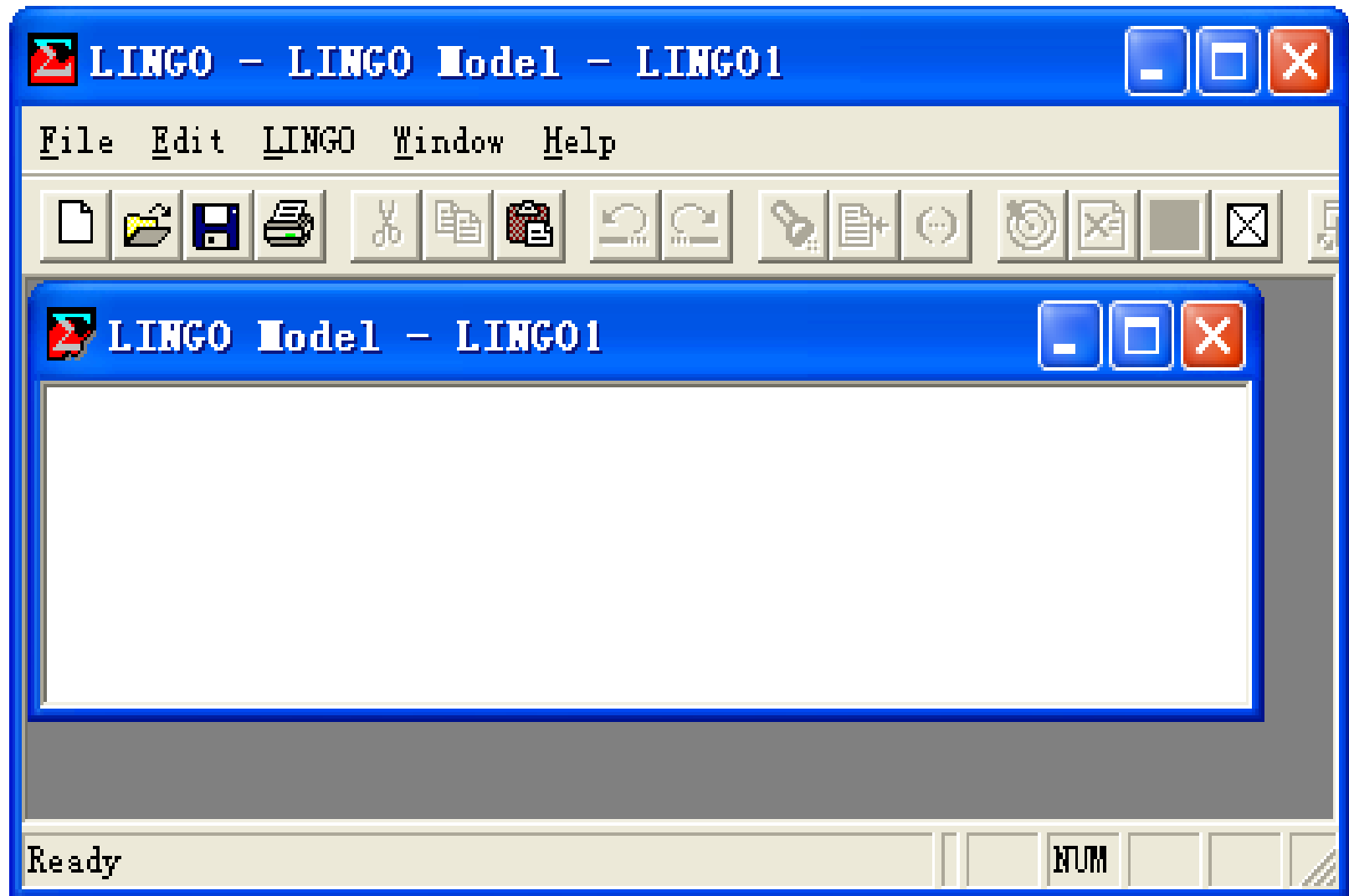


LINGO 教程

LINGO是用来求解线性和非线性优化问题的简易工具。LINGO内置了一种建立最优化模型的语言，可以简便地表达大规模问题，LINGO高效的求解器可快速求解并分析结果。

§1



外层是主框架窗口，包含了所有菜单命令和工具条，其它所有的窗口将被包含在主窗口之下。在主窗口内的标题为LINGO Model – LINGO1的窗口是LINGO的默认模型窗口，建立的模型都都要在该窗口内编码实现。下面举两个例子。

- 例1.1 如何在LINGO中求解如下的LP问题：

$$\min \quad 2x_1 + 3x_2$$

s.t.


$$x_1 + x_2 \geq 350$$

$$x_1 \geq 100$$

$$2x_1 + x_2 \leq 600$$

$$x_1, x_2 \geq 0$$

在模型窗口中输入如下代码：

- $\text{min} = 2 * x_1 + 3 * x_2;$
- $x_1 + x_2 \geq 350;$
- $x_1 \geq 100;$
- $2 * x_1 + x_2 \leq 600;$
- 然后点击工具条上的按钮  即可。

例1.2 使用LINGO软件计算6个发点8个收点的最小费用运输问题。产销单位运价如下表。

单 位 运 价 产地\销地	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇	B ₈	产量
A ₁	6	2	6	7	4	2	5	9	60
A ₂	4	9	5	3	8	5	8	2	55
A ₃	5	2	1	9	7	4	3	3	51
A ₄	7	6	7	3	9	2	7	1	43
A ₅	2	3	9	5	7	2	6	5	41
A ₆	5	5	2	2	8	1	4	3	52
销量	35	37	22	32	41	32	43	38	

使用LINGO软件， 编制程序如下：

- model:
- !6发点8收点运输问题;
- sets:
- warehouses/wh1..wh6/: capacity;
- vendors/v1..v8/: demand;
- links(warehouses,vendors): cost, volume;
- endsets
- !目标函数;
- min=@sum(links: cost*volume);
- !需求约束;
- @for(vendors(J):
- @sum(warehouses(I): volume(I,J))=demand(J));

!产量约束;

@for(warehouses(I):

@sum(vendors(J): volume(I,J))<=capacity(I));

- !这里是数据;
- data:
- capacity=60 55 51 43 41 52;
- demand=35 37 22 32 41 32 43 38;
- cost=6 2 6 7 4 2 9 5
- 4 9 5 3 8 5 8 2
- 5 2 1 9 7 4 3 3
- 7 6 7 3 9 2 7 1
- 2 3 9 5 7 2 6 5
- 5 5 2 2 8 1 4 3;
- enddata
- End
- 然后点击工具条上的按钮 即可

§2 LINGO中的集

- 对实际问题建模的时候，总会遇到一群或多群相联系的对象，比如工厂、消费者群体、交通工具和雇工等等。LINGO允许把这些相联系的对象聚合成**集**（sets）。一旦把对象聚合成集，就可以利用集来最大限度的发挥LINGO建模语言的优势。
- 现在我们将深入介绍如何创建集，并用数据初始化集的属性。学完本节后，你对基于建模技术的集如何引入模型会有一个基本的理解。

2.1 为什么使用集

- 集是LINGO建模语言的基础，是程序设计最强有力的基本构件。借助于集，能够用一个单一的、长的、简明的复合公式表示一系列相似的约束，从而可以快速方便地表达规模较大的模型。

2.2 什么是集

- 集是一群相联系的对象，这些对象也称为集的**成员**。一个集可能是一系列产品、卡车或雇员。每个集成一个可能有一个或多个与之有关联的特征，我们把这些特征称为**属性**。属性值可以预先给定，也可以是未知的，有待于LINGO求解。例如，产品集中的每个产品可以有一个价格属性；卡车集中的每辆卡车可以有一个牵引力属性；雇员集中的每位雇员可以有一个薪水属性，也可以有一个生日属性等等。
- LINGO有两种类型的集：**原始集**(primitive set)和**派生集**(derived set)。
- 一个原始集是由一些最基本的对象组成的。
- 一个派生集是用一个或多个其它集来定义的，也就是说，它的成员来自于其它已存在的集。

2.3 模型的集部分

- **集部分**是LINGO模型的一个可选部分。在LINGO模型中使用集之前，必须在集部分事先定义。集部分以关键字“sets:”开始，以“endsets”结束。一个模型可以没有集部分，或有一个简单的集部分，或多个集部分。一个集部分可以放置于模型的任何地方，但是一个集及其属性在模型约束中被引用之前必须定义了它们。

2.3.1 定义原始集

- 为了定义一个原始集，必须详细声明：
- 集的名字
- 可选， 集的成员
- 可选， 集成员的属性
- 定义一个原始集， 用下面的语法：
- `setname[/member_list/][:attribute_list];`
- *注意： 用“[]”表示该部分内容可选。 下同， 不再赘述。*

- Setname是你选择的来标记集的名字，最好具有较强的可读性。集名字必须严格符合标准命名规则：以拉丁字母或下划线（_）为首字符，其后由拉丁字母（A—Z）、下划线、阿拉伯数字（0，1，…，9）组成的总长度不超过32个字符的字符串，且不区分大小写。
- **注意：该命名规则同样适用于集成员名和属性名等的命名。**
- Member_list是集成员列表。如果集成员放在集定义中，那么对它们可采取显式罗列和隐式罗列两种方式。如果集成员不放在集定义中，那么可以在随后的数据部分定义它们。
- ① 当显式罗列成员时，必须为每个成员输入一个不同的名字，中间用空格或逗号隔开，允许混合使用。

例2.1 可以定义一个名为students的原始集，它具有成员John、Jill、Rose和Mike，属性有sex和age：

- sets:
- students/John Jill, Rose Mike/: sex, age;
- endsets
- ② 当隐式罗列成员时，不必罗列出每个集成员。可采用如下语法：
- setname/member1..memberN/[: attribute_list];
- 这里的member1是集的第一个成员名，memberN是集的最末一个成员名。LINGO将自动产生中间的所有成员名。LINGO也接受一些特定的首成员名和末成员名，用于创建一些特殊的集。

• 列表如下：

隐式成员列表格式	示例	所产生集成员
1..n	1..5	1, 2, 3, 4, 5
StringM.. <u>StringN</u>	Car2.. <u>car14</u>	Car2, Car3, Car4, ..., Car14
<u>DayM..DayN</u>	<u>Mon..Fri</u>	<u>Mon, Tue, Wed, Thu, Fri</u>
<u>MonthM..MonthN</u>	<u>Oct..Jan</u>	<u>Oct, Nov, Dec, Jan</u>
<u>MonthYearM..MonthYearN</u>	<u>Oct2001..Jan2002</u>	Oct2001, Nov2001, Dec2001, Jan2002

例2.2

- !集部分;
- sets:
 - students:sex,age;
- endsets
- !数据部分;
- data:
 - students,sex,age= John 1 16
 - Jill 0 14
 - Rose 0 17
 - Mike 1 13;
- enddata
- **注意：开头用感叹号 (!) ，末尾用分号 (;) 表示注释，可跨多行。**

- 在集部分只定义了一个集students，并未指定成员。在数据部分罗列了集成员John、Jill、Rose和Mike，并对属性sex和age分别给出了值。
- 集成员无论用何种字符标记，它的索引都是从1开始连续计数。在attribute_list可以指定一个或多个集成员的属性，属性之间必须用逗号隔开。
- 可以把集、集成员和集属性同C语言中的结构体作个类比。如下图：
- 集 \longleftrightarrow 结构体
- 集成员 \longleftrightarrow 结构体的域
- 集属性 \longleftrightarrow 结构体实例
- LINGO内置的建模语言是一种描述性语言，用它可以描述现实世界中的一些问题，然后再借助于LINGO求解器求解。因此，集属性的值一旦在模型中被确定，就不可能再更改。在LINGO中，只有在**初始部分**中给出的集属性值在以后的求解中可更改。这与前面并不矛盾，初始部分是LINGO求解器的需要，并不是描述问题所必须的。

2.3.2 定义派生集

- 为了定义一个派生集，必须详细声明：
- 集的名字
- **父集**的名字
- 可选，集成员
- 可选，集成员的属性
- 可用下面的语法定义一个派生集：
- `setname(parent_set_list)/member_list/[:attribute_list];`
- `setname`是集的名字。`parent_set_list`是已定义的集的列表，多个时必须用逗号隔开。如果没有指定成员列表，那么LINGO会自动创建父集成员的所有组合作为派生集的成员。派生集的父集既可以是原始集，也可以是其它的派生集。

§ 3 模型的数据部分和初始部分

说明：

处理模型的数据时，需要为集指派一些成员并且在求解模型之前为集的某些属性指定值。

两个可选部分：

输入集成员和数据的数据部分 (Data Section) ；

为决策变量设置初始值的初始部分 (Init Section) 。

3.1 模型的数据部分

3.1.1 数据部分入门

数据部分：提供了模型相对静止部分和数据分离的可能性，对模型维护和维数缩放非常便利。

以关键字 “data:” 开始，“enddata” 结束。可指定集成员、集的属性。

语法格式： object_list = value_list;

对象列 (object_list)：包含要指定值的属性名、要设置集成员的集名，用逗号或空格隔开。

一个对象列中至多有一个集名，属性名可有任意多。若对象列中有多个属性名，则其类型必须一致。若对象列中有一个集名，则对象列中所有的属性的类型就是这个集。

数值列 (value_list)：包含要分配给对象列中的对象的值，用逗号或空格隔开。注意属性值的个数必须等于集成员的个数。

```
sets:
```

```
    set1/A,B,C/: X,Y;
```

```
endsets
```

```
data:
```

```
    X=1,2,3;
```

```
    Y=4,5,6;
```

```
enddata
```

说明:

集set1中定义两个属性X、Y。

X的三个值是1、2、3，Y的三个值是4、5、6。

也可采用如下例子中的复合**数据声明**（data statement）实现同样的功能。

例3.2 `sets:`

```
set1/A,B,C/: X,Y;
```

```
endsets
```

```
data:
```

```
  X,Y=1  4
```

```
      2  5
```

```
      3  6;
```

```
enddata
```

说明：

误解，可能会认为X被指定了1、4、2三个值。

正确答案是1、2、3。

假设对象列有n个对象，在为对象指定值时，首先在n个对象的第1个索引处依次分配数值列中的前n个对象，然后在n个对象的第2个索引处依次分配数值列中紧接着的n个对象，……，以此类推。

模型的所有数据——属性值和集成员——被单独放在数据部分，这是最规范的数据输入方式。

3.1.2 参数

数据部分也可指定一些**标量变量** (scalar variables) 。

参数： 在数据部分确定的标量变量

举例： 假设模型中用利率8.5%作为一个参数，象下面一样输入一个利率作为参数。

例3.3

data:

```
interest_rate = .085;
```

enddata

也可同时指定多个参数。

例3.4

data:

```
interest_rate, inflation_rate = .085 .03;
```

enddata

3.1.3 实时数据处理

说明：

某些情况下，模型中的某些数据并不是定值。

如模型中有一通货膨胀率参数，想在2%至6%范围内，对不同的值求解模型，以观察模型的结果对通货膨胀的依赖有多么敏感。称这种情况称为**实时数据处理**（what if analysis）。

方法：

在本该放数的地方输入一个问号（?）。

例3.5

data:

```
interest_rate,inflation_rate = .085  ?;
```

enddata

每一次求解模型时，都会提示为参数inflation_rate输入一个值。

直接输入一个值再点击OK按钮，会把输入的值指定给inflation_rate，然后继续求解模型。

除参数之外，也可以输入集的属性值，但不许实时输入集成员名

3.1.4 指定属性为一个值

在数据声明右边输入一个值将所有成员的该属性指定为一个值。

例3.6

sets:

```
days /MO,TU,WE,TH,FR,SA,SU/:needs;
```

endsets

data:

```
needs = 20;
```

Enddata 用20指定days集的所有成员的needs属性。多个属性的情形，见下例。

例3.7

sets:

```
days /MO,TU,WE,TH,FR,SA,SU/:needs,cost;
```

endsets

data:

```
needs cost = 20 100;
```

enddata

3.1.5 数据部分的未知数值

有时只想为一个集的部分成员的某个属性指定值，让其余成员的该属性保持未知，以便让LINGO去求出它们的最优值。

在数据声明中输入两个相连的逗号表示该位置对应的集成员的属性值未知。两个逗号间可以有空格。

例3.8

```
sets:  
    years/1..5/: capacity;  
endsets  
  
data:  
    capacity = ,34,20,,;  
enddata
```

属性capacity的第2个和第3个值分别为34和20，其余的未知。

3.2 模型的初始部分

被求解器当作初始点来用，并且仅仅对非线性模型有用。

和数据部分指定变量值不同，求解器可以自由改变初始部分初始化的变量的值。

格式：以“init:”开始，以“endinit”结束。

说明：初始部分的初始声明规则和数据部分的数据声明规则相同。即可在声明左边同时初始化多个集属性，可把集属性初始化为一个值，可用问号实现实时数据处理，还可使用逗号指定未知数值。

例3.9

init:说明：提供的另一个可选部分。

初始部分中，可以输入**初始声明** (initialization statement)，和数据部分中数据声明相同。

实际问题建模时，初始部分并不起到描述模型的作用，初始部分输入的值仅

```
X, Y = 0, .1;
```

```
endinit
```

```
Y=@log(X);
```

$X^2 + Y^2 \leq 1$; 好的初始点会减少模型的求解时间。

§ 4 LINGO 函数

说明：

有了前几节的基础知识，再加上本节的内容，就能够借助于LINGO建立并求解复杂的优化模型。

函数类型（9种）：

1. 基本运算符：算术运算符、逻辑运算符、关系运算符
2. 数学函数：三角函数和常规的数学函数
3. 金融函数：两种金融函数
4. 概率函数：大量概率相关的函数
5. 变量界定函数：定义变量的取值范围
6. 集操作函数：对集的操作提供帮助
7. 集循环函数：遍历集的元素，执行一定的操作的函数
8. 数据输入输出函数：允许模型和外部数据源相联系，
进行数据的输入输出
9. 辅助函数：各种杂类函数