

# 数学建模算法与应用

## 附录B

### Lingo软件的使用



# B.1 Lingo软件的基本语法

## 1.集合

集合部分的语法为

sets:

集合名称 1/成员列表 1/: 属性 1\_1, 属性 1\_2, ..., 属性 1\_n1;

集合名称 2/成员列表 2/: 属性 2\_1, 属性 2\_2, ..., 属性 2\_n2;

派生集合名称(集合名称 1, 集合名称 2): 属性 3\_1, ..., 属性 3\_n3;

endsets

## 例 B.1

sets:

product/A B/;

machine/M N/;

week/1..2/;

allowed(product,machine,week):x;

endsets

## 2.数据

数据部分的语法为

**data:**

属性 1=数据列表;

属性 2=数据列表;

**enddata**

### 3.数据计算段

数据计算段部分不能含有变量，必须是已知数据的运算。

```
calc:
```

```
b=0;
```

```
a=a+1;
```

```
endcalc
```

## 4.变量的初始化

变量初始化主要用于非线性问题赋初始值。

例 B.2

init:

$X, Y = 0, .1;$

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。

## 5.模型的目标函数和约束条件

这里就不具体给出了，下面通过具体例子给出。

### B.1.6 实时数据处理

#### 例 B.3

data:

interest\_rate,inflation\_rate = .085 ?;

enddata

注：（1）Lingo 中是不区分大小写字符的。

（2）Lingo 中数据部分不能使用分式，例如数据部分不能使用  $1/3$ 。

（3）Lingo 中的注释是使用！引导的。

（4）Lingo 中默认所有的变量都是非负的。

（5）Lingo 中矩阵数据是逐行存储的，Matlab 中数据是逐列存储的。



## B.2 Lingo函数

### 1.算数运算符

^ 乘方

\* 乘

/ 除

+

加

- 减

## 2.逻辑运算符

在 Lingo 中，逻辑运算符主要用于集循环函数的条件表达式中，来控制在函数中哪些集成员被包含，哪些被排斥。在创建稀疏集时用在成员资格过滤器中。

Lingo 具有 9 种逻辑运算符

**#not#**      否定该操作数的逻辑值，**#not#** 是一个一元运算符。

**#eq#**      若两个运算数相等，则为 **true**；否则为 **false**。

**#ne#**      若两个运算数不相等，则为 **true**；否则为 **false**。

**#gt#**      若左边的运算数严格大于右边的运算数，则为 **true**；否则为 **false**。

**#ge#**      若左边的运算数大于或等于右边的运算数，则为 **true**；否则为 **false**。

**#lt#**      若左边的运算数严格小于右边的运算数，  
则为 **true**；否则为 **false**。

**#le#**      若左边的运算数小于或等于右边的运算  
数，则为 **true**；否则为 **false**。

**#and#**      仅当两个参数都为 **true** 时，结果为 **true**；  
否则为 **false**。

**#or#**      仅当两个参数都为 **false** 时，结果为 **false**  
否则为 **true**。

### 3.关系运算符

在 Lingo 中，关系运算符主要是被用在模型中来指定一个表达式的左边是否等于、小于等于、或者大于等于右边，形成模型的一个约束条件。关系运算符与逻辑运算符`#eq#`、`#le#`、`#ge#`截然不同，逻辑运算符仅仅判断一个关系是否被满足，满足为真，不满足为假。

Lingo 有三种关系运算符：“=”、“<=”和“>=”。

Lingo 中还能用“<”表示小于等于关系，“>”表示大于等于关系。Lingo 并不支持严格小于和严格大于关系运算符。

## 4.数学函数

Lingo 提供了大量的标准数学函数

@abs(x)返回  $x$  的绝对值。

@sin(x)返回  $x$  的正弦值， $x$  采用弧度制。

@cos(x)返回  $x$  的余弦值。

@tan(x)返回  $x$  的正切值。

@exp(x)返回常数  $e$  的  $x$  次幂。

@log(x)返回  $x$  的自然对数。

@lgm(x)返回  $x$  的 gamma 函数的自然对数。

**@mod(x,y)**返回  $x$  除以  $y$  的余数。

**@sign(x)**如果  $x < 0$  返回 -1；如果  $x > 0$  返回 1；如果  $x = 0$  返回 0。

**@floor(x)**返回  $x$  的整数部分。当  $x \geq 0$  时，返回不超过  $x$  的最大整数；当  $x < 0$  时，返回不低于  $x$  的最小整数。

**@smax(x1,x2,...,xn)**返回  $x_1, x_2, \dots, x_n$  中的最大值。

**@smin(x1,x2,...,xn)**返回  $x_1, x_2, \dots, x_n$  中的最小值。



## 5. 变量界定函数

变量界定函数实现对变量取值范围的附加限制，共 4 种

@bin(x)限制  $x$  为 0 或 1；

@bnd(L,x,U)限制  $L \leq x \leq U$ ；

@free(x)取消对变量  $x$  的默认下界为 0 的限制，即  $x$  可以取任意实数；

@gin(x)限制  $x$  为整数。

在默认情况下，Lingo 规定变量是非负的，也就是说下界为 0，上界为 $+\infty$ 。@free 取消了默认的下界为 0 的限制，使变量也可以取负值。@bnd 用于设定一个变量的上下界，它也可以取消默认下界为 0 的约束。

## 6. 集循环函数

@for: 该函数用来产生对集成员的约束。

@sum: 该函数返回遍历指定的集成员的一个表达式的和。

@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。

**例B.4** 求向量[5, 1, 3, 4, 6, 10]前5个数的最小值, 后3个数的最大值。

```

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
    
```

## 7. 概率函数

(1) @pbn(p,n,x)

二项分布的累积分布函数。当  $n$  和 (或)  $x$  不是整数时，用线性插值法进行计算。

(2) @pcx(n,x)

自由度为  $n$  的  $\chi^2$  分布的累积分布函数。

(3) @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。

(4) @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 繁忙概率。

(5) @pfd(n,d,x)

自由度为  $n$  和  $d$  的  $F$  分布的累积分布函数。

### (6) @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。

### (7) @phg(pop,g,n,x)

超几何(Hypergeometric)分布的累积分布函数。pop表示产品总数，g是正品数。从所有产品中任意取出n ( $n \leq \text{pop}$ ) 件。pop, g, n和x都可以是非整数，这时采用线性插值进行计算。

### (8) @ppl(a,x)

Poisson 分布的线性损失函数，即返回  $\max(0, z-x)$  的期望值，其中随机变量 z 服从均值为 a 的 Poisson 分布。

### (9) @pps(a,x)

均值为  $a$  的 Poisson 分布的累积分布函数。当  $x$  不是整数时，采用线性插值进行计算。

### (10) @psl(x)

单位正态线性损失函数，即返回  $\max(0, z-x)$  的期望值其中随机变量  $z$  服从标准正态分布。

### (11) @psn(x)

标准正态分布的累积分布函数。

### (12) @ptd(n,x)

自由度为  $n$  的  $t$  分布的累积分布函数。



### (13) @qrand(seed)

产生服从(0,1)区间的伪随机数。@qrand 只允许在模型的数据部分使用,它将用伪随机数填满集属性。通常,声明一个  $m \times n$  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。

## 例B.5

**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**

如果没有为函数指定种子 seed, 那么 Lingo 将用系统时间构造种子。

#### (14) @rand(seed)

返回 0 和 1 间的伪随机数, 依赖于指定的种子 seed  
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。

例B.6 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:

!产生一系列正态分布和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( .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

## 8. 集操作函数

Lingo 提供了几个函数帮助处理集。

(1) `@in(set_name, primitive_index_1  
[, primitive_index_2, ...])`

如果元素在指定集中，返回 1；否则返回 0。

**例B.7** 全集为 $I$ ， $B$ 是 $I$ 的一个子集， $C$ 是 $B$ 的补集。

**sets:**

$I/x1..x4/:x;$

$B(I)/x2/:y;$

$C(I)|\#not#\@in(B,&1):z;$

**endsets**

## (2) @index([set\_name,] primitive\_set\_element)

该函数返回在集 set\_name 中原始集成员 primitive\_set\_element 的索引。如果 set\_name 被忽略，那么 LINGO 将返回与 primitive\_set\_element 匹配的第一个原始集成员的索引。如果找不到，则产生一个错误。

**例B.8 如何确定集成员(B,Y)属于派生集S3。**

**sets:**

**S1/A B C/;**

**S2/X Y Z/;**

**S3(S1,S2)/A X, A Z, B Y, C X/;**

**endsets**

**L=@in(S3,@index(S1,B),@index(S2,Y));**

看下面的例子，表明有时为@index指定集是必要的。



## 例B.9

**sets:**

**girls/debble,sue,alice/;**

**boys/bob,joe,sue,fred/;**

**endsets**

**I1=@index(sue);**

**I2=@index(boys,sue);**

I1的值是2，I2的值是3。建议在使用@index函数时最好指定集。

### (3) @wrap(index,limit)

该函数返回  $j = \text{index} - k * \text{limit}$ ，其中  $k$  是一个整数，取适当值保证  $j$  落在区间  $[1, \text{limit}]$  内。该函数在循环、多阶段计划编制中特别有用。

### (4) @size(set\_name)

该函数返回集 `set_name` 的成员个数。在模型中明确给出集大小时最好使用该函数。它的使用使模型更加数据独立，集大小改变时也更易维护。

# B.3 数学规划模型举例

## 1.数据直接放在Lingo程序

例 B.10 使用 LINGO 软件计算 6 个产地 8 个销地的最小费用运输问题。单位商品运价如表 B.1 所示。

表 B.1 单位商品运价表

单位运价 产地 \ 销地	销地								产量
	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	
A <sub>1</sub>	6	2	6	7	4	2	5	9	60
A <sub>2</sub>	4	9	5	3	8	5	8	2	55
A <sub>3</sub>	5	2	1	9	7	4	3	3	51
A <sub>4</sub>	7	6	7	3	9	2	7	1	43
A <sub>5</sub>	2	3	9	5	7	2	6	5	41
A <sub>6</sub>	5	5	2	2	8	1	4	3	52
销量	35	37	22	32	41	32	43	38	

解 设  $x_{ij} (i = 1, 2, \dots, 6; j = 1, 2, \dots, 8)$  表示产地  $A_i$  运到销地  $B_j$  的量,  $c_{ij}$  表示产地  $A_i$  到销地  $B_j$  的单位运价,  $d_j$  表示销地  $B_j$  的需求量,  $e_i$  表示产地  $A_i$  的产量, 建立如下线性规划模型

$$\begin{aligned} & \min \sum_{i=1}^6 \sum_{j=1}^8 c_{ij} x_{ij}, \\ \text{s.t. } & \begin{cases} \sum_{i=1}^6 x_{ij} = d_j, & j = 1, 2, \dots, 8, \\ \sum_{j=1}^8 x_{ij} \leq e_i, & i = 1, 2, \dots, 6, \\ x_{ij} \geq 0, & i = 1, 2, \dots, 6; j = 1, 2, \dots, 8. \end{cases} \end{aligned}$$

使用 Lingo 软件，编制程序如下

```

model:
!6产地8销地运输问题;
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));
!下面是数据;
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
    
```

## 2.使用纯文本文件传递数据

使用 Lingo 函数@file, @text 进行纯文本文件数据的输入和输出。

注：执行一次@file, 输入 1 个记录, 记录之间的分隔符为~。

例 B.11(续例 B.10) 通过纯文本文件传递数据。

Lingo程序如下

```

model:
sets:
    warehouses/wh1..wh6/: capacity;
    vendors/v1..v8/: demand;
    links(warehouses,vendors): cost, volume;
endsets
    min=@sum(links: cost*volume);
    @for(vendors(J): @sum(warehouses(I):
volume(I,J)=demand(J));
    @for(warehouses(I): @sum(vendors(J):
volume(I,J))<=capacity(I));
    data:
        capacity=@file(dataB_11.txt);
        demand=@file(dataB_11.txt);
        cost=@file(dataB_11.txt);
    enddata
end
    
```



其中纯文本数据文件 dataB\_11.txt 中的数据格式如下

60 55 51 43 41 52~ !~是记录分割符，该

第一个记录是产量；

35 37 22 32 41 32 43 38~ !该第二个记录是  
需求量；

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

!最后一个记录是单位运价；

### 3.使用Excel文件传递数据

Lingo 通过@OLE 函数实现与 Excel 文件传递数据，使用@OLE 函数既可以从 Excel 文件中导入数据，也能把计算结果写入 Excel 文件。

@OLE 函数只能用在模型的集合定义段、数据段和初始段。从 Excel 文件中导入数据的使用格式可以分成以下几种类型

(1) 变量名 1, 变量名 2=@OLE ('文件名', '数据块名称 1', '数据块名称 2');

若变量是初始集合的属性, 则对应的数据块应当是一列数据, 若变量是二维派生集合的属性, 则对应数据块应当是二维矩形数据区域。@OLE 函数无法读取三维数据区域。

(2) 变量名 1, 变量名 2=@OLE ('文件名', '数据块名称');

左边的两个变量必须定义在同一个集合中，@OLE 的参数仅指定一个数据块名称，该数据块应当包含类型相同的两列数据，第一列赋值给变量 1，第二列赋值给变量 2。

(3) 变量名 1, 变量名 2=@OLE ('文件名');  
没有指定数据块名称, 默认使用 Excel 文件中与  
属性名同名的数据块。

使用@OLE 函数也能把计算结果写入 Excel 文件, 使用格式有以下三种

(1) @OLE ('文件名', '数据块名称 1', '数据块名称 2') = 变量名 1, 变量名 2;

将两个变量的内容分别写入指定文件的两个预先已经定义了名称的数据块，数据块的长度（大小）不应小于变量所包含的数据，如果数据块原来有数据，则@OLE 写入语句运行后原来的数据将被新的数据覆盖。

(2) @OLE ('文件名', '数据块名称') = 变量名 1, 变量名 2;

两个变量的数据写入同一数据块 (不止 1 列), 先写变量 1, 变量 2 写入另外 1 列。

(3) @OLE ('文件名') = 变量名 1, 变量名 2;

不指定数据块的名称，默认使用 Excel 文件中与变量名同名的数据块。

例 B.12(续例 B.10) 通过 Excel 文件传递数据。



Lingo程序如下：

**model:**

**sets:**

**warehouses/wh1..wh6/: capacity;**

**vendors/v1..v8/: demand;**

**links(warehouses,vendors): cost, volume;**

**endsets**

**min=@sum(links: cost\*volume);**

**@for(vendors(J):@sum(warehouses(I): volume(I,J))=demand(J));**

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

**data:**

**capacity=@ole(.\dataB\_12.xlsx);**

**demand=@ole(.\dataB\_12.xlsx);**

**cost=@ole(.\dataB\_12.xlsx);**

**@ole(.\dataB\_12.xlsx)=volume;**

**enddata**

**end**

注：（1）Excel 中数据块的命名，具体做法是先用鼠标选中数据区域，从菜单上选择“插入”→“名称”→“定义”，弹出“定义名称”对话框，输入适当的名称，然后点击确定。

（2）建议把所有的数据文件和程序文件放在同一个目录下，如果运行时找不到要打开的文件，请核对文件名是否正确，如果文件名无误，仍然找不到文件，则是由于没有用 Excel 打开所操作的数据文件。

(3) 在新版 office 下，必须指明 Excel 文件的路径，程序中的“.”表示当前路径，必须指明，否则找不到 Excel 文件。

## 4. Lingo与数据库的接口

数据库管理系统（data base management system, DBMS）在数据库建立、运行和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在故障发生后对系统进行恢复，它是处理大规模数据的最好工具，许多部门的业务数据大多保存在数据库中。开放式数据库连接（open data base connectivity, ODBC）为 DBMS 定义了一个标准化接口，其它软件可以通过这个接口访问任何 ODBC 支持的数据库。

LINGO 为 Access、DBase、Excel、FoxPro、Oracle Paradox、SQL Sever 和 Text Files 安装了驱动程序，能与这些类型的数据库文件交换数据。

Lingo 提供的名为@ODBC 函数能够实现从 ODBC 数据源导出数据或将计算结果导入 ODBC 数据源中。

为了使 Lingo 模型在运行时能够自动找到 ODBC 数据源并正确赋值，必须满足以下三个条件

(1) 将数据源文件在 Windows 的 ODBC 数据源管理器中进行注册；

(2) 注册的用户数据源名称与 LINGO 模型的标题相同；

(3) 对于模型中的每一条@ODBC 语句，数据源文件中存在与之相对应的表项。

## 5. 钢管订购和运输问题的Lingo求解

例 B.13 用 Lingo 软件求解第 4 章的钢管订购和运输问题。

使用 Lingo 求解时,需要对非线性约束条件(4.27)进行处理。引进0-1变量

$$t_i = \begin{cases} 1, & \text{钢厂} i \text{ 生产,} \\ 0, & \text{钢厂} i \text{ 不生产,} \end{cases} \quad i = 1, 2, \dots, 7.$$

把约束条件 (4.27) 转化为线性约束

$$500t_i \leq \sum_{j=1}^{15} x_{ij} \leq s_i t_i, \quad i = 1, 2, \dots, 7.$$



利用 Lingo 软件求得总费用的最小值为 127.8632 亿。Lingo 程序如下：

本程序语句过多，请参考程序文件夹中源程序。