数学模型与lingo软件

西南交通大学数学系

需要掌握的几个重要方面

- 掌握集合(SETS)的应用；
- 正确阅读求解报告；
- 正确理解求解状态窗口；
- 学会设置基本的求解选项(OPTIONS) ；
- 应用实例

LINGO 8.0有两种命令模式

Windows 模式，通过下拉式菜单命令驱动LINGO 运行

命令行(Command-Line) 模式,仅在命令窗口下操作

与LINDO 相比，LINGO 软件主要具有两大优点

1、除具有LINDO 的全部功能外，还可用于求解非线性规划问题，包括非线性整数规划问题

2、LINGO 包含了内置的建模语言，允许以简练、直观的方式描述较大规模的优化问题，模型中所需的数据可以以一定格式保存在独立的文件中

LP问题在lindo和lingo中不同的输入形式

Lindo:

max 2x+3y

st

4x+3y<10

3x+5y<12

end

Lingo:

```
max=2*x+3*y;
 4*x+3*y<10;
 3*x+5*y<12;
```

这是LINGO 模型的最基本特征

（1） 将目标函数的表示方式从"MAX"变成了"MAX="

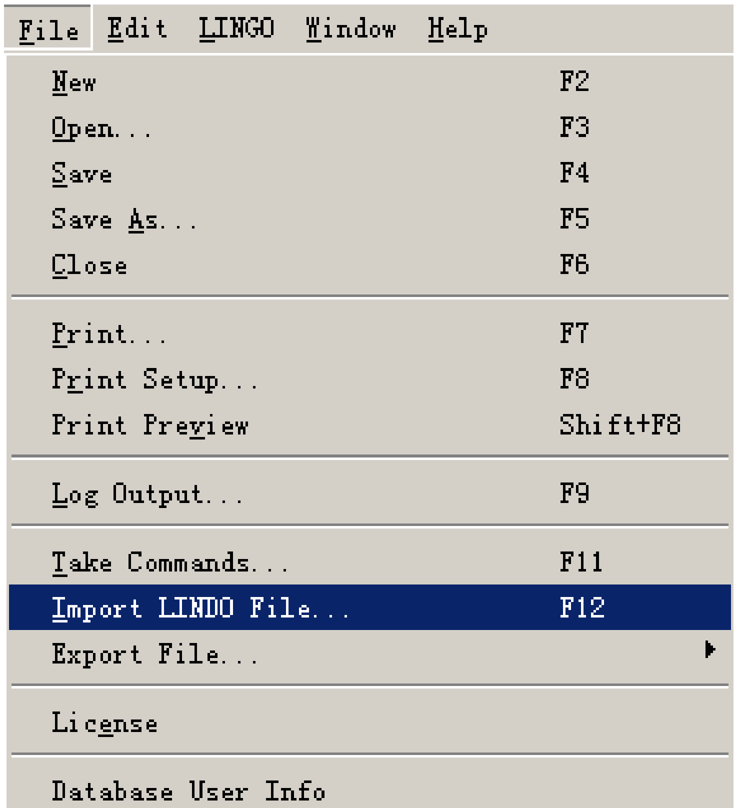
（2） "ST"在LINGO 模型中不再需要，所以被删除了

（3） 每个系数与变量间增加了运算符"*"（即乘号不能省略）

（4） 每行（目标、约束和说明语句）后面均增加了一个分号";"

（5） 模型结束标志"END"也被删除了（LINGO 中只有当模型以"MODEL："开始时才能以"END" 结束）。

直接将lindo模型文件转化为lingo文件



Lindo:

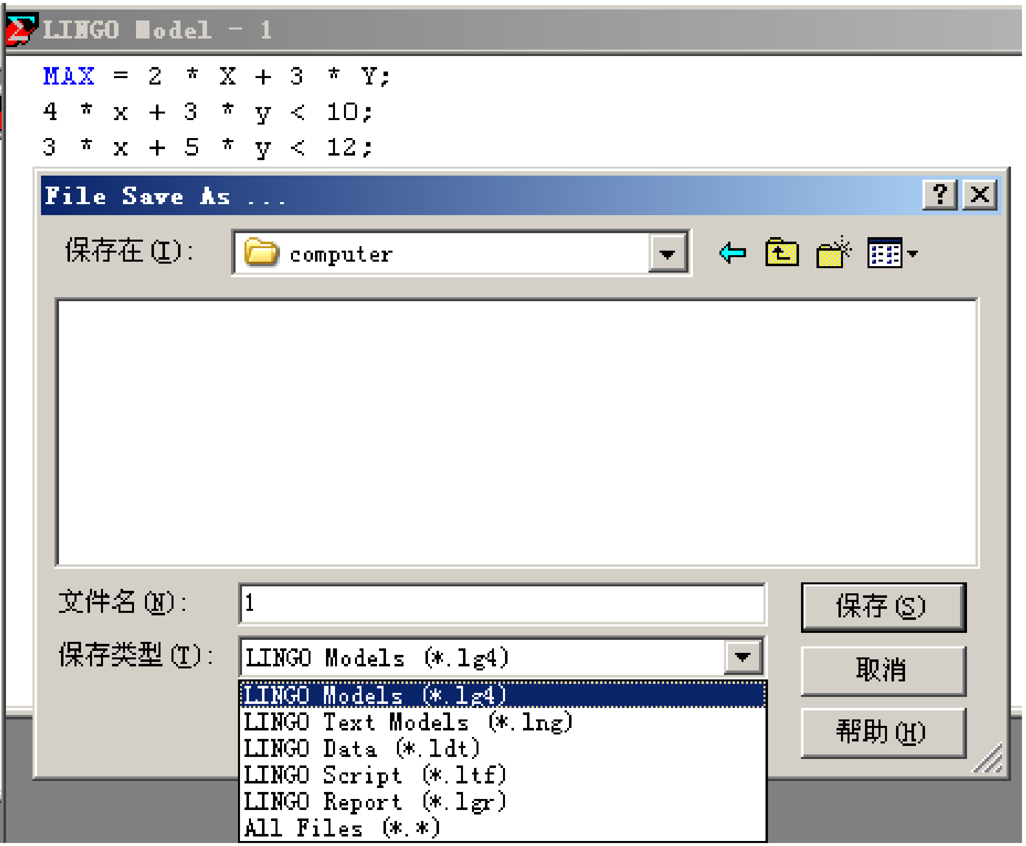
```
max 2x+3y
st
 4x+3y<10
 3x+5y<12
 end
```

Lingo:

```
max=2*x+3*y;
 4*x+3*y<10;
 3*x+5*y<12;
```

为保证能将LINDO 模型移植到LINGO中去，在LINDO 模型输入时应尽量采用“规范化”的格式

Lingo的不同保存类型



“LG4”表示LINGO 格式的模型文件，是一种特殊的二进制格式文件，保存了我们在模型窗口中所能够看到的所有文本和其他对象及其格式信息，只有LINGO 能读出它，用其他系统打开这种文件时会出现乱码

“LNG”表示LINGO文本文件，以这个格式保存模型时系统 将给出警告，因为模型中的格式信息（如字体、颜色等）将会丢失

除“LG4”文件外，这里的另外几种格式的文件其实都是普通的文本文件，可以用任何文本编辑器打开和编辑

“LDT”表示数据文件

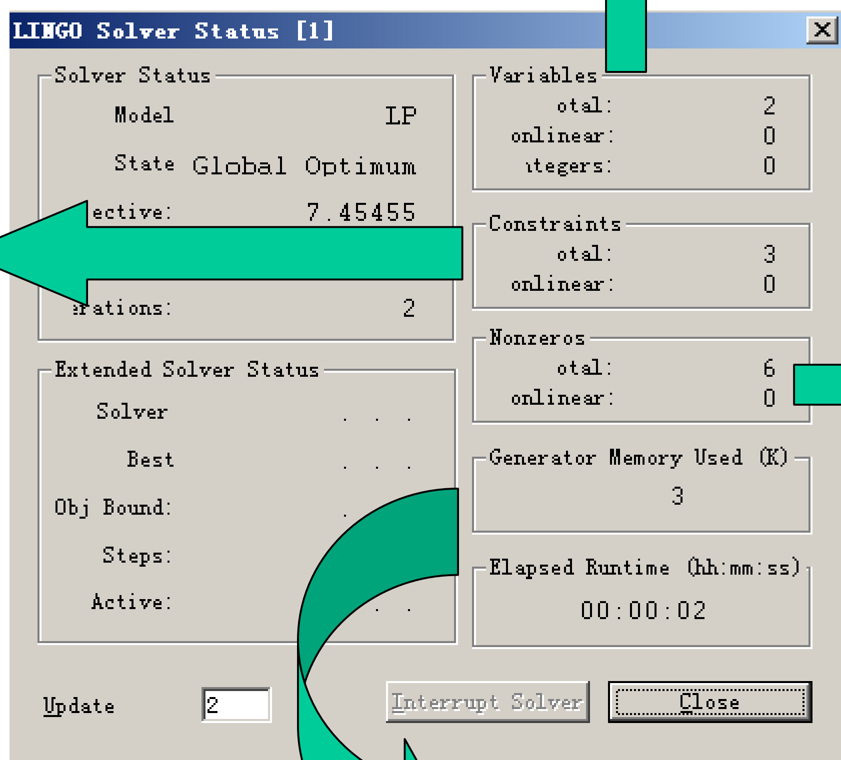
“LTF”表示 命令脚本文件

“LGR”表示 报告文件

状态窗口的参数解释

变量数量（其中包括变量总数、非线性变量数、整数变量数）

约束数量（约束总数、非线性约束个数）



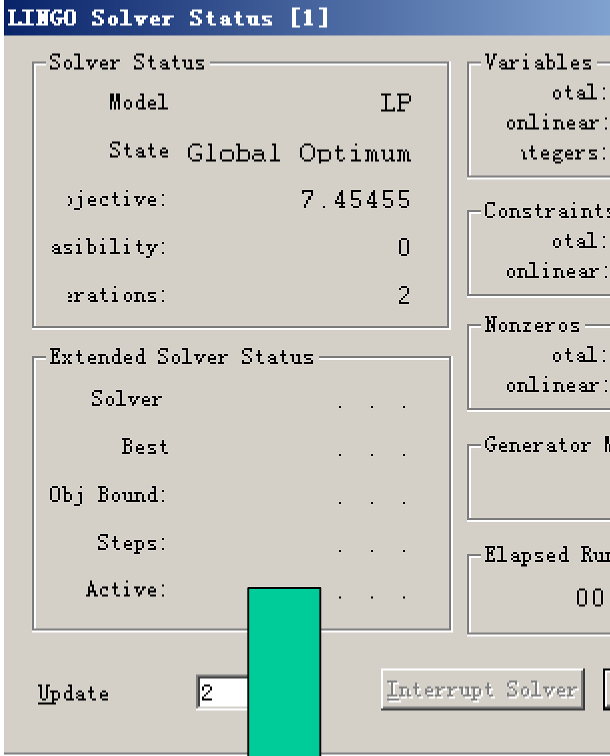
非零系数数量（总数、非线性项的个数）

内存使用量、求解花费的时间

状态窗口的参数解释(2)

求解器状态框

域名	含义	可能的显示
Model Class	当前模型的类型	LP，QP，ILP，IQP，PILP， PIQP，NLP，INLP，PINLP （以 I 开头表示 IP，以 PI 开头表示 PIP）
State	当前解的状态	"Global Optimum"，"Local Optimum"，"Feasible"，"Infeasible"（不可行），"Unbounded"（无界），"Interrupted"（中断），"Undetermined"（未确定）
Objective	当前解的目标函数值	实数
Infeasibility	当前约束不满足的总量（不是不满足的约束的个数）	实数（即使该值=0，当前解也可能不可行，因为这个量中没有考虑用上下界形式给出的约束）
Iterations	目前为止的迭代次数	非负整数



域名	含义	可能的显示
Solver Type	使用的特殊求解程序	B-and-B （分枝定界算法） Global （全局最优求解程序） Multistart（用多个初始点求解的程序）
Best Obj	目前为止找到的可行解的最佳目标函数值	实数
Obj Bound	目标函数值的界	实数
Steps	特殊求解程序当前运行步数： 分枝数（对 B-and-B 程序）； 子问题数（对 Global 程序）； 初始点数（对 Multistart 程序）	非负整数
Active	有效步数	非负整数

扩展的求解器（求解程序）状态框

用LINGO 来解二次规划问题

$$MAXz = 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 2x_2^2$$

$$s.t. x_1 + x_2 \le 100$$

$$x_1 \le 2x_2$$

$x_1, x_2 \ge 0$为整数

```
max=98*x1+277*x2-x1^2-
0.3*x1*x2-2*x2^2;
 x1+x2<100;
 x1<2*x2;
@gin(x1);@gin(x2);
```

注意事项：

1）变量和行名可以超过8 个字符，但不能超过32 个字符，且必须以字母开头

2）LINGO 已假定各变量非负（除非用函数@free或@sub 或@slb 另行说明）

3）变量可以放在约束条件的右端（同时数字也可放在约束条件的左端）。但为了提高效率，应尽可能采用线性表达式定义目标和约束（如果可能）

Lingo的编程

LINGO模型的构成:4个段

集合段（SETS ENDSETS）

数据段（DATA ENDDATA）

初始段（INIT ENDINIT）

目标与约束段

```
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;
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;
d=3,5,4,7,6,11; e=20,20;
!x,y=5,1,2,7;
enddata

init:
!initial locations for the supply;
x,y=5,1,2,7;
endinit

!Objective function;
[OBJ] min=@sum(link(i,j): c(i,j)*((x(j)-a(i))^2+(y(j)-b(i))^2)^(1/2) );
!demand constraints;
@for(demand(i):[DEMAND_CON] @sum(supply(j):c(i,j))
!supply constraints;
@for(supply(i):[SUPPLY_CON] @sum(demand(j):c(j,i)) <=e(i); );
     );
END
```

例1：SAILCO 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 条船。如何安排生产可使总费用最小？

用DEM,RP,OP,INV 分别表示需求、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，也就说他们都应该是一个由4 个元素组成的数组，其中DEM 是已知的，而RP,OP,INV 是未知数。

目标函数：$MIN \sum_{i=1,2,3,4} \{400RP(I)+450OP(I)+20INV(I)\}$

约束条件1(能力限制)：RP(I)<40, I=1,2,3,4;

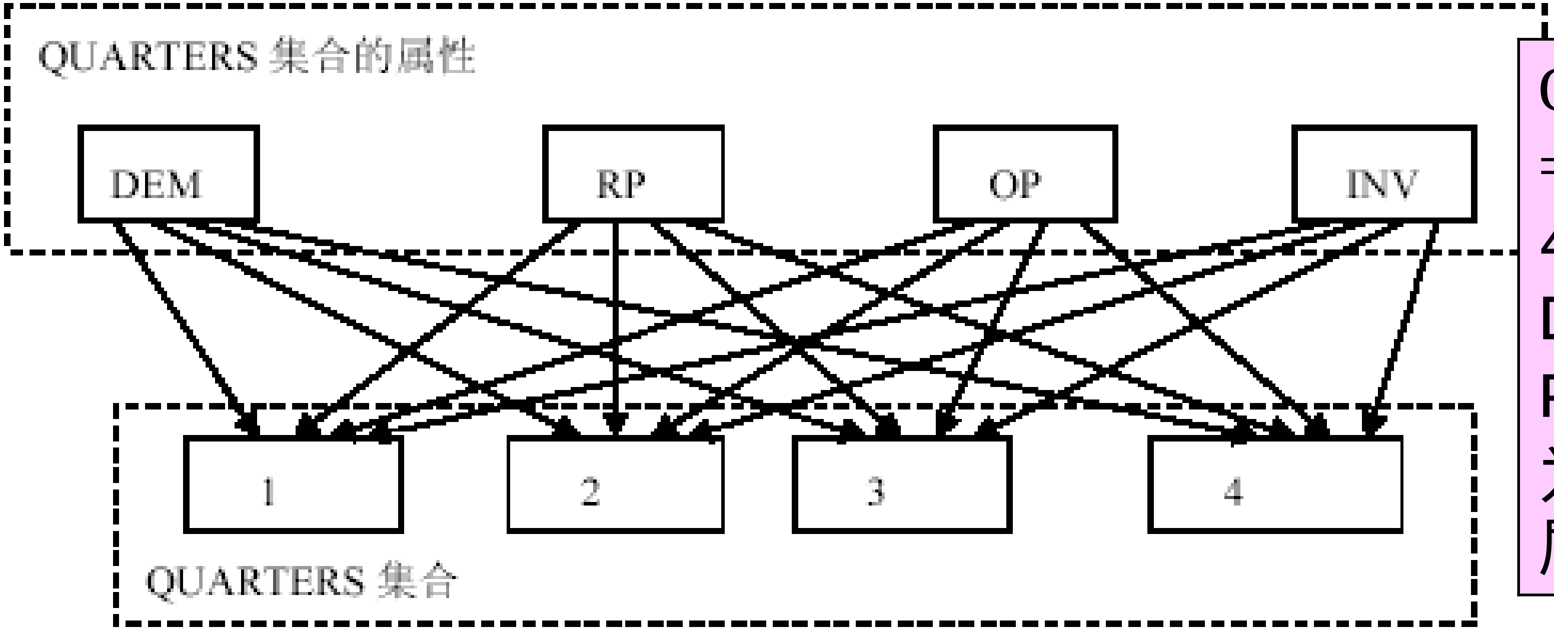
约束条件2(产品数量的平衡方程)：

INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(I),I=1,2,3,4;

INV(0)=10;

利用数组的概念

约束条件3: 变量的非负约束



QUARTERS ={1，2，3，4}称为集合，DEM,RP,OP,INV 称为该集合的属性

循环函数"@FOR(集合：约束关系式)"的方式定义的，意思是对冒号"："前面的集

合的每个元素（下标），冒号"："后面的约束关系式都要成立。由于下标i=1 时的约束关系式与i=2，3，4时有所区别，所以对下标集合的元素（下标）加了一个"i#GT#1"的限制条件，而把i=1 时的约束关系式单独写出。限制条件"i#GT#1"是一个逻辑表达式，意思就是i>1

```
MODEL:
 SETS:
  QUARTE
 ENDSETS
 DATA:
 DEM=40,60,75,25;
 ENDDATA
!初始段省略;
MIN=@SUM(QUARTE   S:400*RP+45
!目标函数;
 @FOR(QUARTERS(I):RP(I)<40);!能力
 @FOR(QUARTERS(I)|I#GT#1:!产品数
  INV(I)=INV(I-1)+RP(I)+OP(I)-DEM(
  INV(1)=10+RP(1)+OP(1)-DEM(1);
 END
```

Objective value: 78450.00

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

基本集合与派生集合

例2: 某公司有6 个建筑工地要开工，每个工地的位置（用平面坐标a, b 表示，距离单位：公里）及水泥日用量d（吨）由下表 给出。目前有两个临时料场位于P (5, 1), Q (2, 7) ，日储量各有20 吨。假设从料场到工地之间均有直线道路相连，试制定每天的供应计划，即从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。为了进一步减少吨公里数，打算舍弃两个临时料场，改建两个新的，日储量仍各为20 吨，问应建在何处，节省的吨公里数有多大。

a

b

d

记工地的位置为(a_i, b_i)，水泥日用量为$d_i, i=1\dots6$；料场位置为(x_j, y_j)，日储量为$e_j, j=1,2$；从料场j向工地i的运送量为c_{ij}。这个优化问题的数学规划模型是：

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

$$st\ \sum_{j=1}^{2} c_{ij}=d_i, i=1\dots6$$

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

当使用现有临时料场时，决策变量只有c_{ij}，是LP 模型；当为新建料场选址时决策变量为c_{ij}和x_j, y_j，由于目标函数f对x_j, y_j是非线性的，所以在新建料场时是NLP 模型。我们现在先解NLP模型，而把现有临时料场的位置作为初始解告诉LINGO。

```
MODEL:
Title Location Problem;
sets:
     demand/1..6/:a,b,d;
     supply/1..2/:x,y,e;
     link(demand,supply):c; (1)
endsets
data:
!locations for the demand(需求点的位
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 suppl
d=3,5,4,7,6,11; e=20,20;

enddata
init:
!initial locations for the supply(
x,y=5,1,2,7; (2)
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: @free(X);    @free(Y);  ); (3)
```

(1):LINK中的元素就是DEMAND 和SUPPLY 的笛卡儿积，也就是LINK={（S，T）|S∈DEMAND，T∈SUPPLY}.因此，其属性C 也就是一个6*2 的矩阵（或数组）。正是由于这种表示方式，LINGO 建模语言也称为矩阵生成器。DEMAND 和SUPPLY 这种直接把元素列举出来的集合，称为基本集合（primary set），而把LINK 这种基于基本集合构造的集合称为派生集合（derived set）

(3):@free 函数取消了变量X、Y 非负限制

(2):LINGO对数据是按列赋值的，而不是按行.其中“X Y =5，1，2，7；”语句的实际赋值顺序是X=(5,2),Y=(1,7),

局部最优解

X(1)=7.249997，

X(2)=5.695940，

Y(1)=7.749998，

Y(2)=4.928524，

（略），最小运量=89.8835(吨公里)

NLP中局部最优解不一定就是全局最优解,在 help中有这样的叙述:

"Thus, when a nonlinear model is solved, we say the solution is merely a local optimum, and the user must be aware other local optimums may, or may not, exist with better objective values."

第一步:利用LINGO|Options"菜单命令激活全局最优求解程序

第二步:为减少计算工作量，对X，Y 的取值再做一些限制。由于最佳料场位置至少不应该超出现在6 个工地所决定的坐标的最大、最小值决定的矩形之外，即:0.5<=x<=8.75, 0.75<=y<=7.75．可以加上@bnd 函数

```
@for(supply: @free(X);    @free(Y
@for(supply: @bnd(0.5,X,8.75);
             @bnd(0.75,Y,7.75);
```

添加部分

```
END
```

最优解X(1)=3.2549，X(2)=7.2500，Y(1)=5.6526， Y(2)=7.7500，最小运量=85.26(吨公里)

把料厂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。只需把初始段的"X Y =5， 1， 2， 7；" 移到数据段就可。此时，得到全局最优解，最小运量136.2275(吨公里)

稠密集合与稀疏集合

若派生集合是基本集合构成的笛卡儿积, 则称它为稠密集合 ;派生集合的元素可以只是这个笛卡儿积的一个真子集合，这种派生集合称为稀疏集合

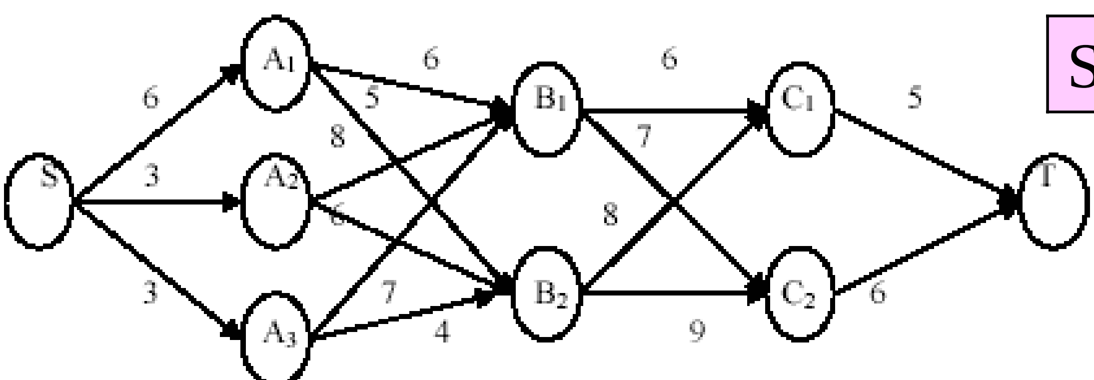
例3 最短路问题 在公路网中，司机希望找到一条从一个城市到另一个城市的最短路. 假设图 表示的是该公路网, 节点表示货车可以停靠的城市,弧上的权表示两个城市之间的距离（百公里）. 那么,货车从城市S 出发到达城市T,如何选择行驶路线,使所经过的路程最短?

S到T的最优行驶路线P

先求出从S到Ck(k=1,2)的最优行驶路线 0

S到Bj(j=1,2)的最优行驶路线

S到Ai(i=1,2,3)的最优行驶路线 **(易求)**



模型的建立

从S到T的行驶过程分成4 个阶段,即S→A*i*(*i*=1,2 或3), A*i* → B*j*(*j*=1或2), B*j* → C*k*(*k*=1或2), C*k* → T

记$d(Y,X)$为城市Y与城市X之间的直接距离（若这两个城市之间没有道路直接相连，则可以认为直接距离为无穷大），用$L(X)$表示城市S到城市X的最优行驶路线的路长, 则:

$$L(S)=0$$

$$\min_{Y\neq X}\left\{L(Y)+d(Y,X)\right\},\ X\neq S$$

```
MODEL:
 SETS:
  CITIES/S,A1,A2,A3,B1,B2,C1,C2,T/:L;
 !集合CITIES(城市)表示点，赋值L(i)表示从城市S到城市i的最短距离；
 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;
!ROADS(CITIES,CITIES)表示连接城 … 有的弧相连，
所以将弧具体列出，而数据D(i，j)表 … 市j的直接距离；
 ENDSETS
 DATA:
  D= 6 3 3
     6 5 8 6 7
     6 7 8 9
     5 6;
  L=0,,,,,,,,;  !因为L(S)=0,但位置必须留出来;
 ENDDATA
 @FOR(CITIES(i)|i#GT#@index(S):
!i 指的正是元素在集合中的位置(顺序)，一般称为元素的索引(INDEX)
 函数“@index”，其作用是返回一个元素在集合中的索引值;
  L(i)=@MIN(ROADS(j,i):L(j)+D(j,i)););
!这就是前面写的最短路关系式;
 end
```

定义稀疏集合ROADS 方法:元素枚举法

从S到T的最优行驶路线的路长为20（进一步分析，可以得到从S到T的最优行驶路线为S→ A3→ B2→ C1 → T）

另一种定义稀疏集合的方法:“元素过滤”法

匹配（MATCHING）问题:8 名同学准备分成4 个调查队（每队两人）前往4 个地区进行社会调查。设两两之间组队的效率如表 所示（由于对称性只列出了上三角部分），问如何组队可以使总效率最高？

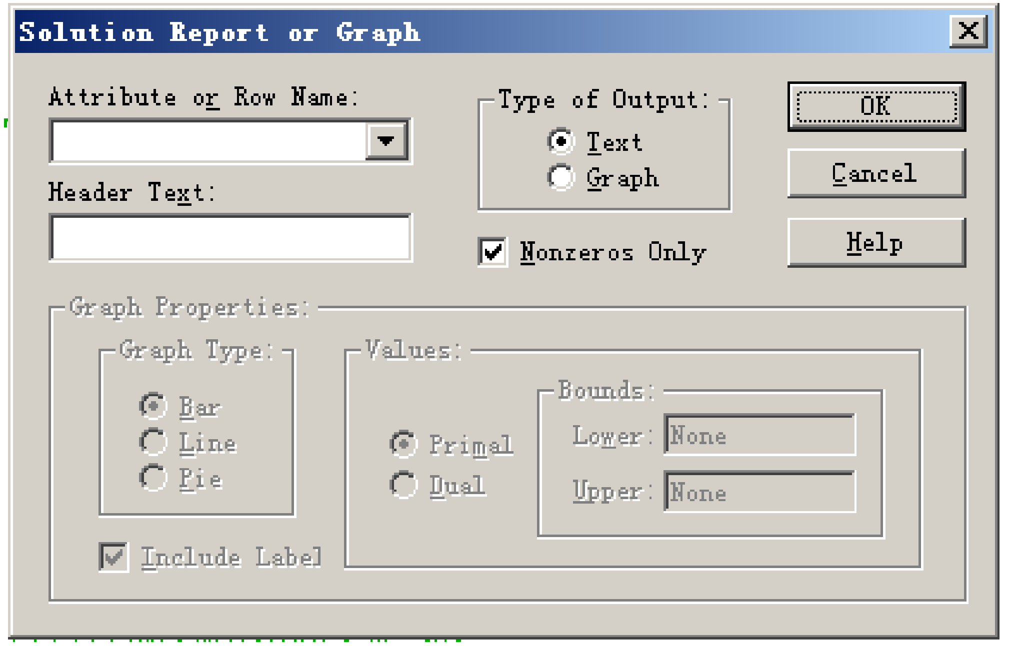
学生	S1	S2	S3	S4	S5	S6	S7	S8
S1	-	9	3	4	2	1	5	6
S2	-	-	1	7	3	5	2	1
S3	-	-	-	4	4	2	9	2
S4	-	-	-	-	1	5	5	2
S5	-	-	-	-	-	8	7	6
S6	-	-	-	-	-	-	2	3
S7	-	-	-	-	-	-	-	4

BENEFIT为效率矩阵，MATCH（Si，Sj）=1 表示Si，Sj 组成一队，0 表示不组队。由于对称性只需考虑i<j 共32 个0-1 变量。

目标函数为BENEFIT（Si，Sj）* MATCH（Si，Sj）之和；约束条件是每个同学只能（而且必须在）某一组，即对于任意i 有：只要MATCH 属性的某个下标为i 就加起来，此和=1。显然，这是一个0-1 线性规划

```
MODEL:
 SETS:
  STUDENTS/S1,S2,S3,S4,S5,S6,S7,S8/;
! 集合STUDENTS表示8位同学;
  PAIRS(STUDENTS,STUDENTS) |&2#GT#&1:
!第2 个父集合的元素的索引值(用"&2"
表示)大于第1 个父集合的元素的索引值(用"&1"
   BENEFIT,MATCH;
ENDSETS
DATA:
  BENEFIT=
   9 3 4 2 1 5 6
     1 7 3 5 2 1
       4 4 2 9 2
         1 5 5 2
           8 7 6
             2 3
               4;
ENDDATA
 MAX= @SUM( PAIRS(I,J): BENEFIT(I,J)*MATCH(I,J));
@FOR(STUDENTS(I ):[CONSTRAINTS]
  @SUM( PAIRS(J,K)|J#EQ#I#OR#K#EQ#I:MATCH(J,K))=1);
!当各个下标不同时，赋值，保证每个同学只能(而且必须在)某一组;
  @FOR(PAIRS(I,J):@BIN(MATCH(I,J)));
!产生约束条件，限制i和j为二进制整数
END
```

由于MATCH 变量中多数为0

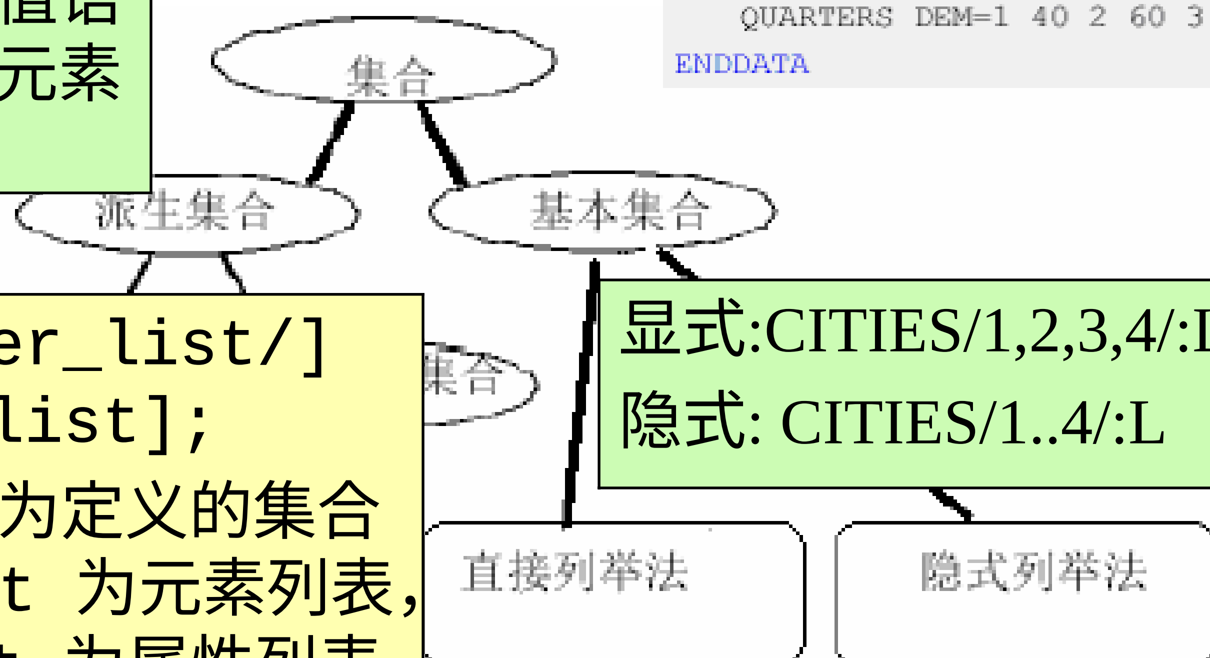


Variable	Value	Reduced Cost
MATCH(S1, S8)	1.000000	-6.000000
MATCH(S2, S4)	1.000000	-7.000000
MATCH(S3, S7)	1.000000	-9.000000
MATCH(S5, S6)	1.000000	-8.000000

类型	隐式列举格式	示例	示例集合表示的元素
数字型	1..n	1..5	1，2，3，4，5
字符-数字型	stringM..stringN	Car101..car208	Car101，car102，…，car208
日期（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

当元素列表不在集合定义中出现时，则必须在程序的数据段以赋值语句的方式直接给出元素列表

```
SETS:
    QUARTERS:DEM,RP,OP,INV;
ENDSETS
DATA:
    QUARTERS DEM=1 40 2 60 3 75 4 25;
ENDDATA
```



显式:CITIES/1,2,3,4/:L

隐式: CITIES/1..4/:L

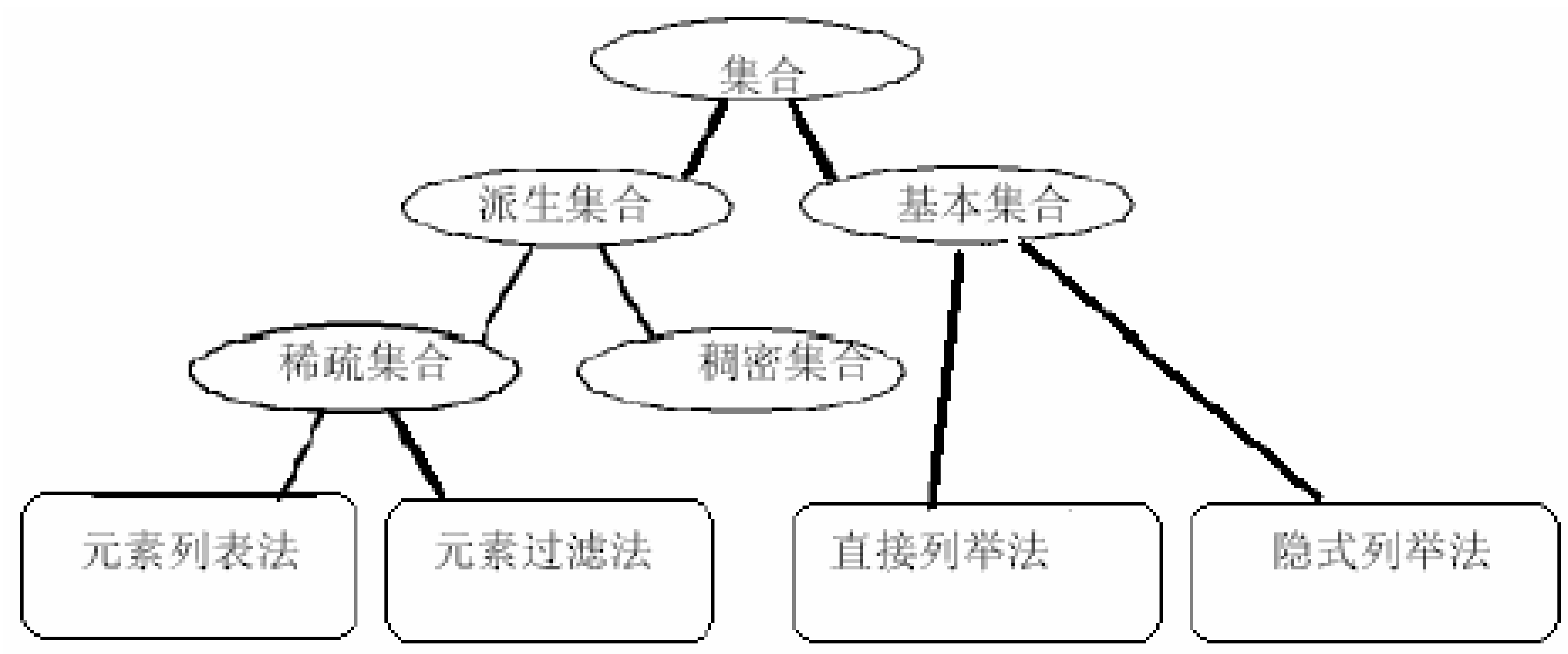
setname[/member_list/]
[: attribute_list];

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

派生集合一般定义格式为:setname(parent_set_list)[/member_list/] [: attribute_list];
称parent_set_list为父集合列表

```
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; 元素列举法
ENDSETS
DATA:
```

```
SETS:
 STUDENTS/S1,S2,S3,S4,S5,S6,S7,S8/;
 PAIRS(STUDENTS,STUDENTS) |&2#GT#&1:
  BENEFIT,MATCH; 元素过滤法
ENDSETS
```



运算符及其优先级

算术运算符：+（加法），—（减法或负号），*（乘法），/（除法），^(求幂)

关系运算符：<（即<=，小于等于),=（等于），>（即>=，大于等于)

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）。结果只有“真”（1）和“假”（0）两个值

优先级	运算符
最高	#NOT# —（负号）
	^
	* /
	+ —（减法）
	#EQ# #NE# #GT# #GE# #LT# #LE#
	#AND# #OR#
最低	< = >

同一优先级按从左到右的顺序执行；如果有括号“（）”，则括号内的表达式优先进行计算

LINGO 函数一览 (1)

1） 常用的数学函数

@ABS(X) 返回变量X 的绝对值。

@COS(X) 返回变量X 的余弦值（X 的单位是弧度）。

@EXP(X) 返回 e^x 的值（其中e 为自然对数的底，即2.718281...）。

@FLOOR(X) 返回的整数部分（向最靠近0 的方向取整）。

@LGM(X) 返回变量X 的gamma（伽玛）函数的自然对数值（当X 为整数时LGM(X) =LOG(X-1)! ；当X 不为整数时，采用线性插值得到结果）。

@LOG(X) 返回变量X 的自然对数值。

@SIGN(X) 返回变量X 的符号值（ X < 0 时返回-1， X >= 0 时返回+1）。

@SIN(X) 返回变量X 的正弦值（X 的单位是弧度）。

@SMAX(list) 返回一列数(list)的最大值。

@SMIN(list) 返回一列数(list)的最小值。

@TAN(X) 返回变量X 的正切值（X 的单位是弧度）。

2） 变量界定函数

变量函数对变量的取值范围附加限制。共四种：

@BND(L， X， U) 限制L <= X <= U。

@BIN(X) 限制X 为0 或1。

@FREE(X) 取消对X 的符号限制（即可取负数、 0 或正数）。

@GIN(X) 限制X 为整数。

LINGO 函数一览 (2)

3）集合循环函数

集合循环函数的用法如下：

@function(setname [(set_index_list)[| condition]] : expression_list);其中function 是集合函数名，有FOR、MAX、MIN、SUM 四种；setname 是集合名；set_index_list 是集合索引列表（不需使用索引时可以省略）；condition 是用逻辑表达式描述的条件（通常含有索引，无条件时可以省略）；expression_list 是一个表达式（对@FOR函数，可以是一组表达式）。集合函数名的含义如下：

@FOR 对集合setname 的每个元素独立地生成约束，约束由expression_list 描述。

@MAX 返回集合setname 上的表达式的最大值。

@MIN 返回集合setname 上的表达式的最小值。

@SUM 返回集合setname 上的表达式的和。

4）财务函数

@FPA(I,N) 返回如下情形下的净现值：单位时段利率为I，连续N 个时段支付，每个时段支付单位费用。即@FPA(I,N) $=\sum_{i=1}^{N} 1/(1+I)^{n}$

@FPL(I,N) 返回如下情形下的净现值：单位时段利率为I，第N 个时段支付单位费用。即@FPL(I,N) $= \left(1/_{1+I}\right)^{N}$

LINGO 函数一览 (3)

5）概率函数

这里我们只是列出这些函数的简要功能，由于牵涉较多概率论和随机过程的概念，请大家参阅有关概率论和随机过程的书籍。

@PSN(X) 标准正态分布的分布函数。

@PSL(X) 单位正态线性损失函数，即返回MAX(0,Z-X)的期望值，其中Z 为标准正态随机变量。

@PPS(A,X) 均值为A 的Poisson 分布的分布函数（当X 不是整数时，采用线性插值进行计算）。

@PPL(A,X) Poisson 分布的线性损失函数，即返回MAX(0, Z-X)的期望值，其中Z为均值为A 的Poisson 随机变量。

@PBN(P,N,X) 二项分布的分布函数（当N 和（或）X 不是整数时，采用线性插值进行计算）。

@PHG(POP,G,N,X) 超几何（Hypergeometric）分布的分布函数（当POP，G，N 和（或）X 不是整数时，采用线性插值进行计算）。也就是说，这个就是如下概率：当总共有POP 个球，其中G 个是白球的，那么随机地从中取出N 个球，白球不超过X 个的概率。

@PEL(A,X) 当到达负荷为A，服务系统有X 个服务器且不允许排队时的Erlang 损失概率。

@PEB(A,X) 当到达负荷为A，服务系统有X 个服务器且允许无穷排队时的Erlang 繁忙概率。

@PFS(A,X,C) 当负荷上限为A，顾客数为C，平行服务器数量为X 时，有限源的Poisson服务系统的等待或返修顾客数的期望值。（A 是顾客数乘以平均服务时间，再除以平均返修时间。当C 和（或）X 不是整数时，采用线性插值进行计算）。

LINGO 函数一览 (4)

@PFD(N,D,X) 自由度为N 和D 的F 分布的分布函数。
@PCX(N,X) 自由度为N 的Chi-squared 分布的分布函数。
@PTD(N,X) 自由度为N 的t 分布的分布函数。
@QRAND(SEED) 返回0 与1 之间的拟均匀随机数（SEED 为种子，缺省时取当前计算机时间）。该函数只能用在数据段，拟均匀随机数可以认为是“超均匀”的随机数。
@RAND(SEED) 返回0 与1 之间的伪均匀随机数（SEED 为种子）

6）文件输入输出函数
@DUAL(variable_or_row_name)
返回解答中变量的判别数（reduced cost）或约束行的对偶（影子）价格（dual prices）。
@FILE(filename)
当前模型引用其他ASCII 码文件中的数据或文本时可以采用该语句（但不允许嵌套使用），其中filename 为存放数据的文件名，该文件中记录之间用“~”分开。
@TEXT(['filename'])
用于数据段中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）。
@RANGED(variable_or_row_name)
为了保持最优基不变，变量的费用系数或约束行的右端项允许减少的量。
@RANGEU(variable_or_row_name)
为了保持最优基不变，变量的费用系数或约束行的右端项允许增加的量（。

LINGO 函数一览 (5)

7)集合处理函数

@IN(set_name, primitive_index_1 [, primitive_index_2 ...])

如果集合set_name 中包含本集合的元素索引primitive_index_1 [, primitive_index_2 ...]所对应的元素，则返回1，否则返回0。元素索引用“&1”、“&2”或@INDEX 函数等形式给出，这里“&1”表示对应于第1 个父集合的元素的索引值，“&2”表示对应于第2 个父集合的元素的索引值。

例如，如果我们想定义一个学生集合STUDENTS（基本集合），然后由它派生一个及格学生的集合PASSED 和一个不及格学生的集合FAILED，可以如下定义：

```
SETS:
STUDENTS / ZHAO, QIAN, SUN, LI/:;
PASSED( STUDENTS) /QIAN, SUN/:;
FAILED( STUDENTS) | #NOT# @IN( PASSED, &1):;
ENDSETS
```

@INDEX([set_name,] primitive_set_element)

给出元素primitive_set_element 在集合set_name 中的索引值(即顺序位置的编号)。如果省略set_name ， LINGO 按模型中定义的集合顺序找到第一个含有元素primitive_set_element 的集合，并返回索引值。如果在所有集合中均没有找到该元素，会给出出错信息。

@WRAP(I,N)

当I 位于区间[1, N]内时直接返回I；一般地，返回J = I - K *N ，其中J 位于区间[1, N]， K 为整数。可见这个，此函数相当于数学上用I 对N 取模函数的值+1，即@WRAP(I,N)=I（mode N）+1。此函数对N<1 无定义。可以想到，此函数的目地之一是可以用来防止集合的索引越界。

@SIZE (set_name)

返回数据集set_name 中包含元素的个数。

LINGO 函数一览 (6)

8) 其他函数

@IF(logical_condition, true_result, false_result)

当逻辑表达式logical_condition 的结果为真时，返回true_result，否则返回false_result。例如@IF(X #LT# 100, 20, 15)语句，当X<100 时，返回20，否则返回15。

@WARN('text', logical_condition)

当如果逻辑表达式logical_condition 的结果为真，显示'text'信息。

$$
\begin{aligned}
&\min \quad f(x)+g(y) \\
&s.t. \\
&\quad f(x)=\begin{cases} 100+2x, & x>0 \\ 2x, & x\le 0 \end{cases} \\
&\quad g(y)=\begin{cases} 60+3y, & y>0 \\ 2y, & y\le 0 \end{cases} \\
&\quad x+y\ge 30 \\
&\quad x,y\ge 0
\end{aligned}
$$

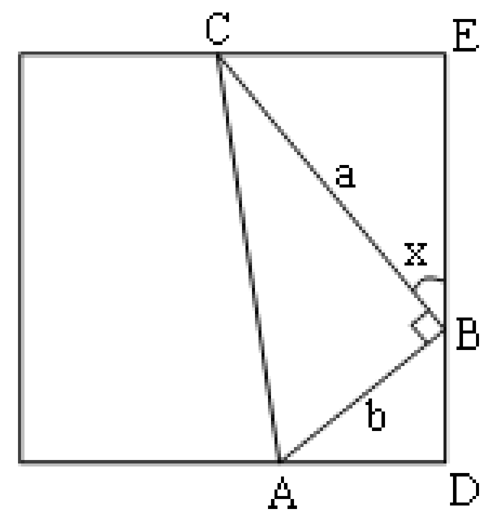
```
model:
  min=fx+fy;
  fx=@if(x #gt# 0, 100,0)+2*x;
  fy=@if(y #gt# 0,60,0)+3*y;
  x+y>=30;
end
```

@smax(x1,x2,...,xn)

给定一个直角三角形，求包含该三角形的最小正方形

$$CE = a\sin x,\quad AD = b\cos x,\quad DE = a\cos x + b\sin x,$$

$$\min_{0\le x\le \frac{\pi}{2}} \max\{CE,\ AD,\ DE\}$$




```
model:
sets:
  object/1..3/: f;
endsets
data:
  a, b = 3, 4; !两个直角边长，修改很方便；
enddata
  f(1) = a * @sin(x);
  f(2) = b * @cos(x);
  f(3) = a * @cos(x) + b * @sin(x);
  min = @smax(f(1),f(2),f(3));
  @bnd(0,x,1.57);
end
```

Variable	Value	Reduced Cost
F(1)	0.7276069	0.000000
F(2)	3.880570	0.000000
F(3)	3.880570	0.000000

@qrand(seed)

产生服从(0,1)区间的拟随机数。@qrand 只允许在模型的数据部分使用，它将用拟随机数填满集属性。通常，声明一个 m×n 的二维表，m 表示运行实验的次数，n 表示每次实验所需的随机数的个数。在行内，随机数是独立分布的；在行间，随机数是非常均匀的。这些随机数是用“分层取样”的方法产生的。

```
model:
data:
  M=4; N=2; seed=1234567;
enddata
sets:
  rows/1..M/;
  cols/1..N/;
  table(rows,cols): x;
endsets
data:
  X=@qrand(seed);
enddata
end
```

Variable	Value
M	4.000000
N	2.000000
SEED	1234567.
X(1, 1)	0.2085788
X(1, 2)	0.1381721
X(2, 1)	0.6283858
X(2, 2)	0.2530084
X(3, 1)	0.3767461
X(3, 2)	0.7546936
X(4, 1)	0.9335576
X(4, 2)	0.5737700

@rand(seed)

返回0和1间的伪随机数，依赖于指定的种子。典型用法是U(I+1)=@rand(U(I))。注意如果seed不变，那么产生的随机数也不变

利用@rand产生15个标准正态分布的随机数和自由度为2的t分布的随机数

```
model:
!产生一列正态分布和t分布的随机数;
sets:
  series/1..15/: u, znorm, zt;
endsets
    !第一个均匀分布随机数是任意的;
  u( 1) = @rand(0.1234);
  !产生其余的均匀分布的随机数;
  @for(series( I)| I #GT# 1:
    u( I) = @rand( u( I - 1))
  );
   @for( series( I):
    !正态分布随机数;
    @psn( znorm( I)) = u( I);
    !和自由度为2的t分布随机数;
    @ptd( 2, zt( I)) = u( I);
    !ZNORM 和 ZT 可以是负数;
    @free( znorm( I)); @free( zt( I));
  );
end
```

@for 函数

该函数用来产生对集成员的约束。基于建模语言的标量需要显式输入每个约束，不过@for函数允许只输入一个约束，然后LINGO自动产生每个集成员的约束

产生序列{1,4,9,16,25}

```
model:
sets:
  number/1..5/:x;
endsets
  @for(number(I): x(I)=I^2);
end
```

@sum 函数

该函数返回遍历指定的集成员的一个表达式的和

求向量[5，1，3，4，6，10]前5个数的和

```
model:
data:
  N=6;
enddata
sets:
  number/1..N/:x;
endsets
data:
  x = 5 1 3 4 6 10;
enddata
  s=@sum(number(I) | I #le# 5: x);
end
```

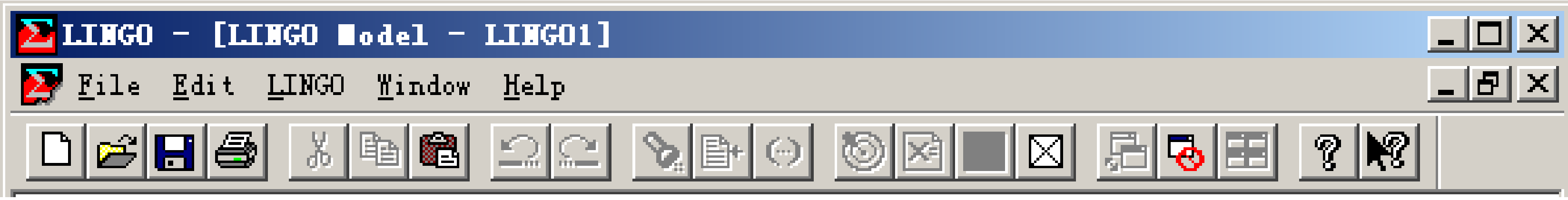
@min和@max

返回指定的集成员的一个表达式的最小值或最大值

求向量[5，1，3，4，6，10]前5个数最小值，后3个数最大值

```
model:
data:
  N=6;
enddata
sets:
  number/1..N/:x;
endsets
data:
  x = 5 1 3 4 6 10;
enddata
  minv=@min(number(I) | I #le# 5: x);
  maxv=@max(number(I) | I #ge# N-2: x);
end
```

LINGO的主要菜单命令



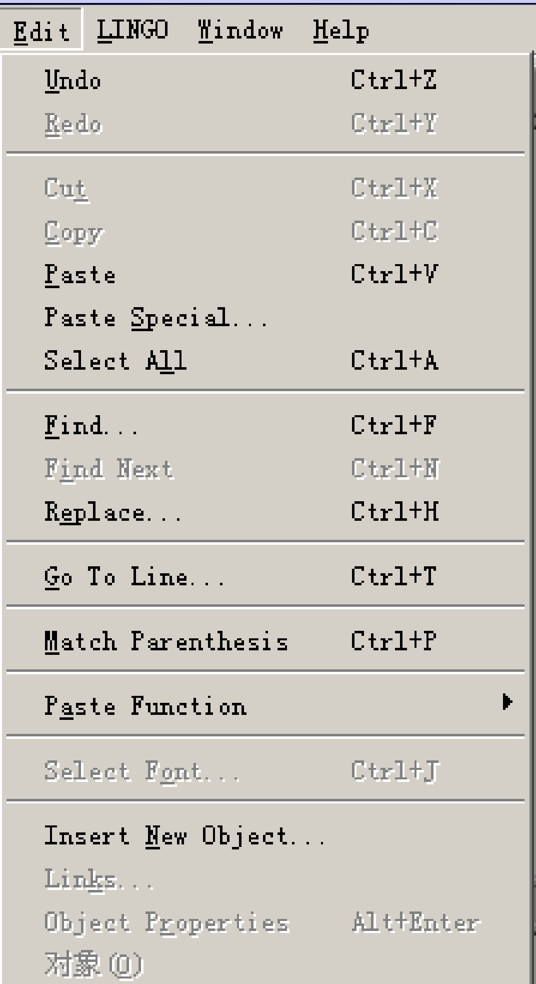
与LINDO 主菜单比较，LINGO 相当于合并了LINDO 中的Solve（求解）菜单和REPORTS（报告）菜单。这些菜单的用法都是和WINDOWS 下其他应用程序的标准用法类似的，下面只对前3 个主菜单中与LINDO 不同而有一定LINGO 特色的主要命令进行简要介绍。

1）文件主菜单

File|Import LINDO File:将以LINDO 文件格式保存的模型转换为LINGO 格式的模型

File|User Database Info:输入用户使用的数据库需要验证的用户名（User ID）和密码（Password）

编辑（Edit）主菜单



Paste Special：用于剪贴板中的内容不是文本的情形，如可以插入其它应用程序中生成的对象（Object）或对象的链接（Link）

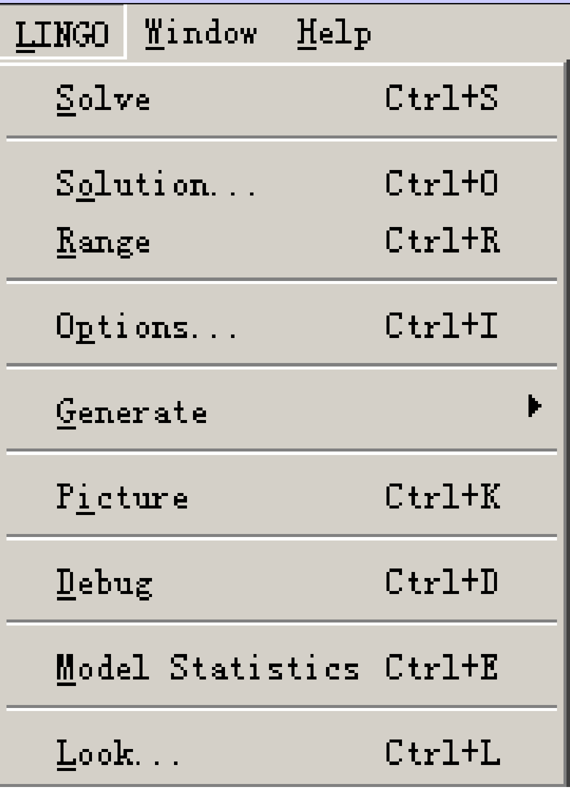
Match Parenthesis：匹配模型中的括号

Edit|Paste Function：选择LINDO 的某个函数，粘贴到当前光标处

Select Font:控制显示字体、字形、大小、颜色、效果等。注意：这些显示特性只有当文件保存为LINGO 格式（*.LG4）的文件时才能保存下来，

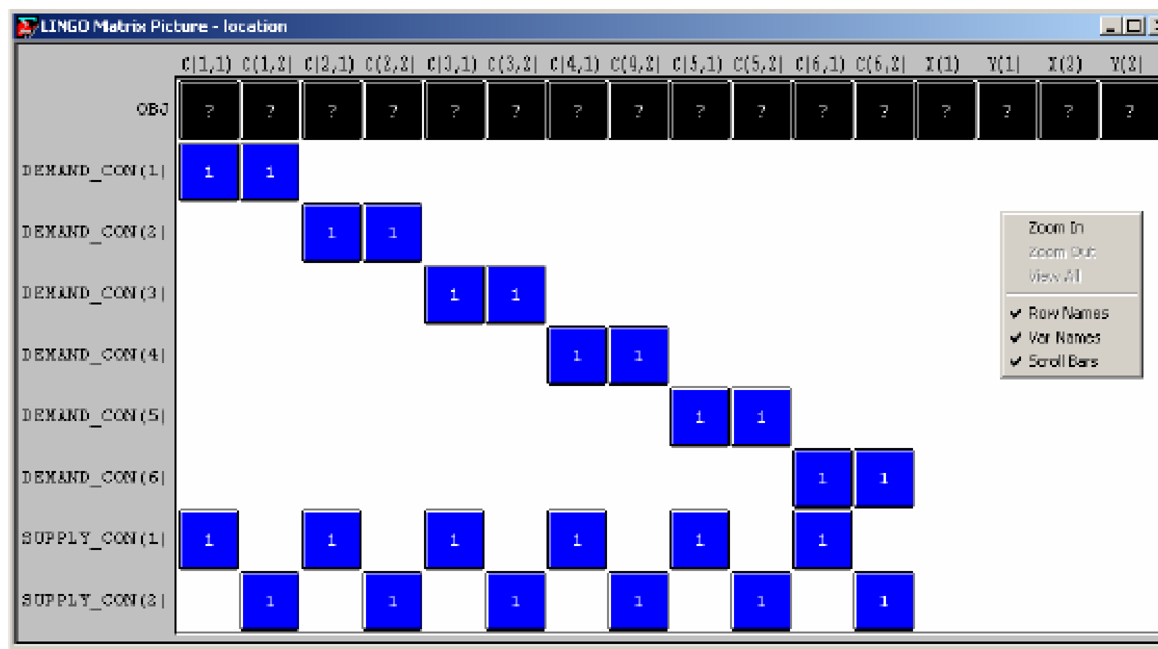
Edit|Link：在模型窗口中选择一个外部对象的链接，可以修改这个对象的属性;**Object Properties**：在模型窗口中选择修改一个链接或嵌入对象（OLE）的属性，

LINGO 主菜单



Generate：以线性形式显示目标函数和约束（只有非零项会显示）。如果有非线性变量项，对应的非线性变量前的系数将以问号（“？”）显示.按照是否在屏幕上显示结果的要求，“Display”和“Don’t display”两个子菜单供选择。在屏幕上不显示时，运行该命令的目的可能仅仅是为了以后选择适当的求解程序使用。

Picture：按照矩阵形式以图形方式给出.非线性项的系数以黑色显示为“？”，线性项的系数为正时显示为兰色，为负则为红色



此部分结束