

# LINGO软件简介

## -----求解线性和非线性优化问题

[1]刘焕彬, 库在强等编著.数学模型与实验.科学出版社, 2008年5月第1版.

[2]谢金星, 薛毅编著.优化建模与LINDO/LINGO软件.清华大学出版社, 2005年7月第1版.

# 内容提要

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题

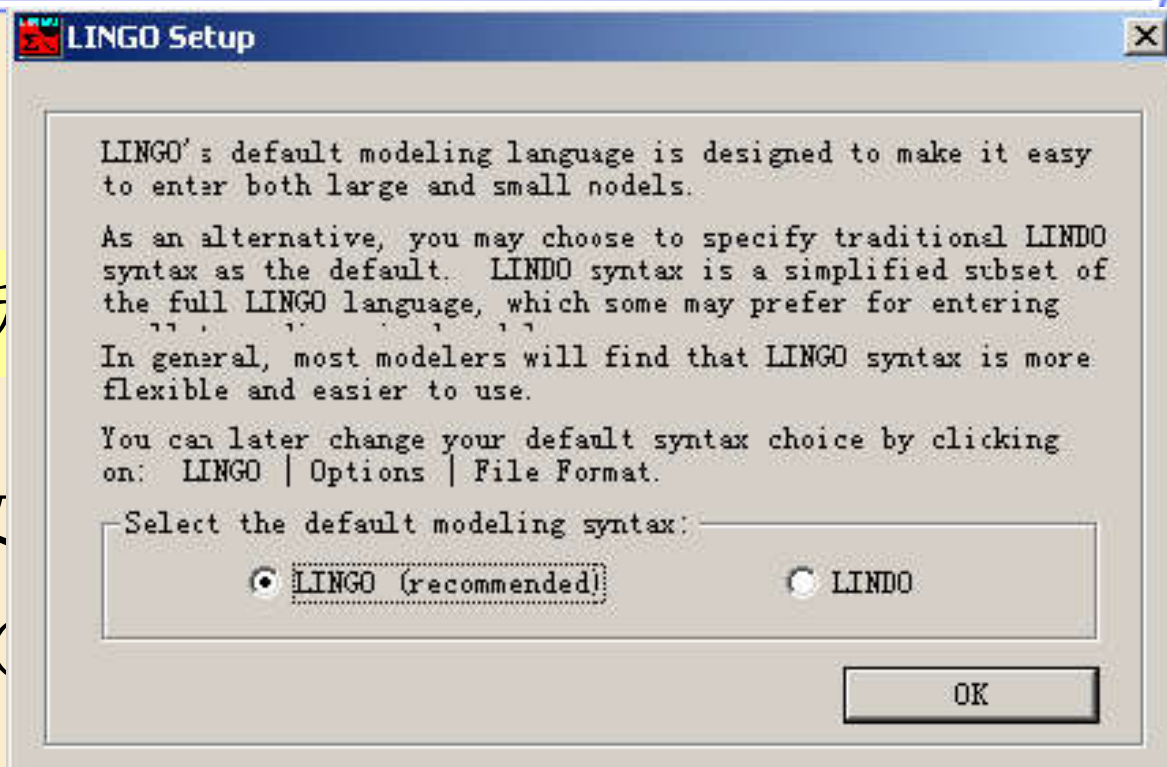
# 1. LINGO入门

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题

安装过

安装文

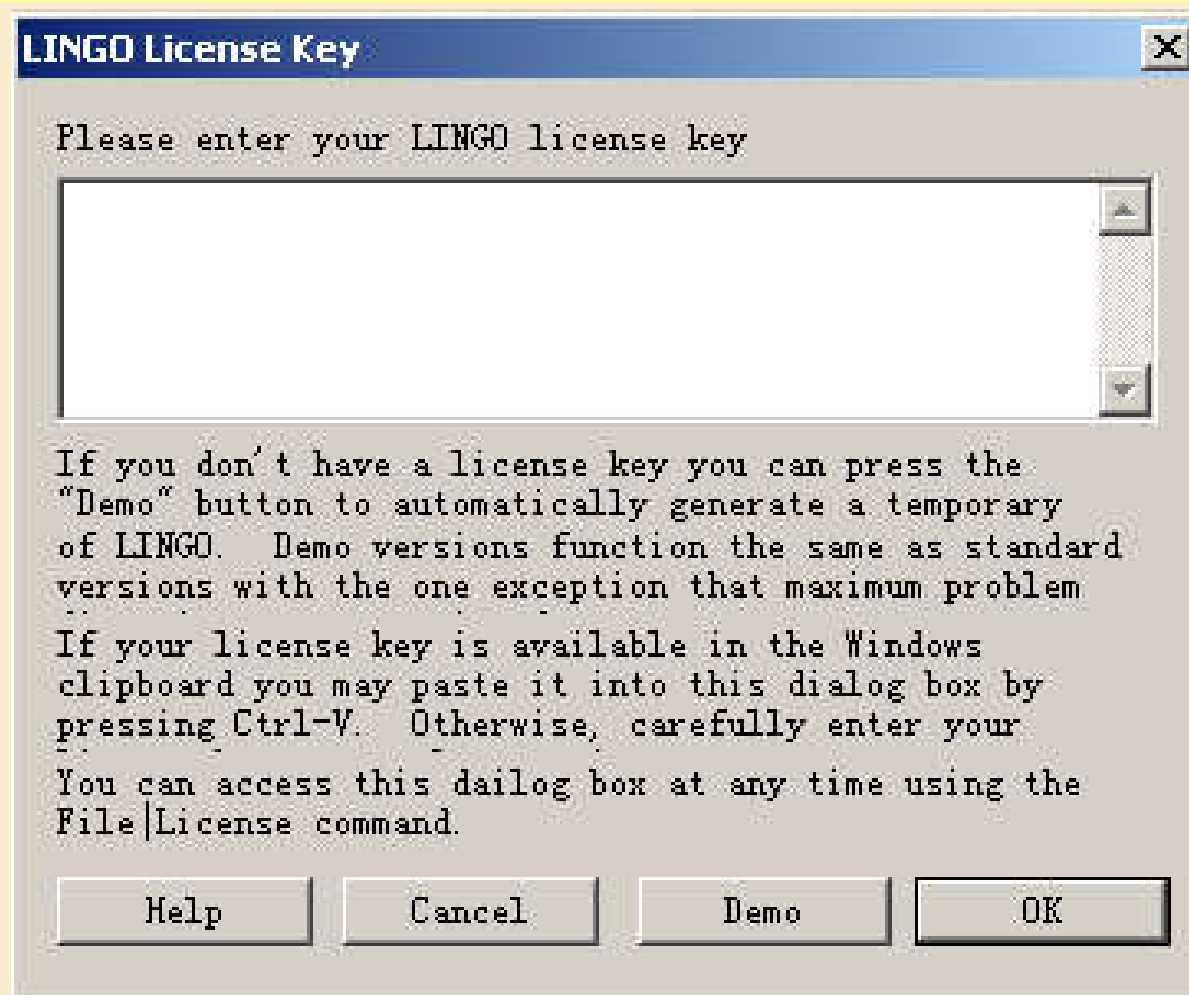
目录 (



选择安装

安装完成前，在出现的对话框(如图)中选择缺省的建模(即编程)语言，系统推荐的是采用LINGO。安装后可通过“LINGO|Options|Interface|File Format”命令修改缺省的建模（即编程）语言。

第一次运行时提示输入授权密码，如图：



点击**Demo**(试用版)，可以不用输密码。

# LINGO软件的主要特色

## 两种命令模式

**Windows模式：** 通过下拉式菜单命令驱动LINGO运行（多数菜单命令有快捷键，常用的菜单命令有快捷按钮），图形界面，使用方便； (这里主要介绍这种模式)

**命令行 模式：** 仅在命令窗口(Command Window)下操作，通过输入行命令驱动LINGO运行。

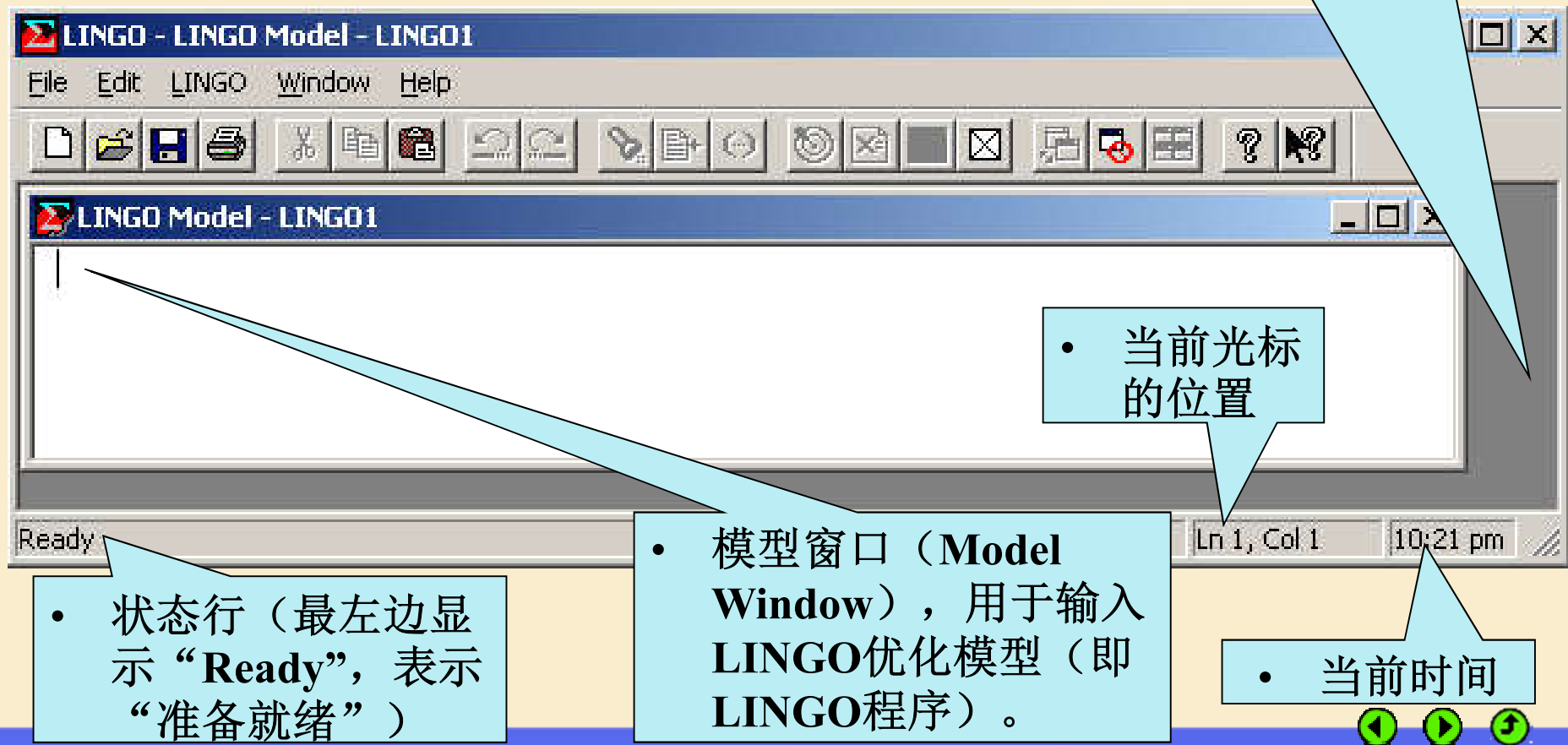
## 两大优点:

**LINGO 10.0**功能增强, 性能稳定, 解答结果可靠。

- 除具有求解线性规划问题外, 还可用于求解非线性规划问题, 包括非线性整数规划问题;
- 内置建模语言, 允许以简练、直观的方式描述较大规模的优化问题, 所需的数据可以以一定格式保存在独立的文件中, 例如excel文件。

## LINGO的界面

- LINGO软件的主窗口（用户界面），所有其他窗口都在这个窗口之内。





## LINGO的文件类型

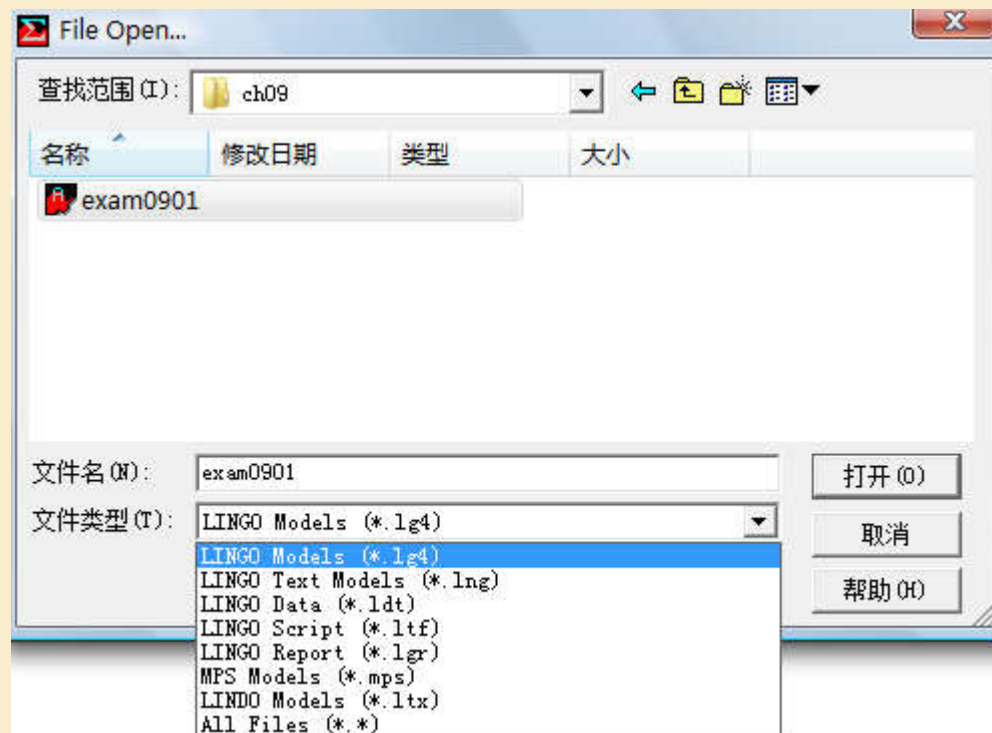
- .LG4**: LINGO格式的模型文件，保存了模型窗口中所能看到的所有文本和其他对象及其格式信息；
- .LNG**: 文本格式的模型文件，不保存模型中的格式信息（如字体、颜色、嵌入对象等）；
- .LDT**: LINGO数据文件；
- .LTF**: LINGO命令脚本文件；
- .LGR**: LINGO报告文件；
- .LTX**: LINDO格式的模型文件；
- .MPS**: MPS（数学规划系统）格式的模型文件。

除“**LG4**”文件外，另外几种格式的文件都是普通的文本文件，可以用任何文本编辑器打开和编辑。

## 在LINGO中打开模型

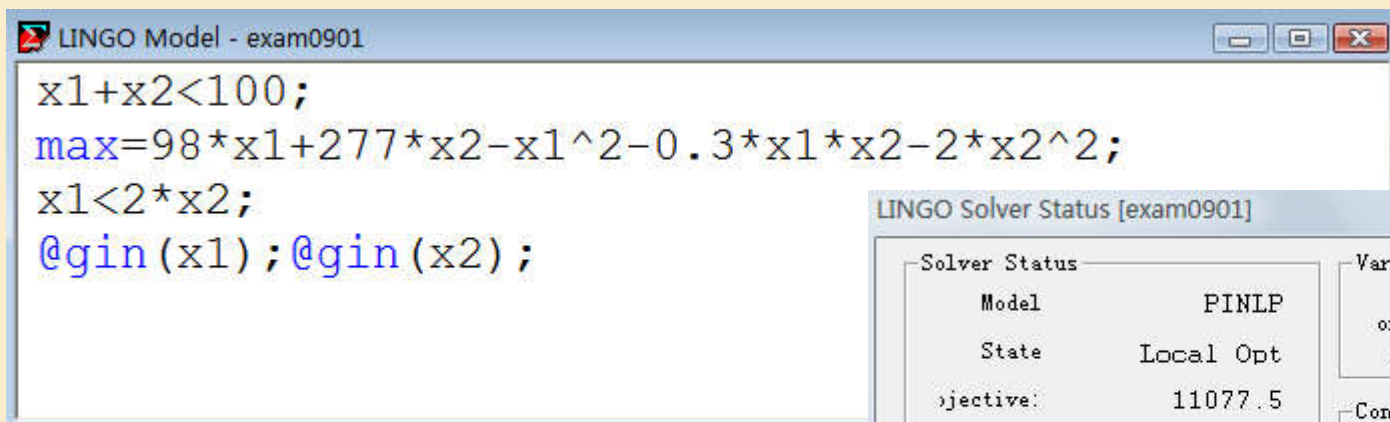
作为一个最简单的例子，在名为EXAM0901.lg4的模型文件中保存了一个LINGO模型，我们现在看看如何用LINGO把它打开。

- ① 选择菜单命令  
“File|Open(F3)”，  
可以看到“打开文件”对话框。（如图）



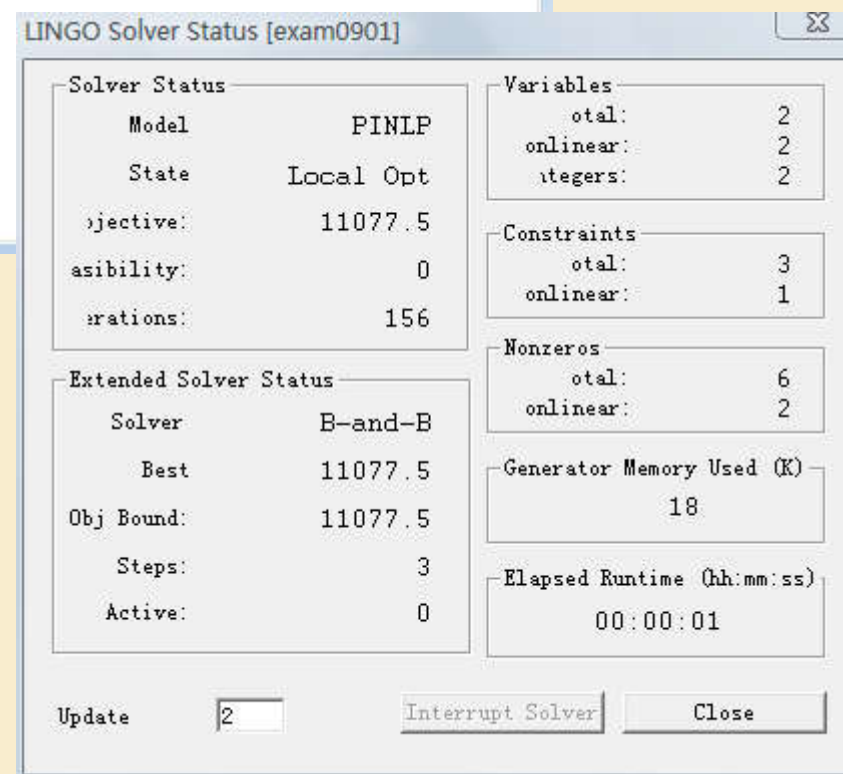
## 在LINGO中打开模型

② 打开“EXAM0901.LG4”文件（如下图）



```
LINGO Model - exam0901
x1+x2<100;
max=98*x1+277*x2-x1^2-0.3*x1*x2-2*x2^2;
x1<2*x2;
@gin(x1);@gin(x2);
```

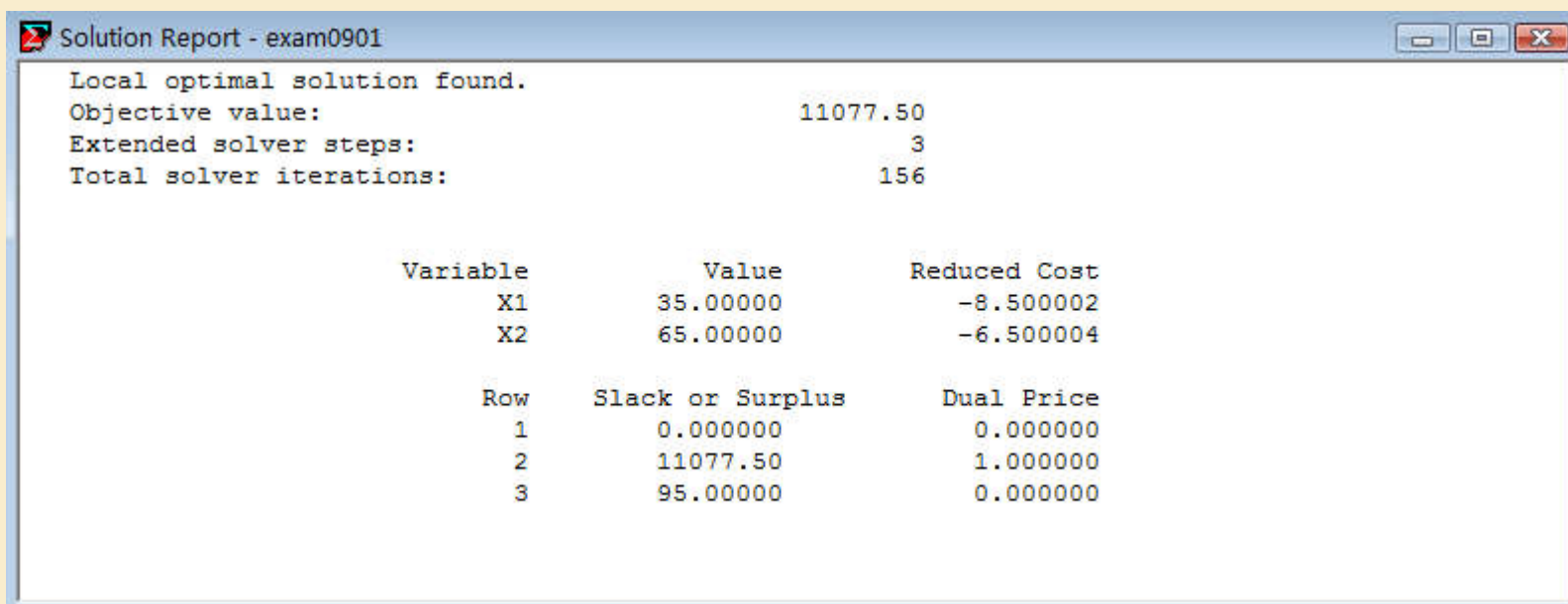
③ 选择“LINGO|Solve  
(Ctrl+S)”来运行这个程序（运行状态窗口如右图）



Solver Status		Variables	
Model	PINLP	total:	2
State	Local Opt	nonlinear:	2
Objective:	11077.5	integers:	2
Feasibility:	0	Constraints	
Iterations:	156	total:	3
		nonlinear:	1
Extended Solver Status		Nonzeros	
Solver	B-and-B	total:	6
Best	11077.5	nonlinear:	2
Obj Bound:	11077.5	Generator Memory Used (K)	
Steps:	3	18	
Active:	0	Elapsed Runtime (hh:mm:ss)	
		00:00:01	
Update 2		Interrupt Solver Close	

## 在LINGO中打开模型

### ④ 运行程序的LINGO报告窗口（如下图）



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

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

注：LINGO不询问是否进行敏感性分析，敏感性分析需要将来通过修改系统选项启动敏感性分析后，再调用“**REPORT|RANGE**”菜单命令来实现。现在同样可以把模型和结果报告保存在文件中。

## • 运行状态窗口

**Variables (变量数量):**

变量总数 (Total)、  
非线性变量数 (Nonlinear)、  
整数变量数 (Integer)。

**Constraints (约束数量):**

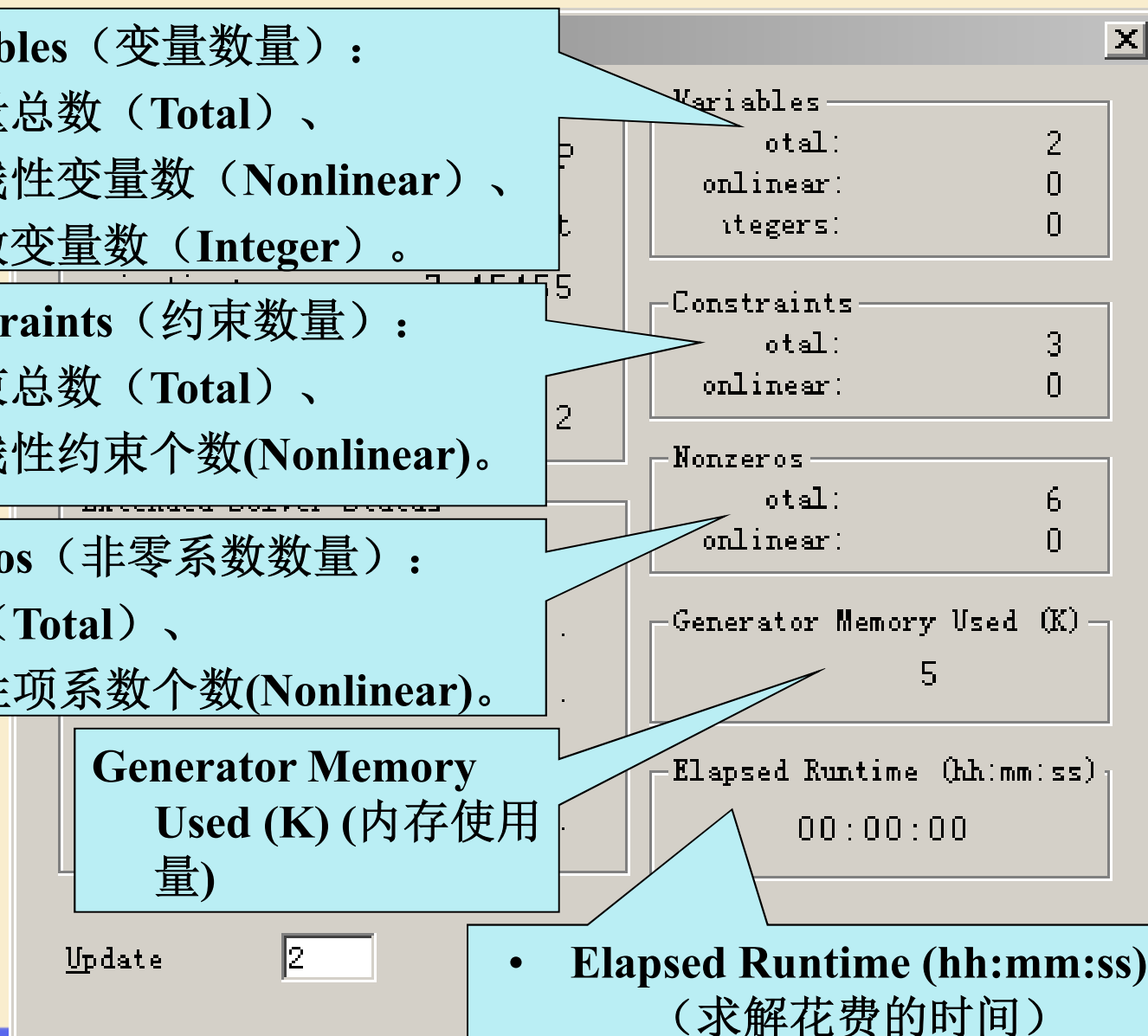
约束总数 (Total)、  
非线性约束个数 (Nonlinear)。

**Nonzeros (非零系数数量):**

总数 (Total)、  
非线性项系数个数 (Nonlinear)。

**Generator Memory  
Used (K) (内存使用  
量)**

• **Elapsed Runtime (hh:mm:ss)**  
(求解花费的时间)



# 运行状态窗口

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)

求解器(求解程序)状态框

LINGO Solver Status [example1.lbr]

Solver Status		Warning
Model	LP	
State	Global Optimum	
Objective:	1.000000000	
Feasibility:	0.000000000	
Iterations:	2	
Extended Solver Status		Nonzeros
Solver	Simplex	
Best	1.000000000	
Obj Bound:	1.000000000	
Steps:	...	
Active:	...	
Elap		00:00:00
Update	2	
Interrupt Solver		Close

解的目标函数值

目前为止的迭代次数

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)

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命令形式给出的约束)

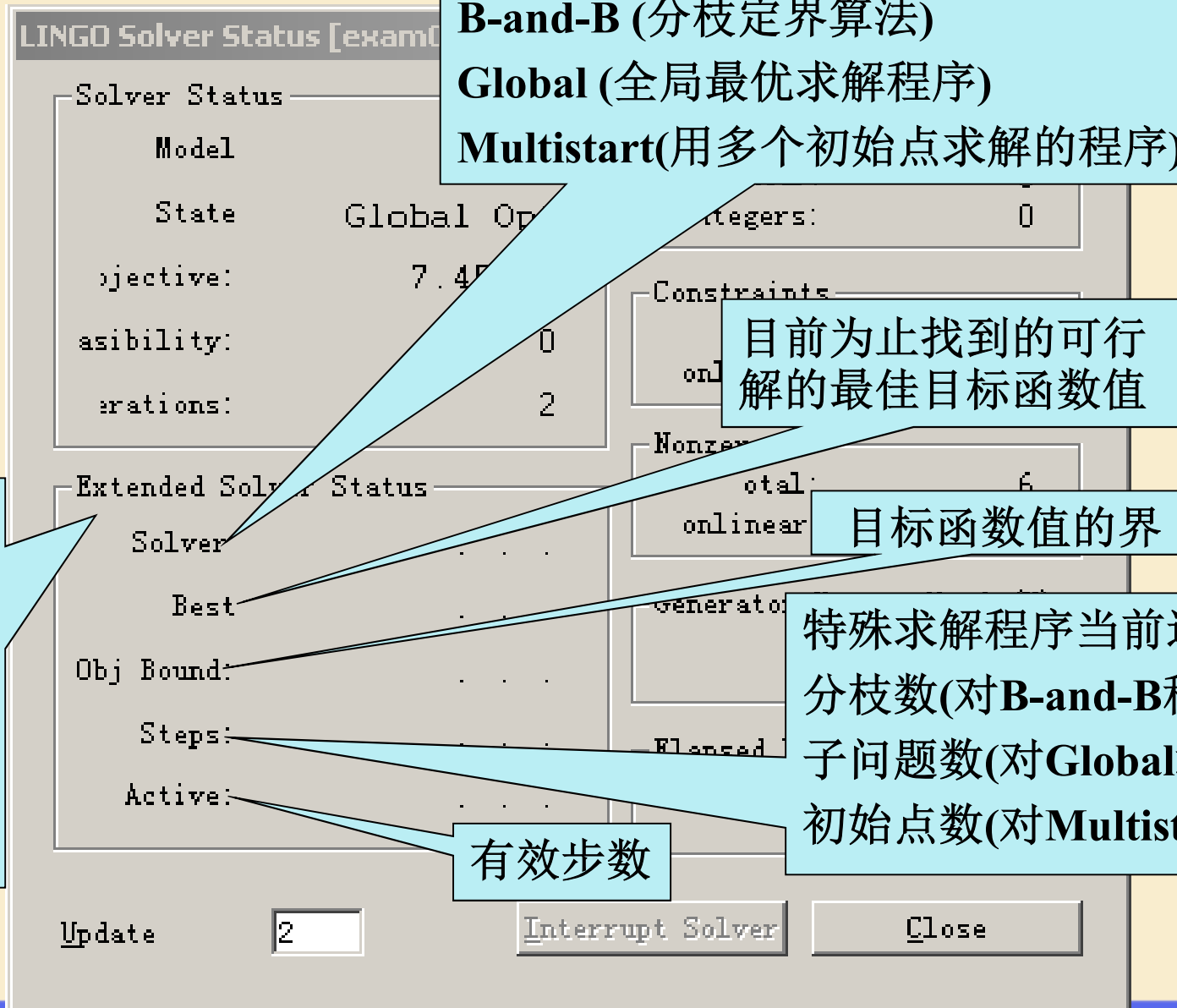
- 运行状态窗口

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)



- 运行状态窗口

注：凡是可以从一个约束直接解出变量取值时，这个变量就不认为是决策变量而是固定变量，不列入统计中；只含有固定变量的约束也不列入约束统计中。



## LINGO模型的最基本特征

- 将目标函数的表示方式从“MAX”变成了“MAX=”;
- “ST”(SubjectTo)在LINGO模型中不需要, 被删除;
- 在系数与变量之间增加运算符“\*” (即乘号不能省略);
- 每行(目标、约束和说明语句)后面增加一个分号“;”;
- 约束的名字被放到 “[ ]”中;
- LINGO中模型以“MODEL: ”开始, 以“END”结束。对简单的模型, 这两个语句也可以省略。

## 一个简单的LINGO程序

例 直接用LINGO来解如下二次规划问题：

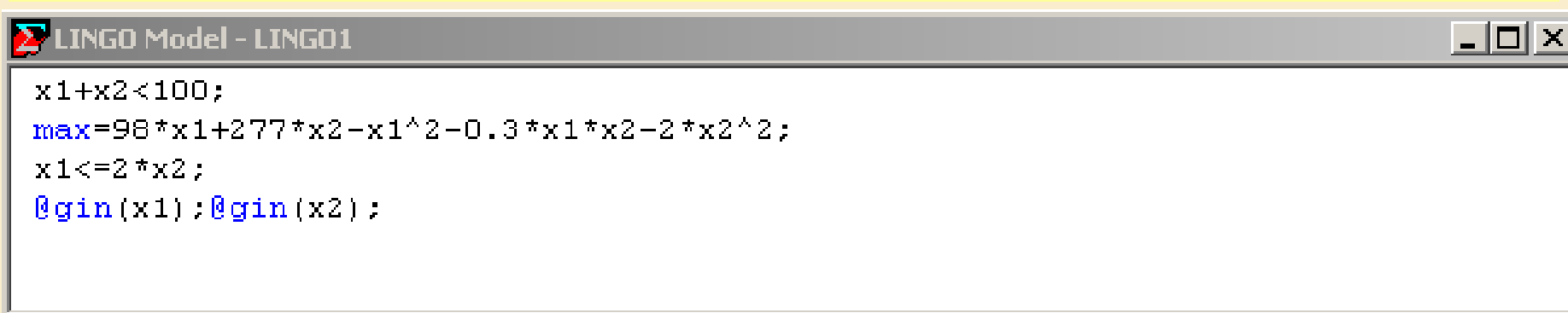
$$\text{Max} \quad 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 2x_2^2 \quad (1)$$

$$\text{s.t.} \quad x_1 + x_2 \leq 100 \quad (2)$$

$$x_1 \leq 2x_2 \quad (3)$$

$$x_1, x_2 \geq 0 \quad \text{为整数} \quad (4)$$

输入窗口如下：



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

## 程序语句输入的备注:

- **LINGO**总是根据“**MAX=**”或“**MIN=**”寻找目标函数，而除注释语句和**TITLE**语句外的其他语句都是约束条件，因此语句的顺序并不重要。
- 限定变量取整数值的语句为“**@GIN(X1)**”和“**@GIN(X2)**”，不可以写成“**@GIN(2)**”，否则**LINGO**将把这个模型看成没有整数变量。
- **LINGO**中函数一律需要以“**@**”开头，其中整型变量函数（**@BIN**、**@GIN**）和上下界限定函数（**@FREE**、**@SUB**、**@SLB**）。而且0/1变量函数是**@BIN**函数。

输出结果:

运行菜单命令“LINGO|Solve”

最大利润=11077.5

Solution Report - LINGO1

Local optimal solution found at iteration: 457  
Objective value: 11077.50

Variable	Value	Reduced Cost
X1	35.00000	-8.500020
X2	65.00000	-6.500069

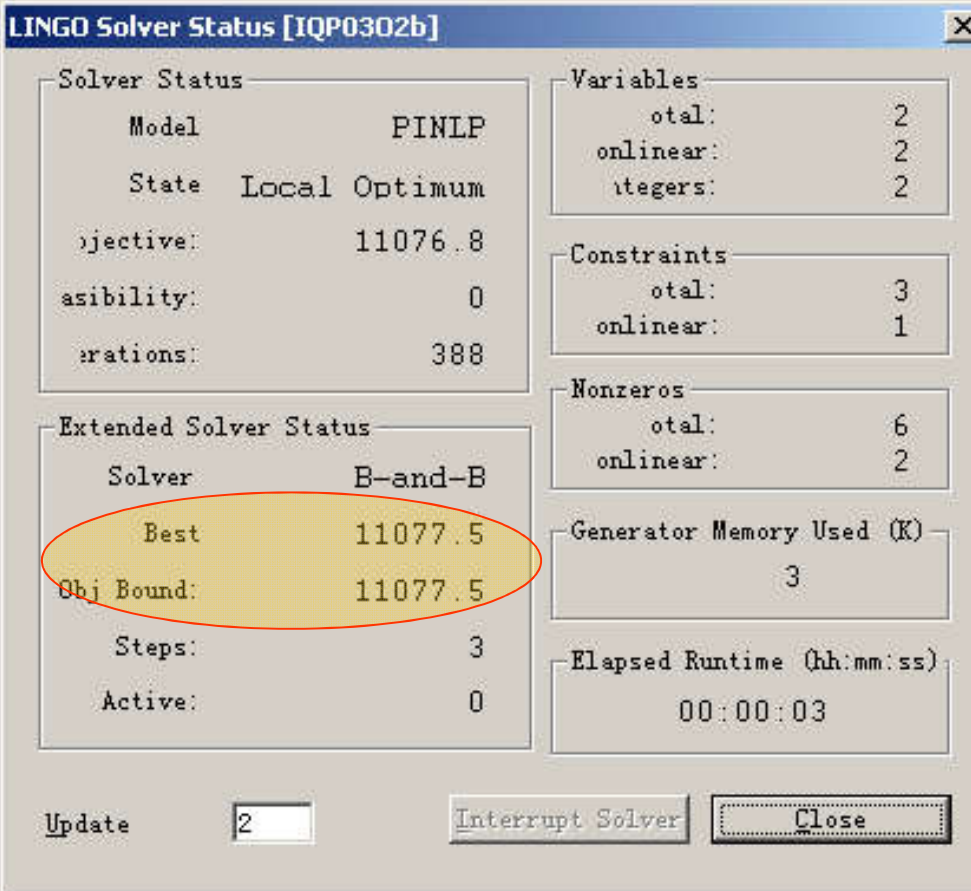
Row	Slack or Surplus	Dual Price
1	0.000000	0.000000
2	11077.50	1.000000
3	95.00000	0.000000

最优整数解  
 $X=(35, 65)$

## 输出结果备注:

LINGO是将它作为PINLP(纯整数非线性规划)来求解,因此找到的是局部最优解。

通过菜单“WINDOW|  
Status Window”看到状态窗  
口,可看到最佳目标值  
“Best Obj”与问题的上界  
“Obj Bound”已经是一样的,  
当前解的最大利润与这两个  
值非常接近,是计算误差引  
起的。如果采用全局最优求  
解程序(后面介绍),可以验  
证它就是全局最优解。



The screenshot shows the 'LINGO Solver Status [IQP0302b]' window. It contains several sections: 'Solver Status', 'Variables', 'Constraints', 'Nonzeros', 'Generator Memory Used (K)', 'Elapsed Runtime (hh:mm:ss)', and 'Extended Solver Status'. The 'Extended Solver Status' section is highlighted with a red oval, showing 'Best' and 'Obj Bound' both at 11077.5. The 'Solver Status' section shows 'Model' as PINLP, 'State' as Local Optimum, 'Objective' as 11076.8, 'Feasibility' as 0, and 'Iterations' as 388. The 'Variables' section shows 2 total, 2 nonlinear, and 2 integers. The 'Constraints' section shows 3 total and 1 nonlinear. The 'Nonzeros' section shows 6 total and 2 nonlinear. The 'Generator Memory Used (K)' is 3. The 'Elapsed Runtime (hh:mm:ss)' is 00:00:03. At the bottom, there are buttons for 'Update', 'Interrupt Solver', and 'Close'.

Solver Status	
Model	PINLP
State	Local Optimum
Objective:	11076.8
Feasibility:	0
Iterations:	388

Variables	
total:	2
nonlinear:	2
integers:	2

Constraints	
total:	3
nonlinear:	1

Nonzeros	
total:	6
nonlinear:	2

Generator Memory Used (K)	
	3

Elapsed Runtime (hh:mm:ss)	
	00:00:03

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

Update  Interrupt Solver Close

# 一个简单的LINGO程序

## LINGO的基本用法的几点注意事项

- LINGO中不区分大小写字母；变量和行名可以超过8个字符，但不能超过32个字符，且必须以字母或下划线\_开头。
- 用LINGO解优化模型时已假定所有变量非负(除非用限定变量取值范围的函数@free或@sub或@slb另行说明)。
- 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LINGO求解时的效率，应尽可能采用线性表达式定义目标和约束(如果可能的话)。
- 语句是组成LINGO模型的基本单位,每个语句都以分号结尾，编写程序时应注意模型的可读性。例如：一行只写一个语句，按照语句之间的嵌套关系对语句安排适当的缩进，增强层次感。
- 以感叹号开始的是说明语句(说明语句也需要以分号结束)。

## 2.在LINGO中使用集合

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题

## 集合的基本用法和LINGO模型的基本要素

例 理解LINGO建模语言最重要的是理解集合(Set)及其属性(Attribute)的概念。SAILCO公司需要决定下四个季度的帆船生产量及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, 也就是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。



## 问题的模型(可以看出是LP模型)

目标函数是所有费用的和  $\text{MIN} \sum_{I=1,2,3,4} \{400RP(I) + 450OP(I) + 20INV(I)\}$

约束条件主要有两个:

1) 能力限制:  $RP(I) < 40, I = 1, 2, 3, 4$

2) 产品数量的平衡方程:

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

$$INV(0) = 10$$

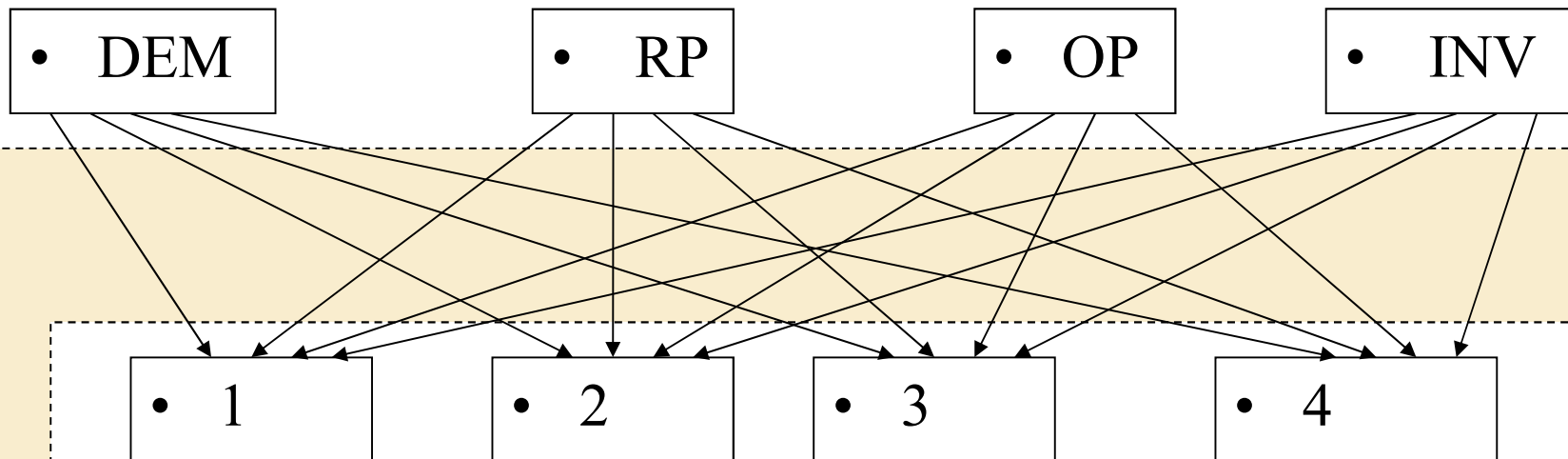
加上变量的非负约束

注：LINDO中没有数组，只能对每个季度分别定义变量，如正常产量就要有RP1，RP2，RP3，RP4 4个变量等。写起来就比较麻烦，尤其是更多(如1000个季度)的时候。

记四个季度组成的集合QUARTERS={1, 2, 3, 4}，它们就是上面数组的下标集合，而数组DEM,RP,OP, INV对集合QUARTERS中的每个元素1, 2, 3, 4分别对应于一个值。LINGO正是充分利用了这种数组及其下标的关系，引入了“集合”及其“属性”的概念，把QUARTERS={1, 2, 3, 4}称为集合，把DEM,RP,OP, INV称为该集合的属性(即定义在该集合上的属性)。

## 集合及其属性

- QUARTERS集合的属性



- QUARTERS集合

# 集合元素及集合的属性确定的所有变量

集合QUARTERS 的元素		1	2	3	4
定义在集 合 QUARTE RS 上的属性	DEM	DEM(1)	DEM(2)	DEM(3)	DEM(4)
	RP	RP(1)	RP(2)	RP(3)	RP(4)
	OP	OP(1)	OP(2)	OP(3)	OP(4)
	INV	INV(1)	INV(2)	INV(3)	INV(4)

# LINGO中定义集合及其属性

## LP模型在LINGO中的一个典型输入方式

```
MODEL:
  SETS:
    QUARTERS/1,2,3,4/:DEM,RP,OP,INV;
  ENDSSETS
  MIN=@SUM(QUARTERS:400*RP+450*OP+20*INV);
  @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);
  DATA:
    DEM=40,60,75,25;
  ENDDATA
END
```

以“MODEL:”开始

集合定义部分从(“SETS:”到“ENDSETS”): 定义集合及其属性

给出优化目标和约束

集合定义部分从(“DATA:”到“ENDDATA”)

以“END”结束

## 目标函数的定义方式

**@SUM(集合（下标）：关于集合的属性的表达式)**

对语句中冒号“:”后面的表达式，按照“:”前面的集合指定的下标（元素）进行求和。

本例中目标函数也可以等价地写成

**@SUM(QUARTERS(i): 400\*RP(i) +450\*OP(i) +20\*INV(i) ),**

“@SUM”相当于求和符号“ $\Sigma$ ”，

“QUARTERS(i)”相当于“iQUARTERS”的含义。

由于本例中目标函数对集合QUARTERS的所有元素(下标) 都要求和，所以可以将下标i省去。

## 约束的定义方式

### 循环函数

**@FOR(集合(下标): 关于集合的属性的约束关系式)**

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

本例中，每个季度正常的生产能力是40条帆船，这正是语句“**@FOR(QUARTERS(I):RP(I)<40);**”的含义。

由于对所有元素(下标I),约束的形式是一样的，所以也可以像上面定义目标函数时一样，将下标i省去，这个语句可以简化成“**@FOR(QUARTERS:RP<40);**”。

## 约束的定义方式

本例中，对于产品数量的平衡方程，由于下标 $i=1$ 时的约束关系式与 $i=2, 3, 4$ 时有所区别，所以不能省略下标“ $i$ ”。实际上， $i=1$ 时要用到变量 $INV(0)$ ，但定义的属性变量中 $INV$ 不包含 $INV(0)$  ( $INV(0)=10$ 是一个已知的)。

为了区别 $i=1$ 和 $i=2, 3, 4$ ，把 $i=1$ 时的约束关系式单独写出，即“ $INV(1)=10+RP(1)+OP(1)-DEM(1);$ ”；而对 $i=2, 3, 4$ 对应的约束，对下标集合的元素（下标 $i$ ）增加了一个逻辑关系式“ $i\#GT\#1$ ”（这个限制条件与集合之间有一个竖线“ $|$ ”分开，称为过滤条件）。

限制条件“ $i\#GT\#1$ ”是一个逻辑表达式，意思就是 $i>1$ ；

“ $\#GT\#$ ”是逻辑运算符号，意思是“大于（Greater Than的字首字母缩写）”。



## 问题的求解：运行菜单命令“LINGO|Solve”

**Solution Report - LINGO1**

Global optimal solution found at iteration: 0  
Objective value: 78450.00

**最小成本=78450**

	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

全局最优解

RP=(40,40,40,25),

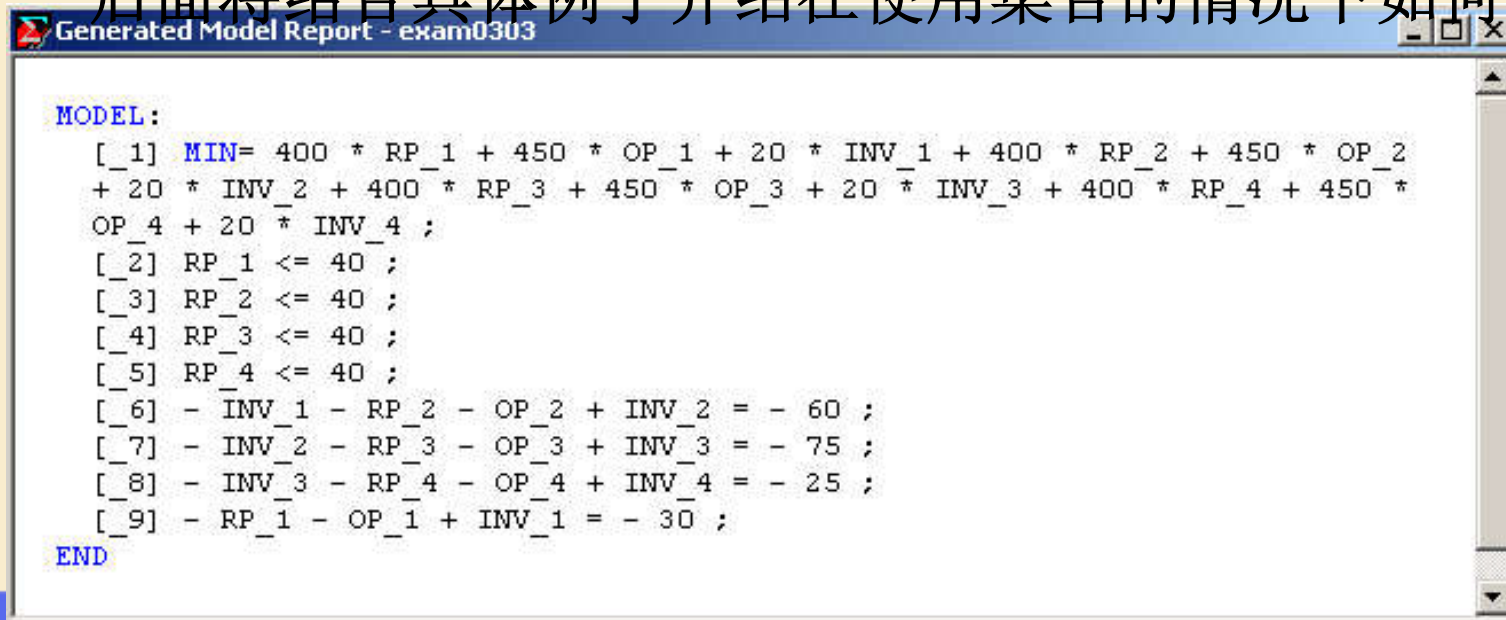
OP=(0,10,35,0)

**注：**由于输入中没有给出行名，所以行名是系统自动按照行号1-9生成的。

选择菜单命令“**LINGO|Generate|Display model (Ctrl+G)**”，可以得到展开形式的模型(如图)，可以看到完整的模型，也能确定行号(行号放在方括号“**[ ]**”中，且数字前面带下划线“**\_**”)。

最好在输入模型时用户主动设定约束的行名(即约束名)，使程序清晰些。单一约束的行名设置方法就是将行名放在方括号“**[ ]**”中，置于约束之前。

后面将结合具体例子介绍在使用集合的情况下如何设置行名。



```
Generated Model Report - exam0303

MODEL:
[_1] MIN= 400 * RP_1 + 450 * OP_1 + 20 * INV_1 + 400 * RP_2 + 450 * OP_2
+ 20 * INV_2 + 400 * RP_3 + 450 * OP_3 + 20 * INV_3 + 400 * RP_4 + 450 *
OP_4 + 20 * INV_4 ;
[_2] RP_1 <= 40 ;
[_3] RP_2 <= 40 ;
[_4] RP_3 <= 40 ;
[_5] RP_4 <= 40 ;
[_6] - INV_1 - RP_2 - OP_2 + INV_2 = - 60 ;
[_7] - INV_2 - RP_3 - OP_3 + INV_3 = - 75 ;
[_8] - INV_3 - RP_4 - OP_4 + INV_4 = - 25 ;
[_9] - RP_1 - OP_1 + INV_1 = - 30 ;
END
```

## 小结:LINGO模型最基本的组成要素

一般来说, LINGO中建立的优化模型可以由五个部分组成, 或称为五“段”(SECTION):

(1) 集合段 (SETS): 以“SETS:”开始, “ENDSETS”结束, 定义必要的集合变量 (SET) 及其元素 (MEMBER, 含义类似于数组的下标) 和属性 (ATTRIBUTE, 含义类似于数组)。

如上例中定义了集合quarters(含义是季节), 它包含四个元素即四个季节指标(1,2,3,4), 每个季节都有需求(DEM)、正常生产量(RP)、加班生产量(OP)、库存量(INV)等属性(相当于数组, 数组下标由quarters元素决定)。一旦这样的定义建立起来, 如果quarters的数量不是4而是1000, 只需扩展其元素为1,2,...,1000, 每个季节仍然都有DEM,RP,OP,INV这样的属性(这些量的具体数值如果是常量, 则可在数据段输入; 如果是未知数, 则可在初始段输入初值)。当quarters的数量不是4而是1000时, 没有必要把1,2, ...,1000全部一个一个列出来, 而是可以如下定义quarters集合: “quarters/1..1000/:DEM,RP,OP,INV;”, “1..1000”的意思就是从1到1000的所有整数。

**(2) 目标与约束段：**目标函数、约束条件等，没有段的开始和结束标记，因此实际上就是除其它四个段(都有明确的段标记)外的LINGO模型。

这里一般要用到LINGO的内部函数，尤其是与集合相关的求和函数@SUM和循环函数@FOR等。

上例中定义的目标函数与quarters的元素数目是4或1000并无具体的关系。约束的表示也类似。

(3) 数据段(DATA): 以“DATA:”开始,“ENDDATA”结束,对集合的属性(数组)输入必要的常数数据。

格式为: “**attribute(属性) = value\_list(常数列表);**”

常数列表(value\_list)中数据之间可以用逗号“,”分开,也可以用空格分开(回车等价于一个空格),如上面对DEM的赋值也可以写成“**DEM=40 60 75 25;**”。

在LINGO模型中,如果想在运行时才对参数赋值,可以在数据段使用输入语句。但这仅能用于对单个变量赋值,输入语句格式为: “**变量名 = ? ;**”。例如,上例中如果需要在求解模型时才给出初始库存量(记为A),则可以在模型中数据段写上语句: “**A = ?;**”

在求解时LINDO系统给出提示界面,等待用户输入变量A的数值。当然,此时的约束语句 **INV(1)=10+RP(1)+OP(1)-DEM(1);**

也应该改写成 **INV(1)=A+RP(1)+OP(1)-DEM(1);**

这样,模型就可以计算任意初始库存量(而不仅仅只能计算初始库存量为10)的情况了。

(4) **初始段**(*INIT*): 以“INIT:”开始, “ENDINIT”结束, 对集合的属性(数组)定义初值(因为求解算法一般是迭代算法, 所以用户如果能给出一个比较好的迭代初值, 对提高算法的计算效果是有益的)。

如果有一个接近最优解的初值, 对LINGO求解模型是有帮助的。定义初值的格式为:

“**attribute (属性) = value\_list (常数列表);**”

这与数据段中的用法是类似的。

上例中没有初始化部分, 我们将在下一个例子中举例说明。



**(5) 计算段(CALC):** 以“CALC:”开始,  
“ENDCALC”结束, 对一些原始数据进行计算处理。

在实际问题中, 输入的数据通常是原始数据, 不一定能在模型中直接使用, 可以在这个段对这些原始数据进行一定的“预处理”, 得到模型中真正需要的数据。

例如上例, 如果希望得到全年的总需求和季度平均需求, 可以增加这个段:

**CALC:**

**T\_DEM = @SUM(quarters: DEM); !总需求;**

**A\_DEM = T\_DEM / @size(quarters); !平均需求;**

**ENDCALC**

在计算段中也可以使用集合函数 (其中函数@size(quarters)表示集合quarters的元素个数, 这里也就是4)。这时, 变量T\_DEM的值就是总需求, A\_DEM的值就是平均需求 (如果需要的话, 这两个变量就可以在程序的其它地方作为常数使用了)。

注: 上面的两个语句不能交换顺序, 因为计算A\_DEM必须要用到T\_DEM的值。此外, 在计算段中只能直接使用赋值语句, 而不能包含需要经过解方程或经过求解优化问题以后才能决定的变量。



## 基本集合与派生集合

**例3.4** 建筑工地的位置(用平面坐标 $a, b$ 表示, 距离单位: 公里)及水泥日用量 $d$ (吨)下表给出。有两个临时料场位于P (5,1), Q (2, 7), 日储量各有20吨。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。两个新的料场应建在何处, 节省的吨公里数有多大?

	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



## 建立模型

记工地的位置为 $(a_i, b_i)$ ，水泥日用量为 $d_i, i = 1, \dots, 6$ ；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 $c_{ij}$ 。

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

$$\text{s.t.} \quad \sum_{j=1}^2 c_{ij} = d_i, \quad i = 1, 2, \dots, 6 \quad (2)$$

$$\sum_{i=1}^6 c_{ij} \leq e_j, \quad j = 1, 2 \quad (3)$$

使用现有临时料场时，决策变量只有 $c_{ij}$ （非负），所以这是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。先解NLP模型，而把现有临时料场的位置作为初始解告诉LINGO。

## 本例中集合的概念

利用集合的概念，可以定义需求点**DEMAND**和供应点**SUPPLY**两个集合，分别有6个和2个元素(下标)。但决策变量(运送量)  $c_{ij}$  与集合**DEMAND**和集合**SUPPLY**都有关系的。该如何定义这样的属性？

集合的属性相当于以集合的元素为下标的数组。这里的相当于二维数组。它的两个下标分别来自集合**DEMAND**和**SUPPLY**，因此可以定义一个由二元对组成的新的集合，然后将  $c_{ij}$  定义成这个新集合的属性。

# 输入程序

定义了三个集合，其中LINK在前两个集合DEMAND 和SUPPLY的基础上定义

LINGO Model - location

```
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);
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;
enddata
```

表示集合LINK中的元素就是集合DEMAND和SUPPLY的元素组合成的有序二元组，从数学上看LINK是DEMAND 和SUPPLY的笛卡儿积，也就是说

$LINK = \{ (S, T) | S \in DEMAND, T \in SUPPLY \}$   
因此，其属性C也就是一个6\*2的矩阵（或者说是含有12个元素的二维数组）。

LINGO建模语言也称为矩阵生成器（MATRIX GENERATOR）。类似DEMAND 和SUPPLY直接把元素列举出来的集合，称为**基本集合(primary set)**，而把LINK这种基于其它集合而派生出来的二维或多维集合称为**派生集合(derived set)**。由于是DEMAND 和SUPPLY生成了派生集合LINK，所以DEMAND 和SUPPLY 称为LINK的**父集合**。

END

# 输入程序

```

LINGO Model - local
MODEL:
Title Local
sets:
    demand/1..6/:a,b
    supply/1..2/:x
    link(demand,supply)
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;
    d=3,5,4,7,6,11; e=20,20;
enddata
init:
    !initial locations for the supply;
    x,y=5,1,2,7;
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: @free(X); @free(Y); );
END

```

同理，数据段中对常数数组A, B的赋值语句也可以写成  
A, B=1.25 1.25 8.75 0.75 0.5 4.75 5.75 5 3 6.5 7.25 7.75;

LINGO对数据是按列赋值的  
语句的实际赋值顺序是  
X=(5,2), Y=(1,7), 而不是  
X=(5,1), Y=(2,7)  
等价写法:

“X=5,2; Y=1,7; ”

初始段

# 输入程序

```

MODEL:
Title Location Problem;
sets:
    demand / 1..6/: a, b, d;
    supply / 1..4/: e;
    link (demand, supply);
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); );
END
  
```

在程序开头用TITLE语句对这个模型取了一个标题“LOCATION PROBLEM;”并且对目标行([OBJ])和两类约束

(DEMAND\_CON、SUPPLY\_CON)分别进行了命名(请特别注意这里约束命名的特点)。

定义目标和约束，与前例的方法是类似(这里包含了派生集合)，请特别注意进一步体会集合函数@SUM和@FOR的用法。

由于新建料场的位置理论上讲可以是任意的，所以在约束的最后(模型的“END”语句上面的一行)用@free函数取消了变量X、Y的非负限制

(初始



解答:运行菜单命令“LINGO|Solve”

局部最优解 $X(1)=7.249997$ ,  $X(2)=5.695940$ ,  
 $Y(1)=7.749998$ ,  $Y(2)=4.928524$ ,  $C$  (略),  
最小运量=89.8835(吨公里)。

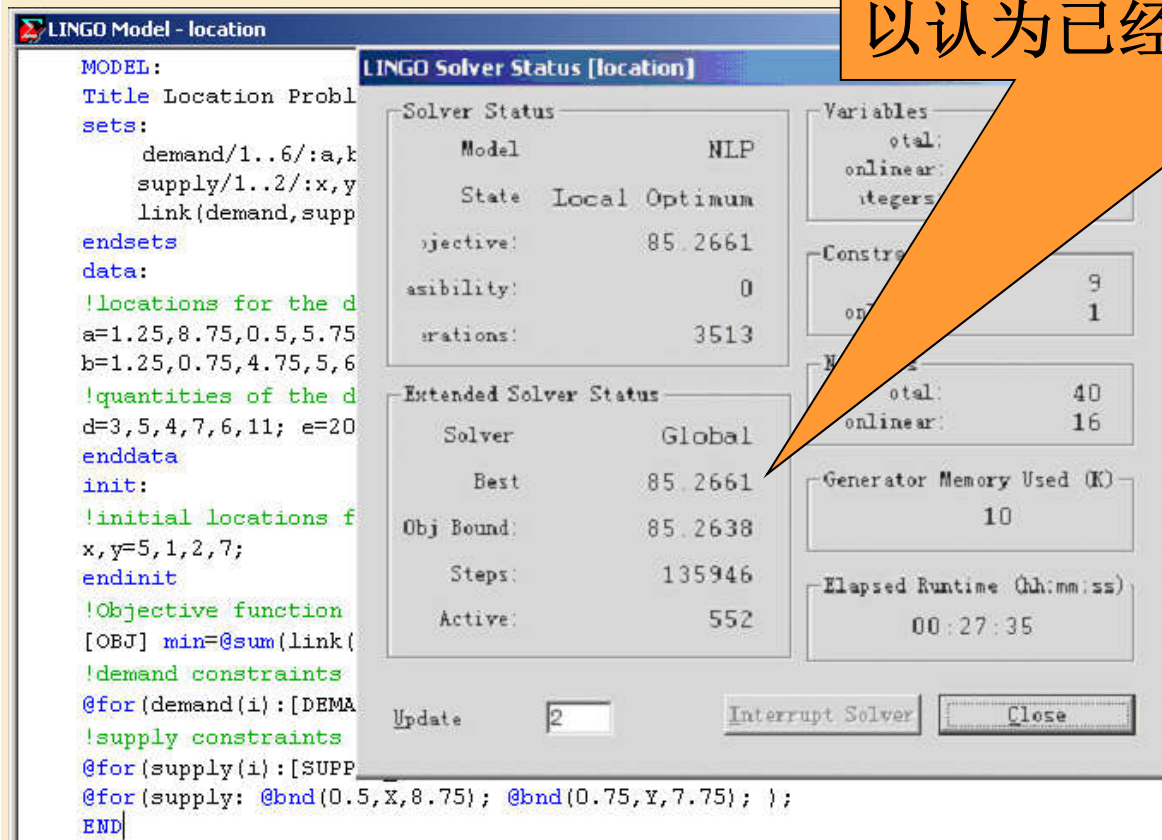
问题:最小运量89.8835是不是全局最优

是用“LINGO|Options”菜单命令打开选项对话框, 在  
“Global Solver”选项卡上选择“Use Global Solver”, 激活全局最优求解程序。

## 问题:最小运量89.8835是不是全局最优

为减少计算工作量,对X, Y  
新建料场的位置可以是任意  
工地太远,至少不应该超出  
最小值决定的矩形之外,即:

此时目标函数值的下界 (Obj Bound=85.2638) 与目前得到的最好的可行解的目标函数值 (Best Obj=85.2661) 相差已经非常小, 可以认为已经得到了全局最优解。



**LINGO Model - location**

```

MODEL:
Title Location Probl
sets:
demand/1..6/:a,k
supply/1..2/:x,y
link(demand,supp
endsets
data:
!locations for the d
a=1.25,8.75,0.5,5.75
b=1.25,0.75,4.75,5,6
!quantities of the d
d=3,5,4,7,6,11; e=20
enddata
init:
!initial locations f
x,y=5,1,2,7;
endinit
!Objective function
[OBJ] min=@sum(link(
!demand constraints
@for(demand(i):[DEMA
!supply constraints
@for(supply(i):[SUPP
@for(supply: @bnd(0.5,X,8.75); @bnd(0.75,Y,7.75); );
END
  
```

**LINGO Solver Status [location]**

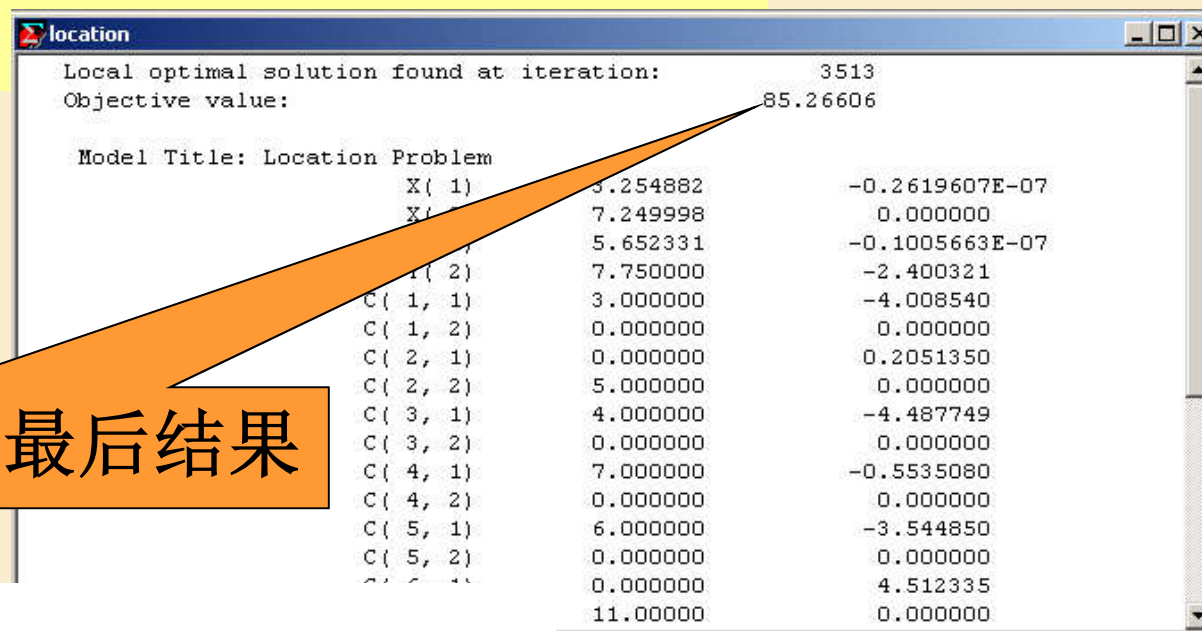
Solver Status	
Model	NLP
State	Local Optimum
Objective:	85.2661
Feasibility:	0
Iterations:	3513

Extended Solver Status	
Solver	Global
Best	85.2661
Obj Bound:	85.2638
Steps:	135946
Active:	552

Update  Interrupt Solver Close

上这个条件取代模型  
END上面的行, 运行  
NLP模型, 全局最优  
求解程序花费的时间  
仍然很长, 运行27分  
35秒时人为终止求解  
(按下“Interrupt  
Solver”按钮)得到左  
边模型窗口和全局求  
解器的状态窗口

# 计算结果



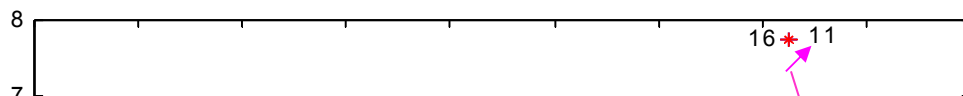
location

Local optimal solution found at iteration: 3513  
Objective value: 85.26606

Model Title: Location Problem

X( 1)	5.254882	-0.2619607E-07
X( 2)	7.249998	0.000000
X( 3)	5.652331	-0.1005663E-07
X( 4)	7.750000	-2.400321
C( 1, 1)	3.000000	-4.008540
C( 1, 2)	0.000000	0.000000
C( 2, 1)	0.000000	0.2051350
C( 2, 2)	5.000000	0.000000
C( 3, 1)	4.000000	-4.487749
C( 3, 2)	0.000000	0.000000
C( 4, 1)	7.000000	-0.5535080
C( 4, 2)	0.000000	0.000000
C( 5, 1)	6.000000	-3.544850
C( 5, 2)	0.000000	0.000000
	0.000000	4.512335
	11.00000	0.000000

可以认为是模型的最后结果



附注：如果要把料厂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。只需要把初始段的“X Y=5, 1, 2, 7;”语句移到数据段就可以了。此时，运行结果告诉我们得到全局最优解（变量C的取值这里略去），最小运量136.2275(吨公里)。

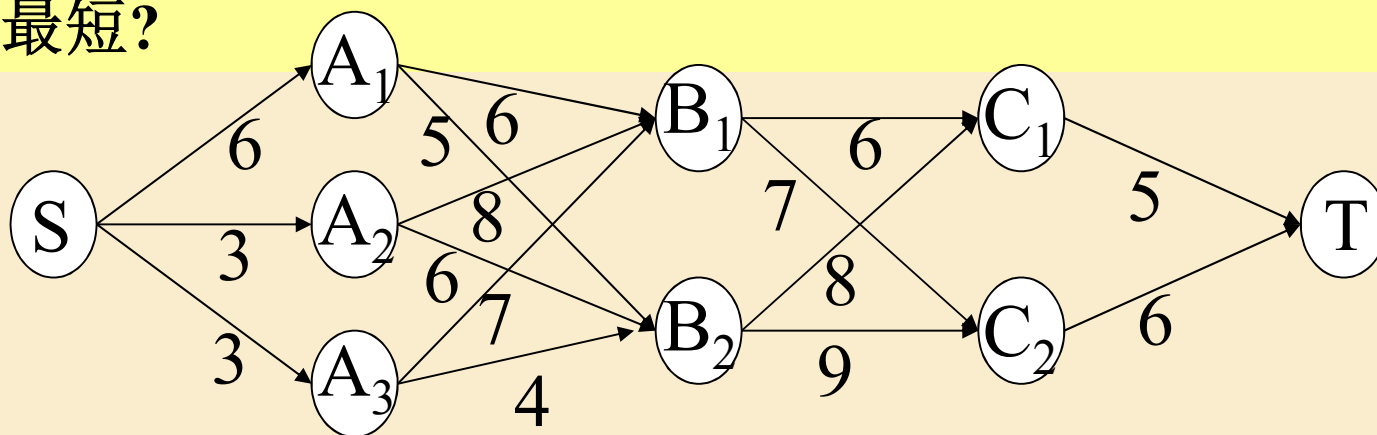
示意图：  
“+”



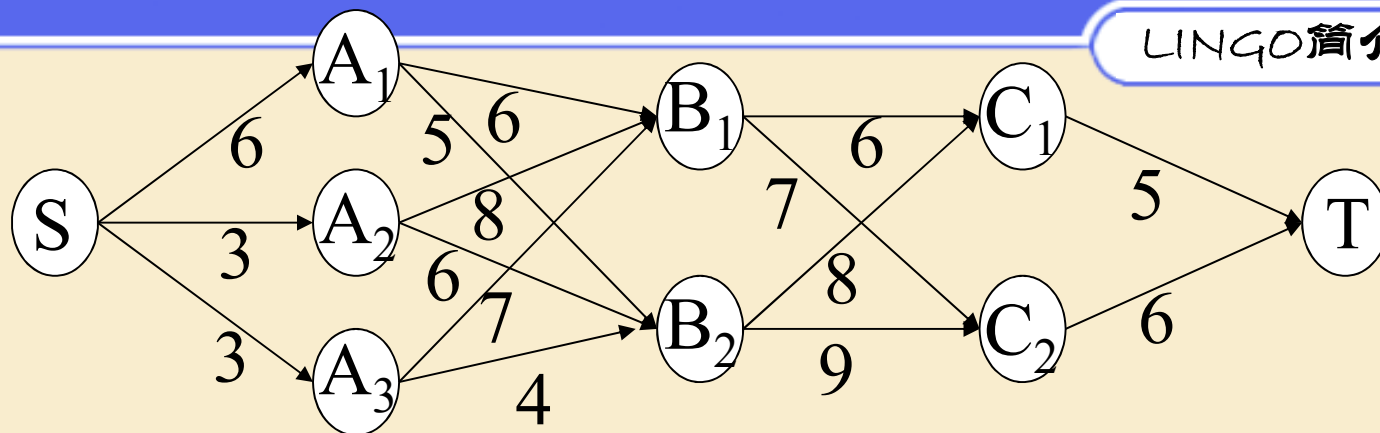
## 稠密集合与稀疏集合

包含了两个基本集合构成的所有二元有序对的派生集合称为**稠密集合(简称稠集)**。有时候,在实际问题中,一些属性(数组)只在笛卡儿积的一个真子集合上定义,这种派生集合称为**稀疏集合(简称疏集)**。

例 (最短路问题) 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. 下图表示的是公路网,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?



## 分析



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  $C_1$  的最优行驶路线;

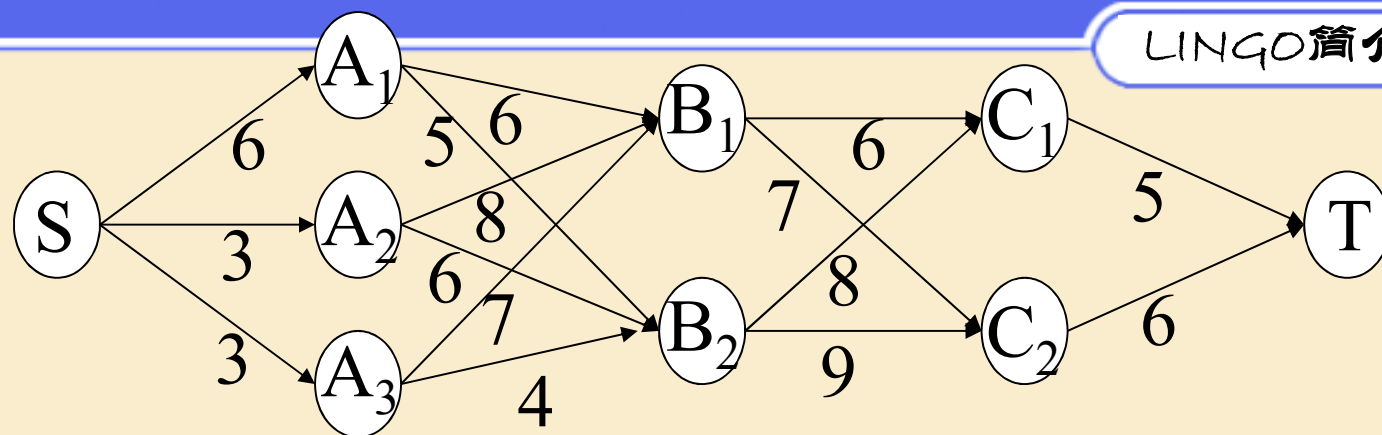
假设  $P$  经过城市  $C_2$ , 则  $P$  中从S到  $C_2$  的子路也一定是从S到  $C_2$  的最优行驶路线.

因此, 为得到从S到T的最优行驶路线, 只需要先求出从S到  $C_k (k=1,2)$  的最优行驶路线, 就可以方便地得到从S到T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, 只需要先求出从S到  $B_j (j=1,2)$  的最优行驶路线;

为了求出从S到  $B_j (j=1,2)$  的最优行驶路线, 只需要先求出从S到  $A_i (i=1,2,3)$  的最优行驶路线. 而S到  $A_i (i=1,2,3)$  的最优行驶路线是很容易得到的(实际上, 此例中S到  $A_i (i=1,2,3)$  只有唯一的道路)

## 分析

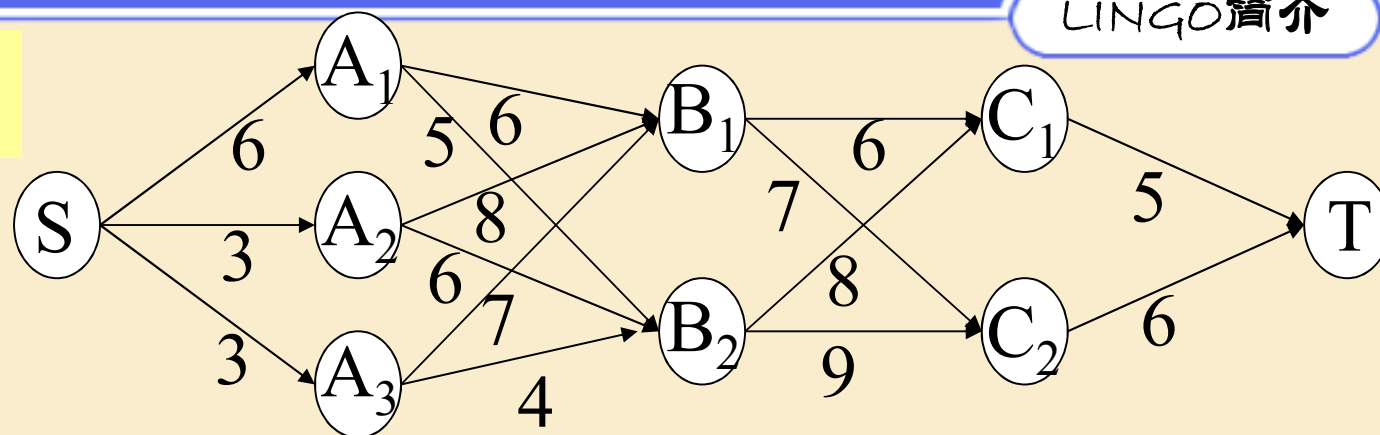


此例中可把从S到T的行驶过程分成4个阶段,即  $S \rightarrow A_i$  ( $i=1,2$ 或 $3$ ),  $A_i \rightarrow B_j$  ( $j=1$ 或 $2$ ),  $B_j \rightarrow C_k$  ( $k=1$ 或 $2$ ),  $C_k \rightarrow T$ . 记  $d(Y,X)$ 为城市Y与城市X之间的直接距离(若这两个城市之间没有道路直接相连,则可以认为直接距离为 $\infty$ ), 用  $L(X)$ 表示城市S到城市X的最优行驶路线的路长:

$$L(S) = 0; \quad (1)$$

$$L(X) = \min_{Y \neq X} \{L(Y) + d(Y, X)\}, \quad X \neq S. \quad (2)$$

## 本例的计算



$$L(A_1) = 6, L(A_2) = 3, L(A_3) = 3;$$

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

$$L(B_2) = \min \{L(A_1) + 5, L(A_2) + 6, L(A_3) + 4\} = 7 = L(A_3) + 4;$$

$$L(C_1) = \min \{L(B_1) + 6, L(B_2) + 8\} = 15 = L(B_2) + 8;$$

$$L(C_2) = \min \{L(B_1) + 7, L(B_2) + 9\} = 16 = L(B_2) + 9;$$

$$L(T) = \min \{L(C_1) + 5, L(C_2) + 6\} = 20 = L(C_1) + 5.$$

这种计算方法  
在数学上称为  
**动态规划**  
(Dynamic  
Programming)

所以, 从S到T的最优行驶路线的路长为20. 进一步分析以上求解过程, 可以得到从S到T的最优行驶路线为

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

# 本例的LINGO求解

“CITIES”(城市):一个基本集合(元素通过枚举给出)

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

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) | i # GT # @index(S):

L( i) = @MIN( ROADS( j, i): L( j) + D( j, i));

end

L:CITIES对应的属性变量(我们要求的最短路长)

属性L(i)表示城市s到城市i的最优行驶路线的路长;

“ROADS”(道路):由CITIES导出的一个派生集合(请特别注意其用法),由于只有一部分城市之间有道路相连,所以不应该把它定义成稠密集合,将其元素通过枚举给出,这就是一个稀疏集合。

D:稀疏集合ROADS对应的属性变量(给定的距离)

因为L(S)=0;

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;



## 本例的LINGO求解

从模型中还可以看出：这个LINGO程序可以没有目标函数，这在LINGO中，可以用来找可行解(解方程组和不等式组)。

```

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)|i#GT#@index(S):
  L( i) = @MIN( ROADS( j, i): L( j) + D( j, i));
end

```

!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) 是城市到j的直接距离(已知);

在数据段对L进行赋值，只有L(S)=0已知，后面的值为空(但位置必须留出来，即逗号“，”一个也不能少，否则会出错)。如果这个语句直接写成“L=0;”，语法上看也是对的，但其含义是L所有元素的取值全部为0，所以也会与题意不符。

!这就是前面写出的最短路关系式;

# 本例的LINGO求解

```

LINGO Model - ShorestPath

model:
SETS:
  CITIES /S,A1,A2,A3,B1,B2,C1,C2/;
  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)|i#GT#@index(S):
  L( i) = @MIN( ROADS( j, i): L( j) + D( j, i));
end

```

虽然集合CITIES中的元素不是数字，但当它以CITIES(I)的形式出现在循环中时，引用下标I却实际上仍是正整数，也就是说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)。

派生集合ROADS表示的是网络中的道路(弧)；

由

属

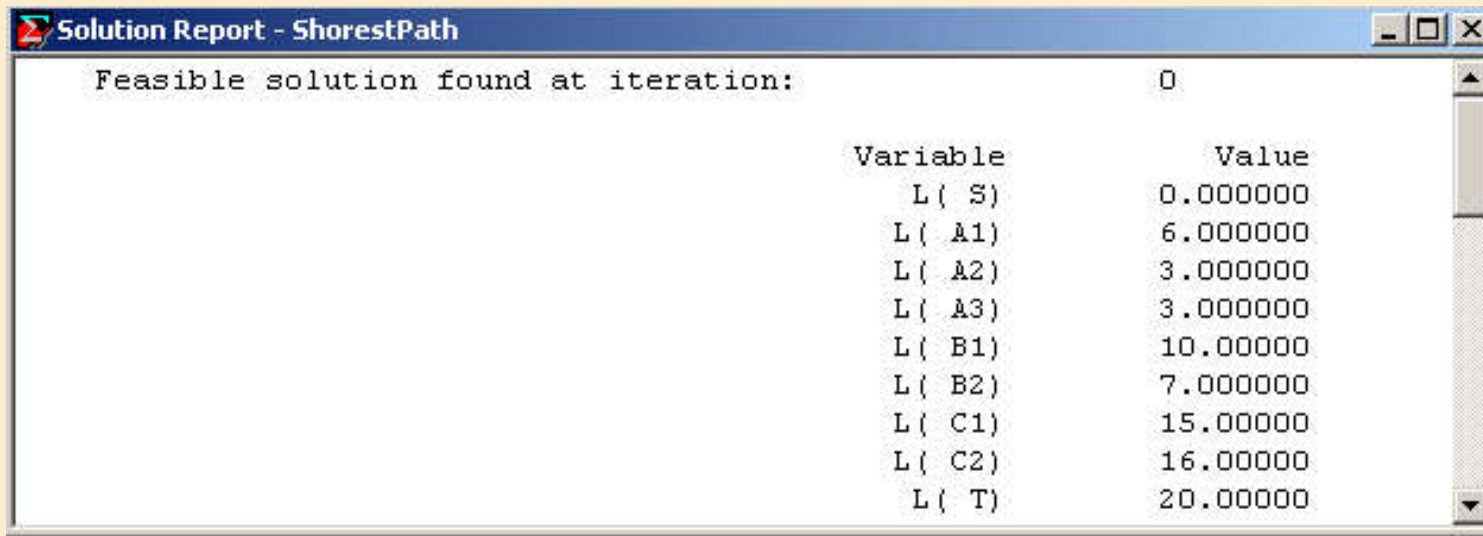
在@for循环中的过滤条件里用了一个函数“@index”，其作用是返回一个元素在集合中的索引值，这里@index(S)=1(即元素S在集合中的索引值为1)，所以逻辑关系式“ $I \#GT \# @index(S)$ ”可以可以等价地写成“ $I \#GT \# 1$ ”。这里@index(S)实际上还是@index(CITIES,S)的简写，即返回S在集合CITIES中的索引值。

因为L(S)=0

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；

## 本例的LINGO求解结果



Feasible solution found at iteration:		0
Variable	Value	
L( S)	0.000000	
L( A1)	6.000000	
L( A2)	3.000000	
L( A3)	3.000000	
L( B1)	10.00000	
L( B2)	7.000000	
L( C1)	15.00000	
L( C2)	16.00000	
L( T)	20.00000	

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

本例中定义稀疏集合ROADS的方法是通过枚举给出, 有时如果元素比较多, 用起来不方便。另一种定义稀疏集合的方法是“元素过滤”法, 能够从笛卡儿积中系统地过滤下来一些真正的元素。



例 某班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。这8名同学两两之间组队的效率如下表所示(由于对称性, 只列出了严格上三角部分), 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

## 分析

这是一个匹配 (MATCHING) 问题。把上表的效率矩阵记为BENEFIT(由于对称性, 这个矩阵只有严格上三角部分共28个数取非零值)。

用MATCH (Si, Sj) =1表示同学Si, Sj组成一队, 而MATCH (Si, Sj) =0表示Si, Sj不组队。由于对称性, 只需考虑*i*<*j*共28个0-1变量(而不是全部32个变量)。

显然, 目标函数正好是BENEFIT(Si,Sj)\*MATCH(Si,Sj)对*i,j*求和。

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: 只要属性MATCH的某个下标为*i*就加起来, 此和应该等于1。

由上面的分析, 因此, 完整的数学模型如下(显然, 这是一个0-1线性规划):

$$\text{Min} \quad \sum_{I < J} \{BENEFIT(I, J) * MATCH(I, J)\} \quad (1)$$

$$\text{s.t.} \quad \sum_{J=I \text{ 或 } K=I} MATCH(J, K) = 1, \quad I = 1, 2, 3, 4, \quad (2)$$

$$MATCH(J, K) \in \{0, 1\}. \quad (3)$$

## 问题的LINGO求解

“S1..S8”等价于写成“S1 S2 S3 S4 S5 S6 S7 S8”，它没有相关的属性列表，只用于表示是一个下标集合

LINGO Model - MATCH

```

MODEL:
SETS:
    STUDENTS /S1..S8/;
    PAIRS( STUDENTS, STUDENTS) | &2 #GT# &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

```

在派生集合PAIRS定义中增加了过滤条件

“&2#GT#&1”，意思是第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”表示)。PAIRS中的元素对应于上表中的严格上三角部分的二维下标(共28个元素)。

BENEFIT和MATCH是PAIRS的属性。

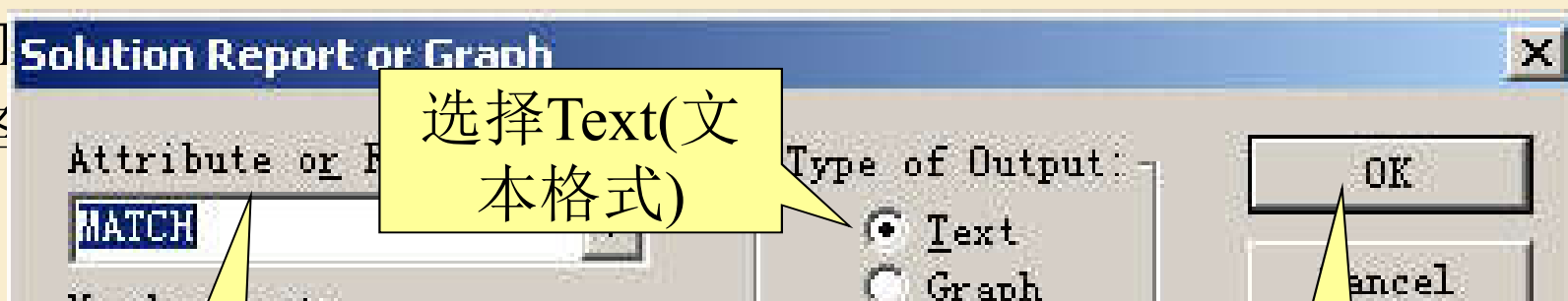
注意数据段对BENEFIT的赋值方式，“LINGO按照列的顺序对属性变量的元素进行赋值。在约束部分，过滤条件“J#EQ#I#OR#K#EQ#I”是由逻辑运算符“#OR#（或者）”连接的一个复合的逻辑关系式，连接由“#EQ#（等于）”表示的两个逻辑关系。由于“#OR#”的运算级别低于“#EQ#”，所以这个逻辑式中没有必要使用括号指定运算次序。

# LINGO求解结果

“LINGO|SOLVE”运行这个程序，可以得到全局最优值为30

MA

选择



**Solution Report - MATCH**

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

变量)

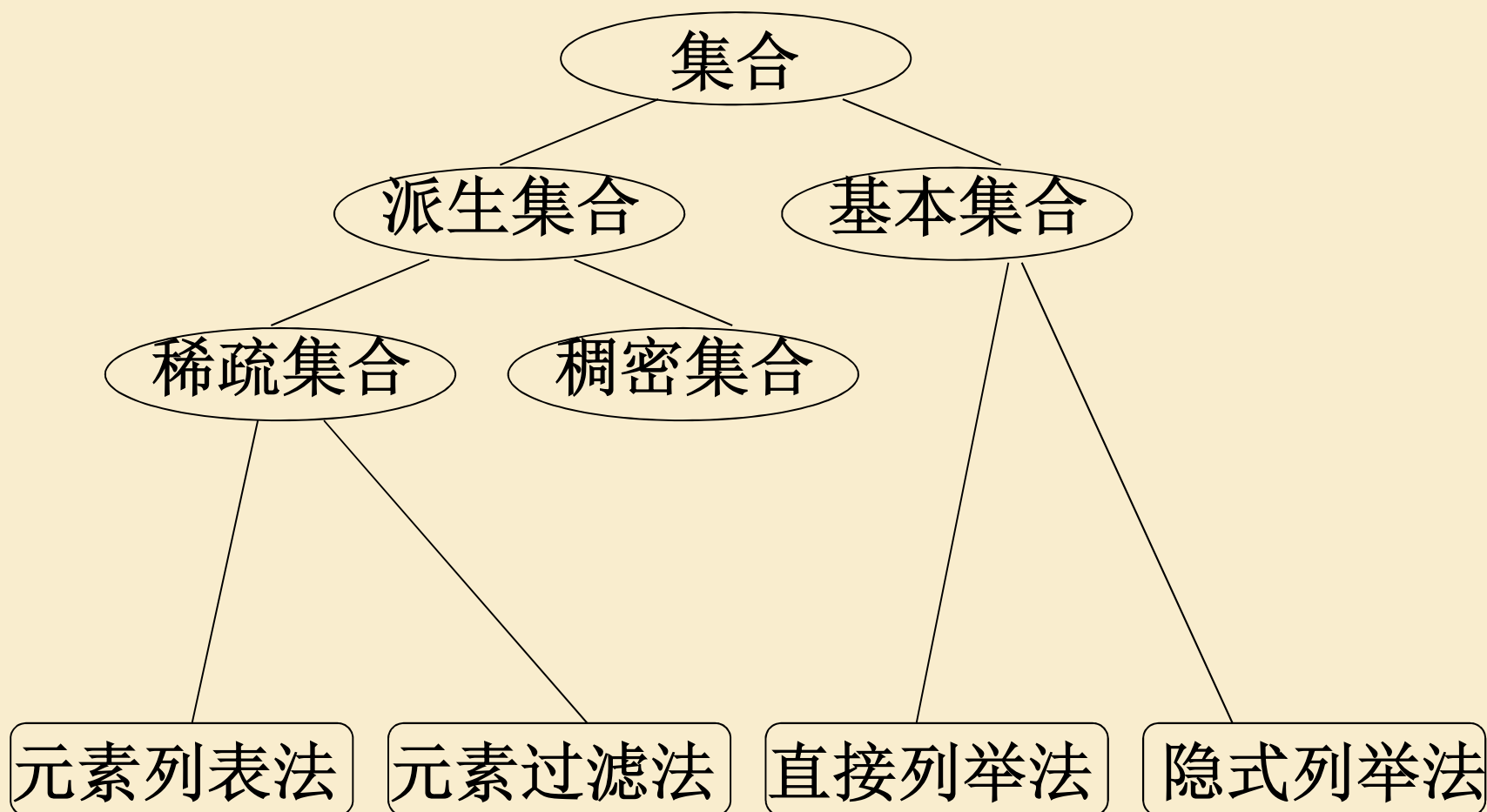
选择Nonzeros  
Only(只显示非零值)

学生最佳的组队方式是(1,8),(2,4),(3,7),(5,6).

点击“OK”按钮，得到关于  
最优解的非零分量的报告

# 集合的使用小结

## 集合的不同类型及其关系



## 基本集合的定义语法

基本集合的定义格式为(方括号 “[ ]” 中的内容是可选项, 可以没有):

```
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。元素列表可以采用显式列举法(即直接将所有元素全部列出,元素之间用逗号或空格分开),也可以采用隐式列举法。隐式列举法可以有几种不同格式,

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

## 基本集合的

元素列表和  
当属性列表  
序中作为一个

LINGO Model - LINGO1

```
MODEL:
SETS:
    QUARTERS/1,2,3,4/:DEM,RP,OP,INV;
ENDSETS
MIN=@SUM(QUARTERS:400*RP+450*OP+20*INV);
@FOR(QUARTERS(I):RP(I)<40);
```

帆船  
的源

生成

列表;

点;

LINGO Model - MATCH

```
MODEL:
SETS:
    STUDENTS /S1..S8/;
    PAIRS( STUDENTS, STUDENTS) | &2 #GT# &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
[objective] 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);
@FOR(PAIRS( I, J): @BIN( MATCH( I, J)));
END
```

匹配  
的源  
程序  
问题



## 派生集合的定义语法

派生集合的定义格式为(方括号 “[ ]” 中的内容是可选项, 可以没有):

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

与基本集合的定义相比较多了一个 `parent_set_list` (父集合列表)。

父集合列表中的集合(如 `set1`, `set2`, ..., 等)称为派生集合 `setname` 的父集合, 它们本身也可以是派生集合。

当元素列表(`member_list`)不在集合定义中出现时, 还可以在程序的数据段以赋值语句的方式给出元素列表;

若在程序的数据段也不以赋值语句的方式给出元素列表, 则认为定义的是稠密集合, 即父集合中所有元素的有序组合(笛卡儿积)都是 `setname` 的元素。

当元素列表在集合定义中出现时, 又有“元素列表法”(直接列出元素)和“元素过滤法”(利用过滤条件)两种不同方式。



### 3.运算符和函数

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题

# 运算符及其优先级

## 算术运算符

加、减、乘、除、乘方等数学运算(即数与数之间的运算，运算结果也是数)。

LINGO中的算术运算符有以下5种：

+（加法），

—（减法或负号），

\*（乘法），

/（除法），

^(求幂)。

## 逻辑运算符

运算结果只有“真”(TRUE)和“假”(FALSE)两个值(称为“逻辑值”), LINGO中用数字1代表TRUE, 其他值(典型的值是0)都是FALSE。

在LINGO中, 逻辑运算(表达式)通常作为过滤条件使用, 逻辑运算符有9种, 可以分成两类:

#AND#(与), #OR#(或), #NOT#(非): 逻辑值之间的运算, 它们操作的对象本身已经是逻辑值或逻辑表达式, 计算结果也是逻辑值。

#EQ#(等于), #NE#(不等于), #GT#(大于), #GE#(大于等于), #LT#(小于), #LE#(小于等于): 是“数与数之间”的比较, 也就是它们操作的对象本身必须是两个数, 计算得到的结果是逻辑值。

## 关系运算符

表示是“数与数之间”的大小关系，在LINGO中用来表示优化模型的约束条件。LINGO中关系运算符有3种：

<(即 $\leq$ ，小于等于)，=(等于)，>(即 $\geq$ ，大于等于)

(在优化模型中约束一般没有严格小于、严格大于关系)

## 运算符的优先级

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

## 基本的数学函数

在LINGO中建立优化模型时可以引用大量的内部函数,这些函数以”@”打头。LINGO中包括相当丰富的数学函数,大部分可以通过点击edit|paste function找到,这些函数的用法非常简单,下面一一列出。

**@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的单位是弧度)。

**@EXP(X)**: 指数函数, 返回  $e^x$  的值(其中 $e=2.718281\dots$ )。

**@FLOOR(X)**: 取整函数, 返回X的整数部分(向最靠近0的方向取整)。

**@LGM(X)**: 返回X的伽玛(gamma)函数的自然对数值(当X为整数时 $LGM(X) = \text{LOG}(X-1)!$ ; 当X不为整数时, 采用线性插值得到结果)。

**@LOG(X)**: 自然对数函数, 返回X的自然对数值。

## 基本的数学函数

@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X和Y应该是整数。

@POW(X,Y): 指数函数, 返回 $X^Y$ 的值。

@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)。

@SIN(X): 正弦函数, 返回X的正弦值 (X的单位是弧度)。

@SMAX(list): 最大值函数, 返回一系列数(list)的最大值。

@SMIN(list): 最小值函数, 返回一系列数(list)的最小值。

@SQR(X): 平方函数, 返回X的平方 (即 $X*X$ ) 的值。

@SQRT(X): 开平方函数, 返回X的正的平方根的值。

@TAN(X): 正切函数, 返回X的正切值 (X的单位是弧度)。

## 集合循环函数

集合上的元素(下标)进行循环操作的函数,一般用法如下:

`@function(setname [ ( set_index_list)[ | condition]] : expression_list);`

其中:

`function` 集合函数名, FOR、MAX、MIN、PROD、SUM之一;

`Setname` 集合名;

`set_index_list` 集合索引列表(不需使用索引时可以省略);

`Condition` 用逻辑表达式描述的过滤条件(通常含有索引,无条件时可以省略);

`expression_list` 一个表达式(对@FOR函数,可以是一组表达式)。

## 集合循环函数

五个集合函数名的含义:

**@FOR**(集合元素的循环函数): 对集合setname的每个元素独立地生成表达式, 表达式由expression\_list描述 (通常是优化问题的约束)。

**@MAX** (集合属性的最大值函数): 返回集合setname上的表达式的最大值。

**@MIN** (集合属性的最小值函数): 返回集合setname上的表达式的最小值。

**@PROD** (集合属性的乘积函数): 返回集合setname上的表达式的积。

**@SUM**(集合属性的求和函数): 返回集合setname上的表达式的和。



## 集合操作函数

`@INDEX( [set_name,] primitive_set_element)`

给出元素`primitive_set_element`在集合`set_name`中的索引值(即按定义集合时元素出现顺序的位置编号)。省略`set_name`, LINGO按模型中定义的集合顺序找到第一个含有该元素的集合,并返回索引值。如果没有找到该元素,则出错。

注: `Set_name`的索引值是正整数且只能位于1和元素个数之间。  
例:定义一个女孩姓名集合(GIRLS)和男孩姓名集合(BOYS):

SETS:

GIRLS /DEBBIE, SUE, ALICE/;

BOYS /BOB, JOE, SUE, FRED/;

ENDSETS

都有SUE, GIRLS在BOYS前定义,调用`@INDEX(SUE)`将返2,相当于`@INDEX(GIRLS,SUE)`。要找男孩中名为SUE的小孩的索引,应该使用`@INDEX(BOYS,SUE)`, 返3。

## 集合操作函数

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

判断一个集合中是否含有某个索引值。如果集合set\_name中包含由索引primitive\_index\_1 [, primitive\_index\_2 ...]所对应元素，则返回1(逻辑值“真”)，否则返回0(逻辑值“假”)。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:定义一个集合STUDENTS(基本集合)，派生出集合PASSED和FAILED，定义：

SETS:

STUDENTS / ZHAO, QIAN, SUN, LI/;;

PASSED( STUDENTS) /QIAN, SUN/;;

FAILED( STUDENTS) | #NOT# @IN( PASSED, &1);;

ENDSETS



## 集合操作函数

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

如果集合C是由集合A, B派生的, 例如:

SETS:

A / 1..3/;;

B / X Y Z/;;

C( A, B) / 1,X 1,Z 2,Y 3,X/;;

ENDSETS

判断C中是否包含元素 (2, Y), 则可以利用以下语句:

`X = @IN( C, @INDEX( A, 2), @INDEX( B, Y));`

对本例, 结果是X=1 (真)。

注: X既是集合B的元素, 又对X赋值1, 在LINGO中这种表达是允许的, 因为前者是集合的元素, 后者是变量, 逻辑上没有关系(除了同名外), 所以不会出现混淆。

## 集合操作函数

**@WRAP(I,N)** 此函数对 $N < 1$ 无定义

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。即

$$@WRAP(I,N) = @MOD(I, N)。$$

但当 $@MOD(I,N)=0$ 时 $@WRAP(I,N)=N$ 。

此函数可以用来防止集合的索引值越界。

用户在编写LINGO程序时，应注意避免LINGO模型求解时出现集合的索引值越界的错误。

**@SIZE (set\_name)**

返回数据集set\_name中包含元素的个数。

## 变量定界函数

对变量的取值范围附加限制，共有以下四种：

- **@BND(L, X, U)**：限制 $L \leq X \leq U$ 。注意LINGO中没有与LINDO命令SLB、SUB类似的函数@SLB和@SUB
- **@BIN(X)**：限制X为0或1。注意LINDO中的命令是INT，但LINGO中这个函数的名字却不是@INT(X)
- **@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

## 财务会计函数

用于计算净现值，包括以下两个函数：

- **@FPA(I,N)** 返回如下情形下总的净现值：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。根据复利的计算公式，很容易知道

$$@FPA(I,N) = \sum_{n=1}^N 1/(1+I)^n = (1 - (\frac{1}{1+I})^N) / I \quad。$$

- **@FPL(I,N)** 返回如下情形下总的净现值：单位时段利率为I，第N个时段支付单位费用。根据复利的计算公式，很容易知道

$$@FPL(I,N) = (\frac{1}{1+I})^N \quad。$$

## 概率相关函数

- **@PSN(X)**: 标准正态分布函数, 即返回标准正态分布的分布函数在X点的取值。
- **@PSL(X)**: 标准正态线性损失函数, 即返回  $\text{MAX}(0, Z-X)$  的期望值, 其中  $Z$  为标准正态随机变量。
- **@PPS(A,X)**: **Poisson**分布函数, 即返回均值为A的**Poisson**分布的分布函数在X点的取值 (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  $\text{MAX}(0, Z-X)$  的期望值, 其中  $Z$  为均值为A的**Poisson**随机变量。
- **@PBN(P,N,X)**: 二项分布函数, 即返回参数为 (N, P) 的二项分布的分布函数在X点的取值 (当N和 (或) X不是整数时, 采用线性插值进行计算) 。



## 概率相关函数

- **@PHG(POP,G,N,X)**: 超几何 (Hypergeometric) 分布的分布函数。也就是说, 返回如下概率: 当总共有POP个球, 其中G个是白球时, 那么随机地从中取出N个球, 白球不超过X个的概率。当POP, G, N和 (或) X不是整数时, 采用线性插值进行计算。
- **@PEL(A,X)**: 当到达负荷 (强度) 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 (强度) 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, 并行服务器数量为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)。

## 概率相关函数

- **@PFD(N,D,X)**: 自由度为N和D的F分布的分布函数在X点的取值。
- **@PCX(N,X)**: 自由度为N的分布的分布函数在X点的取值。
- **@PTD(N,X)**: 自由度为N的t分布的分布函数在X点的取值。
- **@QRAND(SEED)**: 返回0与1之间的多个拟均匀随机数（SEED为种子，缺省时取当前计算机时间为种子）。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，需要详细了解“拟均匀随机数（quasi-random uniform numbers）”请进一步参阅LINGO的使用手册。
- **@RAND(SEED)**: 返回0与1之间的一个伪均匀随机数（SEED为种子）。

## 文件输入输出函数

- **@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。
- **@ODBC** 提供LINGO与ODBC（Open Data Base Connection，开放式数据库连接）的接口。
- **@OLE(spreadsheet\_file[,range\_name\_list])**提供LINGO与OLE(Object Linking and Embedding)接口。

Spreadsheet\_file是电子表格文件的名称，包括扩展名；

Range\_name\_list是文件中包含数据的单元范围。

- **@POINTER(N)**在Windows下使用LINGO的动态连接库DLL，直接从共享的内存中传送数据。
- **@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）。filename中可以带有文件路径，没有指定路径时表示在当前目录,如果这个文件已经存在，将会被覆盖。

## 结果报告函数

- **@ITERS()** 只能在程序的数据段使用，调用时不需要任何参数，返回LINGO求解器计算所使用的总迭代次数。例如：

**@TEXT() = @ITERS();**

将迭代次数显示在屏幕上。

- **@NEWLINE(n)** 在输出设备上输出n个新行。
- **@STRLEN(string)** 返回字符串“string”的长度，如  
**@STRLEN(123)**返回值为3。

## 结果报告函数

- **@NAME(var\_or\_row\_refernce)** 返回变量名或行名。

例：

**SETS:**

**WH/WH1..WH3/; ! WH表示仓库的集合;**

**C/C1..C4/; ! C表示顾客的集合;**

**ROAD (WH,C) : X; !ROAD表示仓库到顾客的道路集合;**

**! X表示某个仓库对某个顾客供货数;**

**ENDSETS**

**DATA:**

**@TEXT() = @WRITEFOR( ROAD( I, J) |**

**X( I, J) #GT# 0: @NAME( X), ' ', X, @NEWLINE( 1));**

**ENDDATA**

## 结果报告函数

- **@NAME(var\_or\_row\_refernce)** 返回变量名或行名。

输出结果示意如下(这里没有详细交待属性X当前的取值):

X( WH1, C1)     2

X( WH1, C2)     17

X( WH1, C3)     1

X( WH2, C1)     13

X( WH2, C4)     12

X( WH3, C3)     21

注: “变量”是指“数组元素” X( WH1, C1)、X( WH2, C4)等, 即属性加上相应的下标(集合元素)。

同理, 约束名也是指模型展开后的约束名(用LINGO|  
**Generate**命令可以看到约束展开后的情况), 即也应该是带有相应的下标(集合元素)的。

## 结果报告函数

- **@WRITE(obj1[, ..., objn])** 只能在数据段中使用，输出一系列结果(obj1, ..., objn)，其中obj1, ..., objn等可以是变量(但不能只是属性)，也可以是字符串(放在单引号中的为字符串)或换行(@NEWLINE)等。

结果可以输出到一个文件，或电子表格（如EXCEL），或数据库，这取决于@WRITE所在的输出语句中左边的定位函数。

例如：

**DATA:**

**@TEXT() = @WRITE('A is', A, ', B is', B, ', A/B is', A/B);**

**ENDDATA**

其中A，B是该模型中的变量，在屏幕上输出A，B以及A/B的值(增加了一些字符串，使结果读起来更方便)。假设计算结束时A=10，B=5,则输出为：

**A is 10, B is 5, A/B is 2**



## 结果报告函数

- **@WRITEFOR( setname[ ( set\_index\_list) [ | condition]]: obj1[,..., objn])** 函数@WRITE在循环情况下的推广，输出集合上定义的属性对应的多个变量的取值。

例：（这里WH，C，X含义同上）：

**DATA:**

**@TEXT() = @WRITEFOR( ROAD( I, J) | X( I, J) #GT# 0:**

**‘从仓库 ’, WH( I),**

**‘ 到顾客 ', C( J), '供货 ', X( I, J), '件 ', @NEWLINE( 1));**

**ENDDATA**

对应的输出效果示意如下：

从仓库 WH1 到顾客 C1 供货 2件

.....

从仓库 WH3 到顾客 C3 供货21件

注：这里省略了部分输出结果

## 结果报告函数

**\* :** 在**@WRITE**和**@WRITEFOR**函数中, 可以使用符号 “\*” 表示将一个字符串重复多次, 用法是将 “\*” 放在一个正整数**n**和这个字符串之前, 表示将这个字符串重复**n**次。

**DATA:**

**LEAD = 3;**

**@TEXT() = '上班人数图示';**

**@TEXT() = @WRITEFOR( DAY( D): LEAD\*' ',  
DAY( D), ' ', ON\_DUTY( D), ' ', ON\_DUTY( D)\*'+',**

**@NEWLINE(1) );**

**ENDDATA**

程序执行的效果示意如下(**DAY**表示一周, **ON\_DUTY**表示每天上班人数):

上班人数图示

**MON 20** ++++++

.....

**SUN 13** ++++++

注: 这里省略了部分输出结果

## 结果报告函数

- **@FORMAT( value, format\_descriptor)**

在@WRITE和@WRITEFOR函数中，@FORMAT对数值设定输出格式。value表示输出的值，format\_descriptor(格式描述符)表示输出格式。格式描述符的含义与C语言中的格式描述类似,如“12.2f”表示十进制数,总共12位，其中有2位小数。

注：使用@FORMAT函数将把数值转换成字符串,所以输出的实际上是字符串,这对于向数据库,电子表中输出不一定合适。

- **@DUAL( variable\_or\_row\_name)**

@DUAL(variable)返回解答中variable的判别数(reduced cost);

@DUAL(row)将返回约束行row的对偶(影子)价格(dual prices)。

例：

DATA:

```
@TEXT() = @WRITEFOR( SET1( I): X( I), @DUAL( X( I), @NEWLINE( 1));
```

```
ENDDATA
```

## 结果报告函数

- **@RANGED( variable\_or\_row\_name)**

为了保持最优基不变，目标函数中变量的系数或约束行的右端项允许减少的量（参见第2章2.2节敏感性分析中的**allowable decrease**）。

- **@RANGEU( variable\_or\_row\_name)**

为了保持最优基不变，目标函数中变量的系数或约束行的右端项允许增加的量（参见第2章2.2节敏感性分析中的**allowable increase**）。

## 结果报告函数

- **@STATUS()** 返回LINGO求解模型结束后的最后状态：
  - 0 **Global Optimum** （全局最优）
  - 1 **Infeasible** （不可行）
  - 2 **Unbounded** （无界）
  - 3 **Undetermined** （不确定）
  - 4 **Interrupted** （用户人为终止了程序的运行）
  - 5 **Infeasible or Unbounded** （通常需要关闭“预处理”选项后重新求解模型，以确定究竟是不可行还是无界）
  - 6 **Local Optimum** （局部最优）
  - 7 **Locally Infeasible** （局部不可行）
  - 8 **Cutoff** （目标函数达到了指定的误差水平）
  - 9 **Numeric Error** （约束中遇到了无定义的数学操作）

## 其他函数

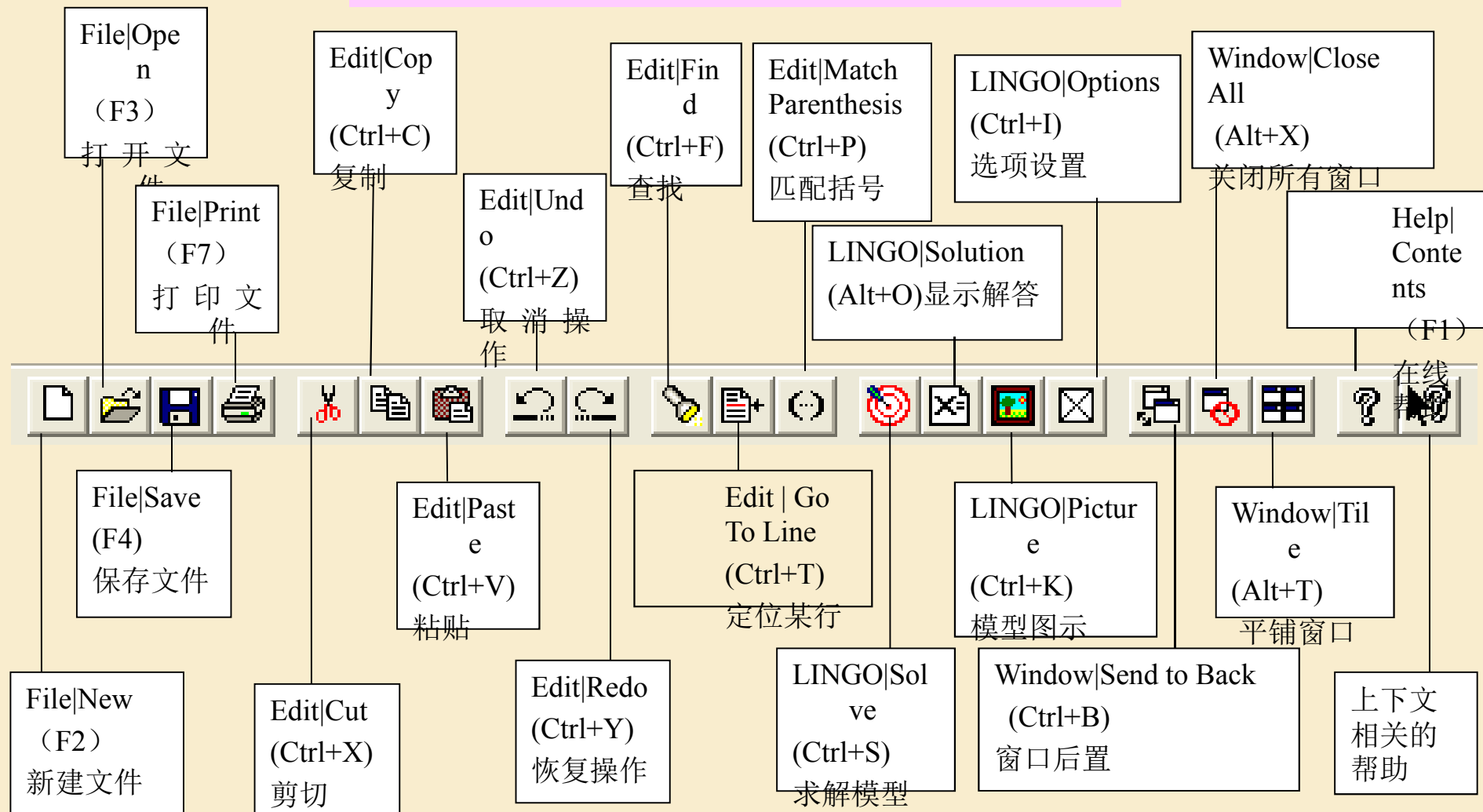
- **@IF(logical\_condition, true\_result, false\_result)** 当逻辑表达式logical\_condition的结果为真时，返回true\_result，否则返回false\_result。
- **@WARN('text', logical\_condition)**  
如果逻辑表达式“logical\_condition”的结果为真，显示‘text’信息。
- **@USER(user\_determined\_arguments)**  
允许用户自己编写的函数（DLL或OBJ文件），可能应当用C或FORTRAN等其他语言编写并编译。

## 4. LINGO的主要菜单命令

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题



# 工具栏



## 文件(File)主菜单

File|Export File...

优化模型输出到文件，有两个子菜单，分别表示两种输出格式（都是文本文件）：

MPS Format（MPS格式）：是IBM公司制定的一种数学规划文件格式。

MPI Format（MPI格式）：是LINDO公司制定的一种数学规划文件格式。

File|User Database Info

弹出对话框，用户输入用户使用数据库时需要验证的用户名（User ID）和密码（Password），这些信息在使用@ODBC()函数访问数据库时要用到。

## 编辑（Edit）主菜单

### Edit|Paste 和Edit|Paste Special ...

将WINDOWS剪贴板中的内容粘贴到当前光标处。

- “Edit|Paste（粘贴命令）” 仅用于剪贴板中的内容是文本的情形。
- “Edit|Paste Special ...（特殊粘贴命令）” 可以用于剪贴板中的内容不是文本的情形，可以插入其它应用程序中生成的对象（Object）或对象的链接（Link）。

例如，LINGO模型中可能会在数据段用到从其它应用程序中生成的数据对象（如EXCEL电子表格数据），这时用“Edit|Paste Special ...”是很方便的。

## 编辑（Edit）主菜单

### Edit|Match Parenthesis

用于匹配模型中的括号：

- 如果当前没有选定括号，则把光标移动到离当前光标最近的一个括号并选中这个括号。
- 当选定一个括号后，则把光标移动到与这个括号相匹配的括号并选中这个括号。

### Edit|Paste Function

还有下一级子菜单和下下一级子菜单，用于按函数类型选择LINGO的某个函数，粘贴到当前光标处。

# 编辑 (Edit) 主菜单

## Edit|Select Font

弹出对话框，控制显示字体、字形、大小、颜色、效果等。

注：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。

此外，如果“按语法显示色彩”选项是有效的(参见“LINGO|Options”)，在模型窗口中将不能通过“Edit|Select Font”菜单命令控制文本的颜色。



## 编辑（Edit）主菜单

### Edit|Insert New Object

插入其它应用程序中生成的整个对象或对象的链接。前面介绍过的“Edit|Paste Special ...”与此类似，但“Paste Special”命令一般用于粘贴某个外部对象的一部分，而这里的命令是插入整个对象或对象的链接。

### Edit|Links

在模型窗口中选择一个外部对象的链接，然后选择“Edit|Links（链接）”命令，则弹出一个对话框，可以修改这个外部对象的链接属性。

# 编辑（Edit）主菜单

## Edit|Object Properties

在模型窗口中选择一个链接或嵌入对象（OLE），然后选择

“Edit|Object Properties（对象属性）”命令，则弹出一个对话框，可以修改这个对象的属性。主要包括以下属性：

- display of the object: 对象的显示;
- the object's source: 对象的源;
- type of update (automatic or manual): 修改方式（自动或人工修改）;
- opening a link to the object: 打开对象的一个链接;
- updating the object: 修改对象;
- breaking the link to the object: 断开对象的链接。



# LINGO系统(LINGO)主菜单

## LINGO|LOOK（模型显示）

模型窗口下才能使用，按照LINGO模型的输入形式以文本方式显示，显示时对输入的所有行(包括说明语句)按顺序编号。将弹出一个对话框，在对话框中选择“ALL”将对所有行进行显示，也可以选择“Selected”输入起始行，这时只显示相应行的内容。

## LINGO|Generate和LINGO|Picture

都是在模型窗口下才能使用，它们的功能是按照LINGO模型的完整形式（例如将属性按下标（集合的每个元素）展开）显示目标函数和约束（只有非零项会显示出来）。

## LINGO系统(LINGO)之基础

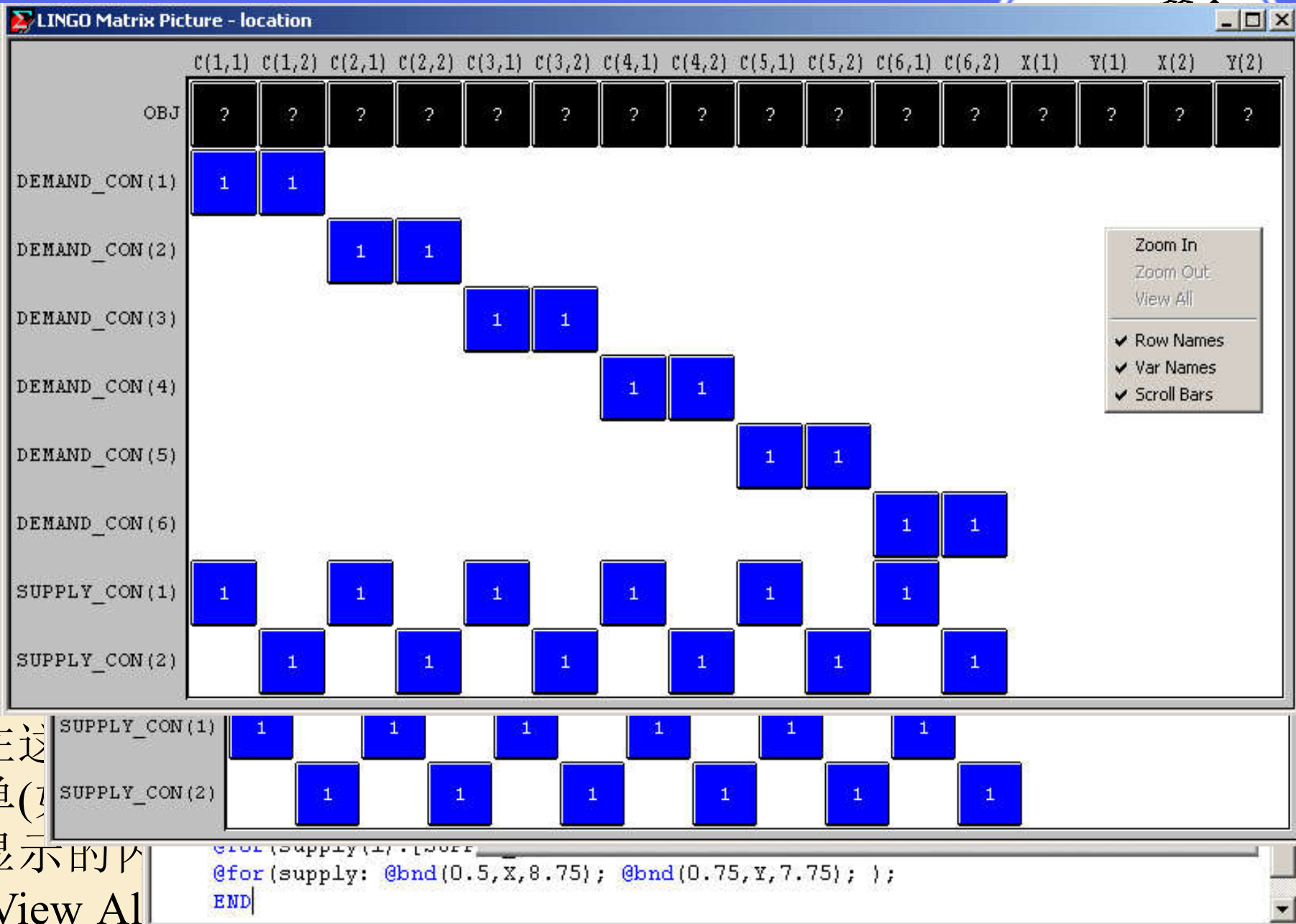
## LINGO

结果以  
果的要  
“Don  
屏幕上  
选择适  
例如,  
LINGO  
注: 在  
量项,  
示。

```

MODEL:
TITLE Location Problem;
[OBJ] MIN= ( C_1_1 * ( ( X_1 - 1.25 ) ^ 2 + ( Y_1 - 1.25 ) ^
2 ) ^ ( 1 / 2 ) + C_1_2 * ( ( X_2 - 1.25 ) ^ 2 + ( Y_2 -
1.25 ) ^ 2 ) ^ ( 1 / 2 ) + C_2_1 * ( ( X_1 - 8.75 ) ^ 2 +
( Y_1 - 0.75 ) ^ 2 ) ^ ( 1 / 2 ) + C_2_2 * ( ( X_2 - 8.75
) ^ 2 + ( Y_2 - 0.75 ) ^ 2 ) ^ ( 1 / 2 ) + C_3_1 * ( (
X_1 - 0.5 ) ^ 2 + ( Y_1 - 4.75 ) ^ 2 ) ^ ( 1 / 2 ) +
C_3_2 * ( ( X_2 - 0.5 ) ^ 2 + ( Y_2 - 4.75 ) ^ 2 ) ^ ( 1
/ 2 ) + C_4_1 * ( ( X_1 - 5.75 ) ^ 2 + ( Y_1 - 5 ) ^ 2 )
^ ( 1 / 2 ) + C_4_2 * ( ( X_2 - 5.75 ) ^ 2 + ( Y_2 - 5 )
^ 2 ) ^ ( 1 / 2 ) + C_5_1 * ( ( X_1 - 3 ) ^ 2 + ( Y_1 -
6.5 ) ^ 2 ) ^ ( 1 / 2 ) + C_5_2 * ( ( X_2 - 3 ) ^ 2 + (
Y_2 - 6.5 ) ^ 2 ) ^ ( 1 / 2 ) + C_6_1 * ( ( X_1 - 7.25 )
^ 2 + ( Y_1 - 7.75 ) ^ 2 ) ^ ( 1 / 2 ) + C_6_2 * ( ( X_2
- 7.25 ) ^ 2 + ( Y_2 - 7.75 ) ^ 2 ) ^ ( 1 / 2 ) ) ;
[DEMAND_CON_1] C_1_1 + C_1_2 = 3 ;
[DEMAND_CON_2] C_2_1 + C_2_2 = 5 ;
[DEMAND_CON_3] C_3_1 + C_3_2 = 4 ;
[DEMAND_CON_4] C_4_1 + C_4_2 = 7 ;
[DEMAND_CON_5] C_5_1 + C_5_2 = 6 ;
[DEMAND_CON_6] C_6_1 + C_6_2 = 11 ;
[SUPPLY_CON_1] C_1_1 + C_2_1 + C_3_1 + C_4_1 + C_5_1 + C_6_1 <= 20 ;
[SUPPLY_CON_2] C_1_2 + C_2_2 + C_3_2 + C_4_2 + C_5_2 + C_6_2 <= 20 ;
@BND( 0.5, X_1, 8.75) ; @BND( 0.75, Y_1, 7.75) ; @BND( 0.5,
X_2, 8.75) ; @BND( 0.75, Y_2, 7.75) ;
END

```



I  
结  
例  
该  
该  
正

在这  
单(1  
显示的  
(View All

(Var Names)、滚动条(Scroll Bars)。



# LINGO系统(LINGO)主菜单

## LINGO|Options命令

打开一个含有7个选项卡的窗口,通过它修改LINGO系统的各种控制参数和选项。

当前设置变为默认设置,下次启动LINGO时这些设置仍然有效

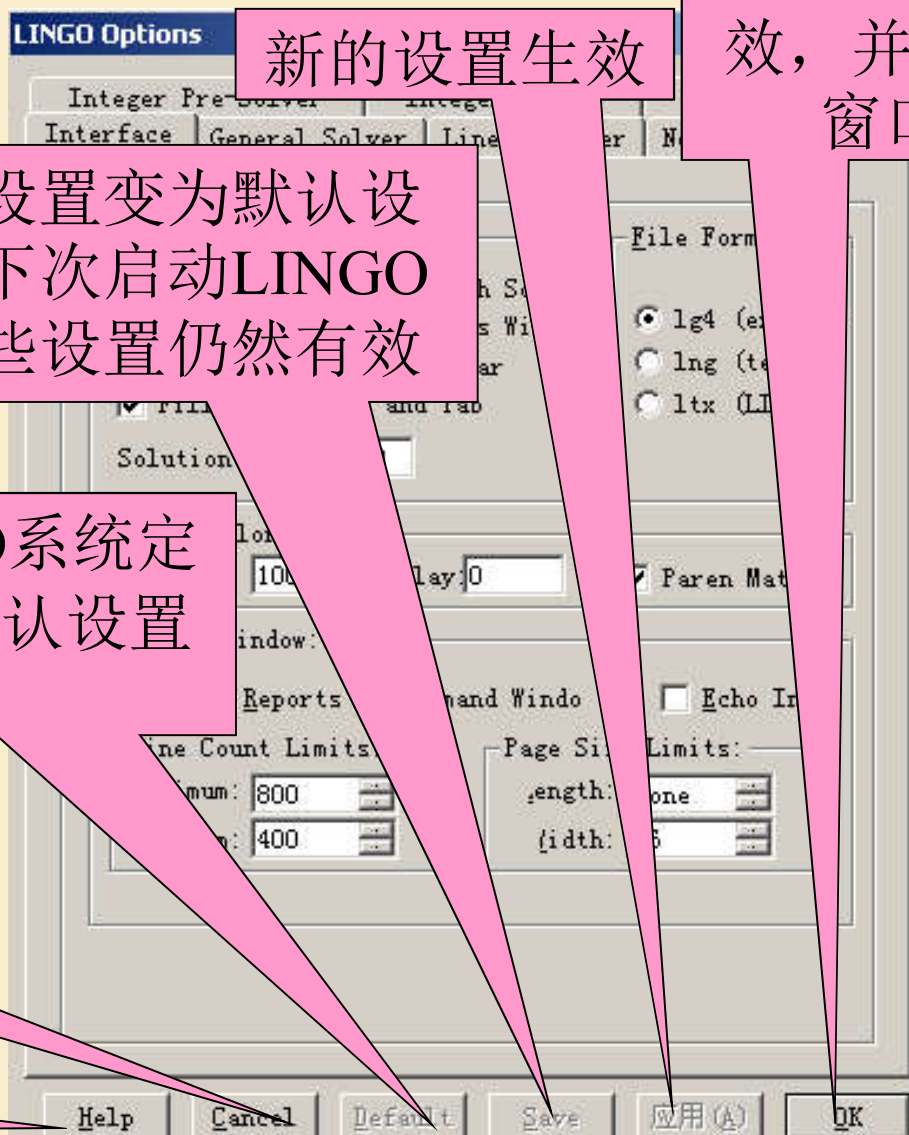
恢复LINGO系统定义的原始默认设置(缺省设置)

废弃本次操作,退出对话框

显示本对话框的帮助信息

新的设置生效

新的设置生效,并关闭窗口



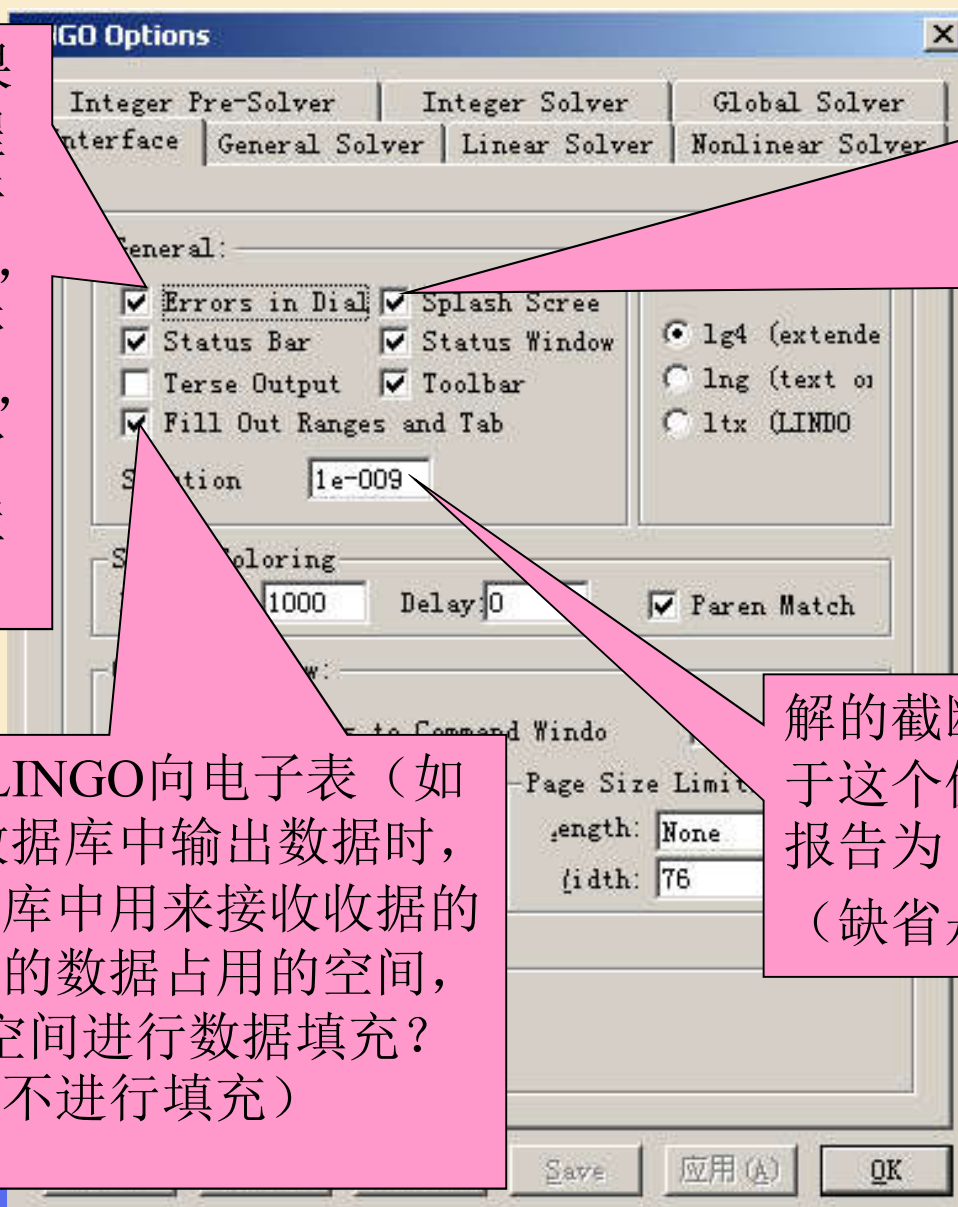
# LINGO|Options|Interface (界面) 选项卡

错误信息对话框:如果选择该选项, 求解程序遇到错误时将打开一个对话框显示错误, 关闭该对话框后程序才会继续执行; 否则, 错误信息将在报告窗口显示, 程序仍会继续执行

弹出屏幕:如果选择该选项, 则LINGO每次启动时会在屏幕上弹出一个对话框, 显示LINGO的版本和版权信息; 否则不弹出

填充数据库表:当LINGO向电子表(如EXCEL文件)或数据库中输出数据时, 如果电子表或数据库中用来接收收据的空间大于实际输出的数据占用的空间, 是否对多余的表空间进行数据填充? (缺省值为不进行填充)

解的截断:小于等于这个值的解将报告为“0”  
(缺省是 $10^{-9}$ )





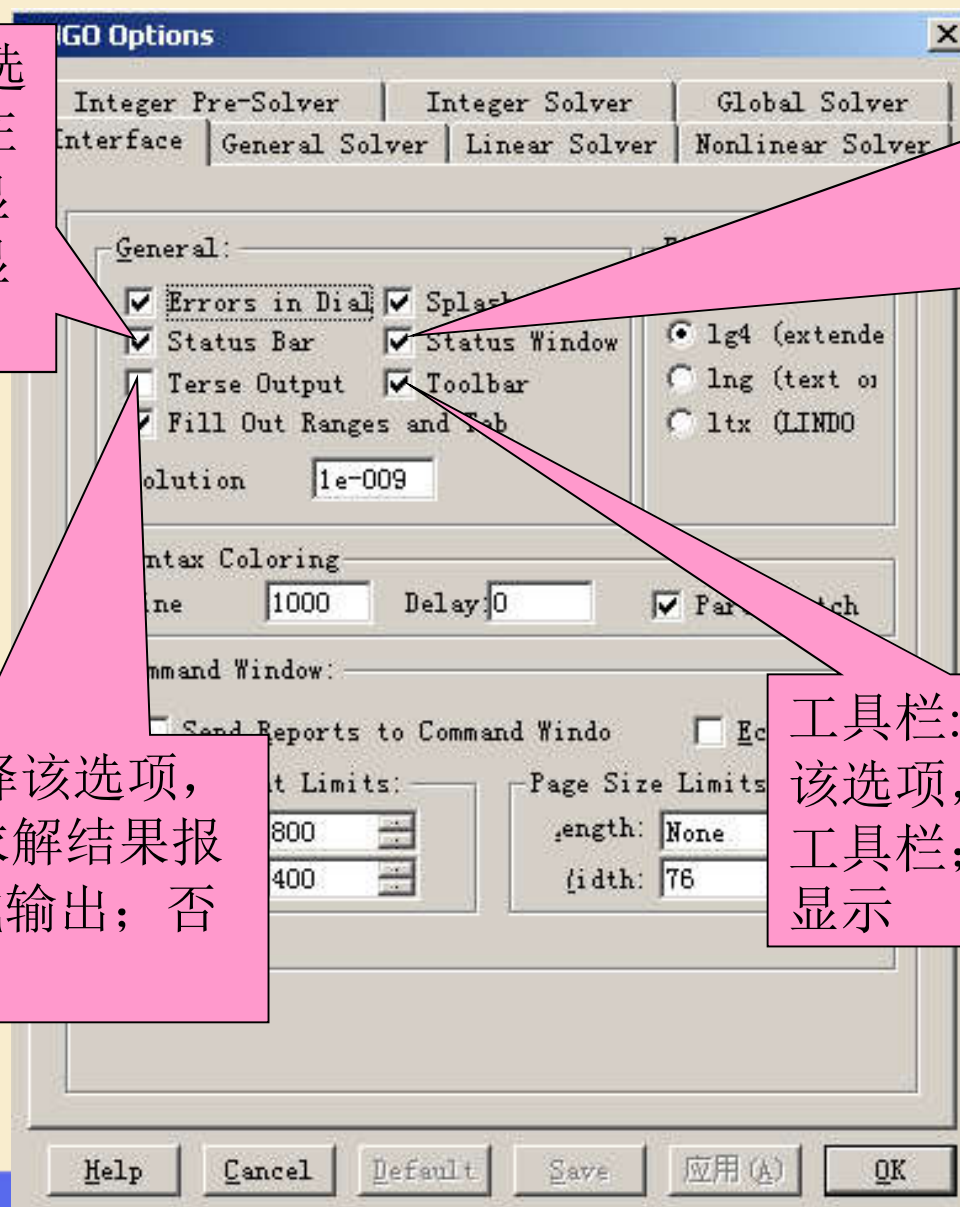
## LINGO|Options|Interface（界面）选项卡

状态栏:如果选择该选项,则LINGO系统在主窗口最下面一行显示状态栏;否则不显示

状态窗口:如果选择该选项,则LINGO系统每次运行LINGO|Solve命令时会在屏幕上弹出状态窗口;否则不弹出

简洁输出:如果选择该选项,则LINGO系统对求解结果报告等将以简洁形式输出;否则以详细形式输出

工具栏:如果选择该选项,则显示工具栏;否则不显示

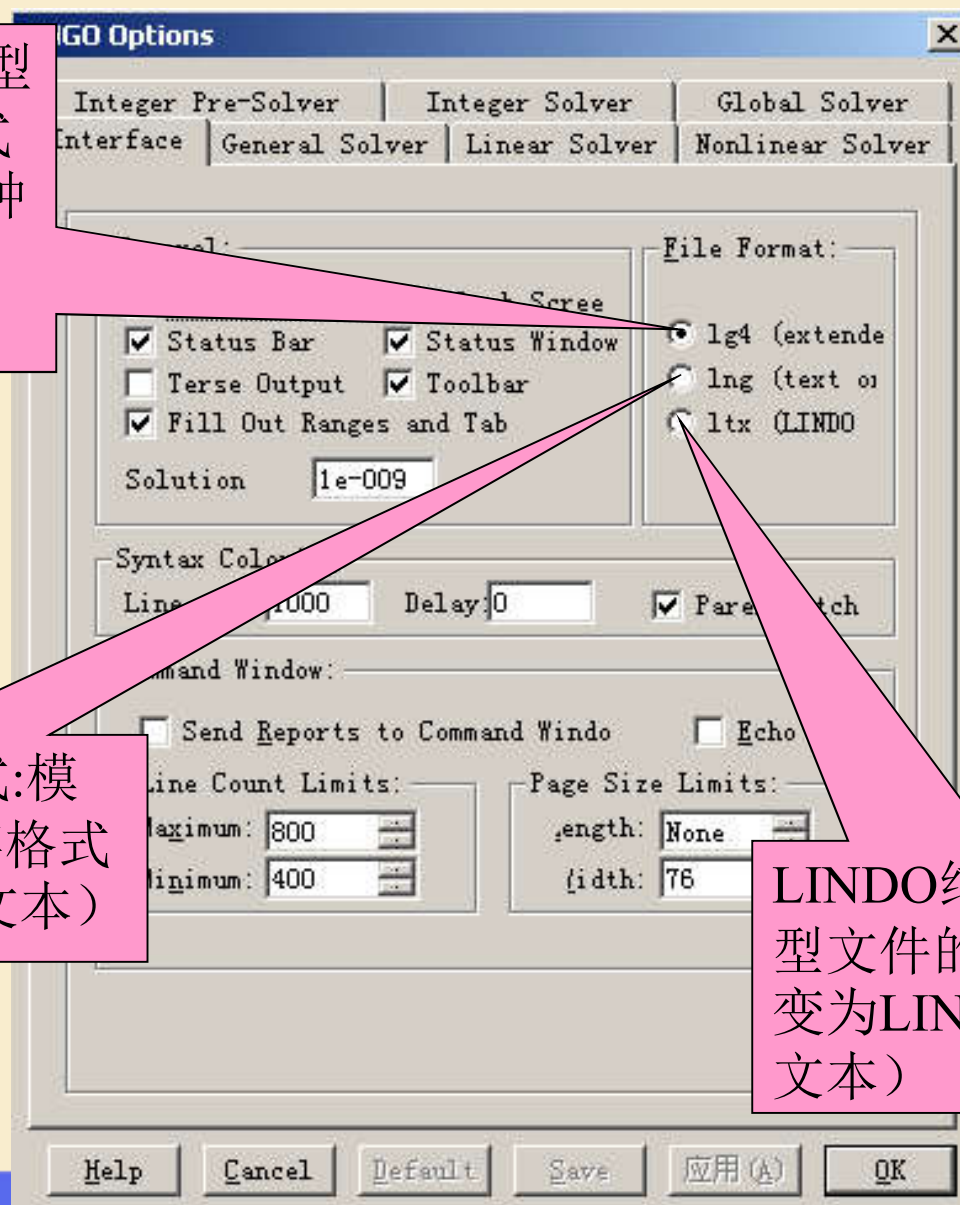


# LINGO|Options|Interface（界面）选项卡

LINGO扩展格式:模型文件的缺省保存格式是lg4格式（这是一种二进制文件，只有LINGO能读出）

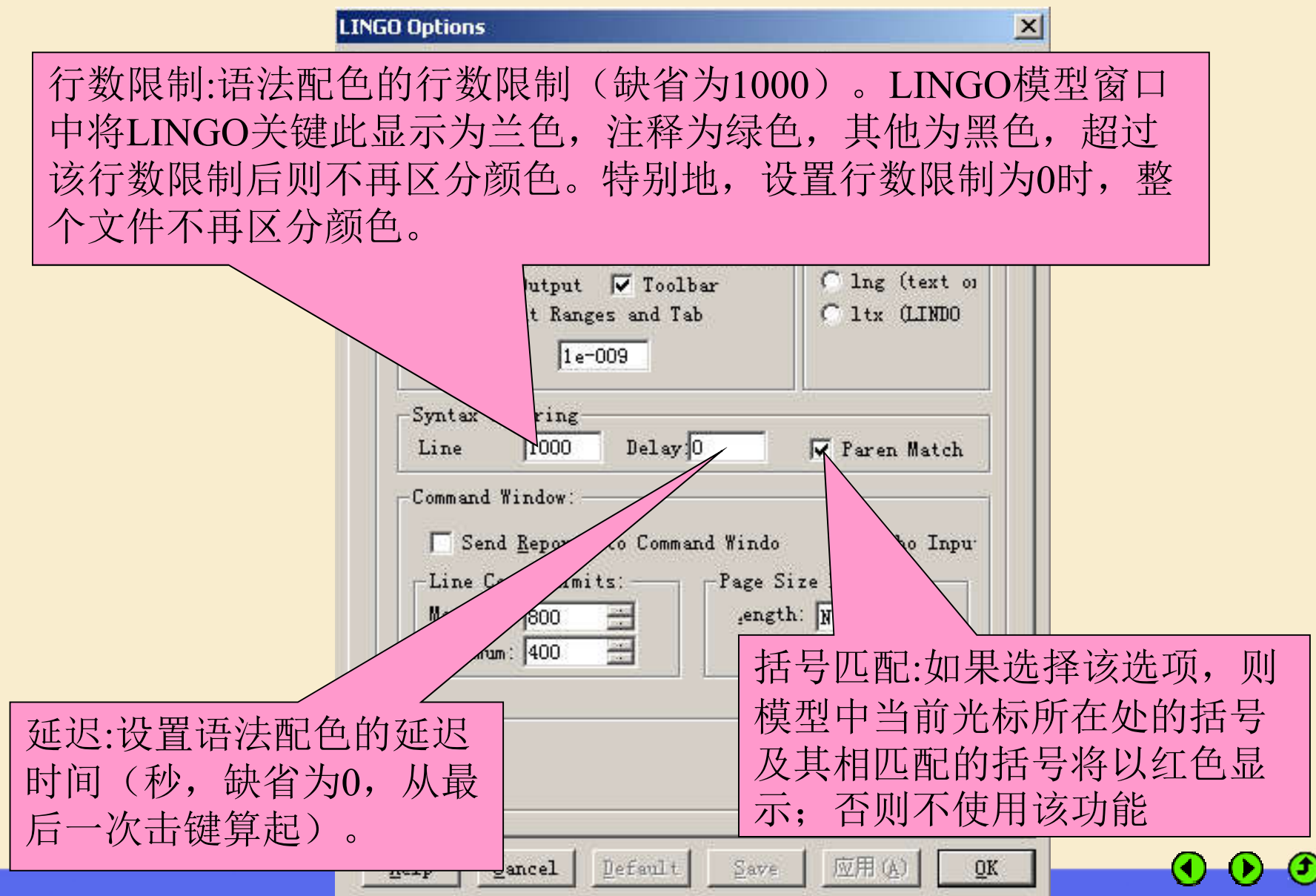
LINGO纯文本格式:模型文件的缺省保存格式变为lng格式（纯文本）

LINDO纯文本格式:模型文件的缺省保存格式变为LINDO格式（纯文本）





## LINGO|Options|Interface（界面）选项卡



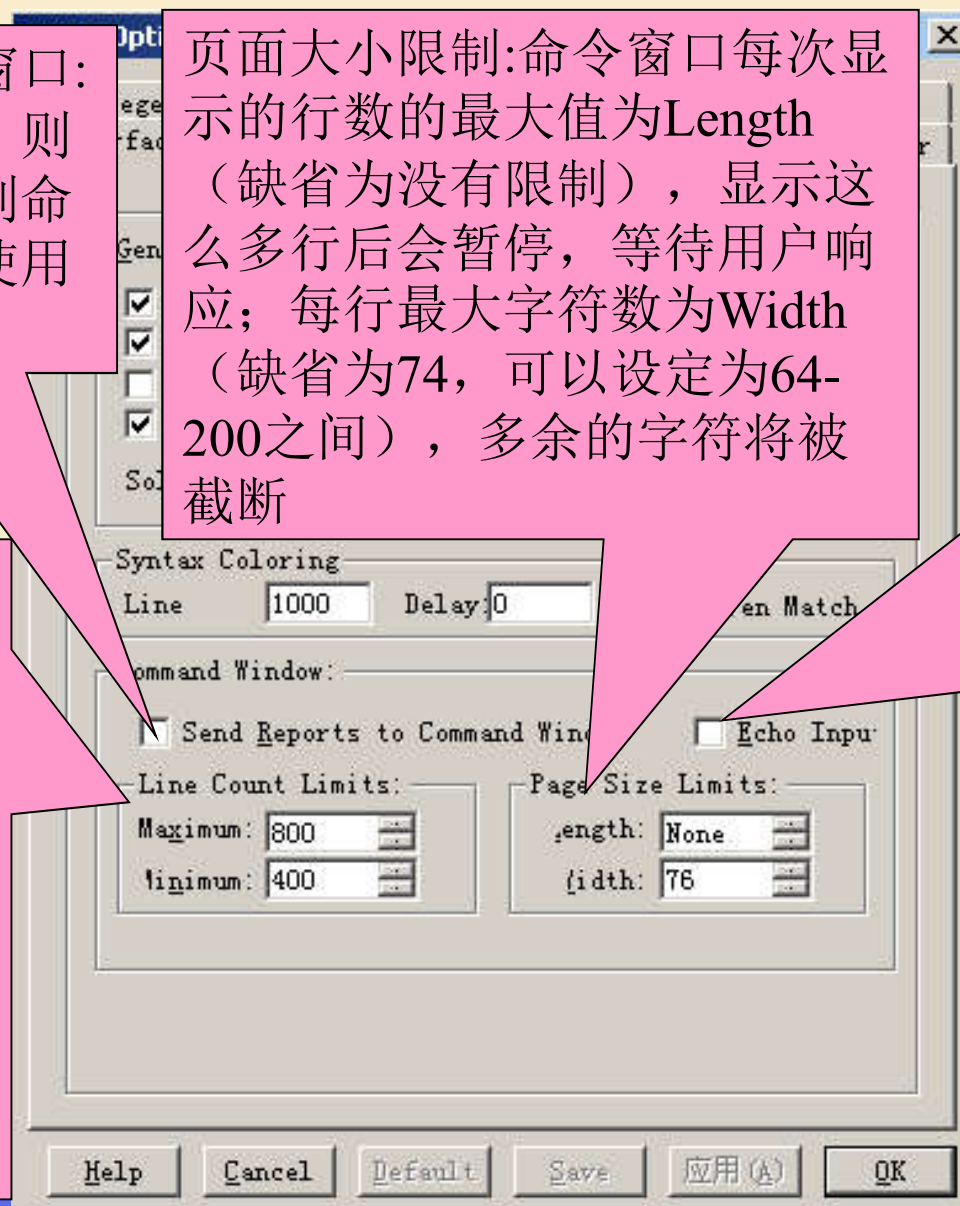
## LINGO|Options|Interface（界面）选项卡

报告发送到命令窗口:  
如果选择该选项, 则  
输出信息会发送到命  
令窗口; 否则不使用  
该功能

页面大小限制:命令窗口每次显  
示的行数的最大值为Length  
(缺省为没有限制), 显示这  
么多行后会暂停, 等待用户响  
应; 每行最大字符数为Width  
(缺省为74, 可以设定为64-  
200之间), 多余的字符将被  
截断

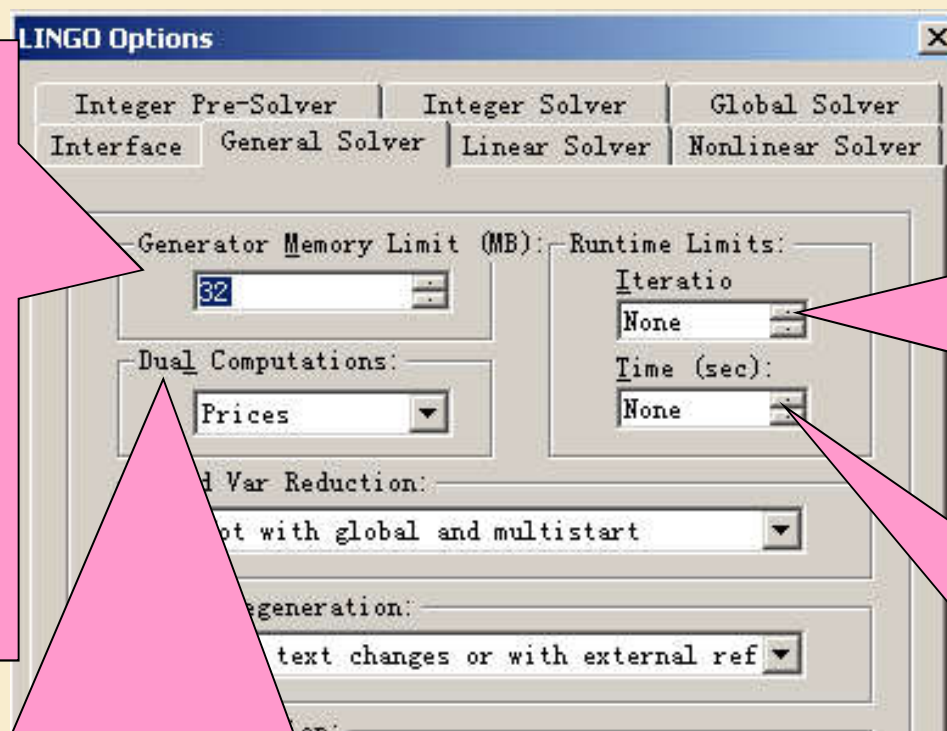
输入信息反馈:  
如果选择该选  
项, 则用  
File|Take  
Command命令  
执行命令脚本  
文件时, 处理  
信息会发送到  
命令窗口; 否  
则不使用该功  
能

行数限制:命令窗口  
能显示的行数的最  
大值为Maximum  
(缺省为800); 如  
果要显示的内容超  
过这个值, 每次从  
命令窗口滚动删除  
的最小行数为  
Minimum (缺省为  
400)



# LINGO|Options|General Solver(通用求解程序)选项卡

矩阵生成器的内存限制:缺省值为32M, 矩阵生成器使用的内存超过该限制, LINGO将报告  
"The model generator ran out of memory"



运行限制:  
迭代次数:求解一个模型时, 允许的最大迭代次数 (缺省值为无限)

运行限制:  
运行时间:求解一个模型时, 允许的最大运行时间 (秒) (缺省值为无限)

对偶计算:求解时控制对偶计算的级别, 有三种可能的设置:

None: 不计算任何对偶信息;

Prices: 计算对偶价格 (缺省设置);

Prices and Ranges:计算对偶价格并分析敏感性;

Prices, Opt Only: 只计算最优行的对偶价格。

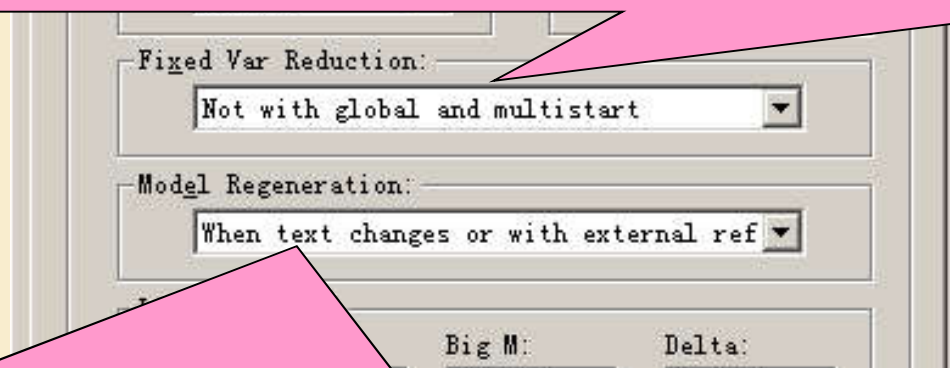
# LINGO|Options|General Solver(通用求解程序)选项卡

固定变量的归结、简化:求解前对固定变量的归结程度（相当于预处理程度）：

None: 不归结；

Always: 总是归结；

Not with global and multistart: 在全局优化和多初值优化程序中不归结。



模型的重新生成:控制重新生成模型的频率，有三种可能的设置：

Only when text changes: 只有当模型的文本修改后才再生成模型；

When text changes or with external references: 当模型的文本修改或模型含有外部引用时（缺省设置）；

Always: 每当有需要时。

# LINGO|Options|General Solver(通用求解程序)选项卡

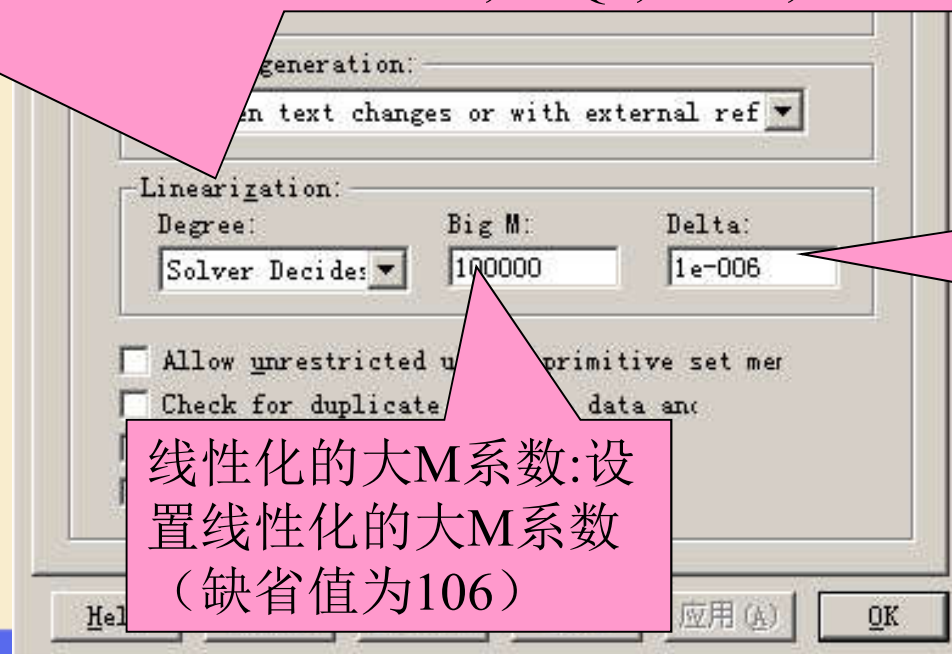
线性化程度:决定求解模型时线性化的程度, 有四种可能的设置:

**Solver Decides:** 若变量数小于等于12个, 则尽可能全部线性化; 否则不做任何线性化 (缺省设置)

**None:** 不做任何线性化

**Low:** 对函数@ABS(), @MAX(), @MIN(), @SMAX(), @SMIN(), 以及二进制变量与连续变量的乘积项做线性化

**High:** 同上, 此外对逻辑运算符#LE#, #EQ#, #GE#, #NE#做线性化



线性化的误差限:  
设置线性化的误差限 (缺省值为  $10^{-6}$ )

线性化的大M系数:设置线性化的大M系数 (缺省值为106)



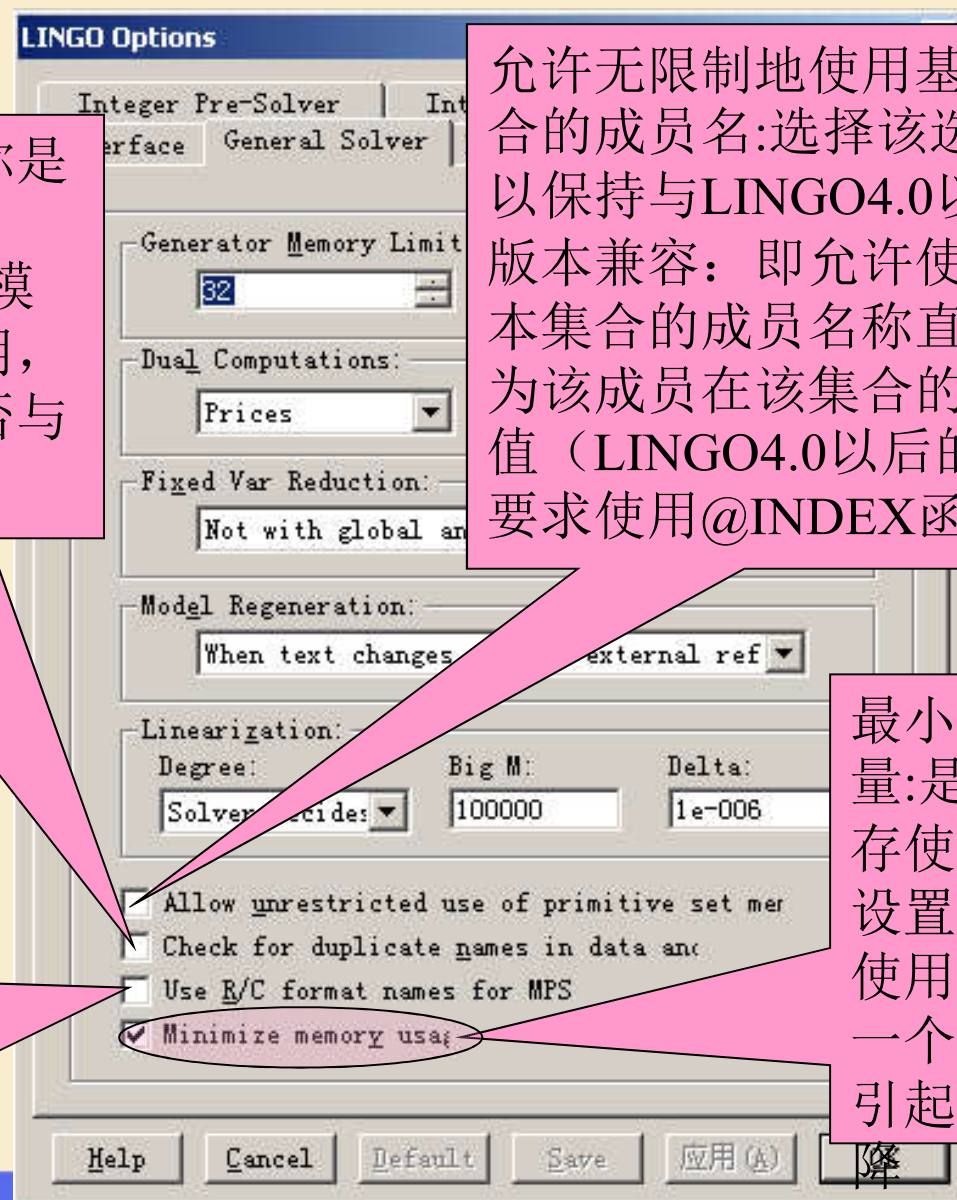
# LINGO|Options|General Solver(通用求解程序)选项卡

检查数据和模型中的名称是否重复使用:选择该选项,LINGO将检查数据和模型中的名称是否重复使用,如基本集合的成员名是否与决策变量名重复

检查数据和模型中的名称是否重复使用:选择该选项,LINGO将检查数据和模型中的名称是否重复使用,如基本集合的成员名是否与决策变量名重复

允许无限制地使用基本集合的成员名:选择该选项可以保持与LINGO4.0以前的版本兼容:即允许使用基本集合的成员名称直接作为该成员在该集合的索引值(LINGO4.0以后的版本要求使用@INDEX函数)

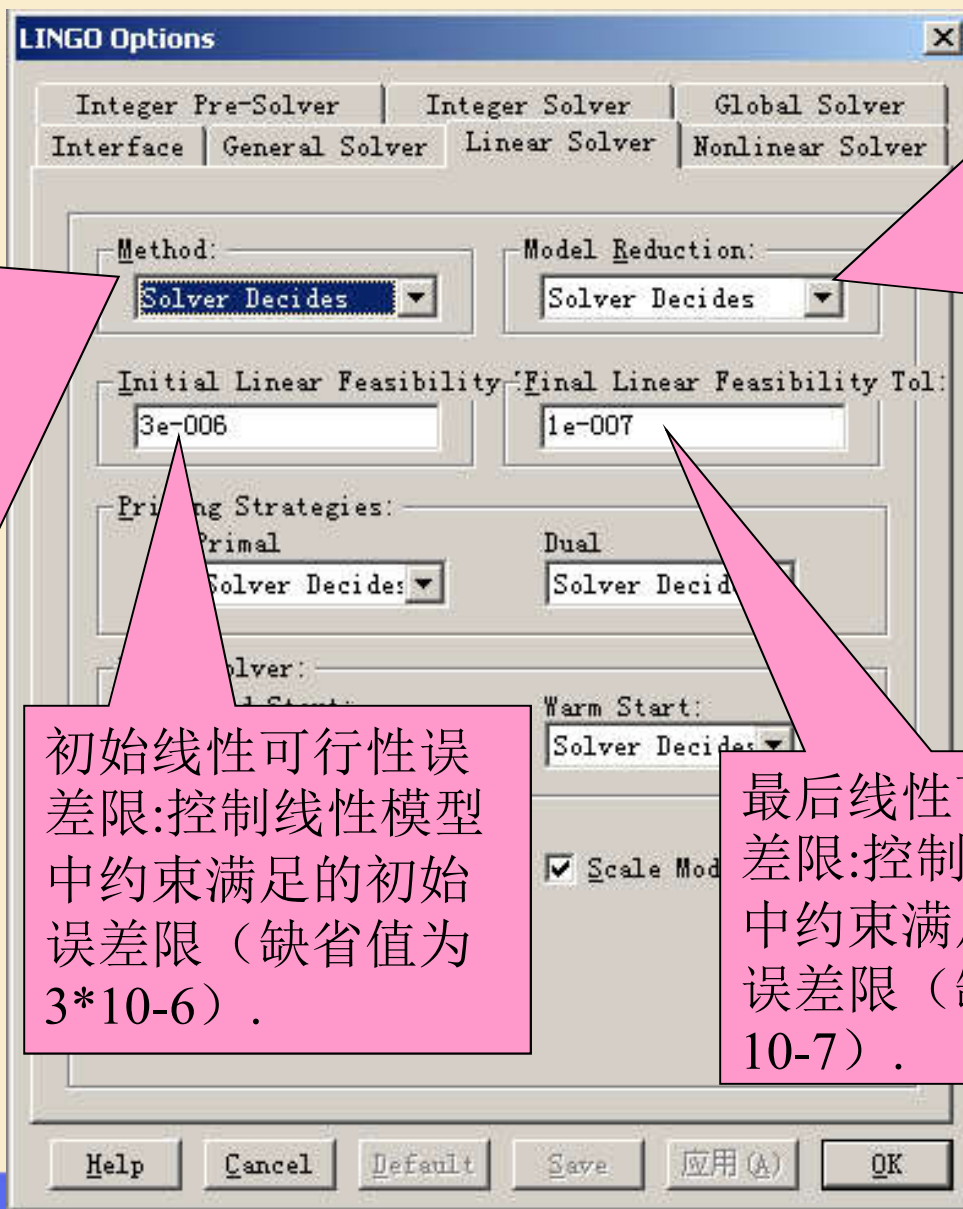
最小化内存使用量:是否最小化内存使用量。缺省设置为“是”。使用这个功能的一个缺点是可能引起计算速度下降



# LINGO|OPTIONS|Linear Solver(线性求解程序)选项卡

求解时的算法, 有四种可能的设置:

- Solver Decides: LINGO自动选择算法 (缺省设置)
- Primal Simplex: 原始单纯形法
- Dual Simplex: 对偶单纯形法
- Barrier: 障碍法 (即内点法)



初始线性可行性误差限: 控制线性模型中约束满足的初始误差限 (缺省值为  $3 \times 10^{-6}$ ) .

最后线性可行性误差限: 控制线性模型中约束满足的最后误差限 (缺省值为  $10^{-7}$ ) .

模型降维: 控制是否检查模型中的无关变量, 从而降低模型的规模:

- Off: 不检查
- On: 检查
- Solver Decides: LINGO自动决定 (缺省设置)



# LINGO|OPTIONS|Linear Solver(线性求解程序)选项卡



价格策略（决定出基变量的策略）

对偶单纯形法:有三种可能的设置:

Solver Decides:  
LINGO自动决定  
(缺省设置)

Dantzig: 按最大下降比例法确定出基变量

Steepest-Edge: 最陡边策略, 对所有可能的变量进行尝试, 找到使目标值下降最多的出基变量

原始单纯形法:有三种可能的设置:

Solver Decides: LINGO自动决定 (缺省设置)

Partial: LINGO 对一部分可能的出基变量进行尝试

Devex: 用Steepest-Edge (最陡边) 近似算法对所有可能的变量进行尝试, 找到使目标值下降最多的出基变量

# LINGO|OPTIONS|Linear Solver(线性求解程序)选项卡

冷启动时:设置LINGO|Debug命令调试所使用的程序（冷启动的含义是不从当前基开始，而是从头开始运行）。有四种可能的设置：

Solver Decides: LINGO自动选择算法(缺省设置)

Primal Simplex: 原始单纯形法

Dual Simplex: 对偶单纯形法

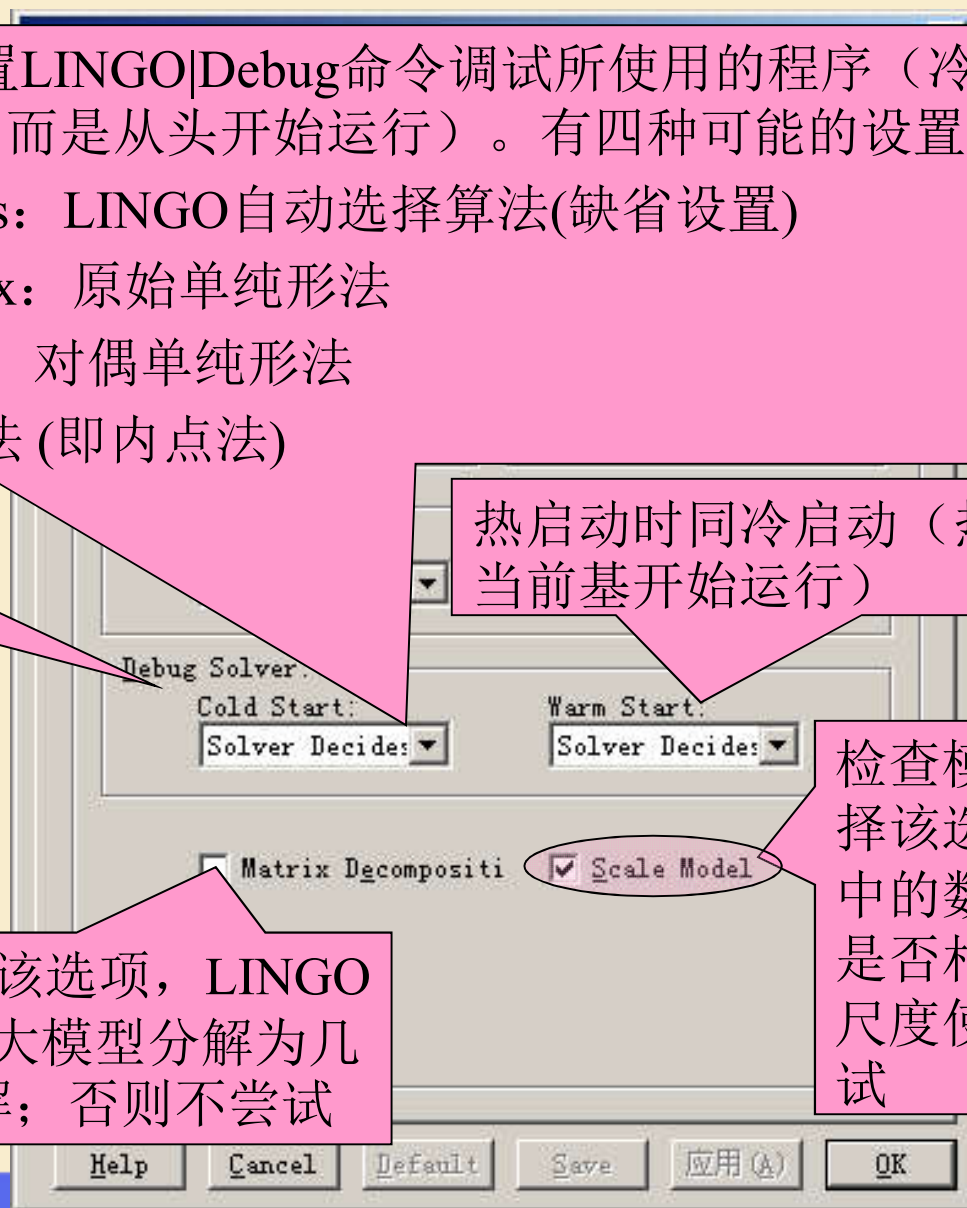
Barrier: 障碍法 (即内点法)

调试时采用的  
求解程序

热启动时同冷启动（热启动的含义是从当前基开始运行）

矩阵分解:选择该选项，LINGO  
将尝试将一个大模型分解为几  
个小模型求解；否则不尝试

检查模型的数据平衡性:选  
择该选项,LINGO检查模型  
中的数据是否平衡（数量级  
是否相差太大）并尝试改变  
尺度使模型平衡；否则不尝  
试



# LINGO|OPTIONS|Nonlinear Solver(非线性求解程序)选项卡

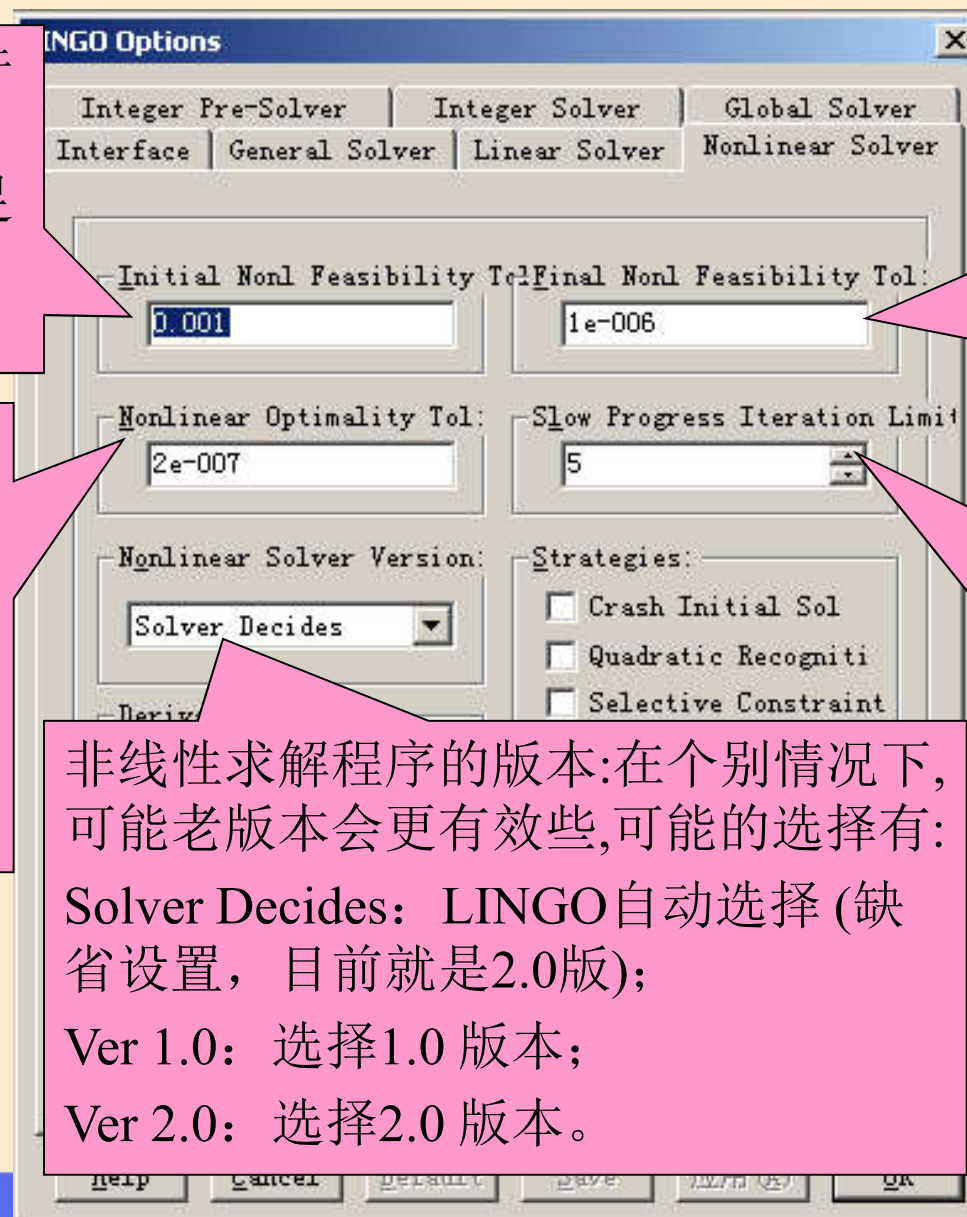
初始非线性可行性误差限:控制模型中约束满足的初始误差限(缺省值为 $10^{-3}$ )

非线性规划的最优性误差限:当目标函数在当前解的梯度小于等于这个值以后,停止迭代(缺省值为 $2 \times 10^{-7}$ )

非线性求解程序的版本:在个别情况下,可能老版本会更有效些,可能的选择有:  
Solver Decides: LINGO自动选择(缺省设置,目前就是2.0版);  
Ver 1.0: 选择1.0 版本;  
Ver 2.0: 选择2.0 版本。

最后非线性可行性误差限:控制模型中约束满足的最后误差限(缺省值为 $10^{-6}$ )

缓慢改进的迭代次数的上限:当目标函数在连续这么多次迭代没有显著改进以后,停止迭代(缺省值为5)



# LINGO|OPTIONS|Nonlinear Solver(非线性求解程序)选项卡

导数计算方式:设置导数计算方式,有5种选择:

Solver Decides: LINGO自动选择 (缺省设置)

Backward analytical: 前向解析法计算导数;

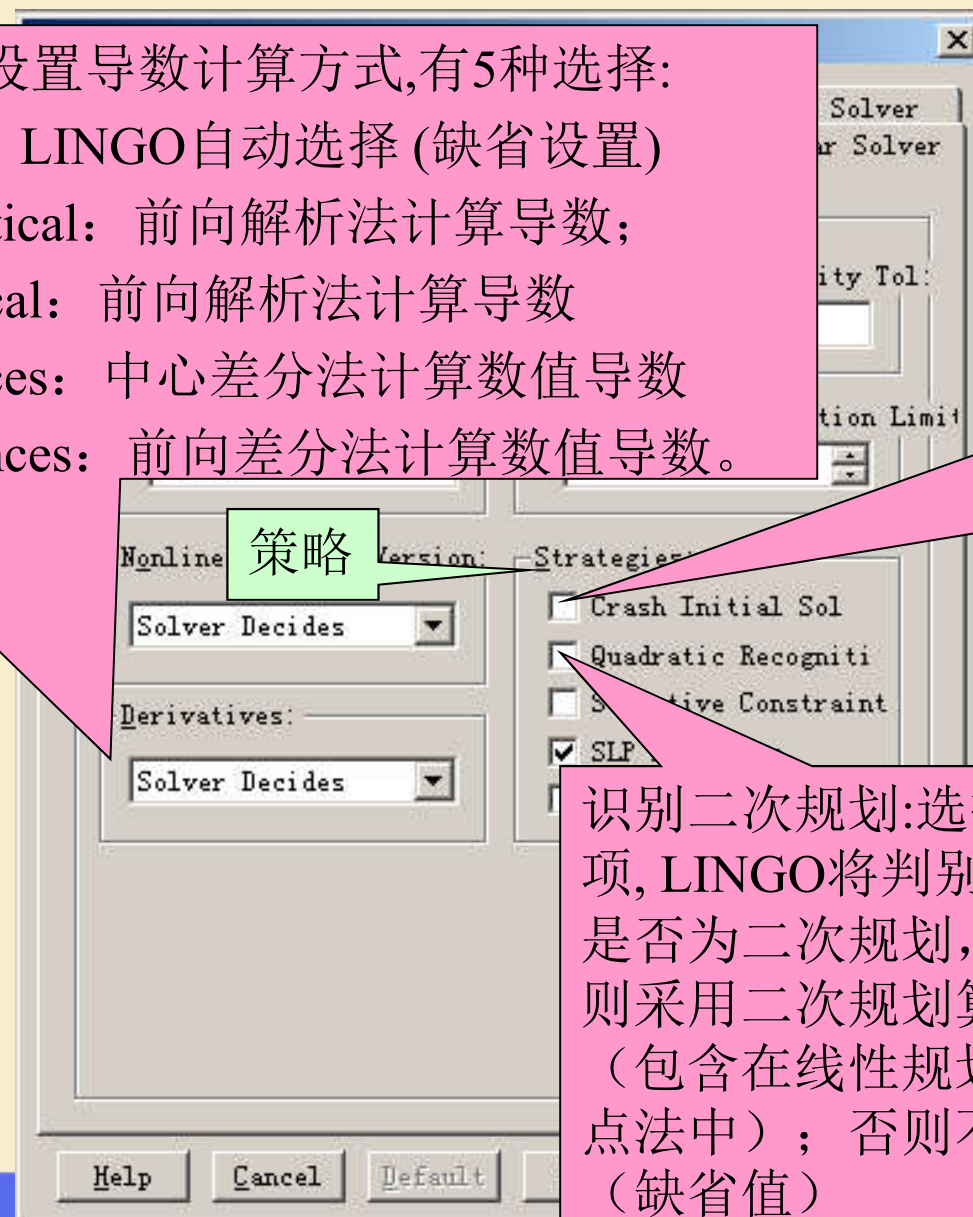
Forward analytical: 前向解析法计算导数

Central differences: 中心差分法计算数值导数

Forward differences: 前向差分法计算数值导数。

生成初始解:选择该选项, LINGO将用启发式方法生成初始解;否则不生成(缺省值)

识别二次规划:选择该选项, LINGO将判别模型是否为二次规划, 若是则采用二次规划算法 (包含在线性规划的内点法中); 否则不判别 (缺省值)

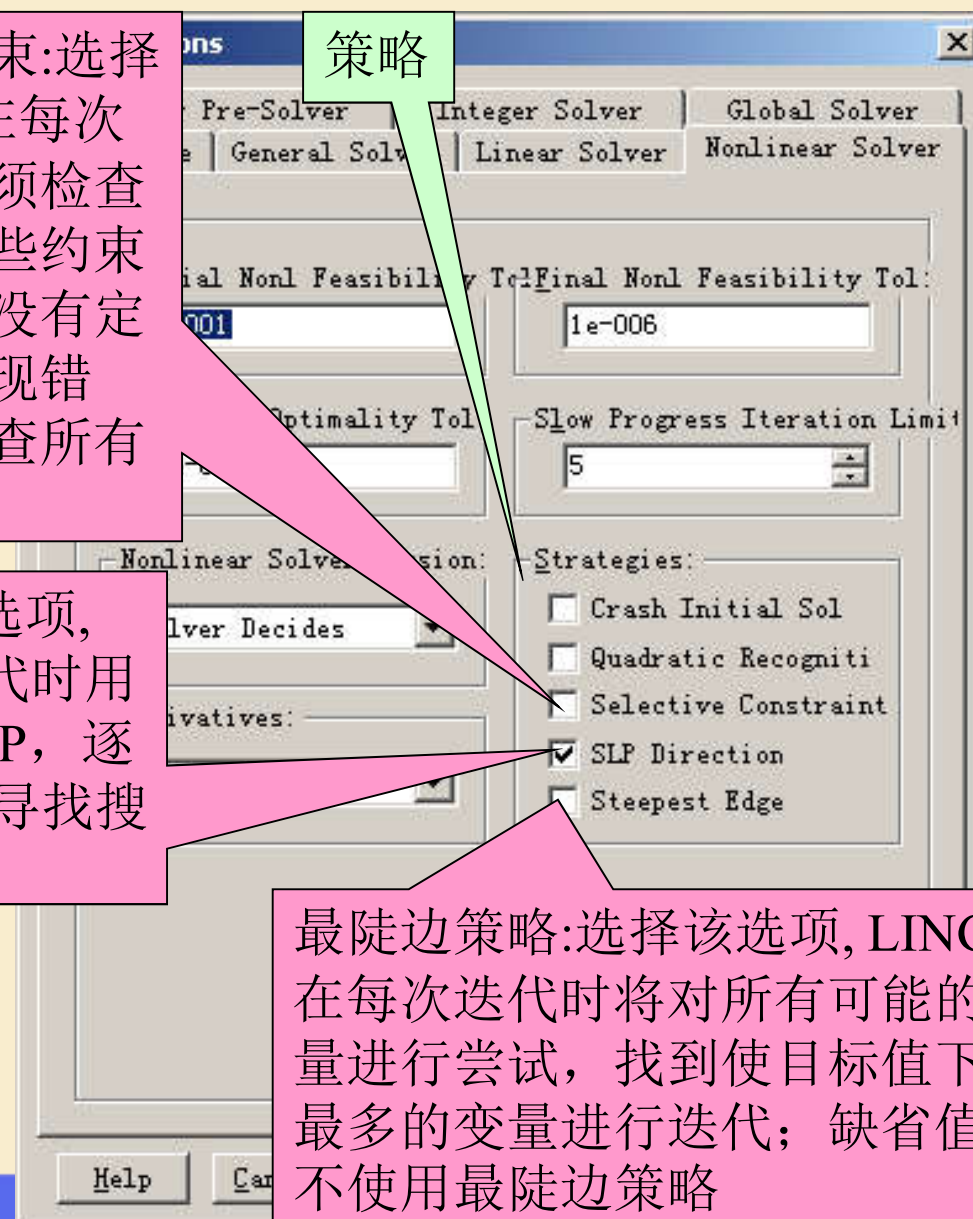


# LINGO|OPTIONS|Nonlinear Solver(非线性求解程序)选项卡

有选择地检查约束:选择该选项, LINGO在每次迭代时只检查必须检查的约束(如果有些约束函数在某些区域没有定义,这样做会出现错误); 否则, 检查所有约束(缺省值)

SLP方向:选择该选项, LINGO在每次迭代时用SLP (Successive LP, 逐次线性规划)方法寻找搜索方向(缺省值)

最陡边策略:选择该选项, LINGO在每次迭代时将对所有可能的变量进行尝试, 找到使目标值下降最多的变量进行迭代; 缺省值为不使用最陡边策略





## LINGO|OPTIONS|Integer Pre-Solver(整数预处理程序)选项卡

水平:控制采用启发式搜索的次数  
(缺省值为3, 可能的值为0-100).  
启发式方法的目的是从分枝节点的连续解出发, 搜索一个好的整数解。

启发式方法

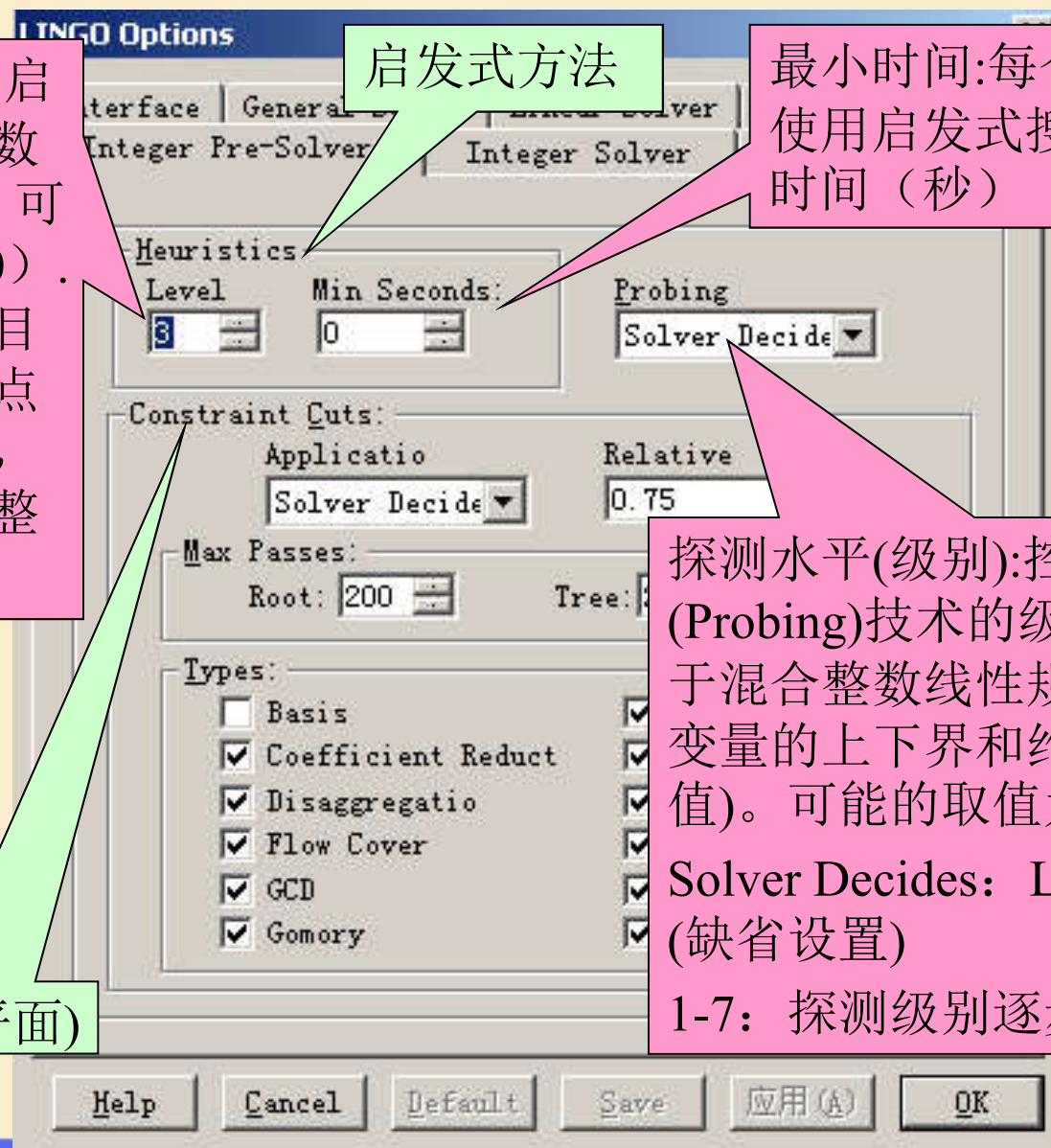
最小时间:每个分枝节点使用启发式搜索的最小时间(秒)

约束的割(平面)

探测水平(级别):控制采用探测(Probing)技术的级别(探测能够用于混合整数线性规划模型, 收紧变量的上下界和约束的右端项的值)。可能的取值为:

Solver Decides: LINGO自动决定(缺省设置)

1-7: 探测级别逐步升高。



## LINGO|OPTIONS|Integer Pre-Solver(整数预处理程序)选项卡

应用节点:控制在分枝定界树中,哪些节点需要增加割(平面),可能的取值为:

Root Only: 仅根节点增加割(平面)

All Nodes: 所有节点均增加割(平面)

Solver Decides: LINGO自动决定(缺省设置)

约束的割  
(平面)

最大迭代检查的次数:为了寻找合适的割,最大迭代检查的次数。

对根节点的次数(缺省值为200)

相对上限:控制生成的割(平面)的个数相对于原问题的约束个数的上限(比值),缺省值为0.75

对其他节点的次数(缺省值为2)

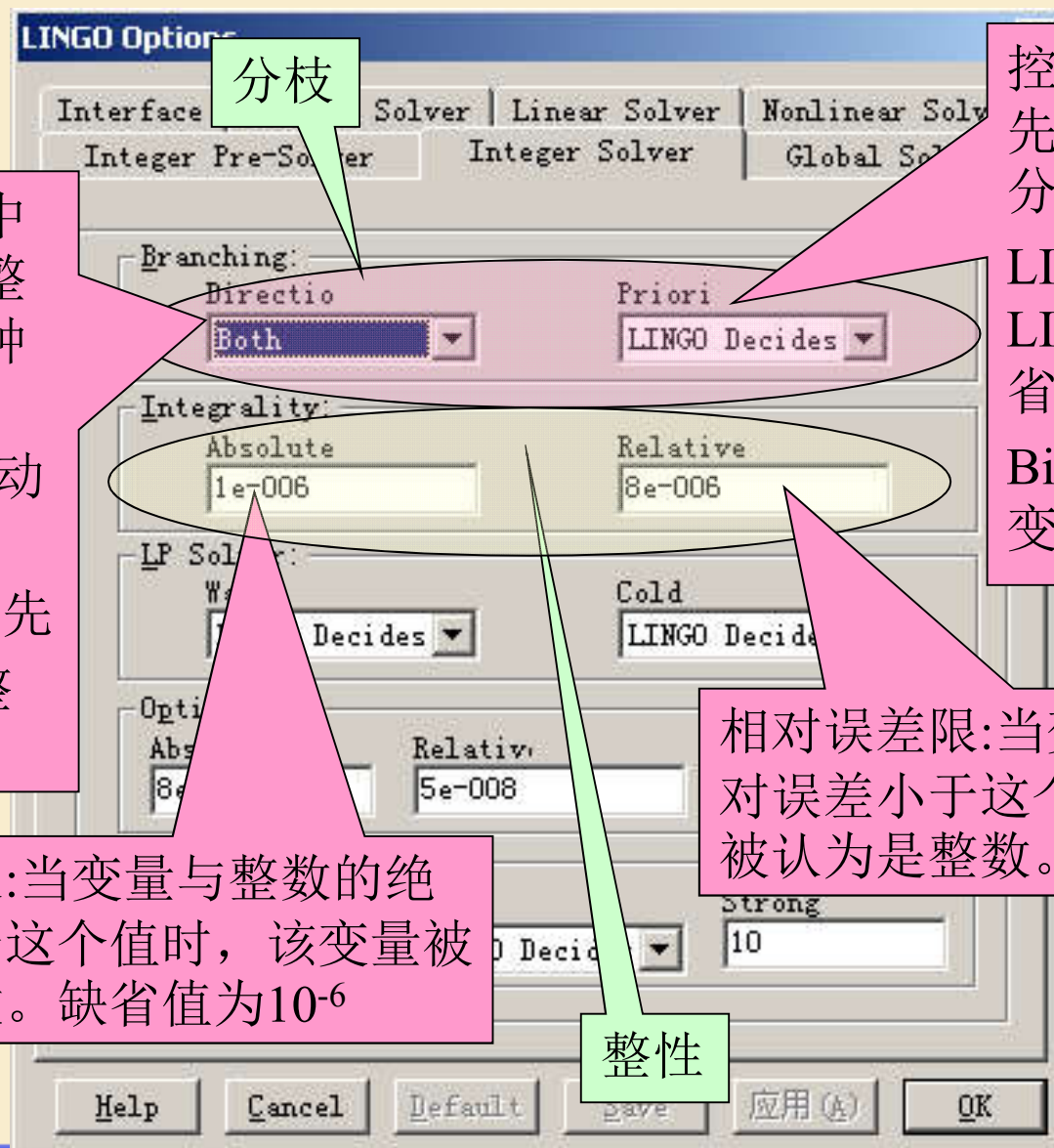
类型:控制生成的割(平面)的策略

The screenshot shows the 'Integer Pre-Solver' options dialog box. It has a title bar with a close button. The main area is divided into several sections:

- Constraint Cuts:** A dropdown menu set to 'Solver Decide' and a text box for 'Relative' set to '0.75'.
- Max Passes:** Two spinners. 'Root' is set to '200' and 'Tree' is set to '2'.
- Types:** A list of checkboxes for different cut types. The following are checked: 'Coefficient Reduct', 'Disaggregatio', 'Flow Cover', 'GCD', 'Gomory', 'GUB', 'Knapsack Cover', 'Lattice', 'Liftin', 'Objective', and 'Plant Locati'. 'Basis' is unchecked.

At the bottom, there are buttons for 'Help', 'Cancel', 'Default', 'Save', '应用(A)', and 'OK'.

# LINGO|OPTIONS|Integer Solver(整数求解程序)选项卡





# LINGO|OPTIONS|Integer Solver(整数求解程序)选项卡

热启动:当以前面的求解结果为基础, 热启动求解程序时采用的算法, 有四种可能的设置:

LINGO Decides: LINGO自动选择算法(缺省)

Primal Simplex: 原始单纯形法

Dual Simplex: 对偶单纯形法

Barrier: 障碍法 (即内点法)

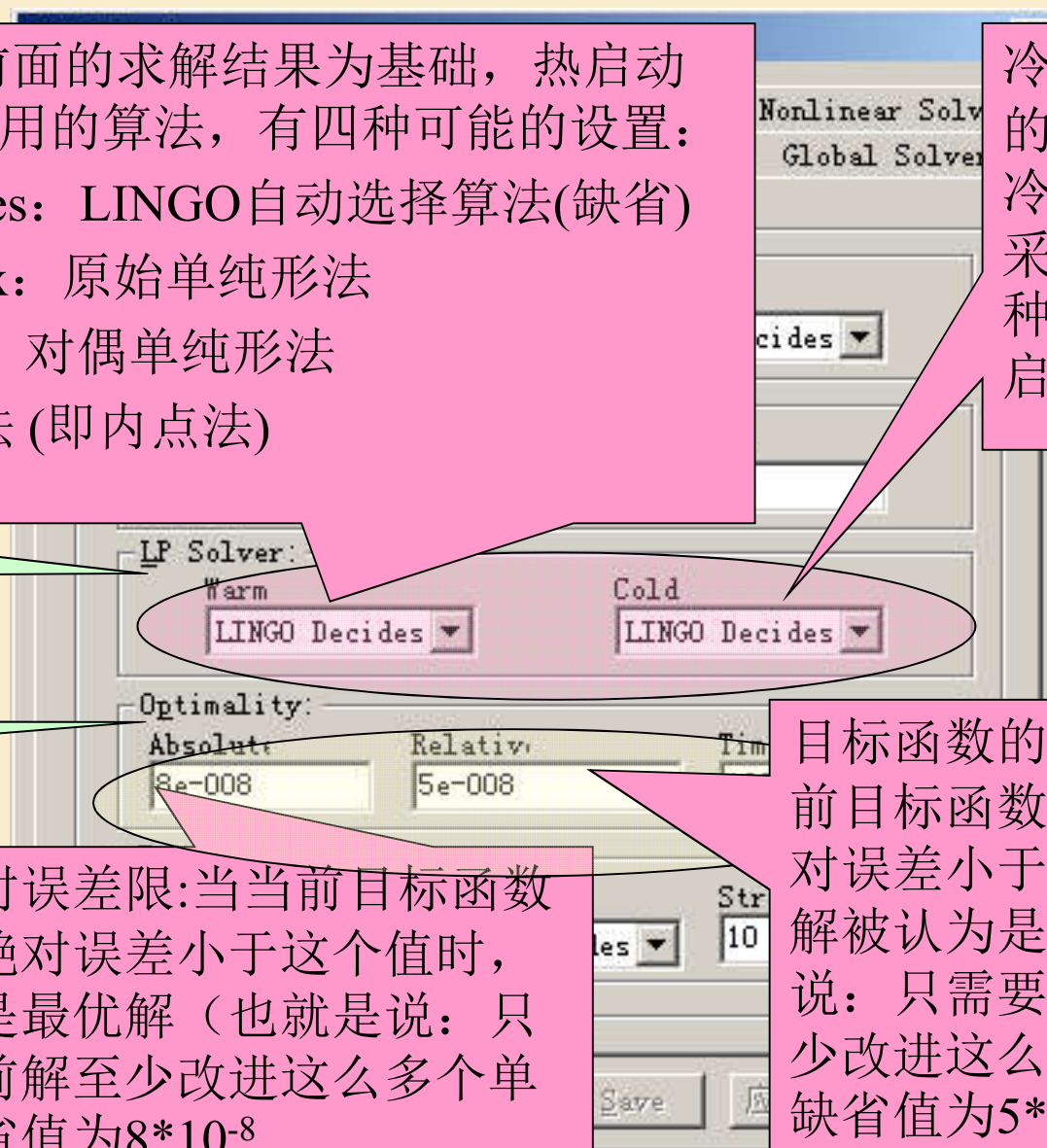
冷启动:当不以前面的求解结果为基础, 冷启动求解程序时采用的算法, 有四种可能的设置(同热启动)

LP求解程序

最优性

目标函数的绝对误差限:当当前目标函数值与最优值的绝对误差小于这个值时, 当前解被认为是最优解 (也就是说: 只需要搜索比当前解至少改进这么多个单位的解)。缺省值为 $8 \times 10^{-8}$

目标函数的相对误差限:当当前目标函数值与最优值的相对误差小于这个值时, 当前解被认为是最优解 (也就是说: 只需要搜索比当前解至少改进这么多百分比的解)。缺省值为 $5 \times 10^{-8}$



# LINGO|OPTIONS|Integer Solver(整数求解程序)选项卡

节点选择:控制如何选择节点的分枝求解, 有以下选项:

LINGO Decides: LINGO自动选择  
(缺省设置)

Depth First:按深度优先

Worst Bound:选择具有最坏界的节点

Best Bound:选择具有最好的界的节点

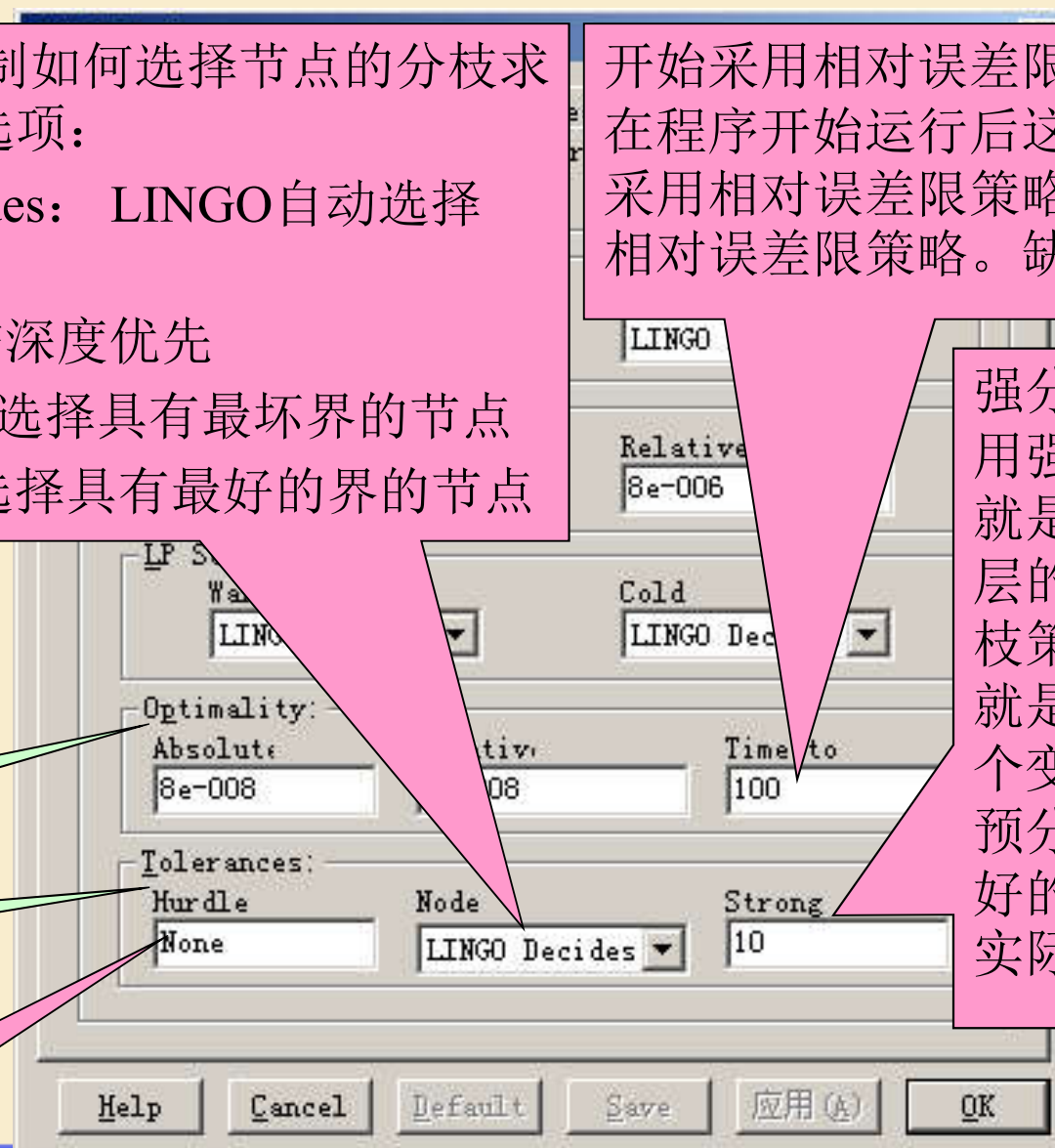
开始采用相对误差限的时间(秒):  
在程序开始运行后这么多秒内, 不采用相对误差限策略; 此后才使用相对误差限策略。缺省值为100秒

强分枝的层数:控制采用强分枝的层数。也就是说, 对前这么多层的分枝, 采用强分枝策略。所谓强分枝, 就是在一个节点对多个变量分别尝试进行预分枝, 找出其中最好的解(变量)进行实际分枝

最优性

误差限

篱笆值(同  
LINDO)



# LINGO|OPTIONS|Global Solver(全局最优求解程序)选项卡

全局最优求解程序

使用全局最优求解程序:选择该选项,LINGO将用全局最优求解程序求解模型,尽可能得到全局最优解(求解花费的时间可能很长);否则不使用全局最优求解程序,通常只得到局部最优解

变量上界

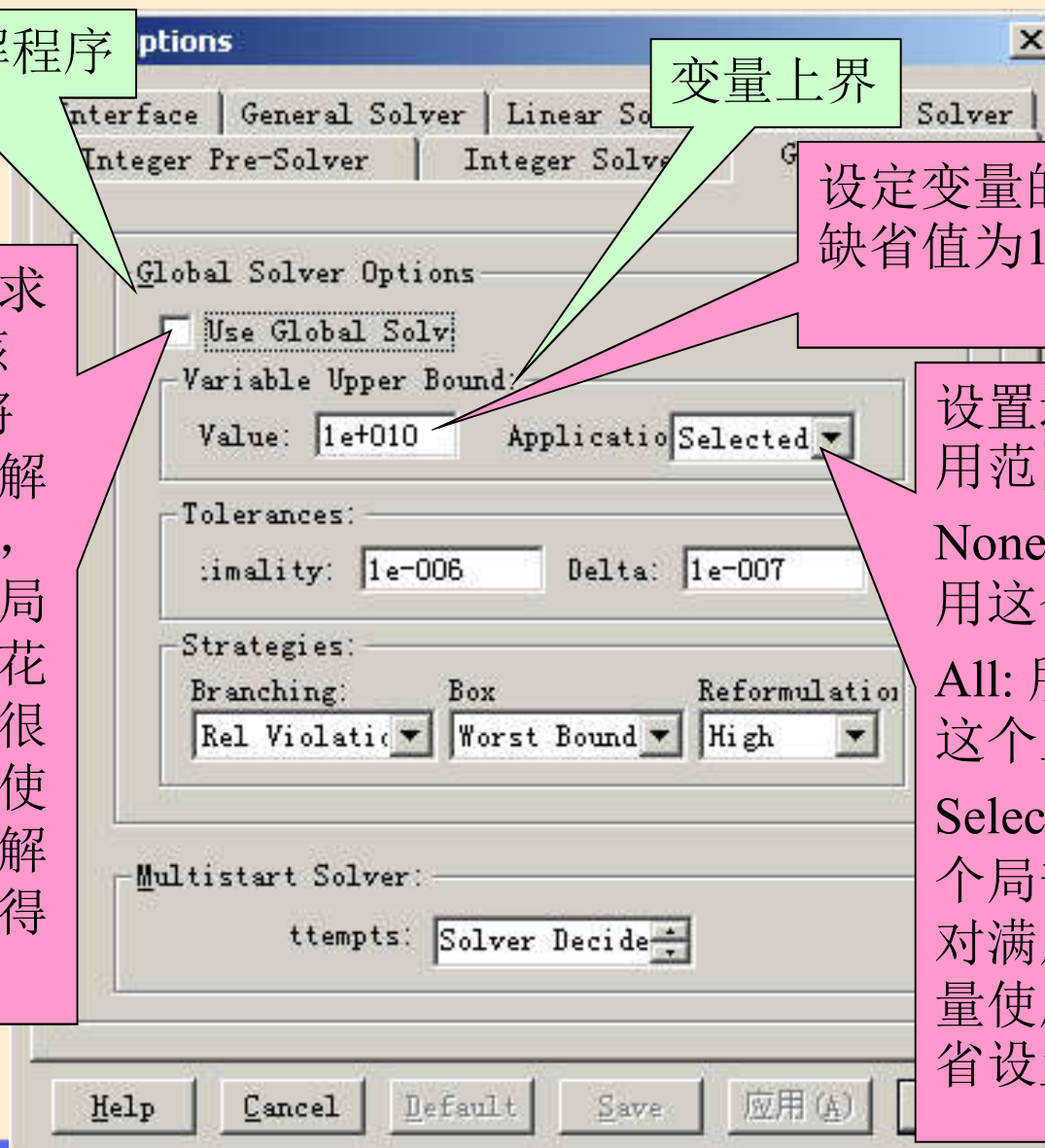
设定变量的上界,缺省值为 $10^{10}$

设置这个界的三种应用范围:

None: 所有变量都不使用这个上界;

All: 所有变量都使用这个上界;

Selected: 先找到第1个局部最优解,然后对满足这个上界的变量使用这个上界(缺省设置)



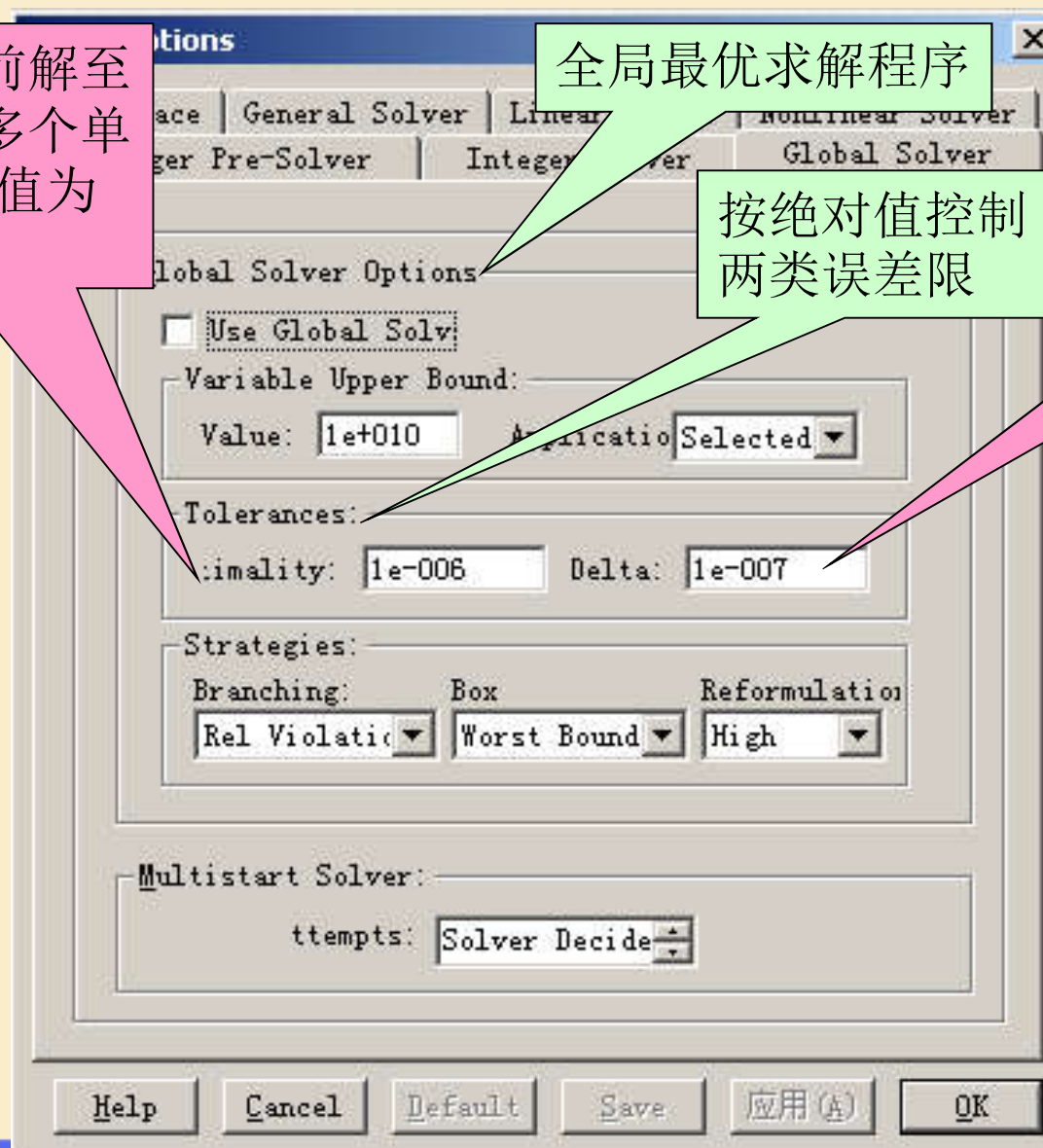
# LINGO|OPTIONS|Global Solver(全局最优求解程序)选项卡

只搜索比当前解至少改进这么多个单位的解(缺省值为 $10^{-6}$ )

全局最优求解程序

按绝对值控制  
两类误差限

全局最优求解程序在凸化过程中增加的约束的误差限(缺省值为 $10^{-7}$ )





# LINGO|OPTIONS|Global Solver(全局最优求解程序)选项卡

第1次对变量分枝时使用的分枝策略

Absolute Width(绝对宽度)

Local Width(局部宽度)

Global Width(全局宽度)

Global Distance(全局距离)

Abs (Absolute)

Violation(绝对冲突)

Rel (Relative) Violation(相对冲突, 缺省设置)

选择活跃分枝节点的方法:

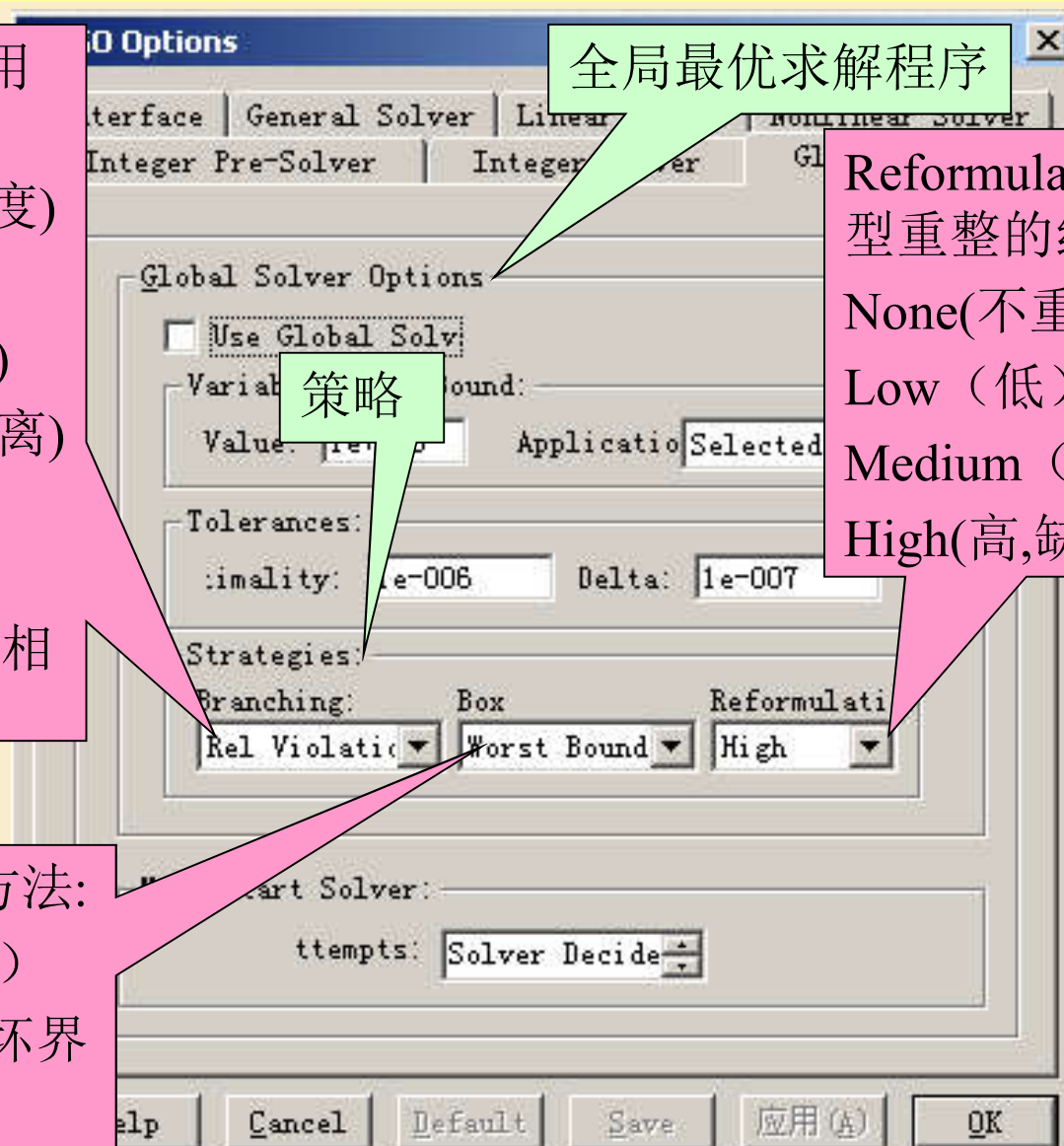
Depth First (深度优先)

Worst Bound (具有最坏界的分枝优先, 缺省)

全局最优求解程序

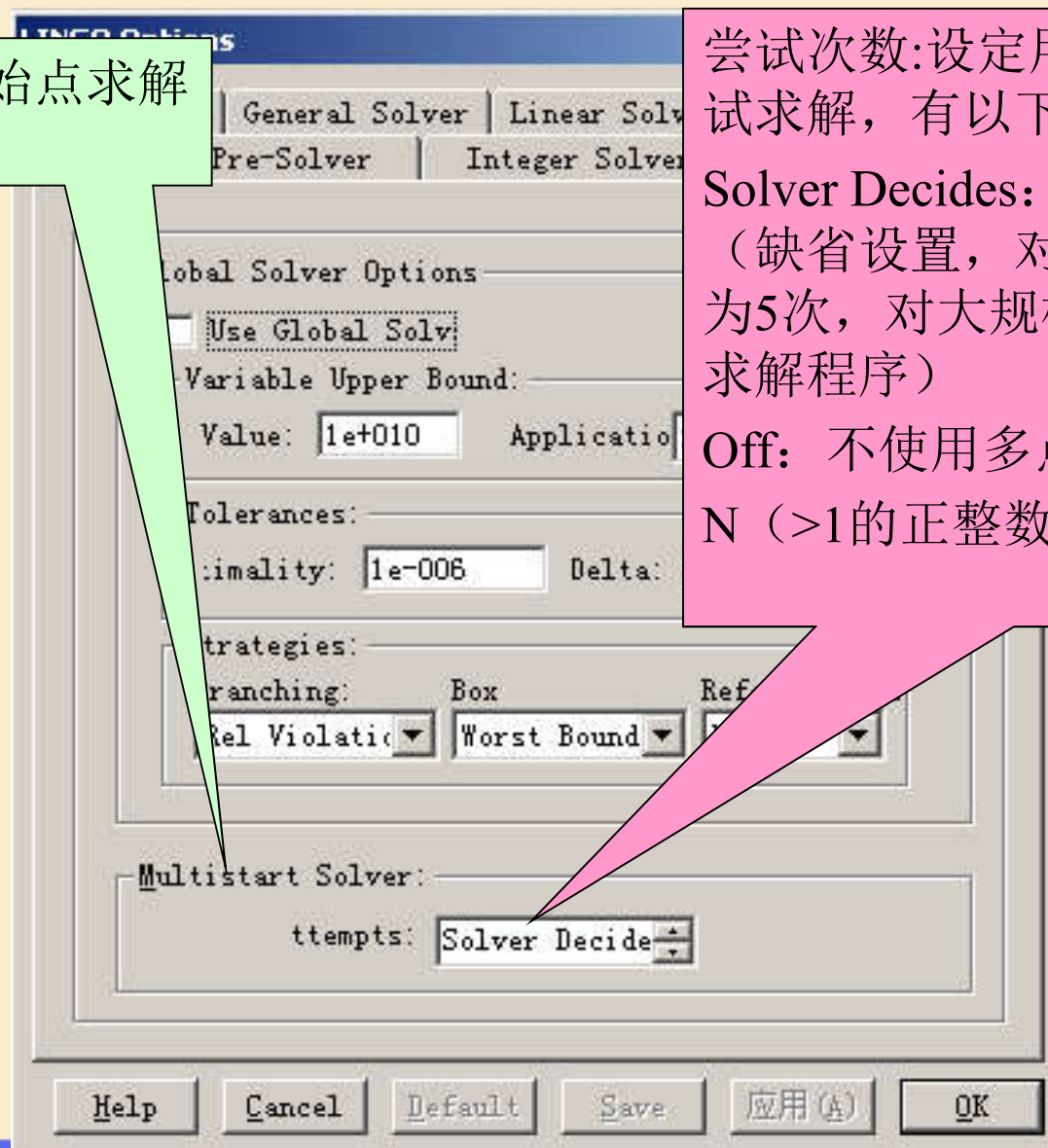
Reformulation: 模型重整的级别:  
None(不重整)  
Low (低)  
Medium (中)  
High(高,缺省)

策略



# LINGO|OPTIONS|Global Solver(全局最优求解程序)选项卡

多初始点求解  
程序



尝试次数:设定用多少个初始点尝试求解,有以下几种可能的设置:  
Solver Decides: 由LINGO决定  
(缺省设置,对小规模NLP问题为5次,对大规模问题不使用多点求解程序)

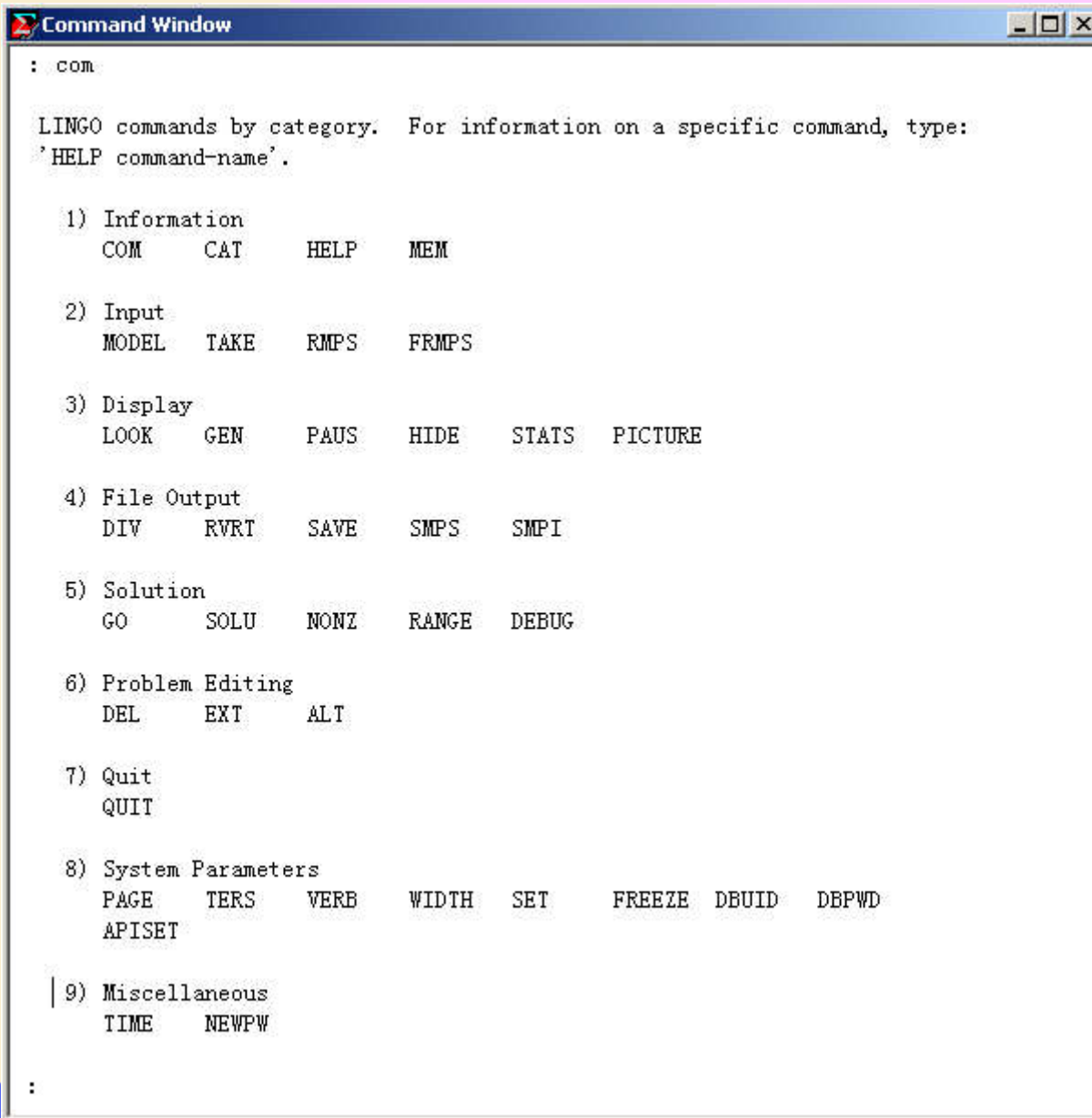
Off: 不使用多点求解程序

N (>1的正整数): N点求解

## 5. LINGO命令窗口

1. LINGO入门
2. 在LINGO中使用集合
3. 运算符和函数
4. LINGO的主要菜单命令
5. LINGO命令窗口
6. 习题

# LINGO命令行 模式



```
Command Window
: com

LINGO commands by category.  For information on a specific command, type:
'HELP command-name'.

1) Information
  COM    CAT    HELP    MEM

2) Input
  MODEL  TAKE    RMPS    FRMPS

3) Display
  LOOK   GEN     PAUS    HIDE    STATS  PICTURE

4) File Output
  DIV     RVRT    SAVE    SMPS    SMPPI

5) Solution
  GO      SOLU    NONZ    RANGE  DEBUG

6) Problem Editing
  DEL     EXT     ALT

7) Quit
  QUIT

8) System Parameters
  PAGE    TERS    VERB    WIDTH  SET     FREEZE  DBUID  DBPWD
  APISET

9) Miscellaneous
  TIME    NEWPW

:
```

LINGO有两种命令模式：  
Windows模式, 命令行  
(Command-Line) 模式。  
这里简单介绍一下命令  
行模式下的主要行命令。

可以通过菜单命令

“Window|Command  
Window (Ctrl+1)” 打  
开命令窗口，在命令窗  
口下操作。

在命令窗口下的提示符  
“: ” 后面键入

"COMMANDS" (COM)  
可以看到LINGO的所有  
行命令



## 部分行命令的基本功能

在LINGO中不再支持LINDO的部分行命令，如DATE，TABL，SDBC，FBS，FPUN，SMPN等。LINGO也增加了一些与LINDO不同的命令（见下表）。

LINGO 行命令	功能简介
MEM	显示矩阵生成器(建模语言)的内存使用情况（不包括求解程序使用的内存）
MODEL	开始输入LINGO模型
FRMPS	读出自由格式的MPS文件（而RMPS命令读出固定格式的MPS文件）
GEN	编译并以代数形式生成展开的模型，参见 “LINGO Generate”菜单命令

## 部分行命令的基本功能

LINGO行命令	功能简介
HIDE	用户对模型设定密码，隐藏模型文本的内容（如：为了保护你的知识产权时）
SMPI	以MPI文件格式保存模型（该文件主要供LINDO API软件阅读，提供接口）
FREEZE	冻结（即保存）系统参数（包括SET命令可以设定的所有参数），下次启动LINGO这些参数仍然有效;实际上，这些参数保存在LINGO目录下的LINGO.CNF文件中；用户随时可以运行“SET DEFAULT”和“FREEZE”两条命令恢复缺省设置。
DBUID	设定数据库的用户名，该用户名在@ODBC()函数存取数据库时使用
DBPWD	设定数据库的使用密码，该密码在@ODBC()函数存取数据库时使用
APISET	该命令用于设定LINDO API所需要的参数（当然，只有当某个参数不能通过LINGO的前端命令“SET”来修改时才需要使用APISET命令）。因此，这是比较专业的参数选项，具体请参见LINDO API的使用手册。

## LINGO中的SET命令

相同的命令，LINGO中与在LINDO中也不完全相同。

LINGO中的SET命令能设定的参数比LINDO中多。

凡是用户能够控制的LINGO系统参数，SET命令都能够对它进行设置。SET命令的使用格式为：

SET parameter\_name | parameter\_index [parameter\_value]

parameter\_name是参数名，parameter\_index是参数索引（编号），parameter\_value是参数值。当不写出参数值时，则SET命令的功能是显示该参数当前的值。

“SET DEFAULT”命令用于将所有参数恢复为系统的默认值（缺省值）。这些设置如果不用“FREEZE”命令保存 to 配置文件LINGO.CNF中，退出LINGO后这些设置失效。

## SET 可设置的参数及简要功能

参数名	缺省值	简要说明
ILFTOL	0.3e-5	初始线性可行误差限
FLFTOL	0.1e-6	最终线性可行误差限
INFTOL	0.1e-2	初始非线性可行误差限
FNFTOL	0.1e-5	最终非线性可行误差限
RELINT	0.8e-5	相对整性误差限
NOPTOL	0.2e-6	非线性规划(NLP)的最优性误差限
ITRSLW	5	缓慢改进的迭代次数的上限
DERCMP	0	导数 (0:数值导数, 1:解析导数)
ITRLIM	0	迭代次数上限 (0:无限制)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
TIMLIM	0	求解时间的上限（秒） (0:无限制)
OBJECTS	1	是否采用目标割平面法 (1:是, 0:否)
MXMEMB	32	模型生成器的内存上限（兆字节） (对某些机器，可能无意义)
CUTAPP	2	割平面法的应用范围(0:根节点, 1:所有节点, 2:LINGO自动决定)
ABSINT	.0000001	整性绝对误差限
HEURIS	3	整数规划（IP）启发式求解次数 (0:无, 可设定为0~100)

## SET 可设置的参数及简要功能

参数名	缺省值	简要说明
HURDLE	none	整数规划（IP）的“篱笆”值(none: 无, 可设定为任意实数值)
IPTOLA	.8e-7	整数规划（IP）的绝对最优性误差限
IPTOLR	.5e-7	整数规划（IP）的相对最优性误差限
TIM2RL	100	采用IPTOLR作为判断标准之前，程序必须求解的时间（秒）
NODESL	0	分枝节点的选择策略（0: LINGO自动选择；1: 深度优先；2: 最坏界的节点优先；3: 最好界的节点优先）

## SET 可设置的参数及简要功能

参数名	缺省值	简要说明
LENPAG	0	终端的页长限制 (0:没有限制; 可设定任意非负整数)
LINLEN	76	终端的行宽限制(0:没有限制; 可设定为64-200)
TERSEO	0	输出级别 (0:详细型, 1:简洁型)
STAWIN	1	是否显示状态窗口 (1:是, 0:否, Windows系统才能使用)
SPLASH	1	弹出版本和版权信息 (1:是, 0:否, Windows系统才能使用)



## SET 可设置的参数及简要功能

参数名	缺省值	简要说明
ORROUTE	0	将输出定向到命令窗口 (1:是, 0:否, Windows系统才能使用)
WNLINE	800	命令窗口的最大显示行数(Windows系统才能使用)
WNTRIM	400	每次从命令窗口滚动删除的最小行数(Windows系统才能使用)
STABAR	1	显示状态栏(1:是, 0:否, Windows系统才能使用)
FILFMT	1	文件格式(0:lng格式, 1:lg4格式, Windows系统才能使用)

# SET 可设置的参数及简要功能

参数名	缺省 值	简要说明
TOOLBR	1	显示工具栏(1:是, 0:否, Windows系统才能使用)
CHKDUP	0	检查数据与模型中变量是否重名 (1:是, 0:否)
ECHOIN	0	脚本命令反馈到命令窗口(1:是, 0:否)
ERRDLG	1	错误信息以对话框显示 (1:是, 0:否, Windows系统才能使用)
USEPNM	0	允许无限制地使用基本集合的成员名 (1:是, 0:否)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
NSTEEP	0	在非线性求解程序中使用最陡边策略选择变量(1:是, 0:否)
NCRASH	0	在非线性求解程序中使用启发式方法生成初始解(1:是, 0:否)
NSLPDR	1	在非线性求解程序中用SLP法寻找搜索方向 (1:是, 0:否)
SELCON	0	在非线性求解程序中有选择地检查约束(1:是, 0:否)
PRBLVL	0	对混合整数线性规划 (MILP) 模型, 采用探测 (Probing) 技术的级别(0:LINGO自动决定; 1:无; 2-7: 探测级别逐步升高)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
SOLVE	0	线性求解程序(0: LINGO自动选择, 1: 原始单纯形法, 2: 对偶单纯形法, 3: 障碍法 (即内点法))
REDUCE	2	模型降维(2:LINGO决定, 1:是, 0:否)
SCALEM	1	变换模型中的数据尺度 (1:是, 0:否)
PRIMPR	0	原始单纯形法决定出基变量的策略 (0: LINGO自动决定, 1: 对部分出基变量尝试, 2: 用最陡边法对所有变量进行尝试)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
DUALPR	0	对偶单纯形法决定出基变量的策略(0: LINGO自动决定, 1:按最大下降比例法确定, 2: 用最陡边法对所有变量进行尝试)
DUALCO	1	指定对偶计算的级别 (0: 不计算任何对偶信息; 1: 计算对偶价格; 2: 计算对偶价格并分析敏感性)
RCMPSN	0	Use RC format names for MPS I/O (1:yes, 0:no)
MREGEN	1	重新生成模型的频率 (0: 当模型的文本修改后;1: 当模型的文本修改或模型含有外部引用时; 3:每当有需要时)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
BRANDR	0	分枝时对变量取整的优先方向(0: LINGO自动决定;1:向上取整优先;2: 向下取整优先)
BRANPR	0	分枝时变量的优先级 (0:LINGO自动决定, 1:二进制 (0-1) 变量)
CUTOFF	.1e-8	解的截断误差限
STRONG	10	指定强分枝的层次级别
REOPTB	0	IP热启动时的LP算法 (0: LINGO自动选择; 1: 障碍法 (即内点法); 2: 原始单纯形法; 3: 对偶单纯形法)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
REOPTX	0	IP冷启动时的LP算法（选项同上）
MAXCTP	200	分枝中根节点增加割平面时，最大迭代检查的次数
RCTLIM	.75	割（平面）的个数相对于原问题的约束个数的上限（比值）
GUBCTS	1	是否使用广义上界（GUB）割（1:是, 0:否）
FLWCTS	1	是否使用流（Flow）割（1:是, 0:否）
LFTCTS	1	是否使用Lift割（1:是, 0:否）
PLOCTS	1	是否使用选址问题的割（1:是, 0:否）



# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
DISCTS	1	是否使用分解割 (1:是, 0:否)
KNPCTS	1	是否使用背包覆盖割 (1:是, 0:否)
LATCTS	1	是否使用格(Lattice)割 (1:是, 0:否)
GOMCTS	1	是否使用Gomory割(1:是, 0:否)
COFCTS	1	是否使用系数归约割 (1:是, 0:否)
GCDCTS	1	是否使用最大公因子割 (1:是, 0:否)
SCLRLM	1000	语法配色的最大行数 (仅Windows系统使用)
SCLRDL	0	语法配色的延时(秒) (仅Windows系统使用)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
PRNCLR	1	括号匹配配色 (1:是, 0:否, 仅 Windows 系统使用)
MULTIS	0	NLP 多点求解的次数 (0:无, 可设为任意非负整数)
USEQPR	0	是否识别二次规划 (1:是, 0:否)
GLOBAL	0	是否对NLP采用全局最优求解程序 (1:是, 0:否)
LNRISE	0	线性化级别 (0:LINGO自动决定, 1:无, 2:低, 3:高)
LNBIGM	100,000	线性化的大M系数

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
LNDLTA	.1e-5	线性化的Delta误差系数
BASCTS	0	是否使用基本（Basis）割 (1:是, 0:否)
MAXCTR	2	分枝中非根节点增加割平面时，最大迭代检查的次数
HUMNTM	0	分枝中每个节点使用启发式搜索的最小时间（秒）
DECOMP	0	是否使用矩阵分解技术 (1:是, 0:否)
GLBOPT	.1e-5	全局最优求解程序的最优性误差限

## SET 可设置的参数及简要功能

参数名	缺省值	简要说明
GLBDLT	.1e-6	全局最优求解程序在凸化过程中增加的约束的误差限
GLBVBD	.1e+11	全局最优求解程序中变量的上界
GLBUBD	2	全局最优求解程序中变量的上界的应用范围(0: 所有变量都不使用上界; 1: 所有变量都使用上界; 2:部分使用)
GLBBRN	5	全局最优求解程序中第1次对变量分枝时使用的分枝策略 (0: 绝对宽度; 1: 局部宽度; 2: 全局宽度; 3: 全局距离; 4: 绝对冲突; 5: 相对冲突)

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
GLBBXS	1	全局最优求解程序选择活跃分枝节点的方法（0：深度优先；1：具有最坏界的分枝优先）
GLBREF	3	全局最优求解程序中模型重整的级别（0：不进行重整；1：低；2：中；3：高）
SUBOUT	2	求解前对固定变量的归结、简化程度，相当于预处理程度（0：不归结；1：总是归结；2：在全局优化和多初值优化程序中不归结。）
NLPVER	0	非线性求解器的版本（0：系统自动选择；1：1.0版本；2：2.0版本）

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
DBGCLD	0	设置Debug调试命令冷启动时所使用的程序，有四种可能的设置：0：自动选择算法；1：原始单纯形法；2：对偶单纯形法；3：障碍法 (即内点法)。
DBGWRM	0	设置Debug调试命令热启动时所使用的程序，设置同上。
LCRASH	1	对非线性规划，使用启发式crashing技术（一种寻找初始解的技术）的程度：（0：不使用；1：低；2：高）。

# SET 可设置的参数及简要功能

参数名	缺省值	简要说明
BCROSS	1	使用内点法解线性规划时，是否将最后的最优解转化成基解（顶点解）的形式：（0：不转化；1:转化）。
LOWMEM	0	是否采用节省内存方式运行求解器：（0：不使用；1：使用）
FILOUT	0	当LINGO向电子表或数据库中输出数据时，如果电子表或数据库中用来接收数据的空间大于实际输出的数据占用的空间，是否对多余的表空间进行数据填充？（0：不填充；1：填充）。