

目 录

| | |
|--|--|
| 第一章 引言 | |
| 1.1 优化模型的基本概念 | |
| 1.1.1 优化模型的一般形式 | |
| 1.1.2 可行解与最优解 | |
| 1.1.3 优化模型的基本类型 | |
| 1.2 优化问题的建模实例 | |
| 1.2.1 线性规划模型 | |
| 1.2.2 二次规划模型 | |
| 1.2.3 非线性规划模型 | |
| 1.2.4 整数规划模型 | |
| 1.2.5 其他优化模型 | |
| 1.3 LINDO/LINGO 软件简介 | |
| 1.3.1 LINDO/LINGO 软件的基本功能 | |
| 1.3.2 LINDO/LINGO 软件的求解过程 | |
| 1.3.3 建立 LINDO/LINGO 优化模型需要注意的几个基本问题 | |
| 习题 1 | |
| 第二章 LINDO 软件的基本使用方法 | |
| 2.1 LINDO 入门 | |
| 2.1.1 LINDO 软件的安装过程 | |
| 2.1.2 编写一个简单的 LINDO 程序 | |
| 2.1.3 一些注意事项 | |
| 2.2 敏感性分析 | |
| 2.3 整数线性规划的求解 | |
| *2.4 二次规划的求解 | |
| *2.5 LINDO 的主要菜单命令 | |
| 2.5.1 文件主菜单 | |
| 2.5.2 编辑主菜单 | |
| 2.5.3 求解主菜单 | |
| 2.5.4 报告主菜单 | |
| *2.6 LINDO 命令窗口 | |
| 2.6.1 INFORMATION (信息类命令) | |
| 2.6.2 INPUT (输入类命令) | |
| 2.6.3 DISPLAY (显示类命令) | |
| 2.6.4 OUTPUT (输出类命令) | |
| 2.6.5 SOLUTION (求解类命令) | |
| 2.6.6 PROBLEM EDITING (编辑类命令) | |
| 2.6.7 QUIT (退出类命令) | |
| 2.6.8 INTEGER, QUADRATIC, AND PARAMETRIC PROGRAMS (整数, 二次与参数规划命令) | |
| 2.6.9 CONVERSATIONAL PARAMETERS (对话类命令) | |
| 2.6.10 USER SUPPLIED ROUTINES (用户过程类命令) | |
| 2.6.11 MISCELLANEOUS (其他命令) | |

| | | |
|------------|-------------------------------|--|
| *2.7 | LINGO 命令脚本文件 | |
| 附录 | MPS 格式数据文件 | |
| 习题 2 | | |
| 第三章 | LINGO 软件的基本使用方法 | |
| 3.1 | LINGO 入门 | |
| 3.1.1 | LINGO 软件的安装过程和主要特色 | |
| 3.1.2 | 在 LINGO 中使用 LINGO 模型 | |
| 3.1.3 | 编写一个简单的 LINGO 程序 | |
| 3.2 | 在 LINGO 中使用集合 | |
| 3.2.1 | 集合的基本用法和 LINGO 模型的基本要素 | |
| 3.2.2 | 基本集合与派生集合 | |
| 3.2.3 | 稠密集合与稀疏集合 | |
| 3.2.4 | 集合的使用小结 | |
| 3.3 | 运算符和函数 | |
| 3.3.1 | 运算符及优先级 | |
| 3.3.2 | 基本的数学函数 | |
| 3.3.3 | 集合循环函数 | |
| 3.3.4 | 集合操作函数 | |
| 3.3.5 | 变量定界函数 | |
| 3.3.6 | 财务会计函数 | |
| 3.3.7 | 概率论中的相关函数 | |
| 3.3.8 | 文件输入输出函数 | |
| 3.3.9 | 结果报告函数 | |
| 3.3.10 | 其他函数 | |
| 3.4 | LINGO 的主要菜单命令 | |
| 3.4.1 | 文件主菜单 | |
| 3.4.2 | 编辑主菜单 | |
| 3.4.3 | LINGO 系统 (LINGO) 主菜单 | |
| 3.5 | LINGO 命令窗口 | |
| 习题 3 | | |
| 第四章 | LINGO 软件与外部文件的接口 | |
| 4.1 | 通过 WINDOWS 剪贴板传递数据 | |
| 4.1.1 | 粘贴命令的用法 | |
| 4.1.2 | 特殊粘贴命令的用法 | |
| 4.2 | 通过文本文件传递数据 | |
| 4.2.1 | 通过文本文件输入数据 | |
| 4.2.2 | 通过文本文件输出数据 | |
| 4.3 | 通过电子表格文件传递数据 | |
| 4.3.1 | 在 LINGO 中使用电子表格文件的数据 | |
| 4.3.2 | 将 LINGO 模型嵌入、链接到电子表格文件中 | |
| 4.4 | LINGO 命令脚本文件 | |
| 附录 | LINGO 出错信息 | |
| 习题 4 | | |
| 第五章 | 生产与服务运作管理中的优化问题 | |

| | |
|------------------------------|-----------------------|
| 5.1 | 生产与销售计划问题 |
| 5.1.1 | 问题实例 |
| 5.1.2 | 建立模型 |
| 5.1.3 | 求解模型 |
| 5.2 | 有瓶颈设备的多级生产计划问题 |
| 5.2.1 | 问题实例 |
| 5.2.2 | 建立模型 |
| 5.2.3 | 求解模型 |
| 5.3 | 下料问题 |
| 5.3.1 | 钢管下料问题 |
| 5.3.2 | 易拉罐下料问题 |
| 5.4 | 面试顺序与消防车调度问题 |
| 5.4.1 | 面试顺序问题 |
| 5.4.2 | 消防车调度问题 |
| 5.5 | 飞机定位和飞行计划问题 |
| 5.5.1 | 飞机的精度定位问题 |
| 5.5.2 | 飞机计划问题 |
| 习题 5 | |
| 第六章 经济与金融中的优化问题 | |
| 6.1 | 经济均衡问题及应用 |
| 6.1.1 | 单一生产商、单一消费者的情形 |
| 6.1.2 | 两个生产商、两个消费者的情形 |
| 6.1.3 | 拍卖与投标问题 |
| 6.1.4 | 交通流均衡问题 |
| 6.2 | 投资组合问题 |
| 6.2.1 | 基本的投资组合模型 |
| 6.2.2 | 存在无风险资产时的投资组合模型 |
| 6.2.3 | 考虑交易成本的投资组合模型 |
| 6.2.4 | 利用股票指数简化投资组合模型 |
| 6.2.5 | 其他目标下的投资组合模型 |
| 6.3 | 市场营销问题 |
| 6.3.1 | 新产品的市场预测 |
| 6.3.2 | 产品属性的效用函数 |
| 6.3.3 | 机票的销售策略 |
| 习题 6 | |
| 第七章 图论与网络模型 | |
| 7.1 | 运输问题与转运问题 |
| 7.1.1 | 运输问题 |
| 7.1.2 | 指派问题 |
| 7.1.3 | 转运问题 |
| 7.2 | 最短路问题和最大流问题 |
| 7.2.1 | 最短路问题 |
| 7.2.2 | 最大流问题 |
| 7.2.3 | 最小费用最大流问题 |

| | |
|------------|--------------------------|
| 7.3 | 最优连线问题与旅游商问题 |
| 7.3.1 | 最优连线问题 |
| 7.3.2 | 旅行商问题 |
| 7.4 | 计划评审方法和关键线路 |
| 7.4.1 | 计划网络图 |
| 7.4.2 | 计划网络的计算 |
| 7.4.3 | 关键线路与计划网络的优化 |
| 7.4.4 | 完成作业期望和实现时间的概率 |
| 习题 7 | |
| 第八章 | 目标规划模型 |
| 8.1 | 线性规划与目标规划 |
| 8.1.1 | 线性规划建模与目标规划建模 |
| 8.1.2 | 线性规划建模的局限性 |
| 8.2 | 目标规划的数学模型 |
| 8.2.1 | 目标规划的基本概念 |
| 8.2.2 | 目标规划模型的建立 |
| 8.2.3 | 目标规划的一般模型 |
| 8.2.4 | 求解目标规划的序贯式算法 |
| 8.3 | 目标规划模型的实例 |
| 8.4 | 数据包络分析 |
| 8.4.1 | 数据包络分析的基本概念 |
| 8.4.2 | C^2R 模型 |
| 8.4.3 | 数据包络分析的求解 |
| 习题 8 | |
| 第九章 | 对策论模型 |
| 9.1 | 二人常数和对策模型 |
| 9.1.1 | 二人零和对策 |
| 9.1.2 | 二人常数和对策 |
| 9.2 | 二人常数和对策 |
| 9.2.1 | 纯对策问题 |
| 9.2.2 | 混合对策问题 |
| 9.3 | n 人合作对策初步 |
| 习题 9 | |
| 第十章 | 排队论模型 |
| 10.1 | 排队服务系统的基本概念 |
| 10.1.1 | 排队的例子及基本概念 |
| 10.1.2 | 符号表示 |
| 10.1.3 | 描述排队系统的主要数量指标 |
| 10.1.4 | 与排队论模型有关的 LINGO 函数 |
| 10.2 | 等待制排队模型 |
| 10.2.1 | 等待制排队模型的基本参数 |
| 10.2.2 | 等待制排队模型的计算实例 |
| 10.3 | 损失制排队模型 |

| | | |
|-------------|--------|-----------------------------|
| | 10.3.1 | 损失制排队模型的基本公式 |
| | 10.3.2 | 损失制排队模型的计算实例 |
| 10.4 | | 混合制排队模型 |
| | 10.4.1 | 混合制排队模型的基本公式 |
| | 10.4.2 | 混合制排队模型的基本参数 |
| | 10.4.3 | 混合制排队模型的计算实例 |
| 10.5 | | 闭合式排队模型 |
| | 10.5.1 | 闭合式排队模型的基本参数 |
| | 10.5.2 | 闭合式排队模型的计算实例 |
| 10.6 | | 排队系统的最优化模型 |
| | 10.6.1 | 系统服务时间的确定 |
| | 10.6.2 | 系统服务台（员）的确定 |
| | 习题 10 | |
| 第十一章 | | 存储论模型 |
| 11.1 | | 存储论模型简介 |
| | 11.1.1 | 问题的引入 |
| | 11.1.2 | 存储论模型的基本概念 |
| 11.2 | | 经济订购批量存储模型 |
| | 11.2.1 | 基本的经济订购批量存储模型 |
| | 11.2.2 | 带有约束的经济订购批量存储模型 |
| | 11.2.3 | 允许缺货的经济订购批量存储模型 |
| | 11.2.4 | 带有约束允许缺货模型 |
| | 11.2.5 | 经济订购批量折扣模型 |
| 11.3 | | 经济生产批量存储模型 |
| | 11.3.1 | 基本的经济生产批量存储模型 |
| | 11.3.2 | 带有约束的经济生产批量存储模型 |
| | 11.3.3 | 允许缺货的经济生产批量存储模型 |
| | 11.3.4 | 带有约束的允许缺货模型 |
| 11.4 | | 单周期随机库存模型 |
| | 11.4.1 | 模型的基本假设 |
| | 11.4.2 | 模型的推导 |
| | 11.4.3 | 模型的求解 |
| 第十二章 | | 数学建模竞赛中的部分优化问题 |
| 12.1 | | 一个飞行管理问题 |
| | 12.1.1 | 问题的描述 |
| | 12.1.2 | 模型 1 及求解 |
| | 12.1.3 | 模型 2 及求解 |
| 12.2 | | 钢管订购和运输 |
| | 12.2.1 | 问题的描述 |
| | 12.2.2 | 运费矩阵的计算模型 |
| | 12.2.3 | 运算量计算模型及求解 |
| 12.3 | | 露天矿生产的车辆安排 |
| | 12.3.1 | 问题的描述 |
| | 12.3.2 | 运输计划模型及求解 |

| | |
|--------|---------------|
| 12.4 | 空洞探测 |
| 12.4.1 | 问题的描述 |
| 12.4.2 | 优化模型及求解 |
| 习题 12 | |
| 参考文献 | |

第一章引言

1.1 优化模型的基本概念

1.1.1 优化模型的一般形式

在工程技术, 经济管理, 科学研究和日常生活等诸多领域中, 人们经常遇到的一类决策问题是: 在一系列客观或主观限制条件下, 寻求使所关注的模个或多个指标达到最大(或最小)的决策, 例如, 结合设计要在满足强度要求的条件下选策材料的尺度, 使其总重量最轻; 运输方案要在有限资源约束下制定各用户的分配数量, 使资源产生的总效益最大; 运输方案要在满足物资需求和装载条件下安排从各供应到各需求点的运量和路线, 使运输总费最底; 生产计划要按照产品工艺流程和顾客需求, 制定原料、零件、部件等订购、投产的日程和数量, 尽量降低成本使利润最高。

上述这种决策问题通常称为最优化 (optimization, 简称为解决这些优化问题的手段大致有以下几种:

- (1) 依赖过去的经验判断面临的问题, 这似乎切实可行, 并且没有太大的风险, 但是其处理过程会融入决策者太多的主观因素, 常常难以客观地给予描述, 从而无法确认结果的最优性。
- (2) 做大量的试验反复比较。这固然比较真实可靠, 但是常要花费太多的钱资金和人力, 而且得到的最优结果基本上跑不出开始设计的试验范围。
- (3) 用数学建模 (mathematical modeling) 的方法建立优化模型 (optimization model) 求解最优决策, 我们将这种方式简称为优化模型 (optimization model)。虽然由于建模时要做适当的简化, 可能得结果不一定完全可行或达到实际上的最优, 但是它基于客观规律和数据, 又不需要多大的费用, 具有前两种手段无可比拟的优点。如果在此基础上再辅之以适当的经验和试验, 就可以期望得到实际问题的一个比圆满的回答。优化建模是解决优化问题的最有效, 最常用的方法之一。在决策科学化, 定量化的呼声日益高涨的今天, 用数学建模方法求解优化问题, 无疑是符合时代的潮流和形势发展需要的。

优化模型是一种特殊的数学模型, 油画模型方法是一种特殊的数学建模方法。优化模型一般有以下几个要素:

- (1) 决策变量 (decision variable), 它通常是该问题要求解的那些未知量, 不妨用 n 维向量 $x = (x_1, x_2, \dots, x_n)^T$ 表示, 当对 x 赋值后它通常称为该问题的一个解或一个点。
- (2) 目标函数, 通常是指该问题要优化 (最小或最大) 的那个目标的数学表达式, 它是决策变量 x 的函数, 可以抽象地记作 $f(x)$ 。
- (3) 约束条件, 由该问题对决策变量的限制条件给出, 即 x 允许取值的范围为 $x \in \Omega$, Ω 称为可行域, 常用一组关于 x 的等式 $h_i(x)=0$ ($i=1, 2, \dots, m_e$) 或不等式 $g_j(x) \leq 0$ ($j=m_e+1, m_e+2, \dots, m_e+m$) 来界定, 分别称为等式

约束和不等式约束。

于是，优化模型从数学可表述成如下一般形式：

$$\text{Opt } z=f(x); \quad (1)$$

$$\text{s. t. } h_i(x)=0 \quad (i=1, 2, \cdots, m) \quad (2)$$

$$g_j(x) \leq 0 \quad (j= m_e+1, m_e+2, \cdots, m_e+m) \quad (3)$$

这里 opt 是最优化的意思，可以是 min（求极小，即 minimize 的缩写）或 max（求极大，即 maximize 的缩写）两者之一；s. t. 是“受约束于”的意思。

1.1.2 可行解与最优解

同时满足余数 (2) 和 (3) 的解称为可行解或可行点，否则称为不可行解或不可行点。

满足 (1) 的可行解 x^* （也就是使目标达到最优的 x^* ）称为最优解或最优点，在最优解 x^* 处

目标函数的取值 $f(x^*)$ 称为最优值。对于极小化问题，则对应的最优解（点）也可以称为最小解（点），最优值称为最小值。类似地，对于极大化问题，则对应的最优解（点）也可以称为最大解，最优值称为最大值。

如果在某个可行解 x^* 的附近（ x^* 的某个邻域）， x^* 使目标函数达到最优（即将可行域限定在 x^* 的某个邻域时 x^* 是最优解），但 x^* 不一定是整个可行域 Ω 上的最优解，则 x^* 称为一个局部最优解或相对最优解，此时的所谓最优解实际上知识极值点。相对于极小化问题，图 1—1 中的 x_1 ， x_1 都是局部最优解，我们把整个可行解（最小点），其中 x_1 不是全局最优解，而 x_2 是全局最优解。对大多数优化问题，求全局最优解是很困难的，所以很多优化软件往往只能求到局部最优解。

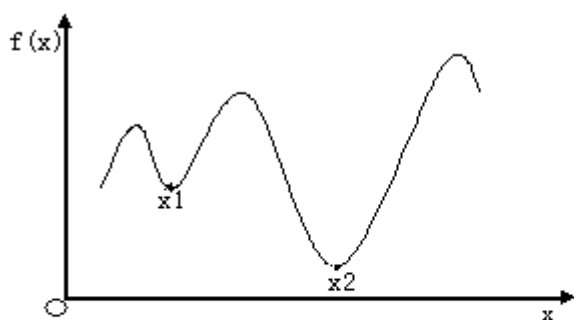


图1-1 局部最优化解与整体最优解

1.1.3 优化模型的基本类型

优化模型可以从不同的角度进行分类。若优化模型中只有 (1) 式而没有 (2)，(3) 式，则这种特殊情况称为无约束化；只有 (2) 或 (3) 式，模型就称为约束化。还有一些更特殊的情况，即只有 (2) 式而没有 (1)，(3) 式，模型就变成了普通的方程组；如果只有 (3)

式而没有(1),(2)式,模型就变成了不等式组;这些都可以看成式约束化的特例。一般说来,实际生活中的优化问题总是没有约束的,但是如果最优解不是在可行域的边界上,而是在它的内部,那么就可以考虑用无约束化来比较简单地处理。另外,在理论和算法上,约束化也式约束优化的基础。

在上面的模型(1),(2),(3)中,除了要求决策变量满足约束(2),(3)外,没有限制决策变量 x 的范围内取值,这是通常表示(默认)决策变量的分量 x_i ($i=1, 2, \dots, n$)

可以在实数范围内取值,即 $x \in R^n$ 。优化问题的另一种分类方法,是按照模型中决策变量

的取值范围以及目标函数 $f(x)$ 和约束函数 $h_i(x)=0$ ($i=1, 2, \dots, m_e$)或不等式 $g_j(x)$

≤ 0 ($j=m_e+1, m_e+2, \dots, m_e+m$)的特性进行感分类。常见的类型如下:

- (1) 当模型中决策变量 x 的所有分量 x_i ($i=1, 2, \dots, n$) 取值均为连续数值(即实数)时,优化模型称为连续优化(continuous optionzation),这也是通常所说的数学规划(mathematical rogramming)。此时,若 f, h_i, g_j 都是线性函数,称为线性规划(linear programming, LP);若 f, h_i, g_j 至少有一个是非线性函数,则称为非线性规划(nonlinear programming, NLP)。特别地,若 f 是一个二次函数,而 h_i, g_j 都是线性函数,则称为二次规划(quadratic programming, QP),

它是一种相对简单的非线性规划。

- (2) 否则,若 x 的一个或多个分量只取离散数值,则优化模型称为离散优化(discrete optimization),或称为组合优化(combinatorial optimization)。这是通常 x 的一个或多个分量只取整数数值,称为整数规划(integer programming, IP),并可以进一步明确地分为纯整数规划(pure integer programming, PIP, 此时 x 的所有分量只取整数数值)和混合整数规划(mixed integer programming, MIP, 此时 x 的部分分量只取整数数值)。特别地,若 x 的分量中取整数数值的范围还限定为只取 0 或 1,则称为 0-1 规划(zero-one programming, ZOP)。此外,与连续优化分程线性规划和非线性规划类似,整数规划也可以分成整数线性规划(ILP)和整数非线性规划(INLP)。

请大家注意,上面括号内的英文中经常出现“programming”(规划)这个词,在与计算机语言连用时它通常时“编程”的意思,如 C++programming(C++语言编程)。但在最优化中,它的意思就是“optimization”(优化),依次偶尔也会有用“optimization”(优化)来代替“programming”(规划)的时候,但反过来通常不行,什么时候用哪个词基本上是约定俗成的,并没有什么特别的道理可言。例如,我们几乎从来不把“组合优化”(combinatorial optimization)说成“整数优化”(integer optimization)。

还可以根据其他标准对优化问题进行分析。例如,根据模型中参数或决策变量是否具有不确定性,可以把优化问题分成确定性规划,不确定性规划(如随机规划,模糊规划等);根据 f, h_i, g_j 是否连续,是否可靠,可以把优化问题分成光滑优化,非光滑优化;根据需要优化目标的多少,把优化问题可以分成单目标优化,多目标优化;此外,还有目标规划,

动态规划，多层规划，等等。总之，出于解决实际问题的需要，人们建立和研究了不同类型的优化问题；反过来，有关优化问题的理论研究成果和所涉及的内容非常丰富，为优化方法的广泛应用提供了支持。

本书不准备对优化理论和方法进行具体，详细地介绍，而是把重点放在如何建立优化模型，然后如何用 LINDO/LINGO 软件来求解所建立的模型和分析所得到的计算结果。即便如此，由于不同类型的优化问题的求解难度和求解方法是有很大大差异的，因此在解决我们所面临的问题时，弄清问题的类型是很有必要的。例如，只能对于连续线性规划或某些特定的二次规划（如凸二次规划）问题，可以比较容易地求到整体最优解，或判断原问题无解；而对于一般的非线性规划和整数规划，当问题的规模比较大时，在可以接受的计算时间内找到整体最优解时非常困难的，因此通常只能求局部最优解。一般来说，离散优化问题比连续优化问题那以求解。对于本书后面选择的求解软件来说，理解下面列出的主要优化类型及其求解难度（如图 1-2 所示）是有帮助的。

对于解决实际优化问题来说，建立其对应的优化模型是极其重要的一步。为了大家对上述类型的优化模型有一个基本的了解，1.2 节将分别介绍一些可以用线性规划，二次规划，非线性规划，整数规划建立模型的实际问题的案例。

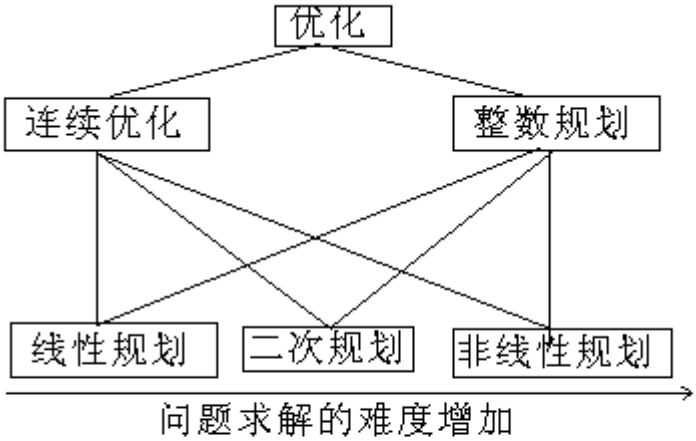


图1-2 优化模型的简单分类和求解难度

1.2 优化问题的建模实例

1.2.1 线性规划模型

下面通过一个加工奶制品的生产计划问题的实例，说明如何建立线性规划模型..

例 1.1 一奶制品加工厂用牛奶生产 A_1 , A_2 两种奶制品，1 桶牛奶可以在甲车间用 12 加工成 3 kg A_1 ，或者在乙车间用 8 h 加工成 4 kg A_2 。根据市场需要，生产的 A_1 , A_2 全部能售出，且每千克 A_1 获得 24 元，每千克 A_2 获得 16 元。现在加工厂每天能得到 50 桶牛奶的供应，每天正式工人总的劳动时间为 480 h，并且甲车间的设备每天至多能加工 100 kg A_1 乙车间的设备的加工能力可以认为没有上限限制（即加工能力足够大）。试为该厂制定一个生产计划，使每天获得最大，并进一步讨论以下 3 个附加问题：

- （1）若用 35 元可以买到 1 桶牛奶，是否作这都是项投资？若投资，每天最少购买多少桶牛奶？
- （2）若可以聘用临时工人以增加劳动时间，会给临时工人的工资资源共享最多是每小时几元？

(3) 由于开市场需求变化, 每千克 A_1 的获利增加到 30 元, 是否应该改变生产计划?

问题分析

这个优化问题的目标是使每天的获利最大, 要作的决策是生产计划, 即每天用多少桶奶生产 A_1 , 用多少桶牛奶生产 A_2 (当然, 不潮湿策变量也可以取每天生产多少年千克 A_1 , 多少千克 A_2 , 得到的模型不会有本质区别) 决策受到 3 个条件的条件的限制: 原料 (牛奶) 供应。劳动时间。甲车间的加工能力, 按照题目所给, 将决策变量, 目标数和约束条件用数学符号及式子表示出来, 就可得到这个优化问题的模型。

优化模型

决策变量: 设每天用小 x_1 桶牛奶生产 A_1 , 用 x_2 桶牛奶 生产 A_2 ,

目标函数: 设每天获利为 z (元)。 x_1 桶牛奶可生产 $3x_1$ (kg) A_1 , 获利 $24 \times 3x_1$, x_2

桶牛奶 可生产 $4x_2$ (kg) A_2 , 获利 $16 \times 4x_2$, 故 $z = 72x_1 + 64x_2$ 。

约束条件

原料供应: 生产 A_1, A_2 的原料 (牛奶) 总量不得超过每天的供应, 即 $x_1 + x_2 \leq 50$ (桶);

劳动时间: 生产 A_1, A_2 的总参加工时间 不得超产过每天正式工人总的劳动时间, 即 $12x_1 + 8x_2 \leq 480$ (h);

设备能力: A_1 的产量不得超产过甲车间役备每天加工能力, 即 $3x_1 \leq 100$;

非负约束: x_1, x_2 均不能为负值, 即 $x_1, x_2 \geq 0$ 。

综上可得

$$\text{Max } z = 72x_1 + 64x_2;$$

$$\text{s. t } x_1 + x_2 \leq 50,$$

$$12x_1 + 8x_2 \leq 480,$$

$$3x_1 \leq 100,$$

$$x_1, x_2 \geq 0。$$

这就是该问题的基本模型, 由于牛奶是任意可分的, 我们可以假设决策变量是在实数范围内取值, 因此这是一个连续规划, 又由于目标函数和约束条件对于决策变量而言都是线性的, 所以是一个 (连续) 线性规划 (LP) 问题。

模型分析与假设

从该实例可以看到, 许多实际的优化问题的数学模型都好是线性规划 (特别是在像生产计划这样的经济管理领域), 这不是偶然的, 让我们分析一下线性规划 具有哪些特征, 或者说, 实际问题具有什么性质, 其模型才是线性规划 。

比例性: 每个决策变量对目标函数的贡献。与该决策变量的取值成正比; 每个决策

变量对每个约束条件右端项的贡献，与该决策变量的取值成正比，

可加性：每个决策变量对目标函数的“贡献”，与其他决策变量的取值无关；每个决策变量对每个约束条件右端项的“贡献”，与其他决策变量的取值无关。

连续性：每个决策变量的取值是连续的。

比例性和可加性保证了目标函数和约束条件对于决策变量的线性性，连续性则允许得到决策变量的实数可行解和实数可行解和实数最优解。

对于本例，能建立上面的线性规划模型，实际上是事先作了如下的假设：

- (1) A_1, A_2 两种奶制品千克的获利是与它们各自产量无关的常数，每桶牛奶加工出 A_1, A_2 的数量和所需的时间是与它们各自的产量无关的常数；
- (2) A_1, A_2 每千克的获利是与它们相互间产量无关的常。每桶牛奶加工出 A_1, A_2 的数量和所需的时间是与它们相互间产量无关的常数。
- (3) 加工 A_1, A_2 的牛奶桶数可以是任意实数（只要不是负数）。

这三假设恰好保证了上面的三条性质，当然，在现实生活中这些假设只是近似成立的，比如 A_1, A_2 的产量很大时，自然会使它们每千克的获利有所减少。由于这些假设对于本书中给出的。经过简化的实际问题是如此明显地成立，本书后面的例题就不再一一列出类似的假设了，不过，读者在打算用线性规划模型解决现实生活中的实际问题时，应该考虑上面三条性质是否近似地满足。

线性规划的解法相对比较简单，为了让大家对有一个简单的理解，下面直观地通过图解法来解这个问题。

模型求解

图解法：这个线性规划模型的决策变量为二维变量，用图解法即简单又直观地把握线性规划的基本性质。

将约束条件 (5) ~ (8) 中的不等号改为等号，可知它们是 $x_1 \sim x_2$ 平面上的 5 条直线，

依次记为 $L_1 \sim L_5$ ，如图 1-3，其中 L_4, L_5 分别是 x_2 轴和 x_1 轴，并且不难判断，(5) ~ (8)

式界定的可行域是 5 条直线上的线段所围成的五边形 OABCD。容易算出，5 个顶点的坐标为：O (0 0)，A (0 50)，B (20 30)，C (100/3, 10)，D (100/3 0)。

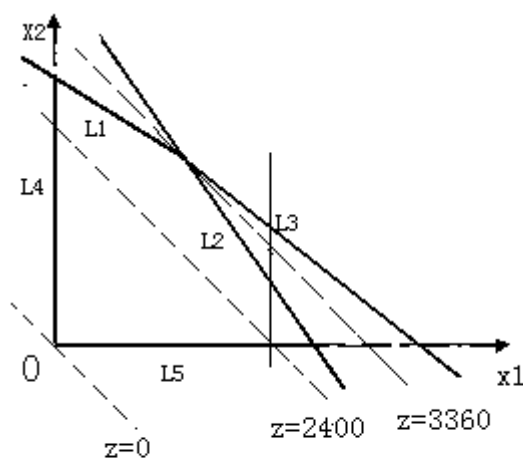


图1-3 线性规划模型的图解法

目标函数 (4) 中的 z 取不同数值时, 在图 1-3 中表示一组平行直线 (虚线), 称等值线族。如 $z=0$ 是过 O 点的直线, $z=2400$ 是过 D 点的直线, $z=3040$ 是过点 C 点的直线……可以看出, 当这族平行线向右上方移动到过 B 点时, $z=3360$, 达到最大值, 所以 B 点的坐标 $(20\ 30)$ 即为最优值: $x_1=20, x_2=30$ 。

在最优点 $B(20\ 30)$ 处, 约束 (5), (6) 是严格取等号的, 此时称它们是积极约束 (active constraints), 或称为是紧约束 (tight constraints), 也就是起作用的约束; 约束 (7), (8) 是严格取不等号的, 此时它们是非积极约束 (inactive constraints), 也是不起作用的约束。显然, 去掉不起作用的约束不会改变模型的最优值。

对于模型中需要讨论的三个问题, 实际上是考虑当模型中的参数发生变化时最优解是否变化, 变化多少的问题, 这种分析称为敏感性分析 (sensitivity analysis)。我们以后将结合 LINDO/LINDO 软件的使用, 再介绍如何作这样的分析。

我们直观地看到, 由于目标函数和约束条件都是线性函数, 在二维的情形, 可行域为直线段围成的凸多边形, 目标函数的等值线为直线, 于是最优化一定在凸多边形的某个顶点取得。

推广到 n 维的情形, 可以猜想, 最优解会在约束条件所界定的一个凸多面体 (可行域) 的某个顶点取得。线性规划的理论告诉我们, 这个猜想是正确的。

可以看出, 由于线性规划的约束条件和目标函数均为线性函数, 所以对于二维的情形, 可行域为直线组成的凸多边形, 目标函数的等值线为直线, 等值线沿着与其垂直似的一个方向 (法线方向, 即等值线是减少的点的梯度方向) 移动时, 函数值是增加的 (沿相反法移动时, 函数值是减少的)。这样, 最优解一定在凸多边形的某个顶点取得。

从二维例子的几何意义可以看出, 除了取得有限的最优值这种情况外, 线性规划的解好可能会有下列情况出现 (见图 1-4):

- (1) 可行域为空集, 原问题不可行 (infeasible)。
- (2) 可行域非空但无机界, 则可能无最优解 (即最优值无界 (unbounded))。注意: 可行域无界时也可能有 (有限的) 最优解。
- (3) 最优解在凸多边形的某一条边上取得则有无穷多个最优解。由于我们一般不关心最优解的数目 (实际上我们很难有效地确定最优解的数目), 主要找到一个最优解就满足了。

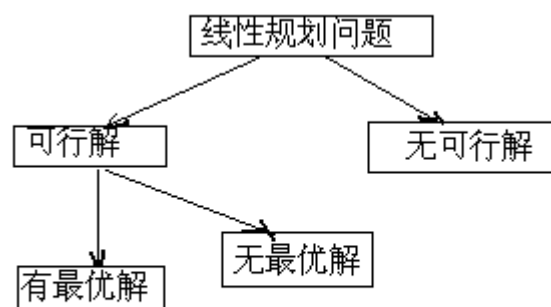


图1-4 线性规划模型的解的几种情况

二维的情况可以推广到多维: 线性规划的可行域是超平面组成的凸多平面体, 等值线是超平面, 最优解在凸多平面体的某个顶点取得。一般认为, 美国数学家 Dantzig 于 1947 年提出的单纯形法 (simplex method) 是最早提出的线性规划的算法, 在 20 世纪 80 年代以前, 它几乎是线性规划的唯一算法。其基本思路是: 将变量的分量分成基变量 (basic variable) 和非基变量 (nonbasic variable), 令非基变量的取值为 0 (模型中变量取值的下界), 此

时的解称为所谓的基本解 (basic solution)。从代数角度看, 基本解对于约束矩阵的一个可逆子矩阵 (称为基矩阵); 从几何上看, 基本解中的可行解正好对应于可行域的顶点。单纯形法以迭代方式从一个顶点 (基本可行解) 转换到另一个顶点, 每一步转换称为一次旋转 (pivot), 每次旋转只将一个非基变量, 称为进基, 同时将一个基变量变为非基变量, 称为离基, 进基和离基的确定使目标函数值不断下降 (对极小化问题)。单纯形法的具体步骤可参阅任何一本有关线性规划的书籍。

20 世纪 80 年代以后, 人们提出了一类新的算法——内点算法 (interior point method)。内点算法也是迭代法, 但不再从可行域的一个顶点转换到另一个顶点, 而是直接从可行域的内部逼近最优解。虽然实践证明单纯形法计算效果很好, 目前仍然经常使用, 但理论上讲内点算法具有单纯形法所不具备的一些优点, 尤其对于特别大规模的问题 (如变量规模上万甚至达到十万, 百万量级), 使用内点算法可能更为有效。内点算法理论较为复杂, 有兴趣的读者请参看有关的专门书籍。

1.2.2 二次规划模型

下面通过一个产品生产销售计划实例, 说明如何建立二次规划模型。

例 1.2 某厂生产的一种产品有甲, 乙两个牌号, 讨论在产销平衡的情况下如何确定各自的产量, 使总的利润最大。所谓产销平衡指工厂的产量等于市场上的销量和 (单件) 价格, 也依赖于产量和 (单件) 成本。按照市场经济归律, 甲的价格 p_1 固会随其销量 x_1 的增长而降低, 同时乙的销量 x_2 的增长也会使甲的价格有稍微的下降, 可以简单地假设价格与销量成线性关系, 即 $p_1 = b_1 - a_{11}x_1 - a_{12}x_2, b_1, a_{11}, a_{12} > 0, a_{11} > a_{12}$; 类似地, 乙的价格 p_2 遵循同样的规律, 即有 $p_2 = b_2 - a_{21}x_1 - a_{22}x_2, b_2, a_{21}, a_{22} > 0, a_{22} > a_{21}$ 。例如, 假定实际中 $b_1 = 100, a_{11} = 1, a_{12} = 0.1, b_2 = 280, a_{21} = 0.2, a_{22} = 2$ 。此外, 假设工厂的生产能力有限, 两种牌号产品的产量之和不可能超过 100 件, 且甲的产量不可能超过乙的产量的两倍, 甲乙的单件生产成本分别是 $q_1 = 2$ 和 $q_2 = 3$ (假定为常数)。求甲、乙两个牌号的产量 x_1, x_2 使总利润最大。

优化模型

决策变量就是甲、乙两个牌号的产量 (也是销量) x_1, x_2 。

目标函数: 显然, 目标函数就是总利润 $z(x_1, x_2)$, 即

$$\begin{aligned} z(x_1, x_2) &= (p_1 - q_1)x_1 + (p_2 - q_2)x_2 \\ &= (100 - x_1 - 0.1x_2 - 2)x_1 + (280 - 0.2x_1 - 2x_2 - 3)x_2 \\ &= 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 2x_2^2. \end{aligned}$$

约束条件: 题中假设工厂的生产能力有限, 两种产品的产量之和不可能超过 100 件, 且产品甲的产量不可能超过乙的产量的两倍。写成数学表达式, 就是

$$x_1 + x_2 \leq 100, x_1 \leq 2x_2.$$

于是，可以得到如下数学规划模型：

$$\begin{aligned} \max \quad & z = 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 2x_2^2; \\ \text{s.t.} \quad & x_1 + x_2 \leq 100, \\ & x_1, x_2 \geq 0. \end{aligned}$$

由于目标函数（9）是一个二次函数，而约束函数（10）~（12）都是线性函数，所以这是一个二次规划（QP）。

如果还要求产量必须是整数，则通常用 $x_1, x_2 \in Z$ 表示（这里 Z 表示全体整数集合），此时就是一个整数二次规划（IQP）。

由于二次规划是一种特殊的非线性规划，因此自然可以直接用求解非线性规划的算法来求解。不过，有一些专门针对二次规划的求解方法，如积极集方法（active set method）是常用的方法。特别是对于凸二次规划，能够有有效的方法求到全局最优解。有关二次规划的更多知识，请读者参阅其他有关的专门书籍。

1.2.3 非线性规划模型

下面通过一个选址问题的实例，说明如何建立非线性规划模型。

例 1.3 某公司有 6 个建筑工地要开工，每个工地的位置（用平面坐标 a, b 表示，距离单位：km）及水泥日用量 d （单位：t）由表 1-1 给出。目前有两个临时料场位于 $P(5,1)$ ， $Q(2,7)$ ，日储量各有 $20t$ 。请回答以下两个问题：

- （1）假设从料场到工地之间均有直线道路相连，试制定每天的供应计划，即从 A、B 两料场分别向各工地运送多少吨水泥，使总的吨公里数最小。
- （2）为了进一步减少吨公里数，打算舍弃目前的两个料场，改建两个新的料场，日储量仍各为 $20t$ ，问应建在何处，与目前相比节省的吨公里数有多大。

表 1-1 工地的位置 (a, b) 及水泥日用量 d

| 料场 | 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 |

优化模型

记工地的位置为 (a_i, b_i) ，水泥日用量为 $d_i, i = 1, 2, \dots, 6$ ；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} 。

决策变量：在问题（1）中，决策变量就是料场 j 向工地 i 的运送量 c_{ij} ；在问题（2）中，决策变量除了料场 j 向工地 i 的运送量 c_{ij} 外，新建料场位置 (x_j, y_j) 也是决策变量。

目标函数：这个优化问题的目标函数 f 是总吨公里数（运量乘以运输距离），所以优化目标可表为

$$\min f = \sum_{j=1}^2 \sum_{i=1}^6 c_{ij} \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2}. \quad (13)$$

约束条件：各工地的日用量必须满足，所以

$$\sum_{j=1}^2 c_{ij} = d_i, i = 1, 2, \dots, 6. \quad (14)$$

各料场的运送量不能超过日用量，所以

$$\sum_{i=1}^6 c_{ij} \leq e_j, j = 1, 2. \quad (15)$$

问题归结为在约束条件 (14)、(15) 及决策变量 c_{ij} 非负的情况下，使目标 (13) 达到最小。当使用临时料场（问题 (1) 中）决策变量只有 c_{ij} ，目标函数和约束关于 c_{ij} 都是线性的，所以这时的优化模型是线性规划模型；当为新建料场选址时（问题 (2) 中），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，所以在新建料场时这个优化模型是非线性规划模型（NLP）。

对于非线性规划问题（包括二次规划问题），通常的解法仍然是迭代法，使目标函数值不断下降（这里指的是最小化问题；如果是最大化问题，当然是使目标函数值不断上升），即从一个初始解甚至不一定是可行解）出发，在可行域中沿着目标函数值下降的某个方向前进到下一个解。不同的算法通常在选择下降方向时采用的方法不同，以及在选定一个方向后所前进的路程（步长）不同，以及判定停止迭代的准则不同，等等。具体的非线性规划方法很多，不同方法可能对具有特定性质的模型求解更有效，理论较为复杂，与感兴趣的读者可参看有关的专门书籍。

一般来说，非线性规划问题比线性规划求解起来要困难得多，除了一些比较特殊的情况外，一般只能找到局部最优解（有时在复杂的实际问题中甚至只要找到一个可行解就满足了）。

1.2.4 整数规划模型

下面通过一个服务员聘用问题的实例，说明任何建立整数规划模型。

例 1.4 某服务部门一周中每天需要不同数目的雇员：周一到周四每天至少需要 50 人，周五至少需要 80 人，周六和周日至少需要 90 人。现规定应聘者需连续工作 5 天，试确定聘用方案，既周一到周日每天聘用多少人，使在满足需要的条件下聘用总人数最少。

优化模型

决策变量：记周一到周日每天聘用的人数分别为 x_1, x_2, \dots, x_7 ，这就是问题的决策变量。

目标函数：目标函数是聘用总人数，即

$$z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7. \quad (16)$$

约束条件：约束条件由每天需要的人数确定。由于每人连续工作 5 天，所以周一工作的雇员应是周四到周一聘用的，按照需要至少有 50 人，于是

$$x_1 + x_4 + x_5 + x_6 + x_7 \geq 50. \quad (17)$$

类似地，有

$$x_1 + x_2 + x_5 + x_6 + x_7 \geq 50, \quad (18)$$

$$x_1 + x_2 + x_3 + x_6 + x_7 \geq 50, \quad (19)$$

$$x_1 + x_2 + x_3 + x_4 + x_7 \geq 50, \quad (20)$$

$$x_1 + x_2 + x_3 + x_4 + x_5 \geq 80, \quad (21)$$

$$x_2 + x_3 + x_4 + x_5 + x_6 \geq 90, \quad (22)$$

$$x_3 + x_4 + x_5 + x_6 + x_7 \geq 90, \quad (23)$$

虽然，人数总应该是整数，所以

$$x_i \geq 0 \quad (i=1, 2, \dots, 7)$$

其中 x_i 是整数。问题归结为在条件 (17) ~ (24) 下求解 $\min z$ 的整数规划模型。由于目标函数和约束条件关于决策变量都是线性函数，所以这是一个整数线性规划模型。

在 1.2.2 节介绍的生产和销售的二次规划模型中，如果我们要求生产的产品数量必须是证书，那么问题将变成了一个证书二次规划模型，这是一种特殊的整数非线性规划模型。

可以用整数规划建立模型的实际问题非常多。例如，实际生活中可能遇到这样的分派或选择问题：若干项任务分给一些候选人来完成，因为每个人的专长不同，他们完成每项任务取得的效益或需要的资源就不一样，如何分派这些任务使获得的总效益最大，或付出的总资源最少？也会遇到这样的选择问题：有若干种策略供选择，不同的策略得到的效益或付出的成本不同，各个策略之间可以有相互制约关系，如何在满足一定条件下作出选择，使得收益最大或最小或成本最小？下面介绍一个结局这种问题的 0-1 规划模型。

例 1.5 某班准备从 5 名游泳队员中选择 4 人组成接力队，参加学校的 $4 \times 100m$ 混合泳接力比赛。5 名队员 4 中泳姿的百米平均成绩如表 1-2 所示，问应如何选拔队员组成接力队？

表 1-2 5 名队员种泳姿的百米平均成绩

| 队员 | 甲 | 乙 | 丙 | 丁 | 戊 |
|-----|---------|--------|--------|--------|--------|
| 蝶泳 | 1'06''* | 57'2 | 1'18'' | 1'10'' | 1'07'4 |
| 仰泳 | 1'15'6 | 1'06'' | 1'07'8 | 1'14'2 | 1'11'' |
| 蛙泳 | 1'27'' | 1'06'4 | 1'24'6 | 1'09'6 | 1'23'8 |
| 自由水 | 58'6 | 53'' | 59'4 | 57'2 | 1'02'4 |

*1'06'8 表示 1 分 6.8 秒，这里沿了习惯表示法。

问题分析

问题要求从 5 名队员中选出 4 人组成接力队，每人一种泳姿，且 4 人的泳姿各不相同，使接力队的成绩最好，容易想到的一个办法是穷举法，组成接力队的方案共有 $5! = 120$ 种，逐一计算并作比较，即可找出最优方案，显然这不是解决这类问题的好办法，随着问题规模的变大，空举法的计算量将是无法接受的。

可以用

可以用 0-1 变量表示一个队员是否入选接力队，从而建立这个问题的 0-1 规划模型，借助现成的数学软件求解。

优化模型

记甲、乙、丙、丁、戊分别为队员 $i=1,2,3,4,5$; 记蝶泳、仰泳、蛙泳、自由泳分别为泳姿 $j=1,2,3,4$. 记队员 i 的第 j 种泳姿的百米最好成绩为 $c_{ij}(s)$ ，则表 1-2 可以表示成表 1-3.

表 1-3 5 名队员 4 种泳姿的百米平均成绩

| c_{ij} | $i=1$ | $i=2$ | $i=3$ | $i=4$ | $i=5$ |
|----------|-------|-------|-------|-------|-------|
| $j=1$ | 66.8 | 57.2 | 78 | 70 | 67.4 |
| $j=2$ | 75.6 | 66 | 67.8 | 74.2 | 71 |
| $j=3$ | 87 | 66.4 | 84.60 | 69.6 | 83.8 |
| $J=4$ | 58.6 | 53 | 59.4 | 57.2 | 62.4 |

决策变量：引入 0-1 变量 x_{ij} 若选择队员 i 参加泳姿 j 的比赛，否则 $c_{ij}x_{ij} = 0$ ，于是接

力队的成绩可以表示为 $f = \sum_{j=1}^4 \sum_{i=1}^5 c_{ij}x_{ij}$ ，这就是该问题的目标函数。

约束条件：根据组成接力队的要求， x_{ij} 应该满足下面两个约束条件：

(1) 每人最多只能入选 4 种泳姿之一，即对于 $i=1,2,3,4,5$, 应有 $\sum_{j=1}^4 x_{ij} \leq 1$.

(2) 每种泳姿必须有 1 人而且只能有 1 人入选，即对于 $j=1,2,3,4$, 应有 $\sum_{i=1}^5 x_{ij} = 1$.

综上所述，这个问题的优化模型可写作

$$\begin{aligned}
 \min \quad & f = \sum_{j=1}^4 \sum_{i=1}^5 c_{ij}x_{ij}; \\
 s.t. \quad & \sum_{j=1}^4 x_{ij} \leq 1, i = 1,2,3,4,5, \\
 & \sum_{i=1}^5 x_{ij} = 1, j = 1,2,3,4, \\
 & x_{ij} = \{0,1\}.
 \end{aligned}$$

这是一个线性 0-1 规划模型，它是一个特殊线性整数规划。一般来讲，对于整数规划问题，即使是线性 0-1 规划问题，当问题的规模比较大时，求解也是非常困难的。虽然整数规划问题的可行解通常只能有有限多个（如果可行域有界的话），可以通过枚举所有可行解比较出最优解，但是对于规模限制，求解相应的线性规划或非线性规划问题（称为原问题对应的松弛问题），然后将得到的解四舍五入到最接近的整数？在有些情况下，尤其当解的分量是非常大的实数时，可能这些解对四舍五入不太敏感，那么这一策略可能是可行的，但在许多实际应用中，整数变量的取值并不太大，如 0-1 规划问题中整数只取 0 或 1，因此这一方法往往行不通。此外把松弛问题的解四舍五入到一个可行的整数解并非易事（参见例 1.6），

有时甚至找这样一个可行解与求解原问题本身的难度可能是一样的。

例1. 6 求解整数规划问题

$$\begin{aligned} \max \quad & z = 5x_1 + 8x_2; \\ \text{s.t.} \quad & x_1 + x_2 \leq 6, \\ & 5x_1 + 9x_2 \leq 45, \quad IP \\ & x_1, x_2 \geq 0 \text{ 且为整数.} \end{aligned}$$

解 将去掉整数限制后的松弛问题记叙 LP，其可行域为图 1-5 中由点 (0, 0), (6, 0), p(2.25,3.75),(0,5)围成的四边形，过 P 点的等值线（图中点划线）为 $z = z_{\max}$ ，最优解在 P 点取得，图中小圆点为整数点（也称为格点），四边形中的小圆点为原问题 IP 的可行解。

为了求 IP 的最优解，交款 P 舍入成整数或者找最靠近它的整数，都行不通。经过在可行解中试探、比较可以得到表 1-4。可见 IP 的最优解不一定能从 LP 的最优解经过简单的“移动”得到。

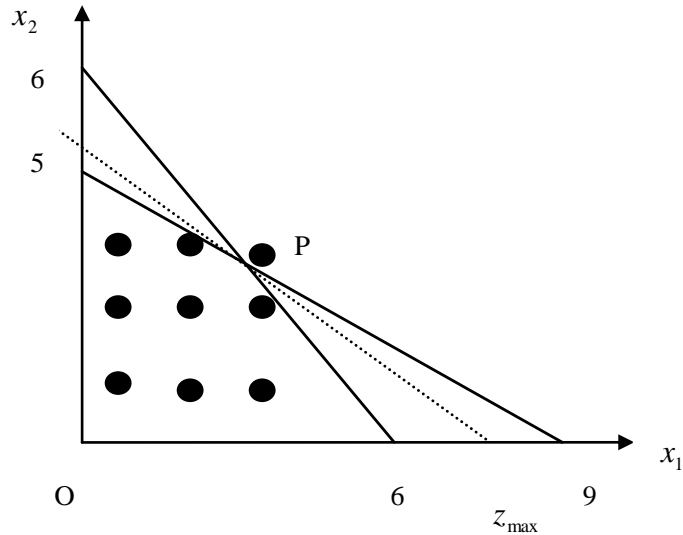


表 1-4 不同解的效果比较

| LP 最优解为 P | P 的舍入解 | 最靠近 P 的可行解 | IP 最优解 |
|----------------|--------|------------|--------|
| (2. 25, 3. 75) | (2,4) | (2,3) | (0,5) |
| Z=41.25 | 不可行 | Z=34 | Z=40 |

求解整数规划没有统一的有效方法,不同方法的效果与问题的性质有很大关系.比较常用的一种求解方法是分支定界法 (branch and bound method)，可看作对枚举法的一种改进。分支定界法采用“分而治之”的策略求解整数规划，其基本思想是隐式地枚举一切可行解。自然，这种枚举不是简单的完全枚举，而是以一种比较“聪明”的方式进行的，即逐次对解空间进行划分。所谓分支，指的就是这个划分过程；而所谓定界，是指对于每个划分在求解过程中判定是否需要目前的解空间进一步划分，也就是尽可能去掉一些明显的非最优点，从而避免完全枚举。分支定界算法的实际计算效果取决于具体的分支策略和定界方法，有兴趣的读者看有关的专门书籍。

1. 2. 5 其他优化模型

实际中的许多优化决策问题，并不一定像上面的例子中的问题那样，很明显、很简单地

就能建立形如 (1)、(2)、(3) 的优化模型。这时可能需要利用较多的数学知识和技巧, 才能将决策问题转化、描述成形如 (1)、(2)、(3) 的优化模型。例如, 和生产和运作管理中的决策问题、经济与金融中的决策问题、图论和网络优化相关的决策问题、目标规划问题、对策论中的决策问题、排队论中的决策问题、存储论中的决策问题, 以及更加综合、更加复杂的决策问题等。对这些方面的决策问题和优化建模方法, 我们将介绍过 LINGO/LINGO 软件的使用方法之后, 从第 5 章一个专题一个专题地分别通实例进行介绍和讨论。

1. 3 LINDO/LINGO 软件简介

1. 3. 1 LINGO/LINGO 软件的基本功能

美国芝加哥大学的 linus Schrage 教授于 1980 年前后开发了一套专门用于求解最优化问题的软件包, 后来又经过了多年的不断完善和扩充, 并成立了 LINDO 系统公司 (LINDO Systems Inc.) 进行商业化动作, 取得了巨大成功, 这套软件的主要产品有 4 种: LINDO, LINGO, LINDO API 和 WHAT'SBEST!, 在最优化软件的市场上占有很大的份额, 尤其在供微机上使用的最优化软件的市场上, 上述软件产品具有绝对的优势。例如, 根据 LINGO 公司主页 (HTTP://WWW.LINGO.CON) 上提供的信息, 位列全球《财富》杂志 500 强的企业中, 一半以上使用上述产品, 其中位列全球《财富》杂志 25 强的企业中有 23 家使用上述产品, 读者可以从该公司的主页了解更多的信息, 特别是可以从这个网页上下载上面 4 种软件的演示版 (试用版) 和大量应用例子。演示版与正式版的基本功能是类似的, 只是试用版能够求解问题的规模 (即决策变量和约束的个数) 受到严格限制, 对于规模稍微大些的问题就不能求解。即使对于正式版式, 通常也被分成求解包 (SOLVER SUITE)、高级版 (SUPER)、超级版 (HYPER)、工业版 (INDUSTRIAL)、扩展版 (EXTENDED) 等不同档次的版式本, 不同档次的版本的差别也在于能够求解的问题的规模大小不同 (参见表 1-5, 只列出了 LINGO 软件的规模限制; LINGO 软件也类似, 只是不能有非线性约束)。当然, 规模越大的版式本的销售价格也越昂贵 (不过, 所有的正式版对教育机构都有特殊的优惠价)。

表 1-5 不同版本 LINGO 程序对求解规模的限制

| 版本类型 | 总变量数 | 整数变量数 | 非线性变量数 | 约束数 |
|------|-------|-------|--------|-------|
| 演示版 | 300 | 30 | 30 | 150 |
| 求解包 | 500 | 50 | 50 | 250 |
| 高级版 | 2000 | 200 | 200 | 1000 |
| 超级版 | 8000 | 800 | 800 | 4000 |
| 工业版 | 32000 | 3200 | 3200 | 16000 |
| 扩展版 | 无限 | 无限 | 无限 | 无限 |

说明: 通过 “HELP|ABOUT...” 菜单命令, 就能知道你所安装的 LINGO/LINDO 软件的版本和所能求解的规模了限制等相关的信息。

LINDO 是英文 LINEAR INTERACTIVE AND DIACRETE OPTIMIZER 字首的缩写形式, 即 “交互式的线性离散优化求解器”, 可以用来求解线性规划 (LP) 和二次规划 (QP); LINGO 是英文 LINEAR INTERACTIVE AND GENERAL OPTIMIZER 字首的缩写形式, 即 “交互式的线性和通用优化求解器”, 它除了具有 LINGO 的全部功能外, 还可以用于求解非线性规划, 也可以用于一些线性和非线性方程组的求解。LINDO 和 LINGO 软件的最大特色在于可以允许决策变量是整数 (即整数规划, 包括 0—1 规划), 而且执行速度很快。LINGO 实际上还是最优化问题的一种建模语言, 包括许多常用的数学函数供使用者建立优化模型时调用, 并可以接受其他数据文件 (如文本文件、Excel 电子表格文件、数据库文件等), 即使对优化方面的专业知识了解不多的用户, 也能够方便地建模和输入、有效地求解和分析实际中遇到的大规模优化问题, 并通常能够快速得到复杂优化问题的高质量解。LINDO 和 LINGO 软件能求解的优化模型参见图

1—6。

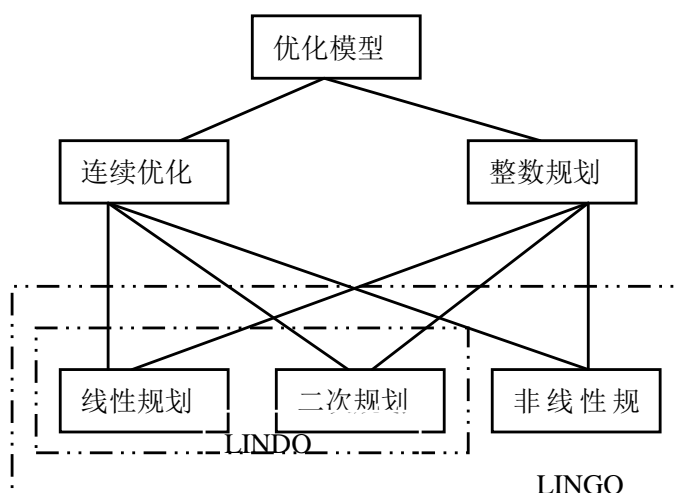


图 1-6 LINDO 和 LINGO 软件能求解的优化

此外，LINDO 系统公司还提供了 LINDO 和 LINGO 软件与其他开发工具（如 C++ 和 Java 等语言）的接口软件 LINDO API (LINDO application program interface)，因此使 LINDO 和 LINGO 软件还能方便地融入到用户应用软件的开发中去。最后，What'sBest! 实际上提供了 LINDO 和 LINGO 软件与电子表格软件（如 Excel 等）的接口，能够直接集成到电子表格软件中使用。

由于上述特点，LINDO、LINGO、LINDO API 和 What'sBest! 软件在教学、科研和工业、商业、服务等领域得到了广泛的应用。本书只详细介绍在 Microsoft windows 环境下运行的 LINDO 和 LINGO 软件最新版本 (LINDO6.1、LINGO9.0) 的使用方法，并包括社会、经济、工程等方面的大量实际应用问题的数学建模和求解实例。需要指出的是，目前 LINGO9.0 的功能完全包含了 LINDO 软件的功能（包括模型程序的书写格式），所以 LINDO 公司已经将 LINDO 软件从产品目录中删除，这意味着以后不会再有 LINDO 软件的新版本了。考虑到国内目前仍有不少读者使用 LINDO 软件，而且 LINDO 软件比起 LINGO 软件更容易入门和掌握，所以我们在本书中还是对 LINDO 软件进行一定的介绍，但把全书重点放在 LINGO 软件的使用上。

1.3.2 LINDO/LINGO 软件的求解过程

在进入后续章节的学习之前，先简单介绍一下 LINDO/LINGO 软件求解一个优化模型的过程。实际上，使用 LINDO/LINGO 时，软件本身并不需要用户知道软件内部的这些算法实际和调用方法。不过，如果对此有所了解，对于更有效地利用 LINDO/LINGO 软件解决实际问题，理解软件的运行状态和阅读结果报告、分析出错信息等，还是有所帮助的。

LINDO/LINGO 软件内部有以下 4 个基本的求解程序用于求解不同类型的优化模型（参见图 1—7）：

- (1) 直接求解程序 (Direct Solver)。
- (2) 线性优化求解程序 (Linear Solver)。
- (3) 非线性优化求解程序 (Nonlinear Solver)。
- (4) 分支定界管理程序 (Branch and Bound Manager)。

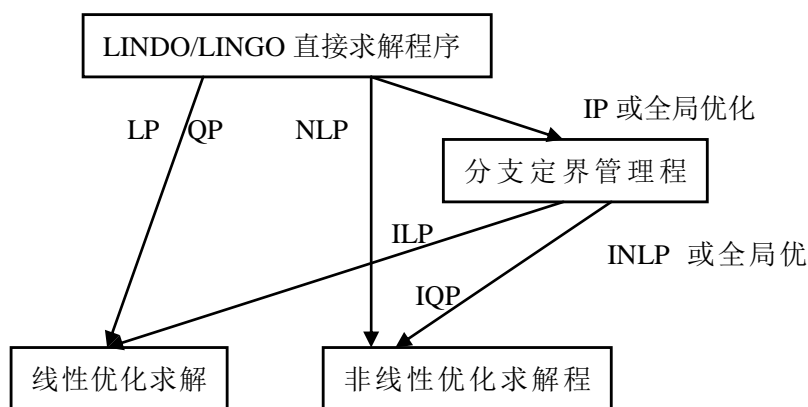


图 1-7 LINDO/LINGO 软件的求解过程

当用户在 LINDO/LINGO 软件中输入完一个优化模型，要求 LINDO/LINGO 软件求解时，LINDO/LINGO 软件首先调用直接求解程序对模型进行一系列的直接处理。主要有以下两个功能：

1. 对等式约束的直接处理

如果一个等式约束只含有一个变量，那么这个变量的值就可以直接确定下来，以后 LINDO/LINGO 软件在求解模型时就把这个变量看成固定变量（常数），而不认为它是决策变量了。这个过程是循环进行的，例如，如果约束中有以下三个等式：

$$xyz=30,$$

$$x+y=8,$$

$$y=5,$$

则 LINDO/LINGO 软件能直接确定出固定变量 $y=5$ ，然后可以确定出 $x=3$ ，最后确定出 $z=2$ ，这三个变量就是常量了，而且这三个约束也就不看成是约束了。显然，经过这样的处理，将尽量减小模型中实际需要求解的决策变量和需要满足的约束条件的个数，可能会使问题的规模有所下降，从而使求解更有效。

2. 识别优化模型的类型

对等式约束的直接处理完成后，直接求解程序将对输入的模型进行分析，自动识别模型的数学结构和性质，确定优化模型的类型，从而决定下一步采取什么求解程序。如果模型是（连续）线性规划，下一步将直接调用线性优化求解程序；如果模型是（连续）非线性规划，下一步将直接调用非线性优化求解程序；如果模型是整数规划，则下一步将直接调用非线性优化求解程序，主要是通过调用分支定界管理程序进行求解。分支定界管理程序主要用于管理整数规划的分支定界算法，在进行中还要不断调用线性优化求解程序和非线性优化求解程序进行定界处理。整数规划求解程序本身通常还有一个预处理程序，主要是为了对整数线性规划模型生成有效的割平面（有效的割平面可以显著地减少求解整数规划所需要的计算时间）。

线性优化求解程序通常使用单纯形算法，为了能解大规模的问题，也提供了内点算法（LINGO 中一般称为障碍法，即 barrier）以备选用。非线性规划求解程序采用的是顺序线性规划法（sequential linear programming, SLP），即通过迭代求解一系列线性规划来逼近、达到求解非线性规划的目的。非线性规划求解程序也可以使用其他的算法，如顺序二次规划法（sequential quadratic programming, SQP），广义既约梯度法（generalized reduced gradient, GRG），并可以让软件自动选择多个初始点开始进行多次求解（LINGO 中称为 multistart），以便通过找到多个局部最优解增加找到全局最优解的可能性。LINGO 中配备了全局优化求解程序（global solver），用户能要求 LINGO 软件求非线性规划问题的全局最

优解，主要的思想是把原问题分解成一系列的凸规划（理论上，凸规划的局部最优解就是全局最优解），这时也要通用分支定界管理程序进行控制。不过，内点算法、GRG 算法、多初始点求解和全局优化功能通常不是 LINGO 软件的标准配置，而是可选的，用户需要额外付费购买。

1.3.3 建立 LINDO/LINGO 优化模型需要注意的几个基本问题

在开始讨论 LINDO/LINGO 的软件使用之前，我们特别指出建立 LINDO/LINGO 优化模型需要注意的几个基本问题。虽然 LINDO/LINGO 软件的功能非常强大，但这并不是说只要将模型随随便便地输入软件中去求解就万事大吉了。事实上，在利用 LINDO/LINGO 求解优化模型之前，为了将来 LINDO/LINGO 能快速地得到高质量的解，对实际问题建立优化模型就必须特别仔细。下面给出值得注意的几个问题。

1. 尽量使用实数优化模型，尽量减少整数约束和整数变量的个数

只有在万不得已时才使用整数变量和整数约束（即含有整数变量的约束）。原因前面已经讲过，求解离散优化问题比连续优化问题难得多。

2. 尽量使用光滑优化模型，尽量避免使用非光滑函数

非光滑函数是指存在不可微点的函数。例如，应尽量少地使用绝对值函数、符号函数（如当变量 x 为正数时取 1，为 0 时取 0，为负数时取 -1）、多个变量求最大（或最小）值、四舍五入函数、取整函数等。这些函数从数学上看是不光滑的，含有尖点（不可微点）甚至间断点（函数的不连续点），因此从数学上看不利于利用其导数信息。我们在前面也已经讲过，求解非光滑优化比光滑优化难得多。

3. 尽量使用线性优化模型，尽量减少非线性约束和非线性变量的个数

应当尽量简化变量之间的约束关系。例如，对于“变量 x 与 y 的比值不超过 5”这样的约束，写成“ $x/y \leq 5$ ”当然是可以的，但这是一个分式约束（同时它还含有间断点 $y=0$ ），因此是非线性约束， x ， y 都是非线性变量（即非线性约束中的变量），最好改写成线性约束“ $x \geq 5y$ ”。原因还是我们在前面已经讲过的，求解非线性模型问题比模型问题难。

4 合理设定变量的上下界，尽可能给出变量的初始值

如果在实际问题中知道变量的取值范围，那就尽量告诉 LINDO/LINGO，不要让软件帮你到大海中去捞针。例如，如果 x 的取值范围在实际中是大于 30 小于 50，就不要让软件在整个实数范围内去寻求最优解。有时实际问题中还能知道或感觉到最优解大致在哪个解附近，那就可以以初始值的形式告诉 LINDO/LINGO，这对于问题的求解是很有帮助的。毕竟，软件是死的，而人要比计算机聪明得多。

5. 模型中使用的单位数量级要适当

如果用一模型甚至同一约束中有的数很小而有的数很大，如 0.01 和 10000000000，则这两个数量级相差太大了，不利于优化模型求解，因为大的数和小的数运算时误差会很大，运算精度降低。LINDO/LINGO 通常希望模型中数据之间的数量级不要相差超过 1000（即最大数与最小数（按绝对值）不要超过 1000 倍以上），否则会给出警告提示信息。有时，可以通过对数据选择适当的单位改变相对尺度（scaling），尽量使数据之间的数量级相差减少。

习题 1

1.1 能否将下面的非线性规划问题等价地转化成线性规划问题？

$$(1) \min \quad x_1^2 + x_2 + 4x_3;$$

$$S. \quad t. \quad 4x_1^2 + x_2 + x_3 \geq 5,$$

$$\begin{aligned}
 (2) \quad & \min \quad x_1^2 - x_2^2 + 4x_3 + 5x_4; \\
 & s.t. \quad 4x_1^2 - 4x_2^2 + x_3 - x_4 = 5, \\
 & \quad \quad -x_3 + 2x_4 \leq 10, \\
 & \quad \quad x_1 \geq 0, x_2 \leq -2, x_3 \geq 0, x_4 \geq 0.
 \end{aligned}$$

$$\begin{aligned}
 (3) \quad & \min \quad |x_1| + 2|x_2| + |x_3|; \\
 & s.t. \quad x_1 + x_2 - x_3 \leq 10, \\
 & \quad \quad x_1 - 3x_2 + 2x_3 = 12.
 \end{aligned}$$

$$\begin{aligned}
 (4) \quad & \min \quad |x_1 - 5| + 2|x_2 + 4|; \\
 & s.t. \quad x_1 + x_2 \leq 10, \\
 & \quad \quad x_1 - 3x_2 \geq 2.
 \end{aligned}$$

$$\begin{aligned}
 (5) \quad & \min \quad \max(x_1, x_2, x_3); \\
 & s.t. \quad x_1 + x_2 - x_3 \leq 10, \\
 & \quad \quad x_1 - 3x_2 + 2x_3 = 12.
 \end{aligned}$$

$$\begin{aligned}
 (6) \quad & \max \quad \min(x_1 - 5, x_2 + 4); \\
 & s.t. \quad x_1 + x_2 \leq 10, \\
 & \quad \quad x_1 - 3x_2 \geq 2.
 \end{aligned}$$

1.2 要从给定的数 c_1, c_2, \dots, c_n 中寻找最大的数，对这个问题建立线性规划模型。

1.3 给定 3 个不同的数 a, b, c ，要找出中位数（即给出的数中不是最大也不是最小的哪个数 x ）。建立这个问题的优化模型，并讨论能否把这个问题建模成线性规划模型。

1.4 你有 100 个 25 美分的硬币和 90 个 10 美分的硬币，没有其他钱币。你必须付给定的款额 C ，不找零给你。用数学形式建立这段话的优化问题。这个问题是线性规划吗？分别对于 $C=15$ 美分， $C=\$1.02$ ， $C=\$100$ 求解。

1.5 例 1.1 给出的 A_1, A_2 两种奶制品的生产条件、利润及工厂的“资源”限制全都不变。为增加工厂的获利，开发了奶制品的深加工技术：用 2h 和 3 元加工费，可将 1kg A_1 加工成 0.8kg 高级奶制品 B_1 ，也可以将 1kg A_2 加工成 0.75kg 高级奶制品 B_2 ，每千克 B_1 能够获利 44 元，每千克 B_2 能够获利 32 元。试为该厂制定一个生产销售计划，使每天的净利润最大。建立该问题的优化模型，并讨论应该如何分析以下问题：

(1) 若投资 30 元可以供应 1 桶牛奶，投资 3 元可以增加 1h 劳动时间，是否应作这些投资？若每天投资 150 元，可多赚回多少元？

(2) 每千克高级奶制品 B_1, B_2 的获利经常有 10% 的波动，对制定的生产销售计划有无影响？若每千克 B_1 的获利下降 10%，计划应该变化吗？

1.6 在当前普片具有健康意识的时代，许多人在分析食物的营养成分。选择不同食

物的组合作为食谱的一般方法是：以最小费用来满足对基本营养的需求。当然，这类实际问题是相当复杂的，我们必须从营养学家那里知道什么是基本营养需求（可能因人而异），另外，为了保持多样性避免对营养食物的厌倦，应该考虑一个很长的可选择食物的名单。线面的例子做了很大的简化。

按照营养学家的建议，一个人一天对蛋白质、维生素 A 和钙的需求如下：50kg 蛋白质、4000IU（国际单位）维生素 A 和 1000mg 钙。我们只考虑以下食物构成的食谱：生的带皮苹果、生的香蕉、生成胡萝卜、切粹并去核的枣和新鲜的生鸡蛋，他们所含的营养成分和收集到的这些食物的价格如下表 1-6 所示。如果可能的话，确定每种食物的用量，以最小费用满足推荐的每日定额。请用这些数据建立一个优化模型。

表 1-6 食物的营养成分和价格

| 食物 | 单位 | 蛋白质/g | 维生素 A/IU | 钙/mg | 价格/元 |
|-----|-------------------|-------|----------|------|------|
| 苹果 | 中等大小一个 (138kg) | 0.3 | 73 | 9.6 | 1 |
| 香蕉 | 中等大小一个 (118kg) | 1.2 | 96 | 7 | 1.5 |
| 胡萝卜 | 中等大小一个 (72kg) | 0.7 | 20253 | 19 | 0.5 |
| 枣 | 一杯 (178kg) | 3.5 | 890 | 57 | 6 |
| 鸡蛋 | 中等大小一个 (44kg) | 5.5 | 279 | 22 | 0.8 |

1.7 某架货机有三个货舱：前仓、中仓、后仓。三个货舱所能装载的货物的最大重量和体积都有限制，如表 1-7 所示。为了保持飞机的平衡，三个货舱中实际装载货物的重量必须与其最大容许重量成比例。现有四类货物供该货机本次飞行装运，其有关信息如表 1-8 所示，最后一列指装运后所获得的利润。应如何安排装运，使该获机本次飞行获利最大？请建立该问题的优化模型。

表 1-7 三个货舱装载货物的最大容许重量和体积

| | 前仓 | 中仓 | 后仓 |
|-------------|------|------|------|
| 重量限制/t | 10 | 16 | 8 |
| 体积限制/ m^3 | 6800 | 8700 | 5300 |

表 1-8 四类装运货物的信息

| | 重量/t | 空间/(m^3 /t) | 利润/(元/t) |
|------|------|----------------|----------|
| 货物 1 | 18 | 480 | 3100 |
| 货物 2 | 15 | 650 | 3800 |
| 货物 3 | 23 | 580 | 3500 |
| 货物 4 | 12 | 390 | 2850 |

1.8 某投资公司经理欲将 50 万元基金用于股票投资，股票的收益使随机的。经过慎重考虑，他从所有上市交易的股票中选择了 3 种股票作为候选的投资对象。从统计数据的分析得到：股票 A 每股的年期望收益为 5 万，标准差为 2 万元；股票 B 每股的年期望收益为 8 元，标准差为 6 元；股票 C 每股的年期望收益为 10 万，标准差为 10 元；股票 A、B 收益的相关系数为 5/24，股票 A、C 收益的相关系数为-0.5，股票 B、C 收益的相关系数为-0.25。

目前股票 A、B、C 的市价分别为每股 20 元、25 元、30 元，在投资时可以用收益的方差或标准差衡量风险（关于随机变量的期望、标准差、相关系数等概念可参看任何一本概率论的教科书。）

- (1) 如果该投资人期望今年得到至少 20% 的投资回报，应如何投资可以使风险最小？
- (2) 投资回报率与风险的关系如何？

1.9 某房地产开发商准备在两片开发区上分别圈出一块长方形土地，并砌围墙将这两块土地分别围起来。每块土地的面积不得小于 $1000m^2$ ，围墙的高度不能低于 2m。能够用于砌围墙的每块砖使一样的，每块的高度为 10cm，长度为 15cm（假设砖的宽度就是围墙的宽度）。该开发商希望用 10 万块砖，使圈出的两块土地的面积之和最大，问应该如何圈地？如果两块土地不要求是长方形，而是三角形，结果如何？请你分别建立优化模型。

1.10 经济学中著名的柯布——道格拉斯（Cobb-Douglas）生产函数的一般形式为：

$$Q(K, L) = aK^\alpha L^\beta, \quad 0 < \alpha, \beta < 1,$$

其中 Q, K, L 分别表示产值、资金、劳动力，式中 α, β, a 要由经济统计数据确定。现有《中国统计年鉴（2003）》给出的统计数据如表 1-9（其中总产值取自“国内生产总值”，资金取自“固定资产投资”，劳动力取自“就业人员”），请建立优化模型求式中的 α, β, a ，并解释 α, β 的含义。

表 1-9 经济统计数据

| 年份 | 总产值/万亿元 | 资金/万元 | 劳动力/亿人 |
|------|---------|--------|--------|
| 1984 | 0.7171 | 0.0910 | 4.8179 |
| 1985 | 0.8964 | 0.2543 | 4.9873 |
| 1986 | 1.0202 | 0.3121 | 5.1282 |
| 1987 | 1.1962 | 0.3792 | 5.2783 |
| 1988 | 1.4928 | 0.4754 | 5.4334 |
| 1989 | 1.6909 | 0.4410 | 5.5329 |
| 1990 | 1.8548 | 0.4517 | 6.4749 |
| 1991 | 2.1618 | 0.5595 | 6.5491 |
| 1992 | 2.6638 | 0.8080 | 6.6152 |
| 1993 | 3.4634 | 1.3072 | 6.6808 |
| 1994 | 4.6759 | 1.7042 | 6.7455 |
| 1995 | 5.8478 | 2.0019 | 6.8065 |
| 1996 | 6.7885 | 2.2914 | 6.8950 |
| 1997 | 7.4463 | 2.4941 | 6.9820 |
| 1998 | 7.8345 | 2.8406 | 7.0637 |
| 1999 | 8.2068 | 2.9854 | 7.1394 |
| 2000 | 9.9468 | 3.2918 | 7.2085 |
| 2001 | 9.7315 | 3.7314 | 7.3025 |
| 2002 | 10.4791 | 4.3500 | 7.3740 |

第 2 章 LINDO 软件的基本使用方法

2.1 LINDO 入门

2.1.1 LINDO 软件的安装过程

使用 LINDO 软件前，首先需要在操作系统下安装 LINDO 软件，笔者完成本书所使用的操作系统是 Windows XP 的简体汉字版，LINDO 软件是 LINDO6.1 for Windows 试用版。

LINDO 软件非常容易安装，只需要在 Windows 操作系统下将安装光盘插入光驱，运行其中的安装程序就可以了，目前从 LINDO 系统公司或其他渠道得到的安装程序，多数情况下直接是一个自解压的可执行性文件（如 LINDO 系统公司下载的 LINDO6.1 for Windows 试用版安装程序为 LND61.exe，大致是 3M 多一些），那么就直接运行程序进行安装就可以了。

安装过程中，用户只需要接受用户协议，然后选择将 LINDO 软件安装到的目的地（一般是硬盘上的某个目录），屏幕将提示默认的安装目录（默认的目录通用是 C:\LINDO61），您可以任意修改，如此反复，完成后屏幕将提示你确认你的选择。确认你的选择之后，安装程序就会自动完成全部后续安装过程。安装过程成功结束后，你就可以在 Windows 操作系统下运行 LINDO 软件了。

第一次运行刚安装的 LINDO 软件时，系统会弹出一个对话框（图 2-1），要求你输入密码（Password）。如果你买的是正版软件，请在密码框中 LINDO 公司给你提供的密码（如果密码已经被复制到 Windows 剪贴板中，则可以使用粘贴命令从 Windows 剪贴板中将密码拷贝到图 2-1 的对话框中），然后按“OK”按钮即可。

查看安装 LINDO 软件的硬盘目录，通常可以看到其中有一个名为 SAMPLES 的子目录，该子目录下有很多名为*.ltx 的文件，后缀 ltx 表明这些文件是 LINDO 文本（LINDO text）文件，每个文件是一个线性规划的小例子，对初学者学习 LINDO 的使用很有帮助，所以请大家使用最好记住这个子目录的位置。

我们以后均假设 LINDO 软件已经正确地安装完毕，所以直接介绍其用法。

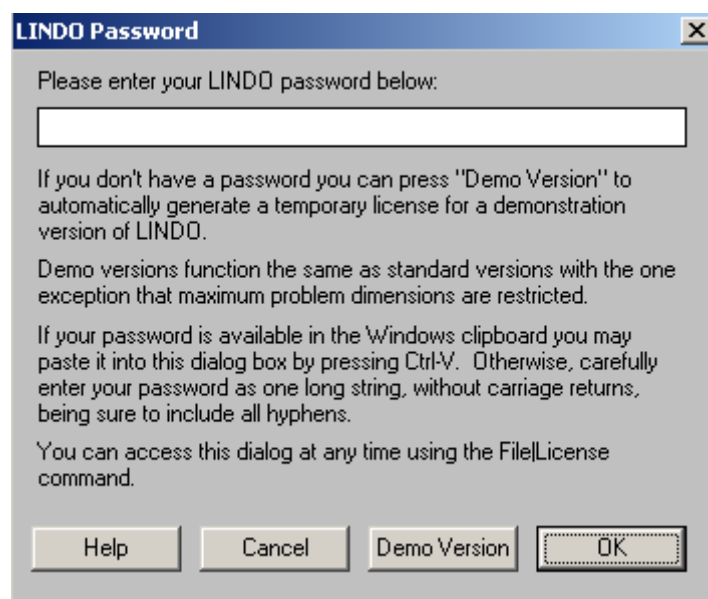


图 2-1 LINDO 要求你输入密码

2.1.1 编写一个简单的 LINDO 程序

下面通过一个非常简单的例子，说明如何编写、运行一个 LINDO 程序的完整过程。

在 Windows 操作系统下双击 LINDO 图标（或在 Windows “开始” 菜单的程序中选择运行 LINDO 软件），可以启动 LINDO 软件，屏幕上首先显示如图 2-2 所示的工作窗口。

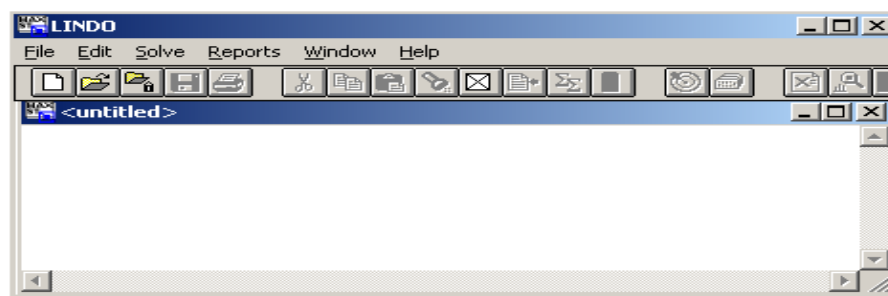


图 2-2 LINDO 初始界面

这就是 LINDO 的初始用户界面，目前光标所在的子窗口称为模型窗口（model windows），是用来供用户输入 LINDO 程序的。所谓一个 LINDO 程序，就是用 LINDO 软件所要求的语法格式对一个优化模型的完整描述，因此一个 LINDO 程序也就是一个 LINDO 优化模型。这两者在 LINDO 中可以认为是一回事，所以下面区分这两个概念。

目前这个模型窗口标有“<untitled>”字样，表示用户还没有为这个程序命名，因此 LINDO 采用了一个自动生成的名字“untitled”，将来用户在保存时可以对它重新命名。

例 2.1 让我们来解如下的简单的线性规划（LP）问题：

$$\text{Max } z=2x+3y; \quad (1)$$

$$\text{s.t. } 4x+3y \leq 10, \quad (2)$$

$$3x+5y \leq 12, \quad (3)$$

$$X,y \geq 0. \quad (4)$$

我们可以直接在<untitled>这个新的、空白的模型窗口中输入这个 LP 模型（图 2-3）。

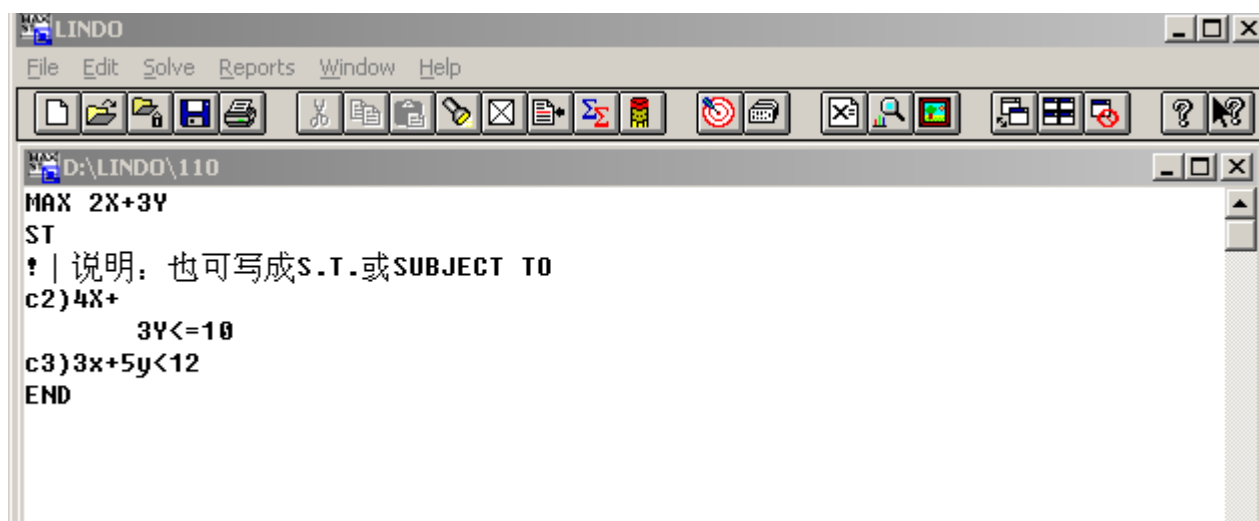


图 2-3 输入一个简单的优化模型

我们看到这段程序（LINDO 优化模型）有以下特点：

- (1) 这个 LINDO 程序以“MAX”开始，表示目的是最大化问题（容易想到，对最小化问题，自然应该用“MIN”开始），后面直接写出目标函数的表达式（目标函数和约束之间用“ST”分开）。程序以“END”结束（请注意：“END”在这里也可以省略）。
- (2) 上面的 LINDO 输入格式与数学模型（1）~（4）的表达式几乎一样，连系数与变量之间的乘号也一样省略了（而且必须省略）。
- (3) 输入的 LINDO 模型中用右括号“)”结束的“C2)”和“C3)”是行名（对于约束，

就是约束名)；用户也可以分别输入“2)”和“3)”等其他行名；行名放在对于的约束之前。请注意：约束不一定非命名不可，也就是说上面程序中的“2)”和“3)”可以省略，省略时 LINDO 会按照输入行的顺序自动生成用数字表示的行名（即行号）。如本例中若输入时省略行名时，系统对约束默认的行名分别是“2)”和“3)”，并对目标函数所在行自动生成行名“1)”。

(4) 我们输入上面模型时故意写得歪七扭八，是为了说明在 LINDO 中模型书写起来是相当灵活的：由于 LINDO 中已假设所有的变量都是非负的，所以非负约束(4)($x, y \geq 0$)不必再输入到计算机中；LINDO 也不区分变量中的大小写字符（实际上任何小写字符将会转换为大写字符）；约束条件中的“<=”及“>=”可分别用“<”及“>”代替；输入的多余的空格和回车也会被忽略；一个约束还可以分成两行甚至多行写；等等。

(5) 一行中感叹号“!”后面的文字将被认为是说明语句（注释语句），不参与模型的

建立；主要的目的是为了增强程序的可读性。


现在我们可以用 LINDO 软件来求解这个模型。用鼠标单击 LINDO 软件工具栏中的 ，或从菜单中选择 Solve | Solve(ctrl+S)命令（即 LINDO 的主菜单“Solve(求解)”中的“Solve(求解)”命令，快捷键是 Ctrl+S,以后我们约定都这样表示），则 LINDO 开始编译这个模型，编译没有错误马上开始求解求解时会显示如图 2-4 所示的 LINDO 求解器运行状态窗口（LINDO Solve Status），其中显示的相应信息的含义见表 2-1。注意，LINDO 求解线性规划的过程默认采用单纯形法，一般是首先寻求一个可行解，在有可行解条件下再寻求最优解。用 LINDO 求解一个 LP 问题会得到如下的结果：不可行或可行；可行时又可分为：有最优解和解无界两种情况。因此替 2-4 中当前状态（Status）除 Optimal（最优解）外，其他显示还有三个：Feasible(可行解)，Infeasible(不可行)，Unbounded(最优值无界)。

表 2-1 LINDO 求解器运行状态窗口显示的相应信息及含义

| 名称 | 含义 |
|-------------------------------|--|
| Status (当前状态) | 显示当前求解状态：“Optimal”表示已经达到最优解；其他显示的可能还有三个：Feasible, Infeasible, Unbounded |
| Iterations (迭代次数) | 显示迭代次数：“2”表示经过了 2 次迭代 |
| Infeasibility (不可行性) | 约束不满足的量（及各个约束条件不满足的“数量”的和；特别注意不是“不满足的约束条件个数”)：“0”表示这个解是可行的 |
| Objective （当前的目标值） | 显示目标函数当前的值：7.45455 |
| Best IP (整数规划当前的最佳目标值) | 显示整数规划当前的最佳目标值：“N/A”（No Answer 或 Not Applicable）表示无答案或无意义，因为这个模型中没有整数变量，不是整数规划 |
| IP Bound (整数规划的界) | 显示整数规划的界(对最大化问题显示界；对最小化问题，显示下界)：“N/A”含义同上 |
| Branches （分支数） | 显示分支定界算法已经显示的分支书：“N/A”含义同上 |
| Elapsed Time (所用时间) | 显示计算所用时间（单位：S）：“0.00”说明时间太快了，用时还不到 0.005S |
| Update Interval （刷 | 显示所用的时间间隔：“1”表示 1S；用户可以直接在界面上修改这 |

| | |
|----------------------------------|---|
| 新本界面的时间间隔) | 个时间间隔 |
| Interrupt Solver (中断求解程序) | 当模型规划比较大是 4 (尤其对整数规划), 可能求解时间会很长, 如果不想再等待下去时, 可以在程序运行过程中用鼠标单击该按钮终止计算。求解结束后这个按钮变成灰色, 再单击就不起作用了 |
| Close (关闭) | 该按钮只是关闭状态窗口, 并不终止计算。如果你关闭了状态窗口, 将来随时可以选择 Window Open Status Window 菜单命令来再次打开这个窗口 |

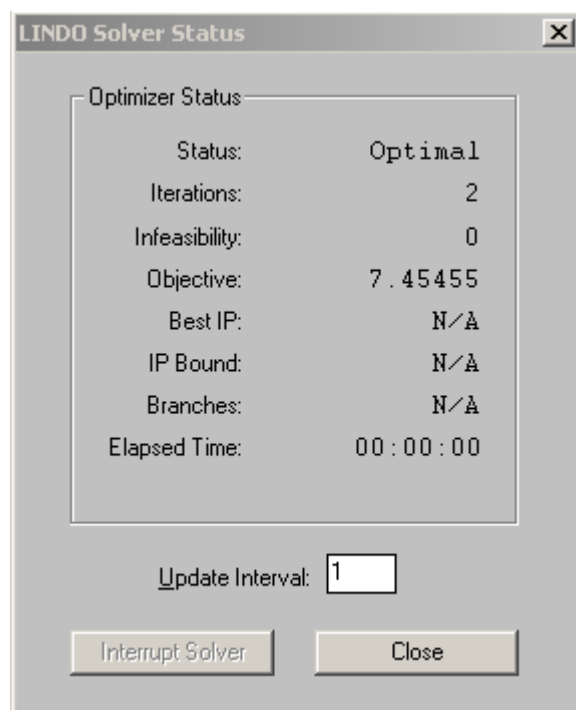


图 2-4 LINDO 运行状态窗口

由于这个例子中的 LP 模型的规划太小了, 我们可能还没来得及看清图 2-4 的界面, LINDO 就解出了最优解, 并马上弹出如图 2-5 的对话框。这个对话框询问你是否需要作灵敏性分析 (DO RANGE (SENSITIVITY) ANALYSIS?) 我们现在先选择“否 (N)”按钮, 这个窗口就会关闭。然后, 我们再把图 2-4 的状态窗口也关闭 (按下图 2-4 的“Close”按钮即可)。



图 2-5 灵敏性分析对话框

现在这个模型就解完了, 那么最优解在哪里呢? 如果你在屏幕上没有看到求解的结果, 那么请你用鼠标选择 LINDO 的主菜单“Window(窗口)”, 你会发现有一个自菜单项“Reports Windows (报告窗口)”, 这就是最终结果的报告窗口。用鼠标选择“Window | Reports

Windows”，就可以查看该窗口的内容（图 2-6）。这些输出结果表示的意思如下：

“LP OPTIMUM FOUND AT STEP 2”表示单纯形法在两次迭代（旋转）后得到的最优解。

“OBJECTIVE FUNCTION VALUE 1) 7.4545450”表示最优目标值为 7.4545450。（注意：在 LINDO 中目标函数所在的行总是被认为是第 1 行，这就是这里 1）的含义）。

“VALUE”给出最优解中各变量（VARIABLE）的值：X=1.272727，Y=1.636364。

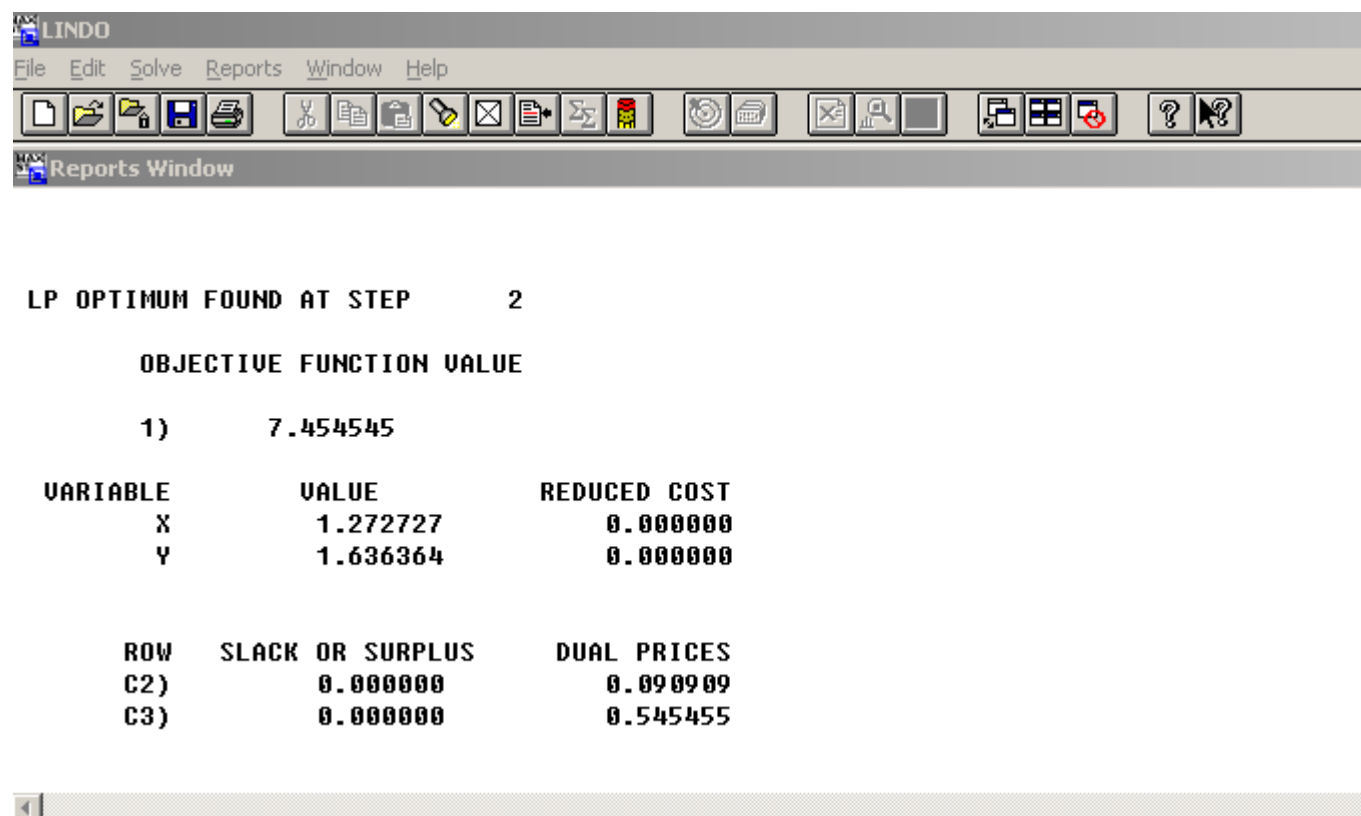



图 2-6 LINDO 的结果报告窗口

“REDUCED COST”给出最优的单纯形表中目标函数行（第 1 行）中变量对应的系数（即各个变量的检查数（也称为判别数））。其中基变量的 REDUCED COST 值一定为 0；对于非基变量（注意：非基变量本身取值一定为 0），相应的 REDUCED COST 值表示当该非基变量增加一个单位（其他非基变量保持不变）时目标减少的量（对 Max 型问题）。本例最优解中两个变量都是基变量，所以对应的 REDUCED COST 的值均为 0。

“SLACK OR SURPLUS（松弛或剩余）”给出约束对应的松弛变量的值：第 2、3 行的松弛变量均为 0，说明对于最优解来讲，两个约束（第 2、3 行）均取等号，即都是紧约束。

“DUAL PRICES”给出对偶价格的值：第 2、3 行对偶价格分别为 0.090909，0.545455。（含义我们以后再介绍）。

“NO. ITERATIONS= 2”表示用单纯形法进行了两次迭代（旋转）。

我们现在可以用鼠标单击工具栏中的图标或选择 File | Save(F5)命令把这个结果报告保存在一个文件中（默认的后缀名为 1tx，即 LINDO 文本文件），以便以后调出来查看。

类似地我们可以回到前面的模型窗口（图 2-3），把我们输入的模型也保存在一个文件中（如保存在文件“exam0201.ltx”中）此时模型窗口中的标题<untitled>将变成文件名 exam0201.ltx）。保存的文件将来可以用 File | Save(F3)和 File | Save(F4)重新打开，用前者打开的程序可以进行修改，而后者只可以浏览。

如果您不想继续使用 LINDO，现在可以选择 File | Exit (Shift+F6) 命令退出 LINDO。现在我们归纳一下上面介绍的输入求解 LP 问题的一般步骤如下：

- (1) 在模型窗口中输入一个 LP 模型，模型以“MAX”或“MIN”开始，按线性规划问题的自然形式输入（如前面例子所示）。若要结束一个模型的输入，只需输入“END”（也可以省略）。
- (2) 求解模型。如果 LINDO 报告有编译错误，则回到上一步修改模型。
- (3) 查看结果，储存结果和模型。

2.1.2 一些注意事项

我们前面已经看到，LINDO 软件对模型的输入格式要求与线性要求问题的自然形式（数学形式）非常类似，几乎没有什么差别，因此几乎不需要专门学习就可以掌握，不过，LINGO 软件对模型的输入格式还是有一些特殊规定的，这些规则值得引起特别注意，我们下面就简单释一下使用 LINGO 软件建立线性规划模型的一些特殊事项：

(1)、LINDO 中的变量名由字母和数字组成，但必须以字母开头，且长度不超过 8 个字符（只能是英文字符，不能含有中文字符）。LINDO 中不区分大小写字母，包括 LINDO 本身的关键字（如 MAX、MIN 等）也不区分大小写字母。

(2)、LINDO 中对优化模型的目标和约束用行号（行名）进行标识，这些标识会在将来的求解结果报告中用到。用户没有指定行号（行名）时，系统将自动产生行号，将目标函数所在行作为第一行，从第 2 行起为约束条件。用户也可以人为定义行号或行名，行号或行名总是以“)”结束，放在相应的约束条件之前；行号或行名

(3)、在 LINDO 模型的任何地方都可以用“TITLE”语句对输入的模型命令，用法是在 TITLE 后面写出其名字（最多 72 个字符，可以有汉字），在程序中单独占一行。请看下面两个例子：

TITLE Example Modle for Chpter 2

TITLE 第 2 章的第一个例子

前者将模型命名为“Example Modle for Chpter 2”后者将模型命名为“第 2 章的一个例子”。实际上这类似于对模型的注释和说明，这是模型命名的第一作用。

对模型命名的另一个目的，是为了方便将来阅读求解结果报告。因为用户有可能同时处理多个模型，很容易混淆模型与求解结果的对应关系。这时如果对不同模型分别进行了命名，就可以随时（例如在求解当前模型前）使用菜单命令“FILE/Title”将当前模型的名字显示在求解结果报告窗口中，这样就容易判别每个求解结果与模型的对应关系。

此外，LINDO 模型中以感叹号（“!”）开头的注释行（注释语句，或称为说明语句），可以帮助他人或自己以后理解这个模型。实际上，每行中“!”符号后面的都是注释或说明。例如：

! This is a comment.

3x+5y <12 ! 这是一个约束。

第一行完全是注释语句；第 2 行则后半部分为注释语句。可以看出，注释语句中也可以有汉字，但是领头的感叹号“!”必须是英文字符，否则会出现错误。

再次总结、提醒一下：行号、“TITLE”语句和注释语句，是 LINDO 中惟一可以使用汉字字符的地方。

(4) LINDO 中变量不能出现一个约束条件的右端（即约束条件的右端只能是常数）；

变量与其系数间可以有空格（甚至回车），但不能有任何运算符号（包括乘号“*”等）。

(5)、LINDO 中不能接受括号“()”和逗号“,”等任何符号（除非在注释语句中），例如：400 (X1+X2) 需写为 400X1+400X2；“10, 000”需写为 10000。

(6)、LINDO 中表达式应当已经经过化简，如不能出现 $2X1+3X2-4X1$ ，而应写成 $-2X1+3X2$ 等

(7)、LINDO 中已假定所有变量非负。可在模型的“END”语句后面用命令“FREE”（设定自由变量）取消变量的非负假定。起用法是“FREE”后面跟变量名，例如，在“END”语句后输入下面命令，可将变量 vname 的非负假定取消：

FREE vnmne

(8)、可以在模型的“END”语句后面用命令“SUB”（即设置上界（set upper bound）的英文缩写）设定变量的上界，用命令“SLB”（即设置下界（set lower bound）的缩写）设定变量的上下界。其用法是：“SUB vname value”将变量 vname 的上限设定为 value；“SLB”的用法类似。例如：

Sub x1 10 ! 作用等价于 “x1 ≤10”

SLB x2 20 ! 作用等价于 “x2 ≥20”

使用“SUB”和“SLB”表示的上下界约束不计入模型的约束，因此 LINDO 也不能给出其松紧判断和敏感性分析。

(9)、数值均衡化及其他考虑：如果约束系数矩阵中各非零元的绝对值的数量级差别很大（相差 1000 倍以上），则称其为数值不均衡的。为了避免数值不均衡引起的计算问题，使用者应尽可能自己对矩阵的行列进行均衡化。此时还有一个原则，即系数中非零元的绝对值不能大于 100000 或者小于 0.0001。LINDO 不能对 LP 中的系数自动进行数值均衡化，但如果 LINDO 觉得矩阵之间很不均衡，将会给出警告。

(10)、简单错误的检查和避免

当你将一个线性规划问题的数学表达式输入 LINDO 系统时，有可能式子中会带有某些输入错误。这类错误虽可能只是抄写和输入错误造成的，但当问题规模较大时，要搜寻它们也是比较困难的。在 LINDO 中有一些客人帮助寻找错误的功能，其中之一就是菜单命令“Report/Picture (Alt+5)”，它的功能是可以将目标函数和约束表达式中的非零系数通过列表（或图形）显示出来。

例2. 2 对 2-7 中的输入，用 Report/Picture 命令，将弹出一个对话框（图 2-8）；在弹出的对话框中采取默认选项（即不采用下三角矩阵形式，而以图形方式显示），直接按“OK”按钮可得到图 2-9 的输出，可以从图 2-9 很直观地发现，其实错误原因只不过是图 2-7 中的输入中，5) 行的表达式中 CO 与 C0 弄混了（英文字母 O 与 0 弄混了）。在图 2-9 中还可以用鼠标控制显示图形的缩放，这对于规模较大的模型是有用的。

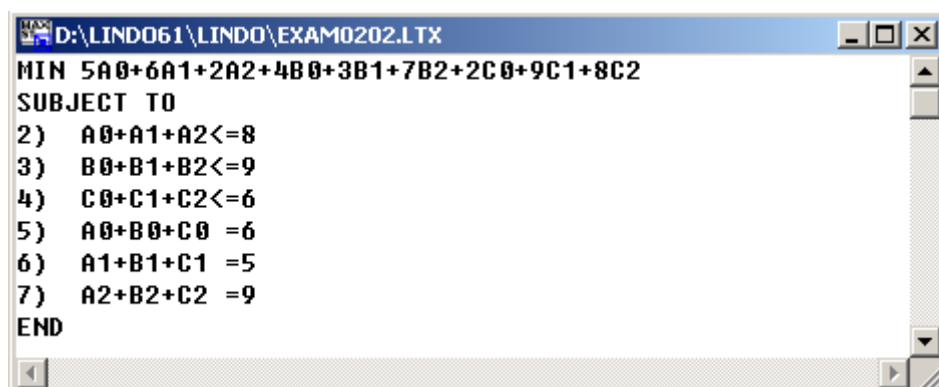


图 2-7 一个输入中含有错误的例子

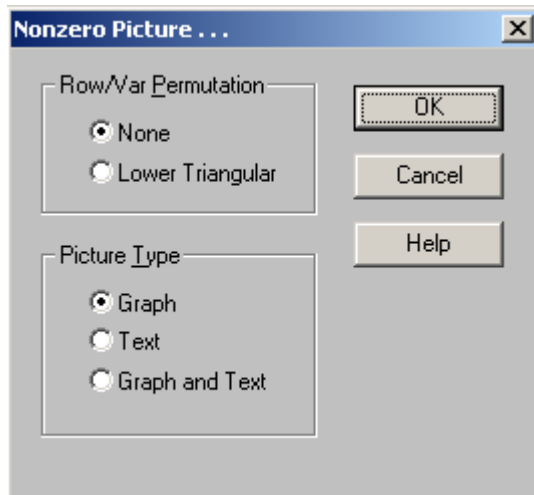


图 2-8 系数矩阵显示方式的控制对话框

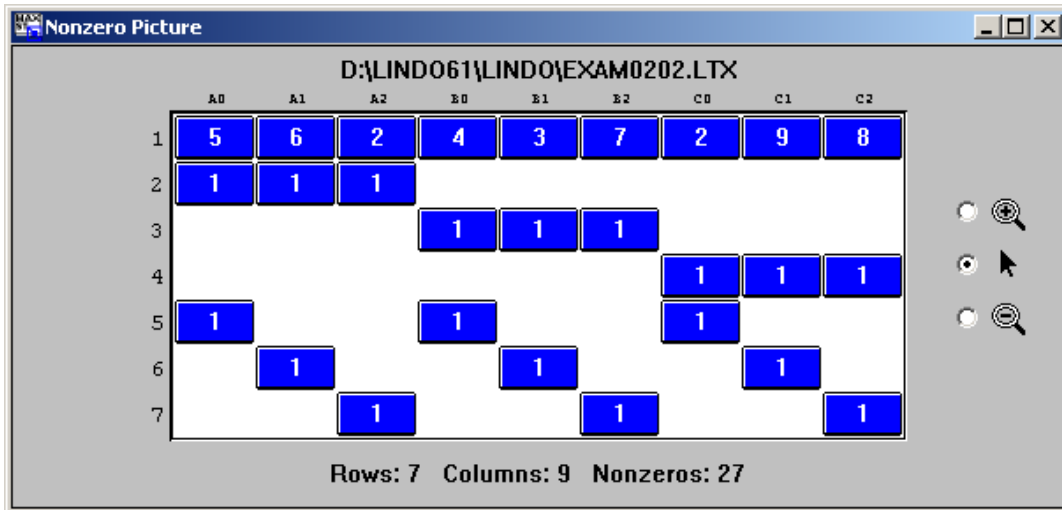


图 2-9 系数矩阵的图形显示

下面我们用一个例子，说明三个变量范围限定命令（FREE、SUB、SLB）的作用。

例 2. 3 来解如下的线性规划（LP）问题：

$$\text{Max } 2x-3y+4z; \quad (5)$$

$$\text{S. t } 4x+3y+2z \leq 10, \quad (6)$$

$$-3x+5y-z \leq 12, \quad (7)$$

$$x+y+5z \geq 8, \quad (8)$$

$$-5x-y-z \geq 2, \quad (9)$$

$$0 \leq y \leq 20, \quad z \geq 30. \quad (10)$$

这个模型中对变量 x 没有非负限制,对 y 有上限限制,对 z 有下限限制.用 FREE、SUB、SLB 三个命令可以实现这些功能,具体输入如图 2-10 所示.

求解得到的结果如图 2-11 所示,即最大值为 122,最优解为 $x=-17, y=0, z=39$.可

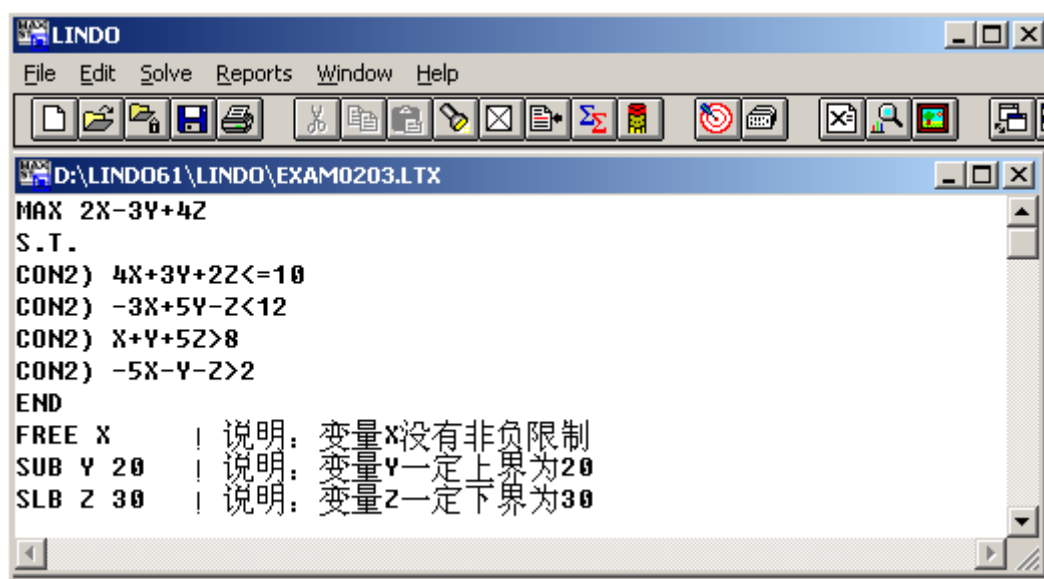


图 2-10 例 2.3 的输入模型

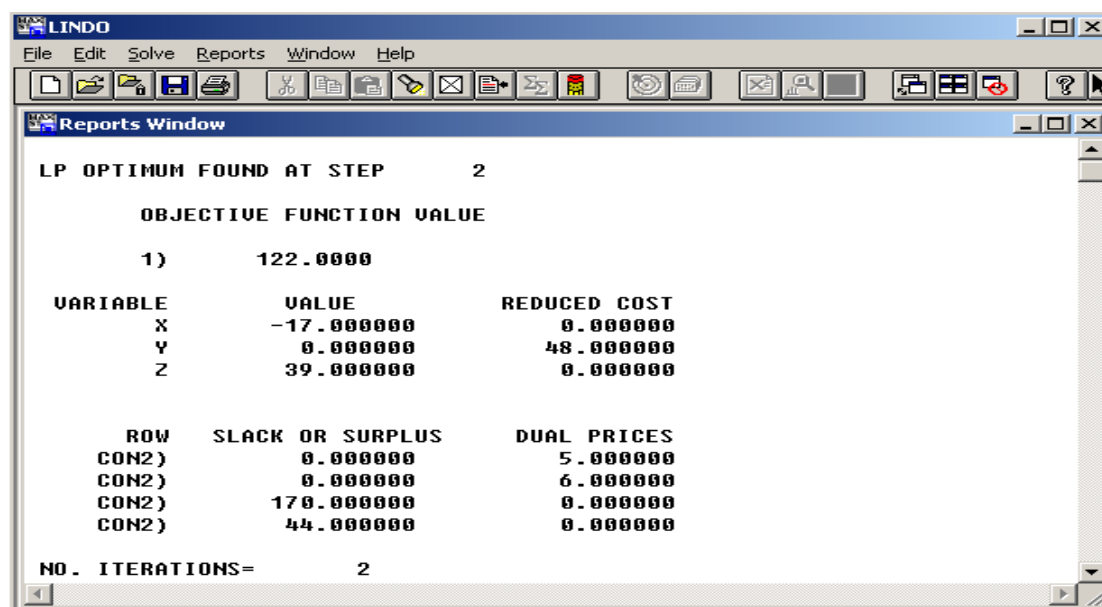


图 2-11 例 2.3 的输出结果

以看 y 的上界(20)在最优解中并没有达到, z 的下界(30)也没有达到,因此模型中去掉"sub y 20"和"slb z 30"两个语句,得到的结果应该是不变的.但由于最优解 x 的取值为负值,所以"free x "这个语句确实是不能少的.读者不妨试一下,去掉这个语句后效果会怎样?这时你会发现模型没有可行解.

2.2 敏感性分析

下面来看一个简单的具体例子.

例 2.4 某家具公司制造书桌、餐桌和椅子,所用的资源有三种:木料、木工和漆工.生产数据如表 2-2 所示.若要求桌子产量不超过 5 件,如何安排三种产品的生产可使利润最大?

表 2-2 家具公司的基础数据

| 每个书桌 | 每个餐桌 | 每个椅子 | 现有资源总 |
|------|------|------|-------|
|------|------|------|-------|

| | | | | |
|---------|-------|--------|--------|-------|
| 木料 | 8 单位 | 6 单位 | 1 单位 | 48 单位 |
| 漆工 | 4 单位 | 2 单位 | 1.5 单位 | 20 单位 |
| 木工 | 2 单位 | 1.5 单位 | 0.5 单位 | 8 单位 |
| 单价 成品单价 | 60 单位 | 30 单位 | 20 单位 | |

用 DESKS、TABLES 和 CHAIRS 分别表示三种产品的生产量(决策变量),容易建立 LP 模型.首先在 LINDO 模型窗口中输入模型,见图 2-12.



图 2-12 家具生产问题的输入模型

求解这个模型,并对图 2-5 的对话框(DO RANGE (SENSITIVITY) ANALYSIS?)选择”是(Y)”按钮,这表示你需要作灵敏性分析.这时,查看报告窗口(Reports Window),可以看到结果如图 2-13 所示.图 2-13 中前半部分的输出结果的解释与 2.1 的结果(图 2-6)类似:

“LP OPTIMUM FOUND AT STEP 2”表示两次迭代(旋转变换)后得到最优解.

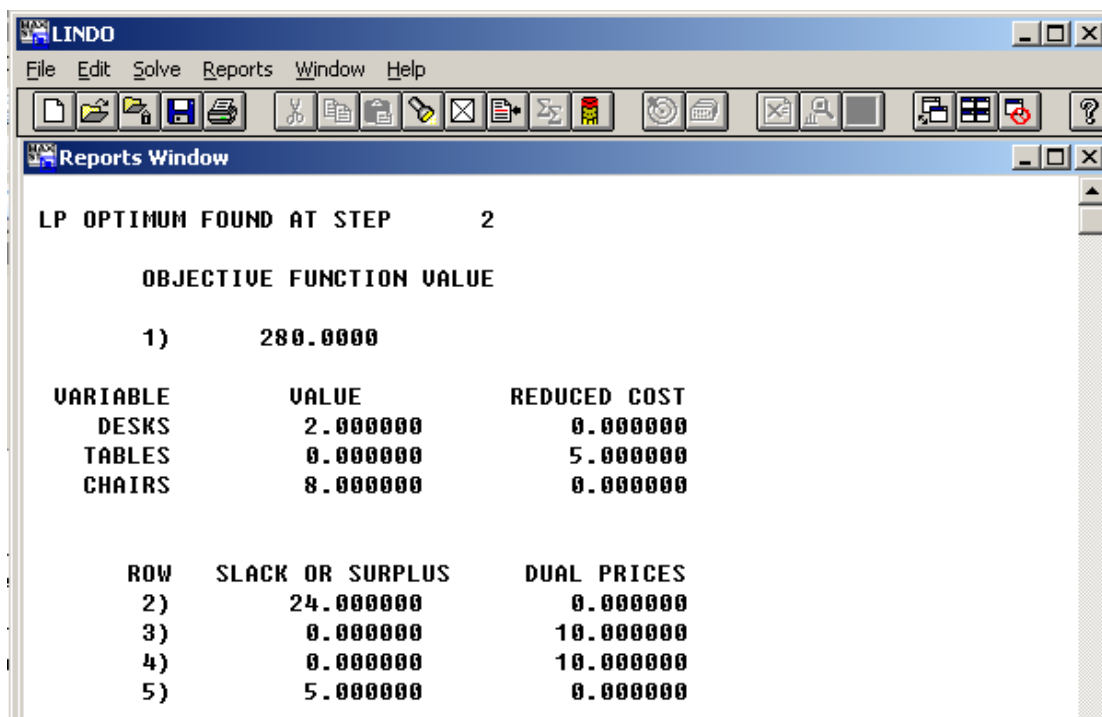


图 2-13 家具生产问题的输出结果

“OBJECTIVE FUNCTION VALUE 1)280.000”表示最优目标值为 280.

“VALUE”给出最优解中各变量的值:造 2 个书桌(DESKS),0 个餐桌(TABLES).8 个椅子(CHATRS).
所以 DESKE、CHAIRS 是基变量(取值非 0),TABLES 是非基变量(取值为 0)。

“SLACK OR SURPLUS”给出松弛变量的值:

第 2 行松弛变量=24; (模型第 1 行表示目标函数,所以第行对应第 1 个约束。)

第 3 行松弛变量=0;

第 4 行松弛变量=0;

第 5 行松弛变量=5;

“REDUCED COST”列出最优单纯形表中判别数所在行的变量的系数,表示当变量有微小变动时,目标函数的变化率。其中基变量的 REDUCED COST 值应为 0,对于非基变量 X_j (请注意,非基变量的取值一定是 0),相应的 REDUCED COST 值表示当某个变量 X_j 增加一个单位时目标函数减少的量(max 型问题)。本例中,变量 TABLES 对应的 REDUCED COST 值为 5,表示当非基变量 TABLES 的值从 0 变为 1 时(此时假定其他变量保持不变,但为了满足约束条件,基变量显然会发生变化),最优的目标函数值=280-5=275.

“DUAL PRICE”(对偶价格)表示当对应约束有微小变动时,目标函数的变化率。输出结果中对应于每一个约束有一个对偶价格。若其数值为 p ,表示对应约束中不等式右端项若增加 1 个单位,目标函数将增加 p 个单位(max 型问题)。显然,如果在最优解处约束正好取等号(也就是“紧约束”,即起作用约束),对偶价格值才可能不是 0。本例中,第 3、4 行是紧约束,对应的对偶价格值为 10,表示当紧约束

3) 4 DESKS+2TABLES+1.5CHAIRS \leq 20

变为

3) 4 DESKS+2TABLES+1.5CHAIRS \leq 21

时,目标函数值=280+10=290.对第 4 行也可类似解释.

对于非紧约束(如本例中第 2、5 行是非紧要约束),DUAL PRICE 的值为 0,表示对应约束中不等式右端项的微小扰动不影响目标函数。有时,通过分析 DUAL PRICE,也可对产生可行问题的原因有所了解。

图 2—13 中后半部分的输出结果是敏感性分析结果(如果来求解模型时你没有要求 LINDO 作敏感性分析,现在想获得敏感性分析也不必从头开始重新求解模型,可直接用菜单命令“Reports| Range”)。敏感性分析的作用是给出“RANGES IN WHICE THE BASIS IS UNCHANGED”,即研究当目标函数的系数和约束右端项在什么范围变化(此时假定其他系数保持不变)时,最优基(矩阵)保持不变.报告中 INFINITY 表示正无穷.这个部分包括两方面的敏感性分析内容:

(1) 目标函数中系数变化的范围(OBJ COEFFICIENT RANGES)

如本例中,目标函数中 DESKS 变量当前的系数(CURRENT COEF)=60,允许增加(ALLOWABLE INCREASE)=4、允许减少(ALLOWABLE DECREASE)=20,说明当这个系数在 $[60-4, 60+20]=[56, 80]$ 范围变化时,最优基保持不变。对 TABLES、CHAIRS 变量,可以类似解释。原因此时约束没有变化(只是目标函数中某个系数发生变化,所以最优基保持不变的意思也就是最优解不变(当然,由于目标函数中系数发生了变化,所以最优值会变化)。

(2) 约束右端项变化的范围(RIGHT HAND SIDE RANGES)

如本例中,第 2 行约束中当前右端项 CURRENT RES)=48,允许增加(ALLOWABLE INCREASE)=INFINITY(无穷),允许减少(ALLOWABLE DECREASE)=24,说明当它在 $[48-24, 48+\infty]=[24, \infty]$ 范围变化时,最优基保持不变。第 3、3、5 行可以类似解释。

不过由于此时约束发生变化，最优基即使不变，最优解，最优值也会发生变化。如何变化呢？我们将在本节后面结合 1.2.1 节例 1.1 给出的实际问题来进行说明。

最后，如果你对单纯形法比较熟悉，你可以直接看最优解的单纯形表，这值要选择菜单命令 Reports| Tableau (Alt+7) 执行即可，输出结果见图 2-14。在图 2-14 中，基变量为 BV={SLK2, CHAIRS, DESKS, SLKS5}, ART 是人工变量 (artificial variable), j 即相应的目标值 z；这样，你就可以知道 $z=5 \text{ TABLES}+10\text{SLK3}+10\text{SLK4}=280$ 。

敏感性分析结果表示的是最优基保持不变的系数范围。由此，也可以进一步确定当目标函数的系数和约束右端项发生小的变化时，最优解，最优值如何变化。下面我们通过求解 1.2.1 节例 1.1 的实际问题来进行说明。

| ROW (BASIS) | DESKS | TABLES | CHAIRS | SLK 2 | SLK 3 | SLK 4 |
|-------------|-------|--------|--------|-------|--------|--------|
| 1 ART | 0.000 | 5.000 | 0.000 | 0.000 | 10.000 | 10.000 |
| 2 SLK 2 | 0.000 | -2.000 | 0.000 | 1.000 | 2.000 | -8.000 |
| 3 CHAIRS | 0.000 | -2.000 | 1.000 | 0.000 | 2.000 | -4.000 |
| 4 DESKS | 1.000 | 1.250 | 0.000 | 0.000 | -0.500 | 1.500 |
| 5 SLK 5 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| ROW | SLK 5 | z |
|-----|-------|---------|
| 1 | 0.000 | 280.000 |
| 2 | 0.000 | 24.000a |
| 3 | 0.000 | 8.000 |
| 4 | 0.000 | 2.000 |
| 5 | 1.000 | 5.000 |

图 2-14 LINDO 输出的单纯形表

例 2.5 继续讨论例 1.1，其模型为

$$\max \quad z=72x_1+64x_2; \quad (11)$$

$$s.t \quad x_1+x_2 \leq 50, \quad (12)$$

$$12x_1+8x_2 \leq 480, \quad (13)$$

$$3x_1 \leq 100, \quad (14)$$

$$x_1, x_2 \geq 0. \quad (15)$$

首先在 LINDO 模型窗口输入模型，见图 2-15 求解这个模型并做灵敏度分析，查看报告窗口 (Reports Window) 看到结果如图 2-16。结果告诉我们：这个线性规划的最优解为 $x_1=20, x_2=30$ ，最优值为 $z=3360$ ，即用 20 桶牛奶生产 A_2 ，可获最大利润 3360 元。这与我们在第 1 章中使用图解法得到的结果是一致的，输出中除了告诉我们问题最优解和最优值以外，还有许多对分析结果有用的信息，下面结合题目中提出的 3 个附加问题给予说明。

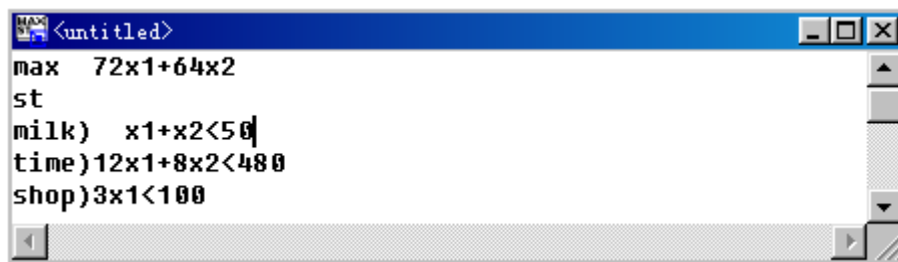


图 2-15 奶制品生产计划问题的模型

3 个约束条件得右端不妨看作 3 种“资源”：原料、劳动时间，车间甲得加工能力。输出中 SLACK OR SURPLUS（松弛或剩余）给出这 3 种资源在最优解下是否有剩余：原料、劳动时间的剩余均为零（即约束为紧约束），车间甲尚余 40kg 加工能力（不是紧约束）。

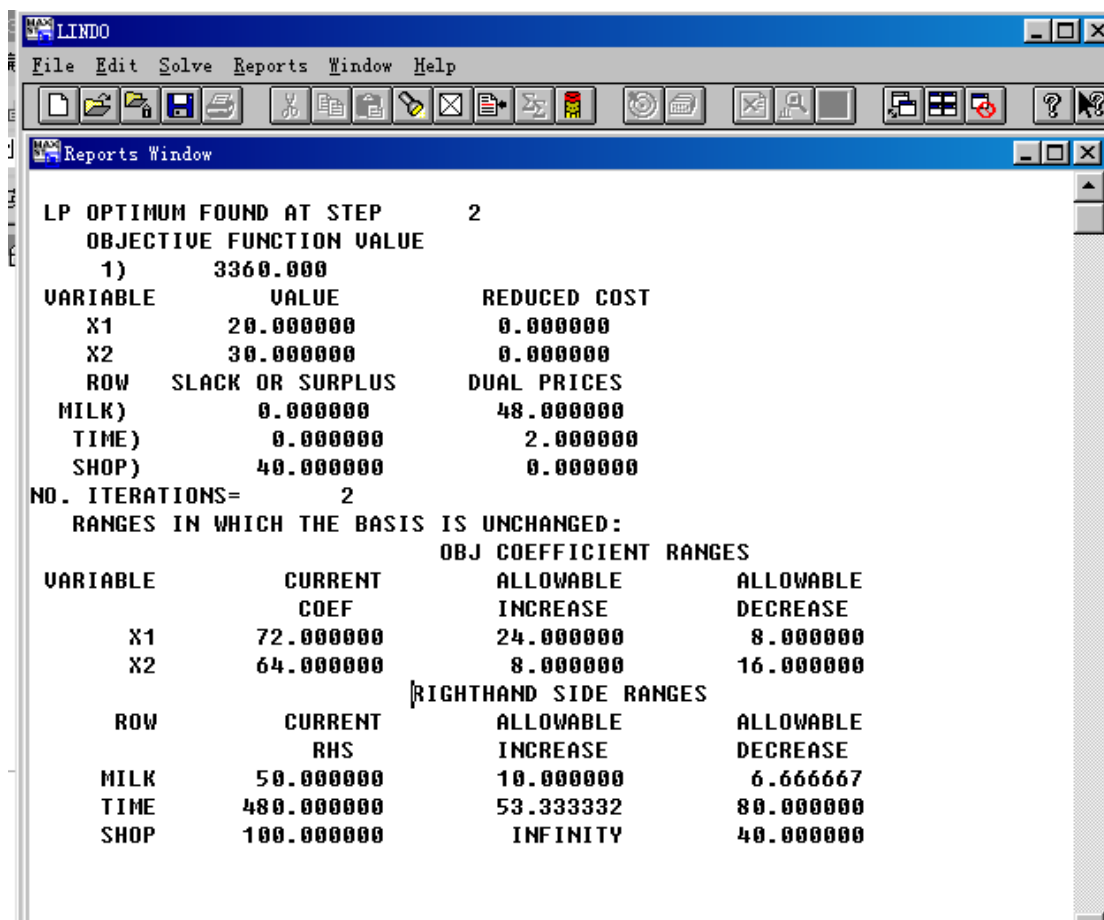


图 2-16 奶制品生产计划问题的计算结果

目标函数可以看作“效益”，成为紧约束的“资源”一旦增加，“效益”必然跟着增长。输出中 DUAL PRICES 给出 3 种资源在最优解下“资源”增加 1 个单位时“效益”的增量：原料增加 1 个单位（1 桶牛奶）时利润增长 48（元），劳动时间增长一个单位（1h）时利润增长 2（元），而增加非紧约束车间甲的能力显然不会使利润增长。这里，“效益”的增量可以看作“资源”的潜在价值，经济学上称为影子价格（shadow price），即 1 桶牛奶的影子价格为 48（元），1h 劳动的影子价格为 2 元，车间甲生产能力的为零。读者可以用直接求解的办法验证上面的结论，即将输入文件中原料约束（milk）右端的 50 改为 51，看看得到的最优值（利润）是否好增长 48（元）。用影子价格的该了概念很容易回答附加问题（1）：用 35 元可以买到 1 桶牛奶，低于 1 桶牛奶的影子价格 48，当然一个作这项投资。回答附加问题

(2): 聘用临时工人以增加劳动时间, 付给的工资低于劳动时间的影子价格才可以增加利润, 剩余工资最多是 2 元/h。

目标函数的系数发生变化时 (假定约束条件不变), 最优解, 最优值会改变吗? 这个问题不能简单地回答。上面的输出结果给出了最优基不变条件下目标函数的允许变化范围: x_1 的系数范围为 $[72-8, 72+24]=[64, 96]$; x_2 的系数范围为 $[64-16, 64+8]=[48, 72]$ 。注意: x_1 系数的允许范围需要 x_2 的系数 64 不变, 反之亦然。由于目标函数的系数变化并不影响约束条件, 因此此时最优基不变可以保证最优解也不变, 但最优值变化用这个结果很容易回答附加问题 (3): 若每千克 A_1 的获利增加到 30 元, 则 x_1 系数变为 $30 \times 3 = 90$, 在允许范围内, 所以不应该改变生产计划, 但最优值为 $90 \times 20 + 64 \times 30 = 3720$ 。

下面对“资源”的影子价格作进一步的分析, 影子价格的作用 (即最优解下“资源”增加 1 个单位时“效益”的增量) 是有限制的。每增加 1 桶牛奶利润增长 48 元 (影子价格), 但是, 从上面输出中可以看出, 约束的右端项 (CURRENT RHS) 的“允许增加” (ALLOWABLE INCREASE) 和“允许减少” (ALLOWABLE DECREASE) 给出了影子价格有意义条件下约束右端的限制范围 (因为此时最优基不变, 所以影子价格才有意义; 如果最优基已经变了, 那么结果中给出的影子价格也就不正确了; 理解这一点可能需要多了解一些线性规划的有关知识)。具体对本例来说: (milk) 原料最多增加 10 桶牛奶, (time) 劳动时间最多增加 53h。现在可以回答附加问题 (1) 的第 2 问: 虽然应该批准用 35 元买 1 桶牛奶的投资, 但每天最多购买 10 桶牛奶。顺便地说, 可以用低于 2 元/h 的工资聘用临时工人以增加劳动时间, 但最多增加 $53 \cdot 3333h$ 。

需要注意的是: 灵敏性分析给出的只是最优基保持不变的充分条件, 而不一定是必要条件。比如对于上面的问题, “原料最多增加 10 桶牛奶” 的含义只能是“原料增加 10 桶牛奶” 时最优基保持不变, 所以影子价格有意义, 即利润的增加大于牛奶的投资。反过来, 原料增加超过 10 桶牛奶, 最优基是否一定改变? 影子价格是否一定没有意义? 一般来说, 这是不能从灵敏性分析报告中直接得到的。此时, 应该重新用新数据求解规划模型, 才能作出判断。所以严格来说, 我们上面回答“原料最多增加 10 桶牛奶” 并不是完全科学的。

2.3 整数线性规划的求解

LINDO 可用于求解线性纯整数规划或混合整数规划 (IP), 模型的输入与 LP 问题类似, 但在标志后需定义整型变量。0/1 型的变量可由 INTEGER (可简写为 INT) 命令来标识, 有以下两种可能的用法:

INT vname

INT n

前者只将决策变量 vname 标识为 0/1 型, 后者将当前模型中 n 个变量标识为 0/1 型 (模型中变量顺序由模型中输入时出现的先后顺序决定, 该顺序可由输出结果中的变量顺序查证是否一致)。

一般的整数变量可用命令 GIN (general integer 的缩写), 其使用方式与 INT 命令相似。再次提醒读者, 对整数变量的说明只能放在模型的“END”语句之后。

下面对 1.2.4 节提出的两个问题, 具体进行演示。

例 2.6 现在我们来解 1.2.4 节例 1.4 的员工聘用问题。首先在 LINDO 模型窗口输入

模型，如图 2-17，其中“GIN 7”表示 7 个变量都是一般整数变量（仍然默认为取直是非负的）。求解后报告窗口见图 2-18。我们特意把状态窗口叠加在这个窗口上，可以看到此时状态窗口中与整数相关的三个域就有了相关结果（其意义也可回顾一下图 2-4 和表 2-1）；

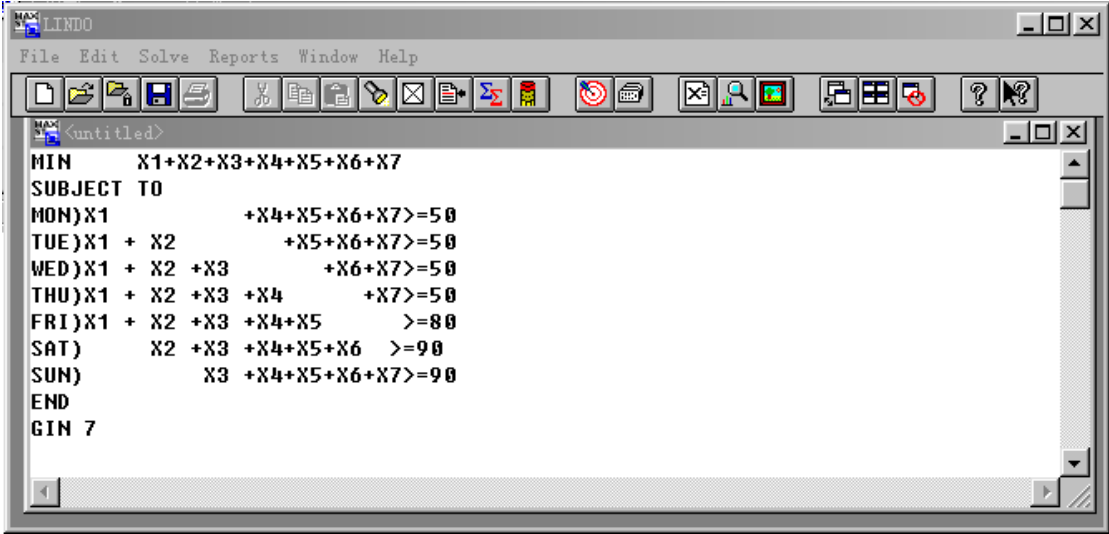


图 2-17 例 2.6 的输入模型

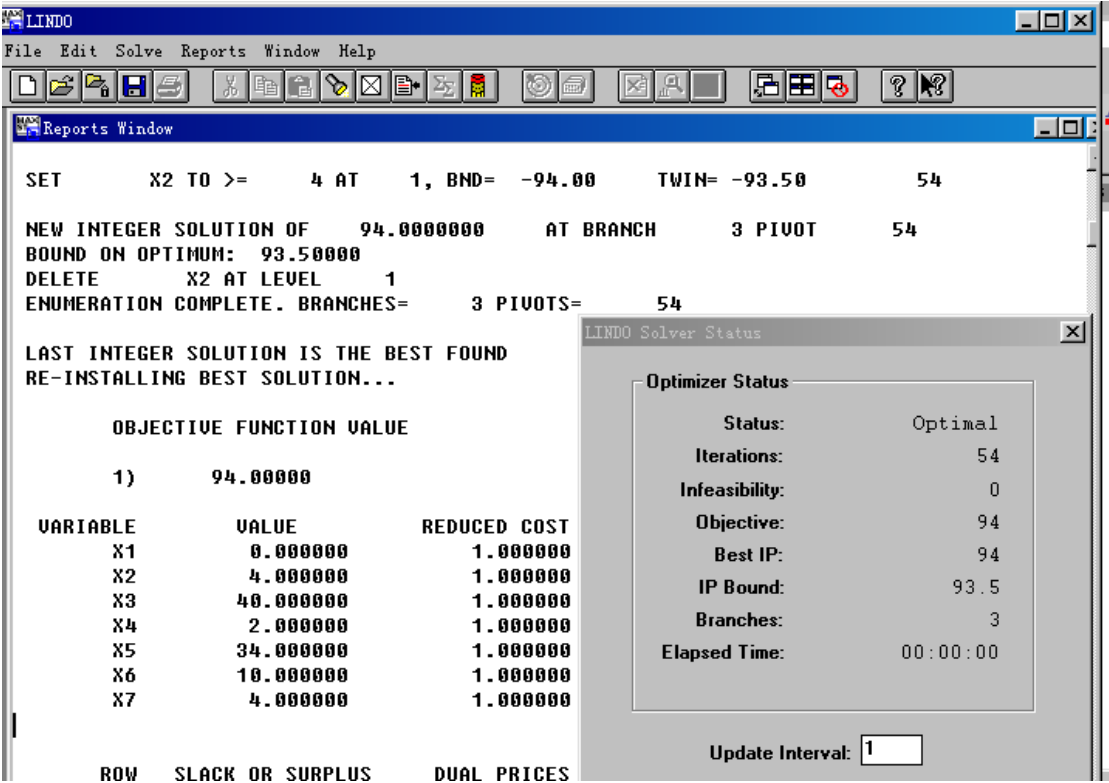


图 2-18 例 2.6 的输出结果

“Best IP: 94”表示当前得到的最好的整数解目标函数值为 94（人）。
 “IP Bound:93.5”表示该整数规划目标值的下界为 93.5（人）。
 “Branches:1”表示分支数为 1（即在第 1 个分支中就找到了最优解）。我们前面说过，LINDO 求解 IP 用的是分支定界法。

显然，上面第二条“整数规划目标值的下界为 93.5（人）”表明至少要聘用 93.5 名员工，由于员工人数只能是整数，所以至少要聘用 94（人），而第一条说明目前得到的解就是 94

人所以已经是最优的了。

报告窗口中前两行告诉我们，在 8 次迭代后找到对应的线性规划（LP）问题的最优解，最优解，最优值等于 93.3333359。LIND 求解 IP 用的是分支定界法，紧接着几行显示的是分支定界的信息，在第一个分支中设定 $x_2 \geq 4$ ，并在该分支中找到了整数解，而且就是全局整数最优解，所以算法停止。旋转迭代（PIVOTS）共 18 次。

后面显示的是最后的最优解 $x=(0,4,40,2,34,10,4)$ 。松弛变量和剩余变量（SLACK ORPLUS）仍然可以表示约束的松紧程度，但目前 IP 尚无相应完善的敏感性分析理论。因此

REDUCED COST 和 DUAL PRICES 的结果在整数规划中意义不大。

例 2.7 下面来解 1.2.4 节 例 1.5 的游泳队员的选拔问题（0-1 规划）。输入见图 2-19，其中“INT20”表示 20 个变量都是 0-1 整数变量；结果报告见图 2-20（只列出变量的取直部分）。求解得到结果为： $x_{14}=x_{21}=x_{32}=x_{43}=1$ ，其他变量为 0，成绩为 253。2s=4'13"2。即应当选派甲乙丙丁 4 人组成接力对，分别参加自由泳、蝶泳、仰泳、蛙泳的比赛。

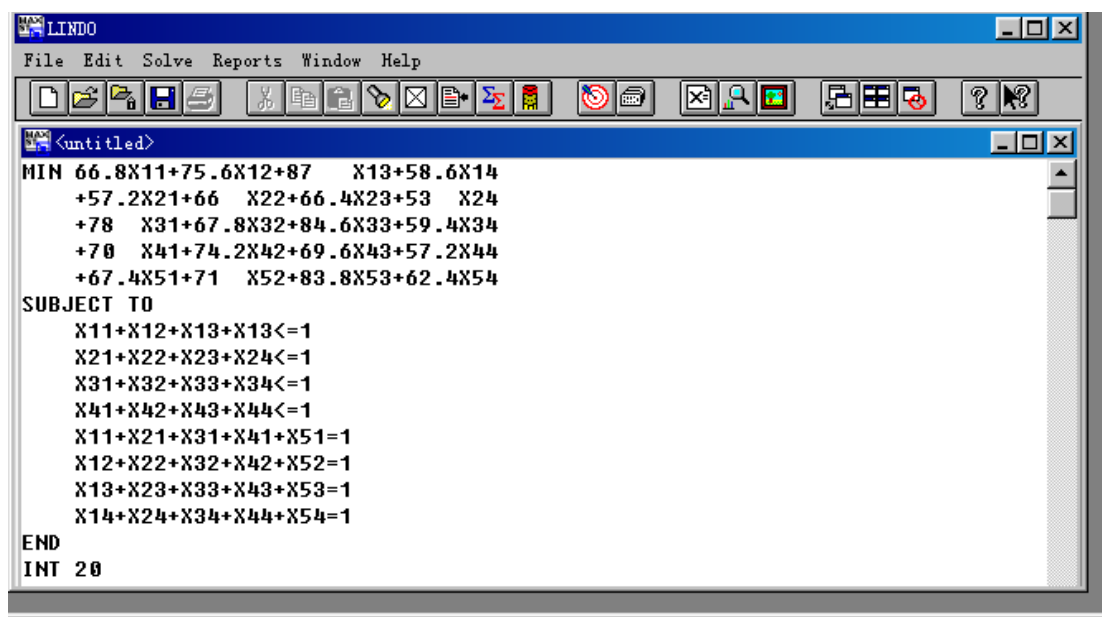


图 2-19 例 2.7 的输入模型

由于这个问题中有 20 个 0-1 变量，而最优解中肯定只有其中的 4 个变量取非零值“1”，所以要在一大堆变量中去找少量的几个变量的几个非零值的变量，这上不大方便的，有没有办法只把非零值的变量显示出来呢？这是可以做到的：选择菜单命令“Reports|solution...(Alt+0)”（这个命令的功能是只要把最优解显示出来），这时会弹出选择对话框（图 2-21）默认的选项是“Noozeros Only(只显示非零值)”。按下图 2-21 对话框中的“OK”按钮，则报告窗口中的显示如图 2-2 所示，可以看到这时显示了 4 个取非零值“1”的变量，这样阅读起来就很方便了。请读者注意，这个功能并不仅仅在整数规划可以使用，在其他模型中也可以使用的，不防试试就知道了。

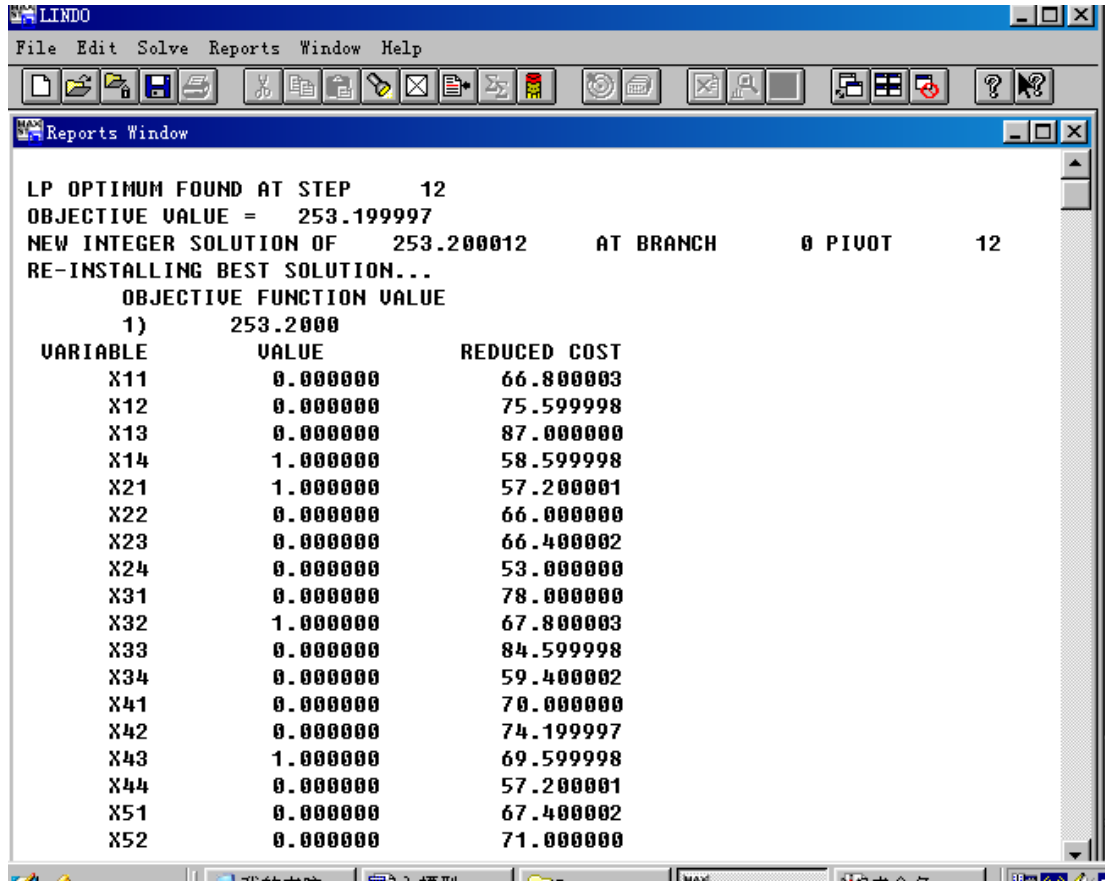


图 2-21 例 2.7 的输出结果

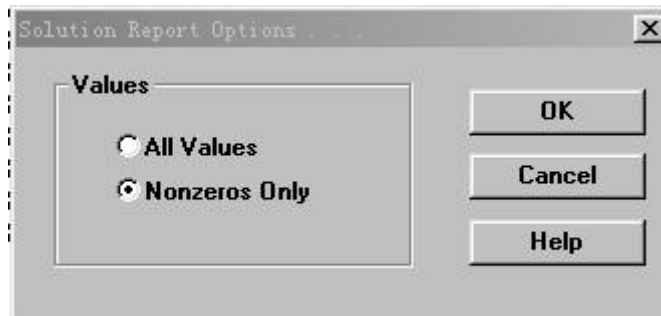


图 2-21 显示解答报告的选择对话框

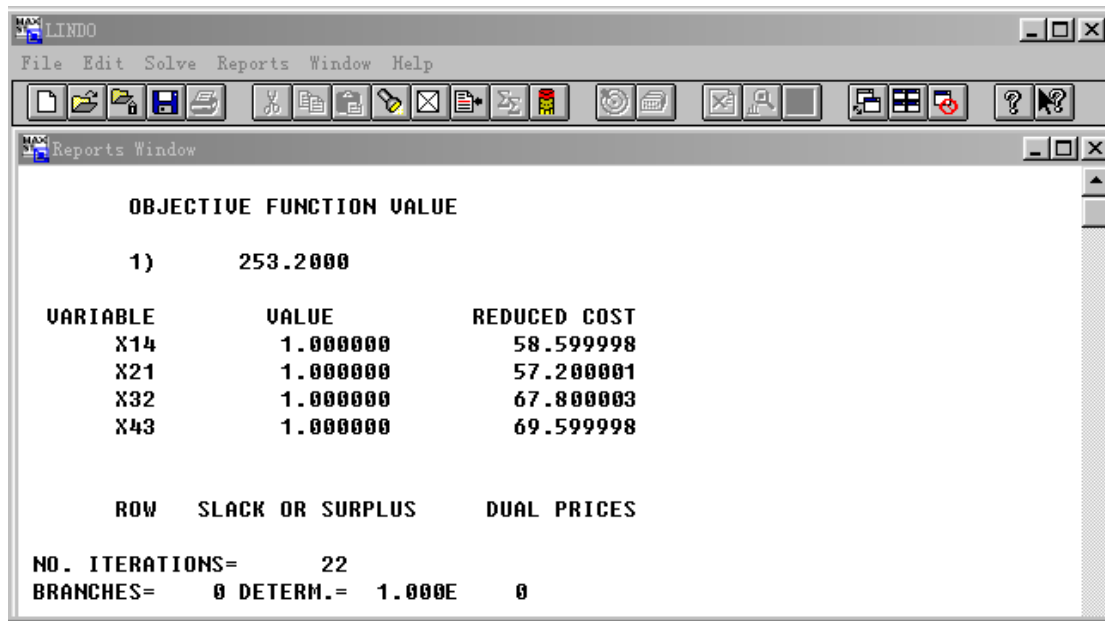


图 2-22 例 2.7 的简洁的输出结果（只输出非零分量的值）

讨论 若考虑到丁、戊最近的状态， c_{43} 由原来的 69.6s 变为 75.2s， c_{54} 由原来的 62.4s 变为 75.5s，讨论对结果的影响。这类似于线性规划中的敏感性分析，但是可惜的是，对于整数规划模型，一般没有与线性规划相类似的理论，此时 LINDO 中所输出的敏感性分析结果通常是没有意义的，因此不能利用这个 输出敏感性分析结果。于是我们只好用 c_{43} ， c_{54} 的新数据重新输入模型，用求解得到 $x_{21}=x_{32}=c_{43}=x_{51}=1$ ，其他的变量为 0，成绩为 257.7s=4' 17" 7。即应当选派乙丙丁戊 4 人组成接力队，分别成绩碟泳、仰泳、蛙泳、自由泳的比赛。请读者最近试试。

下面我们来看一个混合整数规划（既有整数变量、又有实数决策变量）的例子。

例 2·8 一汽车生产小、中、大类型的汽车，已知各类型每辆车对钢材、劳动时间的需求，利润以及每月工厂钢材、劳动时间的现有量如表 2-3 所示。由于各种条件限制，如果生产某一类型汽车，则至少要生产 80 辆。试制定月生产计划，使工厂的利润最大。

表 2-3 汽车生产数据

| | 小型 | 中型 | 大型 | 现有量 |
|--------|-----|-----|-----|-------|
| 钢材/t | 1.5 | 3 | 5 | 600 |
| 劳动时间/h | 280 | 250 | 400 | 60000 |
| 利润/万元 | 2 | 3 | 4 | |

模型建立与求解

设每月生产小、中、大型汽车的数量分为 x_1 ， x_2 ， x_3 （由于生产是一个月一个月连续进行的，所以这里可以合理地认为这个产量不一定非取整数不可，而是可以取实数），工厂的月利润为 z ，在题目所给参数均不随生产数量变化的假设下，立即可得线性规划模型：

$$\max \quad z=2x_1+3x_2+4x_3; \quad (16)$$

$$\text{s. t} \quad 1.5x_1 + 3x_2 + 5x_3 \leq 600, \quad (17)$$

$$280x_1 + 250x_2 + 400x_3 \leq 60000, \quad (18)$$

$$x_1, x_2, x_3 \geq 0. \quad (19)$$

但是，如果生产某一类型汽车，则至少要生产 80 辆，这个约束怎么表达？可以引入 0-1 变量，化为整数规划。设 y_1 只取 0, 1 两个值，则 “ $x_1 = 0$ 或 ≥ 80 ” 等价于

$$x_1 \leq My_1, \quad x_1 \geq 80y_1, \quad y_1 \in \{0, 1\}, \quad (20)$$

其中 M 为相当大的正数，本例可取 1000 (x_1 不可能超过 1000)。类似地有

$$x_2 \leq My_2, \quad x_2 \geq 80y_2, \quad y_2 \in \{0, 1\}, \quad (21)$$

$$x_3 \leq My_3, \quad x_3 \geq 80y_3, \quad y_3 \in \{0, 1\}. \quad (22)$$

于是这个模型构成一个混合整数规划模型（既有一般的整数变量 x ，又有 0-1 变量 y ），用 LINDO 直接求解时，输入的最后（END 语句后）只需要加上 0-1 变量 y 的限定语句。

模型的输入见图 2-23。求解得到输出如下（只列出需要的部分结果）：

OBJECTIVE FUNCTION VALUE

1) 611.2000

| Variable | Value | Reduced Cost |
|----------|----------|--------------|
| Y1 | 1.000000 | 108.8000 |
| Y2 | 1.000000 | 0.000000 |
| Y3 | 0.000000 | 0.000000 |
| X1 | 80.00000 | 0.000000 |
| X2 | 150.4000 | 0.000000 |
| X3 | 0.000000 | 0.800000 |

```

LINDO Model - LINGO1
max 2x1+3x2+4x3
st
1.5x1+3x2+5x3<600
280x1+250x2+400x3<60000
x1-1000y1<0
x2-1000y2<0
x3-1000y3<0
x1-80y1>0
x2-80y2>0
x3-80y3>0
end
int y1
int y2
int y3
For Help, press F1  NUM MOD Ln 16, Col 1 8:08 pm

```

图 2-23 例 2.8 的输入模型

也就是说，只生产小型和中小型汽车，产量分别是 80 辆和 150 辆（近似值），读者不妨试试把产量 x 也限定只取整数，结果会如何呢？

备注 尽管 LINDO 整数规划问题很有威力，但要想有效地使用，有时还是需要一定的技巧的。这是因为，人们很容易将一个本质是很简单的问题列成一个不太好的输入模型，从而有可能会产生一个冗长的分支定界计算。遗憾的是，我们往往难以预先估计什么样的模型才能避免冗长的分支定界计算，也难以判断什么样的模型是“不太好”的输入模型。当然这时 LINDO 会主动砍去一些计算过程，以缩短计算时间，而且越是高版本的 LINDO 软件，这种自动处理的“智能”越强。我们的建议是：如果分支定界计算时间很长仍得不到最优解，你可以试试对输入模型进行一些等价交换：如交换变量的次序，交换约束的顺序等，有时也许会对减少求解所需的时间有所帮助。

* 2.4 二次规划的求解

LINDO 可用于求解二次规划 (QP) 问题，但输入方式比较复杂，因为在 LINDO 中不许出现非线性表达式，我们需要为每一个实际约束增加一个对偶变量(或 lagrange 乘子)，通过在实际约束前增加有关变量的一阶最优条件，从而转化二次型为线性互补型(对线性互补型有兴趣的读者，可参阅其他一些书籍)；并要使用 QCP 命令指明实际约束开始的行号，然后才能求解。下面仅通过两个例子进行说明。

例2.9 求解如下二次规划问题

$$\begin{aligned}
 \min z &= 3x^2 + y^2 - xy - 0.4y; \\
 &1.2x + 0.9y > 1.1, \\
 \text{s.t. } &x + y = 1, \\
 &y < 0.7
 \end{aligned}$$

我们用 RT, ONE 和 UL 作为对偶变量，问题输入格式参见图 2-24.

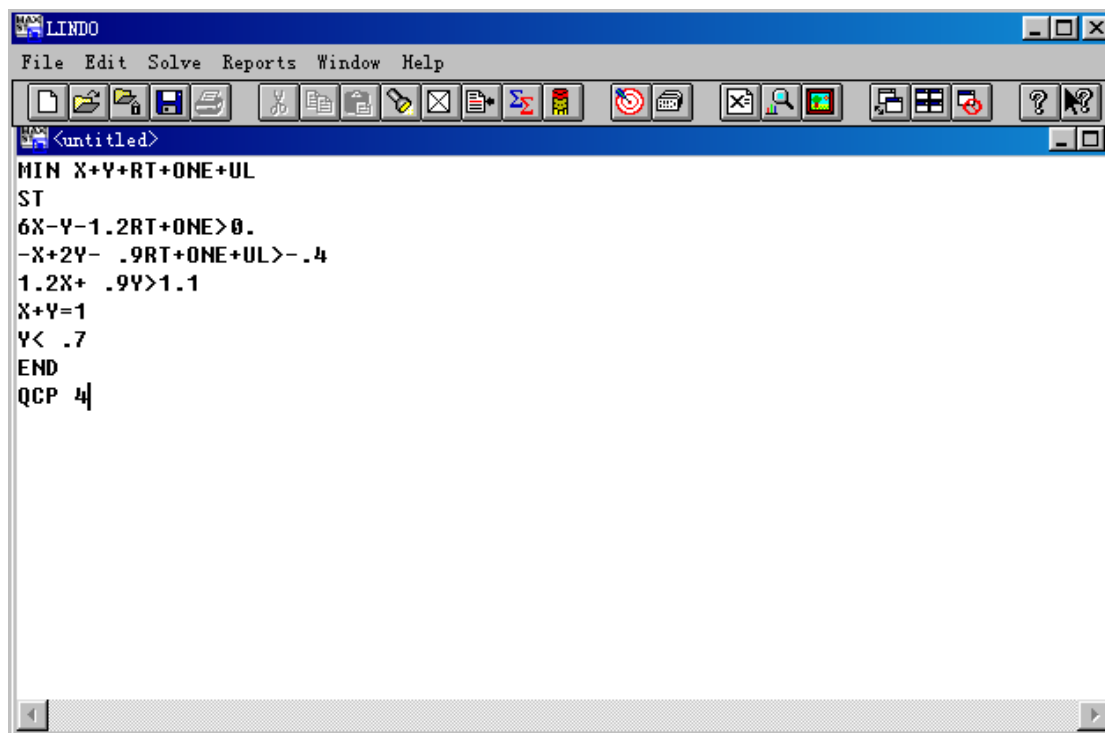


图 2-24 二次规划的输入

输入中的第一行(目标函数)只用于给出模型中相应变量的出现顺序:X, Y, RT, ONE, UL, 用加号连接。

输入中的第二行、三行约束是在实际约束前增加的有关变量的一阶最优条件, 即 lagrange 函数

$$3x^2 + y^2 - xy + 0.4y - RT(1.2x + 0.9y - 1.1) + ONE(x + y - 1) + UL(y - 0.7) \quad (2.7)$$

分别对 x, y 求偏导数, 令其大于 0 可得第二、三行约束(最优条件)。

“END”后面的语句“QCP 4”表示原来的二次规划问题真正的约束是从输入的第 4 行开始的。

求解得到输出如下(只列出需要的部分结果):

QP OPTIMUM FOUND AT STEP 7

OBJECTIVE FUNCTION VALUE

1) 1.355556

| VARIABLE | VALUE | REDUCED COST |
|----------|------------------|--------------|
| X | 0.666667 | 0.000000 |
| Y | 0.333333 | 0.000000 |
| RT | 10.888889 | 0.000000 |
| ONE | 9.400000 | 0.000000 |
| UL | 0.000000 | 0.366667 |
| ROW | SLACK OR SURPLUS | DUAL PRICES |
| 2) | 0.000000 | -0.666667 |
| 3) | 0.000000 | -0.333333 |
| 4) | 0.000000 | -10.888889 |

| | | |
|-----------------|----------|----------|
| 5) | 0.000000 | 9.400000 |
| 6) | 0.366667 | 0.000000 |
| NO. ITERATIONS= | | 7 |

这样,经过 7 次迭代,就得到了最优解 $x=0.666667$, $y=0.333333$, 最优值为 1.3555560
同整数规划时的情况类似,二次规划也没有敏感性分析结果,因此 LINDO 对应的敏感性分析输出结果也是没有意义的,不能被利用。

例2.10 对于 1.2.2 节例 1.2 给出的如下二次规划问题(注意:我们把目标函数取了负号,因此把原来的最大化问题变成了最小化问题):

$$\min z = -98x_1 - 277x_2 + x_1^2 + 0.3x_1x_2 + 2x_2^2$$

$$\begin{aligned} & x_1 + x_2 \leq 100, \\ \text{s. t. } & x_1 \leq 2x_2, \\ & x_1, x_2 \geq 0. \end{aligned}$$

我们需要用对偶变量(Lagrange 乘子)写出其 Lagrange 函数. 设两个“ \leq ”约束的 Lagrange 乘子分别是 LAG1, LAG2, 则其 Lagrange 函数为

$$\begin{aligned} \min z = & -98x_1 - 277x_2 + x_1^2 + 0.3x_1x_2 + 2x_2^2 \\ & + LAG1(x_1 + x_2 - 100) \\ & + LAG2(x_1 - 2x_2). \end{aligned} \quad (32)$$

分别对原问题的决策变量 x_1, x_2 求偏导数, 令其大于等于 0 (这实际上是一阶最优条件), 可得两个新约束如下:

$$\begin{aligned} -98 + 2x_1 + 0.3x_2 + LAG1 + LAG2 & \geq 0, \\ -277 + 4x_2 + 0.3x_1 + LAG1 - 2LAG2 & \geq 0 \end{aligned}$$

QP 问题输入 LINDO 软件求解时, 第一行(目标函数)只用于给出模型中相应变量的出现顺序: $x_1, x_2, LAG1, LAG2$, 用加号连接; 在实际约束前增加刚刚得到的有关变量的一阶最优条件。此外, 必须在“end”语句用“QCP n”语句说明这是一个二次规划, 同时指出实际约束是从第 n 行开始的。最后, 这个问题的输入如图 2-25 (注意: 我们这里故意假设产量必须为整数, 所以最后增加了“gin2”语句)。

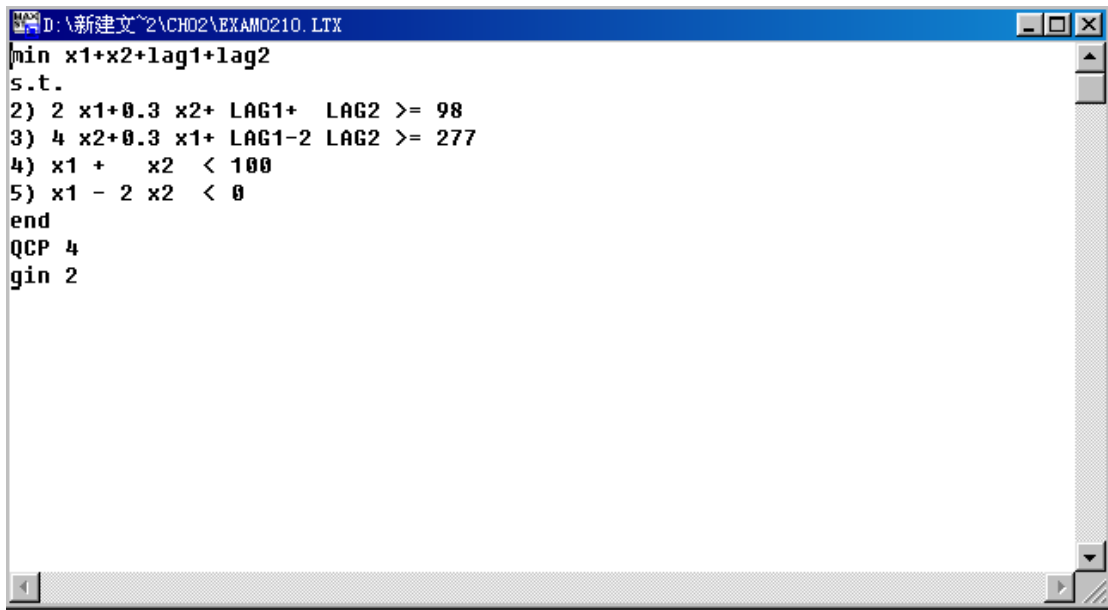


图 2-25 例 2.10 的输入

最后得到的结果见图 2-26，最优整数解 $x = (35, 65)$ ，最大利润等于 11109.17.

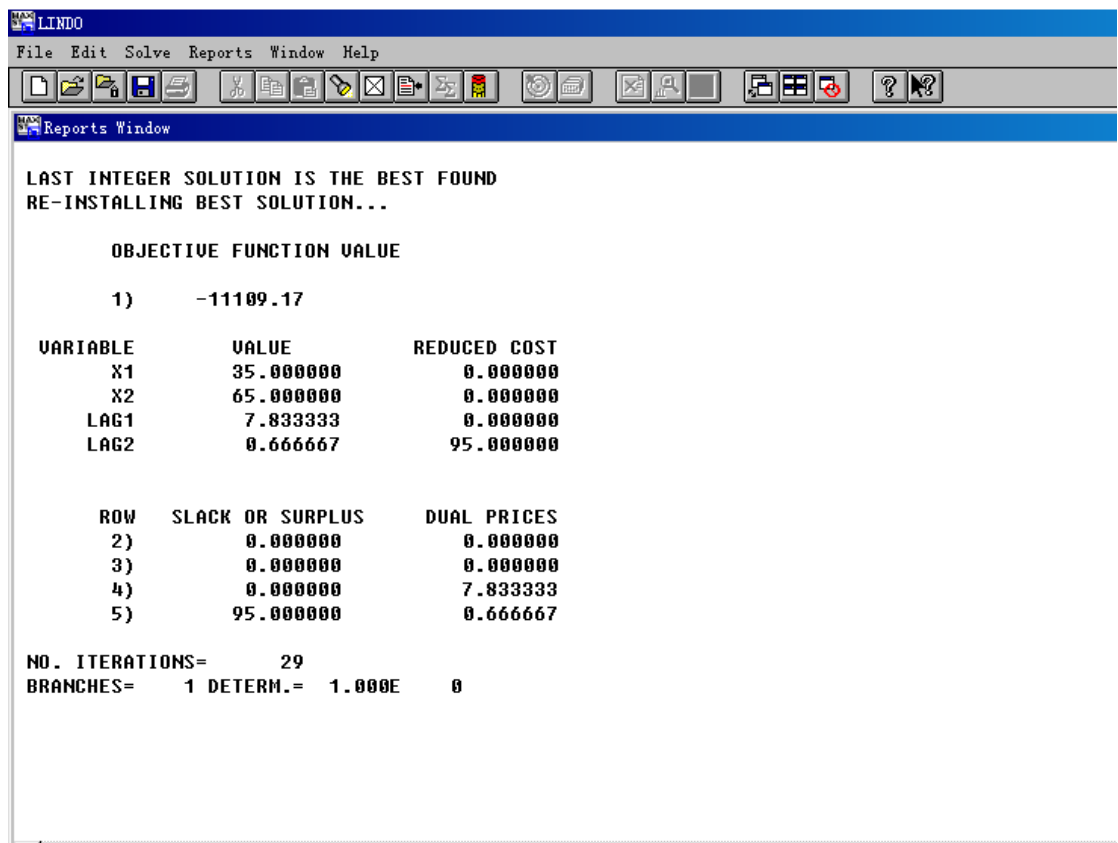


图 2-26 例 2.10 的输出结果

可见，LINDO 用于求解二次规划（QP）问题比较复杂，不容易非常直观地理解。特别要注意：若原约束是“ \geq ”约束，则对应的 Lagrange 乘子应该取 0 或负值（对最小化问题），所以在例 2.9 构造的 Lagrange 函数中，RT 的前面是一个负号（减号），这样 RT 就取非负值了。对最大化问题，这些规律可能又不相同，所以建议您在没有绝对把握时总是将目标一律

化成最小化 (min) 问题, 将约束一律写成 “ \leq ” 约束 (等式约束也可以), 以免出错。

备注 建议最好直接用第 3 章介绍的 LINGO 软件求解 QP 问题, 因为 LINGO 中输入的模型更接近二次规划的数学表达形式, 不容易出错, 而计算效果同样很好。

要想学好和灵活应用 LINGO 软件, 首先要多练习使用 LINDO 来解决问题, 熟能生巧。LINDO 中的显示报告完全是英文的, 大家要熟悉其含义。不要太拘泥于书本或别人教你的方法, 要举一反三, 综合使用, 才能用得巧而精。这就像编程一样, 同样的几条程序命令, 有的人只能生搬硬套, 而有的人却能发挥得淋漓尽致, 这中间的功夫不是光靠一招招向书本学能得来的了。

*2. 5 LINDO 的主要菜单命令

从前面的各个图形窗口中我们已经看到, LINDO 软件的菜单条上有 6 个主菜单:

- File (文件)
- Edit (编辑)
- Solve (求解)
- Reports (报告)
- Window (窗口)
- Help (帮助)

File (文件) 菜单包括了 LINDO 通过文件与外部设备 (如磁盘) 交换信息的命令; Edit (编辑) 菜单包括了在当前窗口下编辑文本的命令; Solve (求解) 菜单包括了求解模型的命令; Reports (报告) 菜单包括了生成解答结果报告的命令; Window (窗口) 菜单包括了窗口切换的命令; Help (帮助) 菜单包括了访问在线帮助文档的命令。

对于几乎所有的菜单命令, LINDO 都提供了快捷键 (快捷键的提示位于每个菜单命令的右侧); 对于常用的菜单命令, LINDO 在工具栏提供了相应的图形按钮 (参见图 2-27)。工具栏是浮动式的, 可以用鼠标拖到屏幕上任何地方。这些用法都是和 WINDOWS 下其它应用程序的标准用法类似的, 所以我们不准备对所有的菜单命令进行完整和详细的介绍, 而是只对前 4 个主菜单中有一定 LINDO 特色的主要命令进行简要介绍。

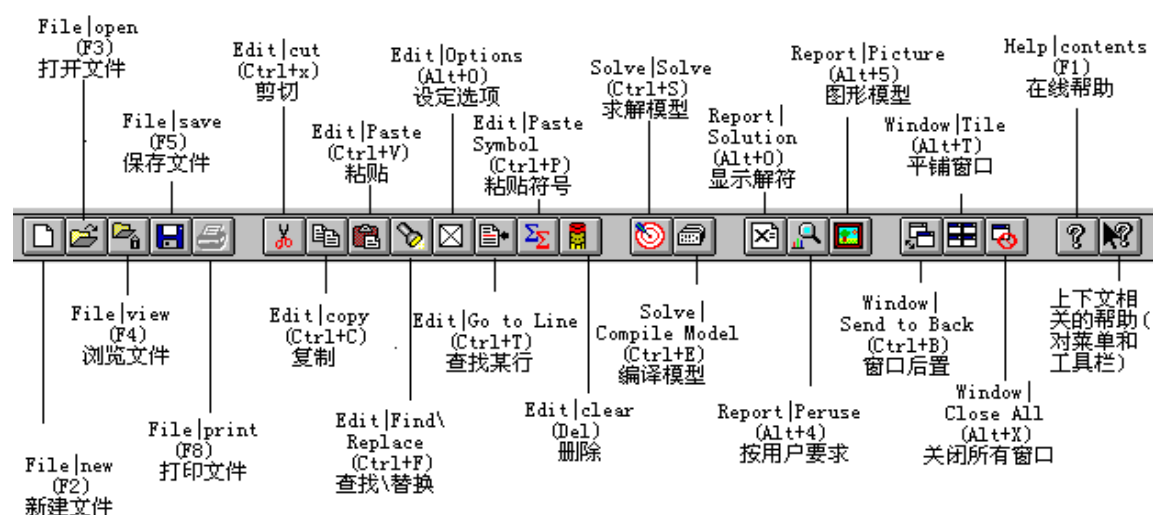


图 2-27 LINDO 工具栏及其对应的菜单命令和快捷键

2. 5. 1 文件主菜单

- File|New、File|Open、File|View 的区别

File|New 用于新建一个模型文件; File|Open 用于打开一个已有文件, 此后可以对这个文件进行编辑、求解、保存等; 而 File|View 只用于打开已有文件供浏览 (也可以求解) 使

用，不能编辑。由于 LINDO 编辑器对文件的大小是有限制的，因此用 File|New 和 File|Open 打开的文件不能太大（通常不一定是由 LINDO 本身的编辑器产生的）是有用的。

- File|Log Output

该命令将打开一个对话框（图 2-28），要求你指定一个文件名（该文件成为“Log(日志文件)”）。此后，LINDO 软件的所有输出都被送到这个日志文件中保存下来，供你以后查看。注意，正常情况下，在菜单驱动模式下，LINDO 的输出应当是被送到报告窗口；在“Command Window(命令窗口)”模式下（参见 2.6 节），LINDO 的输出应当是被送到命令窗口。在图 2-28 的对话框中有两个检验盒：

（1）如果选择“Echo to screen（屏幕显示）”检验盒，屏幕上也会同时显示输出结果，否则屏幕上就不再显示了；

（2）如果选择“Append output（追加输出）”检验盒，则以后所有 LINDO 的输出被追加到这个日志文件的结尾，否则系统将首先清空这个文件，然后开始追加内容。

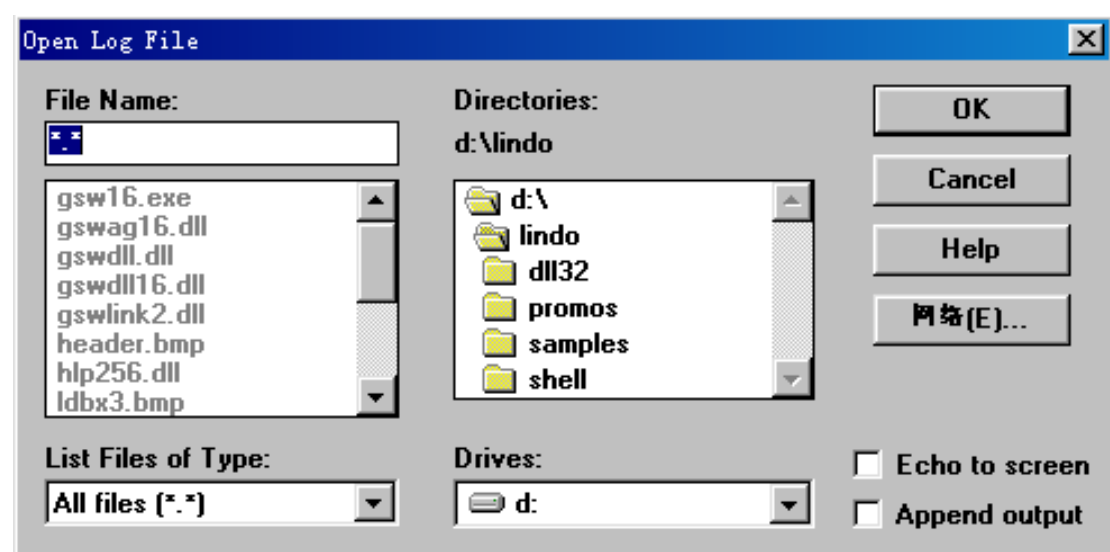


图 2-28 日志文件对话框

- File|Take Commands

File|Take Commands（提取命令）用于打开和执行一个 LINDO 命令脚本文件（命令脚本文件中包含一系列 LINDO 命令组成的命令序列，具体参见 2.6 节对应的行命令“TAKE”以及 2.7 节对应的例子）。

- File|Basis Read 和 File|Basis Save

File|Basis Save（保存基）命令打开一个标准的文件保存对话框，可以将单纯形算法的当前的基（解）以你指定的文件名和文件格式保存下来；将来可以用 File|Basis Read（读取基）命令读出这个基（解），并可以从这个基（解）开始继续运行单纯形算法。

保存时可以有三种文件格式可供选择：*.pun（以 MPS(数学规划系统)的“punch”格式保存）；*.fbs（以 LINDO 格式保存）；*.sdbc（以数据库格式按列(变量)保存）。具体参考 2.6 节中对应的行命令。

- File|Title

显示当前模型的名称（如果该模型被命名过，即模型的程序中出现过 Title 语句）。

- File|Date

显示当前日期和时间。

- File|Elapsed Time

显示本次启动 LINDO 以来已经运行了多长时间。

• File|Licence

输入、验证 LINDO 的许可证密码，功能和界面与图 2-1 相同。

2. 5. 2 编辑主菜单

该菜单下的多数命令基本上是不言自明的，与 Windows 下的其他编辑器类似，这些命令就不具体介绍了。这里只介绍几个 LINDO 软件中的特色命令。

• Edit|Options

该命令打开一个对话框（见图 2-29），用于设置 LINDO 系统运行的内部参数，这对于比较专业的用户是有帮助的。从图中可以看出，可修改的参数分成两大类：左边一类是关于优化程序的（Optimizer 这里是指优化程序，也就是 LINDO 求解器，而不是最优解的意思），右边一类是关于输出格式的（Output）。

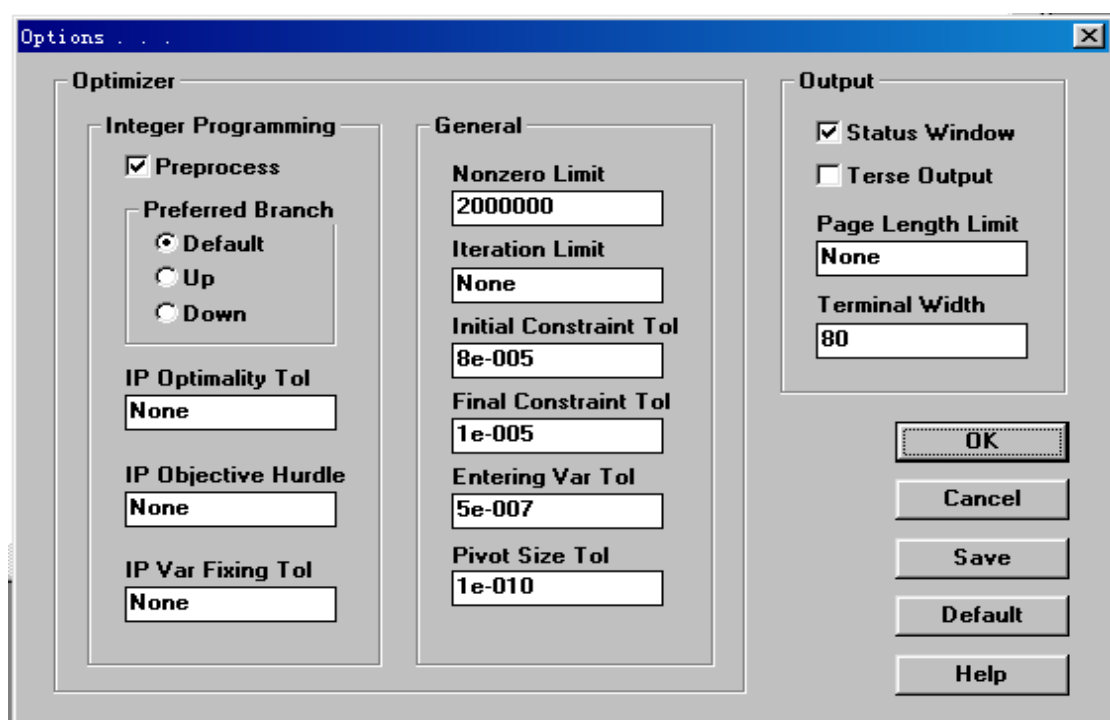


图 2-29 Options 对话框

先看输出格式（Output）中所包括的 4 个选项：

Status Window（状态窗口）选项：用于控制求解模型时是否显示状态窗口（默认设置为显示）。

Terse Output（以简明的形式显示结果）选项：用于控制是否以简明形式报告结果（默认设置为以详细(Verbose)的形式报告结果）。

Page Length Limit（页长限制）：用于控制输出时每页最多显示多少行（可以设置为任意正整数；默认设置是“None”，表示无限制）。

Terminal Width（终端宽度）：每行的最大宽度（每行多少字符），可以设为 40—132 之间的整数（默认设置为 80）。

关于优化程序（Optimizer）的参数又分成两类，左边一类是关于整数规划（Integer Programming，即 IP）的，右边一类是一般参数（General），对于 IP，可以如下设置参数。

Preprocess（预处理）：控制是否进行预处理。预处理中将生成割平面割平面对求解有利，所以系统默认设置是进行预处理但生成割平面也会花费不少时间，所以也可不让 lingo 生成割平面，即不进行预处理。

Preferred Branch(优先的分支方式): 可以选择“Default”(默认方式, 向上或向下都可能)、“Up”(向上取整优先)、“Down”(向下取整优先)。

Ip Optimality Tol(Ip 最优值允许的误差上限): 是一个百分数, 如 0.05 表示 5%, 即当 lingo 得到一个目标值与最优值, 相差不超过 5% 的可行解, 就认为达到了最优, 停止计算。默认值为“None”表示不使用这个判停准则, (即相当于要求误差为 0)

Ip Objective Hurdle(Ip 表示目标函数的篱笆值): 即只在比这个值更优的解中寻找最优解, 这相当于给定了一个最优值的一个界, 因此有利于求解(例如, 当知道当前模型的某个整数可行解时就可以用这个好书的的目标值设置这个值)。默认值为“None”, 表示没有指定这个条件。

Ip Var Fixing Tol(固定一个整数变量取值时允许的上限): 如果在 Ip 模型松弛后的计算中某一个整数变量确实取到了整数, 但对应的判别数 (REDUCED COST) 的值很大, 超过这里指定的上限, 则以后的迭代中就把该整数变量固定下来不再允许变化。这样做的理由是判别数很大的整数很难在以后的迭代中发生变化。默认为“None”, 表示没有指定这个条件。

对于 lingo 的一般参数 (General), 可以如下设置

Nonzero Limit(模型中允许出现的非零系数的个数上限): 这个参数对于不同的 lingo 软件的默认值不同, 试用版中是 2000000。

Iteration Limit(求解时允许的最大迭代步数): 默认值是“None”, 即没有限制; 有时为防止计算时间太长, 拥护可以设置为任意一个正整数。

Initial Constraint Tol(最后阶段求解是约束允许的误差上限): 即只要约束两边相差小于这个数时, 就认为约束成立。计算的初始阶段这个误差可能没有必要设置得过小, 以免找不到可行解, 所以默认值是 0.00008。

Final Constraint Tol(最初阶段时约束允许的误差上限): 含义同上, 即只要约束两边的相差小于这个数时, 就认为约束成立。计算的最后阶段这个误差有必要设置得比较小, 以便提高计算精度, 所以默认值是 0.00001。

Entering Var Tol(进基变量的误差上限): 即只有当变量的判别数大雨这个数时 (按绝对值), 这个变量才可能进基 (相当于认为绝对值小于这个数时, 判别数就是零)。默认值为 0.0000005。

Pivot size Tol(旋转 (迭代) 时采用的误差下限): 即绝对值不能小于该上限 (相当于认为绝对值小于这个数时, 旋转元就是 0)。默认值为 0.000000001。

一旦参数被修改并按下“OK”后, 即对所有以后的运行均有效, 直到退出 lingo 系统或重新设置这个参数为止, 而与具体模型无关。如果将这些参数用“save(保存)”按钮保存下来, 退出 lingo 后下次启动 lingo 时这些参数仍然有效。对话框中右下方的“Default(默认)”按钮用于恢复 lingo 的参数值。“Cancel(取消)”按钮用于废除本次参数修改, 关闭这个选相窗口; “Help(帮助)”按钮用于提供本窗口的在线帮助。

• Edit | Paste Symbol

该命令打开一个对话框, 用于在模型中当前光标处插入符号。例如, 对于前面介绍的 QP 模型 (参见例 2.9), Paste Symbol 打开的对话框如图 2-30, 可以看到, 可选的符号主要是三类:

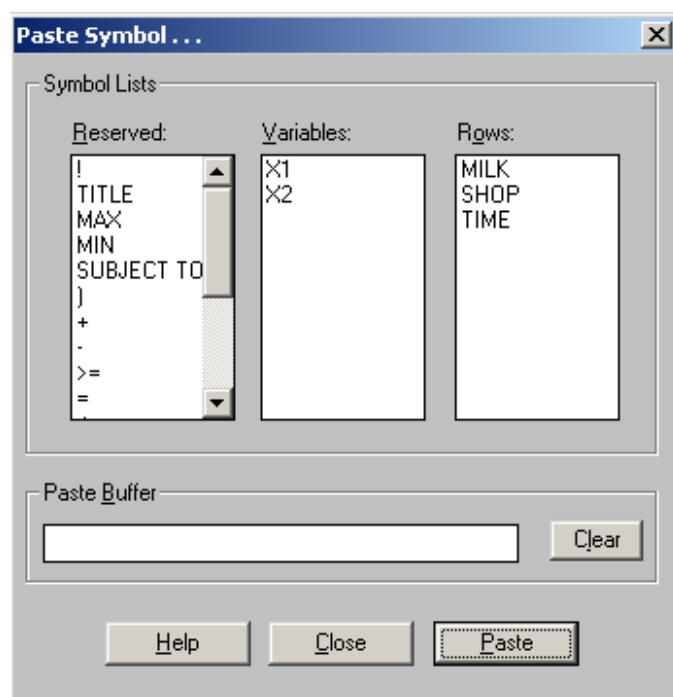


图 2-30 Paste Symbol 对话框

Reserved(保留字): lingo 系统的保留字 (如一些语句关键词和运算符号);

Variables(变量): 当前模型的决策变量;

Rows(行名): 约束的行号或行名。

可以双击其中某个符号, 则该符号显示在图中的缓冲区 (“Paste Buffer”); 也可以直接编辑缓冲区的内容。当单击 “Paste(粘贴)” 按钮时, 缓冲区的内容将被插入当前模型的当前光标后。单击 “Clear (清除)” 按钮将清除缓冲区的内容, 单击 “Close(关闭)” 按钮将关闭对话框。

- Edit | Choose New Font

该命令用于指定显示的字体、字行和文字的大小。对话框如图 2-31 所示。

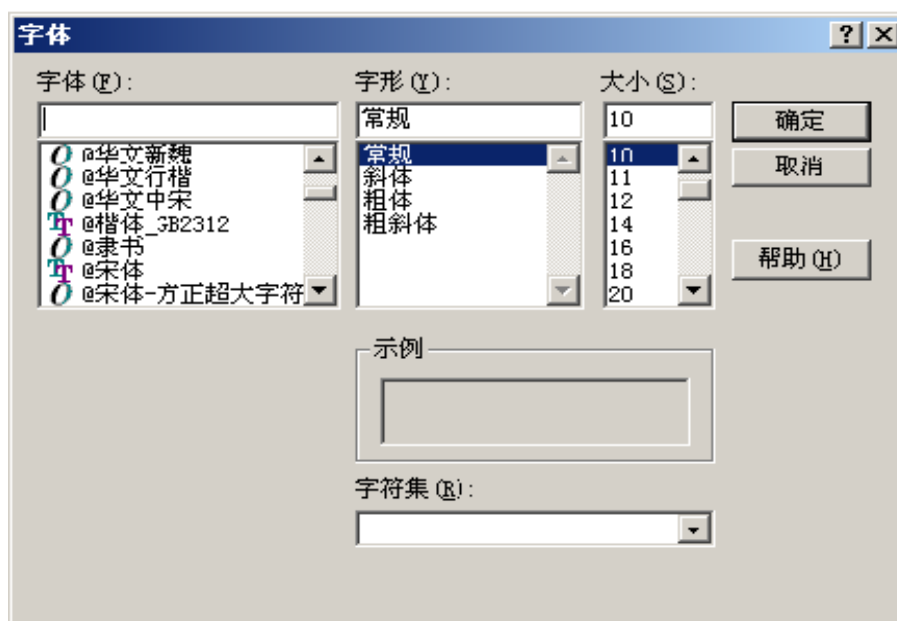


图 2-31 字体选择

2.5.3 求解主菜单

- Solve | Compile Model

Solve | Compile Model(编译模型)命令对当前模型进行编译(使用 Solve | Solve 命令时自然也要先使用该命令)。如果当前模型有输入语法错误,编译时将报告错误。

- Solve | Pivot

Solve | Pivot(旋转)命令从当前解出发进行一次单纯形旋转(即一次迭代)。用这个命令可以跟踪整个单纯形法的运行。

- Solve | Debug

Solve | Debug(调试)命令分析 LP 无解或无界的原因建议如何修改。它仅对 LP 有效,对 IP 或 QP 无意义。对线性规划的高手而言,这个功能可能是有用的。

- Solve | Preemptive Goal

Solve | Preemptive Goal(多目标)命令依次按照多个目标求解模型,参见 2.6 节的命令 GLEX。它仅对 LP 和 IP 有效,对 QP 无意义。

2.5.4 报告主菜单

- Report | Solution

Report | Solution(解答)命令显示当前的解(你必须在此之前求解过当前模型)。对话框参见图 2-21,你可以选择“ALL Values”(把所有变量的值全部显示)、“Nonzeros Only”(只显示非零取值的变量),然后按“OK”即可。

- Report | Range

Report | Range(敏感性分析)命令显示当前解的敏感性分析结果(你必须在此之前求解过当前模型)。敏感性分析可参见 2.2 节。

- Report | Parametrics

Report | Parametrics(参数分析)命令对约束的右端项(RHS)进行参数分析,也就是研究某个约束的右端项发生变化时,最优值如何变化。例如,对于前面介绍过的员工聘用模型,对话框入图 2-32,你可以选择约束的行名(这里选择的是 MON),输入新的右端项值(New RHS Value,这里输入的是 40,原来的值是 50),还可以选择参数分析结果的报告方式(文本(Text),二维图形(2D),三维图形(3D)),单击“OK”按钮得到参数分析结果(如图 2-23 所示),非常方便!从图中和报告中的显示结果都可以看到,这时最优解和最优值没有变化。请你用其它行或其它数试试,看看效果如何。

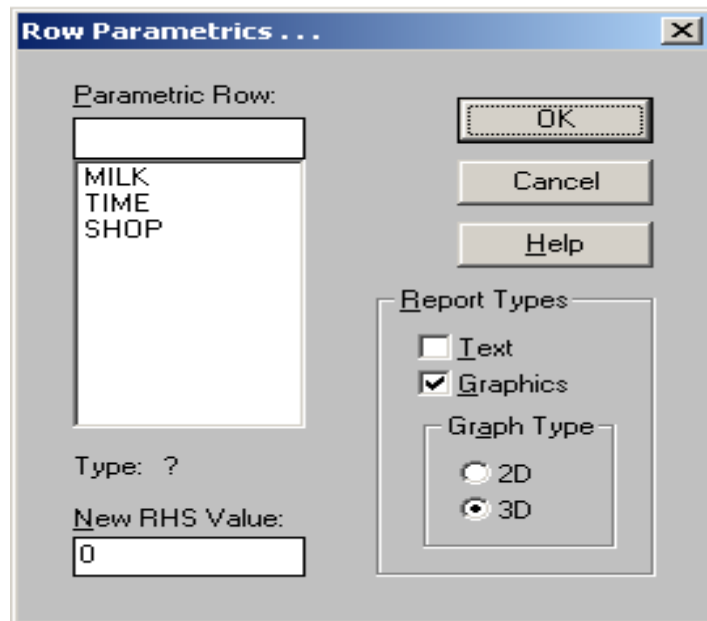


图 2-32 Row Parametrics 对话框

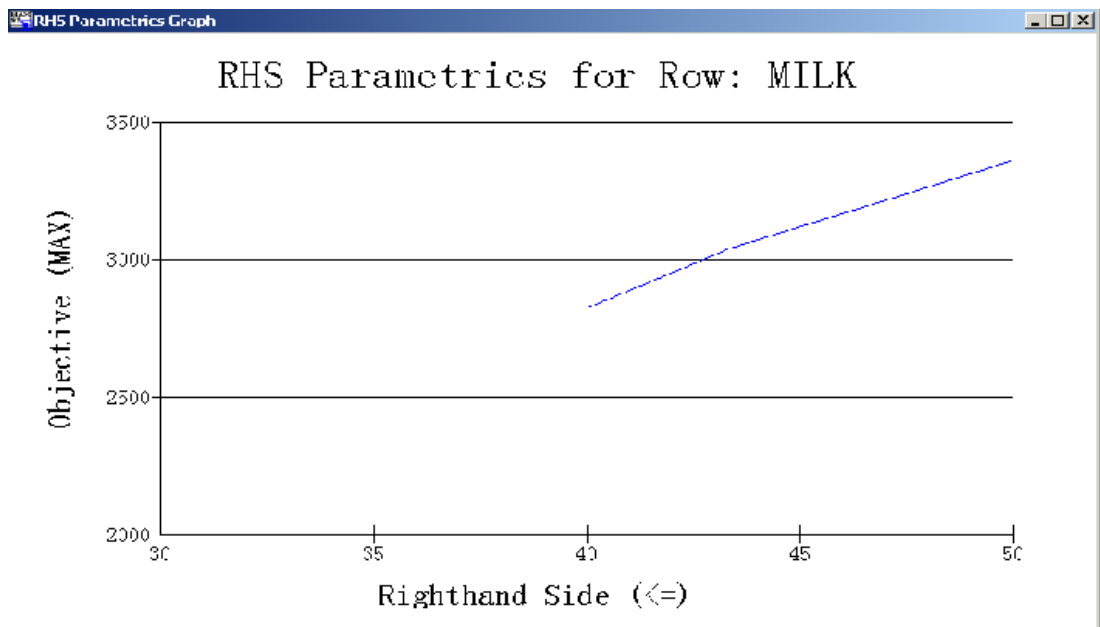


图 2-33 参数分析显示结果

• Report | Statistics

Report | Statistics(统计)命令显示当前模型的统计信息. 例如, 对于前面介绍的 QP 模型 (例 2.9), 该命令将在报告窗口显示如下的信息:

```
ROW= 5 VARS= 4 INTEGER VARS 2 ( 0=0/1) QCP= 4
NONZEROS= 19CONSTRAINT NONZ= 12 ( 6 = + - 1) DESITY=0.760
SMALLEST AND LARGEST ELEMENTS IN ABSOLUTE VALUE= 0.300000 277.000
OBJ=MIN, NO. <, =, >: 2 0 2, GUBS<= 1VUBS>= 0
SINGLE COLS= 0 REDUNDANT COLS= 0
```

第一行的意思: 该模型有 5 行 (当然, 约束只有 4 行), 4 个变量, 其中两个整数变量

(没有 0-1 变量), 从第 4 行开始是二次规划的实际约束。

第二好哪个的意思: 非零系数共有 19 个, 约束中非零系数共有 12 个 (其中 6 个为 1 或-1), 模型密码为 0.760 (密码定义为非零系数/[行数 X (变量数+1)], 分母即模型中所以可能出现的系数个数, 这里系数也包括右端项)。

第三行的意思: 模型中系数的最小值和最大值 (按绝对值看) 分别为 0.3 和 277.

第四行的意思: 模型目标为极小化; 小于等于、等于、大于等于约束分别有 2, 0, 2 个;

广义上界约束 (VUBS) 不少于 0 个. 所谓 GUBS, 是指一组不含有相同变量的约束; 所谓 VUBS, 是指一个蕴涵变量上界的约束, 如从约束 $X_1 + X_2 - X_3 = 0$ 可以看出, 若 $X_3 = 0$, 则 $X_1 = 0, X_2 = 0$ (因为有非负限制), 因此 $X_1 + X_2 - X_3 = 0$ 是一个 VUBS 约束.

第五行的意思: 只含有 1 个变量的约束个数=0 个; 冗余的列数=0 个.

- Report | Peruse

Report | Peruse (用户请求) 命令按照你的要求显示大拿跟前解答的各种信息, 对话框如图 2-34. 用户主要有两类选择:

- Report Parameters (报告参数): 设置用户需要显示的信息项; 选项框 “A” “B” “C” 中项目的含义可参见 2.6 节的命令 CPRI 和 RPRI.

- Report Format (报告参数): 设置用户希望的显示格式.

该命令的具体用法这里不详细介绍了, 大家试试就清楚了.

- Report | Picture 和 Report | Basis Picture

Report | Picture (模型图示) 按照图形或文本方式显示模型中的非零系数 (参见图 2-9), 而 Report | Basis Picture (基图示) 只显示当前基 (Basis) 的非零系数. 参见 2.6 节的命令 PIC 和 BPIC.

- Report | Tableau

.Report | Tableau (单纯形表) 显示当前单纯形表 (参见图 2-12).

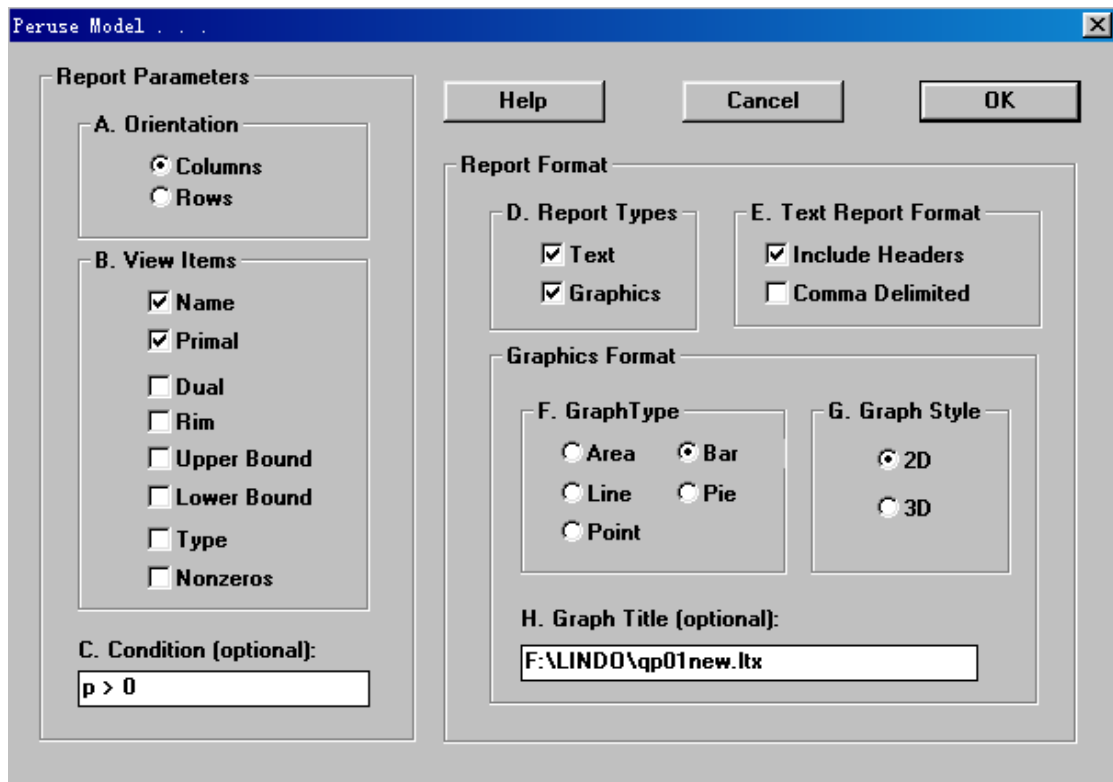


图 2-34 Report Parameters 对话框

- Report | Formulation

Report | Formulation (模型) 显示当前模型 (或其指定的部分)。

- Report | Show Column

Report | Show Column (显示列) 显示模型中你选定的列的信息。

- Report | Positive Definite

Report | Positive Definite (正定) 判断二次规划的目标函数中的二次是否正定 (只对 QP 问题有效, 也就是说只有当前内存中的模型是一个二次规划模型时, 这个命令才有意义)。

*2.6 LINDO 命令窗口

我们前面介绍的基本上的是在 Windows 下菜单模式下驱动 LINDO 运行, 使用起来相当方便. LINDO 还提供了另一种运行模式, 即 “Command-Line” (命令行) 模式. 所谓 “命令行” 模式, 即通过在字符方式下输入一行一行的命令来驱动 LINDO 运行, 因此每个命令也称为 “行命令”. 这操作方式很像老式 DOS 操作系统和 UNIX 操作系统下的运行方式. 在 Windows 下操作系统下, 相信很少有人会选择使用 “命令行” 模式, 但为了对 LINDO 软件介绍的完整性, 我们这里还是简单介绍一下.

你随时可以通过菜单命令 “Window | Open Command Window(Alt+C)” 打开命令窗口, 在命令窗口下操作 (图 2-35)。

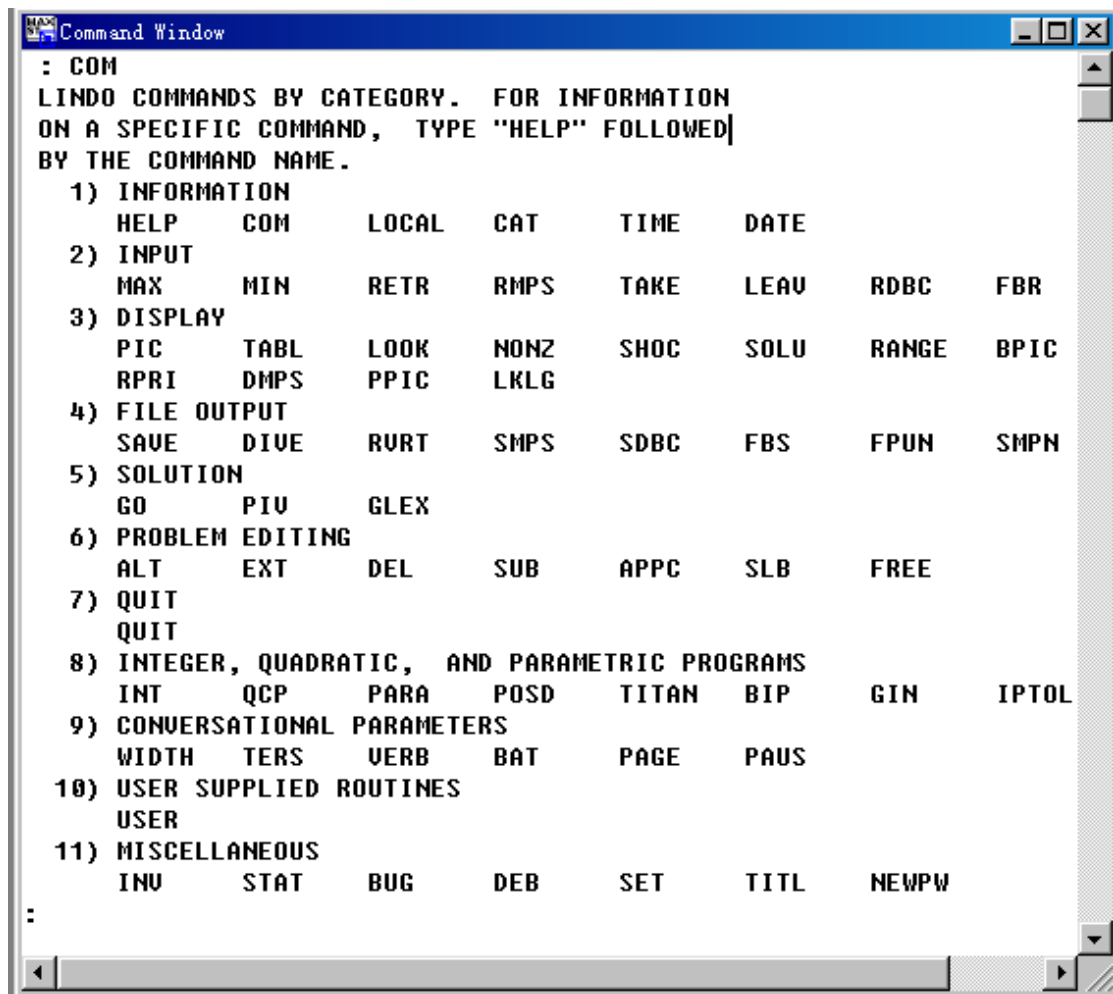


图 2-35 命令窗口

命令窗口下的提示符是“:”，用户在“:”后面可以输入各种 LINDO 的有效命令进行操作，输入也马上显示在命令窗口下。那么，命令窗口下有哪些行命令呢？在命令窗口下，只需输入“COMMANDS”或“COM”（命令），就会看到 LINDO 的所有有效命令（图 2-35）。下面就给出这些命令的简要解释，注意使用命令时如果该命令很长，通过可以只写出前面的若干字母（当然需要注意不要与其他命令的最简单的简写形式）。

2.6.1 INFORMATION（信息类命令）

- HELP 命令

输入“HELP”会显示出 LINDO 的一般帮助信息。通过输入“HELP name”，LINDO 可帮助你了解某个具体的命令，其中“name”是命令名。

- COMMAND（COM）命令

给出按类型分类的 LINDO 命令目录，参见图 2-35。

- LOCAL（LOC）命令

给出该程序的版本信息。

- CATEGORY（CAY）命令

列出 LINDO 命令类型，并可按提示有选择地给出某类型下的所有命令。

- TIME 命令

显示本次启动 LINDO 后累计的运行时间。

- DATE 命令

显示当前命令的日期和时间。

2.6.2 INPUT (输入类命令)

• MAX/MIN 命令

用于输入一个包含目标函数, 约束条件在内的 LP 模型.输入过程如下:在提示符" : "

下输入"MAX"(或"MIN"),继之以自然格式的目标函数作为第一行;再输入"SUBJECT TO"或"SUCH THAT"(可简写为"ST"或"S.T."),后面跟约束条件行.每次回车后将显示"?"提示符.最后,输入"END"回到命令状态模式.

可以看出,这和 Windows 环境下在 LINDO 模式窗口中输入一个程序是类似的.其中,变量名可以由 1~8 个字母或数字型的字符构成,且第一个字符必须是字母.变量系数不能为指数型,列如:258E+29 形成的系数是不允许的,任一系数的整数位数最多为 9 位,小数位数最多位五位.关键词("MAX","ST","END"...)及各行之间必须用一个或多个空格分隔开.空格可以出现在一行之间,但不能出现在变量名中.一个回车符等价一个空格.

例: 下面是同一个问题的两种合法的输入方式:

```
1) MIN 2X + 3Y    SUBJECT TO  -5X - 2Z<=10
    +10X -Y>5    END
```

```
2) MIN 2X + 3Y
    ST  -5X - 2Z
        <10
    10X -Y>+5
    END
```

另外, 任一约束可自由选择名称来代替行号, 例如:

```
DEMAND) 10X -Y>5
```

• RETRIEVE (RETR) 命令

执行该命令可直接从硬盘上的文件中获得一个 LINDO 格式的模型。LINDO 会为你提示可供选择的具体的文件名。能被 RETRIEVE 的模型文件必须是以前经“SAVE”行命令存入的模型(后缀通常为“lpk”, 即 LINDO 压缩格式文件), 而不能是 LINDO 文本格式文件(后缀通常是“ltx”)。

• RMPS 命令

读取一个 MPS 格式文件, 并转化成 LINDI 格式的模型。LINDO 会为你提示可供选

N

! 显示解答

SOLU

! 在屏幕上显示一段提示信息

PAUS 第一模型求解成功, 按 R 键或 Resume 按钮继续!

! 关闭文件“RESULT0201.txt”

RVRT

! 回到执行本命令脚本前的会话模式

BAT

! 结束, 退出本命令脚本文件

LEAV

阅读上面的命令脚本文件 BAT01.txt, 很容易理解其主要功能(实际上, 注释语句已经解释得很清楚了).需要注意的是 RETR exam0201.lpk 命令需要读取 LINDO 模型文件 exam0201.lpk, 所以我们在运行 BAT01.txt 之前需要在当前目录下输入一个 exam0201.lpk 的 LINDO 模型文件.假设这个模型就是 2.1.2 节例 2.1 中输入的简单模型,由于 RETR 只能读取

LINGO 压缩格式的文件(即 LPK 格式文件),所以有两种方法生成这文件:

(1)可以在命令窗口中直接输入这个模型,然后用 SAVE 行命令保存在文件 exam0201.lpk 中.注意:用 SAVE 行命令保存下来的 LINDO 模型文件不是一般的文本文件,而是 LINGO 压缩格式文件,用一般的文本编辑器打开无法正常显示模型,但 LINDO 菜单模式下的模式编辑器可以阅读和编辑它(使用菜单命令“File|Open”打开这个文件即可).

(2)在 2.1.2 节例 2.1 中我们已经把这个模型保存在了文件 exam0201.ltx 中,当时采用的是默认的文件格式,即 LTX 格式(LINDO 文本格式文件).但这个文本文件不能被 RETR 行命令正常打开.不过,我们可以先用菜单命令“File|Open”把文件 exam0201.ltx 打开,然后选择“File|Save As”,这时会出现如图 2-36 的保存文件对话框,将保存文件的格式选择为“LINDO Packed(*.Lpk)”(LINDO 压缩格式的文件,即 LPK 格式文件),这样就可以得到我们想要的文件 exam0201.lpk.

不过,将模型保存为 LPK 格式文件时,说明语句或问中的格式(如字体等)将会自然丢失.所以如果要保留说明语句和原来的文本格式文件千万不要轻易删掉.顺便指出,从图 2-36 可以看出 LINDO 模型文件还有第三种格式,即 MPS 格式,格式的具体规范请参见下一页所提供的信息.

现在,在命令窗口运行“TAKE BAT01.TXT”(也可以在 WINDOWS 模式下选择“File|Take Commands”命令打开脚本文件 BAT01.TXT),将显示如图 2-37 的命令窗口.模型的目标函数、约束条件及解答保存到文件 RESULT0201.TXT 中,然后可以用任何文本处理程序对该结果文件进行编辑和打印输出等.如图 2-38,是用记事本打开的 RESULT0201.TXT 中的内容:首先显示的是模型(执行命令“LOOK ALL”的结果),然后是

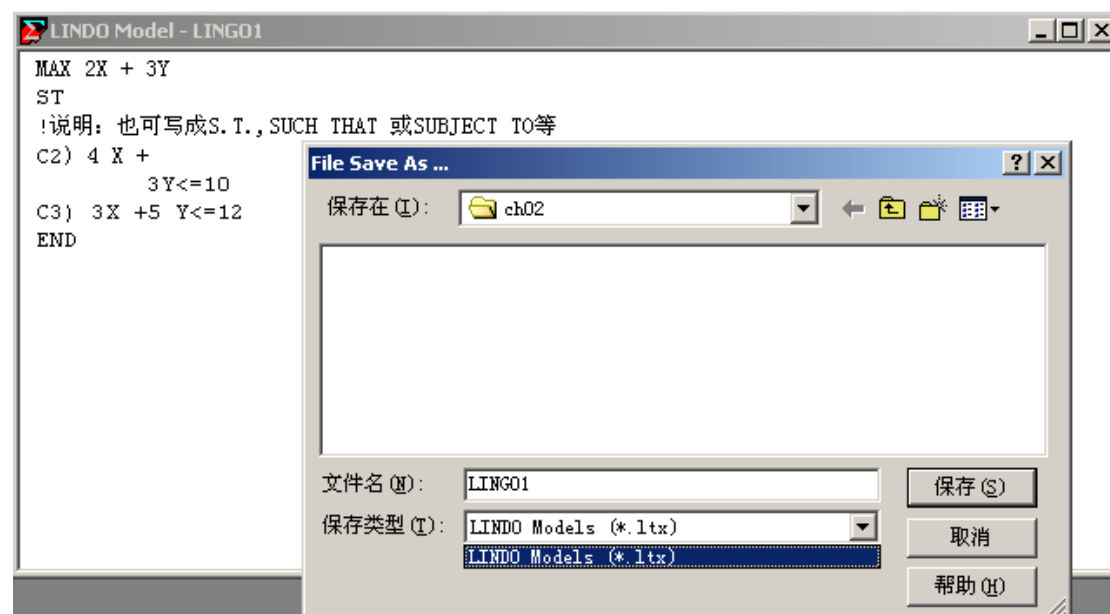
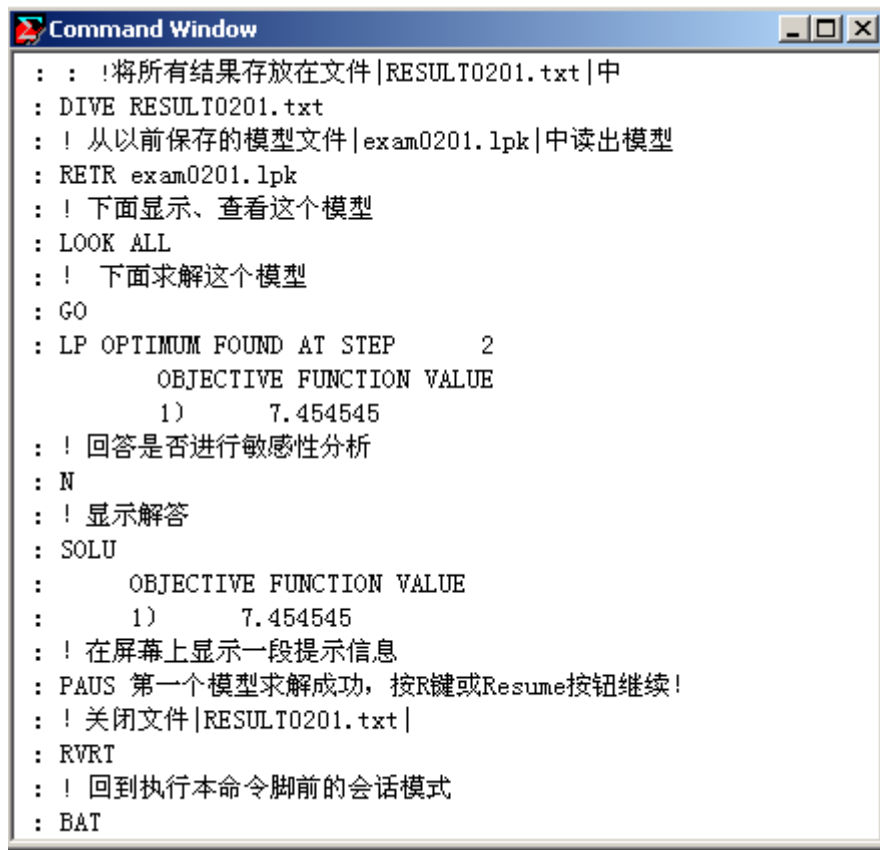


图 2-36 模型文件的保存窗口

求解的答案(执行“GO”命令的结果,且回答“N”表示不需要显示敏感性分析的结果),最后又一次显示答案(执行“SOLU”命令的结果).注意:如果该文件原来有内容,将被覆盖.



```
: : !将所有结果存放在文件|RESULT0201.txt|中
: DIVE RESULT0201.txt
: ! 从以前保存的模型文件|exam0201.lpk|中读出模型
: RETR exam0201.lpk
: ! 下面显示、查看这个模型
: LOOK ALL
: ! 下面求解这个模型
: GO
: LP OPTIMUM FOUND AT STEP      2
      OBJECTIVE FUNCTION VALUE
          1)      7.454545
: ! 回答是否进行敏感性分析
: N
: ! 显示解答
: SOLU
:      OBJECTIVE FUNCTION VALUE
:      1)      7.454545
: ! 在屏幕上显示一段提示信息
: PAUS 第一个模型求解成功, 按R键或Resume按钮继续!
: ! 关闭文件|RESULT0201.txt|
: RVRT
: ! 回到执行本命令脚前的会话模式
: BAT
```

图 2-37 命令脚本文件运行时的窗口

当然,这里我们只是作为一个例子来说明 LINDO 命令的用法,一般情况下在 Windows 环境下实际上这样做的意义不大,因为在 Windows 环境下一切都是所见即所得的图形界面,已经足够直观了.

```

RESULT0201.TXT - 记事本
文件(E) 编辑(E) 格式(O) 帮助(H)

MAX      2 X + 3 U
SUBJECT 10
          C2)  4 X +3 U <=10
          C3)  3 X +5 U <=12
END
LP OPTIMUM FOUND AT STEP   2

      OBJECTIVE FUNCTION VALUE

          1) 7.454545

      VARIABLE           UALUE           REDUCED COST
          X             1.272727           0.000000
          Y             1.636364           0.000000

      ROW    SLACK     OR    SURPLUS     DUAL PRICES
      C2)                0.000000           0.090909
      C3)                0.000000           0.545455

NO. ITERATIONS=         2

      OBJECTIVE FUNCTION VALUE

          1)          7.454545

      VARIABLE           UALUE           REDUCED COST
          X             1.272727           0.000000
          Y             1.636364           0.000000

      ROW    SLACK     OR    SURPLUS     DUAL PRICES
      C2)                0.000000           0.090909
      C3)                0.000000           0.545455

NO. ITERATIONS=         2
  
```

图 2-38 命令脚本文件运行后的输出文件

附录 MPS 格式数据文件

MPS 是 mathematical programming system 的首字母缩写形式,MPS 格式是线性规划中被广泛使用的一种标准数据文件格式,大多数商业性的线性规划软件都能接受这种格式.这种格式据说是由 IBM 公司首先提出来并在其优化软件中使用的.

一个 MPS 文件由下列一些要素构成,其所含信息如下.

- (1)行(ROWS):给出行号、行的类型(等式/不等式)
- (2)列(COLUMNS):给出在目标函数和各约束行的变量名称、行号及数值
- (3)右端项(RHS—right-hang side): 给出非零的右端项
- (4)界(BOUNDS): 给出变量的上下界
- (5)范围(RANGES): 给出约束的上下界

考虑下面这个问题:

```

MAX  500  FIESTA+1600 GRANADA+4300 MARKV+1800 FORD
SUBJECT TO
    2) 12 FIESTA-4 MARKV-2 FORD>=0
    3) FIESTA<=250000
    4) GRANADA+ FORD<=2000000
  
```

```

5) MARKV+ FORD<=1500000
6) FIESTA+GRANADA+ MARKV+ FORD<=3000000
END

```

该问题写成 MPS 格式时,如下(开始和结尾的“12345...”等是为了便于阅读而给出的,并不是模型文件的一部分):

```

      1      2      3      4      5      6
123456789012345678901234567890123456789012345678901234567890-
NAME      LINDO GENERATED MPS FILE(MAX)
ROWS
  N 1
  G 2
  L 3
  L 4
  L 5
  L 6
COLUMNS
  FIESTA      1      500.0000
  FIESTA      2      12.00000000
  FIESTA      3      1.00000000
  FIESTA      6      1.00000000
  GRANADA     1      1600.00000
  GRANADA     4      1.00000000
  GRANADA     6      1.00000000
  MARKV       1      4300.00000
  MARKV       2      -4.00000000
  MARKV       5      1.00000000
  MARKV       6      1.00000000
  FORD        1      1800.00000
  FORD        2      -2.00000000
  FORD        4      1.00000000
  FORD        5      1.00000000
  FORD        6      1.00000000
RHS
  RHS         3      250000.
  RHS         4      2000000.
  RHS         5      1500000.
  RHS         6      3000000.
ENDATA
123456789012345678901234567890123456789012345678901234567890
      1      2      3      4      5      6

```

其中各部分固定书写结构及某些符号注释如下:

ROWS 部分

列 2~3:

L 约束不等式是 \leq 型;

E 约束不等式是 $=$ 型;

G 约束不等式是 \geq 型;

N 目标函数行.

列 5~12: 行名 (不大于 8 个字符).

注 并无特定方式以确定目标函数是 min 型还是 max 型, 一般默认时假设为 min 型.

COLUMNS 部分

列 5~12: 变量名 (不多于 8 个字符);

列 15~22: 变量所在行的行名;

列 25~36: 变量在某行中系数值;

另外可空缺的有:

列 40~47: 另一行名;

列 50~61: 变量在某行中系数值.

RHS 部分

与 **COLUMNS** 部分大致相同.

COUNDS 部分

在这部分, 符号 **UP** 也可用来表示形如 $x \leq 10$ 这样的约束不等式, 符号 **FR** 表示对应变量无上下界 (**FREE**), 例如:

BOUNDS

UP BOUNDNAM X 10.

FR BOUNDNAM XX

表示 $X \leq 10$, **XX** 没有非负限制

RANGES 部分

提供了一种描述约束不等式同时有上下界的格式.

例如: $6 \leq 4X + 3Y - Z \leq 10$.

上式等价于

$$4X + 3Y - Z \leq 10,$$

$$4X + 3Y - Z \geq 6.$$

可在 **ROWS**, **RHS**, **BOUNDS** 三部分中定义如下:

设 $4X + 3Y - Z \leq 10$ 在所在约束行行名为 **ROW1**, 则

ROWS

...

L ROW1

...

RHS

...

RHS1 ROW1 10

RANGES

...

RSET1 ROW1 4

...

注 上面最后一行的 4 表示上下界之差 10^{-6} .

另外, MPS 文件中还可在 COLUMNS 部分中定义整数型变量. 其格式如下:

COLUMNS

| | | |
|----------|------------|------------|
| INTEGERS | ' MARKER ' | ' NTORG ' |
| | | |
| 变量名 | 所在行名 | 系数 |
| | | |
| INTEGERS | ' MARKER ' | ' INTEND ' |

例如:

MIN 3A+4B+6AX+2AY+7BX+4BY

S.T.

| | |
|--------------|--------|
| AX+AY<=8 | ACAP |
| AX+AY>=5 | |
| AX+BX<=7 | XCAP |
| AX+BY>=4 | |
| AY+BY>=6 | YCAP |
| BX+BY-10B<=0 | BSETUP |
| AY-6A<=0 | ASETUP |
| BX <=6 | |
| AX-AY-DIF=0 | DEFN |
| A,B=0,1 | |

则此问题的 MPS 文件为

```

NAME      SAMPLE-PROBLEM
ROWS
  N      COST
  L      ACAP
  L      XCAP
  G      YCAP
  L      BSETUP
  L      ASETUP
  E      DEFN
COLUMNS
INTEGERS      ' MARKER '      ' INTORG '
  A      COST 3.      ASETUP -6.
  B      COST 4.      BSETUP -10.
INTEGERS      ' MARKER '      ' INTEND '
  AX      COST 6.      ACAP 1.
  AX      XCAP 1.      DEFN 1.
  AY      COST 2.      ACAP 1.
  AY      YCAP 1.      ASETUP 1.
  AY      DEFN -1.
  BX      COST 7.      XCAP 1.

```

```

        BX                BSETUP 1.
        BY                COST 4.                YCAP 1.
        BY                BSETUP 1.
        DIF                DEFN -1.
RHS
        RHS1              ACAP 8.
        RHS1              XCAP 7.
        RHS1              YCAP 6.
BOUNDS
        UP BOUNDNAM        BX 6.

        UP BOUNDNAM        A 1.
        UP BOUNDNAM        B 1.
        FR BOUNDNAM        DIF
RANGES
        RSET1              ACAP 3.
        RSET1              XCAP 3.
ENDATA

```

习 题 2

2.1 下面的 LINDO 模型有什么错误？应当如何改正？

MIN=X+y

4. 5 x+10Y>100

51 (x+Y)<1, 000

SLB X-60

SUB Y 40

FREE X, y

END

2.2 下面的 LINDO 模型有什么错误？应当如何改正？

MIN=X+y;

4. 5*x+10Y>100

51*x+51*Y<1000

Gin 2

INT y

2.3 线形规划 (opt 可以是 min 或 max)

$$\begin{aligned}
 \text{opt} \quad & z = -3x_1 + 3x_2 - x_3, \\
 \text{s.t.} \quad & 2x_1 + 3x_2 + x_3 \leq 5, \\
 & 4x_1 + 3x_2 + x_3 \geq 3, \\
 & -x_1 + x_2 + x_3 \geq 2, \\
 & x_1, x_2, x_3 \geq 0
 \end{aligned}$$

的最小值和最大值是多少？相应的最小点和最大点分别是什么？指出积极约束(最优解中取等号的约束)，并指出敏感性分析结果和含义。

2.4 用 LINDO 求解如下整数规划问题 (这里 Z^+ 表示非负整数集合, Z 表示全体整数集合), 并画图进行验证:

(1)

$$\begin{aligned}
 \max \quad & z = 40x_1 + 90x_2; \\
 \text{s.t.} \quad & 9x_1 + 7x_2 \leq 56, \\
 & 7x_1 + 20x_2 \leq 70, \\
 & x_1, x_2 \in Z.
 \end{aligned}$$

(2)

$$\begin{aligned}
 \max \quad & z = 40x_1 + 90x_2; \\
 \text{s.t.} \quad & 9x_1 + 7x_2 \leq 56, \\
 & 7x_1 + 20x_2 \leq 70, \\
 & x_1, x_2 \in Z.
 \end{aligned}$$

2.5 用 LINDO 求解如下整数规划问题:

(1)

$$\begin{aligned}
 \max \quad & z = 5x_1 + 10x_2 + 3x_3 + 6x_4; \\
 \text{s.t.} \quad & x_1 + 4x_2 + 5x_3 + 10x_4 \leq 20, \\
 & x_1, x_2, x_3, x_4 \in Z^+.
 \end{aligned}$$

(2)

$$\begin{aligned} \max \quad & z = 5x_1 + 10x_2 + 3x_3 + 6x_4; \\ \text{s.t.} \quad & x_1 + 4x_2 + 5x_3 + 10x_4 \leq 20, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

2.6 用 LINDO 求解如下整数规划问题:

(1)

$$\begin{aligned} \max \quad & z = -x_1^2 - x_2^2 - x_3^2 - x_4^2 + x_1 + 2x_2 + 3x_3 + 4x_4; \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 \geq 10, \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}^+. \end{aligned}$$

(2)

$$\begin{aligned} \min \quad & z = -x_1^2 - x_2^2 - x_3^2 - x_4^2 + x_1 + 2x_2 + 3x_3 + 4x_4; \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 \geq 2, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

2.7 对习题 2.3 ~ 2.6, 每道题都包含多个模型需要求解. 对于每道题, 将其中一个模型作为初始模型, 编写 LINDO 命令脚本文件, 要求完成下列功能:

(1) 将初始模型在 LINDO 中输入后保存在文件中, 并将计算结果保存在另一个文件中;

(2) 当初始模型中只有部分内容改变时, 修改模型并将修改后的模型保存在文件中, 最后将计算结果保存在另一个文件中.

2.8 习题 1.1 中的哪些问题是连续线形规划模型? 对于其中的连续线形规划模型, 用 LINDO 软件进行求解, 并解释敏感性分析的结果.

2.9 习题 1.1 中的哪些问题是整数线形规划模型? 对于其中的正数线形规划模型, 用 LINDO 软件进行求解. 此时输出的敏感性分析的结果是什么含义?

2.10 习题 1.1 中的哪些问题是二次规划模型? 对于其中的二次规划模型, 用 LINDO 软件进行求解. 此时输出的敏感性分析的结果是什么含义?

第 3 章 LINGO 软件的基本使用方法

3.1 LINGO 入门

3.1.1 LINGO 软件的安装过程和主要特色

LINGO9.0 for Windows 软件的安装过程与 LINDO6.1 for Windows 软件的安装过程类似. 软件安装程序的文件大小通常是 20M 多一点, 当你开始安装后, 仍然需要接受安装协议、选择安装目录 (默认的目录通常是 C: \LINGO9). 安装完成前, 会出现图 3.1 所示的对话框, 这个对话框询问你希望采纳的默认的建模 (即编程) 语言, 系统推荐的是采用 LINGO 语法, 即选项 “LINGO (recommended)” ; 你也可以选择 “LINDO” 将 LINDO 语法作为默认的设置. 在图 3.1 中按下 “OK (确认)” 按钮, 系统就会完成 LINGO 的安装过程. 安装后你也可以随

时通过“LINGO|Options|File Format”命令来修改默认的建模（即编程）语言。

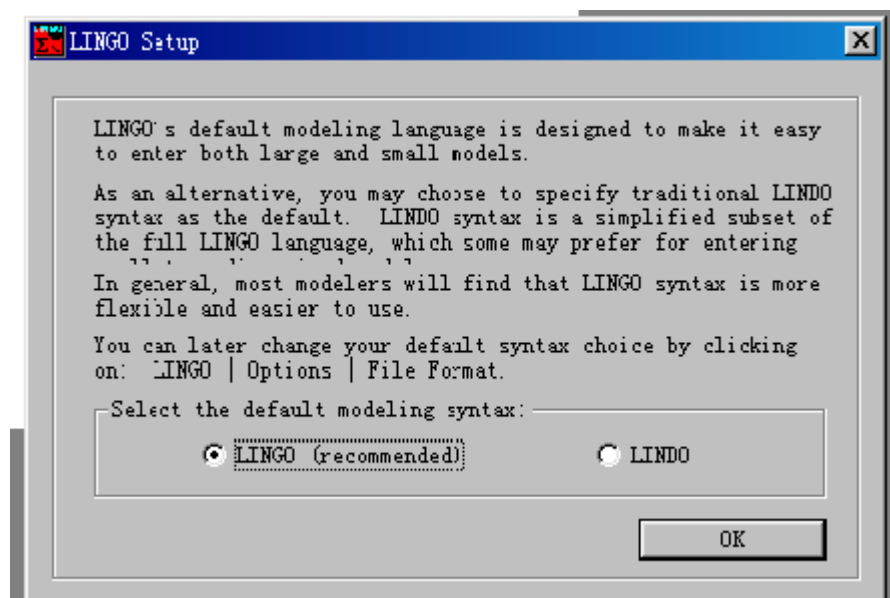


图 3.1 选择编程语法格式

我们下面假设 LINGO9.0 For Windows 软件已经成功安装。第一次运行 LINGO 软件时，系统需要你输入密码（图 3.2），操作方法与 LINDO 完全类似，这里不再重复了。

与 LINDO 类似，LINGO 也有两种命令模式：一种是常用的 Windows 模式，通过下拉式菜单命令驱动 LINGO 运行（多数的菜单命令通常有快捷键，常用的菜单命令在工具

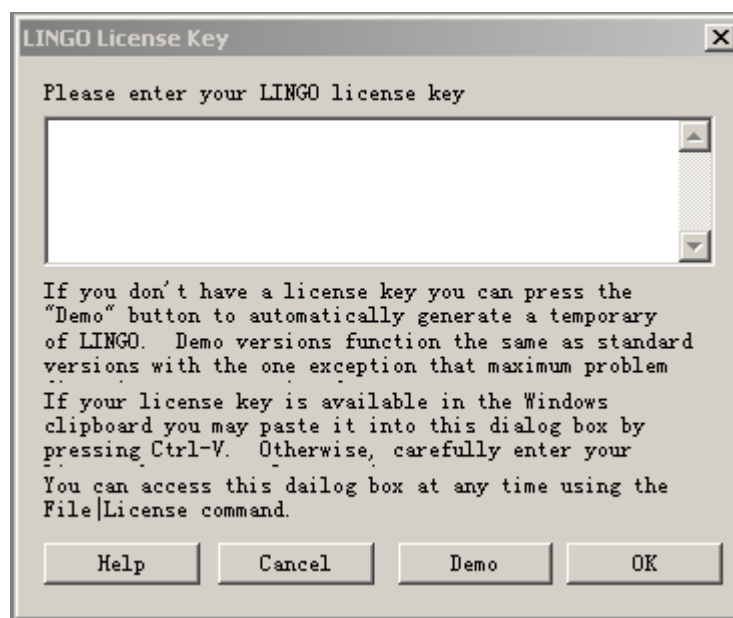


图 3 — 2 输入授权密码

栏中有图标表示的快捷按钮），界面是图形式的，使用起来也比较方便；另一种是命令行（Command-Line）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，其使用界面不是图形式的，而是字符式的，初学者往往不太容易掌握，与第 2 章一样，我们仍然主要在 Windows 菜单驱动下介绍一下命令行模式下的主要行命令。

LINGO9.0 软件比以前的版本有了很大的改进,功能大大加强,性能更加稳定,解答结果更加可靠。我们前面说过,从基本功能看,与 LINDO 相比,LINGO 软件主要具有两大优点:

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

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

前一条是很容易理解的。那么后一条呢?从第 2 章的介绍中可以看出,虽然 LINGO 输入模型的格式与我们数学上对数学规划的表达式非常接近,但是如果我们希望在 LINDO 模型窗口下输入一个规模比较大的模型,那将是一件非常费时费力的事情。例如,如果决策变量有 1000 个,由于 LINGO 不提供数据组或类似的数据结构,我们除了用 $x_1, x_2, \dots, x_{1000}$ 或类似方法表示决策变量外,完全没有其他办法。而对实际企业中的优化问题,决策变量达到几万、几十万个是常有的事,显然前面那种在 LINDO 模型窗口下输入模型的方法几乎是不可能的。而 LINGO 则在这方面通过引入建模语言(常称为“矩阵生成器”)有了很大改进。也就是说,即使你只对解线性规划感兴趣,你也应该学习使用 LINGO。事实上,LINDO 公司目前已将 LINDO 软件从其产品目录中删除了,而将 LINDO 软件的所有功能(包括 LINDO 语法规则)都在 LINGO 中得到了支持,所以不久的将来总有一天人们会废弃 LINDO 软件不在使用的,但 LINDO 的生命力应该还是很强的。

3.1.2 在 LINGO 中使用 LINDO 模型

在 Windows 操作系统下双击 LINGO 图标或从 Windows 操作系统下选择 LINGO 软件运行,启动 LINGO 软件,屏幕上首先显示如图 3-3 所示的窗口。

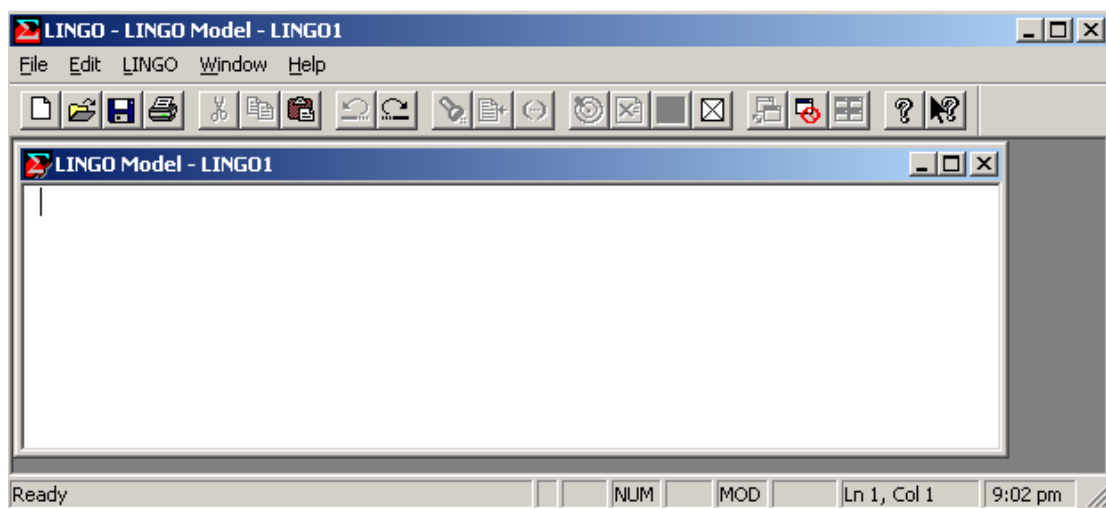


图 3-3 LINGO 初始界面

图 3-3 中最外层的窗口是 LINGO 软件的主窗口(LINGO 软件的用户界面),所有其他窗口都在这个窗口之内。当前光标所在的窗口上标有“LINGO Model - LINGO1”,这就是模型窗口(model window),也就是用于输入 LINGO 优化模型(即 LINGO 程序)的窗口。初步观察可以看到,图 3-3 这个界面与 LINDO 软件的界面非常类似,只是在 LINGO 软件的主窗口中最下面增加了一个状态行(仔细观察,可以发现菜单和工具栏也略有区别)。目前,状态行最左边显示的是“Ready”,表示“准备就绪”;右下角显示的是当前时间,时间前面

是当前光标的位置“Ln 1, Col 1”（即 1 行 1 列）。将来，用户可以用选项命令（**LINGO|Options** 菜单命令）决定是否需要显示工具栏和状态行。

在 LINGO 中可以直接使用 LINDO 语法编写的优化模型（即优化程序）。作为一个最简单的例子，我们看看 2.1.2 节例 2.1 中输入的那个简单例子在 LINGO 下应当如何输入。当时我们把它存入了一个名为 exam0201.ltx 的模型文件中，现在看看如何用 LINGO 把它打开。

例 3.1 在 LINGO 模型窗口中，选择菜单命令“**File|Open(F3)**”，可以看到如图 3—4 所示的标准的“打开文件”对话框。我们看到有各种不同的“文件类型”：

。后缀“lg4”表示 LINGO 格式的模型文件，是一种特殊的二进制格式文件，保存了我们在模型窗口中所能够看到的所有文本和其他对象及其格式信息，只有 LILNGO 能读出它，用其他系统打开这种文件是出现乱码；

。后缀“lng”表示文本格式的模型文件，并且以这个格式保存模型时 LINGO 将给出警告，因为模型中的格式信息（如字体、颜色、嵌入对象等）将会丢失；

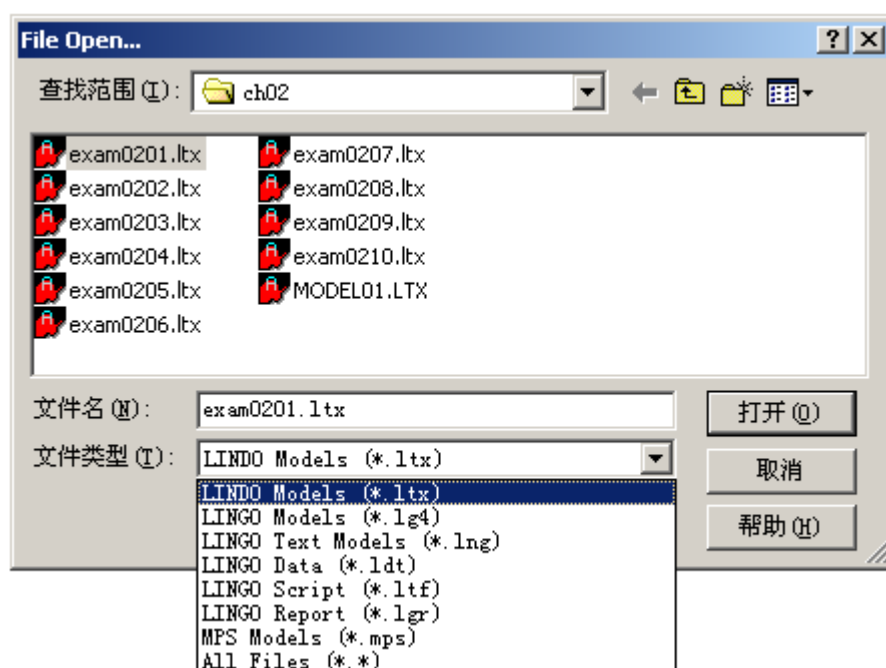
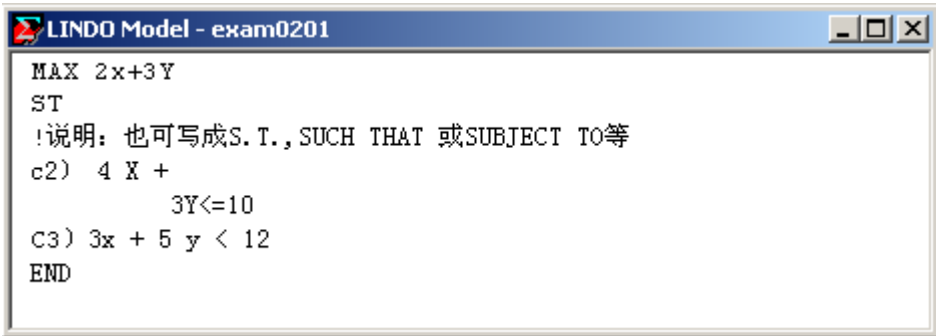


图 3—4 在 LINGO 中打开 LINDO 文件

- 。后缀“ldt”表示 LINGO 数据文件；
- 。后缀“ltf”表示 LINGO 命令脚本文件；
- 。后缀“lgr”表示 LINGO 报告文件；
- 。后缀“ltx”表示 LINDO 格式的模型文件；
- 。后缀“mps”表示 MPS（数学规划系统）格式的模型文件；
- 。“*.*”表示所有文件。

除“lg4”文件外，这里的另外几种格式的文件其实都是普通的文本文件，可以用任何文本编辑器打开和编辑。

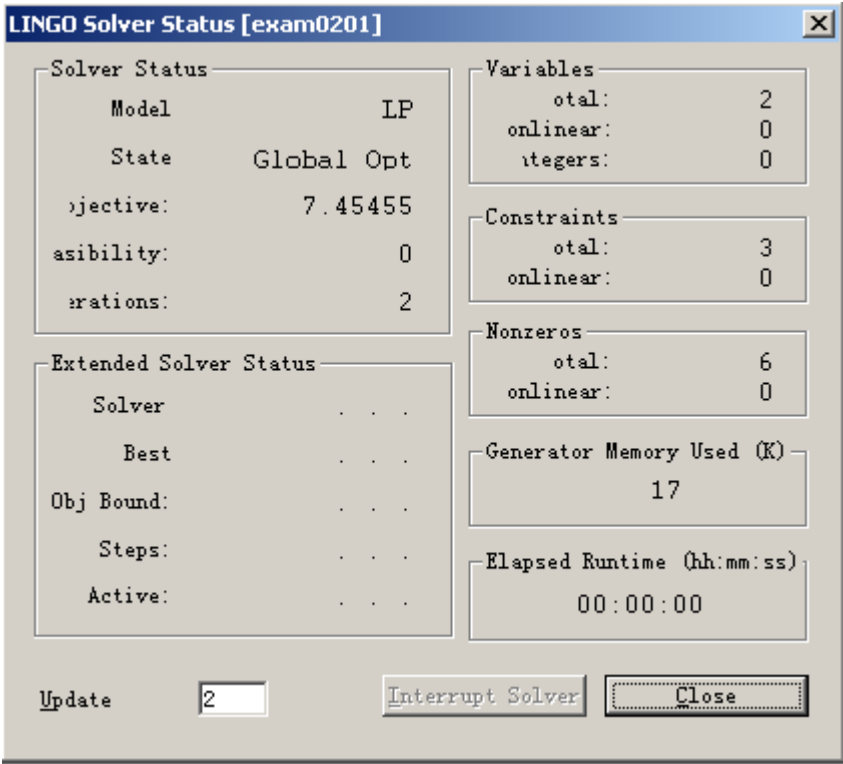
我们找到“exam0201.ltx”文件（需要将“文件类型”选为“*.ltx”），打开这个文件，看到模型窗口中的显示如图 3－5。可以看出，这个模型和在 LINDO 下看到的是一样的。这时我们可以选择“*LINGO*|*Solve*(Ctrl + S)”命令来运行这个程序，屏幕上显示的运行状态窗口如图 3－6 所示，运行结果显示在报告窗口中（如图 3－7 所示，请对照第二章中 LINGO 软件下的结果，看看是否一致）。但这里 LINGO 不询问是否进行敏感性分析，因为敏感性分析需要将来通过修改系统选项启动敏感性分析后，再调用“*Re port*|*Range*”菜单命令来实现。现在同样可以把模型和结果报告保存在文件中（注意上面提到的“文件格式”问题）。



```

LINDO Model - exam0201
MAX 2x+3Y
ST
!说明：也可写成S.T., SUCH THAT 或SUBJECT TO等
C2) 4 X +
      3Y<=10
C3) 3x + 5 y < 12
END
  
```

图 3－5 LINGO 模型窗口中显示的线形规划程序



| Solver Status | |
|---------------|------------|
| Model | LP |
| State | Global Opt |
| Objective: | 7.45455 |
| Feasibility: | 0 |
| Iterations: | 2 |

| Variables | |
|------------|---|
| total: | 2 |
| nonlinear: | 0 |
| integers: | 0 |

| Constraints | |
|-------------|---|
| total: | 3 |
| nonlinear: | 0 |

| Nonzeros | |
|------------|---|
| total: | 6 |
| nonlinear: | 0 |

| Generator Memory Used (K) | |
|---------------------------|--|
| 17 | |

| Elapsed Runtime (hh:mm:ss) | |
|----------------------------|--|
| 00:00:00 | |

| Extended Solver Status | |
|------------------------|-------|
| Solver | . . . |
| Best | . . . |
| Obj Bound: | . . . |
| Steps: | . . . |
| Active: | . . . |

Update

图 3－6 LINGO 状态窗口

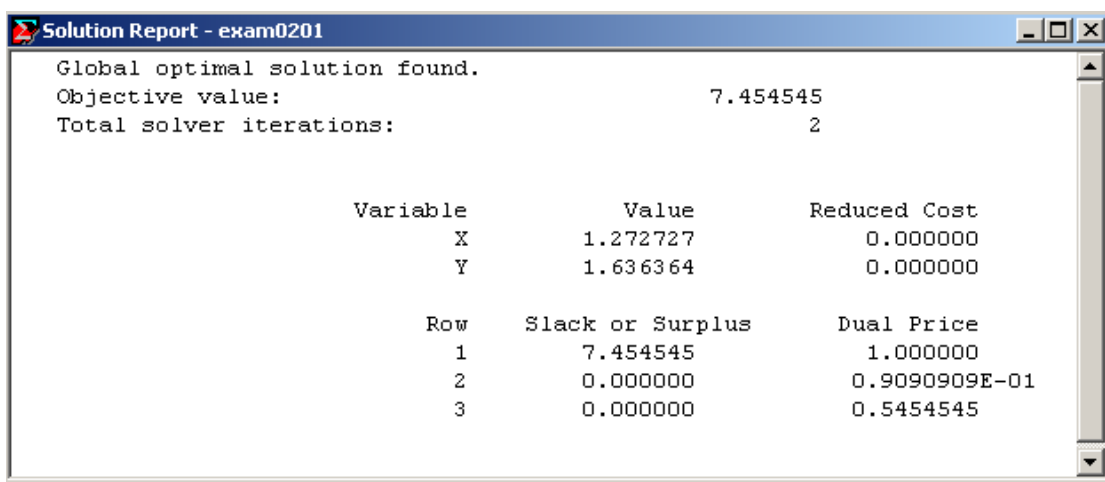


图 3 — 7 LINGO 报告窗口

图 3 — 6 所示的 LINGO 的运行状态窗口也和 LINDO 的运行状态窗口类似，但包含的内容更多些（注意：可能由于 LINDO 和 LINGO 对中文 Windows 系统的兼容性不太好，所以图 3 — 6 中有些显示字符和单词被截掉了）。下面我们给出相应的解释。

Variables(变量数量): 其中包括变量总数 (Total)、非线性变量数 (Nonlinear)、整数变量数 (Integer)。

Constraints(约束数量): 包括约束总数 (Total)、非线性束个数 (Nonlinear)。

Nonzeros(非零系数数量): 包括总数 (Total)、非线性项的系数个数 (Nonlinear)。

Generator Memory Used(K)(内存使用量): 单位为千字节 (K)。

Elapsed Runtime(hh:mm:ss)(求解花费的时间): 显示格式是“时：分：秒”。

需要注意的是，凡是可以从一个约束直接解出变量取值是，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（参见 1.3 节的说明）。总的来说，这些统计值的意义比较清楚，图 3 — 6 中最下面一行的含义也与 LINDO 状态窗口类似，我们下面主要详细介绍一下图 3 — 6 左边的两个框中内容。左上角是求解器（求解程序）状态框 (Solver Status), 含义见表 3 — 1；左下角是扩展的求解器（求解程序）状态框 (Extended Solver Status), 含义见表 3 — 2。

表 3 — 1 LINGO 状态窗口中关于求解器各项的含义

| 域名 | 含义 | 可能的显示 |
|---------------|-------------------------|---|
| Model Class | 当前模型的类型（请 参阅本书第 1 章的介绍） | LP, QP, ILP, IQP, PILP, PIQP, NLP, INLP, PINLP 以 I 开头表示 IP, 以 PI 开头表示 PIP) |
| State | 当前解的状态 | Global Optimum, Local Optimum, Feasible, Infeasible (不可行), Unbounded(无界), Interrupted(中断), Undetermind(未确定) |
| Objective | 当前解的目标函数值 | 实数 |
| Infeasibility | 当前约束不满足的总数（不是不满足的约束个数） | 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） |
| Iterations | 到目前为止的迭代次数 | 非负整数 |

表 3—2 LINGO 状态窗口中关于扩展的求解器各项的含义

| 域名 | 含义 | 可能的显示 |
|--------|-----------|-----------------|
| Solver | 使用的特殊求解程序 | B-and-B（分支定界算法） |

| | | |
|-----------|---|--|
| | | Global (全局最优求解程序) Multistart(最多个初始点求解的程序) |
| Best Obi | 但目前为止找到的可行解的最佳目标函数值 | 实数 |
| Obj Bound | 目标函数值的界 | 实数 |
| Steps | 特殊求解程序当前运行步数: 分支数 (对 B-and-B 程序); 子问题数 (对 Global 程序); 初始点数 (对 Multistart 程序) | 非负整数 |
| Active | 有效步数 | 非负整数 |

备注 在 LINGO 9.0 以前的版本中 (如 LINGO 8.0 中), 一般不能直接用 *File|Import LINGO File(F12)* 命令可以直接把 LINDO 的模型文件转化成 LINGO 模型。

这个菜单命令的意思是“导入 LINDO 文件”(LINGO 9.0 中已无必要, 所以已经被取消了), 运行后屏幕上会显示一个标准的“打开文件”的对话框, 我们在目录下找到 exam0201.ltx, 选定该文件后, 屏幕显示如图 3-8. 可以看出, 这个命令在 LINGO 主窗口中又打开了两个窗口, 一个是命令窗口 (Command

Window, 根据版本不同, 这个窗口也可能不显示出来), 另一个是名为“exam0201”的模型窗口。还可以看出, 当前光标位于命令窗口 (从主窗口左上角的显示结果也可以知道当前的活动窗口是命令窗口), 命令窗口显示的正是从 exam0201.ltx 读出的原始文件; 而“exam0201”窗口才是由 exam0201.ltx 转化而来的等价的 LINGO 模型, 请大家注意, 在第二章的最后, 我们曾经用行命令“SAVE”把同样的 LINDO 模型以压缩文件格式存入了一个名为 exam0201.lpk 的模型文件中。但是经过试验, 笔者发现 LINGO8.0 的菜单命令 *File/Import LINDOFile/saveF (12)* 不能把 exam0201.lpk 正确地转化成 LINGO 模型。即使对于在 LINDO 中去用菜单命令“File/save ‘保存下来的模型, 笔者也多次发现有时不能正确地转化 (转化时出现严重错误)。因此, 本人的经验是: 为了保证将来能将 LINDO 模型移植到 LINGO 中去, 在 LINDO 模型输入时尽量采用“规范化”的格式并以文本文件保存 (例如, 说明语句最好单独占据一行; 行名 (目标和约束的名字) 不要以数字开头; 尽量避免出现汉字和非标准的英文符; 二次规划 (QP) 模型不能被正确转化; 等等), 当然, 由于现在的 LINGO9.0 版本能直接接受 LINDO 格式的输入, 所以不需要进行格式转化, 这个问题也就不存在了。

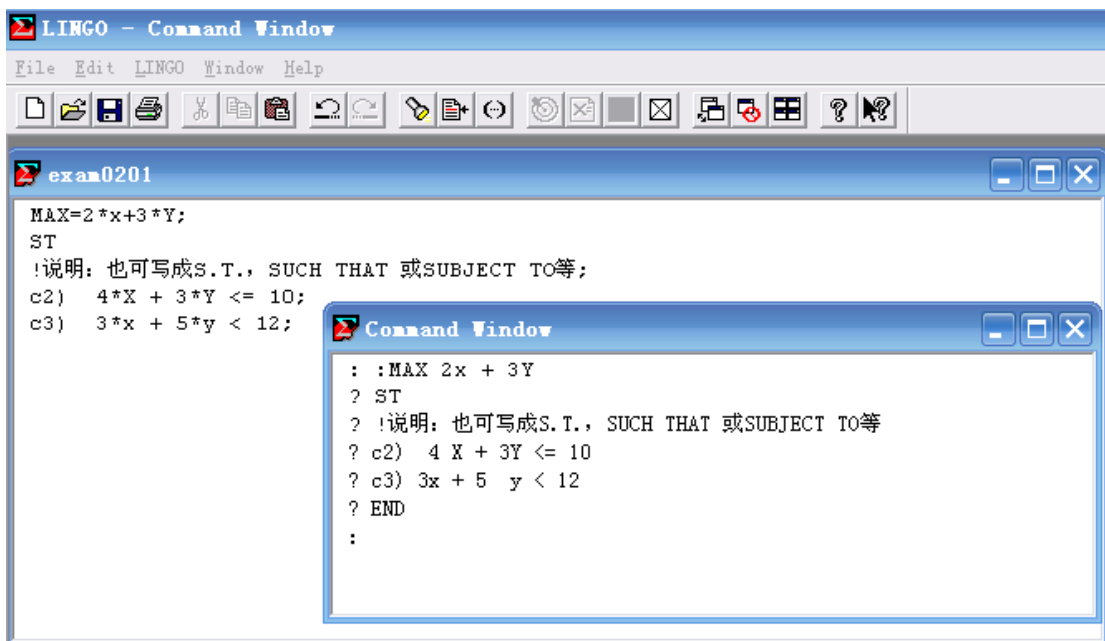


图 3-8 从 LINDO 模型转化成的 LINGO 模型 (LINGO8.0 下)

实际上, 在 LINGO9.0 中, 一样可以把 LINDO 格式的文件转化成 LINGO 格式的文件显示出来。例如, 对图 3-5 的模型, 运行 “LINGO/Generate/DisplayModelModel (Ctrl+G)” 命令, 就会显示图 3-6 的 LINGO 状态窗口和一个如图 3-9 的报告窗口。图 3-9 中报告显示的正是标准格式的 LINGO 模型, 与图 3-8 中的模型类似, 但增加了以 “MODEL:” 开头, “END” 结束的语句, 不过删去了注释语句, 增加了目标行的行号。

比较图 3-8, 图 3-9 和图 3-5 可以发现, 从 LINDO 模型到 LINGO 模型的实质性转化

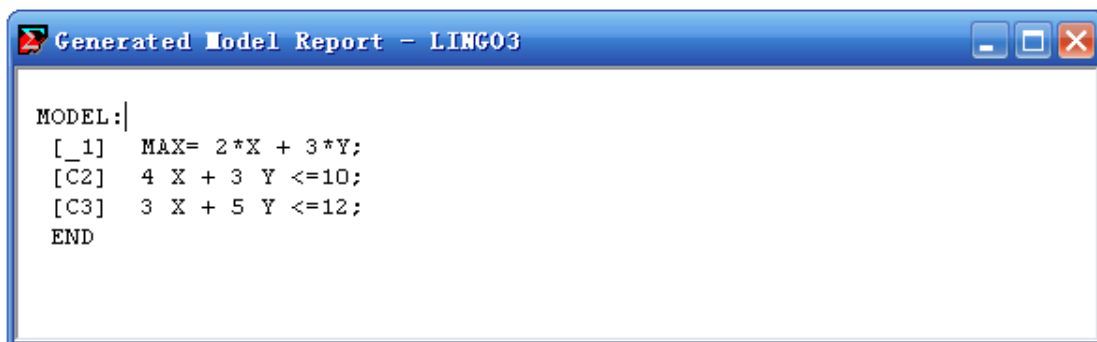


图 3-9 与世隔绝 LINDO 模型等价的 LINGO 模型(LINGO9.0 下)

工作主要在于以下几个方面 (这也是 LINGO 模型的最基本特征):

- (1) 将目标函数的表示方式从 “MAX” 变成了 “MAX=”;
- (2) “ST” (subject to) 在 LINGO 模型中不再需要, 所以删除了;
- (3) 在每个系数与变量之间增加了运算符 “*” (即乘号不能省略);
- (4) 每行 (目标、约束和说明语句) 后面均增加了一个分号 “;”;
- (5) 约束的名字被放到了一对方括号 “[]” 中, 而不是放在右半括号 “)” 之前;
- (6) 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。

3. 1. 3 编写一个简单的 LINGO 程序

如果直接在 LINGO 模型窗口中输入程序, 应该如何做呢? 下面通过一个例子说

明。

例 3.2 我们现在直接用 LINGO 来解 1.2.3 节和 2..4 节的二次规划问题：

$$\begin{aligned} \max \quad & 98x_1 + 277x_2 - x_1^2 - 0.3x_1x_2 - 0.2x_2^2 \\ \text{s. t} \quad & x_1 + x_2 \\ & x_1 \leq 2x_2, \\ & x_1, x_2 \geq 0 \text{ 为整数。} \end{aligned}$$

该模型输入模型窗口 LINGO1 后的形式见图 3-10。对照 2.4 节，我们可以看出用 LINGO 解 QP 比用 LINDO 解要容易输入模型。请注意以下几点：

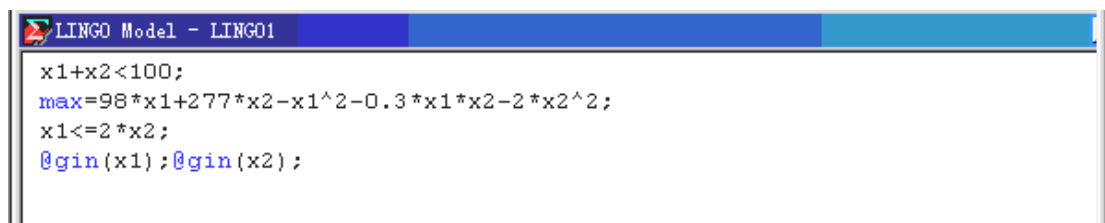


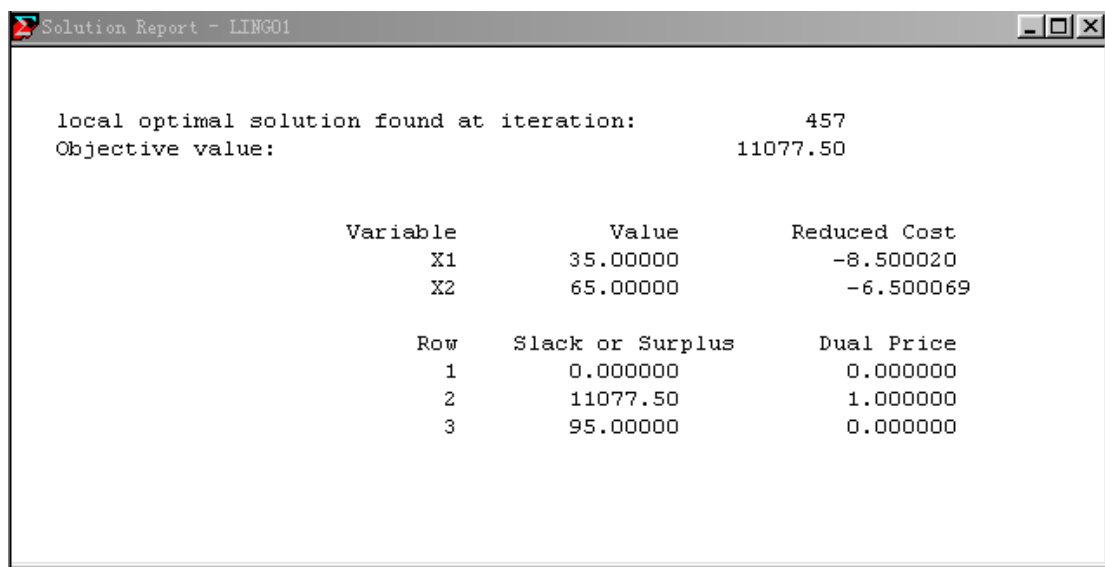
图 3-10

- 我们故意把目标函数没有放在程序的最前面，是为了表明 LINGO 中的语句的顺序是不重要的，因为 LINGO 总是根据“MIN=”语句寻找目标函数，而其他语句都是约束条件（注释语句和 TITLE 语句出外），所以语句的顺序不重要。

- 原来 LINDO 模型中“END”语句后面的限定变量取整数值的语句“GIN X1”和“GIN X2”，这里变成了“@GIN(x1)”和“@GIN(X2)”;但是，在 LINDO 下也可以写成“GIN2”，这里却不可写成“@GIN(2)”，否则 LINDO 将把这个模型看成没有整数变量。

。在 LINGO 中，以“@”开头的都是函数调用，其中整型变量函数（@BIN、@GIN）和上下界限定函数（@FREE、@SUB、@SLB）与 LINDO 中的命令非常类似。只是 LINGO 中函数一律需要以“@”开头；而且 0-1 变量函数不是与 LINDO 中的 INT 命令对应的 @INT 函数（LINGO 中没有 @INT 函数），而是改成了 @BIN 函数。我们将在后面（3.4 节）详细介绍 LINGO 中能够使用的所有函数。

现在运行菜单命令“LINGO/Solve”，则可以得到图 3-11 所示得解答报告，最优整数解 $x = (35, 65)$ ，最大利润=11077.5。结果中最优整数解与 2.4 节相同，但最优值略有不同，估计是计算机误差引起的。你还可以选择运行菜单命令“WINDOW/Status Window”看到图 3-12 所示的状态窗口（这时我们已经把该规划模型保存到了文件 IQP0302.lg4 中，所以这个名字现在也出现在了状态窗口中），从中可以看到目前为止找到的最佳目标值“Best Obj”与问题的上界“Obj BounD”已经是一样的。实际上，如果采用全局最优求解程序（我们将在后面介绍“LINGO/Options”菜单命令时介绍如何激活全局最优求解程序），可以验证它就是全局最优解。此外，LINGO 是将它作为 PINLP（纯整数非线性规划）来求解，因此只告诉我们找到的是局部最优解（为什么 LINGO 不将它作为 PIQP（纯整数二次规划）来求解？本人也不清楚）。



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 |

图 3-11

在本节的最后，我们对 LINGO 的基本用法指出几点注意事项：

- (1) 和 LINGO 一样，LINGO 中不区分大小写字母；但 LINGO 中的变量和行名可以超过 8 个字符，只是不能超过 32 个字符，且仍然必须以字母开头



| Solver Status | | Variables | |
|------------------------|----------------|----------------------------|---|
| Model | PIQP | total: | 2 |
| State | Global Optimum | nonlinear: | 2 |
| Objective: | 11077.5 | integers: | 2 |
| Feasibility: | 0 | Constraints | |
| Iterations: | 51 | 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: | 2 | 3 | |
| Active: | 0 | Elapsed Runtime (hh:mm:ss) | |
| | | 00:00:00 | |

Update

图 3 - 12

- (2) 与 LINDO 相同，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@s1b 另行说明）。
- (3) 与 LINDO 不同，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）。

- (4) LINGO 模型是由一系列语句组成 LINGO 模型的基本单位。每个语句都是以分号“;”结尾的，编写程序时应注意保持模型的可读性，例如：虽然 LINGO 允许每行写多个语句，但最好一行只写一个语句，并且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。
- (5) 与 LINDO 相同，LINGO 中以感叹号“!”开始的是说明语句（说明语句也需要以分号“;”结束）。

3.2 在 LINGO 中使用集合

3.2.1 集合的基本语法和 LINGO 模型的基本要素

我们前面说过，LINGO 同时也是优化问题的一种建模语言，有了它，使用者可以只用输入一行文字就可以建立起含有大规模变量的目标函数和成千上万条件约束。掌握这种最优化模型语言是非常重要的，与 LINGO 相比，这可使输入较大规模问题的过程得到简化。

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。什么是集合呢？我们通过下面的一个简单例子开始来进行介绍。

例 3.3 SAILCO 公司需要决定下四个季度的帆船生产量。下四个季度的帆船需求量分别是 40 条,60 条,75 条,25 条,这些需求必须按时满足。每个季度正常的生产能力是 40 条帆船,每条船的生产费用为 400 美元。如果加班生产,每条船的生产费用为 450 美元,每个季度末,每条船的库存费用为 20 美元。假定生产提前期为 0。初始库存为 10 条船。如何安排生产可使总费用最小？

我们用 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,现在我们可以写出这个问题的模型。首先,目标函数是所有费用的和:

$$\min \sum_{I=1,2,3,4} \{400RP(I) + 450OP(I) + 20INV(I)\}。 \quad (5)$$

约束条件主要有两个:

- (1) 能力限制

$$RP(I) \leq 40, I=1, 2, 3, 4; \quad (6)$$

- (2) 产品数量的平衡方程

$$INV(I) = INV(I-1) + RP(I) + OP(I) - DEM(I), \\ I=1, 2, 3, 4; \quad (7)$$

$$INV(0) = 10; \quad (8)$$

当然,还要加上变量的非负约束,构成了这个问题的模型(可以看出是 LP 模型)。

可以看出,如果利用数组的概念,这个模型是比较容易建立的。然而,由于 LINGO 中没有数组这样的数据结构,我们只能对每个季度分别定义变量,如正常产量就要有 4 个变量 RP1,RP2,RP3,RP4 等;对未知数 OP,INV 也是一样。这样,写起来就比较麻烦,尤其如果不是 4 个季度而是更多(如 1000 个季度)的时候。

记 4 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,如图 3-13 所示。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)。表 3-3 更清楚地列出了集合元素及其属性所确定的所有变量。

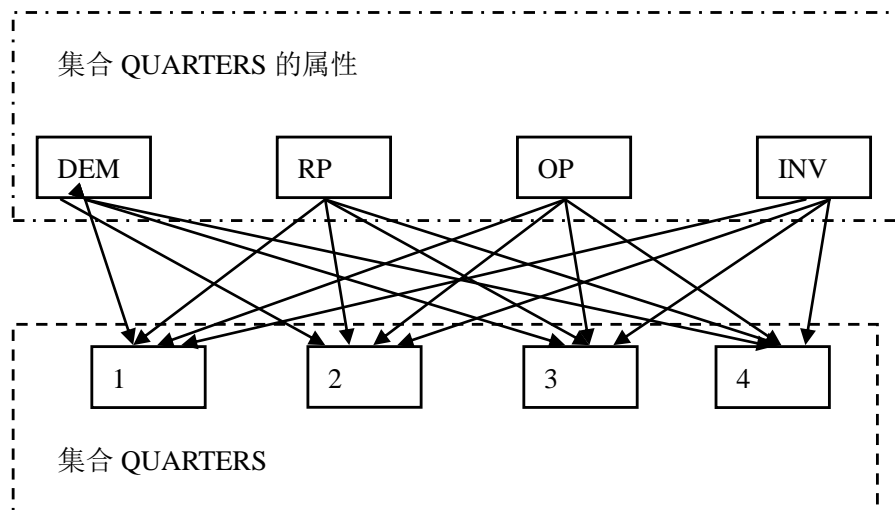


图 3-13 集合及其属性

表 3-3 集合元素及其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 QUARTERS 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) |

下面我们看看在 LINGO 中具体如何定义集合及其属性.例 3.3 的 LP 模型在 LINGO 中的一个典型输入方式见图 3-14.我们可以看到这个输入以"MODEL:"开始,以"END"结束,它们之间由语句组成,可以分成三个部分:

(1) 集合定义部分(从"SETS"到"ENDSETS"):定义集合及其属性,语句"QUARTERS/1, 2, 3, 4/:DEM,RP,OP,INV;"就是定义了上面所说的集合 $QUARTERS=\{1, 2, 3, 4\}$,以及对应于该集合的属性 DEM,RP,OP,INV,其结果正是定义了表 3-3 所列的 16 个变

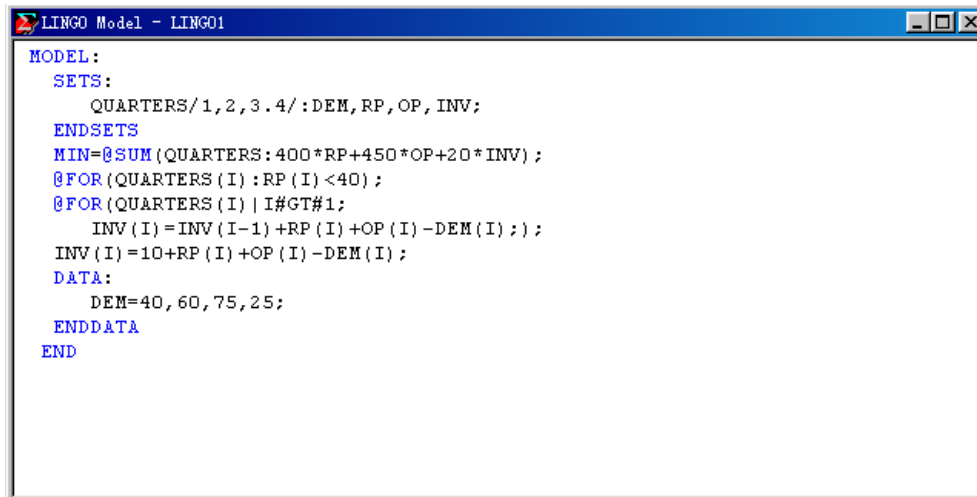


图 3-14

量名(并不一定都是决策变量,如下面马上就看到 DEM 对应的 4 个变量是已知的).

(2) 数据输入部分(从"DATA:"到"ENDDATA"): "DEM=40,60,75,25;"给出常量 DEM(给定的需求量)的值,即 DEM(1=40),DEM(2)=60,DEM(3)=75,DEM(4)=25.

(3) 其他部分:给出优化目标和约束.

目标函数("MIN="后面所接的表达式)是用求和函数

"@SUM(集合(下标):关于集合的属性的表达式)"

的方式定义的,这个函数的功能是对语句中冒号":"后面的表达式,按照":"前面的集合的方式进行指定的下标(元素)进行求和,本例中函数也可以等价地写成"@SUM(QUARTERS(i):400*RP(i)+450*OP(i)+20*INV(i))",这里"@SUM"相当于求和符号" \sum ",而"QUARTERS(i)"相当于" $i \in \text{QUARTERS}$ "的含义(请与目标函数的数学表达式(5)比较一下,你会发现它们几乎式一样的!).只是由于本例中目标函数对集合 QUARTERS 的所有元素(下标)都要求和,所以我们在图 3-14 的相应语句中将下标 i 省去.

约束是循环函数"@FOR(集合(下标):关于集合的属性的约束关系式)"的方式定义的,意思是对冒号":"前面的集合的每个元素(下标),冒号":"后面的约束关系式都要成立.我们先看能力限制(6),即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<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,即这个语句可以简化成"@FOR((QUARTERS(I): RP<40);",效果相同.

但是,对于产品数量的平衡方程(7)及(8)而言,由于下标 I=1 时的约束关系式与 I=2,3,4 时有所区别,所以我们这时不能省略下标"I",实际上,I=1 时的约束关系式(7)要用到变量 INV(0),但我们定义的属性变量中 INV 是不包含 INV(0)的(不过,我们知道 INV(0)是一个已知的常数,即 INV(0)=10),为了区别 I=1 和 I=2,3,4,我们在程序中把 I=1 时的约束关系式单独写出,即"INV(1)=10+RP(1)+OP(1)-DEM(1);"(这样一来,约束(8)实际上已经不再需要了);而对 I=2,3,4 对应的约束,我们对下标集合的元素(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,意思是"大于(greater than 的字首字母缩写)"(其他逻辑运算符将在后面 3.3 节介绍).

现在运行菜单命令"LINGO|Solve",则可以得到图 3-15 所示的解答报告,全局最优解 RP=(40,40,40,25),OP=(0,10,35,0),最小成本=78450.这就是我们模型的计算结果.

不过有一点值得注意:由于我们在图 3-14 的输入中没有给出行名,所以行名是系统自动生成的.图 3-15 中约束的行名分别是行号 1~9,我们怎么知道那个约束对应那个行号?可以这样做:选择菜单命令"LINGO|Generate|Disply Model(Ctrl+G)"(你应该

| Variable | Value | Reduced Cost |
|----------|----------|--------------|
| DEM(1) | 40.00000 | 0.000000 |
| DEM(2) | 60.00000 | 0.000000 |
| DEM(3) | 75.00000 | 0.000000 |
| DEM(4) | 25.00000 | 0.000000 |
| RP(1) | 40.00000 | 0.000000 |
| RP(2) | 40.00000 | 0.000000 |
| RP(3) | 40.00000 | 0.000000 |
| RP(4) | 25.00000 | 0.000000 |
| OP(1) | 0.000000 | 20.00000 |
| OP(2) | 10.00000 | 0.000000 |
| OP(3) | 35.00000 | 0.000000 |
| OP(4) | 0.000000 | 50.00000 |
| INV(1) | 10.00000 | 0.000000 |
| INV(2) | 0.000000 | 20.00000 |
| INV(3) | 0.000000 | 70.00000 |
| INV(4) | 0.000000 | 420.0000 |

| Row | Slack or Surplus | Dual Price |
|-----|------------------|------------|
| 1 | 78450.00 | -1.000000 |
| 2 | 0.000000 | 30.00000 |
| 3 | 0.000000 | 50.00000 |

图 3-15 例 3.3 模型的求解结果

记得在上一节中,我们也使用过这个命令的),可以得到展开形式的模型如图 3-16 所示,这样就可以看到完整的模型了,也能确定行号了(行号放在方括号"[]"中,且数字前面带有下划线"-").不过,最好还是在输入模型时用户主动设定约束的行名(即约束名),这样使程序更清晰些.前面已经看到过单一约束的行名的行名设置方法,就是将行名放在方括号"[]"中,置于约束之前,那么,在使用集合的情况下,如何设置行名?后面将结合具体例子再进一步介绍.

```

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 *INV4_;
[ _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

```

图 3-16 例 3.3 模型的展开形式

下面小结一下 LINGO 模型最基本的组成要素,LINGO 中建立的优化模型出除了上面例子

中给出的三大部分语句外,还可以包括一个“初值设定(LINT)”部分;此外,从LINGO9.0开始,还可以增加一个“计算(CALC)”部分。因此,一般来说,LINGO中建立的优化模型可以由5个部分组成,或称为5段(section):

(1) 集合段(SETS):这部分要以“SETS:”开始,以“ENDSETS”结束,作用在于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不是4而是1000时,我们也没有必要把1、2、……、1000全部一个一个列出来,而可以如下定义QUARTERS集合:

QUARTERS/1..1000/: DEM, RP, OP, INV;

即“1..1000”的意思就是从1到1000的所有整数(前面的例子中只有4个元素,所以没有写成“1..4”而是全部列出来了)。

(2) 目标与约束段:这部分实际上定义了目标函数、约束条件等,但这部分并不有段的开始和结束标记,因此实际上就是除其他4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,可在具体使用中体会其功能和用法(详见3.3节)。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):这部分要以“DATA:”开始,以“ENDDATA”结束,作用在于对集合的属性(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(回车的作用也等价于一个空格),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。但这仅用于对单个变量赋值,而不能用于属性变量(数组),输入语句格式为:“变量名=? ;”。例如,上面的例子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LINGO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):这部分要以“INIT:”开始,以“ENDINIT”结束,作用在于对集

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): 这部分要心 “CALC:” 开始，以 “ENDCALC” 结束，作用在于对一些原始数据进行计算处理（这种处理是在数据段的数据输入完成以后、LINGO 开始正式求解模型之前进行的）。为什么要设计这个段？这是因为在实际问题中，输入的数据通常是原始数据，不一定能在模型中直接使用，我们就可以在这个段对这些原始数据进行一定的“预处理”，得到我们模型中真正需要的数据。例如，对上面的例子，如果我们希望得到全年的总需求，则可以增加这个段（这里在只是作为“计算段”的一个例子，其实总需求和季度平均需求在这个问题的模型中并没有实际应用价值）：

```
CALC:
T_DEM=@SUM(QUARTERS:DEM);!总需求;
A_DEM=T_DEM/@size(QUARTERS);!平均需求;
ENDCALC
```

可以看出，在计算段中也可以使用集合函数（其中函数 @size(QUARTERS) 表示 QUARTERS 示的元素个数，这里了就是 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他地方作为常数使用了）。

请大家注意，在计算段中语句是顺序执行的，所以上面的一两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。

3.2.2 基本集合与派生集合

下面再用 LINGO 来解在 1.2.3 节中介绍的料场选址问题。

例 3.4 建筑工地的位置（用平面坐标 a, b 表示，距离单位：km）及水泥日用量 d （单位：t）由表 3-4 给出。目前有两个临时料场位于 P（5，1），Q（2，7），日储量各有 20t。求从 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，节省的吨公里数最大？

表 3-4 工地的位置 (a b) 及水泥日用量 d

| | 1 | 2 | 3 | 4 | 5 | 6 |
|----------|------|------|------|------|-----|------|
| <i>a</i> | 1.25 | 8.75 | 0.5 | 5.75 | 3 | 7.25 |
| <i>b</i> | 1.25 | 0.75 | 4.75 | 5 | 6.5 | 7.75 |
| <i>d</i> | 3 | 5 | 4 | 7 | 6 | 11 |

1.2.3 节中已经建立了这个问题的优化模型。记工地位置为 (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) ，日用量为 e_j ， $j=1, 2$ ；从料场 j 向工地 i 的运送量为 c_{ij} 。这个优化问题的数学规化模型是

$$\begin{aligned} \min f &= \sum_{j=1}^2 \sum_{i=1}^6 c_{ij} \sqrt{(x_j - a_i)^2 + (y_j - b_i)^2} \\ s.t. \sum_j c_{ij} &= d_i, i = 1, 2, 3, 4, 5, 6 \\ \sum c_{ij} &\leq e_j, j = 1, 2 \end{aligned}$$

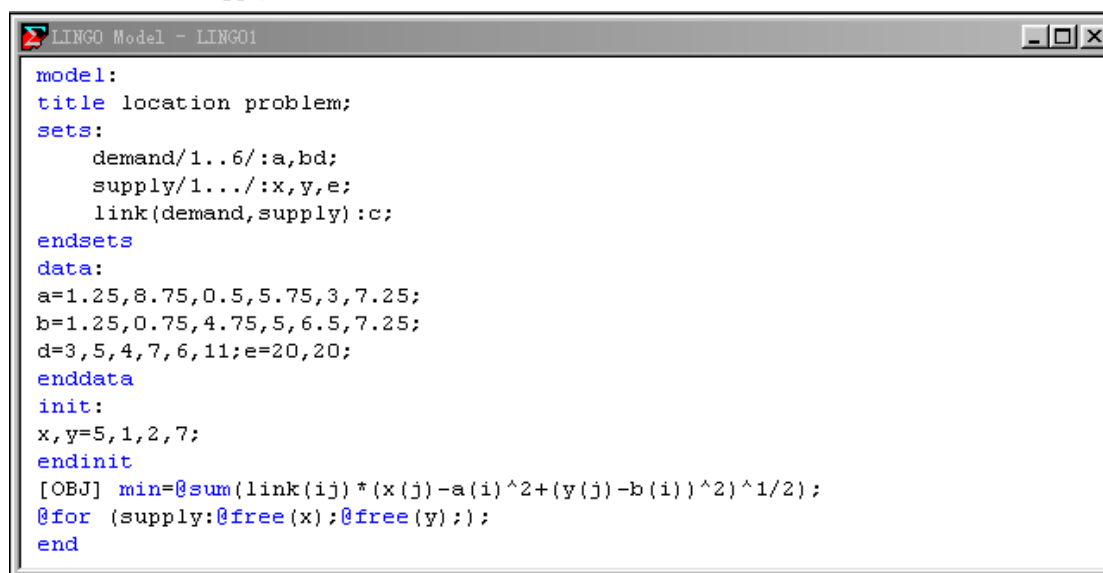
当使用现在临时料场时，决策变量只有 c_{ij} （非负），所以这是 LP 模型；当为新建料场选址时决策变量为 c_{ij} 和 x_j, y_j ，由于目标函数 f 对 x_j, y_j 是非线性的，所以在新建料场时是 NLP 模型。我们现在先解 NLP 模型，而把瑞临时料场作为初始解告诉 LINGO。

尝试将这个模型输入 LINGO 时，利用上面介绍的集合的概念，显然可以定义需求点 demand 和供应点 supply 两个集合分别有 6 个和 2 个元素（下标）。但是，你可能会遇到一个困难：决策变量（运送量） c_{ij} 既不是只与集合 demand 相关的属性，而与这两个集合都有关系的。这样的属性应该如何定义？

我们前面说过，集合的属性相当以集合的元素为下标的数组。那么，这里的 c_{ij} 不正是相当于二元数组吗？它的两个下标分别来自于集合 demand 和 supply，这就启发我们可以利用集合 demand 和 supply，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。

具体的输入程序如图 3-17 所示。我们在集合段定义了三个集合，其中集合 demand 和 supply 的属性的含义与上一个例子类似，而 link 则是在前两个集合的基础上定义的一个新集合。语句

link (demand, supply): c;



```

LINGO Model - LING01

model:
title location problem;
sets:
demand/1..6/:a,b,d;
supply/1..2/:x,y,e;
link(demand,supply):c;
endsets
data:
a=1.25,8.75,0.5,5.75,3,7.25;
b=1.25,0.75,4.75,5,6.5,7.25;
d=3,5,4,7,6,11;e=20,20;
enddata
init:
x,y=5,1,2,7;
endinit
[OBJ] min=@sum(link(i,j)*(x(j)-a(i))^2+(y(j)-b(i))^2)^1/2;
@for(supply:@free(x);@free(y));
end

```

图 3-7 例 3.4 的模型窗口

表示集合 link 中的元素就是集合 demand 和 supply 就是的元素组合成的有序二元组，从数学上看 link 就是 demand 和 supply 的笛卡积，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）。

正是由于这种表示方式与矩阵的表示非常类似，LINGO 建模语言也称为矩阵生成器

(matrix generator)。类似于 demand 和 supply 这种直接把元素列举出来的集合称为**基本集合** (primary set, 也称为“原始集合”), 而把 link 这种基于其他集合而派生出来的二维或多维集合称为**派生集合** (derived set 也为“导出集合”)。由于 demand 和 supply 生成派生集合 link 的父集合。

本模型中数据段的含义也是容易理解的。本模型中还包括了初始段, 请特别注意其中 “x, y=5, 1, 2, 7;” 语句的实际赋值顺序是 $x=(5, 2)$, $y=(1, 7)$, 而不是 $x=(5, 1)$, $y=(2, 7)$ 。也就是说, LINGO 对数据是按列赋值的, 而按行。当然, 直接写成两个语句 “x=5, 2, y=1, 7;” 也是等价的。同样道理, 数据段中对常数数组 a, b 的赋值语句也是可以写成 $a, b=1.25, 1.25, 8.75, 0.75, 0.5, 4.75, 5.75, 5, 3, 6.5, 7.25, 7.25$;

请注意我们前面说过, 这时分割数据的空格与逗号 “,” 或 “回车” 的作用是等价的。

程序接下来定义目标约束, 与前面例 3.3 中的方法是类似的 (但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, 所以我们在约束的最后 (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。此外, 我们在程序开头用 Title 语句对这个模型取了一个标题 “Location Problem” (见模型开始的 “MODEL:” 下面一行); 并且对目标行 ([OBJ]) 和两类约束 (DEMAND-CON、SUPPLY-CON) 分别进行了命名 (请特别注意这里命名的特点)。

大家仔细阅读、理解了图 3-17 的程序后, 现在就可以运行菜单命令 “LINGO|Solve”, 很快得到解答报告 (显示界面略, 请特别注意结果中的约束条件 (行名) 也好似有下标的): 局部最优解 $x(1)=7.249997, x(2)=5.695940, y(1)=7.749998, y(2)=4.928524, c(\text{略})$, 最小运量 $=89.8835 (t \cdot km)$ 。

现在我们来看看对于这个问题的 NLP 模型, 最小运量 89.8835 是不是全局最优解。我们考虑用全局最优求解器, 求解图 3-17 中的模型 (如果你使用的是试用版软件, 则可能不能用全局最优求解器求解本例, 因为问题规模已经太大了)。激活全局最优求解程序的方法, 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” (我们将在后面介绍 “LINGO|Options” 菜单命令的具体用法)。全局最优求解程序花费的时间可能是很长的, 所以为了减少计算工作量, 我们对 x, y 的取值再做一些限制。虽然理论上新建料场的位置可以是任意的, 但我们可以很直观地想到, 最佳的料场位置不应该离工地太远, 无论如何至少不应该超出在 6 个工地所决定的坐标的最大、最小值决定的矩形之外, 即: $0.5 \leq x \leq 8.75, 0.75 \leq y \leq 7.75$ 。可以用 @bnb 函数加上这个条件取代模型 END 上面的行, 运行 NLP 模型, 发现全局最优求解程序花费的时间仍然很长, 图 3-18 是运行 27 分 35 秒时人为终止求解 (按下 “Interrupt Solver” 按钮) 时对应的求解状态窗口。

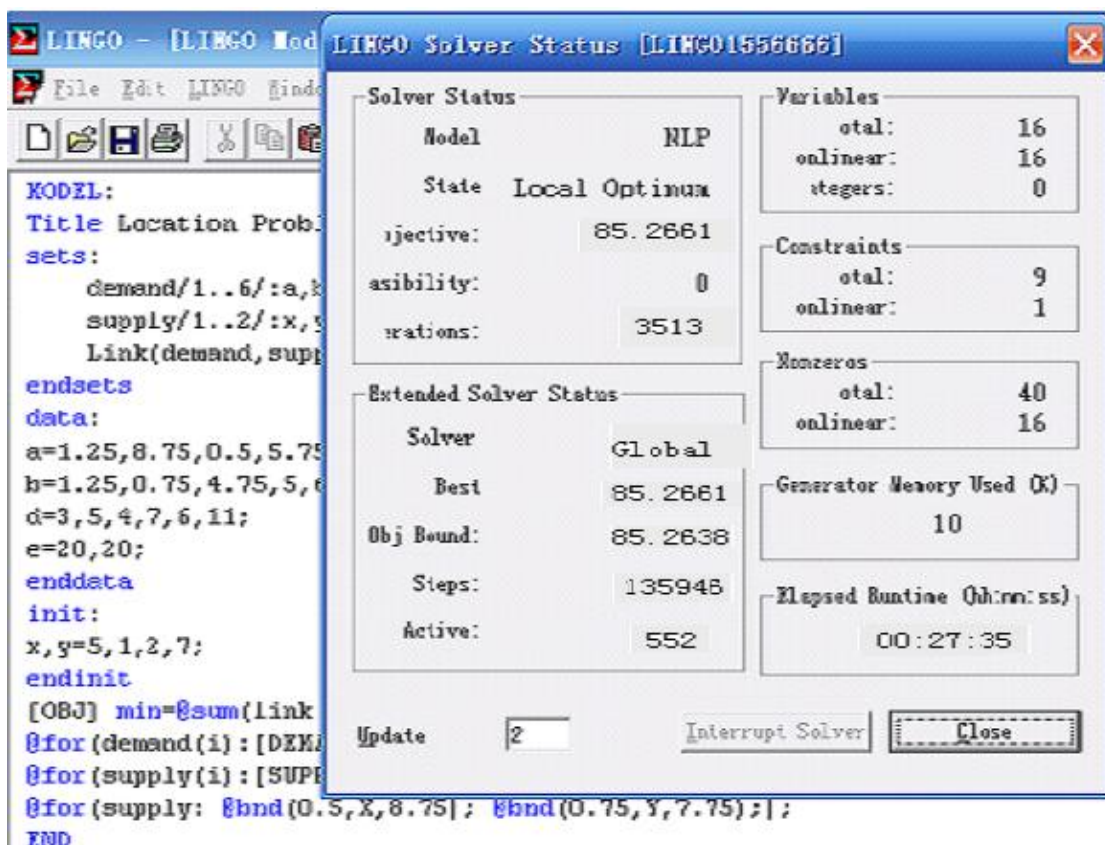


图 3-18 例 3.4 的模型窗口和全局求解器的状态窗口

从图 3-18 可以看出，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=85.2661）相差已经非常小，可以认为已经得到了全局最优解。部分结果见图 3-19，这就可以认为是我们模型的最后结果。在图 3-20 中，我们可以画出料场和工地的位置示意图，其中标有“*”号的是料场，标有“+”号的是工地。

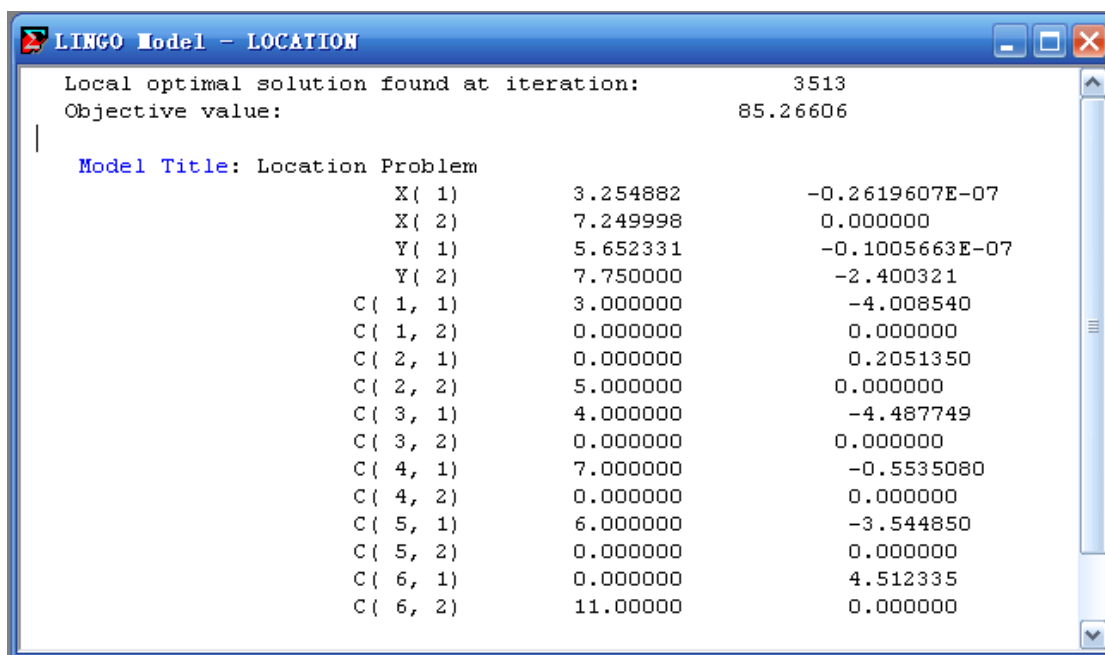


图 3-19 例 3.4 计算结果

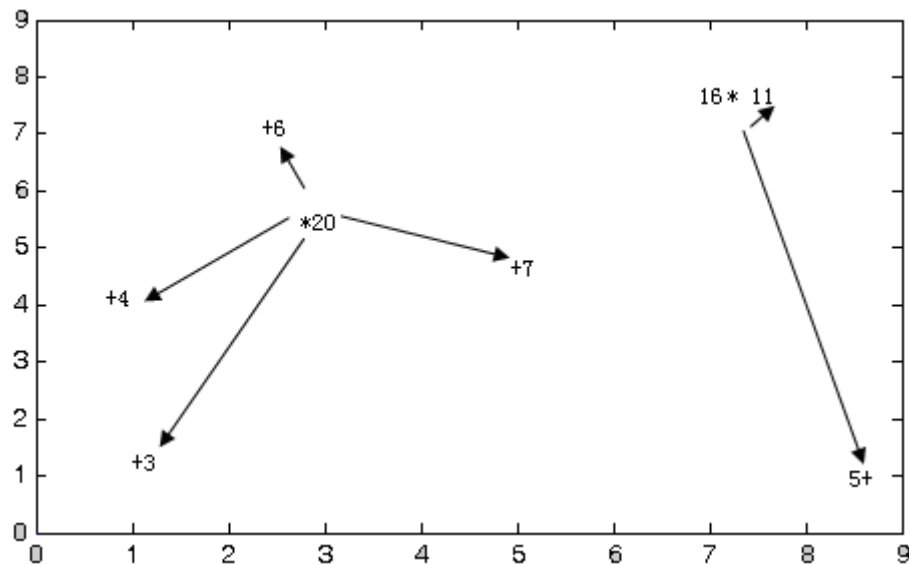


图 3-20 工地与料场示意图

我们还可以指出：如果要把料场 P (5, 1), Q (2, 7) 的位置看成是已知并且固定的，这时 LP 模型。只需要在图 3-17 中把初阶段的 “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 ($t \bullet km$)。请读者自己不妨一试。

3.2.3 稠密集合与稀疏集合

在 3.2.3 节我们介绍了在 LINGO 中可以定义和使用两类集合：基本集合和派生集合。前面的例子中我们把派生集合 link 的元素定义为 demand 和 supply 的笛卡儿积，即包含了两个基本集合构成的所有二元有序对。这种派生集合称为**稠密集合**（简称稠集）。有时候，在实际问题中，一些属性（数组）可能只在笛卡儿积的一个真子集合上定义，而不是在整个稠集上定义，LINGO 能不能做到这一点呢？

答案是肯定的。其实，在 LINGO 中，派生集合的元素可以定义为只是这个笛卡儿积的一个真子集合，这种派生集合称为**稀疏集合**（简称疏集）。下面我们通过一个例子来说明。

例 3.5（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。假设图 3-21 表示的是该公路网，节点表示货车可以停靠的城市，弧上的权表示两个城市之间的距离（百公里）。那么，货车从城市 S 出发到达城市，如何选择行驶路线，使所经过的路程最短？

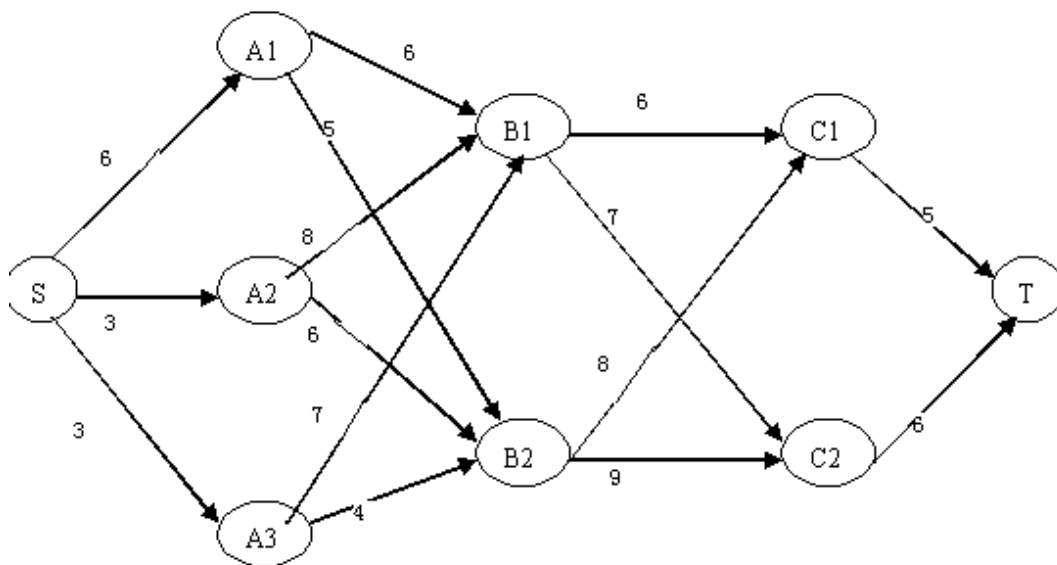


图 3-21 最短路问题的例子

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, 为了得到从 S 到 T 的最优行驶路线, 我们只需要先求出从 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。而从 A_i ($i=1, 2$) 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)。

也就是说, 此例中我们可以把从 S 到 T 的行驶过程分成 4 个阶段, 即 S 到 A_i ($i=1, 2, 3$), A_i 到 B_j ($j=1, 2$), B_j 到 C_k ($k=1, 2$), C_k 到 T 。记 $d(Y, X)$ 为城市 Y 到城市 X 之间的直接距离 (若两个城市没有道路直接相连, 则可以认为直接距离无穷大), 用 $L(X)$ 表示 S 到 X 的最优行驶路线的路长, 则有

$$L(S) = 0; \dots\dots\dots (12)$$

$$L(X) = \min_{Y \neq X} \{L(Y) + d(Y, X)\}, X \neq S \dots\dots\dots (13)$$

对于本例的具体问题, 可以直接计算如下:

$$\begin{aligned} L(A_1) &= 6, L(A_2) = 3, L(A_3) = 3; \\ L(B_1) &= \min\{L(A_1) + 6, L(A_2) + 8, L(A_3) + 7\} = 10 = L(A_3) + 7, \\ L(B_2) &= \min\{L(A_1) + 5, L(A_2) + 6, L(A_3) + 4\} = 7 = L(A_3) + 4; \\ L(C_1) &= \min\{L(B_1) + 6, L(B_2) + 8\} = 15 = L(B_2) + 8, \\ L(C_2) &= \min\{L(B_1) + 7, L(B_2) + 9\} = 16 = L(B_2) + 9, \\ L(T) &= \min\{L(C_1) + 5, L(C_2) + 6\} = 20 = L(C_1) + 5. \end{aligned}$$

所以, 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 S 到 T 的

最优行驶路线的路线为 $S \rightarrow A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow T$.

上面这种计算方法在数学上称为动态规划 (dynamic programming), 是最优化的一个分支。作为一个例子, 我们用 LINGO 来解决最短路问题, 我们可以编写如图 3-22 的 LINGO 程序。集合段定义的 "CITIES" (城市) 是一个基本集合 (元素通过枚举给出), L 是起对应的属性变量 (我们要求最短路长); "ROADS" (道路) 是由 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由道路相连, 所以不应该把它定义成稠密集合, 我们进一步将其元素通过枚举给出, 这就是一个稀疏集合, D 是稀疏集合 ROADS 对应的属性变量 (给定的距离)。

```

model:
SETS:
  CITIES /S,A1,A2,A3,B1,B2,C1,C2,T/; L;      !属性L(i)表示城市S到城市i的最优行驶路线的路长;
  ROADS(CITIES, CITIES)/                     !派生集合ROADS表示的是网络中的道路(弧);
    S,A1 S,A2 S,A3                           !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;
    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;                           !属性D(i,j)是城市i到j的直接距离(已知);
ENDSETS
DATA:
  D =
    6 3 3
    6 5 8 6 7 4
    6 7 8 9
    5 6;
  L = 0, , , , , , , ;                       !因为L(S)=0;
ENDDATA
@FOR( CITIES( i) | #GT# @index(S):           !这行中"@index(S)"可以直接写成"1";
  L( i) = @MIN( ROADS( j, i): L( j) + D( j, i); ) ; !这就是前面写出的最短路关系式;
end
  
```

图 3-22 最短路问题的模

从模型我们还可以看出: 这个 LINGO 程序可以没有目标函数, 这在 LINGO 中是允许的, 可以用来找可行解 (解方程组和不等式组)。此外, 在数据段我们对 L 进行了赋值, 但只有 $L(S)=0$ 是已知的, 所以后面的值为空 (但位置必须留出来, 即用逗号“, ”一贯也不能少, 否则回出错)。如果这个语句直接写成 “ $L=0;$ ” 语法上看是对的, 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。

从这个例子还可以看出, 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)。我们在 @FOR 循环中的过滤条件里故意用了一个函数: “@ (index)”, 其作用是返回一个元素在集合中是索引值, 这里 @ (index) = 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)” 可以直接等价地写成 “i#GT#1” 也是可以的。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。

运行以上程序得到结果 (图 3-23)。可以看出, 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 S 到 T 的最优行驶路线的路线为 $S \rightarrow A_3 \rightarrow B_2 \rightarrow C_1 \rightarrow T$ 。)

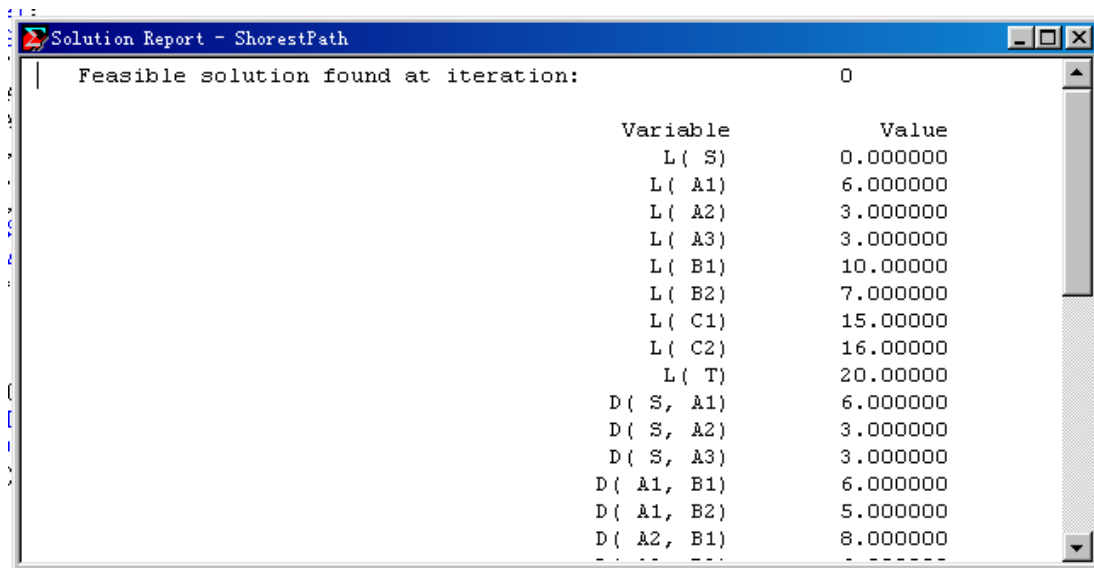


图 3-23 最短路问题的结果

上面这个例子中定义稀疏集合 **ROADS** 的方法是将其元素通过枚举给出,有时候如果元素比较多,这还是太麻烦了,用起来不方便。LINGO 提供了另一种定义稀疏集合的方法,这就是“元素过滤法”,能够从构成派生集合的父集合的笛卡儿积中的系统地过滤下来的一写真正需要的元素。请看下面的例子。

例 3.6 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。假设这 8 名同学两两之间组队的效率如表 3-5 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?

这是一个典型的匹配(matching)问题,把表 3-5 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 $MATCH(S_i, S_j) = 1$ 表示同学 S_i, S_j 组成一队,而 $MATCH(S_i, S_j) = 0$ 表示同学 S_i, S_j 不组队。由于对称性,只需考虑 $i < j$ 共 28 个 0-1 变量。显然,摸表函数正好是 $BENEFIT(S_i, S_j) \times MATCH(S_i, S_j)$ 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 i 有:只要属性 $MATCH$ 的某下标为 i 就加起来,此和应该等于 1。因此,完整的数学模型如下(显然,这是 0-1 线性规划):

$$\min \sum_{I < J} \{BENEFIT(I, J) \times MATCH(I, J)\}; \dots \dots \dots (14)$$

$$s.t. \sum_{J=I \text{ 或 } K=J} \{MATCH(J, K)\} = 1, I = 1, 2, 3, 4, \dots \dots \dots (15)$$

$$MATCH(J, K) \in \{0, 1\} \dots \dots \dots (16)$$

表 3-5 同学两两之间组队的效率

| 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 |

模型输入 LINGO 见图 3-24, 其中 STUDENTS 集合的元素列表“S1..S8”等价于写成“S1 S2 S3 S4 S5 S6 S7 S8”, 它没有相关属性列表, 只用于表示一个下标集合。我们应该特别注意, 在挨批声集合“PAIRS”的定义中, 增加了过滤条件, 即逻辑关系式“&2#GT#&1”, 意思是第 2 个父集合的索引值 (用“&2”表示) 大于第 1 个父集合的元素的索引值 (用“&1”表示)。这样, “PAIRS”中的元素就正好对应上面表 3-5 中是严格上三角部分的二维下标 (共 28 个元素)。BENEFIT 和 MATCH 都是这个集合 PAIRS 的属性。

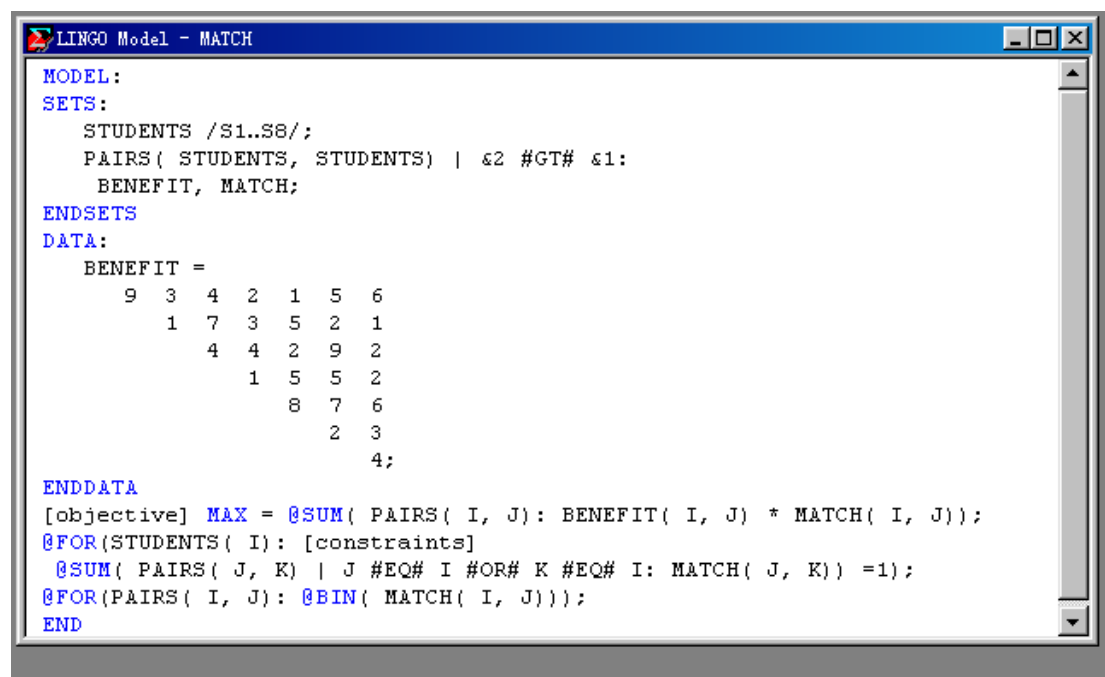


图 3-24 稀疏集合的例子

读者还应注意数据段对 BENEFIT 的赋值方式, 体会我们前面提到的“LINGO 是按照列的顺序对属性变量的元素进行赋值的”, 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方式, 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#”的, 所以这个逻辑式中没有必要使用括号指定运算次序。LINGO 中的运算符及其优先级关系可以参见 3.3.1 节。

选择菜单命令“LINGO|Solve”运行这个程序, 可以得到全局最优值为 30。由于 MATCH 变量中多数为 0 我们这里练习一下如何更清晰地浏览最优解, 选择菜单命令“LINGO|Solution”, 可以看到图 3-25 所示的对话框, 在对话框中的“Attribute or Row Name (属性或行名)”里选择“Text (文本格式)”, 并选择“Nonzeros Only 只显示非零值”选项, 然后单击“OK”按钮, 得到的正是我们想看的关于最优解的非零分量的报告 (如图 3-26 所示)。学生最佳的组队方式是: (1, 8), (2, 4), (3, 7), (5, 6)。

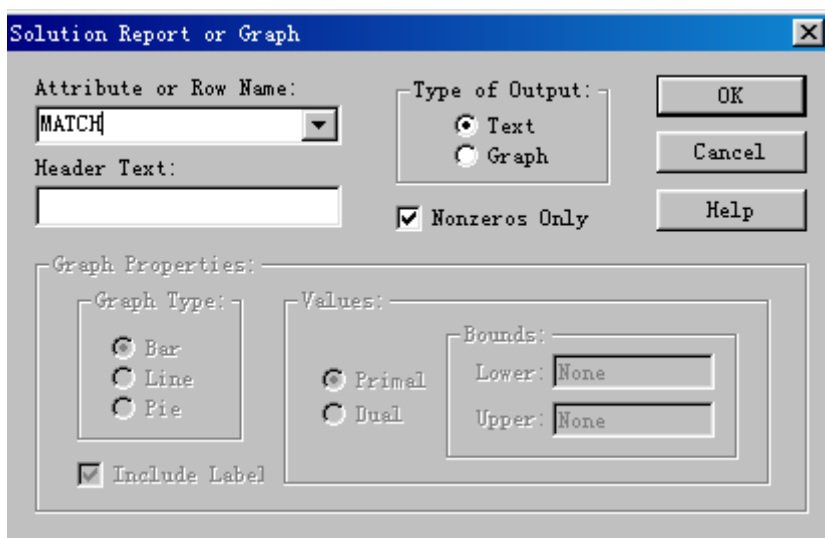


图 3-25 解答报告对话框

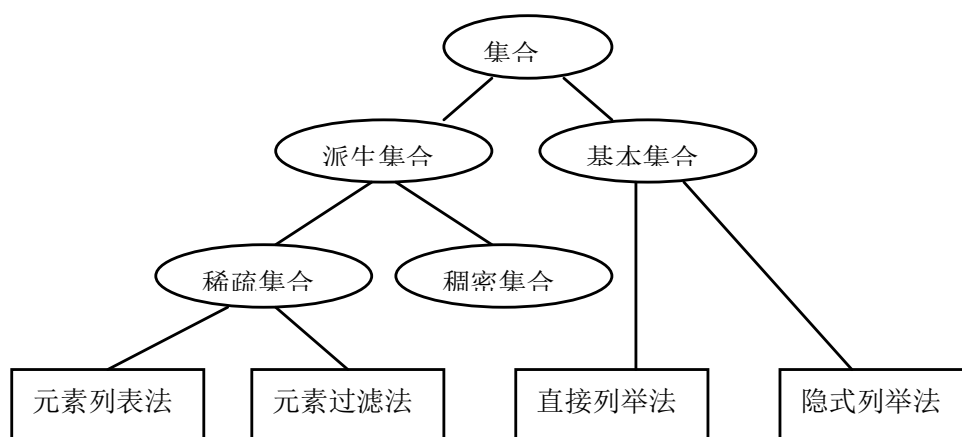
| Variable | Value | Reduced Cost |
|----------------|----------|--------------|
| MATCH(S1, S8) | 1.000000 | -6.000000 |
| MATCH(S2, S4) | 1.000000 | -7.000000 |
| MATCH(S3, S7) | 1.000000 | -9.000000 |
| MATCH(S5, S6) | 1.000000 | -8.000000 |

图 3-26 例 3.6 的解答结果

3.2.4 集合的使用小结

下面把 前面介绍的不同类型及其小结以下，表示在图 3-27 中

选择，我们归纳一下基本集合和派生集合的定义语法。基本集合的定义格式为（以下语法中凡是在方括号“[]”中的内容，表示是可



选的项，即该项可以有也可以没有）：

图 3-27 集合的的不同类型及其关系

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，元素列表利用采用显示列举法。隐式列举法可以有几种不同格式，见表 3-6。

| 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... | OCT2001... | OCT2001,NOV2001, |
| | monthYearN | JAN2002 | DEC2001,JAN2002 |

上面的语法还告诉我们元素列表 和属性列表都是可选的。当属性列表不在集合定义中出现时，这样的集合往往只是为了将来在程序中作为一个循环变量了来使用，或者作为构造更复杂的派生集合的负集使用（参见前面例 3.6（匹配问题）中的集合 **STUDENTS**，段以赋值语句的方式直接给出元素列表。例如，3.2.1 节的模型（图 3-14）的集合段和数据段可以分别改为

SETS:

QUARTERS: DEM, RP, INV;

ENDSETS

DATA:

QUARTERS DEM=1 40 2 60 3 75 4 25;

！注意 LINGO 按列赋值的特点；

ENDDATA

派生集合的一般定义格式为；

setname (parent-set-list) [/member-list][:attribute-list];

其中与基本集合的定义相比较只是多了一个 **parent-set-list**（父集合列表）。父集合列表中的集合（如 **set1, set2, ...**）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**）不在集合项和中出现时，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，则认为定义的时刻密集集合，即父集合中所有元素的有序组合（笛卡儿积）都是 **setname** 的元素（利用过滤条件）两种不同方式，请参看前面的介绍。

3.3 运算符和函数

3.3.1 运算符及其优先级

在前面的很多例子里，我们陆续用到了一些运算符，现在归纳一下 LINGO 中的三运算符：算术运算符，逻辑运算符和关系运算符。

算术运算符实际上就是加、减、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）。

逻辑运算符就是结果只有“真(TRUE)”和“假(FALSE)”两个值(称为“逻辑值”)的运

算,LINGO 中用数学 1 代表 TRUE,其它值(典型的值是 0)都是 FALSE.Z 在 LINGO 中,逻辑运算(表达式)通常作为过滤条件使用(回顾一下,在例 3.3 中定义约束时\例 3.6 中定义稀疏集合约束时,都多次使用了逻辑表达式作为过滤条件使用)LINGO 中的辑运算符有 9 种,可以分为两类:

(1)#AND#(与),#NOD#(非):这 3 个运算是逻辑值之间的运算,也就是它们操作的对象本身必须已经是逻辑值或逻辑表达式..计算结果也是逻辑值,'

(2)#EQ#(等于),#NE#(不等于),#GT#(大于),#GE#(大于等于),#LT#(小于),#LE#(小于等于):这 6 个操作实际上是"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.

而逻辑表达式计算得到的结果是逻辑值
关系运算符表示的是"数与数之间"的大小关系.因此在 LINGO 中用来表示优化模型的约束条件,所以可以认为不是真正的操作运算符.LINGO 中关系运算符有三种:

<(即<=,小于等于),=(等于),>(即>=,大于等于)
请注意在优化模型中约束一般,,没有严格小于,严格大于关系.此外,请注意区分运算符与"数与数之间"进行比较的 6 个逻辑运算符的不同之处.

这些运算符的优先级如表 3-7 所示(同以优先级按从左到右的顺序执行:如果有括号"),则括号;诶的表达式优先计算).

表 3-7 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 | < = > |

3.3.2 基本的数学函数

在 LINGO 中建立优化模型时引用大量的内部函数这些函数以"@ "负号打头.
LINGO 中包括相当丰富的数学函数,这些函数的用法非常简单,我们直接在下一一列出
@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 为自然对数值的底,即 2.718281...)
@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 的伽马(Gamma)函数的自然对数值(当 X 为整数时 $LGM(X)=LOG(X-1)!$;
当 X 不为整数时,采用线性插值得到结果)
@LOG(X):自然对数函数,返回自然对数值
@MOD(X,Y): 模函数,返回 X 对 Y 的取模的结果,即 X 除以 Y 的余数,这里 X 余 Y 应是整数
@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 是返回-1,X>=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,返回一列数(list)的:最大值

@SMIN.(list)::最小值函数,返回一列数(list)的:最小值

@SQR(X):平方函数,返回 X 的 平方即(X*X)的值

@SQRT(X):平方根函数,返回 X 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)。

3.3.3 集合循环函数

集合循环函数是指对集合上的元素(下标)进行循环操作的函数,如前我们用过的@FOR和@SUM等,一般用法如下:

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

其中:

function 是集合函数名, 是 FOR,MAX,PROD,SUM 五种之一;

setname 是集合名;

set_index_list 是集合索引列表(不需使用索引时可以省略);

condition 是逻辑表达式描述的过滤条件(通常含有索引,无条件时可省略);

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

五个集合函数名的含义如下:

@FOR(集合元素循环函数):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。

3.3.4 集合操作函数

集合操作函数是指对集合进行操作函数,主要有@IN,@INDEX,@WRAP,@SIZE四种,下面分别简要介绍一般用法。

• @INDEX[set_name,[primitive_set_element])

这个函数给出元素 primitive_set_element 在集合 set_name 的索引值(即按定义集合时元素出现顺序的位置编号)。如果省略集合名 set_name, LINGO 按模型在定义的集合顺序找到第一个含有元素 primitive_set_element 的集合,并返回索引值。如果在所有集合中均没找到该元素,会给出出错信息。

请注意,按照上面所说的索引值的含义,集合 set_name 的一个索引值是一个正整数(即对集合中一个对应元素的编号),且只能位于 1 和集合的元素个数(即@SIZE(set_name),该函数是含义见后面)之间,超出这个范围就没意义了。

例如,假如定义一个女孩姓名的集合(GIRLS)和一个男孩姓名的集合(BOYS)如下:
SETS:

GIRLS/DEBBIE,SUE,ALLCE/;

BOYS/BOB,JOE,SUM,FRED/;

ENDSETS

可以看到女孩和男孩中都有名为 SUM 的小孩。这时,调用函数@INDEX(BOYS,SUE)将返回索引值 2,这相当于@INDEX(GRILS,SUE),因为集合 GRILS 的定义出现在集合 BOYS 之前。如果真的要找男孩中名为 SUE 的小孩的索引,应该使用@INDEX(GRILS,SUE),这时将返回索引 3。

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

这个函数用于判断一个集合中是否有某个索引值。如果集合 `set_name` 中包含由索引 `primitive_index_1[,primitive_index_2...]` 所表示的对应元素，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”表示对应第 1 个父集合的元素的索引值，“&2”表示对应第 2 个父集合的元素的索引值。

例如，如果我们想定义一个学生集合 STUDENT（基本集合），然后由它派一个及格学生的集合 PASSED 和一个不及格学生集合 FAILED，可以如下定义：

SETS:

STUDENTS/ZHAO, QIAN. SUN, LI/;

PASSED (STUDENTS) /QIAN, SUN/;

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

ENDSETS

又如，如果集合 C 由集合 A, B 派生的，例如：

SETS:

A/1...3/;

B/X Y Z/;

C (A, B) /1, X1, Z2, Y3, X/;

ENDSETS

现在假设我们判断 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，C 中是否包含元素 (2, Y)，所以上面语句的结果是 X=1（真）。你可能已经注意到，这里 X 既是集合 B 的元素，后来又对 X 赋值 1，这样不混淆吗？这两个 X 表示的是一个东西吗（后者回冲掉前者吗）？在 LINGO 中这种表达式是允许的，因为前者的 X 是集合元素，后者 X 是变量，二者逻辑上没有任何关系（除了同名外），所以不回出现混淆，更谈不上后者回冲掉前者的问题。

- @WRAP (I, N)

当 I 为于区间 [I, 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 为函数。可见这个函数相当于数学上用 I 对 N 取模函数的值+1，即 $@WRAP(I, N) = @MOD(I, N) + 1$ 。此函数对 N < 1 无定义。

可以想到，此函数的目的之一集合 S 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。

3.3.5 变量定界函数

变量定界函数对变量的取值范围附加限制，共有以下四种函数：

@BND(L,X,U)：限制 $L \leq X \leq U$ 。注意 LINGO 中命令 SLB, SUB 类似函数@SLB 和 @SUM。

@BIN(X)：即限制 X 为 0 或 1。注意 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 后正数）。

@GIN(X)：限制 X 为整数。

3.3.6 财务会计函数

财务会计函数主要用于计算净现值，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，从下个时段开始连续 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

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

• @FPL(I,N)

返回如下情形下的净现值：单位时段利率为 I ，从下个时段开始连续 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

$$@FPL(I, N) = (\frac{1}{1+I})^N。$$

3.3.7 概率论中的相关函数

这里我们只是列出这些函数的简要功能，由于牵涉较多概率论和随机过程个概念，请大家参阅有关概率论和随机过程的书籍。2332

@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，即均值为 A 的 Poisson 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 A 的 Poisson 随机变量。

@PBN(P, N, X)：二项分布函数，即返回参数 (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，采用线性插值计算）。

@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 个服务器且不允许无穷排队时 Erlang 繁忙概率。

@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

Erlang 繁忙概率。

@PFS(A, X, C): 当负荷上限为 A, 顾客数为 C, 并行服务器数量有 X 个时, 有限源的 Poisson 服务系统等待或返修顾客的期望值。(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, 采用线性插值计算。

@PFD(A, X, C): 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 χ^2 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, 默认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)”的读者请进一步参阅 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)。

3.3.8 文件输入输出函数

这些函数用于控制通过文件输入数据和输出结果, 包括以下五个函数。

- @FILE (filename)

当前模型引用其他 ASCII 码文件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 (可以带有文件路径, 没有指定路径时表示在当前目录), 该文件记录之间用 “~” 分开。我们将在第四章详细介绍。

- @Odbc

这个函数提供 LINGO 与 ODBC (open data base connection, 开放式数据库连接) 的接口, 需要详细了解的读者请进一步阅读 LINGO 的使用手册, 或参与 @OLE 函数的用法。

- @OLE

这个函数提供 LINGO 与 ODBC (open data base connection, 对象连接与嵌入) 的接口, 我们将在第四章详细介绍。

- @POINTER (N)

在 Windows 下使用 LINGO 的动态连接库 (dynamic link library DLL), 直接从共享的内存中传给数据。需要详细了解的读者请进一步参阅 LINGO 的使用手册。

- @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下生成这个文件 (如果这个文件已经存在, 将会被覆盖)。具体用法我们将在第 4 章详细介绍。

3.3.9 结果报告函数

这些函数用于输出计算结果和与之相关的一些其他结果, 以及控制输出格式等。

- @ITERS ()

这个函数只能在程序的数据段使用, 调用时不需要任何参数, 总是返回 LINGO 求解器计算所使用的总迭代次数。例如:

- @TEXT () = @ITERT();

将迭代次数显示在屏幕上。

。@NEWLINE (n)

这个函数在输出设备上输出 n 行新行 (n 为一个正整数)。

。@STRLEN (string)

这个函数返回字符串 “string” 的长度，如@STRLEN(123)返回值为 3。

。@NAME (Var _ or _ row _ reference)

这个函数返回变量名或行名。请看下面的例子：

SETS:

WH/WH1。。WH3/; ! WH 表示仓库的集合;

C/C1。。C4/; ! C 表示顾客的集合;

RODA (WH, C): X; ! RODA 表示仓库到顾客的道路集合;

! X 是表示某个仓库对某个顾客的供货数量;

ENDSETS

DATA:

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

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

ENDDATA

输出结果示意如下 (之所以说只是 “示意”，是因为我们这里没有详细交代属性 X 当前的取值)：

| | |
|------------|----|
| x (WH1,C1) | 2 |
| x (WH1,C2) | 17 |
| x (WH1,C3) | 1 |
| x (WH2,C1) | 13 |
| x (WH2,C2) | 12 |
| x (WH2,C3) | 21 |

注意 从上面的例子还可以看出，“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，可以想象，约束名也是指模型展开后的约束名 (用 LINGO| Generate 命令可以不清楚第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。

。@WRITE (obj1 [, . . . , objn])

这个函数只能在数据段中使用，用于输出一系列结果 (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，则上面语句的作用是在屏幕上输出 A, B 以及 A/B 的值 (注意上面语句中还增加了一些字符串，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

。@WRITEFOR (set_index_list)

[| condition]: obj1 [, . . . , objn])

这个函数可以看作是函数@WRITE 在循环情况下的推广，它输出集合上定义的属性对

应的多个变量的取值（因此它实际上也是一个集合循环函数）。例如（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 wh1 到顾客 c1 供货 2 件
从仓 wh1 到顾客 c2 供货 17 件
从仓 wh1 到顾客 c3 供货 1 件
从仓 wh2 到顾客 c1 供货 13 件
从仓 wh2 到顾客 c2 供货 12 件
从仓 wh2 到顾客 c3 供货 21 件
。符号 “*”
```

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 和这个字符串之间，表示将这个字符串重复 n 次，例如：

```
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（上面的集合 DAY 表示一个星期 7 天的集合，对应的属性 ON_DUTY 表示每天的上班人数，正如前面介绍的员工聘用的例子中一样）：

```
上班人数图示
MON 20 ++++++
TUE 16 ++++++
WED 12 ++++++
THU 16 ++++++
FRI 19 ++++++
SAT 14 ++++++
SUN 13 ++++++
```

。@FORMAT (value, format _descriptor)

在@WRITE 和@WRITEFOR 函数中，可以使用@FORMAT 函数对数值设定输出格式。其中 value 表示要输出的数值，而 format _descriptor(格式描述符)表示输出格式。格式描述符的含义于 C 语言中的格式描述是类似的，如 “12. 2f” 表示输出一个十进制数，总共占 12 位，其中有 2 位小数。

注意 使用@FORMAT 函数将把数值转换成字符串，所以输出的实际是字符串，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)将返回解答中变量 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）。

例如，一个程序的数据段中可以有如下语句：

DATA:

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

ENDDATA

。@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增加的量（参见 2.2 节敏感性分析中的 allowable increase）。

。@STATUS ()

返回 LINGO 求解模型结束的最后状态：

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)

3.3.10 其他函数

。@IF (logical _condition, true _result, false _result)

当逻辑表达式 logical _condition 的结果为真时，返回 true _result, 否则返回 false _result。

例如@(x # LT # 100,20,15)语句，当 X <100 时，返回 20，否则返回 15。

。@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)

该函数是允许用户自己编写的函数(dl 或 Obj 文件),该用户函数可能当用 C 或 FORTRAN 等其他语言编写并编译,返回值为用户函数计算的结果.从编程角度来看,@USER 函数包含两个参数: 第一个用于指定参数个数, 第二个用于指定参数向量: 而在 LINGO 中调用 •USER 时则直接指定对应的参数“user _determined_”(类似于 C 语言中的 main (argc , argv) 的编程和运行方式).更多细节请参考 LINGO 使用手册。

3.4 LINGO 的主要菜单命令

从前面的各个窗口中我们已经看到,LINGO 有 5 个主菜单:

。File(文件)

。Edit(编辑)

。LINGO(LINGO 窗口)

。Window(窗口)

。Help(帮助)

与第 2 章介绍的 LINGO 主菜单比较,LINGO 软件中的“LINGO”菜单相当于合并了 LINGO 中的 Solve(求解)菜单和 REPORTS(报告)菜单.这些菜单的用法都是和 Window 下其他应用程序的标准用法类似的,下面只对前 3 个主菜单中与 LINGO 不同而有一定 LINGO 特色的主要命令进行简要介绍.图 3—28 给出了工具栏的简要功能说明。

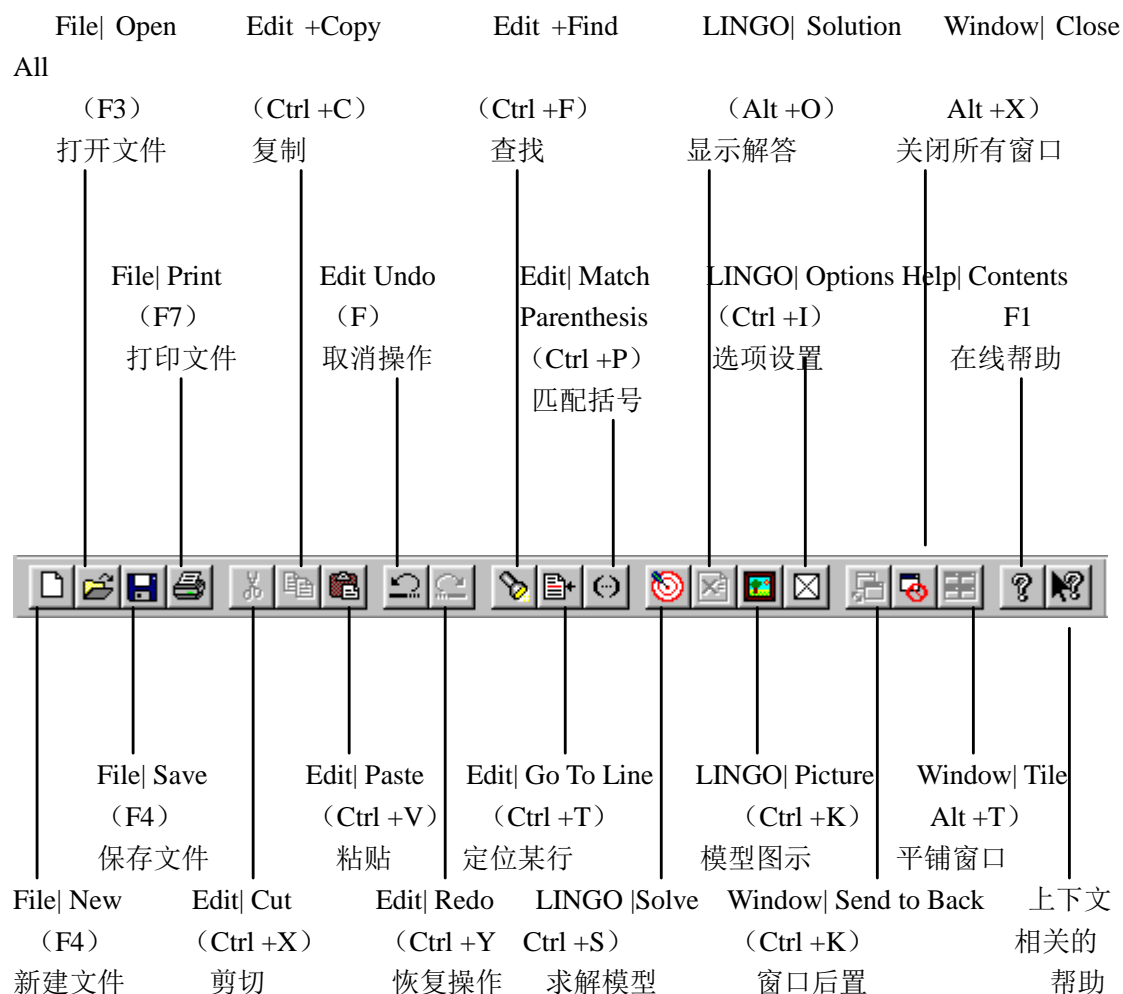


图 3-28 LINGO 工具栏及其对应的菜单命令和快捷键

3.4.1 文件主菜单

。 File| Export File....

该命令将优化模型输出到文件,它有两个子菜单,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(参见第 2 章的附录).
MPI Format(MPI 格式): 是 LINGO 公司制定的一种数学规划文件格式。

。 File| User Database Info

该命令弹出一个对话框,请求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()函数访问数据库时是要用到的。

3.4.2 编辑主菜单

。 Edit| Paste 和 Edit| Paste Special....

这两个命令都是将 Windows 剪贴板中的内容粘贴到当前光标处.不同之处在于:

(1) 前者是通常的“Edit| Paste(粘贴命令)”,它仅用于剪贴板中的内容是文本的情形。

(2) “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,如可以插入其他应用程序中生成的对象(object)或对象的链接(link)。例如,编译时将代码与数据分离是一种很好的习惯,所以 LINGO 模型中可能会在数据段用到从其他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...”是很方便的。

。 Edit| Match Parenthesis

该命令用于匹配模型中的括号:

(1)如果当前没有选定括号,则执行这个命令的结果是把光标移动到当前光标最近的一个

括号并选择这个括号。

(2)当选定一个括号后,执行这个命令的结果是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

该命令还有下一记子菜单和下下级子菜单,用于按函数类型选择 LINGO 的某个函数(参见 3.3 节),粘贴到当前光标处。

。Edit| Select Font

先用鼠标选择一段文本,然后选择“Edit| Select Font(选择字体)”菜单命令,则出现一个对话框(如图 3—29),你可以通过这个对话框控制显示字体、字形、大小、颜色、效果等。注意,这些显示特性只有当文件保存为 LINGO 格式(*.lg4)的文件时才能保存下来,否则下次打开文件时将不会还原这次你修改的显示特性。此外,如果“按语法显示色彩”选项是有效的(参见菜单命令“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。



图 3-29 字体选择对话框

。Edit| Insert New Object

该命令插入其他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,所以二者有所不同。

。Edit| Links

在模型窗口中选择一个链接,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。

。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 Objects 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 to the object: 修改对象;

breaking the link to the object: 断开对象的链接;

3.4.3 LINGO 系统 (LINGO) 主菜单

该主菜单下的多数命令与 LINGO 下相同, 或者我们已经在前面结合例子具体地介绍过了。我们这里只介绍 LINGO| Look (模型显示)、LINGO| Generate (模型生成)、LINGO| Picture (模型图示) 和 LINGO| Options (选项) 命令。其中 LINGO| Options (选项) 命令包括的内容特别多, 一般情况下我们没有必要修改选项 (即采用默认值就可以了); 除非你确实需要改变选项, 并且你对将要修改的选项的含义非常清楚, 否则建议你尽量不要轻易改变选项的值。

。LINGO| Look 命令

LINGO| Look (Ctrl +L) 命令在模型窗口下才能使用, 功能是按照 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。这个命令将导致弹出一个对话框 (图形略), 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。(我们在后续章节的例子中, 有时显示给大家的程序就是这样的, 主要是为了方便对每行程序的功能进行解释和说明。)

LINGO| Generate 和 LINGO| Picture 命令

LINGO| Generate 和 LINGO| Picture (Ctrl +K) 命令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| Generate 命令的结果是以代数表达式的形式给出的, 按照是否在屏幕上显示结果的要求, 你可以选择 “Display Model (Ctrl +G)” 和 “Dot Display Model (Ctrl +Q)” 两个子菜单之一。在屏幕上不显示时, 运行该命令的目的可能仅仅是为了以后选择适当的求解程序使用。例如, 对于 3.2.2 节的选址问题 (参见图 3—18), LINGO| Generate 命令显示的结果如图 3—30 所示。注意: 在 LINGO8.0 以及更早以前的版本中, 如果有非线性变量项, 对应的非线性变量前的系数将以问号 (“?”) 显示。

LINGO| Picture 命令的结果是按照矩形形式给出的。例如, 对应 3.2.2 节的下选址问题 (参见 3—18), 该命令的结果如图 3—31。该显示中非线性项的系数以黑色显示为 “?”, 线性项的系数为正时显示为蓝色, 为负则显示为红色 (本例没有红色)。在这个图形上单击鼠标右键, 可以出现一个相关联的显示控制菜单 (如图中写有 “Zoom In” 的菜单所示), 这个菜单可以控制图形显示的内容的放大 (Zoom In)、缩小 (Zoom Out)、显示全部内容 (View All), 也可以控制该窗口是否显示行名 (Row Names)、变量名 (Var Names)、滚动条 (Scroll Bars)。

。LINGO| Options 命令

LINGO| Options (Ctrl +I) 命令将打开一个含有 7 个选项卡的窗口 (如图 3—32), 你可以通过它修改 LINGO 系统的各种控制参数和选项。修改完以后, 你如果单击 “应用 (A)”

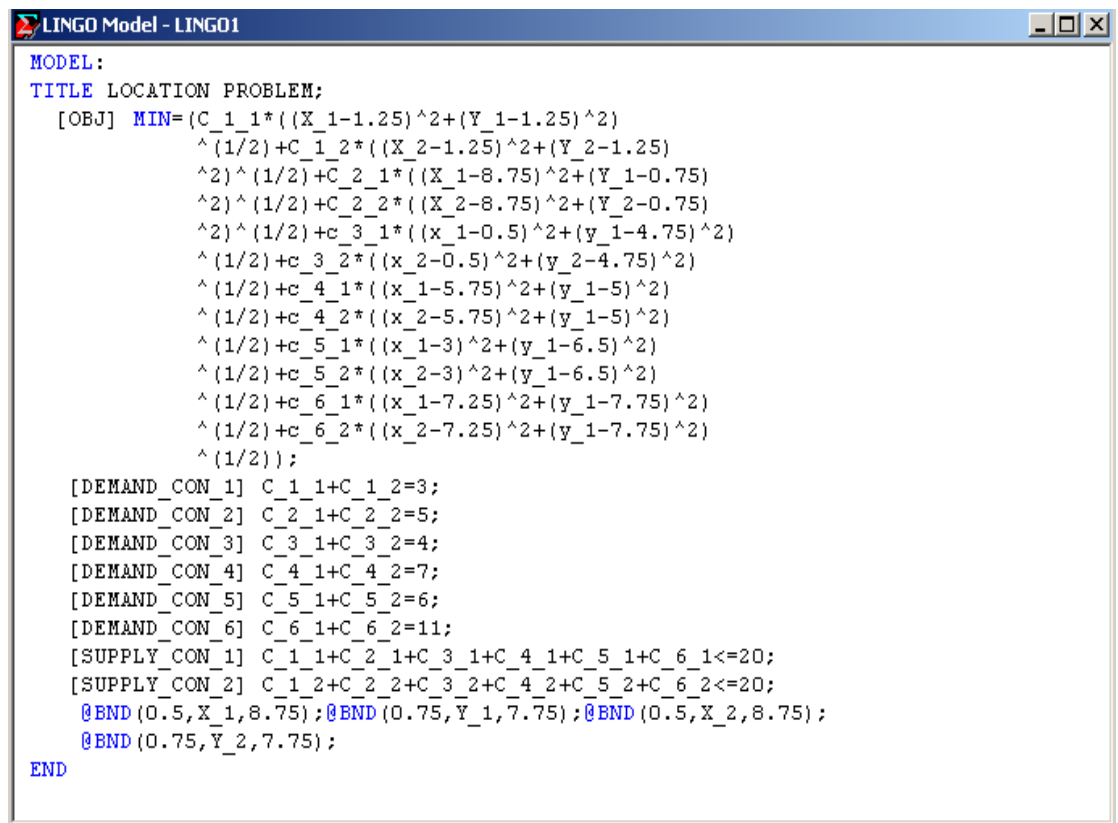


图 3-30 完整的 LINGO 模型

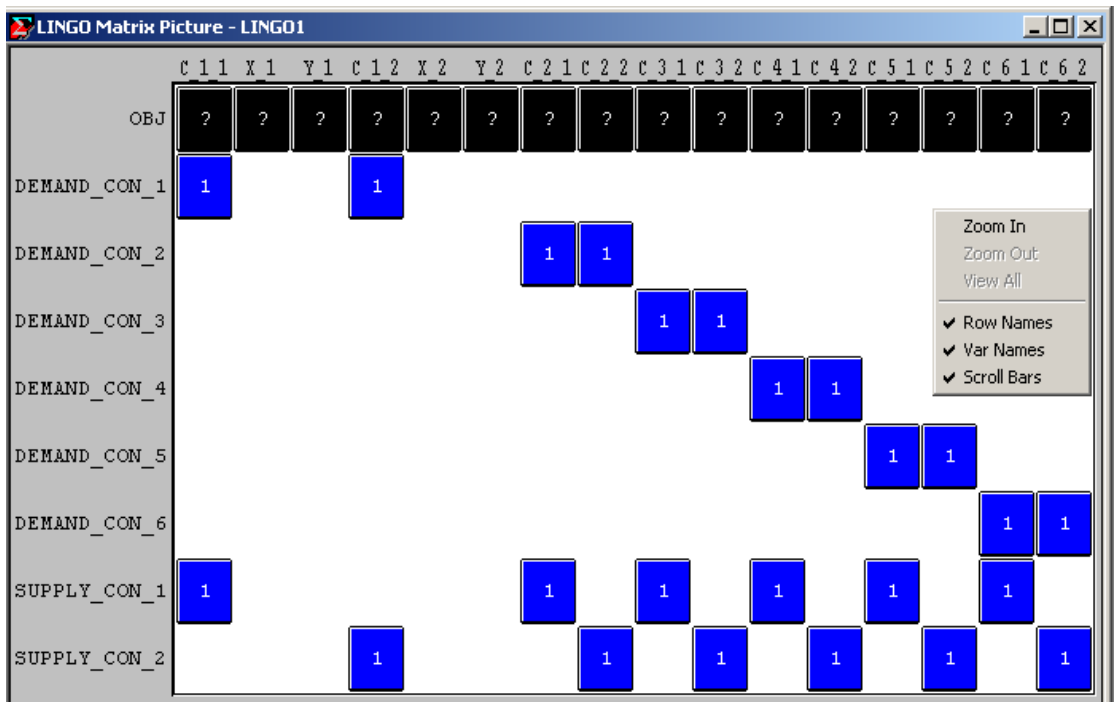


图 3-31 完整的 LINGO 图示

按钮，则新的设置马上生效；如果单击“OK（确定）”按钮，则新的设置马上生效，并且同时关闭该窗口，如果单击“Save”按钮，下次启动 LINGO 时这些设置仍然。单击“Default（默认值）”按钮，则恢复 LINGO 系统定义的原始默认设置。单击“Cancel（取消）”按钮将废弃本次操作，退出对话框；单击“Help（帮助）”按钮将显示对话框的帮助信息。

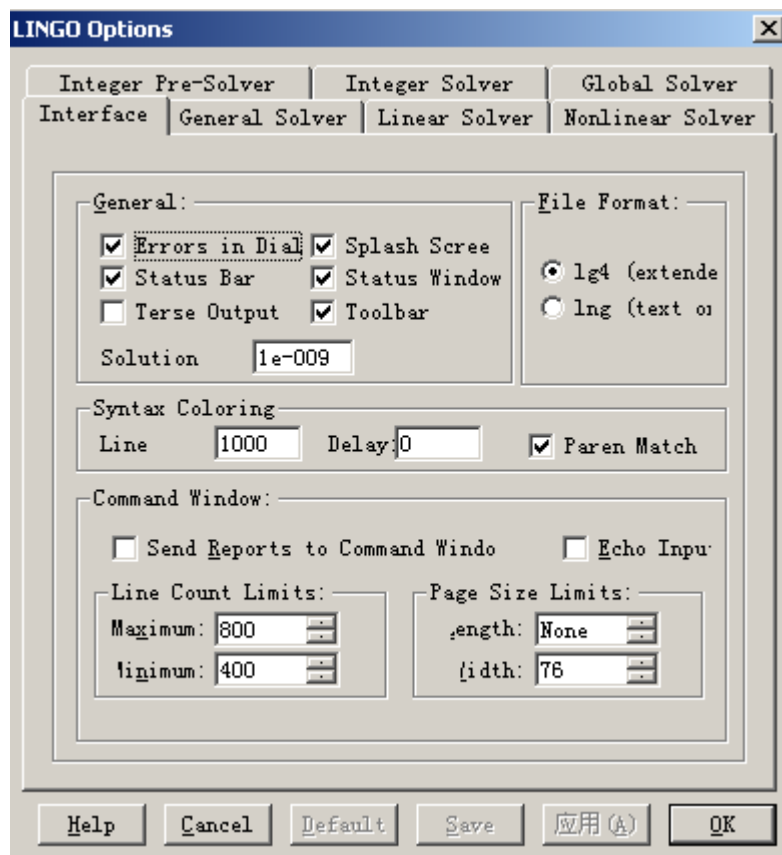


图 3-32 LINGO 选项卡

- LINGO| Options| Interface(界面)选项卡

图 3-32 显示的是界面选项卡，主要控制 LINGO 的界面、输出方式、文件格式等. 具体可以控制的参数和选项的含义见表 3-8.

3-8 LINGO 界面选项卡的控制的参数和含义

| 选项组 | 选项 | 含义 |
|-------------------|----------------------------|--|
| General (一般选项) | Errors In dialogs(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，你关闭该对话后程序才会继续执行；否则，错误信息将在报告框口，程序仍将进行 |
| | Splash screen (弹出屏幕) | 如果选择该选项，则 LINGO 每次启动时会在屏幕上一个对话框，显示 LINGO 的版本和版权信息；否则不弹出 |
| | Status Bar (状态栏) | 如果选择该选项，则 LINGO 系统在主框口最下面一行显示状态栏，否则不显示 |
| | Status Windows (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框口栏；否则不弹出 |
| | Terse Output(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 |
| | Toolbar (工具栏) | 如果选择该选项，则显示工具栏；否则不显示 |

| | | |
|---------------------------|---|--|
| General (一般选项) | Fill Out Ranges and Tables (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数据占用的空间, 是否对多余的表空间进行数据填充? (默认值为不进行填充) |
| | Solution Cutoff (截的截断) | 小于等于这个值的解的报告为 “0” (默认值为 10^{-9}) |
| File format (文件格式) | Lg4(extended) (LINGO 扩展格式) | 模型文件的默认值格式是 Lg4 格式 (这是一种二进制文件, 只有 LINGO 能读出) |
| | Lng(text only) (LINGO 纯文本格式) | 如果选择该选项, 模型文件的默认保存格式将变为 Lng 格式 (纯文本) |
| | Ltx(LINDO) (LINDO 纯文本格式) | 如果选择该选项, 模型文件的默认保存格式将变为 LINDO 格式 (纯文本) |
| Syntax coloring (语法配色) | Line limit (行数限制) | 语法配色的行数限制 (默认值为 1000)。LINGO 模型框口中将 LINGO 的关键词显示为蓝色, 注释为绿色, 其他为黑色, 超过该行数限制后则不再区分颜色。特别的, 设置行数为 0 时, 整个文件不再区分颜色 |
| | Delay (延迟) | 设置语法配色的延迟时间 (秒, 默认为 0, 从最后一次击键算起) |
| | Paren match (括号匹配) | 如果选择该选项, 则模型中当前光标所在处的括号及其相配的括号将以红色显示; 否则不使用该功能 |
| Command Windows (命令窗口) | Send reports to Command Windows (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 |
| | Echo Input (输入信息反馈) | 如果选择该选项, 则使用 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 |
| | Line count limits (行数限制) | 命令窗口每次显示的行数的最大值 Maximum (默认为 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 (默认为 400) |
| | Page size limit (页面大小控制) | 命令窗口每次显示的行数的最大值为 Length (默认为没有限制), 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 (默认为 74, 可以设定 64~200 之间), 多余的字符将被截断 |

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

该界面见图 3-33, 主要控制 LINGO 求解程序的一些通用参数, 具体参数和含义见表 3-9。

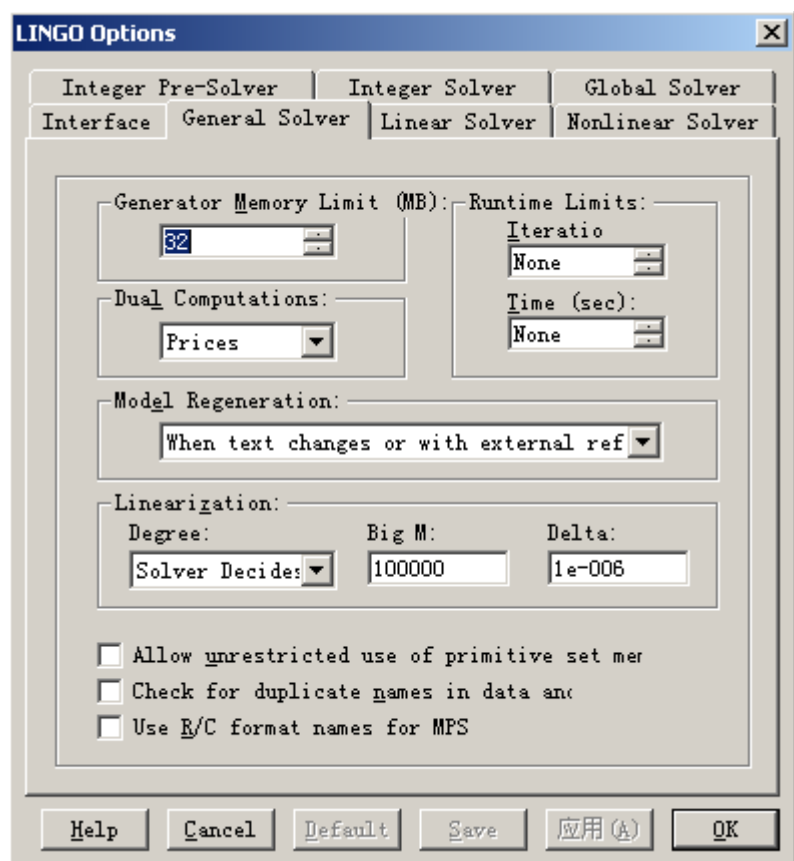


图 3-33 LINGO 通用求解程序的参数选项卡

表 3-9 LINGO 通用求解程序的选项卡的控制参数和含义

| 选项组 | 选项 | 含义 |
|---|--------------------|--|
| Generator memory limit(MB)矩阵生成器的内存限制（兆） | | 默认值为 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” |
| Runtime Limits 运行限制 | Iterations 迭代次数 | 求解一个模型时，允许的最大迭代次数（此时默认值为无限） |
| | Time(sec) 运行时间（秒） | 求解一个模型时，允许的最大运行时间（此时默认值为无限） |
| Dual computations （对偶计算） | | 求解时控制对偶计算的级别，有三种可能设置： <ul style="list-style-type: none"> • None：不计算任何对偶信息； • Prices：计算对偶价格（默认设置）； • Prices and Ranges：计算对偶价格并分析敏感性； • Prices Opt Only：只计算最优的对偶价格 |
| Fixed Var Reduction （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）： <ul style="list-style-type: none"> • 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 |

续表

| 选项组 | 选项 | 含义 |
|-----|----|----|
|-----|----|----|

| | | |
|---|---------------------------|--|
| Model Regeneration (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： <ul style="list-style-type: none"> • Only when text change: 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是（默认设置）； • Always: 每当有需要时； |
| Linearization (线性化) | Degree (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： <ul style="list-style-type: none"> • Solver Decides: 若变量小于等于 12 个，则尽可能全部线性化；否则不做任何线性化（默认设置）； • None: 不做任何线性化； • Low: 对函@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； |
| | Big M (线性化的大 M 系数) | 设置线性化的大 M 系数（默认值为 10^6 ） |
| | Delta (线性化的误差限) | 设置线性化的误差限（默认值为 10^{-6} ） |
| Allow Unrestricted Use of Primitive Set Member Names (允许无限制的使用基本集合的成员名) | | 选择该选项可以保持与 LINGO 4.0 以前版本兼容：即允许使用基本集合的成员名称直接作为该成员在校改集合的索引值（LINGO 4.0 以后的版本要求使用@INDEX 函数） |
| Check for Duplicate Names in Data and Model (检查数据和模型中的名称是否重复使用) | | 选择该项，LINGO 将检查数据和模型中的名称是否重复使用，如基本成员名是否与决策变量名重复 |
| Use R/C format names for MPS I/O (在 MPS 文件格式的输入输出中使用 R/C 格式的名称) | | 在 MPS 文件格式的输入输出中使用 R/C 格式的名称 |
| Minimize Memory Usage (最小化内存使用) | | 是否最小化内存使用量默认设置为‘是’.使用这个功能的一个缺点是可能引起计算速度的下降. |

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

界面见图 3-34，用于控制线性求解程序的相关参数.可以控制的参数和选项的含义见表 3-10.

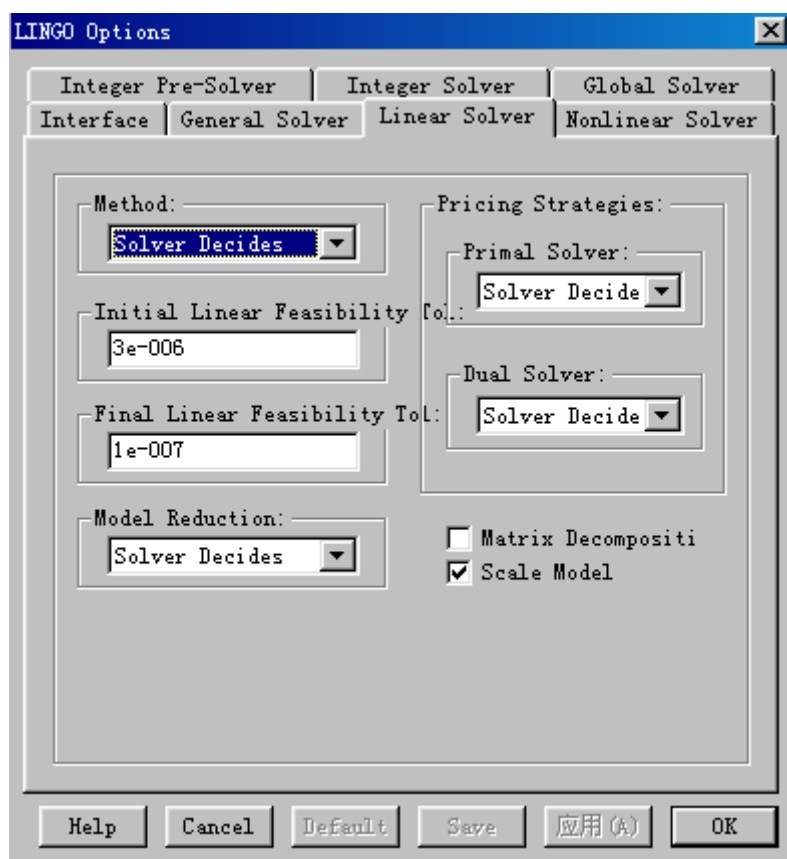


图 3-34 LINGO 线性求解程序的选项卡

表 3-10 LINGO 线性求解程序的选项卡的控制参数和含义

| 选项组 | 选项 | 含义 |
|---|--------------------------|---|
| Method 求解方法 | | 求解时的算法，有四种可能的设置： <ul style="list-style-type: none"> • Solver Decides: LINGO 自动选择算法（默认设置） • Primal Simplex: 原始单纯型算法 • Dual Simplex: 对偶单纯型算法 • Barrier: 障碍法（即内点法） |
| Model Reduction 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： <ul style="list-style-type: none"> • Off: 不检查 • On: 检查 • Solver Decides: LINGO 自动决定（默认值） |
| Initial Linear Feasibility Tol. 初始线性可行性误差限 | | 控制线性模型中的约束满足的初始误差限（默认值为） |
| Final Linear Feasibility Tol. 最后线性可行性误差限 | | 控制线性模型中的约束满足的最后误差限（默认值为） |
| Pricing Strategies 价 格策略（决 定出基的策 略） | Pricing Solver 原始单纯型法 | 有三种可能的设置： <ul style="list-style-type: none"> • Solver Decides: LINGO 自动决定（默认设置） • 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，找到是目标值下降最多的出基变量 |
| 续表 | | |
| 选项组 | 选项 | 含义 |

| | | |
|---|----------------------------|---|
| Pricing Strategies 价 格策略（决 定出基变量 的策略） | Dual Solver 对偶型单纯形 法 | 有三种可能的设置： <ul style="list-style-type: none"> • Solver decides: LINGO 自动决定（默认设置） • 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 |
| Debug Solver 调试 时采用的求 解程序 | Cold Start 冷启动时 | 设置 LINGO Debug 命令调试所使用的程序（冷启动的含义就是不从当前基开始，而是从头开始运行）。有四种可能的设置： <ul style="list-style-type: none"> • Solver Decides: LINGO 自动选择算法（默认设置） |
| | Warm Start 热启动时 | 同上（热启动的含义是从当前基开始运行） |
| Matrix Decomposition 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 |
| Scale Model 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

界面见图 3-35.可以控制的参数和选项的含义见表 3-11.

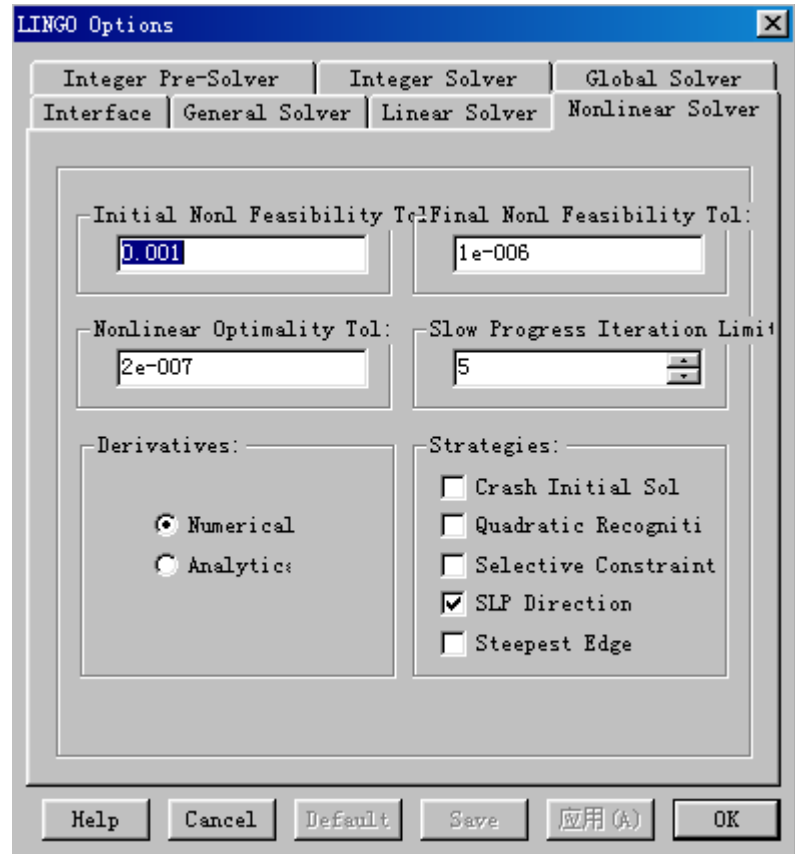


图 3-35 LINGO 非线性求解程序选项卡

表 3-11 LINGO 非线性求解程序选项卡的控制参数和含义

| 选项组 | 选项 | 含义 |
|-----|----|----|
|-----|----|----|

| | | |
|---|----------------------------------|--|
| Initial Nonlinear Feasibility Tol. 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（默认值为 10^{-3} ） |
| Initial Nonlinear Feasibility Tol. 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（默认值为 10^{-6} ） |
| Nonlinear Optimality Tol. 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后，停止迭代（默认值为 $2 * 10^{-7}$ ） |
| Slow Progress Iteration Limit 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后，停止迭代（默认值为 5） |
| Nonlinear Solver Version (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： <ul style="list-style-type: none"> • Solver Decides: LINGO 自动选择（默认设置） • Ver1.0: 选择 1.0 版本 • Ver2.0: 选择 2.0 版本 |
| Derivatives 导数计算方式 | | 设置导数计算方式，有五种选择： <ul style="list-style-type: none"> • Solver Decides: LINGO 自动选择（默认设置）； • Backward analytical: 后向解析法计算导数； • Forward analytical: 后向解析法计算导数； • Central differences: 中心差分法计算数值导数； • Forward differences: 向前差分法计算数值导数； |
| Strategies 策略 | Crash Initial Solution 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（默认值） |
| | Quadratic Recognition 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（默认值） |
| | Selective Constraint 有选择的检查约束 | 选择该选项，LINGO 在每次进行迭代是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，这样就会出现错误）；否则检查所有约束（默认值） |
| | SLP Directions 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（默认值） |
| | Steepest Edge 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；默认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

设置整数预处理程序的控制参数，界面见图 3-36.可以控制的参数和选项的含义见表 3-12.整数预处理程序只用于整数线性规划模型（ILP 模型），对连续规划和非线性模型无效。

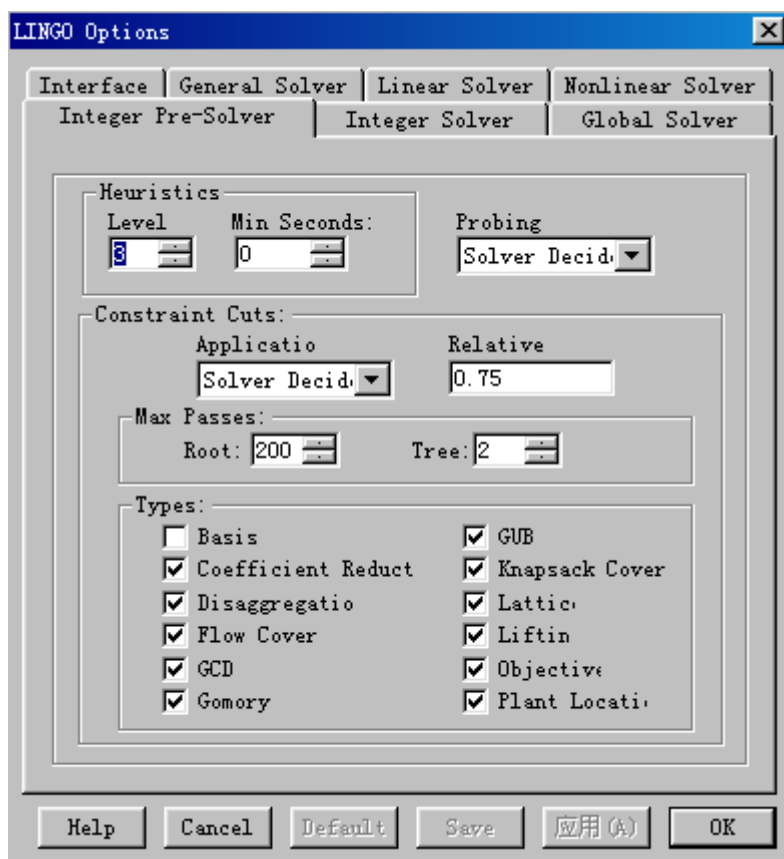


图 3-36 整数预处理程序的选项卡

表 3-12 整数预处理程序选项卡的控制参数和选项的含义

| 选项组 | 选项 | 含义 |
|-----------------------------|------------------------|---|
| Heuristics 启发式方法 | Level 水平 | 控制采用启发式搜索的次数（默认值为 3，可能的值为 0~100）。启发式方法的目的是从分支节点的连续解出发，搜索一个好的整数解 |
| | min Seconds 最小时间 | 每个分支节点使用启发式的最小时间（秒） |
| Probing Level 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先行规划模型，收紧变量的上下界和约束的右端项的值）。可能的取值为： <ul style="list-style-type: none"> • Solver decides: LINGO 自动选择（默认设置） • 1~7: 探测级别逐步升高 |
| Constraint Cuts 约束的割（平面） | Application 应用节点 | 控制在分支定界树中，哪些节点需要增加割（平面），可能的取值为： <ul style="list-style-type: none"> • 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 自动选择（默认设置） |
| | Relative Limit 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），默认值为 0.75 |
| | Max Passes 最大迭代检查 | 为了寻找合适的割，最大迭代检查的次数。有两个参数： <ul style="list-style-type: none"> • Root: 对根节点的次数（默认值为 200） |

| | | |
|--|-------------|---|
| | 的次数 | • Tree: 对其他节点的次数（默认值为 2） |
| | Types 类型 | 控制生成割（平面）的策略，共有 12 种策略可供选择。（如 果想了解细节，请参阅整数规划方面的专著） |

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

可以控制的参数和选项的含义见表 3-13，界面见图 3-37.

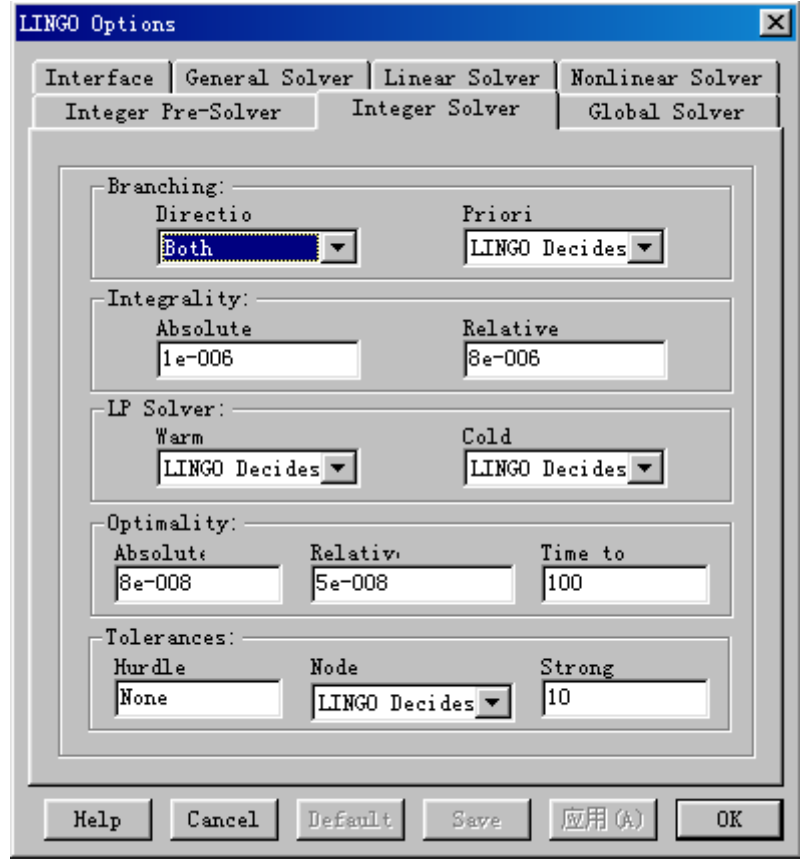


图 3-37 LINGO 整数求解程序的选项卡

表 3-13 LINGO 整数求解程序的选项卡的控制参数和含义

| 选项组 | 选项 | 含义 |
|-------------------|-------------------|--|
| Branching 分支 | Direction 方向 | 控制分支策略中优先对变量取整数的方向，有三种选择： <ul style="list-style-type: none"> • Both: LINGO 自动选择（默认设置） • Up: 向上取整优先 • Down: 向下取整优先 |
| | Priority 优先级 | 控制分支策略中优先对哪些变量进行分支，有两种选择： <ul style="list-style-type: none"> • LINGO Decides: LINGO 自动选择（默认设置） • Binary: 二进制（0-1）变量优先 |
| Integrality 整性 | Absolute 绝对 误差 | 但变量与整数的绝对误差小于这个数时，该变量被认为是整数.默认值为 10^{-6} |
| | Relative 相对误 差 | 但变量与整数的相对误差小于这个数时，该变量被认为是整数.默认值为 8×10^{-6} |
| LP Solver | Warm Start 热 | 当以面前的求解结果为基础，热启动求解程序时采用的算 |

| | | |
|---------|----|---|
| LP 求解程序 | 启动 | 法，有四种可能的设置： <ul style="list-style-type: none"> • LINGO Decides: LINGO 自动选择（默认设置） • Primal Simplex: 原始单纯形算法 • Dual Simplex: 对偶单纯形算法 • Barrier: 障碍法（即内点法） |
|---------|----|---|

续表

| 选项组 | 选项 | 含义 |
|----------------|----------------------------------|--|
| | Cold Start 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：（同上，略） |
| LP 求解程序 | Absolute 目标函数的绝对误差限 | 当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）。默认值为 8×10^{-8} |
| Optimality 最优性 | Relative 目标函数的相对误差限 | 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）。默认值为 5×10^{-8} |
| | Time To Relative 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，默认值为 100 秒 |
| Tolerances | Hurdle 篱笆值 | 同第二章 LINDO 部分的介绍 |
| | Node Selection 节点选择 | 控制如何选择节点的分支求解，有以下选项： <ul style="list-style-type: none"> • LINGO Decides: LINGO 自动选择（默认设置） • 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 |
| | Strong Branch 强分支的层数 | 控制采用强分支的层数，也就是说，对前这么多层的分支，采用强分支的策略。所谓强分支，就是在一个节点对多个变量分别尝试进行预分支，找出其中最好的解（变量）进行实际分支 |

• LINGO|OPTIONS|Global Solver（全局最优解程序）选项卡

控制全局优化程序的参数。可以控制的参数和选项的含义见表 3-14，界面如图 3-38。

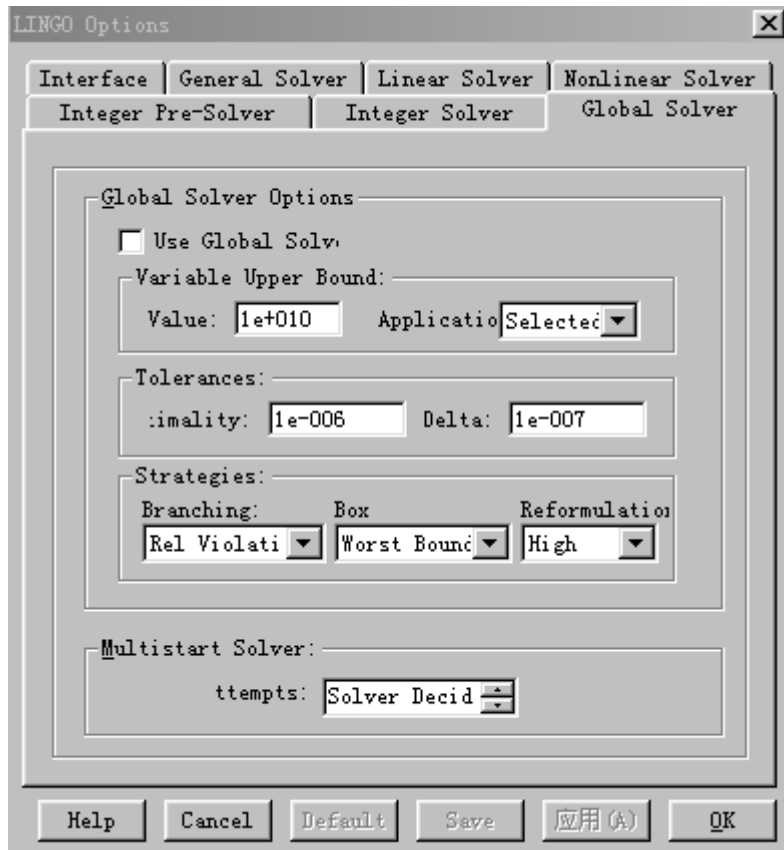


图 3-38 LINGO 全局优化程序的选项卡

表 3-14 LINGO 全局优化程序的选项卡的控制参数和含义

| 选项组 | 选项 | 含义 |
|------------------------|------------------------------|---|
| Global Solver 全局最优求解程序 | Use Global Solver 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 |
| | Variable Upper Bound 变量上界 | <p>有两个域可以控制变量上界（按绝对值）：</p> <ol style="list-style-type: none"> 1. Value: 设定变量的上界，默认为 10^{10}； 2. Application 列表框设置这个界的三种应用范围： <ul style="list-style-type: none"> • None: 所以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（默认设置） |
| | Tolerances 误差限 | <p>有两个域可以控制两类误差限（按绝对值）：</p> <ol style="list-style-type: none"> 1. Optimality: 只搜索比当前解至少改进这么个单位的解（默认值为 10^{-6}）； 2. Delta: 全局最优求解程序在凸化过程中增加的约束的误差限（默认值为 10^{-7}） |

| | | |
|----------------------------|---------------|---|
| | Strategies 策略 | <p>可以控制最优求解程序的三类策略：</p> <p>1. Branching: 第 1 次对变量分支时使用的分支策略：</p> <ul style="list-style-type: none"> • 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, 默认设置) <p>2. Box Selection: 选择活跃分支节点的方法：</p> <ul style="list-style-type: none"> • Depth First (深度优先) • Worst Bound (具有最坏界的分支优先, 默认设置) <p>3. Reformulation: 模型重整的级别：</p> <ul style="list-style-type: none"> • None (不进行重整) • Low (低) • Medium (中) • High (高, 默认设置) |
| Multistart Solver 多初始点求解程序 | Attempts 尝试次数 | <p>设定用多少个初始点尝试求解, 有以下几种可能的设置：</p> <ul style="list-style-type: none"> • Solver Decides: 由 LINGO 决定 (默认设置, 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 |

3.5 LINGO 命令窗口

与 LINGO 类似, LINGO 也有两种命令模式: 一种是常用的 Windows 模式, 另一种是命令行 (Command-Line) 模式。由于在 Windows 模式下使用 LINGO 非常方便, 所以这里仅仅是简单介绍一下命令行模式下的主要行命令, 供有兴趣的读者作为入门参考。

与 LINGO 类似, 你随时可以通过菜单命令 (“Window|Command Window(Ctrl+1)”) 打开命令窗口, 在命令窗口下操作。LINGO 行命令大多数与 LINDO 类似, 如在命令窗口下的提示符 “:” 后面输入 COMMANDS (COM) 可以看到 LINGO 的所有命令 (图 3-39)。

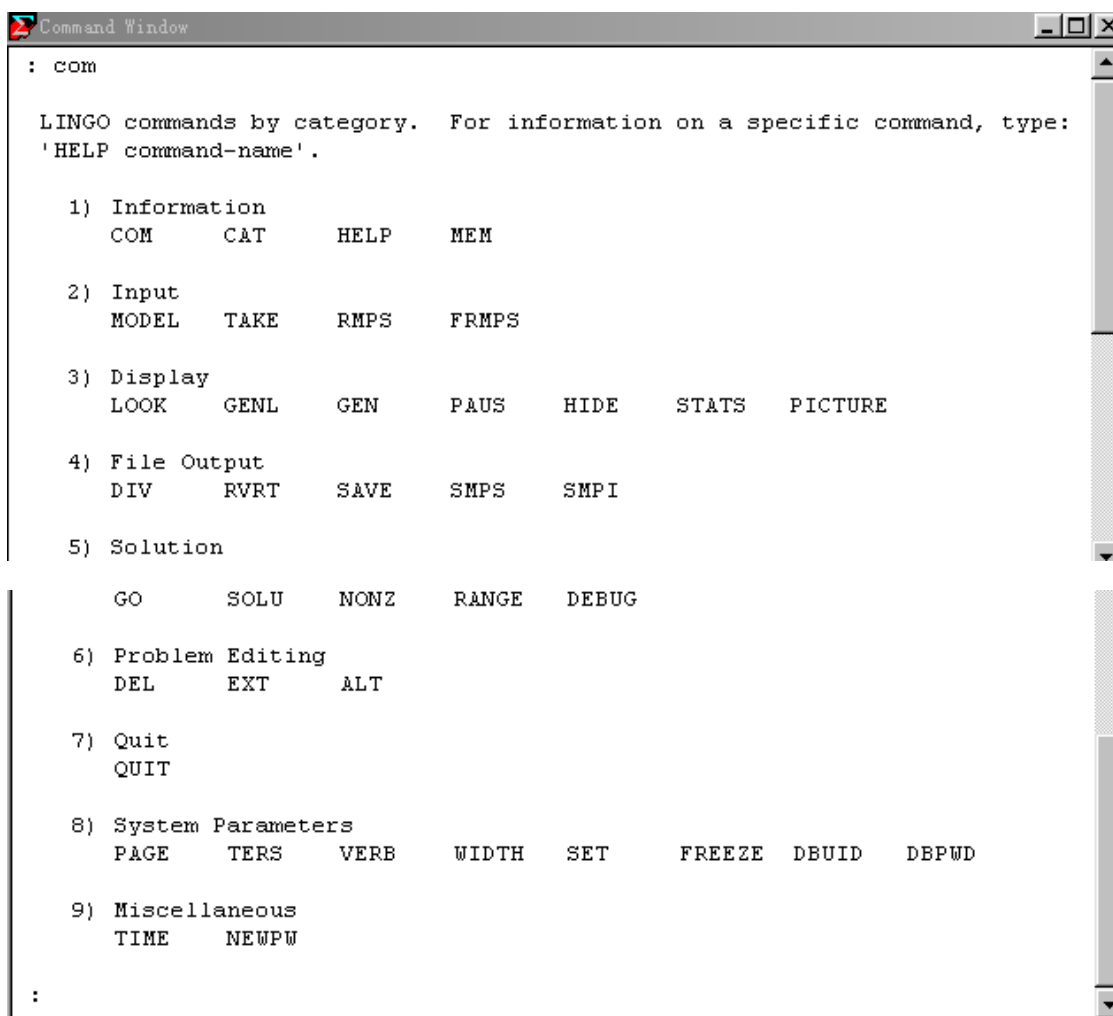


图 3-39 LINGO 的所有行命令

可以看出，LINDO 的不少行命令在 LINGO 中不再支持，如 DATE, TABL, SDBC, FBS, FPUN, SMPN 等。LINGO 也增加了一些与 LINDO 不同的命令，如 MEM, MODEL, FRMPS, GEN, HIDE, SMPI, FREEZE, DBUID, DBPWD 等，表 3-15 简要列出了这些新增命令的基本功能。

表 3-15 部分 LINGO 行命令的基本功能

| 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，才参见“LINGO Generate”菜单命令 |
| 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 DEEAUT”和“FREEZE”两条命令恢复默认设置 |
| 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 APL 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 |

即使双方都有的命令, 也可能在 LINGO 中的功能与在 LINDO 中不完全相同, 如 LINGO 中的 SET 命令能设定的参数数量要比 LINDO 中多出很多。凡是用户能控制的 LINGO 系统参数, SET 命令都能够对它进行设置。SET 命令使用格式为:

SET paramter_name | parameter_index[parameter_value]

这里 paramter_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。此外, SET DEFAULT 命令用于将所有参数恢复为系统的默认值(默认值)。这些设置如果不用 FREEZE 命令保存到配置文件 LINGO.CNF 中, 则退出 LINGO 后这些设置就失效了。

可供设置的参数及其简要说明如表 3-16 所示(参见 LINGO 菜单命令 LINGO | Options 的功能)。

表 3-16 SET 命令可设置的参数及其简要功能

| 索引 | 参数名 | 默认值 | 简要说明 |
|----|--------|---------|---|
| 1 | ILFTOL | 0.3e-5 | 初始线性可行误差限 |
| 2 | FLFTOL | 0.1e-6 | 最终线性可行误差限 |
| 3 | INFTOL | 0.1e-2 | 初始非线性可行误差限 |
| 4 | FNFTOL | 0.1e-5 | 最终非线性可行误差限 |
| 5 | RELINT | 0.8e-5 | 相对整性误差限 |
| 6 | NOPTOL | 0.2e-6 | 非线性规划(NLP)的最优性误差限 |
| 7 | ITRSLW | 5 | 缓慢改进的迭代次数的上限 |
| 8 | DERCMP | 0 | 导数(0: 数值导数, 1: 解析导数) |
| 9 | ITRLIM | 0 | 迭代次数上限(0: 无限制) |
| 10 | TIMLIM | 0 | 求解时间的上限(秒)(0: 无限制) |
| 11 | OBJCTS | 1 | 是否采用目标割平面法(1: 是, 0: 否) |
| 12 | MXMEMB | 32 | 模型生成器的内存上限(兆字节)(对某些机, 可能无意义) |
| 13 | 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) |
| 14 | ABSINT | 0.00001 | 整性绝对误差限 |
| 15 | HEURIS | 3 | 整数规划(IP)启发式求解次数(0: 无, 可设定为 0~100) |
| 16 | HURDLE | none | 整数规划(IP)的“篱笆”值(none: 无, 可设定为任意实数值) |
| 17 | IPTOLA | 0.8e-7 | 整数规划(IP)的绝对最优性误差限 |

| | | | |
|----|--------|--------|---|
| 18 | IPTOLR | 0.5e-7 | 整数规划（IP）的相对最优性误差限 |
| 19 | TIM2RL | 100 | 采用 IPTOLR 作为判断标准之前，程序必须求解的时间（秒） |
| 20 | NODEST | 0 | 分支节点的选择策略（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） |
| 21 | LENPAG | 0 | 终端的页长限制（0：没有限制；可设定任意非负整数） |
| 22 | LINLEN | 76 | 终端的行宽限制（0：没有限制；可设定为 64~200） |
| 23 | TERSEO | 0 | 输出级别（0：详细型，1：简洁型） |
| 24 | STAWLN | 1 | 是否显示状态窗口（1：是，0：否，Windows 系统才能使用） |
| 25 | SPLASH | 1 | 弹出版本和版权信息（1：是，0：否，Windows 系统才能使用） |
| 26 | OROUTE | 0 | 将输出定向到命令窗口（1：是，0：否，Windows 系统才能使用） |
| 27 | WNLIN | 800 | 命令窗口的最大显示行数（Windows 系统才能使用） |
| 28 | 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） |
| 29 | STABAR | 1 | 显示状态栏（1：是，0：否，Windows 系统才能使用） |
| 30 | FILFMT | 1 | 文件格式（0：LNG 格式，1：LG4 格式，Windows 系统才能使用） |
| 31 | TOOLBR | 1 | 显示工具栏（1：是，0：否，Windows 系统才能使用） |
| 32 | CHKDUP | 0 | 检查数据与模型中变量是否重名（1：是，0：否） |
| 33 | ECHOIN | 0 | 脚本命令反馈到命令窗口（1：是，0：否） |
| 34 | ERRDLG | 1 | 错误信息以对话框显示（1：是，0：否，Windows 系统才能使用） |
| 35 | USEPNM | 0 | 允许无限制地使用基本集合的成员名（1：是，0：否） |
| 36 | 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） |
| 37 | 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） |
| 38 | NSLPDR | 1 | 在非线性求解程序中用 SLP 法寻找搜索方向（1：是，0：否） |
| 39 | SELCON | 0 | 在非线性求解程序中有选择地检查约束（1：是，0：否） |
| 40 | 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） |
| 41 | SOLVEL | 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）） |
| 42 | REDUCE | 2 | 模型降维（2：LINGO 决定，1：是，0：否） |
| 43 | SCALEM | 1 | 变换模型中的数据尺度（1：是，0：否） |
| 44 | 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） |
| 45 | DUALPR | 0 | 对偶单纯形法决定出基变量的策略（0：LINGO 自动决定，1：按最大下降比例确定，2：用最陡边法对所有变量进行尝试） |
| 46 | 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） |
| 47 | RCMPSN | 0 | 是否在 MPS 格式的模型中使用 R/C 格式的名称（1：是，0：否） |
| 48 | 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） |
| 49 | BRANDR | 0 | 分支时对变量取整的优先级方向（0：LINGO 自动决定，1：向上取整优先，2：向下取整优先） |
| 50 | BRANPR | 0 | 分支时变量的优先级（0：LINGO 自动决定，1：二进制（0—1）变量） |
| 51 | CUTOFF | 0.1e-8 | 解的截断误差限 |

| | | | |
|----|--------|---------|--|
| 52 | STRONG | 10 | 指定强分支的层次级别 |
| 53 | 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) |
| 54 | REOPTX | 0 | IP 冷启动时的 LP 算法 (选项同上) |
| 55 | MAXCTP | 200 | 分支中根节点增加割平面时, 最大迭代检查的次数 |
| 56 | RCTLIM | 0.75 | 割 (平面) 的个数相对于原问题的约束个数的上限 (比值) |
| 57 | GUBCTS | 1 | 是否使用广义上界 (GUB) 割 (1: 是, 0: 否) |
| 58 | FLWCTS | 1 | 是否使用流 (Flow) 割 (1: 是, 0: 否) |
| 59 | LFTCTS | 1 | 是否使用 Lift 割 (1: 是, 0: 否) |
| 60 | PLOCTS | 1 | 是否使用选址问题的割 (1: 是, 0: 否) |
| 61 | DISCTS | 1 | 是否使用分解割 (1: 是, 0: 否) |
| 62 | KNPCTS | 1 | 是否使用背包覆盖割 (1: 是, 0: 否) |
| 63 | LATCTS | 1 | 是否使用格 (Lattice) 割 (1: 是, 0: 否) |
| 64 | GOMCTS | 1 | 是否使用 Gomory 割 (1: 是, 0: 否) |
| 65 | COFCTS | 1 | 是否使用系数归约割 (1: 是, 0: 否) |
| 66 | GCDCTS | 1 | 是否使用最大公因子割 (1: 是, 0: 否) |
| 67 | SCLRLM | 1000 | 语法配色的最大行数 (仅 Windows 系统使用) |
| 68 | SCLRDL | 0 | 语法配色的延时 (仅 Windows 系统使用) |
| 69 | PRNCLR | 1 | 括号匹配配色 (1: 是, 0: 否, 仅 Windows 系统使用) |
| 70 | MULTIS | 0 | NLP 多点求解的次数 (0: 无, 可设为任意非负整数) |
| 71 | USEQPR | 0 | 是否识别二次规划 (1: 是, 0: 否) |
| 72 | GLOBAL | 0 | 是否对 NLP 采用全局最优求解程序 (1: 是, 0: 否) |
| 73 | LNRISE | 0 | 线性化级别 (0: LINGO 自动决定, 1: 无, 2: 低, 3: 高) |
| 74 | LNBIGM | 100,000 | 线性化的大 M 系数 |
| 75 | LNDLTA | 0.1e-5 | 线性化的 Delta 误差系数 |
| 76 | BASCTS | 0 | 是否使用基本 (Basis) 割 (1: 是, 0: 否) |
| 77 | MAXCTR | 2 | 分支中非根节点增加割平面时, 最大迭代检查的次数 |
| 78 | HUMNTM | 0 | 分支中每个节点使用启发式搜索的最小时间 (秒) |
| 79 | DECOMP | 0 | 是否使用矩阵分解技术 (1: 是, 0: 否) |
| 80 | GLBOPT | 0.1e-5 | 全局最优求解程序的最优性误差限 |
| 81 | GLBDLT | 0.1e-6 | 全局最优求解程序在凸化过程增加的约束的误差限 |
| 82 | GLBVBD | 0.1e+11 | 全局最优求解程序中变量的上界 |
| 83 | 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) |
| 84 | GLBBRN | 5 | 全局最优求解程序中第 1 次对变量分支时使用的分支策略 (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) |
| 85 | GLBBXS | 1 | 全局求解程序选择活跃分支节点的方法 (0: 深度优先, 1: 具有最坏界的分支优先) |

| | | | |
|----|--------|---|---|
| 86 | 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） |
| 87 | 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） |
| 88 | 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） |
| 89 | 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） |
| 90 | DBGWRM | 0 | 设置 Debug 调试命令热启动时所使用的程序,设置同上 |
| 91 | 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） |
| 92 | 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：不使用，1：转化） |
| 93 | LOWMEM | 0 | 是否采用节省内存方式运行求解器：（0：不使用，1：转化） |
| 94 | 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？（0：不填充，1：填充） |

习 题 3

3.1 方程 $\sin x - x^2/2 = 0$ 有几个根？取不同的初值计算，用 LINGO 软件求方程 $\sin x - x^2/2 = 0$ 的所有根的近似解。

3.2 对 $k = 2, 3, 4, 5, 6$ ，用 LINGO 软件分别求一个 3 阶实方阵 A ，使得 $A^k = [1, 2, 3, 4, 5, 6, 7, 8, 9]$ 。这样的 3 阶实方阵 A 一定存在吗？（分别在精确解和近似解的含义下回答这个问题）？

3.3 先用解析方法求出方程组 $\begin{cases} x_1^2 + x_2^2 = 4 \\ x_1^2 - x_2^2 = 1 \end{cases}$ 的精确解，再用 LINGO 软件解这个方程组，

并与精确进行比较，如何才能用 LINGO 求出这个方程组的所有解？

3.4 用 LINGO 软件求解：

$$\begin{aligned}
 \max z &= c^T x + \frac{1}{2} x^T Q x; \\
 \text{s.t.} \quad &-1 \leq x_1 x_2 + x_3 x_4 \leq 1 \\
 &-3 \leq x_1 + x_2 + x_3 \leq 2 \\
 &x_1, x_2, x_3, x_4 \in \{-1, 1\}
 \end{aligned}$$

其中 $c = (6, 8, 4, -2)^T$ ， Q 是三对角线矩阵，主对角线上元素全为-1，两条次对角线上元素全为 2。

3.5 取不同的初值用 LINGO 软件计算下列平方和形式的非线性规划，尽可能求出所有局

部极小点，进而找出全局极小点，并对结果进行分析、比较。

$$(1) \min (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2;$$

$$(2) \min (x_1^2 + 12x_2 - 1)^2 + (49x_1^2 + 49x_2^2 + 84x_1 + 2324x_2 - 681)^2;$$

$$(3) \min (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

$$(4) \min 100\{[x_3 - 10\theta(x_1, x_2)]^2 + [(x_1^2 + x_2^2)^{1/2} - 1]^2\} + x_3^3;$$

$$\text{其中 } \theta(x_1, x_2) = \begin{cases} \frac{1}{2\pi} \arctan(x_2/x_1), & x_1 > 0 \\ \frac{1}{2\pi} \arctan(x_2/x_1) + \frac{1}{2}, & x_1 < 0 \end{cases}$$

3.6 取不同的初值用 LINGO 软件计算下列非线性规划，尽可能求出所有局部极小点，进而找出全局极小点，并对结果进行分析、比较。

$$(1) \min z = (x_1 x_2)^2 (1 - x_1)^2 [1 - x_1 - x_2 (1 - x_1)^5]^2;$$

$$(2) \min z = e^{-x_1 - x_2} (2x_1^2 + 3x_2^2);$$

$$(3) \min z = (x_1 - 2)^2 + (x_2 - 1)^2 + \frac{0.04}{-0.25x_1^2 - x_2^2 + 1} + 5(x_1 - x_2 + 1)^2;$$

(4)

$$\min z = -\frac{1}{(x - a_1)^T (x - a_1) + c_1} - \frac{1}{(x - a_2)^T (x - a_2) + c_2}, x \in R^2, \text{其中 } c = (0.7, 0.73),$$

$$a_1 = (4, 4)^T, a_2 = (2.5, 3.8)^T.$$

3.7 对于如下线性规划问题(有 3n 个决策变量(x,r,s)和 2n 个约束);

$$\min = (-x_n);$$

$$\text{s.t. } 4x_1 - 4r_1 = 1,$$

$$x_1 + s_1 = 1,$$

$$4x_j - x_{j-1} - 4r_j = 0, \quad j = 2, 3, \dots, n,$$

$$4x_j + x_{j-1} + 4s_j = 4, \quad j = 2, 3, \dots, n,$$

$$x_j, r_j, s_j \geq 0, \quad j = 1, 2, \dots, n.$$

- (1) 选用单纯形法，分别对 n 的不同取值（如 n=2, 10, 50, 500 等）求解上述规划。
- (2) 选用内点法，分别对 n 的不同取值（如 n=2, 10, 50, 500 等）求解上述规划。
- (3) 观察这两种不同算法的计算效果是否（只有 x 为决策变量）：

$$\min z = -0.5 \sum_{i=1}^{20} \lambda_i (x_i - 2)^2;$$

$$\text{s. t. } Ax \leq b, x \geq 0.$$

已知 $b = (-5, 2, -1, -3, 5, 4, -1, 0, 9, 40)^T$, A 为 10×20 的矩阵, $A^T = (a_{ij})_{20 \times 10}$, 且

$$a_{i,10} = 1(i = 1, 2, \dots, 20); a_{i,22-i} = -3(i = 13, 14, \dots, 21); a_{i,23-i} = 7(i = 14, 15, \dots, 22); a_{i,24-i} = 0(i = 15, 16, \dots, 23);$$

$$a_{i,25-i} = -5(i = 16, 17, \dots, 24); a_{i,26-i} = 1(i = 17, 18, \dots, 25);$$

$$a_{i,27-i} = 1(i = 18, 19, \dots, 26);$$

$$a_{i,28-i} = 0(i = 19, 20, \dots, 27); a_{i,29-i} = 2(i = 20, 21, \dots, 28); a_{i,30-i} = -1(i = 21, 22, \dots, 29);$$

注意: 在上面的表达中, 当 a_{ij} 中的下标 i 超过 20 时, 应理解为将该下标减去 20 (即对 20

取模), 如 $a_{21,1} = -3$ 的含义是 $a_{1,1} = -3$, $a_{22,1} = 7$ 的含义是 $a_{2,1} = 7$, 依此类推.

假设还已知 $\lambda_i (i = 1, 2, \dots, 20)$ 的取值, 请分别对它的不同取值 (如以下两种取值) 求解上述规划.

$$(1) \quad \lambda_i = 1(i = 1, 2, \dots, 20);$$

$$(2) \quad \lambda_i = i(i = 1, 2, \dots, 20);$$

3.9 取不同的初值计算下列非线性规划, 尽可能用 LINGO 软件求出所有局部极小点, 进而找出全局极小点:

(1)

$$\min z = 0.000089248x - 0.0218343x^2 + 0.998266x^3 - 1.6995x^4 + 0.2x^5;$$

$$\text{s.t. } 0 \leq x \leq 10$$

$$(2) \quad \min z = \cos x_1 \sin x_2 - \frac{x_1}{x_2^2 + 1};$$

$$\text{s.t. } -1 \leq x_1 \leq 2, -1 \leq x_2 \leq 1.$$

$$(3) \quad \min z = -x_1 - x_2;$$

$$\text{s.t. } x_2 \leq 2x_1^4 - 8x_1^3 + 8x_1^2 + 2,$$

$$x_2 \leq 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 + 36,$$

$$0 \leq x_1 \leq 3, 0 \leq x_2 \leq 4.$$

$$(4) \quad \min z = (x_1 - 1)^2 + (x_1 - x_2)^2 (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^5;$$

$$\text{s.t.} \quad x_1 + x_2^2 + x_3^3 = 3\sqrt{2} + 2,$$

$$x_2 - x_3^2 + x_4 = 2\sqrt{2} + 2,$$

$$-5 \leq x_i \leq 5. \quad i=1,2,3,4,5.$$

$$(5) \quad \min \quad z = -25(x_1 - 2)^2 - (x_2 - 2)^2 - (x_3 - 1)^2 - (x_4 - 4)^2 - (x_5 - 1)^2 - (x_6 - 4)^2;$$

$$\text{s.t.} \quad (x_3 - 3)^2 + x_4 \geq 4,$$

$$(x_5 - 3)^2 + x_6 \geq 4,$$

$$x_1 - 3x_2 \leq 2,$$

$$-x_1 + x_2 \leq 2,$$

$$2 \leq x_1 + x_2 \leq 6,$$

$$0 \leq x_1, x_2,$$

$$1 \leq x_3, x_5 \leq 5,$$

$$0 \leq x_4 \leq 6,$$

$$0 \leq x_6 \leq 10.$$

3.10 对问题

min

$$\{ 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(1 - x_2)^2 + (1 - x_4)^2] + 19.8(x_2 - 1)(x_4 - 1) \}$$

增加以下条件，并分别取初值 $(-3, -1, -3, -1)$ 和 $(3, 1, 3, 1)$ ，求解非线性规划：

$$(1) \quad -10 \leq x_i \leq 10;$$

$$(2) \quad -10 \leq x_i \leq 10; x_1 x_2 - x_1 - x_2 + 1.5 \leq 0, \quad x_1 x_2 + 10 \geq 0, \quad -100 \leq x_1 x_2 x_3 x_4 \leq 100;$$

$$(3) \quad -10 \leq x_i \leq 10; x_1 x_2 - x_1 - x_2 + 1.5 \leq 0, \quad x_1 x_2 + 10 \geq 0, \quad x_1 + x_2 = 0, \quad x_1 x_2 x_3 x_4 = 16;$$

再试取不同的初值计算，比较计算结果，你能从中得到什么启示。

第4章 LINGO 软件与外部文件的接口

在第3章中，我们已经介绍了LINGO软件的基本用法，学会了编写简单的LINGO程序建立LINGO优化模型，并进行求解和观察结果，我们也知道，LINGO建模语言允许以简洁、直观的方式描述较大规模的优化问题，模型中所需要的数据可以以一定格式保存独立

的文件中,计算得到的结果也能够输入到文件中保存下来。通过文件输入输出数据对编写好的程序来说是非常重要的,至少有两个好处:

(1) 通过文件输入输出数据可以将 LINGO 程序和程序处理的数据分离开来。“程序和数据”的分离”是结构化程序设计、面向对象编程的基本要求。

(2) 实际问题中的 LINGO 程序通常需要处理大规模的实际数据,而这些数据通常都是其他应用系统中生成的,或者已经存放在其他应用系统中的某个文件或数据库中,也希望 LINGO 计算的结果以文件方式提供给其他应用系统使用。因此,通过文件输入输出数据是编写实用 LINGO 程序的基本要求。

由于通过文件输入输出数据如此之重要,所以本章在第三章的基础上,主要对 LINGO 软件与外部文件的接口进行更进一步的介绍,通过一些简单的例子说明 LINGO 如何通过外部文件输入输出数据。

4.1 通过 Windows 剪贴板传递数据

通过 Windows 剪贴板传递数据是 Windows 应用程序之间传递数据的一种最简捷的方式,不过这种方式传递数据实际上是通过人工干预进行的,严格来说算不上通过文件传递数据。

第 3 章介绍过 LINGO 软件的两个菜单命令“Edit| Paste(Ctrl+V)”和“Edit| Paste Special...”,这两个命令都是将 Windows 剪贴板中的内容粘贴到 LINGO 模型的当前光标处。不同之处在于:

- 前者是通常的“Edit| Paste (粘贴命令)”,它一般仅用于剪贴板中的内容是文本(包括多信息的文本,即 RTE 格式的文本)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,如可以嵌入(插入)其他应用程序中生成的对象(object)或对象的链接(link)。

下面我们通过例子来说明。

4.1.1 粘贴命令的用法

例 4.1 假想一个最简单的采购问题:有多个城市都需要采购一定的物品,但每个城市只允许在自身所在的城市采购,城市 I 的最低需求量为 $NEED(I)$,最大供应量是 $SUPPLY(I)$,单件采购成本是 $COST(I)$. 如何采购总成本最小?

设采购量用 $ORDERED$ 表示,这个问题的优化模型是一目了然的:

$$\begin{aligned} \min & \sum COST(I) \times ORDERED(I) \\ s.t. & NEED(I) \leq ORDERED(I) \leq SUPPLY(I) \end{aligned}$$

显然,只要所给的数据都是有限的非负数,这个问题的最优解也是一目了然的:最优的采购量 $ORDERED$ 就应该是等于最低需求量 $NEED$ (除非问题本身就没有可行解,即某个城市的最低需求量严格大于其最大供应量)。即使如此,我们还是用这个问题作为一个例子,说明如何通过剪贴板传递数据。

编写 LINGO 模型(不妨存放在文件 exam0401.lg4 中),如图 4-1。这个模型目前还不完整,因为集合段定义的城市集合 MYSET 中没有给出元素列表,数据段的 $COST$, $NEED$, $SUPPLY$ 也没有给出数据。我们假设这些数据放在一个 Word 文件的一个表格中(图 4-2),则可以直接利用剪贴板功能从 Word 中把相应的数据复制过来。

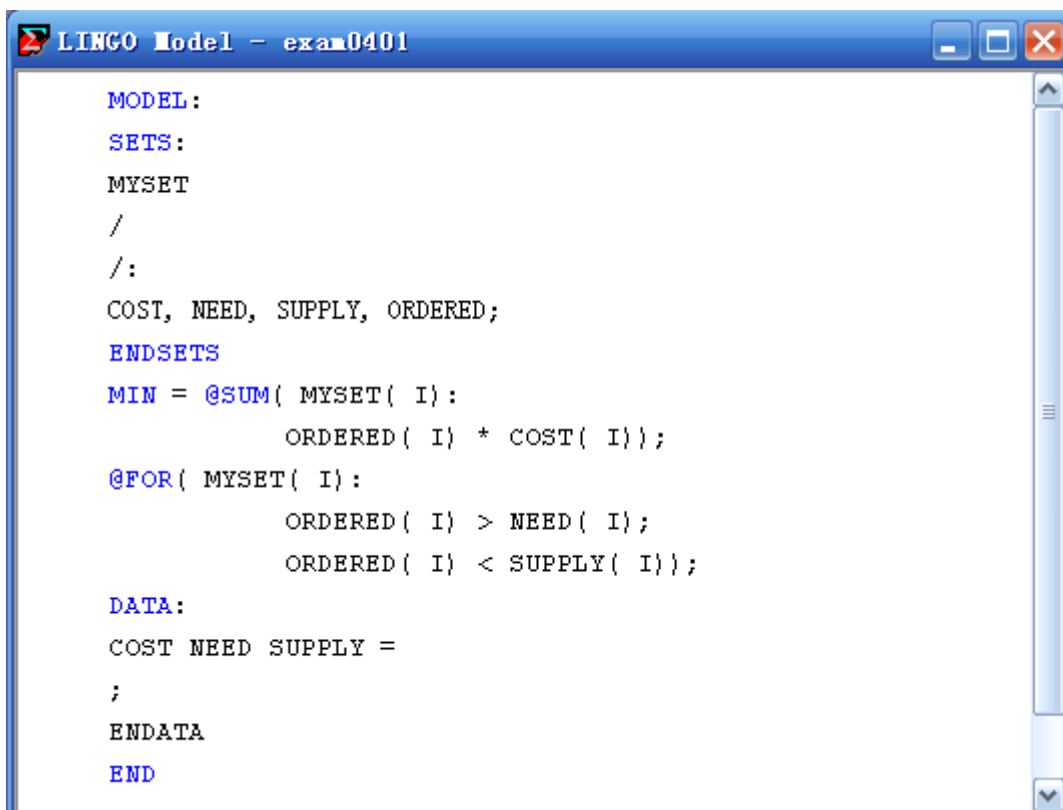


图 4-1 例 4.1 的不完整的模型

我们先在 Word 中将四个城市名所在的单元格复制(Ctrl+C)到剪贴板, 然后回到 LINGO 模型窗口, 利用菜单命令“Edit|Past”或快捷键“Ctrl+V”, 将剪贴板的内容粘贴到集合 MYSET 的元素列表位置。重复一遍上述过程, 将 Word 中的 COST, NEED, SUPPLY 的三行数据也粘贴到 LINGO 模型的数据段, 此时图 4-1 的模型变成如图 4-3 所示的模型。

例 4.1 原始数据表 (a)

| | Seattle | Detroit | Chicago | Denver |
|---------|---------|---------|---------|--------|
| COST | 12 | 28 | 15 | 20 |
| NEED | 1600 | 1800 | 1200 | 1000 |
| SUPPLY | 1700 | 1900 | 1300 | 1100 |
| ORDERED | | | | |

图 4-2 例 4.1 的一种 Word 数据格式

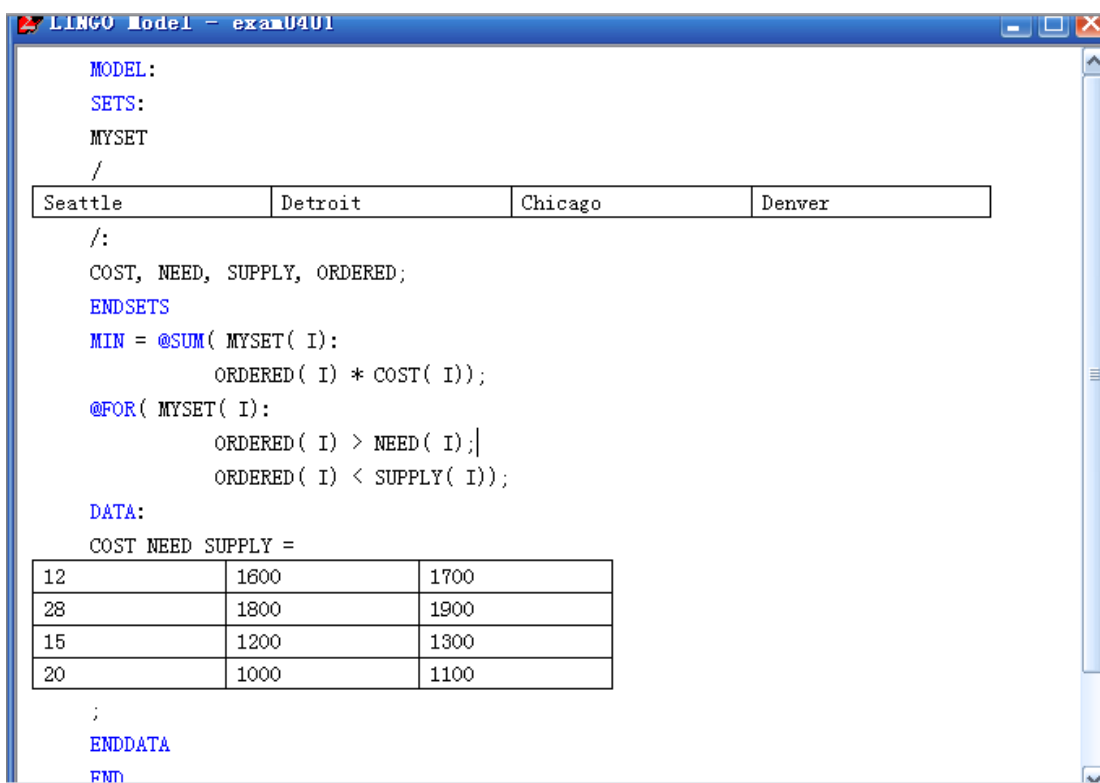


图 4-3 例 4.1 的数据输入错误的模型

在图 4-3 中,我们惊讶地发现粘贴过来的数据仍然保持了 Word 表格的风格,你可以用 LINGO 的菜单命令“Edit|Select Font”重新设定它们的字体,甚至可以在 LINGO 中直接编辑、修改表格中的数据。这些数据能被 LINGO 软件识别,这个模型能正常运行吗?那就用“LINGO|Solve”命令试试!试验的结果告诉我们,这个模型确实能被 LINGO 软件识别和运行,不过 LINGO 运行的时候却报告出错,并告诉我们这个模型没有可行解(具体显示这里略去)。为什么会这样呢?你能发现问题所在吗?

检查一下此时 LINGO 运行后的报告窗口,仔细核对一下报告中显示的数据是否正好是我们希望输入的数据。你会发现集合 MYSET 中给出的元素列表确实是对的,但数据段的 COST, NEED, SUPPLY 给出的数据却不对。这正是问题所在!因为我们在第 3 章中已经说过, LINGO 对集合的属性是按列赋值的,所以 LINGO 处理数据段时,将 12 赋给 COST(1),将 28 赋给 NEED(1),15 赋给 SUPPLY(1),20 赋给 COST(2),1600 赋给 NEED(2),1800 赋给 SUPPLY(2),依此类推,这并不是我们赋值的原意,赋值时将表格的行和列的关系颠倒了!

有两种方法都可以解决这个问题:一是在模型的数据段中对 COST, NEED, SUPPLY 分别赋值(三个语句),这时从 Word 中分三次一行一行地拷贝数据,就不会有错误了。我们这里采用另一种方法,即把 Word 原始数据文件中的数据的行和列换过来,也就是说我们开始建立 Word 数据表时,应该就建成如图 4-4 所示的形式。仍然同前面一样,采用剪贴板复制 Word 表格中的数据到 LINGO 模型的数据段,这时的模型看起来如图 4-5 所示。这时使用“LINGO|Solve”,命令求解模型,一切都正常了,求到的最优值是 107600。



图 4—4 例 4.1 的另一种 Word 数据格式

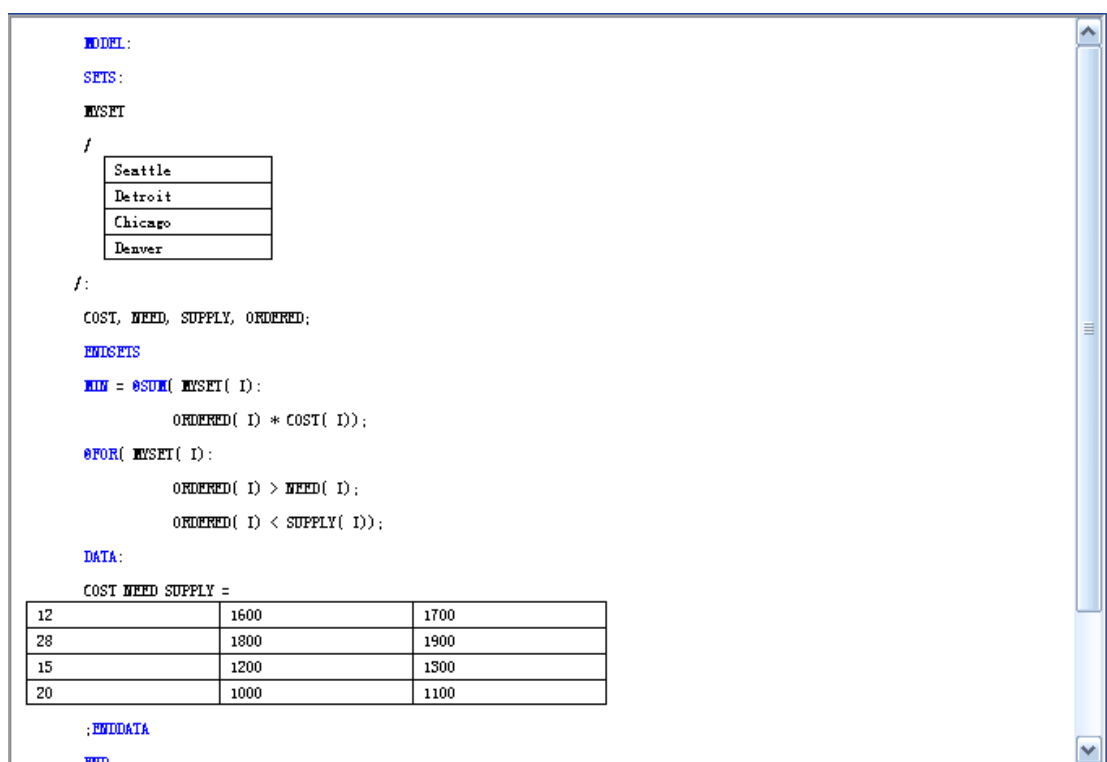


图 4—5 例 4.1 的正确模型

模型求解完后，如果你希望把 LINGO 报告窗口中的答案复制到其他文件（如 Word 文件）中保存下来，同样可以使用剪贴板来完成，只不过是先在 LINGO 报告窗口中使用复制命令，再在 Word 中选择粘贴命令罢了。这里就不再赘述了。

4.1.2 特殊粘贴命令的用法

你可能会想到，由于 Word 文件本身不是文本文件，使用特殊粘贴命令“Edit|Paste Special...”应该更适合。如果选择特殊粘贴命令，则会出现图 4-6 所示的“选择性粘贴”对话框，请你选择粘贴格式。对于本例，你应该选择“多信息文本（RTF）”或“未格式化文本”，实际上，选择“多信息文本（RTF）”时的效果与直接使用“Ctrl+V”的效果是一样的，粘贴的是格式化的（会保留字体、对齐方式、表格线等）；选择“未格式化文本”则会丢失格式信息，如字体、表格的分隔线等信息就会被废弃，变成了纯文本文件。

那么，如果选择其他粘贴方式，效果会如何呢？

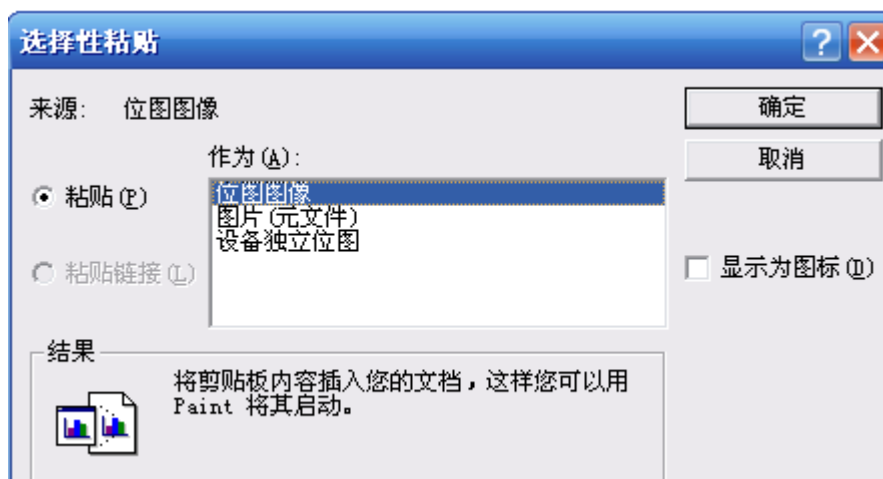


图 4-6 特殊粘贴命令对话框

如果在图 4-6 中选择“Microsoft Office Word 文档”，则粘贴的是一个 Word 对象（object），将来如果用鼠标双击这个对象就会自动用 Word 打开它进行编辑；如果选择“图片”，则将剪贴板中的内容以图形格式插入到 LINGO 模型中，相当于粘贴的是一个图形对象，就无法用 Word 编辑它了。如果在图 4-6 中选择 Word 对象的同时又选择了“显示为图标”选项，则 LINGO 模型中只会显示一个“文档”图标而不显示剪贴板中的具体内容；如果不选择“显示为图标”，虽然 LINGO 中显示的对象的效果看起来与原来 Word 文件中的效果一样，但 LINGO 在运行时完全将它们忽略掉，所以模型无法通过这种方式正确输入数据（如同说明语句一样，只能用作辅助提示信息）。粘贴图形对象的效果也是这样。从专业一点的程序设计角度来看，这也就是“对象嵌入（object embed）”在 LINGO 模型中的含义。此外，插入一种对象后，可以随时用 LINGO 的菜单命令“Edit|Object Propertier(对象属性)”修改这个对象的属性。

如果在图 4-6 中选择“粘贴链接”，效果又如何呢？这时的效果与上面选择“Microsoft Office Word 文档”时是类似的，同样是插入的 word 对象，不同之处在于这时还会同时建立起 LINGO 中插入的文档与原始的数据文件（如原始的 WORD 数据文件）的链接（LINK）。也就是说，当原始的 WORD 数据文件改变时，则 LINGO 中这部分的内容也会随之改变。或者简单地说：“粘贴链接”就是指粘贴了原始文件（“正本”）的一个“副本”，副本会随正本的改变而改变。此外，在 LINGO 模型窗口直接用鼠标双击这个“副本”，也会激活 WORD 开始编辑原来的“正本”文件。虽然 LINGO 中不能通过这种方式来输入数据，但这种功能有助于保持所见到的数据与实际数据的一致性，而且阅读起来非常直观，有时在实际应用中是非常值得提倡的。从专业一点的程序设计角度来看，这就是“对象链接（object link）”在 LINGO 模型中的含义。另外，建立这种链接关系后，可以随时用“Edit|Links...”命令修改这个链接的属性。

实际上，如果希望直接将某个文件的内容全部完整地作为一个对象“嵌入”或“链接”到 LINGO 模型中，可以不通过 Windows 剪贴板，而是直接使用 LINGO 软件的“Edit|Insert New Objects...”命令。这个命令将弹出一个如图 4-7 所示的对话框，你输入（或通过“浏览”选择）希望插入的文件名，按后按下“确定”按钮就可以了，效果同上所述。注意，图 4-7 中如果选择“链接（L）”选项，将建立起 LINGO 中插入的文档与原始的数据文件（这里就是 exam0401.doc）的链接，否则只是插入一个 Word 对象。

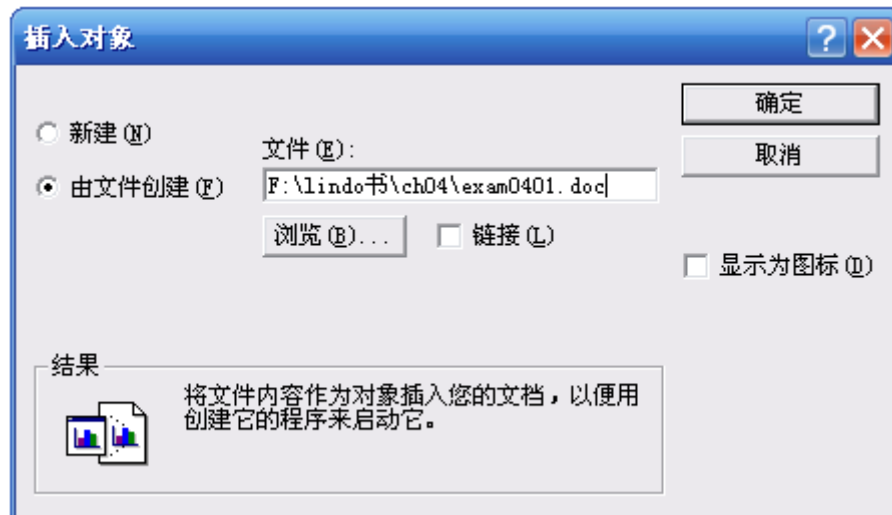


图 4-7 插入对象命令对话框

最后我们说明一下：如果数据不是放在 WORD 文件，而是 EXCEL 电子表格文件或其他应用程序的文件（实际中数据放在 EXCEL 文件中的情形可能是更常见的），操作和结果与上面介绍的过程完全类似，这里不再赘述了。

4.2 通过文本文件传送数据

第 3 章曾简单介绍过，在 LINGO 软件中，通过文本文件输入数据使用的是 @FILE 函数，输出结果采用的是 @TEXT。下面介绍这两个函数的详细用法。

4.2.1 通过文本文件输入数据

@FILE 函数通常可以在集合段和数据段使用，但不允许嵌套使用。这个函数的一般用法是：

@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，其中 filename 为存放数据的文件名（可以包含完整的路径名，没有指定路径时表示在当前目录下寻找这个文件），该文件中记录之间必须用“~”分开。下面通过一个简单的例子来说明。

例 4.2 我们还是用例 4.1 中的问题，说明 @FILE 函数的用法。

假设存放数据的文本文件 myfile.ltd（后缀 ltd 表示 LINGO 数据文件）的内容如下：

Seattle, Detroit, Chicago, Denver~

COST, NEED, SUPPLY, ORDERED~

12, 28, 15, 20~

1600, 1800, 1200, 1000~

1700, 1900, 1300, 1100

现在，在 LINGO 模型窗口中建立如下 LINGO 模型（存放在文件 exam0402.lg4 中）：

MODEL:

SETS:

MYSET / @FILE(myfile.ltd) / : @FILE(myfile.ltd);

ENDSETS

MIN = @SUM(MYSET(I):ORDERED(I) * COST(I));

@FOR(MYSET(I):

ORDERED(I) > NEED(I);

```

        ORDERED( I) < SUPPLY( I));

DATA:
    COST = @FILE( myfile.ldt);
    NEED = @FILE( myfile.ldt);
    SUPPLY = @FILE( myfile.ldt);
ENDDATA
END

```

这时,模型的集合段中对集合MYSET的定义要两次用到@FILE函数(每次从文件中读取一个记录,记录之间用“~”分开). 第一次遇到@FILE(myfile.ldt)时正好是请求从myfile.ldt中读入集合MYSET的元素(因为这个函数放在集合MYSET的元素应该占有的语句的位置上),即MYSET的元素是”Seattle,Detroit,Chicago,Denver”(美国的四个城市名). 第二次遇到@FILE(myfile.ldt)时,这个函数放在集合MYSET的属性应该占有的语句的位置上,所以是请求从myfile.ldt中读入集合MYSET的属性,即MYSET的属性是”COST NEED,SUPPLY,ORDERED”(含义是成本、需求量、供应量、运输量).

在程序的数据段还有三个@FILE函数:

```

    COST=@FILE(myfile.ldt);
其作用是从myfile.ldt中读出数据赋值给属性变量COST,即COST=12,28,15,20;

    NEED=@FILE(myfile.ldt);
其作用是从myfile.ldt中读出数据赋值给属性变量NEED,即NEED=1600,1800,1200,1000;

    SUPPLY = @FILE( myfile.ldt);

```

其作用是从myfile.ldt中读出数据赋值给属性变量SUPPLY,即SUPPLY=1700,1900,1300,1100;

所以,LINGO系统现在就知道了这个模型中只有ORDERED是决策变量. 这时使用”LINGO|Solve”命令求解模型,一切正确,求到的最优值是107600.

显然,当仅仅是输入数据改变时(包括城市的个数及其具体名称、供需量、成本等),只需要改变输入文件myfile.ldt,而程序无需改变,这是非常有利的,因为这样就做到程序与数据的分离.

4.2.2 通过文本文件输出数据

@TEXT函数用于文本文件输出数据,通常只在数据段使用这个函数.这个函数的一般用法是:

```
@TEXT(['filename'])
```

它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中可以包含完整的路径名,没有指定路径时表示在当前目录下生成这个文件(如果这个文件已经存在,将会被覆盖).

例 4.3 在上面例4.2的例子中(模型 exam0402. lg4),如果在数据段增加一些@text语句,则可以输出更多的结果.我们把上面的程序修改为(模型 exam0403. lg4):

```

MODEL:
SETS:
    MYSET / @FILE(myfile.ldt) / : @FILE(myfile.ldt);
ENDSETS
MIN = @SUM( MYSET( I):ORDERED( I) * COST( I));
@FOR( MYSET( I):
    [con1] ORDERED( I) > NEED( I);
    [con2] ORDERED( I) < SUPPLY( I));
DATA:

```

```

COST = @FILE( myfile.lbt);
NEED = @FILE( myfile.lbt);
SUPPLY = @FILE( myfile.lbt);
    @TEXT('exam0403.txt')=@write(4*' ','Value',12*' ','Dual',13*' ',
                                'Decrease',8*' ','Increase',@newline(2));
    @TEXT('exam0403.txt')=@write('Variables:',@newline(2));

@TEXT('exam0403.txt')=Ordered,@DUAL(Ordered),@RANGED(ordered),@RANGEU(ordered);

    @TEXT('exam0403.txt')=@write(@newline(1),'NEED
Constraints:',@newline(2));
    @TEXT('exam0403.txt')=CON1,@DUAL(CON1), @RANGED(CON1),
    @RANGEU(CON1);
    @TEXT('exam0403.txt')=@write(@newline(1),'SUPPLY Constraints:
',@newline(2));
    @TEXT('exam0403.txt')=CON2,@DUAL(CON2), @RANGED(CON2),
    @RANGEU(CON2);
    @TEXT('exam0403.txt')=@write(@newline(1),'Final status for exam0403:
',@status(),
                                @if(@status(),' (Maybe Not Global Optimal)',' (Global
Optimal)'),@newline(1));
ENDDATA
END

```

在数据段,我们总共加了8个含有@TEXT函数的语句:

第1个语句的作用只是输出一个表头;

第2个语句的作用只是输出一个提示行(Variables:)并换行;

第3个语句的作用是输出最优解、对偶价格、敏感性分析(费用系数的允许范围);

第4个语句的作用只是输出一个提示行(Need Constraints:)并换行;

第5个语句的作用是需求约束松弛量、对偶价格、敏感性分析(右端项的允许范围);

第6个语句的作用只是输出一个提示行(Supply Constraints:)并换行;

第7个语句的作用是供应约束松弛量、对偶价格、敏感性分析(右端项的允许范围);

第8个语句的作用是输出求解程序最后的状态及说明(0达到全局最优,否则不一定).

需要注意,这里我们使用了函数@RANGED,@RANGEU,这就要求输出敏感性分析信息(即参数的范围分析),所以在使用这些函数前必须进行敏感性分析.但LINGO的默认设置是不进行敏感性分析的,因此要想以上程序能正常运行,必须先修改选项.这个选项位于LINGO|Options|General Solver 选项卡上,将其中”Dual Computations(对偶计算)”选项的默认设置”Prices(价格)”改为”Prices and Ranges(价格及范围)”即可.

现在就可以求解模型了.当模型求解结束后,LINGO 会将我们用@TEXT 函数请求的值存入文件 exam0403.txt 中(当前目录下将生成这个文件).用文本编辑器打开这个文件,发现其中显示的结果是:

| Value | Dual | Decrease | Increase |
|------------|------|----------|----------|
| Variables: | | | |

| | | | |
|-----------|-----------|-----------|----------------|
| 1600.0000 | 0.0000000 | 12.000000 | 0.10000000E+31 |
| 1800.0000 | 0.0000000 | 28.000000 | 0.10000000E+31 |
| 1200.0000 | 0.0000000 | 15.000000 | 0.10000000E+31 |
| 1000.0000 | 0.0000000 | 20.000000 | 0.10000000E+31 |

NEED Constraints:

| | | | |
|-----------|------------|-----------|-----------|
| 0.0000000 | -12.000000 | 1600.0000 | 100.00000 |
| 0.0000000 | -28.000000 | 1800.0000 | 100.00000 |
| 0.0000000 | -15.000000 | 1200.0000 | 100.00000 |
| 0.0000000 | -20.000000 | 1000.0000 | 100.00000 |

SUPPLY Constraints:

| | | | |
|-----------|-----------|-----------|----------------|
| 100.00000 | 0.0000000 | 100.00000 | 0.10000000E+31 |
| 100.00000 | 0.0000000 | 100.00000 | 0.10000000E+31 |
| 100.00000 | 0.0000000 | 100.00000 | 0.10000000E+31 |
| 100.00000 | 0.0000000 | 100.00000 | 0.10000000E+31 |

Final status for exam0403: 0 (Global Optimal)

值得指出的是，在输出时，可以使用一系列控制输出格式的语句，具体可以参见 3.3.9 节介绍的各种结果报告函数。使用这些函数，将能够输出丰富多彩的结果报告。我们也将 4.4 节中再介绍一个简单的例子。

4.3 通过电子表格文件传递数据

4.3.1 在 LINGO 中使用电子表格的数据

在实际应用中,可能有大量数据是存放在电子表格中的(最常用的大概就是 Excel 软件了,所以我们这里只讨论 Excel 电子表格的情况)。通过 Excel 文件与 LINGO 系统传递数据的函数的一般用法是通过@OLE 函数,与@FILE 函数一样,该函数只能在 LINGO 模型的集合段、数据段和初始段使用。无论用于输入或输出数据,这个函数的使用格式都是:

@OLE(spreadsheet_file [, rang_name_list]

其中 spreadsheet_file 是电子表格文件名称,应当包括扩展名(如*.xls 等),还可以包含完整的路径名,只要字符书不超过 64 均可; rang_name_list 是指文件中包含数据的单元范围(单元范围的格式与 Excel 中工作表的单元范围的格式一样)。

具体来说,当从 Excel 中向 LINGO 模型中输入数据时,在集合段可以直接采用“@OLE (...)”的形式,但在数据段和初始段应当采用“属性(或变量)=@OLE (...)”的赋值形式;当从 LINGO 向 Excel 中输出数据时,应当采用“@OLE (...) =属性(或变量)”的赋值形式(自然,输出语句只能出现在数据段中)。请看下面的例子。

例 4.4 继续考虑上面例 4.2 的例子中(模型 exam0402.lg4),但通过@OLE 语句输入输出数据。

首先,我们用 Excel 建立一个名为 mydata.xls 的 Excel 数据文件,参见图 4-8。为了能够通过@OLE 函数与 LINGO 传递数据,我们需要对这个文件中的数据进行命名。具体做法是:我们用鼠标选中这个表格的 B4:B7 单元,然后选择 Excel 的菜单命令“插入|名称|定义”,这时将会弹出一个对话框,请您输入名称,例如可以将它命名为 cities(这正是图 4-8 所显

示的情形，即 B4:B7 所在的 4 个单元被名为 cities)。同理，我们将 C4:C7 单元命名为 SOLUTION。一般说来，这些单元取什么名字都无所谓，但最好还是取有一定提示作用的名字；另外，这里取什么名字，LINGO 中调用时必须用什么名字，只要二者一致就可以了。

| | A | B | C | D | E | F |
|---|---|------------|------|------|--------|---------|
| 1 | | | | | | |
| 2 | | 例4.5 原始数据表 | | | | |
| 3 | | | COST | NEED | SUPPLY | ORDERED |
| 4 | | Seattle | 12 | 1600 | 1700 | |
| 5 | | Detroit | 28 | 1800 | 1900 | |
| 6 | | Chicago | 15 | 1200 | 1300 | |
| 7 | | Denver | 20 | 1000 | 1100 | |

图 4-8 Excel 文件存放的数据

下面，我们把例 4.2 中的程序修改为（存入文件 exam0404.lg4）：

MODEL:

SETS:

MYSET / @OLE('mydata.xls','CITIES') / :

COST, NEED, SUPPLY, ORDERED;

ENDSETS

MIN = @SUM(MYSET(I): ORDERED(I) * COST(I));

@FOR(MYSET(I):

[CON1] ORDERED(I) > NEED(I);

[CON2] ORDERED(I) < SUPPLY(I));

DATA:

COST, NEED, SUPPLY = @OLE('mydata.xls');

@OLE('mydata.xls','SOLUTION')=ORDERED;

ENDDATA

END

这个程序中有三个@OLE 函数调用，其作用分别说明如下：

- @OLE('mydata.xls','CITIES')：从文件'mydata.xls'的 cities 所指示的单元中取出数据，作为集合 MYSET 的元素。

- COST, NEED, SUPPLY = @OLE('mydata.xls')：这个@OLE 函数调用中没有指明从 mydata.xls 的哪些单元给 COST, NEED, SUPPLY 赋值，这时表示应该采用默认方式（与接收变量的名称相同），即从 mydata.xls 的 COST 指定的单元给 COST 赋值（NEED, SUPPLY 类似）。

- @OLE('mydata.xls','SOLUTION')=ORDERED：将 ORDERED 的值输出赋给 mydata.xls 文件中由 SOLUTION 指定的单元格。

现在运行这个程序，报告窗口将首先显示以下总结报告（Export Summary Report）：

Export Summary Report

Transfer Method: OLE BASED

| | |
|-------------------------|------------|
| Workbook: | mydata.xls |
| Ranges Specified: | 1 |
| SOLUTION | |
| Ranges Found: | 1 |
| Ranges Size Mismatches: | 0 |
| Values Transferred | 4 |

这些信息的意思依次是：采用 OLE 方式传输数据；Excel 文件为 mydata.xls；指定的接收单元范围为 SOLUTION；在 mydata.xls 正好找到一个名为 SOLUTION 的域名；不匹配的单元数为 0；传输了 4 个数值。

如果这时再打开 mydata.xls 查看，你会发现 ORDERED 对应的一列也被自动协商了解答的结果（这正是上面的总结报告的含义所在）。

备注 实际应用中，可能有大量数据是存放在各种数据库中的。不同的数据库可能是由不同的数据库管理系统（DBMS:database management system）进行管理的，所以在使用函数@ODBC 之前，需要在 LINGO 和 DBMS 之间提供接口。Windows 环境下的 ODBC(open database connectivity)接口就是一种绝大多数 DBMS 都支持的标准借口，所以只要你熟悉 ODBC 接口的设置和使用方法（一般应该在 Windows 控制面板中设置），就不难建立起 LINGO 和具体的某个数据库之间的连接。建立起这种连接后，就能够和使用@OLE 函数完全类似地使用@ODBC 函数了。其一般用法是：

• ODBC(['data_source' [, 'table_name' [, 'col_1' [, 'col_2' .. .]]]])

其中 data_source 是数据库名，table_name 是数据表名，col_1,col_2,...是数据列名（数据域名）。具体用法这里就不详细介绍了，有兴趣的读者请参阅 LINGO 的使用手册或在线帮助文档。

4. 3. 2 将 LINGO 模型嵌入、链接到电子表格文件中

如果在求解模型 exam0404.lg4 之前，再采用我们在 4. 1. 2 节中介绍的方法将数据文件 mydata.xls 作为对象链接到模型 exam0404.lg4 中，则求解模型后，模型中的对象（就是文件 mydata.xls）显示的内容应该自动修改，所以 ORDERED 所在列也就应该自动填上解答。可以看出，这是非常直观、方便的。

那么，反过来行不行呢？也就是说，能否在电子表格文件中嵌入一个 LINGO 模型？如果能，那么用户就可以不用每次都打开和运行 LINGO，而是可以直接在 EXCEL 中进行数据维护，同时进行模型优化等多项操作了，这将是非常受实用人员欢迎的。

确实可以做到这一点。下面仍然以 EXCEL 为例，说明如何在 EXCEL 中嵌入一个 LINGO 模型。

例 4. 5 继续考虑上面例 4. 4 的例子。用 EXCEL 打开 mydata.xls（我们将它另存为了 exam0405.xls），执行菜单命令“插入|对象”，看到图 4-9 所示的对话框，我们选择“新建|LINDO Document”，然后按“确定”按钮，则会在 Excel 文件中插入一个空的 LINGO 对象。在这个空的对象框中输入 LINGO 模型（或从 exam0404.lg4 中把文本直接粘贴过来，但把程序中的 mydata.xls 改为 exam0405.xls）。激活这个对象（鼠标双击这个对象），你会看到 Excel 的菜单和工具栏变成了 LINGO 的菜单和工具栏，此时 Excel 的显示如图 4-10。这时，如果想运行这个模型，直接执行“LINGO|SOLVE”命令就可以了，执行命令的结果与前面一个例子相同（ORDERED 所在的列被填上了答案 1600, 1800, 1200, 1000）

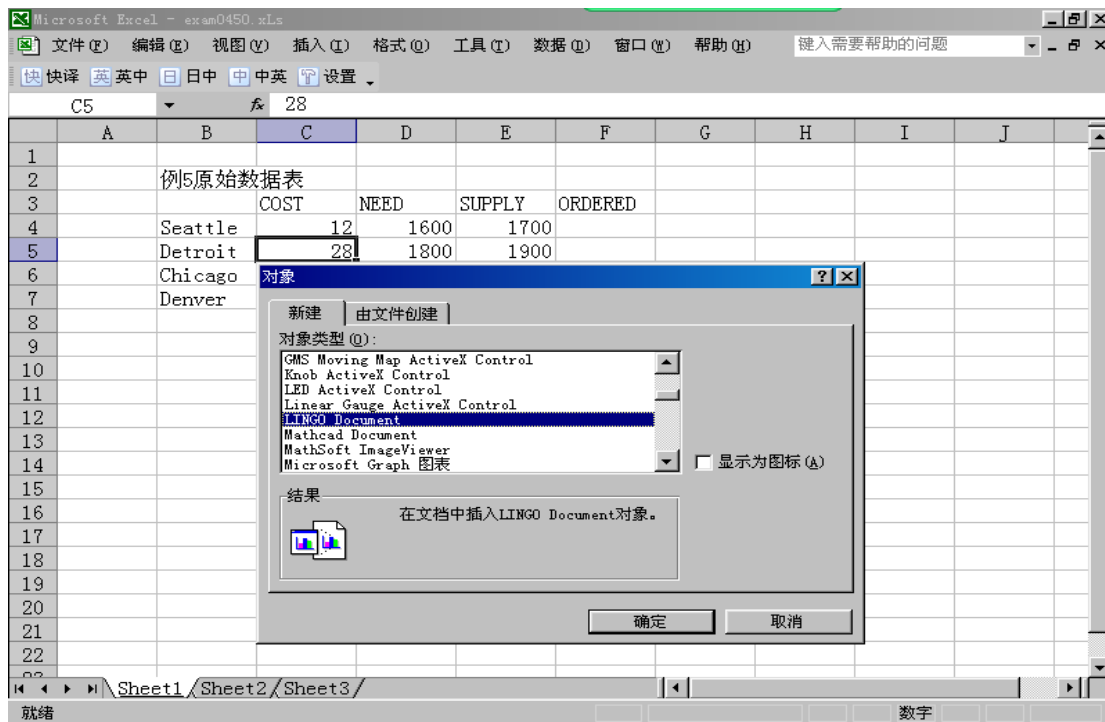


图 4-9 插入 LINDO 对象对话框

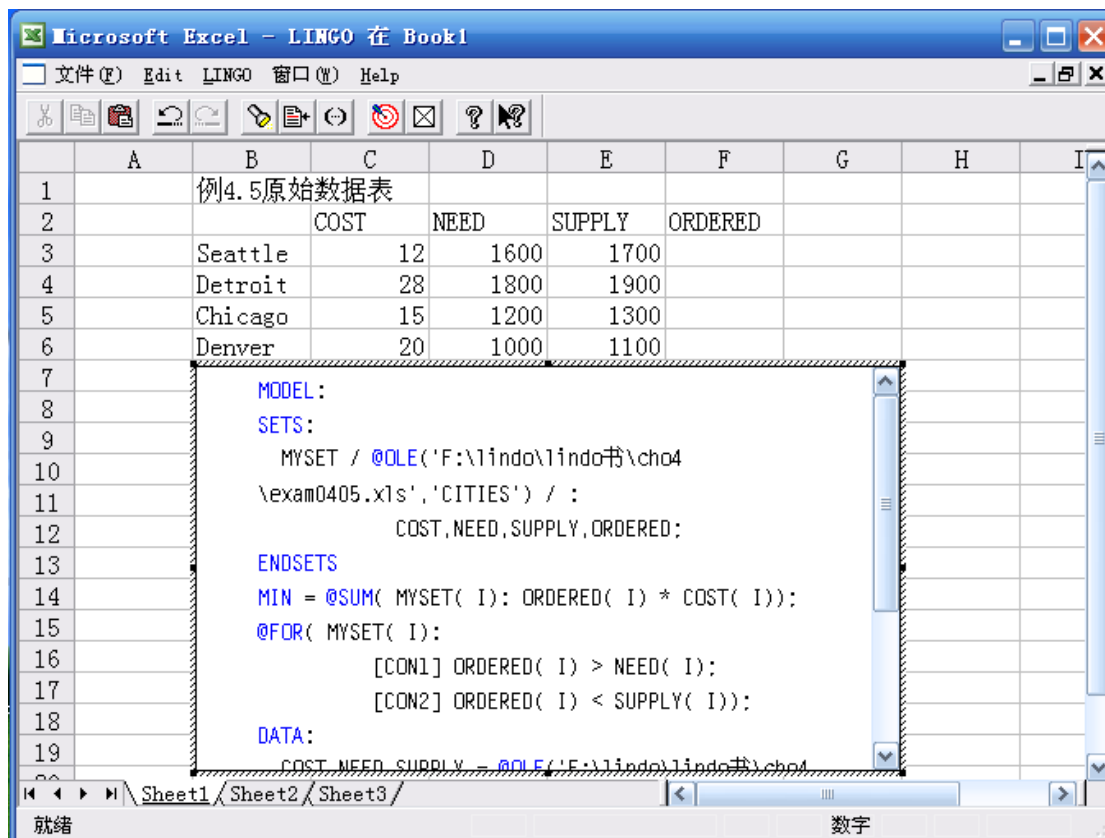


图 4-10 Excel 文件中的 LINGO 模型

你可能已经注意到，在图 4-9 的对话框中除了选项卡“新建”外，还有另选一个选项卡“由文件创建”。当你要插入的对象（LINGO 模型）是已有的文件时，你就可以直接用这个选项卡插入 LINGO 模型对象，如果愿意的话也可以建立同这个对象的链接。这样用法的

含义和我们在 4.1.2 节中介绍的是类似的（只不过当时是把其他对象插入或链接到 LINGO，而这里是把 LINGO 对象插入或链接到 EXCEL），这里就不再多作解释了。

备注 这个例子中的方法虽然在 Excel 文件嵌入了 LINGO 对象，但需要人工干预才能运行这个对象。有时候，人们还希望在 Excel 中自动运行一个 LINGO 程序用命令脚本（行命令序列）进行描述，并需要用到 Excel 的宏命令，我们这里就不再介绍了，有兴趣的读者请参考 LINGO 用户手册或在线帮助文档。

4. LINGO 命令脚本文件

LINGO 命令脚本文件是一个普通的文件，但是文件中的内容是由一系列的 LINGO 命令构成的命令序列。使用命令脚本文件，你可以同时运行一系列的 LINGO 批处理命令。下面举例说明。

例 4.6 假设我们面对下面的实际问题：一家快餐公司有多家分店，每家分店都要确定每天所雇佣的服务员的人数。假设该公司决定采用 1.2.4 节例题 1.4 的模型来确定每家分店每天所雇佣的服务员数量。那么每家分店的优化模型的结构本质上是一样的，只是具体数据不同。我们当然可以分别为每家分店分别进行数据输入和优化，分别求解模型得到决策结果。但一种更好的方法是把每个分店的人员需求数据存入各自的一个数据文件中，并建立一个统一的程序（命令脚本文件）逐个调用这些数据进行求解的结果。

下面只以 3 家分店（分别表示为 AAA，BBB，CCC）为例，分店 AAA 周一到周四每天至少需要 50 人，周五每天至少 80 人，周六和周日至少 90 人；分店 BBB 周一到周四每天至少需要 80 人，周五每天至少 120 人，周六和周日至少 140 人；分店 CCC 周一到周四每天至少需要 90 人，周五每天至少 120 人，周六和周日至少 150 人；

建立一个 LINGO 数据文件 AAA.dlt(分店 AAA 的人员需求数据)，实际上是一个文本文件，内容如下：

```
50 50 50 50 80 90 90
```

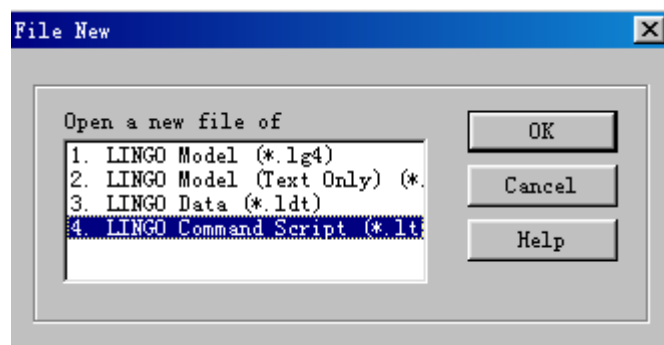
同理建立一个 LINGO 数据文件 BBB.dlt(分店 AAA 的人员需求数据)；

```
80 80 80 80 120 140 140
```

再建立一个 LINGO 数据文件 CCC.dlt(分店 AAA 的人员需求数据)；

```
90 90 90 90 120 150 150
```

这时，用命令脚本文件实现上面功能是方便的。首先，可用任何文件编辑生成命令脚本文件。实际上，LINGO 本身就带有一个很好的编辑器。使用“File|New(新建文件)”菜单命令，系统将会弹出图 4-11 所示的对话框，请求用户选择新建的文件类型，如果将选择项移到“4: LINGO Command Script (*.ltf)”，然后按“OK”按钮，则表示要新建一个 LINGO 命令脚本文件。



假设建立的文件名为 BAT02.ltf(把它保存爱当前目录下；注意其中有不少说明语句使运行结果更有可读性)，文件内容为：

!开始输入关于员工聘用的优化模型；

```

MODEL:
SETS:
    DAYS/MON TUE WED THU FRI SAT SUN/;
    EQUIPED, START;
ENDSETS
DATA:
    !读入需求数据 REQUIRED;
    EQUIPED=@FILE('AAA.LDT');
    将结果START写入文件（这里还特意设计了表头和表尾）;
    @TEXE('F:\lindo\lindo 书\ch04\result\AAA.TXT')=
@WRITE('员工聘用计划表', @NEWLINE(1));
    @TEXE('F:\lindo\lindo 书\ch04\result\AAA.TXT')=
@WRITE('-----', @NEWLINE(1));
    @TEXE('F:\lindo\lindo 书\ch04\result\AAA.TXT')=
@WRITEFOR(DAYS(I)):
    DAYS(I), '(星期', I, '):', 4*'', @FORMAT(START(I), '3.0F'),
    @NEWLINE(I));
    @TEXT('F:\lindo\lindo 书\ch04\result\AAA.TXT')=
@WRITE('-----', @NEWLINE(1));
    @TEXT('F:\lindo\lindo 书\ch04\result\AAA.TXT')=
@WRITE(6*'', '合计:', 6*'', @SUM(DAYS:START),
aNEWLINE(1));
ENDDATA
!目标函数是聘用员工的人数之和;
MIN=@SUM(DAYS(I):START(I));
@FOR(DAYS(J));
!约束条件是满足每天对服务人员的数量要求;
@SUM(DAYS(I)|1#1e#5:
    START(@WRAR()J-I+1,7)))
>=REQUIRED(J)
);
@FOR(DAYS:@GIN(START));
END
!下面求解分店AAA的决策问题;
GO
!下面求解分店BBB的决策问题;
ALTER ALL'AAA'BBB'
GO
!下面求解分店CCC的决策问题;
ALTER ALL'BBB'CCC'
GO
!恢复参数（恢复以正常方式显示解答结果）;
SET TERSEO 0
    提请注意，命令之间的说明语句不需要以“；”结束；但再程序段中（即位于“MODEL：”

```

和“END”之间)的说明语句必须以“;”结束,否则会应为读入的程序不符合LINGO语法而出错。

另外,我们上面的命令脚本程序中还特意安排了对文件的输入输出的处理。

建立好上面的文件后,用“File|take commends”命令打开这个脚本文件,运行是命令窗口将显示一些脚本文件输入和求解模型相关的信息,输出结果再这三个文件“AAA.TXT”、“BBB.TXT”、“CCC.TXT”中(按照程序中的要求,LINGO生成的这三个文件位于目录F:\lindo\lindo 书\ch04\resul),分别是三个分店的解答报告,这正是我们所希望的效果,例如,AAA.txt中的内容是:

员工聘用计划表

| | |
|------------|----|
| ----- | |
| MON (星期1): | 0 |
| TUE (星期2): | 4 |
| WED (星期3): | 40 |
| THU (星期4): | 3 |
| FRI (星期5): | 40 |
| AST (星期6): | 3 |
| SUN (星期7): | 4 |
| ----- | |
| 合计: | 94 |

| 错误代码 | 含义 |
|------|--|
| 117 | 无效的命令 |
| 118 | 命令不明确（例如，可能输入的是命令的缩写名，而这一缩写可有多个命令与之对应） |
| 119 | 命令脚本文件中的错误太多，LINGO 放弃对它继续处理 |
| 120 | LINGO 无法将配置文件（LINGO.CNF）写入启动目录或工作目录(可能是权限问题) |
| 121 | 整数规划没有敏感性分析 |
| 122 | 敏感分析选项没有激活,敏感性分析不能进行(可通过”LINGO OPTIONS”命令对 GENERAL SOLVER 选项卡中的”DUAL+COMPUTATION”进行修改) |
| 123 | 调试(DEBUG)命令只对线性模型,且模型不可行或无界时才能使用 |
| 124 | 对一个空集合的属性进行初始化 |
| 125 | 集合中没有元素 |
| 126 | 使用 ODBC 连接输出时,发现制定的输出变量名不存在 |
| 127 | 使用 ODBC 连接输出时,同时输出的变量的维数必须相同 |
| 128 | 使用 SET 命令时指定的参数索引无效 |
| 129 | 使用 SET 命令时指定的参数的取值无效 |
| 130 | 使用 SET 命令时指定的参数名无效 |
| 131 | FREEZE 命令无法保存配置文件 LINGO.CNF(可能是权限问题) |
| 132 | LINGO 读配置文件(LINGO.CNF)时发生错误 |
| 133 | LINGO 无法通过 OLE 连接电子表格文件(如:当其他人正在编辑这个文件时) |
| 134 | 输出时出现错误,不能完成所有输出操作 |
| 135 | 求解时间超出了限制(可通过” LINGO OPTIONS”命令对 GENERAL SOLVER 选项卡中的”TIME”选项进行修改) |
| 136 | 使用@TEXT 函数输出时出现错误操作 |
| 137 | (该错误编号目前没有使用) |
| 138 | DIVERT(输出重新定向)命令的嵌套次数太多(最多不能超过 10 次嵌套) |
| 139 | DIVERT(输出重新定向)命令不能打开指定文件 |
| 140 | 只求原始最优解时无法给出敏感性分析信息(可通过” LINGO OPTIONS”命令对 GENERAL SOLVER 选项卡中的”DUAL COMPUTATION”选项进行修改) |
| 141 | 对某行约束的敏感性分析无法进行,因为这一行已经是固定约束(即该约束中所有变量都已经在直接求解程序进行预处理时被固定下来了) |
| 142 | 出现了意想不到的错误(请与 LINDO 公司联系解决这个问题) |
| 143 | 使用接口函数输出时,同时输出的对象的维数必须相同 |
| 144 | @POINTER 函数的参数列表无效 |
| 145 | @POINTER 函数出错,:2-输出变量无效;3-内存耗尽;4-只求原始最优解时无法给出敏感想分析信息;5-对固定行无法给出敏感性分析信息;6-意想不到的错误 |
| 146 | 基本集合的元素名与模型中的变量名重名(当前版本的 LINGO 中这本来是允许的,但如果通过” LINGO OPTIONS” 命令对 GENERAL SOLVER 选项卡选择”CHECK FOR DUPLICATES NAMES IN DATA AND MODEL”.则会检查重名,这主要是为了与以前的 LINGO 版本兼容) |

| 错误代码 | 含义 |
|------|---|
| 213 | 当前操作参数不应该是文本，但模型中指定的是文本 |
| 214 | 多次对同一个变量初始化 |
| 215 | @DUAL,@RANGEU,@RANGED 函数不能在此使用(参阅错误代码 168) |
| 216 | 这个函数应该需要输入文本作为参数 |
| 217 | 这个函数应该需要输入数值作为参数 |
| 218 | 这个函数应该需要输入行名或变量名作为参数 |
| 219 | 无法找到指定的行 |
| 220 | 没有定义的文本操作 |
| 221 | @WRITE 或@WRITEFOR 函数的参数溢出 |
| 222 | 需要指定行名或变量名 |
| 223 | 想 EXCEL 文件中写数据时,动态接收单元超出了限制 |
| 224 | 想 EXCEL 文件中写数据时,需要写的数据的个数多于指定的接收单元的个数 |
| 225 | 计算段(CALC)的表达式不正确 |
| 226 | 不存在默认的电子表格文件,请为@OLE 函数指定一个电子表格文件 |
| 227 | 为 APISET 命令指定的参数索引不正确 |
| 228 | 通过 EXCEL 输入输出数据时,如果 LINGO 中的多个对象对应于 EXCEL 中的一个单元范围名,则隶书应该一致 |
| 229 | 为 APISET 命令指定的参数类型不正确 |
| 230 | 为 APISET 命令指定的参数值不正确 |
| 231 | APISET 命令无法完成 |
| 232~ | (该错误编号目前没有使用) |
| 1000 | (错误 为 1000 以上的信息,只对 WINDOWS 系统有效) |
| 1001 | LINGO 找不到与指定括号匹配的括号 |
| 1002 | 当前内存中没有模型,不能求解 |
| 1003 | LINGO 现在正忙,不能马上响应你的请求 |
| 1004 | LINGO 不能写 LOG(日志)文件,也许磁盘已满 |
| 1005 | LINGO 不能打开指定的 LOG(日志)文件 |
| 1006 | 不能打开文件\ |
| 1007 | 没有足够的内存完成命令 |
| 1008 | 不能打开新窗口(可能内存不够) |
| 1009 | 没有足够内存空间生成解答报告 |
| 1010 | 不能打开 EXCEL 文件的链接(通常是由于系统资源不足) |
| 1011 | LINGO 不能完成对图形的请求 |
| 1012 | LINGO 与 ODBC 连接是出错误 |
| 1013 | 通过 ODBC 传递数据时不能完成初始化 |
| 1014 | 向 EXCEL 文件传递数据时,指定的参数不够 |

| 错误代码 | 含义 |
|------|--|
| 147 | @WATN 函数中的条件表达式中只能包含固定变量 |
| 148 | @OLE 函数在当前操作系统下不能使用(只在 WINDOWS 操作系统下可以使用) |
| 149 | (该错误编号目前没有使用) |
| 150 | @ODBC 函数在当前操作系统下不能使用(只在 WINDOWS 操作系统下可以使用) |
| 151 | @POINTER 函数在当前系统下不能使用(只在 WINDOWS 操作系统下可以使用) |
| 152 | 输入的命令在当前操作系统下不能使用 |
| 153 | 集合的初始化(定义元素)不能在初始段中进行,只能在集合段或数据段进行 |
| 154 | 集合名只能被定义一次 |
| 155 | 在数据段对集合进行初始化(定义元素)时,必须显示地列出只有元素,不能省略元素 |
| 156 | 在数据段对结合和(或)变量进行初始化时,给出的参数个数不符 |
| 157 | @INDEX 函数引用的集合名不存在 |
| 158 | 当前函数需要集合的成员名作为参数 |
| 159 | 派生集合中的一个成员(分量)不是对应的父集合的成员 |
| 160 | 数据段中的一个语句不能对两个(或)更多的结合进行初始化(定义元素) |
| 161 | (该错误编号目前没有使用) |
| 162 | 电子表格文件中指定的单元范围内存在不同类型的数据(即有字符,又有增量),LINGO 无法通过这些单元同时输入(或输出)不同类型的数据 |
| 163 | 在初始段对变量进行初始化时,给出的参数个数不符 |
| 164 | 模型中输入的符号不符合 LINGO 的命令规则 |
| 165 | 当前的输出函数不能按集合进行输出 |
| 166 | 不用长度的输出对象无法同时输出到表格型的文件(如数据库和文本文件) |
| 167 | 在通过 EXCEL 进行输入输出时,一次指定了多个单元范围 |
| 168 | @DUAL,@RANGEU,@RANGED 函数不能对文本数据(如集合的成员名)使用,而只能对变量和约束行使用 |
| 169 | 运行模型才能输入集合成员是不允许的 |
| 170 | LINGO 系统的密码输入错误,请重新输入 |
| 171 | LINGO 系统的密码输入错误,系统将以演示版方式运行 |
| 172 | LINGO 的内部求解程序发生了意想不到的错误(请与 LINDO 公司联系解决这个问题) |
| 173 | 内部求解程序发生了数值计算方面的错误 |
| 174 | LINGO 预处理阶段(preprocessing)内存不足 |
| 175 | 系统的虚拟内存不足 |
| 176 | LINGO 后处理阶段(preprocessing)内存不足 |
| 177 | 为集合分配内存时出错(如内存不足等) |
| 178 | 为集合分配内存时堆栈溢出 |
| 179 | 将 MPS 格式的模型文件转化成 LINGO 模型文件时出现错误(如变量名冲突等) |
| 180 | 将 MPS 格式的模型文件转化成 LINGO 模型文件时,不能分配内存(通常是内存不足) |

| 错误代码 | 含义 |
|------|---|
| 181 | 将 MPS 格式的模型文件转化成 LINGO 模型文件时,不能生成模型(通常是内存不足) |
| 182 | 将 MPS 格式的模型文件转化成 LINGO 模型文件时出现错误(会给出出错的行号) |
| 183 | LINGO 目前不支持 MPS 格式的二次规划模型文件 |
| 184 | 敏感性分析选项没有激活,敏感性分析不能进行(可通过“LINGO OPTIONS”命令对 GENERAL SOLVER 选项卡中的”DUAL COMPUTATION”选项进行修改) |
| 185 | 没有使用内点法的权限(LINGO 中的内点法是选件,需要额外购买) |
| 186 | 不能用@QRAND 函数对集合进行初始化(定义元素) |
| 187 | 用@QRAND 函数对属性进行初始化时,一次只能对一个属性进行处理 |
| 188 | 用@QRAND 函数对属性进行初始化时,只能对稠密集合对应的属性进行处理 |
| 189 | 随机函数中指定的种子(SEED)无效 |
| 190 | 用隐式方法定义集合时,定义方式不正确 |
| 191 | LINDO API 返回了错误(请与 LINDO 公司联系解决这个问题) |
| 192 | LINGO 不在支持@WKX 函数,请改用@OLE 函数 |
| 193 | 内存中没有当前模型的解,(模型可能还没有求解,或者求解错误) |
| 194 | 无法生产 LINGO 的内部环境变量(通常是因为内存不足) |
| 195 | 写文件时出现错误(如磁盘空间不足) |
| 196 | 无法为当前模型计算对偶解(这个错误非同寻常,欢迎你将这个模型提供给 LINDO 公司进行进一步分析) |
| 197 | 调试程序目前不能感处理整数规划模型 |
| 198 | 当前二次规划模型是不凸的,不能使用内点法,请通过“LINGO OPTIONS”命令取消对二次规划的判别 |
| 199 | 求解二次规划需要使用内点法,但你使用的 LINGO 版本没有这个权限(请同过“LINGO OPTIONS”命令取消对二次规划的判别) |
| 200 | 无法为当前模型计算对偶解,请通过“LINGO OPTIONS”命令取消对对偶计算的要求 |
| 201 | 模型是局部不可行的 |
| 202 | 全局优化时,模型中非线性变量的个数超出了全局优化求解程序的上限 |
| 203 | 无权使用全局优化求解程序 |
| 204 | 无权使用多初始点求解程序 |
| 205 | 模型中的数据不平衡(数量级差异太大) |
| 206 | “线性化”和“全局优化”两个选项不能同时存在 |
| 207 | 缺少左括号 |
| 208 | @WRITEFOR 函数只能在数据段出现 |
| 209 | @WRITEFOR 函数中不允许出现关系运算符 |
| 210 | @WRITEFOR 函数使用不当 |
| 211 | 输出操作中出现了算术运算错误 |
| 212 | 集合下标越界 |
| | |

建模论文写作问题

一. 科技(数学)论文的撰写

A.科技（数学）论文的写作应包括以下部分内容：

1. **题目**（题目应体现主要内容，取题要实在不要夸大）；
2. **姓名、单位、地名、邮编、(e-mail)**；
3. **摘要**（一般为 50~200 字；包括解决了什么问题、使用的方法、结论各是什么；要求简述且客观实际）；
4. **关键词**（3~5 个），中图分类号；
5. **正文**（一般包括引言、引理（性质）、主要结果、推广、应用、总结等）；
6. **参考文献**：

1⁰.文献是期刊应按以下格式书写：

编号.作者姓名.论文题目[J].期刊名称.出版年.期号：起止页码.

2⁰.文献是专著（书籍）应按以下格式书写：

编号.作者姓名.书名[M].出版地.出版社名称.年份（起止页码）.

注：“编号”按论文中引用的先后次序编号，并在论文中注明[i].

7.英文摘要，包括：题目、作者姓名、单位、地名、邮编、摘要、关键词.

国内的文章（论文）应是：中文论文，英文摘要；英文论文，中文摘要.

B.科技（数学）论文的有关要求

1.论文应该是：创造性论文，研究简报，综述论文，专题论文，应用成果等.创造性论文应是建立某个理论，或某理论的新成果（重要结论、性质等），解决某个尚未解决的问题（如著名人物提出的问题或猜想）.

2.论文要求表达清楚、推理严谨、逻辑性强、数据准确.

3.前人的成果与自己的结论要分明.

4.参考文献只选主要的列入，未引用的不列，未公开发表的不能引用，按引用顺序编号.

5.不能一稿多投，投稿要按论文格式打印（最好用激光打印），并付审稿费.

二. 数学建模论文的撰写

A. 数学建模论文应包括以下内容：

1. 题目

题目的选取要突出问题和模型（即什么问题，哪类数学模型），反映主题思想.避免出现

技术性的术语.例如 96—98 年全国大学生数学建模竞赛题,其建模论文可取这样的题目:最优捕鱼策略模型、关于洗衣机节水的数学模型、零件参数的优化设计、最小费用切割策略、切割次序的优化模型、风险投资组合的线性规划模型、投资组合方案的模糊规划模型、灾情巡视路线的图论模型等.

2.摘要 (200~300 字).包括模型的主要思想、特点、建模方法和主要结果(数据),论文特色要讲清楚,让人看到论文的新意.

3.关键词 (3~5 个).

4.正文.一般由以下几部分组成:

- 1⁰.问题的提出或问题的重述.
- 2⁰.基本假设(模型的假设)与符号约定(包括竞赛题中的假设与数据).
- 3⁰.问题的分析、模型的建立(要求明确写出数学模型).
- 4⁰.模型的求解,包括计算方法设计与实现(程序及计算机输出的计算结果).
- 5⁰.结果分析和检验、优缺点和改进方向(包括误差分析).
- 6⁰.模型的应用与推广.

5.参考文献. (见前面)

6.附录.

7. (英文摘要).

B.数学建模论文的有关要求

1.假设要以严格、确切的数学语言来表达,不致于使读者产生任何曲解;所提出的假设确实是建立数学模型所必需的,与建立模型无关的假设只会扰乱读者的思考;使用的符号在全文中要统一,假设要合理(要与常识、实际、经验、科学相符).

2.全国大学生数学建模竞赛章程规定:“竞赛评奖以假设的合理性、建模的创造性、结果的正确性和文字表达的清晰程度为主要标准”.

3.竞赛论文评审的指标说明:

- 1⁰.假设的合理性:经过抽象、简化,抓住要点,将实际问题用数学语言表达出来;
- 2⁰.建模的创造性:思路的新意和创新在论证及推导中的联想、类比和推广等;

3⁰.结果的正确性：综合分析、求解，计算机及软件包的使用、程序编制、误差分析；

4⁰.表达的清晰性：答卷的文字表达，规范清楚，相对简炼阐明问题及求解过程.

例如：“投资的收益和风险”量化标准：

模型 40%，求解和结果 40%，推广和发挥 10%，文字表达 10%；

“灾情巡视路线”量化标准：

模型与方法 40%，结果 30%，讨论 10%，文字表达 20%.

C.数学建模论文的写作技巧

1. 边建模边写作，建模时得出的信息要记载好或存放在计算机中，先写出部分段落；“假设部分”按序号列出，然后不断修改和充实；
2. 摘要写得要实在，不要夸大，也不要缩小，要有数据结果。一般在论文写完后，最后写摘要和关键词；国家组委会从 2002 年起规定：摘要在整篇论文中占有重要的权重，要认真书写好；
3. 论文语言要精炼、清楚，数学模型要明确写出；某些说明或补充内容可采用附录；
4. 解决得成功的部分、新颖之处要写好写清楚；没有把握的地方不要乱写，有时还要作文字处理。

附：2009 高教社杯全国大学生数学建模竞赛论文格式规范

- 本科组参赛队从 A、B 题中任选一题，专科组参赛队从 C、D 题中任选一题。
- 论文用白色 A4 纸单面打印；上下左右各留出至少 2.5 厘米的页边距；**从左侧装订。**
- 论文第一页为承诺书，具体内容和格式见本规范第二页。
- 论文第二页为编号专用页，用于赛区和全国评阅前后对论文进行编号，具体内容和格式见本规范第三页。
- 论文题目和摘要写在论文第三页上，从第四页开始是论文正文。
- 论文从第三页开始编写页码，页码必须位于每页页脚中部，用阿拉伯数字从“1”开始连续编号。
- 论文不能有页眉，论文中不能有任何可能显示答题人身份的标志。
- 论文题目用三号黑体字、一级标题用四号黑体字，并居中；二级、三级标题用小四号黑体字，左端对齐（不居中）。论文中其他汉字一律采用小四号宋体字，行距用单倍行距，打印时应尽量避免彩色打印。
- 提请大家注意：摘要应该是一份简明扼要的详细摘要（包括关键词），在整

篇论文评阅中占有重要权重,请认真书写(注意篇幅不能超过一页,且无需译成英文)。全国评阅时将首先根据摘要和论文整体结构及概貌对论文优劣进行初步筛选。

- 引用别人的成果或其他公开的资料(包括网上查到的资料)必须按照规定的参考文献的表述方式在正文引用处和参考文献中均明确列出。正文引用处用方括号标示参考文献的编号,如[1][3]等;引用书籍还必须指出页码。参考文献按正文中的引用次序列出,其中书籍的表述方式为:

[编号] 作者,书名,出版地:出版社,出版年。

参考文献中期刊杂志论文的表述方式为:

[编号] 作者,论文名,杂志名,卷期号:起止页码,出版年。

参考文献中网上资源的表述方式为:

[编号] 作者,资源标题,网址,访问时间(年月日)。

- 在不违反本规范的前提下,各赛区可以对论文增加其他要求(如在本规范要求的第一页前增加其他页和其他信息,或在论文的最后增加空白页等);从承诺书开始到论文正文结束前,各赛区不得有本规范外的其他要求(否则一律无效)。
- 本规范的解释权属于全国大学生数学建模竞赛组委会。

[注]

赛区评阅前将论文第一页取下保存,同时在第一页和第二页建立“赛区评阅编号”(由各赛区规定编号方式)，“赛区评阅纪录”表格可供赛区评阅时使用(各赛区自行决定是否在评阅时使用该表格)。评阅后,赛区对送全国评阅的论文在第二页建立“全国统一编号”(编号方式由全国组委会规定,与去年格式相同),然后送全国评阅。论文第二页(编号页)由全国组委会评阅前取下保存,同时在第二页建立“全国评阅编号”。

全国大学生数学建模竞赛组委会

2009年3月16日修订