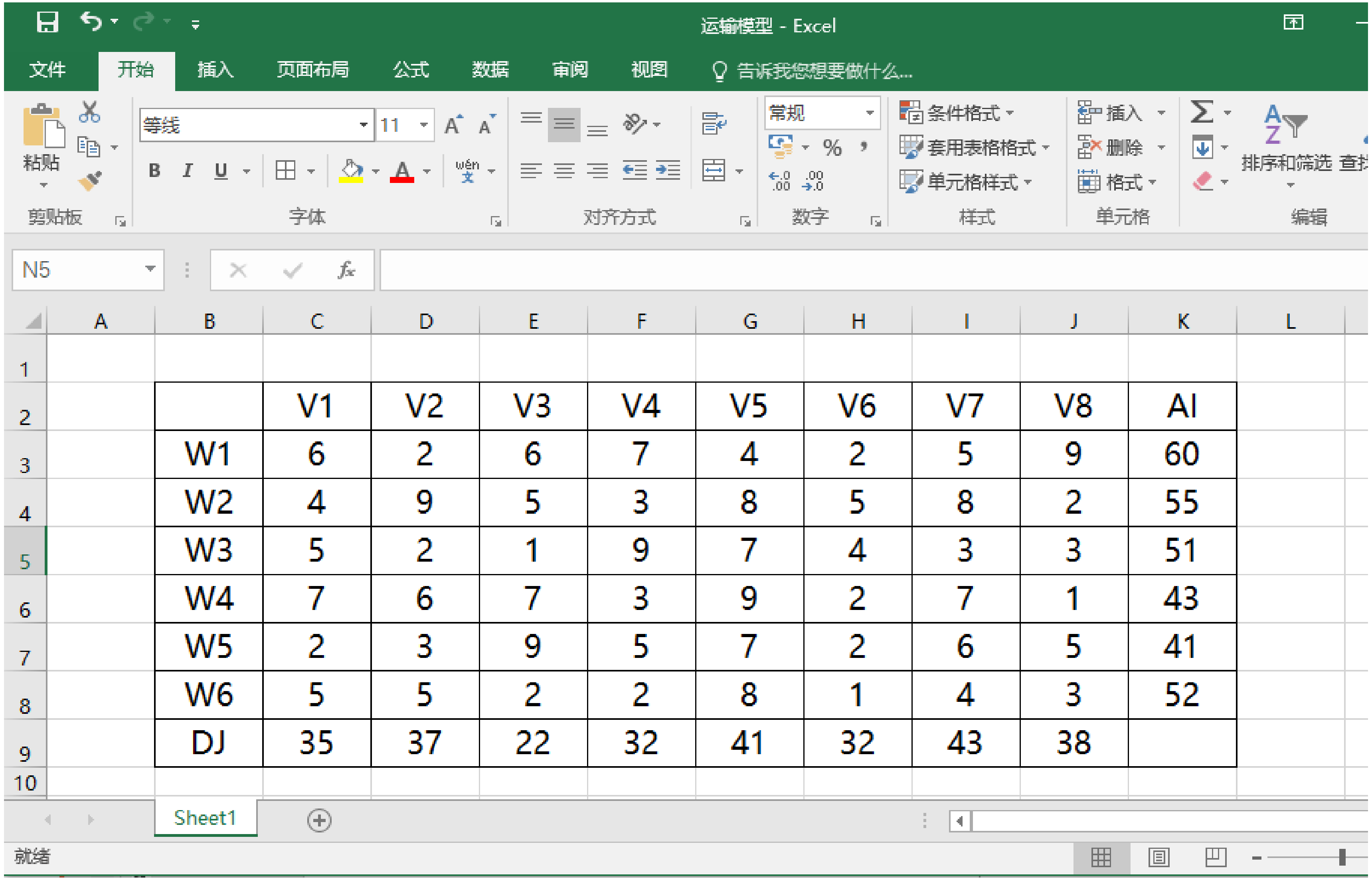
◆ 选择性粘贴（Paste Special）

该命令是把Windows剪粘板中的内容插入到光标所在位置。

例：将上例中运输模型的已知数据保存到名为“运输模型 3.1.xlsx”的Excel文件中，然后按以下步骤操作：

（1）在Excel文件中用鼠标选中数据表所在的区域 B2：K9，点击“复制”将选中的内容复制到Windows剪粘板中，如下图，

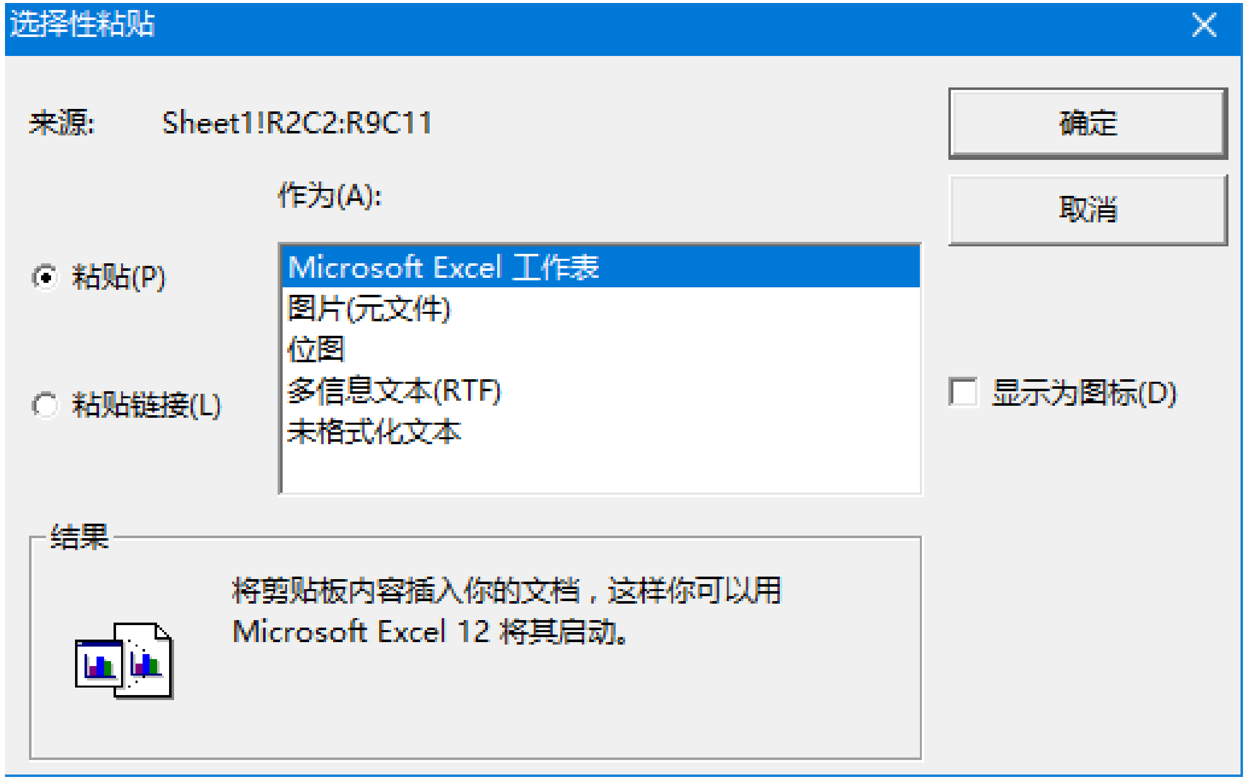
编辑菜单（2）



	V1	V2	V3	V4	V5	V6	V7	V8	AI
W1	6	2	6	7	4	2	5	9	60
W2	4	9	5	3	8	5	8	2	55
W3	5	2	1	9	7	4	3	3	51
W4	7	6	7	3	9	2	7	1	43
W5	2	3	9	5	7	2	6	5	41
W6	5	5	2	2	8	1	4	3	52
DJ	35	37	22	32	41	32	43	38	

编辑菜单（3）

（2）回到 Lingo 的运输模型窗口，将光标定位到数据段，点击 Paste Special 命令，弹出如下图所示“选择性粘贴”对话框。

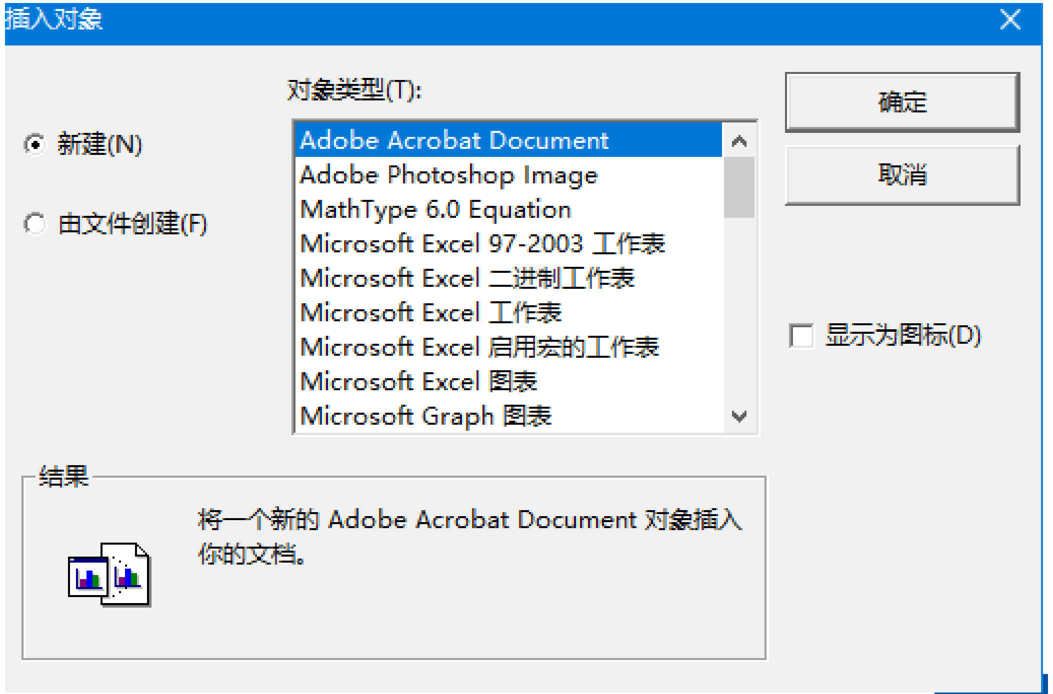


编辑菜单（4）

- **粘贴：** 在 Lingo 中光标所在位置插入剪粘板的当前内容，以后如果原始数据文件（“运输模型.xls”）中的数据发生变化，插入的内容并不随之改变；
- **粘贴链接：** 在 Lingo 中光标所在位置插入的内容与原始文件直接建立了链接关系，当你修改原始文件中的数据时， Lingo 模型中的插入内容也会随之变化，即能够自动更新；

  注意： 可以与外部对象文件的一部分建立链接；

  - Insert New Object：选择“由文件创建”，可以与整个对象建立链接。



Lingo 菜单—灵敏度分析（Range）（1）

(1）最优解保持不变的情况下，目标函数的系数的变化范围；

(2）在影子价格和缩减成本系数都不变的前提下，约束条件右边的常数的变化范围；

- **激活**：Lingo　Opions　General Solver　Dual Computations　Prices and Ranges；
- **运行**：Lingo　Solver　Solve；
- **再运行**：Lingo　Solver　Range
- 灵敏度分析耗费相当多的求解时间，因此当速度很关键时，就没有必要激活它；

Lingo 菜单—灵敏度分析（Range）（2）

Lingo Model - 1.1.1

```
max = 200 * x1 + 300 * x2;
x1 <= 100;
x2 <= 120;
x1 + 2 * x2 <= 160;
x1 >= 0;
x2 >= 0;
```

Range Report - 1.1.1

Ranges in which the basis is unchanged:

Objective Coefficient Ranges:

Variable	Current Coefficient	Allowable Increase	Allowable Decrease
X1	200.0000	INFINITY	50.00000
X2	300.0000	100.0000	300.0000

Righthand Side Ranges:

Row	Current RHS	Allowable Increase	Allowable Decrease
2	100.0000	60.00000	100.0000
3	120.0000	INFINITY	90.00000
4	160.0000	180.0000	60.00000
5	0.000000	100.0000	INFINITY
6	0.000000	30.00000	INFINITY

目标函数系数的变化范围

约束条件右边常数的变化范围

Lingo 菜单—灵敏度分析（Range）（3）

Ranges in which the basis is unchanged:

Objective Coefficient Ranges:

Variable	Current Coefficient	Allowable Increase	Allowable Decrease
X1	200.0000	INFINITY	50.00000
X2	300.0000	100.0000	300.0000

Righthand Side Ranges:

Row	Current RHS	Allowable Increase	Allowable Decrease
2	100.0000	60.00000	100.0000
3	120.0000	INFINITY	90.00000
4	160.0000	180.0000	60.00000
5	0.000000	100.0000	INFINITY
6	0.000000	30.00000	INFINITY

最优解保持不变

$x_1 \geq 150,\ 0 \leq x_2 \leq 400.$

第三列和第四列是在影子价格和缩减成本系数都不变的前提下，约束条件右边的允许上调界限以及允许下调的界限。

Lingo 菜单—Window|Status Window（1）

(1) Model Class 当前模型的类型

- LP（线性规划）、QP（二次规划）、ILP（整数线性规划）、IQP（整数二次规划）、
- PILP（纯整数线性规划）、PIQP（纯整数二次规划）、NLP（非线性规划）、
- MIP（混合整数规划）、INLP（整数非线性规划）、PINLP（纯整数非线性规划）

注:

- ✓ 以 I 开头表示 IP（整数规划），以 PI 开头表示 PIP（纯整数规划）

Lingo 菜单—Window|Status Window（1）

（2）State 当前的状态

- 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）、
- Interrupted（中断）、Undetermined（未确定）

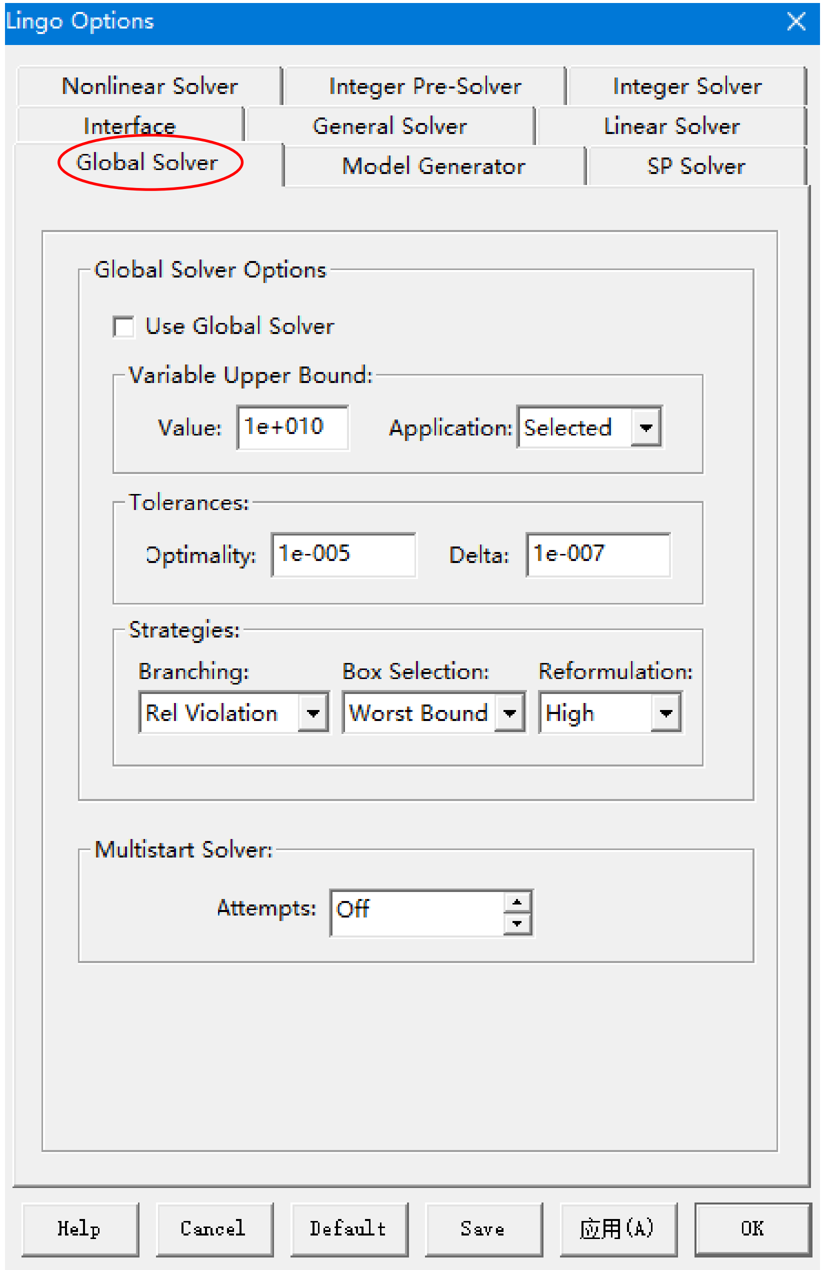
注：

- ✓ （1）如果模型没有非线性约束，那么局部最优解也就是全局最优解；
- ✓ （2）如果模型有一个或多个非线性约束，那么局部最优解不一定就是全局最优解，也许存在一个比目前找到的最优解更优的解，只是算法找不着它。

Lingo 的参数设置

◆ Lingo　Options

- Interface（界面）、General Solver（通用求解器）、Linear Solver（线性求解器）、Nonlinear Solver（非线性求解器）、
- Integer Pre-Solver（整数预处理程序）、Integer Solver（整数求解器）、Global Solver（全局求解器）、Model Generator（模型生成器）、SP Solver（随机规则求解器）



Lingo 的参数设置——Global Solver（1）

例 3.1：求函数 f(x) = xcos(3x) 在区间 [0, 6.25) 内的最小值。

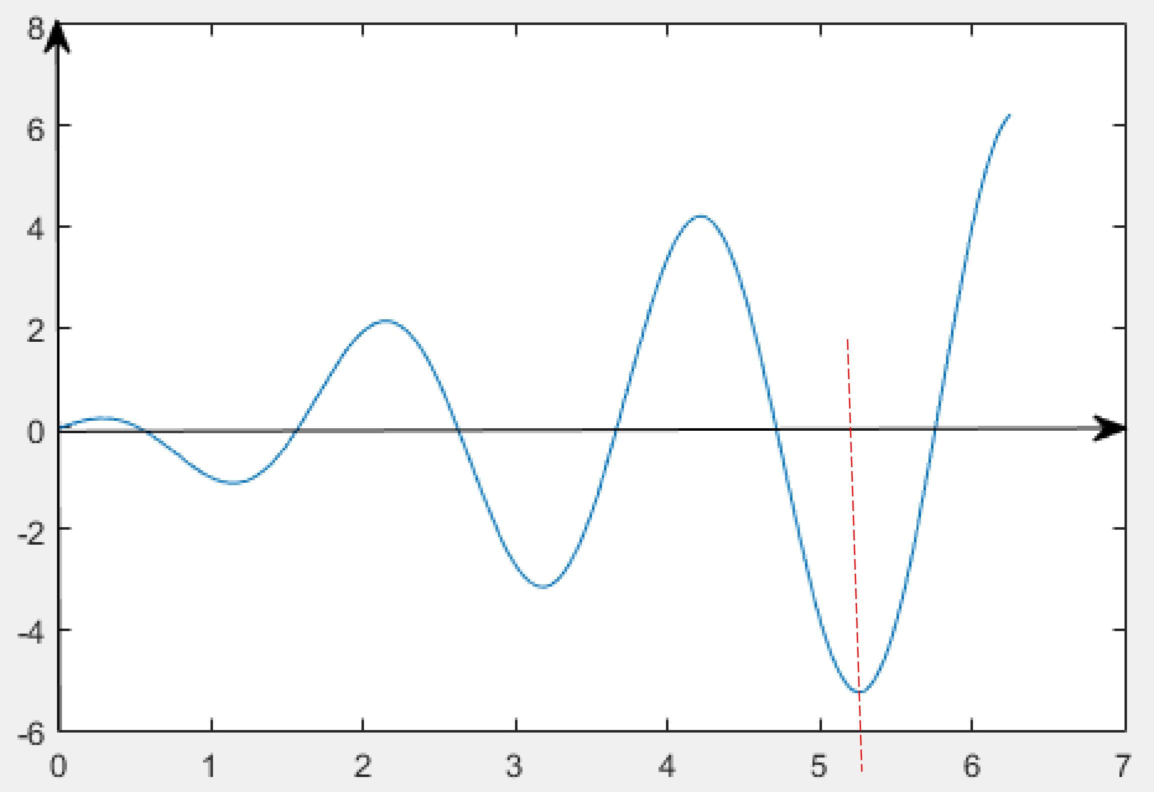
代码 3.1:

```
Model:

    min = x * @cos( 3 * x );

    @bnd( 0, x, 6.25 );

end
```



Lingo 的参数设置——Global Solver（2)

（1）不选择Use Global Solver, 并且多初始点求解程序的尝试次数设置为off

```
Local optimal solution found.
Objective value:                          -1.096124
Infeasibilities:                           0.000000
Total solver iterations:                          7
Elapsed runtime seconds:                       0.11

Model Class:                                    NLP

Total variables:                    1
Nonlinear variables:                1
Integer variables:                  0

Total constraints:                  1
Nonlinear constraints:              1

Total nonzeros:                     1
Nonlinear nonzeros:                 1


                     Variable           Value        Reduced Cost
                            X        1.141873      -0.5631142E-08

                          Row    Slack or Surplus      Dual Price
                            1           -1.096124       -1.000000
```

（2）选择Use Global Solver, 并且多初始点求解程序的尝试次数设置为off

```
Global optimal solution found.
Objective value:                          -5.246559
Objective bound:                          -5.246559
Infeasibilities:                           0.000000
Extended solver steps:                            1
Total solver iterations:                         34
Elapsed runtime seconds:                       0.42

Model Class:                                    NLP

Total variables:                    1
Nonlinear variables:                1
Integer variables:                  0

Total constraints:                  1
Nonlinear constraints:              1

Total nonzeros:                     1
Nonlinear nonzeros:                 1


                     Variable           Value        Reduced Cost
                            X        5.257095            0.000000

                          Row    Slack or Surplus      Dual Price
                            1           -5.246559       -1.000000
```

Lingo 的参数设置——Global Solver（3）

（3）不选择Use Global Solver, 并且多初始点求解程序的尝试次数设置为 Solver Decides

```
Local optimal solution found.
Objective value:                              -5.246559
Infeasibilities:                               0.000000
Extended solver steps:                                5
Best multistart solution found at step:               2
Total solver iterations:                             31
Elapsed runtime seconds:                           0.29

Model Class:                                        NLP

Total variables:                      1
Nonlinear variables:                  1
Integer variables:                    0

Total constraints:                    1
Nonlinear constraints:                1

Total nonzeros:                       1
Nonlinear nonzeros:                   1



                                Variable           Value        Reduced Cost
                                       X        5.257095            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1       -5.246559           -1.000000
```

该例子说明：

- 选择 Use Global Solver，或设置多初始点求解程序的尝试次数为 Solver Decides，两种情况都得到了全局最优解，只是求解时间不同而已。对于规模较小的模型，求解时间不会成为问题。

Lingo 的参数设置——Global Solver（4）

◆ 测试 lingo 11 与 lingo 18 之间区别

例： MAX = @SIN(R) / R + 1;

R = @SQRT((X – 50) ^ 2 + (Y – 50) ^ 2) + 2.71828;

@BND(0, X, 100);

@BND(0, Y, 100);

lingo 11 选 Global Solver

```
Global optimal solution found.
Objective value:                              1.128375
Objective bound:                              1.128375
Infeasibilities:                              0.000000
Extended solver steps:                               3
Total solver iterations:                           126


                       Variable           Value        Reduced Cost
                              R        7.725252            0.000000
                              X        50.31850            0.000000
                              Y        45.00317            0.000000

                            Row    Slack or Surplus      Dual Price
                              1        1.128375            1.000000
                              2        0.000000            0.000000
```

lingo 18 选 Global Solver

```
Global optimal solution found.
Objective value:                              1.151119
Objective bound:                              1.151119
Infeasibilities:                              0.000000
Extended solver steps:                               5
Total solver iterations:                           145
Elapsed runtime seconds:                          0.33

Model Class:                                       NLP

Total variables:                      3
Nonlinear variables:                  3
Integer variables:                    0

Total constraints:                    2
Nonlinear constraints:                2

Total nonzeros:                       4
Nonlinear nonzeros:                   3


                                Variable           Value        Reduced Cost
                                       R        2.718280            0.000000
                                       X        50.00000            0.000000
                                       Y        50.00000            0.000000
```

Lingo 的常用运算符（1）

1. 算术运算符

- ^ 次方， * 乘， / 除， + 加， - 减

  （1）双目运算符，需要两个运算对象；

  - 但，“ - ”号也可以作为单目运算符，表示取运算对象的负值；

  （2）优先级：单目“ - ”最高，其余依次为 ^, * 和 /, + 和 -，同级自左至右，加括号可以改变运算次序。

Lingo 的常用运算符（2）

2. 逻辑运算符

- #EQ#, #NE#, #GT#, #GE#, #LT#, #LE#, #NOT#, #AND#, #OR#

  (1) #NOT# 为单目运算符；

  (2) 其余都是双目运算符，需要两个运算对象，中间用逻辑运算符连接起来，构成逻辑表达式；

  (3) 逻辑值：真（TRUE）＝1，或假（FLASE）＝0；

  (4) 优先级：大部分都是平级的；

  - 最高：#NOT#，
  - 最低：#AND#, #OR#

Lingo 的常用运算符（3 - 1)

3. 关系运算符

- =, <=, >=

  (1) 没有单独的“>”, “<”关系；

  - 如果出现单个“>”或“<”，Lingo 认为是省略了“=”号，即“<”等同于“<=”，“>”等同于“>=”；

  (2) 如果模型中确实需要严格小于，或者严格大于，

  - 如 A < B, 那么就可以把它变成：A + ε <= B, 这里的 ε 是一个很小的正数，它的值依赖于模型中 A 小于 B 多少才算不等；

  (3) 优先级：单目优于双目，算术优于逻辑，逻辑优于关系，平级从左至右，括号改变次序；

Lingo 的常用运算符（3 - 2）

示例讲解

◆ 例： 逻辑运算符示例

```
2 #gt# 3 #and# 4 #gt# 2;
```

其结果为假，值为0.

➢ 建议代码写成：

```
( 2 #gt# 3 ) #and# ( 4 #gt# 2 );
```

数学函数

（1）所有函数都以字符 @ 开头；

（2）数学函数中的角度单位为弧度；

（3）常用函数：

- @sin(x), @cos(x), @tan(x);
- @ABS(x) , @log(x), @exp(x), @sign(x), @floor(x), @lgm(x);
- @smax(x1, x2, ..., xn), @smin(x1, x2, ..., xn);
- @mod(x, y), @pow(x, y), @sqr(x), @sqrt(x);

（4）lingo 提供了50多个内部函数；

集合操作函数（1）

@for(s : e)	对集合 s 中的每一个成员都生成一个约束条件表达式，表达式的具体形式由参数 e 描述；
@sum(s : e)	对集合 s 中的每个成员，分别得到表达式 e 的值，然后返回所有这些值得和；
@max(s : e)	对集合 s 中的每个成员，分别得到表达式 e 的值，然后返回所有这些值中的最大值；
@min(s : e)	对集合 s 中的每个成员，分别得到表达式 e 的值，然后返回所有这些值中的最小值；
@prod(s : e)	对集合 s 中的每个成员，分别得到表达式 e 的值，然后返回所有这些值得乘积；
@size(s)	返回集合 s 中的成员个数；
@index(s : ek)	返回成员 ek 在集合 s 中的顺序号，该值在 1 和集合 s 的成员个数之间，如果不在集合 s 中，则给出出错信息；
@in(s : et)	如果成员 et 在集合 s 中，则返回 1; 否则返回 0;
@wrap(I, N)	在集合循环函数中，利用 @wrap 重新调回，如： @wrap(3, 10) 返回值为 3；@wrap(54, 10) 返回值为 4； @wrap(29, 6) 返回值为 5；@wrap(30, 6) 返回值为 6

集合操作函数（2）

（1）函数形式：

@函数名（集合名|条件： 表达式）；

- 参数之间用“：”号隔开，
- “|”后面的条件可有可无，若有，通常是逻辑表达式，表示满足一定条件的情况下进行操作；

例： 集合 VD 中有 8 个成员，但对应的约束条件不足 8 个，即对该集合中某些成员（如索引为 5 的成员），不要生成约束条件，则代码可以修正为：

@for(VD(J) | J #NE# 5 : 表达式 e);

- 代码中的条件“ J #NE# 5 ”表示对集合 VD，当成员序号 J 不等于 5 时，生成约束表达式

e 即第 5 个成员不生成约束表达式。

集合操作函数（3-1）

例 3.2 员工时序安排模型. 某项工作一周 7 天都需要有人上班，周一至周日所需的最少人数分别为 20、16、13、16、19、14 和 12. 要求员工一周连续工作 5 天，然后休息 2 天，试求每周所需最少总人数，并给出安排（注意这是稳定后的情况）。

集合操作函数（3-2）

解： 周一至周日分别安排 $X(i)$, $i=1, 2, \cdots, 7$ 个人上班，其中周一上班的人，必定周六、周日休息，一次类推。合理安排轮休可使既满足每天所需的最少人数，又使总人数最少。

设总人数为 Z，在周一有多少人上班呢？除了周二和周三开始上班的人正处在休息中，其他人都在上班，在岗人数为：$Z-X(2)-X(3)$，应满足不等式：

$$Z-X(2)-X(3)\geq R(1),$$

式中 $R(1)$ 是周一所需最少上岗人数，依次类推。

目标函数：$\min\ Z=\sum_{i=1}^{7}X_i$

约束条件：$Z-X(i+1)-X(i+2)\geq R(i)$, $i=1, 2, \cdots, 7.$

集合操作函数（3-3）

```
model:
    sets:
        days / mon .. sun /: R, X;
    endsets

    data:
        R = 20, 16, 13, 16, 19, 14, 12;
    enddata

    min = Z;
    Z = @sum( days : X );
    N = @size( days );
    @for( days( i ) : Z - X( @wrap( i + 1, N ) )
    - X( @wrap( i + 2, N ) )  >= R( i ) );
end
```

- 注：@wrap(i + 1, 7)，可以把 8 转换成 1，9 转换成 2；

```
Global optimal solution found.
Objective value:                              22.00000
Infeasibilities:                              0.000000
Total solver iterations:                             5
Elapsed runtime seconds:                          0.06

Model Class:                                        LP

Total variables:                      8
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    9
Nonlinear constraints:                0

Total nonzeros:                      30
Nonlinear nonzeros:                   0
```

Variable	Value	Reduced Cost
Z	22.00000	0.000000
N	7.000000	0.000000
R(MON)	20.00000	0.000000
R(TUE)	16.00000	0.000000
R(WED)	13.00000	0.000000
R(THU)	16.00000	0.000000
R(FRI)	19.00000	0.000000
R(SAT)	14.00000	0.000000
R(SUN)	12.00000	0.000000
X(MON)	8.000000	0.000000
X(TUE)	2.000000	0.000000
X(WED)	0.000000	0.000000
X(THU)	6.000000	0.000000
X(FRI)	3.000000	0.000000
X(SAT)	3.000000	0.000000
X(SUN)	0.000000	0.3333333

变量定界函数（1-1）

@bin(X)	限制 X 为 0 或 1， 该函数在 0-1 规划中特别有用
@bnd(L, X, U)	限制 L <= X <= U，可用作约束条件
@gin(X)	限制 X 为整数，该函数在整数规划中特别有用
@free(X)	取消对变量 X 的限制，即 X 可取任意实数值

例 3.2（新） 在例 3.2中，如果周一至周日所需的最少人数分别为 20, 12, 18, 16, 19, 14 和 12. 重新求解。

解： 需要增加整数规划约束，增加语句：

```
@for( days : @gin( X ) );
```

变量定界函数（1-2）

➢ 未做约束

```
Global optimal solution found.
Objective value:                              23.66667
Infeasibilities:                              0.000000
Total solver iterations:                             5
Elapsed runtime seconds:                          0.06

Model Class:                                        LP

Total variables:                      8
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    9
Nonlinear constraints:                0

Total nonzeros:                      30
Nonlinear nonzeros:                   0


                                Variable           Value        Reduced Cost
                                       Z        23.66667            0.000000
                                       N        7.000000            0.000000
                                R( MON)         20.00000            0.000000
                                R( TUE)         12.00000            0.000000
                                R( WED)         18.00000            0.000000
                                R( THU)         16.00000            0.000000
                                R( FRI)         19.00000            0.000000
                                R( SAT)         14.00000            0.000000
                                R( SUN)         12.00000            0.000000
                                X( MON)         9.666667            0.000000
                                X( TUE)         2.000000            0.000000
                                X( WED)         1.666667            0.000000
                                X( THU)         5.666667            0.000000
                                X( FRI)         0.000000            0.000000
                                X( SAT)         4.666667            0.000000
                                X( SUN)         0.000000           0.3333333
```

➢ 做整数约束

```
Global optimal solution found.
Objective value:                              24.00000
Objective bound:                              24.00000
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                            11
Elapsed runtime seconds:                          0.31

Model Class:                                      MILP

Total variables:                      8
Nonlinear variables:                  0
Integer variables:                    7

Total constraints:                    9
Nonlinear constraints:                0

Total nonzeros:                      30
Nonlinear nonzeros:                   0


                                Variable           Value        Reduced Cost
                                       Z        24.00000            0.000000
                                       N        7.000000            0.000000
                                R( MON)         20.00000            0.000000
                                R( TUE)         12.00000            0.000000
                                R( WED)         18.00000            0.000000
                                R( THU)         16.00000            0.000000
                                R( FRI)         19.00000            0.000000
                                R( SAT)         14.00000            0.000000
                                R( SUN)         12.00000            0.000000
                                X( MON)         10.00000            1.000000
                                X( TUE)         2.000000            1.000000
                                X( WED)         1.000000            1.000000
                                X( THU)         6.000000            1.000000
                                X( FRI)         0.000000            1.000000
                                X( SAT)         5.000000            1.000000
                                X( SUN)         0.000000            1.000000
```

变量定界函数（2-1)

例 3.3 背包问题. 某人打算外出旅游并登山，路程比较远，途中要坐火车和飞机，考虑要带许多必要的旅游和生活用品，如照相机、摄像机、食品、衣服、雨具、书籍等，共 n 件物品，重量分别为 a_i，而受航空行李重量限制，以及个人体力所限，能带的行李总重量为 b，n 件物品的总重量超过了 b，需要裁剪，该旅行者为了决策带哪些物品，对这些物品的重要性进行了量化，用 c_i 表示，试建立该问题的数学模型。这个问题称为背包问题。

变量定界函数（2-2）

解：引入 0-1 型决策变量 x_i, $i=1, 2, \cdots, 8$,

- $x_i=1$：表示物品 i 放入背包中；
- $x_i=0$：表示物品 i 未放入背包中；
- 重量 a（千克）：1, 3, 4, 3, 3, 1, 5, 10；
- 价值 c（元）：2, 9, 3, 8, 10, 6, 4, 10；
- 总重量限制不超过 b = 15 千克

0-1 线性规划：

$$\max \quad z=\sum_{i=1}^{8} c_i \cdot x_i,$$

$$s.t. \quad \begin{cases} \sum_{i=1}^{8} a_i \cdot x_i \le b, \\ x=1, \text{或 } 0, \ i=1, 2, \cdots, 8. \end{cases}$$

变量定界函数（2-3）

```
model:

    sets:
        WP / W1 .. W8 /: A, C, X;
    endsets

    data:
        A = 1, 3, 4, 3, 3, 1, 5, 10;
        C = 2, 9, 3, 8, 10, 6, 4, 10;
    enddata

    max = @sum( WP( i ) : C( i ) * X( i ) );

    @sum( WP( i ) : A( i ) * X( i ) ) <= 15;
    @for( WP( i ) : @bin( X( i ) ) );

end
```

```
Global optimal solution found.
Objective value:                              38.00000
Objective bound:                              38.00000
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                             0
Elapsed runtime seconds:                          0.16

Model Class:                                      PILP

Total variables:                      8
Nonlinear variables:                  0
Integer variables:                    8

Total constraints:                    2
Nonlinear constraints:                0

Total nonzeros:                      16
Nonlinear nonzeros:                   0
```

X(W1)	1.000000	-2.000000
X(W2)	1.000000	-9.000000
X(W3)	1.000000	-3.000000
X(W4)	1.000000	-8.000000
X(W5)	1.000000	-10.00000
X(W6)	1.000000	-6.000000
X(W7)	0.000000	-4.000000
X(W8)	0.000000	-10.00000

Row	Slack or Surplus	Dual Price
1	38.00000	1.000000
2	0.000000	0.000000

THANKS