第 4 章　优化建模与 LINGO 软件

在工程技术、经济管理、科学研究、生产实践等领域，经常会遇到如何利用现有资源进行决策，以实现最大效益，这类问题统称为最优化问题。最优化问题在数学建模竞赛（乙组）中也占据着极其重要的地位，基本每年都有一个赛题涉及到最优化问题。本章将重点介绍如何建立优化模型和如何用 LINGO 软件求解模型。

4.1　优化模型概述

在人们的日常生产实践中，常常面临这样的问题：在一系列客观或主观限制的条件下，寻求使所关注的某个或多个指标达到最小（或最大）的决策，这类决策问题统称为最优化（简称优化）问题。优化问题的求解需先建立优化模型，然后借助软件（如 LINGO 或 MatLab 等）求解。

优化模型由以下三个要素构成：

（1）决策变量：是指所求解问题中的未知量，若有 k 个未知量，常用 k 维向量 $x=(x_1,x_2,\cdots,x_k)^T$ 来表示。

（2）目标函数：是决策变量 x 的函数，常用 $f(x)$ 表示，在优化问题中，都是求目标函数的最大或最小值。

（3）约束条件：是对决策变量的限制的表达式。可以是等式约束，也可是不等式约束。

综上所述，优化模型的一般形式为：

$$
\begin{aligned}
&\text{opt}\quad f(x)\\
&s.t.\quad g_i(x)=0\qquad (i=1,2,\cdots,m)\\
&\qquad\ \ h_j(x)\le 0\qquad (j=1,2,\cdots,n)
\end{aligned}
$$

这里的 opt 是最优化的意思，是 max（极大）和 min（极小）两者之一。*s.t.* 是受约束于的意思，其后面的内容都是约束条件。

根据决策变量、目标函数和约束条件的不同类型，优化模型的分类也不同，常见的分类如下：

（1）按目标函数的个数分类：单目标规划模型、多目标规划模型。

（2）按模型中 $f(x)$，$g_i(x)$ 和 $h_j(x)$ 函数类型分类：线性规划、二次规划、非线性规划等。

（3）按决策变量的类型分类：连续优化模型、整数规划模型、0-1 型整数

规划模型。

【例 4-1】 奶制品的生产计划

问题重述 一个奶制品加工厂用牛奶可加工生产 *A* 和 *B* 两种奶制品，加工工艺及利润说明见表 4-1

表 4-1 *A* 和 *B* 两种奶制品的加工工艺及利润说明

奶制品	牛奶原料	设备	耗时(小时)	重量 (kg)	获利 (元/kg)
A	1 桶	甲	12	3	24
B	1 桶	乙	8	4	16

现每天有 50 桶牛奶原料供应，每天正式工人总的劳动时间为 480 小时，且甲设置每天最多能加工 100kg 的 *A* 奶制品，假设 *A* 和 *B* 奶制品均能全部售完。

问题：如何安排生产计划，使每天获利最多？

问题分析 这个优化问题的目标是使获利最多，决策是生产计划的安排，即每天用多少桶牛奶来加工生产 A，多少桶加工生产 B，决策受到 3 个条件限制：原料、劳动时间、设备甲的加工能力。按题目要求，将决策变量、目标函数和约束条件用数学符号及式子表示，得到下面的模型。

模型的建立

决策变量：设每天用 x_1 桶牛奶加工生产 A，用 x_2 桶牛奶加工生产 B。

目标函数：设每天获利为 z 元，x_1 桶牛奶加工生产 $3x_1$ kgA，获利 $3x_1 \times 24$ 元，x_2 桶牛奶加工生产 $4x_2$ kgB，获利 $4x_2 \times 16$ 元，故 $z = 72x_1 + 64x_2$。

约束条件：

原料约束 加工生产 A 和 B 的牛奶原料总量不得超过每天的供应量，即 $x_1 + x_2 \le 50$。

劳动时间 加工生产 A 和 B 的总劳动时间不能超过正式工人总的劳动时间，即 $12x_1 + 8x_2 \le 480$。

设备能力 加工生产 A 的产量不能超过设备甲每天的加工能力，即 $3x_1 \le 100$。

非负约束　x_1 和 x_2 均不能取负取，即 $x_1 \ge 0$，$x_2 \ge 0$。

综上可得

$$
\begin{aligned}
\max \quad & z = 72x_1 + 64x_2 \\
s.t. \quad & x_1 + x_2 \le 50 \\
& 12x_1 + 8x_2 \le 480 \\
& 3x_1 \le 100 \\
& x_1 \ge 0,\ x_2 \ge 0
\end{aligned}
$$

这就是该问题的模型。由于目标函数和约束条件对决策变量而言都是线性的，所以称为线性规划模型，简记 LP。该模型的 LINGO 求解见下节内容。

4.2　LINGO 入门

4.2.1　LINGO 软件概述

1. LINGO 软件介绍

LINGO 的全称为 Linear Interactive and General Optimizer（交互式的线性和通用优化求解器），是美国芝加哥（Chicago）大学的 Linus Schrage 教授于 1980 年前后开发的，由他成立的 LINDO 系统推出面世。LINGO 用于求解线性规划和非线性规划问题，被广泛用于教育、科研、经济管理和工业等领域。

LINGO 是求解优化模型的最佳选择，主要的特点有：

① 既能求线性规划问题，也可求非线性规划问题；
② 模型的输入简单直观；
③ 执行速度快，计算能力强；
④ 内置建模语言，提供几十个内部函数，方便灵活；
⑤ 方便与 Excel 文件、文本文档进行数据传递。

2. LINGO 界面介绍

双击打开 LINGO 软件就出现 LINGO 操作的初始界面，包括菜单栏、

工具栏和模型窗口三部分（见图 4-1）。光标所在的子窗口称为模型窗口，是用来输入 LINGO 程序的。

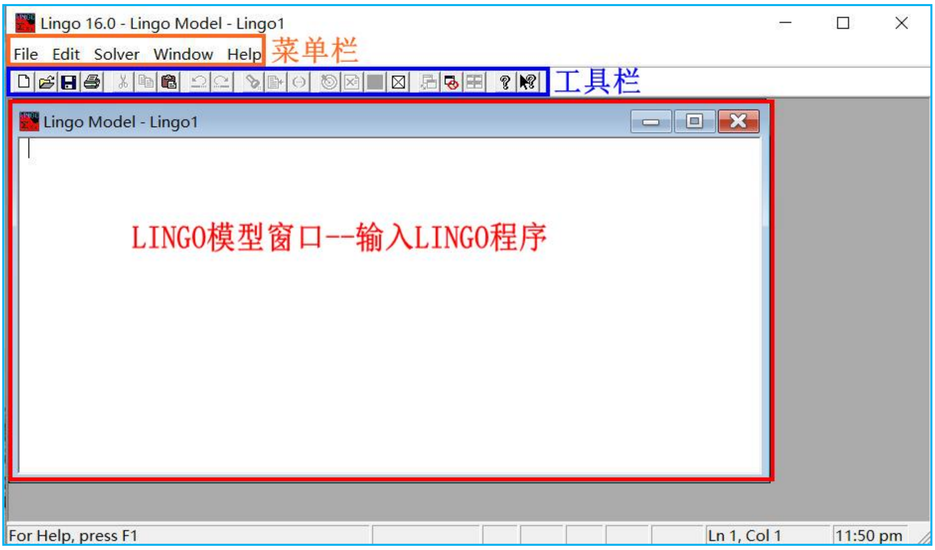


图 4-1　LINGO 初始界面

菜单栏中的 5 个窗口分别为 File（文件）、Ediit（编辑）、Solver（求解）、Window（窗口）和 Help（帮助），这里就不再介绍这些菜单下的目录。为了快速掌握 LINGO 的操作，介绍工具栏中常用的工具（见图 4-2）。

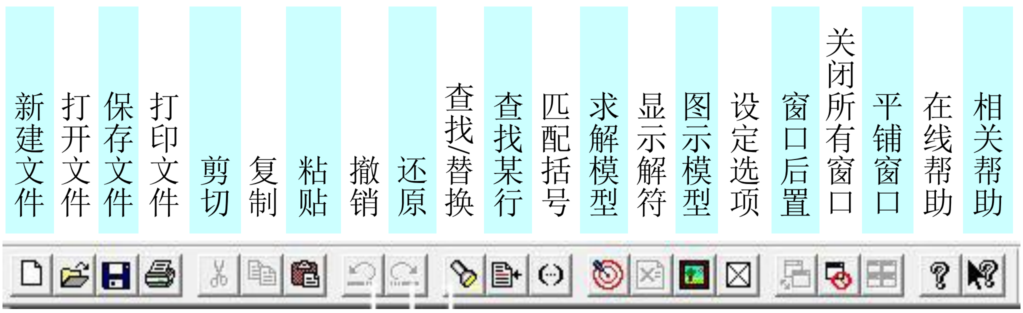


图 4-2　工具栏说明

4.2.2　LINGO 语言规则

LINGO 程序语言的输入和修改都是在模型窗口进行的，程序语言输入简单直观。

下面以 4.1 节的【例 4-1】奶制品的生产计划模型为例，在 LINGO 模型窗口输入该程序，见图 4-3.

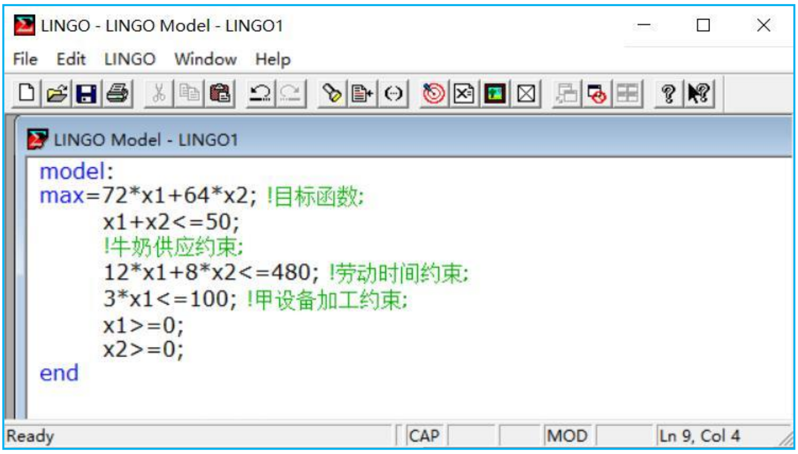


图 4-3 【例 4-1】奶制品的生产计划模型的 LINGO 程序

从图 4-3 中可以看出，LINGO 程序语句的输入与原模型基本相同，不同之处在于：

（1）目标函数用“max=”表示；

（2）模型中的约束条件“s.t.”在 LINGO 程序不需要写。

（3）LINGO 程序以以“model:”开始，“end”结束，目标函数和约束条件都放在这两个语言之间。

通过原模型和 LINGO 程序对比发现，LINGO 程序语句简单直观，基本与原模型保持一致，下面着重介绍 LINGO 程序语句的规则。

1. LINGO 程序的基本规则

LINGO 程序由一系列语句组成的，即语句是 LINGO 程序的基本单位。LINGO 程序输入基本规则包括以下几点：

（1） LINGO 程序语言不区分大小写字母的，即“MAX”、“max”和“Max”都是一样的。

（2） LINGO 每个程序以“model:”开始，“end”结束，且各自独占一行。LINGO 程序语句写在两者之间。对简单的模型程序，这两个语句可省略。

（3） LINGO 程序语句的顺序没有严格要求，总是以“max=”或“min=”语句来寻找目标函数，其他语句都是约束条件。

（4） LINGO 程序的每一个语句必须以分号“;”结束。一行允许写多个语句。

（5） 叹号“！”是注释符，其后的内容为解释，可以写在语句后面，也可以另写一行，且必须以以分号“;”(英文状态输入的分号）结束，允许跨行。
（6） LINGO 程序中都是用“@”开关调用内部函数。

注意：为保证模型程序的可读性，建议一行只写一个语句。按语句之间的嵌套关系对语句安排适当的缩进，以增强程序的层次感。

2. LINGO 变量

在 LINGO 程序中变量占据着重要地位，变量的基本使用规则包括：
（1） LINGO 变量的命名规则：只能由字母、数字、和下划线构成，长度不能超过 32 个字符，且必须以字母或下划线开头。如：x1，_y，_1，z_1 都是合法的变量。
（2） 在 LINGO 中，所有的变量默认都是非负的实数，不需特别说明。
（3） 变量的特殊声明：

```
@gin(x);   !变量 x 为整数;

@bin(x1);   !0-1 整数变量，变量 x1 为 0 或 1;

@free(y);   !取消 y 的非负限制;

@bnd(l,x,u);   !限制变量 l≤x≤u ;
```

注意：变量声明中，每个变量声明命令只能直接声明一个变量，若多个变量则需分开声明，除非采用循环命令，后面章节中会有相应的介绍。

3. LINGO 运算符及优先级

LINGO 运算符包括算术运算符、逻辑运算符和关系运算符三种，其 LINGO 表达调用格式见表 4-2，表 4-3，表 4-4.

表 4-2 LINGO 算术运算符

数学符号名称	LINGO 表达格式	调用格式示例
加法 ＋	+	2+5
减法（负号） －	-	2-5

乘法　×	*	2*5
除法　÷	/	2/5
求幂 2^5	^	2^5

注意：

（1）在 LINGO 中，任何运算符都不能省略，特别是乘号，很多同学会习惯性的按手写规则省略不写，省略后会出错。

（2）在 LINGO 中，没有专门的开方运算符，需将开方转化成幂计算。例如：数学计算公式 $\sqrt[3]{5}$，化成幂 $5^{\frac{1}{3}}$，则 LINGO 表示为：5^(1/3)

（3）LINGO 算术运算的优先级与手写演算一样：有括号先算括号，先幂、再乘除、后加减。

逻辑运算符运算结果只有“真”和“假”两个逻辑值，在 LINGO 中用数字 1 代表“TRUE”，其他值（常用 0）表示“FALSE”。LINGO 逻辑运算符有两类共 9 个，见表 4-3.

表 4-3　LINGO 逻辑运算符

LINGO 逻辑运算符	意义	归类
#and#	与	逻辑值或逻辑表达式之间的运算，运算结果还是逻辑值。如 1#and#0 的结果是 0（假）
#or#	或	
#not#	非	
#eq#	等于	两个数（或相应表达式）之间的比较，结算结果是逻辑值。如：2#gt#1 的结果是 1（真）
#ne#	不等于	
#gt#	大于	
#ge#	大于等于	
#lt#	小于	
#le#	小于等于	

关系运算符表示“数与数之间”的大小关系，常用于模型的约束条件，LINGO 有 3 种关系运算符，分别为：

＞（即<=，小于等于）、=（等于）、＞（即>=，大于等于）

一般地，LINGO 的约束没有严格小于和严格大于关系。

LINGO 运算符的优先级如表 4-4 所示。

表 4-4　LINGO 运算符的优先级

优先级	运算符
最高	#not#、-（负号）
	^
｜	*、/
｜	+、-（减法）
↓	#eq#、#ne#、#gt#、#ge#、#lt#、#le#
	#and#、#or#
最低	>、=、<

有一优先级按从左到右的顺序运算，如果有括号“（）”，则优先计算括号中的表达式。

4. 常用的 LINGO 函数

LINGO 包含很多内部的数学函数,这些函数的用法很简单,都是以“@”开头调用的。常用的 LINGO 函数调用格式见表 4-5.

表 4-5　LINGO 常用的数学函数

LINGO 函数	意义
@abs(x)	绝对值函数，返回 x 的绝对值
@sin(x)	正弦函数，返回 x 的正弦值，x 的单位是弧度
@cos(x)	余弦函数，返回 x 的余弦值，x 的单位是弧度
@pow(x,y)	指数函数，返回 x^y 的值
@exp(x)	以 e（≈2.718）为底的指数函数，返回 e^x 的值
@log(x)	自然对数函数，返回 $\log_e x$ 的值
@floor(x)	取整函数，返回 x 的整数部分（靠 0 取整）
@mod(x,y)	模函数，返回 x 除以 y 的余数（x 与 y 为整数）
@smax(list)	最大值函数，返回一列数（list）中的最大值
@smin(list)	最小值函数，返回一列数（list）中的最小值
@size(list)	元素个数，返回一列数（list）中的元素个数

5. LINGO 程序的运算结果分析

以 4.1 节的【例 4-1】奶制品的生产计划模型为例，在 LINGO 模型窗口输入该程序（见图 4-3）后，单击求解模型键 （见图 4-2）弹出求解器状态窗口如图 4-4 所示，当求解器状态运行完成时，运行结果就会显示在报告窗口中，如图 4-5 所示。

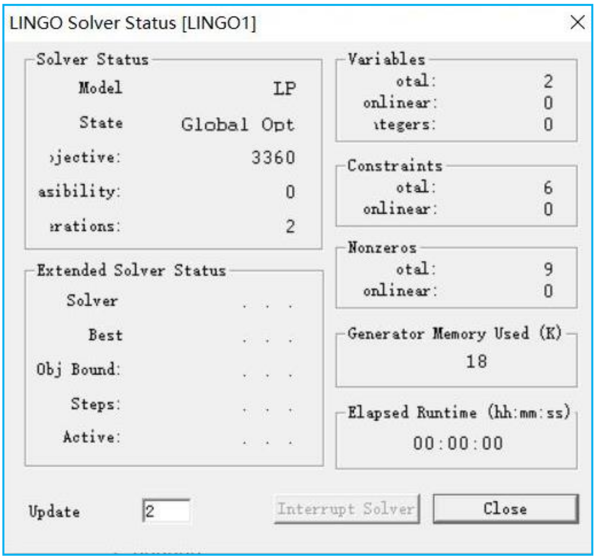
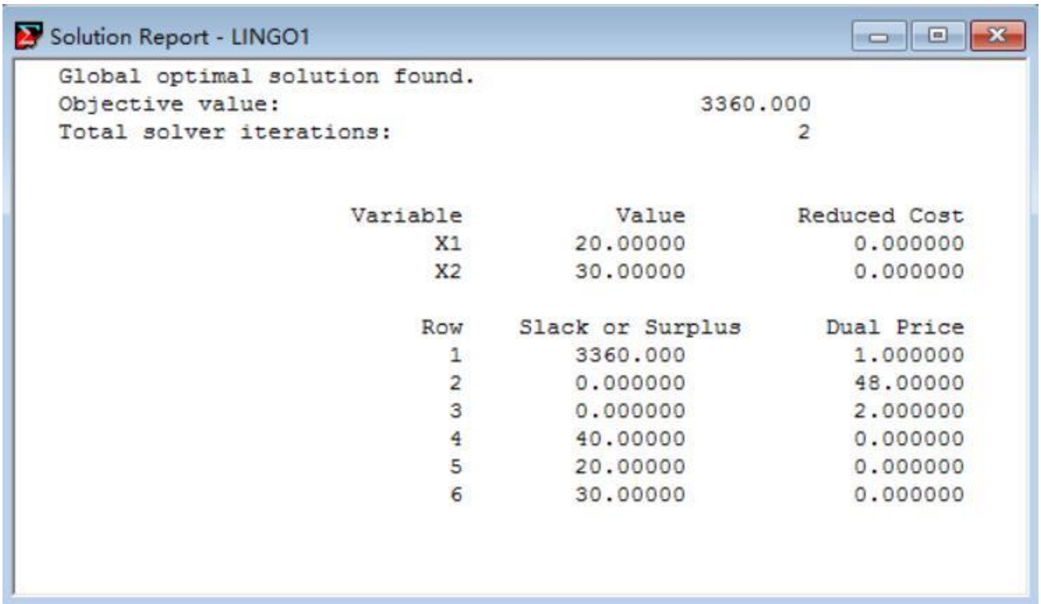


图 4-4 【例 4-1】LINGO 程序的求解器状态窗口


Solution Report - LINGO1

```
Global optimal solution found.
Objective value:                              3360.000
Total solver iterations:                             2


                  Variable           Value        Reduced Cost
                        X1        20.00000            0.000000
                        X2        30.00000            0.000000

                       Row    Slack or Surplus      Dual Price
                         1        3360.000            1.000000
                         2        0.000000            48.00000
                         3        0.000000            2.000000
                         4        40.00000            0.000000
                         5        20.00000            0.000000
                         6        30.00000            0.000000
```

图 4-5 【例 4-1】LINGO 程序的运行结果报告窗口

LINGO 报告窗口的解读：当 $x_1=20, x_2=30$ 时，目标函数最大值为 3360.

【例 4-2】 钢管下料问题

问题重述 从钢管厂进货的原料钢管都是 19m 长，某钢管零售商将钢管按顾客要求切割后售出。有一客户需要 50 根 4m 长、20 根 6m 长和 15 根 8m 长的钢管。

问题：如何下料最节省？

问题分析 首先，19m 长的原料钢管在切割 4m 或 6m 或 8m 长时，为达到下料最节省，余料应该小于客户所需的最小尺寸。其次，在满足前面的条件下，19m 长的原料钢管如何切割？如 19m=4*4m+3m，或 19m=3*6m+1m，或 19m=3*4m+6m+1m 等切割模式。所有的切割模式检有 7 种，如表 4-6 所示。

表 4-6 合理的钢管切割模式

切割模式	4m 的根数	6m 的根数	8m 的根数	余料 (m)
1	4	0	0	3
2	3	1	0	1
3	1	2	0	3
4	2	0	1	3
5	1	1	1	1
6	0	3	0	1
7	0	0	2	3

本问题转化为选择哪些切割模式，每种切割模式分别切割多少根原料钢管，使得切割原料钢管的总根数最少。

模型建立

决策变量：用 x_i 表示按照第 $i(i=1,2,\cdots,7)$ 种切割模式切割原料钢管的根数，显然 x_i 为非负整数。

目标函数：切割原料钢管的总根数最少，即

$$\min \quad Z=x_1+x_2+x_3+x_4+x_5+x_6+x_7$$

约束条件：所切割的各尺寸钢管不得少于客户需求，则有

4m 钢管需求约束： $4\times x_1+3\times x_2+1\times x_3+2\times x_4+1\times x_5+0\times x_6+0\times x_7\geq 50$

6m 钢管需求约束：$0\times x_1+1\times x_2+2\times x_3+0\times x_4+1\times x_5+3\times x_6+0\times x_7\geq 20$

8m 钢管需求约束：$0\times x_1+0\times x_2+0\times x_3+1\times x_4+1\times x_5+0\times x_6+2\times x_7\geq 15$

综上可得

$$\min \quad Z=x_1+x_2+x_3+x_4+x_5+x_6+x_7$$

$$s.t.\quad 4x_1+3x_2+x_3+2x_4+x_5\geq 50$$

$$x_2+2x_3+x_5+3x_6\geq 20$$

$$x_4+x_5+2x_7\geq 15$$

$$x_1,x_2,x_3,x_4,x_5,x_6,x_7\geq 0$$

该问题的模型为整数线性规划模型。

模型求解

上述模型用 LINGO 软件输入如下：

```
model:
min=x1+x2+x3+x4+x5+x6+x7; !目标函数;
        4*x1+3*x2+x3+2*x4+x5>=50;!4m 钢管需求约束;
        x2+2*x3+x5+3*x6>=20;!6m 钢管需求约束;
        x4+x5+2*x7>=15;!8m 钢管需求约束;
        @gin(x1);!x1 整数约束;
        @gin(x2);!x2 整数约束;
        @gin(x3);!x3 整数约束;
        @gin(x4);!x4 整数约束;
        @gin(x5);!x5 整数约束;
        @gin(x6);!x6 整数约束;
        @gin(x7);!x7 整数约束;
end
```

单击求解模型键 ，运行结果报告如下：

```
Global optimal solution found.
Objective value:                              25.00000
                    Variable           Value        Reduced Cost
```

X1	0.000000	0.3333333
X2	15.00000	0.1666667
X3	0.000000	0.1666667
X4	0.000000	0.1666667
X5	5.000000	0.000000
X6	0.000000	0.000000
X7	5.000000	0.000000

即按第 2 种切割模式切割 15 根原料钢管，按第 5 种切割模式切割 5 根原料钢管，按第 7 种切割模式切割 5 根原料钢管，共 25 根原料钢管。

【思考】切割模式越多，导致生产过程越复杂，从而增加生产和管理成本，现规定不同的切割模式不能超过 3 种。客户的需求为：50 根 4m 长，10 根 5m 长，20 根 6m 长，15 要 8m 长。

【例 4-3】 旅行包优化模型

问题重述 某人要外出旅行，需带 10 种物品，但是只有一个容积为 20L。所有物品均可在目的地购买，且通过网上查询得知这 10 件物品的当地价格。每件物品的体积及价格见表 4-7

表 4-7 物品的体积（单位：L）和价格（单位：元）

物品	1	2	3	4	5	6	7	8	9	10
体积	2	3	5	4	3	1	8	4	7	6
价格	20	42	85	60	50	18	170	90	100	70

问题：设计一个最省钱的装包方案。

问题分析 最省钱装包就是到达目的地后购买物品的最价值最低，即旅行包中所装物品的价值之和最高。由于每件物品只有两个结果，要么装进旅行包，要么不装包，所以可用 0-1 变量表示物品装包情况，从而建立 0-1 规划模型。

符号说明

x_i 0-1 变量，表示第 i 个物品的装包情况，$i=1,2,\cdots,10$。

p_i 表示第 i 件物品的价格，$i=1,2,\cdots,10$。

v_i 表示第 i 件物品的体积，$i=1,2,\cdots,10$。

V 表示旅行包的容积。

模型建立

决策变量： $x_i=\begin{cases}0, & 表示第i件物品不装包\\ 1, & 表示第i件物品装包\end{cases}$

目标函数：装包物品的总价值

$$Z=\sum_{i=1}^{10}x_ip_i=x_1p_1+x_2p_2+x_3p_3+x_4p_4+x_5p_5+x_6p_6+x_7p_7+x_8p_8+x_9p_9+x_{10}p_{10}$$

约束条件：装包物品的体积之和不能超过旅行包的容量，即 $\sum_{i=1}^{10}x_iv_i\le V$

综上可得

$$\max \quad Z=\sum_{i=1}^{10}x_ip_i$$

$$s.t. \quad \sum_{i=1}^{10}x_iv_i\le V$$

$$x_i=0 \text{ 或 } x_i=1，\quad i=1,2,\cdots,10$$

该问题的模型为 0-1 线性规划模型。

模型求解

上述模型用 LINGO 软件输入如下：

```
model:
max=x1*20+x2*42+x3*85+x4*60+x5*50+x6*18+x7*170+x8*90+x9*100+x10*
70; !目标函数;
x1*2+x2*3+x3*5+x4*4+x5*3+x6*1+x7*8+x8*4+x9*7+x10*6<=20;!约束条件;
    @bin(x1);!0-1 约束;
    @bin(x2);
    @bin(x3);
    @bin(x4);
    @bin(x5);
    @bin(x6);
    @bin(x7);
    @bin(x8);
    @bin(x9);
    @bin(x10);
end
```

单击求解模型键 ，运行结果报告如下：

```
Global optimal solution found.
Objective value:                    395.0000
Variable           Value          Reduced Cost
   X1            0.000000          -20.00000
   X2            0.000000          -42.00000
   X3            1.000000          -85.00000
   X4            0.000000          -60.00000
   X5            1.000000          -50.00000
   X6            0.000000          -18.00000
   X7            1.000000          -170.0000
   X8            1.000000          -90.00000
   X9            0.000000          -100.0000
   X10           0.000000          -70.00000
```

结果显示：只有 x3、x5、x7 和 x8 的值为 1，即物品编号为 3、5、7 和 8 时携带的物品的总价值最高，价值为 395 元。且这 4 种物品的体积之和刚好是旅行包的容积 20L，没有浪费旅行包的空间。

【例 4-4】 帆船生产计划模型

问题重述 S 造船公司需决定下四个季度的帆船生产计划。已知下四个季度帆船的需求量分别为：40 条、60 条、75 条、25 条，必须按时满足客户需要。S 公司每个季度正常生产 40 条帆船，生产成本为 1000 美元/条。如果加班生产，成本增加到 1200 美元/条，每个季度末，帆船的库存费为 100 美元/条。假定生产提前期记为 0，初始库存的帆船有 10 条。如何安排生产，可使总费用最小？

问题分析 总费用包括正常生产的成本、加班生产的成本和库存费。库存费是由生产的帆船数量决定的，库存帆船数等于当前季度生产的（正常生产和加班生产的）帆船数与上一季度库存数之和减去当前季度的需求量。

符号说明

x_i 表示第 i 个季度正常生产帆船的数量，$i=1,2,3,4$。

y_i 表示第 i 个季度加班生产帆船的数量，$i=1,2,3,4$。

c_i　表示第 i 个季度帆船的库存数量，$i=1,2,3,4$。

d_i　表示第 i 个季度帆船的需求量，$i=1,2,3,4$。

模型建立

决策变量：正常生产帆船量 x_i，加班生产帆船量 y_i

目标函数：所有费用之和

$$Z=\sum_{i=1}^{4}1000x_i+1200y_i+100c_i$$

约束条件有两个：

（1）正常生产能力限制，即 $x_i\le 40,\ i=1,2,3,4$

（2）产品供需平衡方程，$c_i=x_i+y_j+c_{i-1}-d_i,\ i=1,2,3,4$

$$c_0=10$$

综上可得

$$\min\quad Z=\sum_{i=1}^{4}1000x_i+1200y_i+100c_i$$

$$s.t.\quad x_i\le 40,\ i=1,2,3,4$$

$$c_i=x_i+y_j+c_{i-1}-d_i,\ i=1,2,3,4$$

$$c_0=10$$

$$x_i,\ y_i\in Z$$

该问题的模型整数线性规划模型。

模型求解

上述模型用 LINGO 软件输入如下：

```
model:
min=1000*x1+1200*y1+100*c1+1000*x2+1200*y2+100*c2+1000*x3+1200*y
3+100*c3+1000*x4+1200*y4+100*c4; !目标函数;
x1<=40;!正常生产能力的约束;
```

```
x2<=40;
x3<=40;
x4<=40;
c1=x1+y1+10-40;!平衡方程;
c2=x2+y2+c1-60;
c3=x3+y3+c2-75;
c4=x4+y4+c3-25;
@gin(x1);!整数约束;
@gin(x2);
@gin(x3);
@gin(x4);
@gin(y1);
@gin(y2);
@gin(y3);
@gin(y4);
end
```

单击求解模型键，运行结果报告如下：

```
 Global optimal solution found.
 Objective value:                    200000.0
 Variable        Value           Reduced Cost
     X1        40.00000          1400.000
     Y1        0.000000          1600.000
     C1        10.00000          0.000000
     X2        40.00000          1300.000
     Y2        10.00000          1500.000
     C2        0.000000          0.000000
     X3        40.00000          1200.000
     Y3        35.00000          1400.000
     C3        0.000000          0.000000
     X4        25.00000          1100.000
     Y4        0.000000          1300.000
     C4        0.000000          0.000000
```

结果显示：总费用最小为 200000.0 美元，每个季度正常生产的数量分别为：40、40、40、25，每个季度加班生产量为：0、10、35、0。

注：由这几个例子发现，决策变量都可用数组（或矩阵）表示，当变量越多时，模型中的表达式展开后会很长，编写 LINGO 程序麻烦。LINGO 中没有数组和矩阵这样的数据结构，但是 LINGO 可用集合来实现数组和矩阵的表示。

4.2.3 LINGO 中集合的使用

LINGO 建模语言最重要的是集合及属性，用上集合函数后，一行简短的语言就可以表示含有很多变量的目标函数或成千上万的约束条件，从而实现大规模 LINGO 建模语言的简化。

1. 集合的基本用法和 LINGO 模型的基本要素

大规模模型的 LINGO 语言由 5 个部分组成，分别为：集合段、目标和约束段、数据段、初始段和计算段。

以【例 4-3】旅行包优化模型为例，LINGO 语言如下：

```
model:
sets:  !集合段开始;
iset/1..10/:x,p,v;  !集合声明;
endsets !集合段结束;

max=@sum(iset(i):x(i)*p(i)); !目标函数;
@sum(iset(i):x(i)*v(i))<=vmax; !约束条件;
@for(iset(i):@bin(x(i))); !0-1 变量;

data:  !数据段开始;
vmax=20;
p=20,42,85,60,50,18,170,90,100,70;
v=2,3,5,4,3,1,8,4,7,6;
enddata !数据段结束;

calc:  !计算段开始;
pt=@sum(iset(i):p(i)); !计算所有物品总价;
endcalc !计算段结束;
end
```

单击求解模型键 ，运行结果报告如下：下面仅列出 LINGO 求解的目标值和决策变量值

```
Global optimal solution found.
Objective value:                    395.0000
Variable            Value          Reduced Cost
  VMAX            20.00000           0.000000
   PT             705.0000           0.000000
   X1             0.000000          -20.00000
   X2             0.000000          -42.00000
   X3             1.000000          -85.00000
   X4             0.000000          -60.00000
   X5             1.000000          -50.00000
   X6             0.000000          -18.00000
   X7             1.000000          -170.0000
   X8             1.000000          -90.00000
   X9             0.000000          -100.0000
   X10            0.000000          -70.00000
```

从这个例子可以看出，利用集合后目标函数和约束条件相比前面直接展开的方法更简短，变量再多也只需在定义集合段里改一下变量个数即可。因此，利用集合编写 LINGO 语言是有必要的，下面对各段进行说明。

（1）集合段

集合段以“sets:”开始，以“endset”结束。用于定义集合及其元素（数组的下标）和属性（数组）。集合定义格式如下：

```
sets:
setname/member_list/:attribute_list;
endsets
```

其中：

setname 表示集合名，要符合变量命名规则；

member_list 表示元素列表，两种写法：①全列举，如“1,2,3,4,5”，②两点省略法，如“1..5”；

attribute_list 表示属性列表（同元素个数的属性写在一个集合中）。

注意：①一个 LINGO 建模语言中可定义多个集合。

②“sets:”和“endset”都各自独占一行

（2）目标与约束段

这部分没有段开始和段结束（其他 4 部分都有段开始和段结束），目标函数、约束条件、变量约束等内容。这里会常常用到集合函数对集合中

的元素（下标）进行循环操作，集合函数的调用格式如下：

```
@function(setname(set_index_list)|condition:exprssion_list
```

其中：

function 是集合函数名，常用的有 for、sum、max、min、prod 五种；
setname 是集合名，必须是集合段中已定义的集合名；
set_index_list 表示集合索引列表，相当于数组中的下标变量（可省略）；
condition 是用逻辑表达式描述的条件（可省略）；
exprssion_list 是一个表达式（@for 循环函数中，表示多个表达式）。

5 个常用的集合函数的含义见表 4-9

表 4-9　5 个集合函数的含义

集合函数	含义	说明
@for	循环函数	setname 中每个元素生成一个表达式，有几个元素就能生成几个表达式，常用于约束条件或变量约束
@sum	求和函数	setname 上的表达式的和
@max	最大值函数	setname 上的表达式的最大值
@min	最小值函数	setname 上的表达式的最小值
@prod	乘积函数	setname 上的表达式的积

如【例 4-3】的 LINGO 语言中：

```
@sum(iset(i):x(i)*v(i))<=vmax; !对 iset 的表达式 x(i)*v(i)求和<=vmax;
@for(iset(i):@bin(x(i))); !iset 集合的每个元素进行 0-1 约束;
```

（3）数据段

数据段以“data:”开始，以“enddata”结束。作用是对已知数据的输入，常用于以下两种情况：

① 集合属性的数据输入，格式为

```
data:
attribute=constant_list;
enddata
```

其中：

attribute 是集合中已定义的属性（名）；
constant_list 是常数列表，属性的常数列表至少有 2 个（含 2 个）以上，数据之间可用逗号“,”或空格间隔，注意顺序。

如：x=5,2;　或 x=5　2;

② 参数赋值：指的是对单个变量赋值，格式为

```
name=constant
```

如：vmax=20;

（4）初始段

初始段以“init:”开始，以“endinit”结束。作用是集合中的属性定义初值，在迭代算法中，一个好的迭代初值，能提高算法的计算效率。其格式与属性的数据输入相同，如下：

```
init:
attribute=constant_list;
endinit
```

注意：初始部分不是每个 LINGO 建模语言都有的，这部分可以省略。

（5）计算段

计算段以“calc:”开始，以“endcalc”结束。作用是原始数据进行计算(这部分必须数据段的数据输入完成后,LINGO 开始求解模型之前进行)。

如【例 4-3】中的计算段

```
calc:
pt=@sum(iset(i):p(i)); !计算所有物品总价;
endcalc
```

注意：① 计算段不是每个 LINGO 建模语言都有的，这部分可以省略；

② 计算段中的语句是按从上到下的顺序执行的；

③ 计算段中只能使用已有的赋值数据，不能包含需求解后才得到的变量值。

并不是每个 LINGO 建模编程语言都会完全包含这 5 个部分的，很多时候是没有初始段和计算段，这就需要具体模型具体分析了。

利用集合法对【例 4-4】求解，其 LINGO 建模语言如下：

```
model:
sets: !集合段开始;
iset/1..4/:x,y,c,d;
endsets !集合段结束;

min=@sum(iset(i):1000*x(i)+1200*y(i)+100*c(i));
@for(iset(i):x(i)<=40);
```

```
@for(iset(i)|i#ge#2:c(i)=x(i)+y(i)+c(i-1)-d(i););
c(1)=x(1)+y(1)+c0-d(1);
@for(iset(i):@gin(x(i)));
@for(iset(i):@gin(y(i)));

data: !数据段开始;
c0=10;
d=40,60,75,25;
enddata !数据段结束;
end
```

【例 4-5】 选址问题

问题重述 某建筑公司同时对 6 个工地施工，每个工地的平面坐标位置（单位：km）和水泥日用量（单位：吨）见表 4-8。目前有两个临时料场，分别位于 A(5,1)，B(2,7)，水泥日储量各有 20t。

表 4-8 6 个工地的坐标及水泥日用量

工地编号	1	2	3	4	5	6
横坐标(km)	1.5	9	0.5	6	3	7.5
纵坐标(km)	1.5	1	5	5	6.5	8
水泥日用量(t)	3	5	4	7	6	11

问题：

（1）假设从料场到工地之间均有直线道路相连，制定水泥日供应计划，使总的吨公里数最小。

（2）为减少吨公里数，新建两个日储量为 20 吨的料场。新料场应建在何处，与原 A、B 两个料场相比减少的吨公里数是多少？

符号说明

x_i 表示第 i 个料场的横坐标，$i=1,2$。

y_i 表示第 i 个料场的纵坐标，$i=1,2$。

n_i 表示第 i 个料场水泥的日储量，$i=1,2$。

a_j 表示第 j 个工地的横坐标，$j=1,2,\cdots,6$。

b_j　表示第 j 个工地的纵坐标， $j=1,2,\cdots,6$ 。

d_j　表示第 j 个工地的水泥日用量， $j=1,2,\cdots,6$ 。

m_{ij}　表示料场 i 向工地 j 运送水泥的数量， $i=1,2$ ， $j=1,2,\cdots,6$ 。

模型建立

决策变量：问题（1）中，决策变量是料场 i 向工地 j 运送水泥的数量 m_{ij} ，在问题（2）中，决策变量除了 m_{ij} 外，新料场位置 (x_i, y_i) 也是决策变量。

目标函数：吨公里数（运送的数量乘以运送距离）

$$Z=\sum_{i=1}^{2}\sum_{j=1}^{6} m_{ij}\cdot\sqrt{(x_i-a_j)^2+(y_i-b_j)^2}$$

约束条件有两个：

（1）满足各工地的水泥日用量，即 $\sum_{i=1}^{2} m_{ij}=d_j$， $j=1,2,\cdots,6$

（2）各料场的运送量不能超过日储量，即 $\sum_{j=1}^{6} m_{ij}\le n_i$， $i=1,2$

综上可得

$$\min\quad Z=\sum_{i=1}^{2}\sum_{j=1}^{6} m_{ij}\cdot\sqrt{(x_i-a_j)^2+(y_i-b_j)^2}$$

$$s.t.\quad \sum_{i=1}^{2} m_{ij}=d_j,\quad j=1,2,\cdots,6$$

$$\sum_{j=1}^{6} m_{ij}\le n_i,\quad i=1,2$$

问题（1）的决策变量只有 m_{ij} （非负整数），该问题模型是线性规划模型。问题（2）中决策变量为 m_{ij} （非负整数）和 x_i 、 y_i ，该问题模型是

非线性规划模型。

用 LINGO 对该模型的求解，若直接用展开法求解很麻烦，显然是用集合法求解更有效。但是本模型与前面的例子区别在于有双下标的变量，因此，在前面（基本）集合（一维的）的基础上派生二维的甚至多维的集合。

2. 基本集合与派生集合

直接列举元素的集合称为基本集合，可以理解为一维数组。由一个或多个基本集合派生出来的集合称为派生集合，可以理解为二维（或多维）数组，其调用格式如下：

```
sets:
setname(parent_set_list):attribute_list;
endsets
```

其中：

setname 表示集合名，要符合变量命名规则；

parent_set_list 表示父集合列表，父集合（基本集合）已定义的才有效；

attribute_list 表示属性列表。

利用基本集合与派生集合对【例 4-5】求解，问题（1）LINGO 建模语言如下：

```
model:
sets: !集合段开始;
iset/1,2/:x,y,n;
jset/1..6/:a,b,d;
ijset(iset,jset):m;!派生集合;
endsets !集合段结束;

min=@sum(ijset(i,j):m(i,j)*((x(i)-a(j))^2+(y(i)-b(j))^2)^(1/2));
@for(jset(j):@sum(iset(i):m(i,j))=d(j););
@for(iset(i):@sum(jset(j):m(i,j))<=n(i););
@for(ijset:@gin(m););

data: !数据段开始;
x=5,2;
y=1,7;
n=20,20;
a=1.5,9,0.5,6,3,7.5;
b=1.5,1,5,5,6.5,8;
d=3,5,4,7,6,11;
```

```
enddata !数据段结束;
end
```

单击求解模型键，运行结果报告如下：下面仅列出 LINGO 求解的目标值和决策变量值

```
Global optimal solution found.
Objective value:                    139.5113
Variable             Value          Reduced Cost
M( 1, 1)         3.000000            3.535534
M( 1, 2)         5.000000            4.000000
M( 1, 3)         0.000000            6.020797
M( 1, 4)         7.000000            4.123106
M( 1, 5)         0.000000            5.852350
M( 1, 6)         1.000000            7.433034
M( 2, 1)         0.000000            5.522681
M( 2, 2)         0.000000            9.219544
M( 2, 3)         4.000000            2.500000
M( 2, 4)         0.000000            4.472136
M( 2, 5)         6.000000            1.118034
M( 2, 6)         10.00000            5.590170
```

问题（2）LINGO 建模语言如下：

```
model:
sets: !集合段开始;
iset/1,2/:x,y,n;
jset/1..6/:a,b,d;
ijset(iset,jset):m;
endsets !集合段结束;

min=@sum(ijset(i,j):m(i,j)*((x(i)-a(j))^2+(y(i)-b(j))^2)^(1/2));
@for(jset(j):@sum(iset(i):m(i,j))=d(j););
@for(iset(i):@sum(jset(j):m(i,j))<=n(i););
@for(ijset:@gin(m););
@for(iset(i):@free(x(i));@free(y(i)););
```

```
data:  !数据段开始;
n=20,20;
a=1.5,9,0.5,6,3,7.5;
b=1.5,1,5,5,6.5,8;
d=3,5,4,7,6,11;
enddata !数据段结束;

init:!初始段开始;
x=5,2;
y=1,7;
endinit !初始段结束;

end
```

单击求解模型键 ，运行结果报告如下：下面仅列出 LINGO 求解的目标值和决策变量值

```
  Global optimal solution found.
  Objective value:                  85.67942
  Variable           Value        Reduced Cost
      X( 1)        3.226248         0.000000
      X( 2)        7.500000       -0.1439972E-07
      Y( 1)        5.767724         0.000000
      Y( 2)        8.000000       -0.1170255E-06
  M( 1, 1)        3.000000        -4.242274
  M( 1, 2)        0.000000         0.3289063
  M( 1, 3)        4.000000        -4.783490
  M( 1, 4)        7.000000       -0.4760646
  M( 1, 5)        6.000000        -3.976985
  M( 1, 6)        0.000000         0.000000
  M( 2, 1)        0.000000         0.000000
  M( 2, 2)        5.000000         0.000000
  M( 2, 3)        0.000000         0.000000
  M( 2, 4)        0.000000         0.000000
  M( 2, 5)        0.000000         0.000000
  M( 2, 6)        11.00000        -4.821619
```

问题（1）的求解的最小吨公里数为 139.5113。问题（2）中记新料场为 C 和 D，平面坐标位置分别为（3.226248, 5.767724）和（7.5, 8），料场 D 与工地 6 的位置相同。此时，求得最小吨公里数为 85.67942，相比旧料场节省的吨公里数为 53.8。新、旧两个料场分别向各工地运送水泥数量见表 4-10

表 4-10　新旧两个料场分别向各工地运送水泥数量

料场		运送水泥量（t）						合计（t）
		1	2	3	4	5	6	
旧	A	3	5	0	7	0	1	16
	B	0	0	4	0	6	10	20
新	C	3	0	4	7	6	0	20
	D	0	5	0	0	0	11	16

由表 4-10 可以看出，料场位置改变后，向各工地运送水泥路线也不相同。工地与新旧料场的平面坐标图见图 4-6。

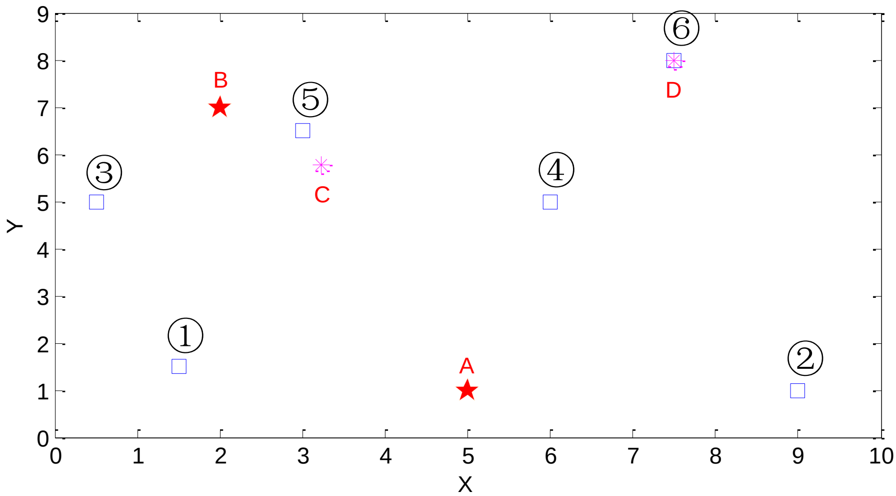


图 4-6　工地与新旧两料场平面示意图

【例 4-6】　指派问题

问题重述　从 5 名游泳队员中选择 4 人参加 4×100m 的混合泳接力赛，5 名队员 4 种泳姿的百米平均成绩见表 4-11 所示。

表 4-11　5 名队员 4 种泳姿的百米平均成绩（s）

队员编号	蝶泳	仰泳	蛙泳	自由泳
1	66.8	75.8	87	58.6
2	57.2	66	66.4	53
3	78	67.8	84.6	59.4
4	70	74.2	69.6	57.2
5	67.4	71	83.8	62.4

问题　如何选拔游泳队员参加混合泳接力赛？

问题分析　要求从 5 名游泳队员中选出 4 人参加混合泳接力赛，每人一种游资，且 4 人的泳姿各不相同，使混合泳接力赛总用时最短。若采用穷举法，有 5！=120 种组合方案，计算量太大不实用。该问题可以用 0-1 变量建立规划模型。

符号说明

x_{ij}　0-1 变量，表示队员 i 是否参加第 j 种泳姿，$i=1,2,\cdots,5$，$j=1,2,3,4$。

c_{ij}　表示队员 i 对应泳姿 j 的百米平均成绩，$i=1,2,\cdots,5$，$j=1,2,3,4$。

模型建立

决策变量：$x_{ij}=\begin{cases}0，表示队员i不参加泳姿j\\1，表示队员i参加泳姿j\end{cases}$

目标函数：混合泳接力赛总用时，即 $Z=\sum_{i=1}^{5}\sum_{j=1}^{4}x_{ij}c_{ij}$

约束条件：

（1）每个最多只能选 1 种泳姿，即 $\sum_{j=1}^{4}x_{ij}\le 1,\ \ i=1,2,\cdots,5$

（2）每种泳姿都只能有 1 人入选，即 $\sum_{i=1}^{5}x_{ij}=1,\ \ j=1,2,3,4$

综上可得

$$\min Z=\sum_{i=1}^{5}\sum_{j=1}^{4}x_{ij}c_{ij}$$

$$s.t.\quad \sum_{j=1}^{4}x_{ij}\le 1,\quad i=1,2,\cdots,5$$

$$\sum_{i=1}^{5}x_{ij}=1,\quad j=1,2,3,4$$

$$x_{ij}=0 \text{ 或 } x_{ij}=1，\quad i=1,2,\cdots,5,\ j=1,2,3,4$$

该问题的模型为 0-1 线性整数规划模型。

模型求解

上述模型用 LINGO 软件输入如下：

```
model:
sets:
is/1..5/:;
js/1,2,3,4/:;
lij(is,js):x,c;
endsets
min=@sum(lij(i,j):x(i,j)*c(i,j));
@for(is(i):@sum(js(j):x(i,j))<=1;);
@for(js(j):@sum(is(i):x(i,j))=1;);
@for(lij(i,j):@bin(x(i,j)));
data:
c=66.8  75.8   87    58.6
  57.2  66     66.4  53
  78    67.8   84.6  59.4
  70    74.2   69.6  57.2
  67.4  71     83.8  62.4;
enddata
end
```

单击求解模型键，运行结果报告如下：下面仅列出 LINGO 求解的目标值和决策变量值

```
  Global optimal solution found.
```

```
Objective value:                  253.2000
Variable           Value        Reduced Cost
   X( 1, 1)        0.000000         66.80000
   X( 1, 2)        0.000000         75.80000
   X( 1, 3)        0.000000         87.00000
   X( 1, 4)        1.000000         58.60000
   X( 2, 1)        1.000000         57.20000
   X( 2, 2)        0.000000         66.00000
   X( 2, 3)        0.000000         66.40000
   X( 2, 4)        0.000000         53.00000
   X( 3, 1)        0.000000         78.00000
   X( 3, 2)        1.000000         67.80000
   X( 3, 3)        0.000000         84.60000
   X( 3, 4)        0.000000         59.40000
   X( 4, 1)        0.000000         70.00000
   X( 4, 2)        0.000000         74.20000
   X( 4, 3)        1.000000         69.60000
   X( 4, 4)        0.000000         57.20000
   X( 5, 1)        0.000000         67.40000
   X( 5, 2)        0.000000         71.00000
   X( 5, 3)        0.000000         83.80000
   X( 5, 4)        0.000000         62.40000
```

结果显示：当队员 1 选自由泳、队员 2 选蝶泳、队员 3 选仰泳、队员 4 选蛙泳时，混合泳接力赛成绩最好，总用时为 253.2 秒。

【例 4-6】 营养搭配问题

问题重述 营养师给病人拟定一周蔬菜菜单，可选的蔬菜品种、价格和营养成分含量，以及病人一周所需营养成分的要求如表 4-12 所示。

表 4-12 蔬菜营养成分及价格

蔬菜品种 \ 养分		每份蔬菜所含营养成份数量					价格(元/份)
		铁	磷	维生素 A	维生素 B	烟酸	
1	青豆	0.45	20	415	22	0.3	4.2
2	胡萝卜	0.45	28	4065	5	0.35	2
3	花菜	0.65	40	850	43	0.6	3.6
4	卷心菜	0.4	25	75	27	0.2	2.4

5	芹菜	0.5	26	76	48	0.4	4
6	土豆	0.5	75	235	8	0.6	2.8
每周最低需求		6	125	12500	345	5	

问题 设计一周费用最少的方案。满足以下条件：每周蔬菜不得少于 14 份，且每种蔬菜必须有，其中卷心菜不能超过 2 份，胡萝卜不能超过 3 份，其他蔬菜不能超过 4 份。

符号说明

x_i 表示第 i 种蔬菜的份数，$i=1,2,\cdots,6$。

p_i 表示第 i 种蔬菜的价格（元/份），$i=1,2,\cdots,6$。

m_j 表示第 j 种营养成份的最低周需求，$j=1,2,\cdots,5$。

a_{ij} 表示每份蔬菜 i 的第 j 种营养成份的含量，$i=1,2,\cdots,6$，$j=1,2,\cdots,5$。

模型建立

决策变量：每周各种蔬菜的份数 x_i

目标函数：一周蔬菜的总费用，即 $Z=\sum_{i=1}^{6} x_i p_i$

约束条件：

（1）每周营养成份最低要求，即 $\sum_{i=1}^{6} x_i a_{ij} \ge m_j$，$j=1,2,\cdots,5$

（2）每周至少 14 份蔬菜，即 $\sum_{i=1}^{6} x_i \ge 14$

（3）每种蔬菜必须有，即 $x_i \ge 1, i=1,2,\cdots,6$

（4）卷心菜不能超过 2 份，即 $1 \le x_4 \le 2$

（5）胡萝卜不能超过 3 份，即 $x_2 \le 3$

（6）其他蔬菜不得超过 4 份，即 $x_i \le 4, i=1,3,5,6$

综上可得

$$\min Z = \sum_{i=1}^{6} x_i p_i$$

$$s.t. \quad \sum_{i=1}^{6} x_i a_{ij} \ge m_j, \quad j=1,2,\cdots,5$$

$$\sum_{i=1}^{6} x_i \ge 14$$

$$x_4 \le 2$$

$$x_2 \le 3$$

$$x_i \ge 1, i=1,2,\cdots,6$$

$$x_i \le 4, i=1,3,5,6$$

$$x_i \in N, i=1,2,\cdots,6$$

该问题的模型为线性整数规划模型。

模型求解

上述模型用 LINGO 软件输入如下：

```
model:
sets:
is/1..6/:p,x;
js/1..5/:m;
lij(is,js):a;
endsets
min=@sum(is(i):x(i)*p(i));
@for(js(j):@sum(is(i):x(i)*a(i,j))>=m(j););
@sum(is(i):x(i))>=14;
x(4)<=2;
x(2)<=3;
```

```
@for(is(i):x(i)>=1);
@for(is(i)|i#ne#2#and#i#ne#4:x(i)<=4);
@for(is(i):@gin(x(i)););
data:
m=6  125  12500  345  5;
p=4.2  2  3.6  2.4  4  2.8;
a=0.45 20  415     22  0.3
  0.45  28  4065    5   0.35
  0.65  40  850     43  0.6
  0.4   25  75      27  0.2
  0.5   26  76      48  0.4
  0.5   75  235     8   0.6;
enddata
end
```

单击求解模型键，运行结果报告如下：下面仅列出 LINGO 求解的目标值和决策变量值

```
  Global optimal solution found.
  Objective value:                    42.60000
  Variable             Value           Reduced Cost
      X( 1)         1.000000            4.200000
      X( 2)         3.000000            2.000000
      X( 3)         2.000000            3.600000
      X( 4)         2.000000            2.400000
      X( 5)         3.000000            4.000000
      X( 6)         3.000000            2.800000
```

结果显示：一周蔬菜菜单为，青豆 1 份、胡萝卜 3 份、花菜 2 份、卷心菜 2 份、芹菜 3 份、土豆 3 份，总费用取得最小为 42.6 元。

由【例 4-5】和【例 4-6】可以看出，当数据量比较多时，在数据段中直接输入数据比较麻烦，占用程序窗口篇幅，且不利于 LINGO 语言和数据的检查。因此，LINGO 与外部文件之间的数据传递是有必要学习的。

4.2.4 LINGO 软件与外部文件数据的传递

在实际问题中，LINGO 建模用到的数据规模较大，且这些数据通常保存在 Word、记事本（文档 txt）、Excel 等这些文件中。由此，通过外部文

件输入输出数据是编写 LINGO 程序的基本要求。

LINGO 与外部文件输入输出数据的方法有 3 种。

1. 直接用“复制”和“粘贴”命令操作

这个方法用于数据量较少的情况，若是复制表格中的数据（Word 或 Excel 表格）直接粘贴到 LINGO 程序中，还是有表格的。若不需要表格可以先把数据粘贴到记事本（txt 文档）后，然后再复制记事本（txt 文档）的数据粘贴到 LINGO 中即可把表格去掉。

2. LINGO 与文本文件数据的输入与输出

（1）@file 输入函数

LINGO 获取文本文件的数据可利用函数@file 来实现，通常用在集合段和数据段中，这个函数的调用格式为：

```
@file(filename)  !filename 是存放数据的文件名或文件的路径
```

filename 有两个格式：

① 若数据文本文件保存在 LINGO 安装程序的同一文件夹中，则 filename 表示数据的文件名

② 数据文件存放在别的文件夹中，则 filename 表示数据文件的路径。

注意：@file 只能用于记事本（.txt）文件或 LINGO 文件（.ldt），且必须写文件的后缀。

以【例 4-6】为例，把 a_{ij} 的数据存放在记事本（.txt）文件或 LINGO 文件（.ldt）中，将文件重命名为 a_data，则数据段可表示为：

```
data:
m=6  125  12500  345  5;
p=4.2  2  3.6  2.4  4  2.8;
a=@file(a_data.txt);
enddata
```

此时，数据文件文档 a_data.txt 存放在 LINGO 安装目录的文件夹中，且 a_data.txt 文档中只有 a_{ij} 的数据（见图 4-7）。a_data.txt 存

放在别的文件夹中，可表示为：

```
data:
m=6  125  12500  345  5;
p=4.2  2  3.6  2.4  4  2.8;
a=@file(E:\a_data.txt);
enddata
```

a_data - 记事本

文件(F) 编辑(E) 格式(O) 查看(V)

```
0.45  20  415   22  0.3
0.45  28  4065  5   0.35
0.65  40  850   43  0.6
0.4   25  75    27  0.2
0.5   26  76    48  0.4
0.5   75  235   8   0.6
```

图 4-7 a_data 文件中的数据

a_data1 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮

```
 6    125  12500  345  5~
4.2   2    3.6    2.4  4  2.8~
0.45  20   415    22   0.3
0.45  28   4065   5    0.35
0.65  40   850    43   0.6
0.4   25   75     27   0.2
0.5   26   76     48   0.4
0.5   75   235    8    0.6
```

图 4-8 a_data1 文件中的数据

把记录结束标记“~”之间的数据文件部分称为记录，图 4-8 的 a_data1 文件中的数据被分为 3 个记录，用@file 调用数据时，是按顺序先调用第 1 个记录、然后调用第 2 个记录、最后调用第 3 个记录。而图 4-7 中没有记录结束标记“~”，则整个文件被看作单个记录。

若 a_data1 文件存放在存放在 LINGO 安装目录的文件夹中，此时【例 4-6】LINGO 程序的数据段表示为：

```
data:
m=@file(a_data1.txt);
p=@file(a_data1.txt);
a=@file(a_data1.txt);
enddata
```

若是 a_data1 文件存放在 E 盘，则表示为

```
data:
m=@file(E:\a_data1.txt);
p=@file(E:\a_data1.txt);
a=@file(E:\a_data1.txt);
enddata
```

注意：若用“~”按记录调用数据时，要注意数据的顺序，一个@file 调用一个记录，从上往下依次调用。在 LINGO 不允许嵌套调用@file 函数。

（2）@text 输出函数

LINGO 程序的计算结果输出到文本文件中，可用函数@text 来实现，可以输出集合成员和集合属性值，其调用格式为：

```
@text('filename')=集合成员名或集合属性名 !单引号不能缺
```

filename 是文件名，也是有两个表达方式，直接写文件名则在 LINGO 软件安装目录下自动生成一个 filename 文件存放等号右边的变量值。也可写路径表示指定 filename 文件保存在该个路径下。

@text 只能在数据段中使用，见图 4-9

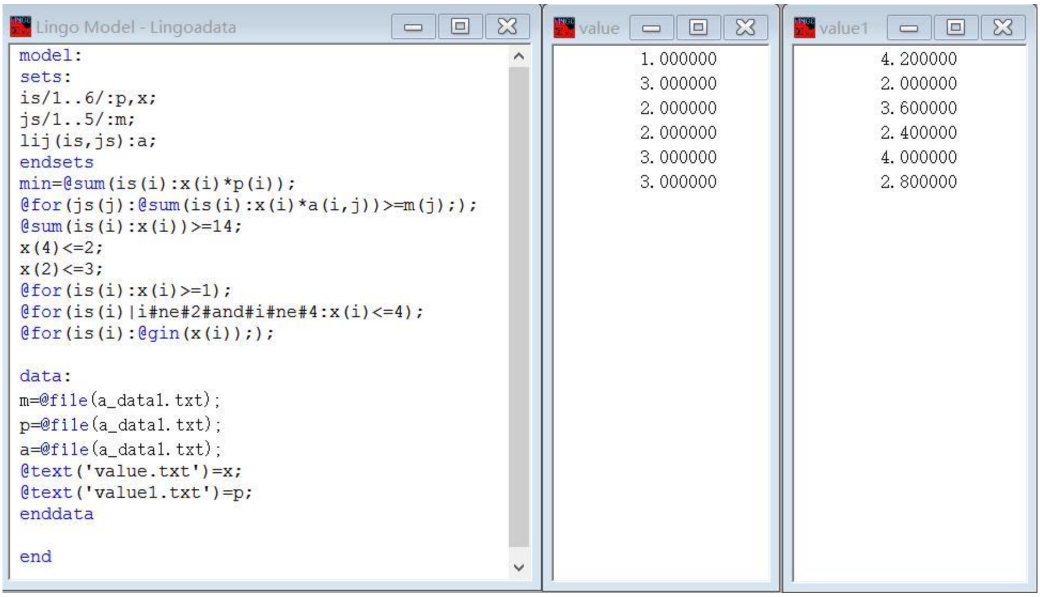


图 4-9　@tetx 输出数据的表示形式

3. LINGO 与 EXCEL 文件数据的输入与输出

LINGO 与 EXCEL 数据文件的输入或输出都是用@ole 函数实现，该函数只能用在 LINGO 模型中的集合段、数据段和初始段，其输入和输出调用格式分别：

```
变量名=@ole('filename.xls','数据块名称')!输入调用格式
```

```
@ole('filename.xls','数据块名称')=变量名 !输出调用格式
```

注意：

① filename 也可用两种表示形式，文件名或路径（与@file 用法一致）；

② EXCEL 数据文件要在打开状态下运行 LINGO 程序才能运行；

③ LINGO 程序中的“数据块名称”必须与 EXCEL 中“数据块名称”一致。

EXCEL 中“数据块名称”命名步骤：选择数据区域，单击菜单中的“插入”→“名称”→“定义”，在弹出“定义名称”对话框（见图 4-10）中输入“数据块名称”（符合变量命名规则），然后单击“添加”→“确定”，这样就完成了第 1 个数据块名称的命名。

图 4-10　EXCEL 表格定义数据块名称

若是 WPS 版本的表格，则需在菜单“公式”→“名称管理器”，弹出对话框（见图 4-11）后，单击“新建”→填写“名称”→“引用位置”选择数据区域完成第 1 个数据块名称的命名。

图 4-11　WPS 表格定义数据块名称

以【例 4-6】为例，把 EXCEL 命名为“data”，则数据段可表示为：

```
data:
m=@ole('data.xls',m);
p=@ole('data.xls',p);
a=@ole('data.xls',a);
enddata
```

此时名为“data”的 EXCEL 文件须保存在 LINGO 程序安装的目录下，若“data.xls”文件保存在 E 盘时，则表示为

```
data:
m=@ole('E:\data.xls',m);
p=@ole('E:\data.xls',p);
a=@ole('E:\data.xls',a);
enddata
```

不管“data.xls”文件保存在哪里，都必须要打开“data”文件，才能运行。

@ole 数据输出也要先命名“数据块名称”，然后将所需输出的变量写在@ole 等号右边即可，如下所示：

```
data:
m=@ole('E:\data.xls',m);
p=@ole('E:\data.xls',p);
a=@ole('E:\data.xls',a);
@ole('data.xls',xvalue)=x;
enddata
```

将 x 变量的值输出到名为 xvalue 的数据块，输出结果见图 4-12.

xvalue　fx　1

A	B	C	D	E	F	G	H	I
		每份蔬菜所含营养成份数量						
蔬菜品种		铁	磷	维生素A	维生素B	烟酸	价格(元/份)	x值
1	青豆	0.45	20	415	22	0.3	4.2	1
2	胡萝卜	0.45	28	4065	5	0.35	2	3
3	花菜	0.65	40	850	43	0.6	3.6	2
4	卷心菜	0.4	25	75	27	0.2	2.4	2
5	芹菜	0.5	26	76	48	0.4	4	3
6	土豆	0.5	75	235	8	0.6	2.8	3
每周最低需求		6	125	12500	345	5		

图 4-12　@ole 输出数据到 EXCEL 文件中

用@ole 函数与 EXCEL 数据输入或者输出都必须先打开该 EXCEL 文件才能运行，且要先定义好数据块。