

附录 LINDO 软件包介绍

提 要

LINDO 是一种专门用于求解数学规划问题的软件包。由于 LINDO 执行速度很快，易于输入、求解和分析数学规划问题，因此在教育、科研和工业界得到广泛应用。LINDO 主要用于求解线性规划、非线性规划、二次规划和整数规划等问题，也可用于一些线性和非线性方程组的求解以及代数方程求根等。LINDO 中包含了一种建模语言和许多常用的数学函数(包括大量概率函数)，可供使用者建立数学规划问题模型时调用。

LINDO 有多种组件和多种版本，本章主要介绍在 PC/DOS 环境下运行的 LINDO、GINO 和 LINGO(包括 LINGO NL)软件的基本使用方法。

f.1 简介与安装

f.1.1 LINDO 软件包简介

LINDO 是一种专门用于求解数学规划问题的优化计算软件包，版权现在由美国 LINDO 系统公司(Lindo System Inc.)所拥有。LINDO 软件包的特点是程序执行速度很快，易于输入、修改、求解和分析一个数学规划(优化问题)，因此 LINDO 在教育、科研和工业界得到广泛应用。有关该软件的发行版本、发行价格和其他最新信息都可以从 LINDO 系统公司的 INTERNET 网络站点 <http://www.lindo.com> 获取，该站点还提供部分 LINDO 软件的演示版本或测试版本。学生版和演示版与发行版的主要区别在于对优化问题的规模(变量和约束个数)有不同的限制。

LINDO 由 Linus Schrage 首先开发，随后又推出了 GINO、LINGO、LINGO NL(又称 LINGO2)和“*What's Best!*”等优化软件，现在一般仍用 LINDO 作为这些软件的统称。各组件的功能各有侧重，分别简要介绍如下：

(1) LINDO 是 Linear Interactive and Discrete Optimizer 字首的缩写形式，可以用来求解线性规划(LP——Linear Programming)、整数规划(IP——Integer Programming)和二次规划(QP——Quadratic Programming)问题。LINDO 学生版可求解多达 200 个变量和 100 个约束的规划问题。

(2) GINO 是 General Interactive Optimizer 字首的缩写形式，可用于求解非线性规划(NLP——Non-Linear Programming)问题，也可用于求解一些线性和非线性方程(组)

以及代数方程求根等. GINO 中包含了各种一般的数学函数(包括大量的概率函数),可供使用者建立问题模型时调用. GINO 学生版可求解多达 50 个变量和 30 个约束的问题.

(3) LINGO 可用于求解线性规划和整数规划问题.

(4) LINGO NL(LINGO2)可用于求解线性、非线性和整数规划问题.

与 LINDO 和 GINO 不同的是, LINGO 和 LINGO NL (LINGO2)包含了内置的建模语言,允许以简练、直观的方式描述较大规模的优化问题,模型中所需的数据可以以一定格式保存在独立的文件中. LINGO 和 LINGO NL (LINGO2)学生版可求解多达 200 个变量和 100 个约束的问题.

(5) “What’s Best!”组件主要用于数据文件是由电子表格软件(如 LUTOS1-2-3 和 MS OFFICE 等)生成的情形. 教材[1]的学生版软件中包括了该书中的大部分例题的相应文件.

LINDO 软件包有多种版本,但其软件内核和使用方法基本上是类似的. 本章主要介绍在微机上 DOS 环境下运行的 LINDO、GINO 和 LINGO (包括 LINGO NL)组件的基本使用方法. 在 DOS 环境下,其使用界面不是图形式的,而是字符式的;不是菜单式的,而是面向具体的命令(Command).

f. 1.2 LINDO 软件包的安装

本章以运筹学教材[1]所附 LINDO 软件的学生版本为基础介绍 LINDO 的使用方法. 该版本在微机上 DOS 环境下通过安装后运行. 安装盘为一片 3 英寸软盘,安装盘上有安装程序(install.exe)和简要介绍文本(read.me),还带有多个压缩文件和一个解压缩程序(lha.exe).

安装时,将安装盘直接插入软驱,然后运行该软盘上的安装程序. 即在 DOS 提示符(如 A: >或 B: >)下键入“install”,按 Enter 键后,只需遵照屏幕上的提示即可完成整个安装过程:

(1) 用户首先必须选择所需安装的内容:LINDO 学生版包含了 5 部分内容,除 f. 1.1 中介绍的(1)~(4)外,还包含教材[1]中的大部分例题(以电子表格形式的文件提供),但不包括 f. 1.1 中介绍的(5). 如果想安装其中某部分内容,就在它前面键入“Y”;否则键入“N”,然后“回车”. 如此反复,完成后屏幕将提示确认用户的选择.

(2) 用户必须选择所需安装到的目的地(一般是硬盘上的某个目录). 屏幕将提示默认的目录,用户可以任意修改,如此反复,完成后屏幕将提示确认用户的选择.

(3) 确认选择之后,安装程序就会自动完成全部后续安装过程. 安装过程结束后,就可以进入安装好的目录,运行相应的优化软件了. 如:如果将 LINDO 安装在 C 盘的 LD 目录下,则在 C: >提示符下键入

```
C: >cd LD
```

```
C: \LD>lindo
```

又如:如果将 GINO 安装在 C 盘的 GINO 目录下,则在 C:>提示符下键入

```
C : >cd GINO
C : \GINO>gino
```

f. 2 用 LINDO 软件求解线性规划、整数规划和二次规划

f. 2.1 初试 LINDO

见“3.5 用 LINDO 软件求解线性规划问题”。

f. 2.2 求解 LP 问题的一般步骤

见“3.5 用 LINDO 软件求解线性规划问题”。

f. 2.3 计算结果显示及敏感性分析

见“4.6 用 LINDO 软件求对偶变量及进行灵敏度分析”。

f. 2.4 注意事项

- (1) 进入 LINDO 后,“:”表示 LINDO 已准备接受一个命令。
- (2) 目标函数及各约束条件之间一定要有“SUBJECT TO(ST)”分开。
- (3) 变量名不能超过 8 个字符。
- (4) 变量与其系数间可以有空格,但不能有任何运算符号(如乘号“*”等)。
- (5) LINDO 中已假定所有变量非负,可用“FREE name”命令将变量 name 的非负假定取消,还可以用“SUB”、“SLB”命令设定变量的上下限。
- (6) 如要输入 \leq 或 \geq 型约束,相应以 $<$ 或 $>$ 代替即可。
- (7) LINDO 不允许变量出现在一个约束条件的右端。
- (8) 一般 LINDO 中不能接受括号()和逗号“,”,例:400(X1+X2)需写为 400X1+400X2;10,000 需写为 10000。
- (9) 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。
- (10) EDIT 命令调用一个全屏幕编辑器,可对当前模型进行全屏幕编辑。编辑完成后用“Esc”键保存当前修改,退出全屏幕编辑器;此时若模型有错误,则要求改正错误后再退出。用“Ctrl+Break”键废弃当前修改,退出全屏幕编辑器。
- (11) LINDO 有 DEL、EXT 及 ALTER 等其他编辑命令,虽然全屏幕编辑器 EDIT 使这些命令用处减少了,但 DEL 在大块地清除一个模型时是有用的,而 ALTER 可允许做全局性的替换。
- (12) LOOK 命令会在屏幕上显示模型(EDIT 也可如此)。

(13) 如想获得敏感性分析可用 RANGE 命令.

(14) SAVE 命令用来存储一个问题模型到文件中, RETR 命令用来读取一个以 LINDO 文件格式存储的模型.

(15) TAKE 命令可用于解读一个以文本格式存储的 LP 问题模型或一系列 LINDO 命令序列.

(16) DIVERT 会导致大多数屏幕信息被输送到一个文件中(文件名要求用户输入), 而只有少量信息被传送到屏幕. RVRT 用于结束 DIVERT, 如果输入的文件名为 PRN, 结果将被直接传到打印机.

(17) LINDO 文件中常有注释间杂于各命令之中, 前面注有[!]符号, 例如: !This is a comment.

(18) LINDO 将目标函数所在行作为第一行, 从第二行起为约束条件. 行号自动产生, 也可以人为定义行号或行名. 行名和变量名一样, 不能超过 8 个字符.

(19) 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, 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. 一个原则是, 系数矩阵中非零元的绝对值不能大于 100000 或者小于 .0001. LINDO 不能对 LP 中的系数矩阵自动进行数值均衡化, 但如果 LINDO 觉得矩阵元素之间很不均衡, 将会给出警告.

(20) 一般错误的避免.

当将一个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, 这类错误虽只是抄写造成的, 但当问题规模较大时, 要搜寻它们也是比较困难的. 在 LINDO 中有一些可帮助寻找错误的功能, 其中之一就是 PICTURE 命令, 它的功能是可以将表达式中的系数通过列表显示出来. 例:

```
MIN 5 A0+6 A1+2 A2+4 B0+3 B1+7 B2+2 C0+9 C1+8 C2
SUBJECT TO
2) A0+A1+A2<=8
3) B0+B1+B2<=9
4) C0+C1+C2<=6
5) A0+B0+C0=6
6) A1+B1+C1=5
7) A2+B2+C2=9
END
```

用 PICTURE 命令可得到

```
  A A A B B B C C C C
  0 1 2 0 1 2 0 1 2 0
```

1:	5	6	2	4	3	7	2	9	8	MIN
2:	1	1	1							<8
3:			1	1	1					<9
4:						1	1	1		<6
5:	1		1					1		=6
6:		1		1		1				=5
7:			1		1		1			=9

从上表可发现,其原因是在表达式 5) 行中将 C0 误写成 CO 了.

f. 2. 5 整数规划(IP)

见“7. 5 用 LINDO 软件求解整数规划并求解例 6. 4. 1 及 6. 4. 2”

f. 2. 6 二次规划(QP)

LINDO 可用于求解二次规划(QP)问题. LINDO 求解 QP 问题时, 第一行(目标函数)只用于给出模型中相应变量的出现顺序(见下面的例子). 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. 并要使用 QCP 命令指明实际约束开始的行号, 然后才可采用 GO 命令求解.

例如: 对于如下二次规划问题

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

用 RT, ONE 和 UL 作为对偶变量, 问题输入如下:

```

MIN X+Y+RT+ONE+UL
ST
6X-Y-1.2RT+ONE>0.
-X+2Y-.9RT+ONE+UL>-.4
1.2X+.9Y>1.1
X+Y=1
Y<.7
END

```

第 1 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. 第 2, 3 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

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

分别对 x, y 求偏导数, 令其大于 0 可得第 2, 3 行约束. 随后操作如下:

```
:QCP
  ROW NO. OF FIRST REAL CONSTRAINT;
      (LINDO 要求输入第一个实际约束的行号)
  4      (第 4 行, 即 1.2X+.9Y>1.1 这一行)
:GO
QP OPTIMUM FOUND AT STEP    7
  OBJECTIVE FUNCTION VALUE
1) 1.3555560
VARIABLE      VALUE      REDUCED COST
X              .666667      .000000
Y              .333333      .000000
RT            10.888890      .000000
ONE            9.399998      .000000
UL             .000000      .366667
ROW           SLACK OR SURPLUS  DUAL PRICES
2)            .000000          -.666667
3)            .000000          -.333333
4)            .000000         -10.888890
5)            .000000          9.399998
6)            .366667          .000000
NO. ITERATIONS=    7
```

f. 2.7 LINDO 命令的译细解释

注意: 需要时, 可键入“COMMANDS”(COM)即可得到 LINDO 的所有有效命令. 如需得到其中某个命令的帮助信息, 可键入“HELP name”, 其中“name”为该命令名.

1) INFORMATION(信息类命令)

HELP 命令:

键入“HELP”会显示出 LINDO 的一般帮助信息. 通过键入“HELP name”, LINDO 可帮助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.

LOCAL(LOC)命令:

给出该程序的版本信息.

CATEGORY(CAT)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:

显示执行 LINDO 后累计的 CPU 时间.

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.

输入程序如下: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”(可简称为“ST”),后面跟约束条件行.最后,输入“END”回到命令状态模式,以后只需给出“GO”命令即可开始优化求解过程.

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,变量系数不能是指数型.例如:258E+29 形式的系数是不允许的.任一系数的整数位数最多为 9 位,小数位数最多为 5 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中,但不能出现在变量名中.一个回车符等价于一个空格.

例:下面是同一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问题模型. LINDO 会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”命令存入的.

RMPS 命令:

转化一个 MPS 格式文件,形成问题输入模型.

TAKE 命令:

用该命令可执行由一系列 LINDO 命令组成的文本文件.该文件内容只能同终端输入一致,例如:文件中不能有行号,且文件中最后一个命令应为“LEAVE”.

LEAVE 命令:

该命令表示结束一个可由“TAKE”命令访问的文件的输入。

RDBC 命令:

给出当前模型的一个初始解,该解是以前由“SDBC”命令存入的。

FBR 命令:

从一个由 FBS 命令建立的文件中得到一个(可行)基,一个与该(可行)基匹配的数学问题表达式必须是内存中已有的。

3) DISPLAY(显示类命令)

PICTURE(PIC)命令:

给出一个 LP 问题中系数矩阵的简图.该命令有助于检查一个模型的输入是否正确。

下面是图中对应字母代表的数字大小:

Z	.000000--	.000001
Y	.000001--	.000009
X	.000010--	.000099
W	.000100--	.000999
V	.001000--	.009999
U	.010000--	.099999
T	.100000--	.999999
A	1.000001--	10.000000
B	10.000001--	100.000000
C	100.000001--	1000.000000
D	1000.000001--	10000.000000
E	10000.000001--	100000.000000
F	100000.000001--	1000000.000000
G	>1000000	

TABLEAU 命令:

显示当前单纯形表中的系数。

LOOK 命令:

可用“LOOK”查看当前问题模型的全部或部分.键入“LOOK ALL”可看全部问题模型.键入“LOOK row1,row2”或“LOOK row1-row2”可看行 row1 至行 row2.模型中系数只能有 5 个小数位,最多有 9 位整数,任何更大的数只能显示为“* * * * *”。

NONZEROES(NONZ)命令:

显示一个小型的解报告,其中只显示非零的变量及相应的行.“NONZ”命令并不能求解问题,所以需首先执行“GO”命令,且“NONZ”只有在设置了“TERSE”交互型模式后才能起作用。

SHOCOLUMN(SHOWC)命令:

键入“SHOC variable-name”可显示出变量 variable-name 的系数列。

SOLUTION(SOLU)命令:

显示解的标准报告.若要存到硬盘上,请见“DIVERT”命令.若需要更简明的解报告,请见“NONZ”、“CPRI”及“RPRI”命令.

RANGE 命令:

显示关于 RHS(右端项)及 OBJ(目标函数费用系数)的范围报告(敏感性分析).

BPICTURE 命令:

按上一次转置/三角化的行序显示当前(可行)基的逻辑图.

4) OUTPUT(输出类命令)

CPRI/RPRI 命令:

可有选择地显示列(用 CPRI 命令)或行(用 RPRI 命令)的有关信息.CPRI 的命令格式如下:

CPRI print-list;conditional-expression

例如,执行

CPRI N P;N="M% %X".AND.D>0

将显示满足下列条件的列(变量)的名称及原始值(PRIMAL VALUE),其条件为:列名的第 1 个字符=M,第 2、3 个字符任意,第 4 个字符=X,且有一对偶值(DUAL VALUE)大于零.

如果“print-list”一项被省略了,将只显示匹配的数目.下面是 print-list 及 conditional-expression 中有关列/行特征的常用符号及意义:

N=NAME(名称).

P=PRIMAL VALUE 原始值(相当于行的松弛量).

D=DUAL VALUE(相当于列的 REDUCED COST).

R=RIM(列对应的目标函数的系数;行的右端项).

U=SIMPLE UPPER BOUND(上界),L=SIMPLE LOWER BOUND(下界).

T=变量类型("C","I",or"F";"<","=",OR">").

Z=一系列或一行中的非零元.

%=名称(N)中的不确定字符.

其他一些有用的符号有:

算术运算符	逻辑运算符	关系运算符	顺序运算符
+ - / * ^	.AND.	><=#	()
LOG()EXP()	.OR.		
ABS()	.NOT.		

SAVE 命令:

将当前的问题模型用 LINDO 格式存储到一个文件中,该问题模型可由“RETR”命令

重新提取.

DIVERT(DIVE)命令:

要求提供一个文件名,随后所有的结果输出(如解集报告等),将转至该文件中,直至给出一个 RVRT(REVERT)命令为止.

RVRT 命令:

重新使以后的所有结果输出都回到终端.该命令的作用与“DIVERT”命令相反.

SMPS 命令:

将当前的问题模型用 MPS 格式存储到文件中.

SDBC 命令:

用数据库格式存储一列.变量名称,大小,REDUCED COST,变量类型,上(SUB)和下界(SLB)分别以(A8,2G15.8,A1,2G15.8)格式存储.(A表示字符型,G表示数值型)

FBS 命令:

将当前的(可行)基存入一个文件.若重新提取,可参见 FBR 命令.

5) SOLUTION(求解类命令)

GO PIV

GO 命令:

求解当前的问题模型.该模型在求解过程中不会被改变.若在 GO 之后有一正整数,表示用单纯形法迭代(旋转,PIVOT)的次数.

PIVOT 命令:

演示单纯形方法的每一步(PIVOT STEP).如果键入“PIVOT”+变量名,则该变量会被输入到解中(入基).变量名后可给出该变量所在的行号(旋转元).

6) PROBLEM EDITING(编辑类命令)

ALTER 命令:

可用“ALTER”编辑当前的问题模型,使用格式为:“ALTER row-id var-id”.“row-id”是需要改动的行的行号,“var-id”是需要改动的变量系数的变量名.随后,LINDO 会提示输入一个新的值.如果想改动 RHS(右端项)、DIRECTION(不等号方向)或某个行的名称,“var-id”一项可用“RHS”、“DIR”或“NAME”.

注:对目标函数而言,有效的 DIRECTIONS 有“MAX”或“MIN”;

对所有其他行可用“<”,“=”,或“>”;

请用“DELETE”命令,而勿用“ALTER”来消去一行.

EXTEND 命令:

可为一个以前定义的问题模型增添约束行.输入新的行,不要再键入“MAX”、“MIN”或“SUBJECT TO”这样的关键词.新的行将会附在原问题模型的最后,键入“END”结束.

DELETE 命令:

键入“DELETE row-id”可从当前的问题模型中消除行“row-id”.也可键入“DELETE

row1 row2”消去行 row1 至行 row2. “DELETE ALL”可用来清除当前的整个问题模型.
SUB 命令:

键入“SUB var-id bound”可以输入或改变一个变量的上界,其中“var-id”是变量名,而“bound”是作为上界的具体数值.

APPEND COLUMN 命令:

键入“APPC var-id”可为问题模型添加以“var-id”命名的新的一列,随后是关于该对应变量出现的行/系数值. 每行有一对行/系数值;行和系数值要用一个空格分开,输入 0 作为行名可结束. 若以“RHS”为列名,将使输入成为新的 RHS(右端项).

SLB 命令:

键入“SLB var-id bound”可以输入或改变一个变量的下界,其中“var-id”是变量名,而“bound”是作为下界的具体数值.

FREE 命令:

键入 FREE X 可使当前数学表达式中的 X 成为一个自由变量,既无上界亦无下界.

EDIT 命令:

LINDO 转为全屏幕编辑状态,光标可游动于问题模型中. 有用的编辑设置包括:

<Home>	光标移动到正文开始处
<End>	光标移动到正文结尾处
<PgUp>/<PgDn>	翻页
Cntrl-S	光标移动到当前行的开始
Cntrl-E	光标移动到当前行的结尾
Cntrl-P	光标移动到相应的匹配括号处
Cntrl-right arrow	光标移动到当前词的结尾
Cntrl-left arrow	光标移动到当前词的开始
Cntrl-Break	废弃当前输入,退出 Edit
<Esc>	退出 Edit(退出前检查当前输入有无语法错误,若有错则要求改正)

用 EDIT 命令编辑的问题模型不能超过 30998 个字符. 问题若更大可用 ALTER、DEL 和 EXTEND 命令.

7) QUIT(退出类命令)

QUIT 命令:

“QUIT”用于退出 LINDO. 任何未存储下来的问题模型会被丢失!

8) INTEGER, QUADRATIC, AND PARAMETRIC PROGRAMS(整数、二次与参数规划类命令)

INTEGER 命令:

INTEGER 命令可将问题模型中的变量标为 0/1 型. 第一种格式为“INTEGER n”.

其中 n 是整型变量的个数, 此时 LINDO 要求整型变量应放在问题模型的最前面. 第二种格式为“INTEGER var-id”, 其中“var-id”是变量名. 执行 GO 命令后, 首先将会显示 LP 最优解, 然后显示一系列渐优的整数规划解.

QCP 命令:

“QCP”用于定义一个二次规划问题. 问题通过在实际约束前增加有关变量的一阶条件转化为线性(互补)型. 这需要我们为每一个实际约束增加一个对偶变量. 要使用 QCP, 第一行(目标函数)只用于给出相应变量的顺序, QCP 用于指出第一个实际约束的行号.

PARA 命令:

键入“PARA row-id new-rhs”可对行“row-id”的 RHS(右端项)进行参数分析. “PARA”会将该行 RHS(右端项)的当前值改为新值“new-rhs”, 同时演示出在由此方式改变任一(可行)基产生的最优目标值. 在此之前, 该问题需被优化过.

POSD 命令:

检验一个“QCP”问题中二次型对应的子约束矩阵是否正定.

TITAN 命令:

该命令可收紧一个 LP/IP 问题模型(加强条件). 第一, 它将收紧上界, 例如:

$$2X + Y < 12$$

$$-X + 2Y < 3$$

执行 TITAN 命令后可将上界减为:

$$\text{SUB } X \ 6$$

$$\text{SUB } Y \ 4.5$$

第二, 会收紧整型变量的系数, 如除上述条件外另外有条件:

$$30W - 3X + 2Y > 3$$

其中 W 是 0/1 型, 那么执行 TITAN 命令收紧后为:

$$21W - 3X + 2Y > 3$$

BIP 命令:

键入“BIP bound-val”会标出“bound-val”在最坏的 IP 解的一个界. 然后 LINDO 会剪除在分枝定界树中的值较差的任一分枝. 例如, 任何已知的可行解可看作一个“BIP”.

GIN 命令:

GIN 命令可将问题模型中的变量标为整数型. 第一种格式为“GIN n ”, 其中 n 是整型变量的个数. LINDO 要求整型变量应放在问题模型的最前面. 第二种格式为“GIN var-id”, 其中“var-id”是变量名. 执行 GO 命令后, 将会显示一系列渐优的 IP 解.

IPTOL 命令:

格式为: IPTOL F , 其中 F 是一个非负分数. 当搜索另一个 IP 解时, 只考虑比目前最好的解至少优 F 的解.

9) CONVERSATIONAL PARAMETERS(对话类命令)

WIDTH 命令:

键入“WIDTH n”可告知 LINDO 你的终端屏幕宽为 n 字符. 例如:“WIDTH 132”表示你的终端使用宽行打印纸.

TERSE 命令:

改变对话方式为 TERSE(简明)型, 例如, 它将不会自动地显示 LP 问题的最优解报告. 使用者可以命令“NONZ”、“CPRI”或“RPRI”浏览解. 命令“VERBOSE”可消除“TERSE”状态.

VERBOSE 命令:

令对话方式回到 VERBOSE(详细)状态(默认状态).

BATCH 命令:

设置对话方式到 BATCH(批处理)状态模式. 分批运行任务, 可使输出更具可读性, 并可在第一个主要错误处停止运行.

PAGE 命令:

设置帧幅. 例如, PAGE 24 将使每屏幕显示 24 行出现一次暂停. 触击一次 CR(回车)将显示下一幅. PAGE 0 表示不设限制, 这对于硬拷屏是适宜的.

PAUSE 命令:

用于一个 TAKE 文件, 它将导致一个暂停直至键入下一个回车. 任何 PAUSE 后的内容(在同一行)被显示到终端上.

10) USER SUPPLIED ROUTINES(用户过程类命令)**USER 命令:**

执行一个用户应用子程序可使用命令 USER(IN), 其中 IN 是当前的输入号. LINDO 中提供了一些子过程可被用户应用子程序调用, 用户应用子程序可用于: 构造一个问题模型; 修改现存的模型; 求解一个问题模型; 重获解的有关信息等(LINDO 学生版中不提供此功能).

11) MISCELLANEOUS(其他命令)**INVERT 命令:**

INVERTS 当前的(可行)基求逆. 通常将使结果更趋精确.

STATS 命令:

给出当前的问题模型统计分析, 如行数、列数(变量数)、整数变量数等.

BUG 命令:

出错报告.

DEBUG 命令:

如因约束系数或右端项中的错误造成问题无可行解, DEBUG 将标出一个包含错误的约束条件的最小集合, 但不能用于二次规划.

SET 命令:

重新设置 LINDO 的内部参数, 格式为:

```
SET PARAM-ID NEW-VALUE
```

可行的参数(PARAM-ID)有:

PARAM-ID	作用
1	最后的约束容限(精度)
2	初始的约束容限(精度)
3	进基变量的 REDUCED COST 容限
4	固定整数变量的阈值
5	旋转次数的阈值

TITLE 命令:

TITLE 'TEXT' 将 'TEXT' 与当前问题相连; TITLE 显示当前的题目.

f. 2. 8 熟练掌握 LINDO

要想学好 LINDO, 首先要掌握 LINDO 中所有的命令, 熟能生巧. 掌握了命令的功能及使用范围, 也不要太拘泥于 LINDO 或别人所教的方法, 要会举一反三, 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向书本学能得来的.

下面举一例说明可同时运行一系列的 LINDO 命令(批处理), 并可使运行结果更有可读性. 例如, 可在 DOS 下用文本编辑器做如下命令序列文件 EXAM001. TXT:

```
BAT
! WE WILL DIVERT OUTPUT TO FILE "RESULT"
DIVE RESULT
! WE WILL RETRIEVE AN OLD PROBLEM PREVIOUSLY
SAVED IN FILE "DIET"
RETR DIET
! WE WILL SEE THE MODEL
LOOK ALL
! SOVLE THE MODEL
GO
! DO WE NEED SENSITIVE ANALYSIS (y/n)?
N
! GIVE SOLUTION?
SOLU
! WE WILL GIVE AN MESSAGE TO THE TERMINAL TO SHOW THE END OF THE
SOLUTION PAUS END OF THE SOLUTION,TYPE THE ENTER KEY,PLEASE!
! CLOSE FILE "RESULT",RETURN TO TERMINAL
```

```

RVRT
! RETURN TO PREVIOUS CONVERSATIONAL STYLE
BAT
! LEAVE
LEAV

```

然后在 LINDO 中运行“TAKE EXAM001.TXT”,就会将一个已有的问题模型“DIET”的目标函数、约束条件及解答保存到文件 RESULT 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.

f.3 用 GINO 求解非线性规划

f.3.1 GINO 命令

GINO 用于求解非线性规划问题. GINO 命令大多数与 LINDO 类似,可用 HELP 来查找其英文注释. 与 LINDO 相比减少了一些命令,如方便的全屏幕编辑器 EDIT. 同时,与 LINDO 相比也增加了一些命令,如模型输入必须用 MODEL 命令开始;GUES 命令用于对某个变量猜想一个初始解;QS 命令用于退出并保存当前模型;SETP 用于修改当前系统参数;OS 命令用于调用 DOS 命令等.

SETP 命令的用法如下: ‘SETP param-index param-value’ 其中

PARAM	INDEX	默认值	含义
EPNEWT	1	.100000E-04	最后紧约束容差
EPINIT	2	.100000E-03	初始紧约束容差
EPSTOP	3	.100000E-04	分数目标函数值的容差
EPSPIV	4	.100000E-02	旋转拒绝的容差
PH1EPS	5	.000000E+00	第一阶段的目标增量
NSTOP	6	3	目标函数值无改进的迭代限制
ITLIM	7	10	NEWTON 迭代的次数限制
LIMSER	8	10000	一维搜索的迭代限制
IPR	9	0	输出结果的打印水平
IPN #	10-13		用于控制结果的强制打印输出
IDUMP	14	0	放弃计算,终止程序执行
IQUAD	15	0	基本变量估计(0:线性,1:四次)
KDERIV	16	0	导数计算方法(0:前向差商,1:中央差商)
MODCG	17	1	控制共轭梯度法
MAXHES	18	-1	HESSIAN 阵大小(-1:变量个数,0:采用共轭梯度法)

f. 3. 2 GINO 的使用

见“12.5 用 GINO 软件解非线性规划”

f. 4 用 LINGO 求解大规模数学规划

f. 4. 1 LINGO、LINGO2 命令

LINGO 和 LINGO NL(即 LINGO2)可分别用于求解线性和非线性规划问题,也可用于求解整数规划问题,同时也是一个矩阵生成器.所谓矩阵生成器,实际上是提供了建立最优化问题实例(模型)的一种语言,有了它,使用者只需键入一行文字就可以建立起成千条约束或目标函数项.掌握这种最优化模型语言是非常重要的,与 LINDO 或 GINO 相比,可使输入较大规模问题的过程得到简化.

LINGO 命令大多数与 LINDO 或 GINO 类似.注意也有一些与 LINDO 不同的命令,如 MODEL、GENL、GEN、EXP 等. MODEL 命令用于输入数学规划模型, GEN 和 GENL 命令可产生一个与当前 LINGO 模型等价的 LINDO 模型形式(在 NLP 问题中 GEN 只能列出所有线性约束或目标函数式). EXP 命令可将当前的解存入已有的文件中(以电子表格文件格式即 WKS 或 WK1 等格式存入,将来可用 @import 函数调用).

f. 4. 2 LINGO 的使用

例 f. 4. 1 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?

该问题模型在 LINGO 中的一个典型输入如下(即学生版中的例子文件 SAIL. LNG):

```
MODEL:
  1] SETS:
  2] QUARTERS/Q1,Q2,Q3,Q4/; TIME,DEM,RP,OP,INV;
  3] ENDSETS
  4] MIN=@SUM(QUARTERS:400 * RP+450 * OP+20 * INV);
  5] @FOR(QUARTERS (I);RP(I)<40);
  6] @FOR(QUARTERS (I)|TIME (I) #GT#1;
  7] INV (I)=INV(I-1)+RP (I)+OP (I)-DEM(I);
```

```

8] INV(1)=10+RP(1)+OP(1)-DEM(1);
9] DATA;
10] DEM=40,60,75,25;
11] TIME=1,2,3,4;
12] ENDDATA
END

```

从上述模型(MODEL)中,我们可看到一个LINGO模型应以MODEL开始,END结束.其间为语句,可分为四大部分(SECTION):

(1) 集合部分(SETS):这部分要以“SETS:”开始,以ENDSETS结束,其作用在于定义必要的变量,在LINGO中称为集合(SET)及其元素(MEMBER或ELEMENT,有点类似于数组的下标)和属性(ATTRIBUTE,有点类似于数组).如上例中定义了集合quarters(含义是季节),这里它包含四个元素即四个季节指标(Q1,Q2,Q3,Q4),每个季节都有时间(TIME)、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40,只需扩展其元素为Q1,...,Q40,每个季节仍然都有TIME,DEM,RP,OP,INV这样的属性.这些量的具体数值可在数据部分(DATA)或初始化部分(INIT)输入.LINGO中的集合有两类:一类是原始集合(PRIMITIVE SETS),如上例中的集合quarters,其定义格式为:

```
SETNAME/member list(or 1..n)/;attribute,attribute,etc.
```

另一类是导出集合(DERIVED SETS),即引用其他集合定义的集合,其定义格式为:

```
SETNAME(set1,set2,etc.);attribute,attribute,etc.
```

(2) 目标与约束部分:这部分实际上定义了目标函数、约束条件等.一般要用到LINGO的内部函数,可在具体使用中体会其功能和用法(详见f.4.3).如上例中,第4行定义了目标函数为MIN=...,其中函数@SUM(quarters:400*RP...)表示对所有的quarters元素计算 $400 * RP + 450 * OP + 20 * INV$ 并求和.注意,如此定义的目标函数与quarters的元素数目是4,40,400或4000并无具体的关系.第5行表示对每个季节,RP不能超过40;第6,7,8行也是对每个季节中生产约束的定义,通常称为“物料守恒方程”.

(3) 数据部分(DATA):这部分要以“DATA:”开始,以ENDDATA结束,其作用在于对集合的属性(数组)输入必要的数值(属性值).格式为:attribute=value_list;如上例中的9-12行.

(4) 初始化部分(INIT):这部分以“INIT:”开始,以ENDINIT结束,其作用在于对集合的属性(数组)定义初值(属性值).格式为:attribute=value_list;上例中没有初始化部分(早期LINGO版本中不包含这部分).

实际上,LINGO的问题模型也可以用类似LINDO的方式输入.如果键入GEN命令,则上述SAILCO问题模型可转化为如下LINDO形式:

```

MIN      400 RP(Q1)+450 OP(Q1)+20 INV(Q1)+400 RP(Q2)+450 OP(Q2)+20INV(Q2)
        +400RP(Q3)+450 OP(Q3)+20 INV(Q3)+400 RP(Q4)+450 OP(Q4)+20 INV(Q4)
SUBJECT TO
2) RP(Q1)≤40
3) RP(Q2)≤40
4) RP(Q3)≤40
5) RP(Q4)≤40
6) -INV(Q1)-RP(Q2)-OP(Q2)+INV(Q2)=-60
7) -INV(Q2)-RP(Q3)-OP(Q3)+INV(Q3)=-75
8) -INV(Q3)-RP(Q3)-OP(Q3)+INV(Q4)=-25
9) -RP(Q1)-OP(Q1)+INV(Q1)=-30
END

```

用 GO 命令求解结果如下:

LP OPTIMUM FOUND AT STEP 7

OBJECTIVE VALUE=78450.0000

VARIABLE	VALUE	REDUCED COST
TIME(Q1)	1.000000	0.000000E00
TIME(Q2)	2.000000	0.000000E00
TIME(Q3)	3.000000	0.000000E00
TIME(Q4)	4.000000	0.000000E00
DEM(Q1)	40.00000	0.000000E00
DEM(Q2)	60.00000	0.000000E00
DEM(Q3)	75.00000	0.000000E00
DEM(Q4)	25.00000	0.000000E00
RP(Q1)	40.00000	0.000000E00
RP(Q2)	40.00000	0.000000E00
RP(Q3)	40.00000	0.000000E00
RP(Q4)	25.00000	0.000000E00
OP(Q1)	0.000000E+00	20.00000
OP(Q2)	10.00000	0.000000E00
OP(Q3)	35.00000	0.000000E00
OP(Q4)	0.000000E00	50.00000
INV(Q1)	10.00000	0.000000E00
INV(Q2)	0.000000E00	20.00000
INV(Q3)	0.000000E00	70.00000
INV(Q4)	0.000000E00	420.00000
ROW	SLACK OR SURPLUS	DUAL PRICE
1	78450.00	1.000000
2	0.000000E00	30.00000
3	0.000000E00	50.00000
4	0.000000E00	50.00000
5	15.00000	0.000000E00

6	0.0000000E00	450.0000
7	0.0000000E00	450.0000
8	0.0000000E00	400.0000
9	0.0000000E00	430.0000

学习 LINGO 使用时应注意以下事项:

(1) 掌握定义数据集及其元素和属性的方法,熟练使用内部函数(尤其是集合函数)来灵活地定义目标和约束(参见 f. 4. 3).

(2) 所有的语句除 SETS、ENDSETS、DATA、ENDDATA、TNIT、ENDINIT 和 MODEL、END 之外必须以一个分号“;”结尾.

(3) 类似 LINDO, LINGO 有 DEL、EXT 和 ALTER 等编辑命令和全屏幕编辑器“EDIT. DEL”可适于大块的删除, ALTER 命令适用于全局性的替换.

(4) 与 LINDO 不同的是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.

(5) 与 GINO 不同,用 LINGO 解非线性规划时已假定各变量非负.

(6) 在 LINGO 模型中,如果想在运行时(执行 GO 命令时)才对变量赋值,可以在数据部分(DATA-ENDDATA 部分)使用输入语句. 但这仅用于对单个变量赋值,而不能用于数组(属性). 输入语句格式为“:变量名=?”. 如:A=? 执行 GO 命令时系统等待用户输入变量 A 的数值.

f. 4. 3 内部函数详细注释

在 LINGO 中建立优化模型时可以引用大量的内部函数,这些函数以“@”符号打头. 以下(1)、(4)、(5)函数中的大部分也可以在上节介绍的 GINO 中引用,但 GINO 中引用时不须以“@”符号打头.

(1) 常用的数学函数

@ABS(X)返回变量 X 的绝对值.

@COS(X)返回变量 X 的余弦值(X 的单位是弧度).

@EXP(X)返回 e^x 的值(其中 e 为自然对数的底,即 2.718281...).

@LGM(X)返回变量 X 的 gamma(伽玛)函数的自然对数值(当 X 为整数时 $LGM(X)=LOG(X-1)$).

@LOG(X)返回变量 X 的自然对数值.

@SIGN(X)返回变量 X 的符号值($X < 0$ 时返回 -1, $X \geq 0$ 时返回 +1). (GINO 中无此函数)

@SIN(X)返回变量 X 的正弦值(X 的单位是弧度).

@SMAX(list)返回一系列数的最大值(GINO 中无此函数).

@SMIN(list)返回一系列数的最小值(GINO 中无此函数).

@TAN(X)返回变量 X 的正切值(X 的单位是弧度).

(2) 集合函数

集合函数的用法如下:

set _ operator (set _ name | condition ; expression)

其中 set _ operator 部分是集合函数名(见下), set _ name 是数据集合名, expression 部分是表达式, | condition 部分是条件, 用逻辑表达式描述(无条件时可以省略). 逻辑表达式中可以有 3 种逻辑算符(#AND#(与), #OR#(或), #NOT#(非))和 6 种关系算符(#EQ#(等于), #NE#(不等于), #GT#(大于), #GE#(大于等于), #LT#(小于), #LE#(小于等于)).

@FOR (set _ name ; constraint _ expressions)对集合(set _ name)的每个元素独立地生成约束, 约束由约束表达式(constraint _ expressions)描述.

@MAX(set _ name ; expression)返回集合上的表达式(expression)的最大值.

@MIN(set _ name ; expression)返回集合上的表达式(expression)的最小值.

@SUM(set _ name ; expression)返回集合上的表达式(expression)的和.

(3) 变量界定函数

变量函数对变量的取值范围附加限制, 共 4 种:

@BND(L, X, U)限制 $L \leftarrow X \leftarrow U$.

@BIN(X)限制 X 为 0 或 1.

@FREE(X)取消对 X 的符号限制(即可取任意实数值).

@GIN(X)限制 X 为整数.

(4) 财务函数

@FPA(I, N)返回如下情形的净现值: 单位时段利率为 I, 连续 N 个时段支付, 每个时段支付单位费用. 即

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

@FPL(I, N)返回如下情形的净现值: 单位时段利率为 I, 第 N 个时段支付单位费用, 即

$$@FPL(I, N) = \left(\frac{1}{1+I}\right)^N.$$

(5) 概率函数

@PSN(X)标准正态分布的分布函数.

@PSL(X)单位正态线性损失函数(即返回 $\text{MAX}(0, Z-X)$ 的期望值, 其中 Z 为标准正态随机变量. 损失函数的定义可参见[1]).

@PPS(A, X)均值为 A 的 Poisson 分布的分布函数(当 X 不是整数时, 采用线性插值进行计算).

@PPL(A,X) Poisson 分布的线性损失函数(即返回 $\text{MAX}(0, Z-X)$ 的期望值, 其中 Z 为均值为 A 的 Poisson 随机变量).

@PBN(P,N,X) 二项分布的分布函数(当 N 和(或) X 不是整数时, 采用线性插值进行计算).

@PHG(POP,G,N,X) 超几何(Hypergeometric)分布的分布函数(当 POP, G, N 和(或) X 不是整数时, 采用线性插值进行计算).

@PEL(A,X) 当到达负荷为 A , 服务系统有 X 个服务器且不允许排队时的 Erlang 损失概率.

@PEB(A,X) 当到达负荷为 A , 服务系统有 X 个服务器且允许无穷排队时的 Erlang 繁忙概率.

@PFS(A,X,C) 当负荷上限为 A , 顾客数为 C , 平行服务器数量为 X 时, 有限源的 Poisson 服务系统的等待或返修顾客数的期望值(A 是顾客数乘以平均服务时间, 再除以平均返修时间. 当 C 和(或) X 不是整数时, 采用线性插值进行计算).

@PFD(N,D,X) 自由度为 N 和 D 的 F 分布的分布函数.

@PCX(N,X) 自由度为 N 的 Chi-squared 分布的分布函数.

@PTD(N,X) 自由度为 N 的 t 分布的分布函数.

@RAND(X) 返回 0 与 1 之间的伪随机数(X 为种子, 典型用法是 $U(I) = \text{@RAND}(U(I+1))$).

(6) 文件输入函数

@FILE

当前模型引用其他 ASCII 码文件中的数据或文本时可以采用该语句(但不允许嵌套使用), 格式为

@FILE(filename);

其中 filename 为存放数据的文件名, 该文件中记录之间用“~”分开. 例如:

MODEL:

SETS:

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

ENDSETS

MIN=@SUM(MYSET(I):

SHIPPED(I)*COST(I);

@FOR(MYSET(I):

SHIPPED(I)>NEED(I);

SHIPPED(I)<SUPPLY(I);

DATA:

COST=@FILE(myfile.txt);

NEED=@FILE(myfile.txt);

```

SUPPLY=@FILE(myfile.txt);
ENDDATA
END
myfile.txt 文件的内容可以是如下格式:
Seattle,Detroit,Chicago,Denver~
COST,NEED,SUPPLY,SHIPPED~
12,28,15,20~
1600,1800,1200,1000~
1700,1900,1300,1100

```

@IMPORT

该语句仅用于当前模型引用其他工作单文件(worksheet)中的数据,且该函数只能在 LINGO 模型的数据部分(DATA-END-DATA)和初始化部分(INIT-ENDINIT)使用. 语句格式为

```
@IMPORT(worksheet filename,data range name);
```

其中 worksheet filename 应当包含扩展名(如. wk1 和. xls 等),还可以包含全路径名,只要字符数不超过 64 均可;data range name 是指工作单中包含数据的单元范围(单元范围的格式与具体的工作单格式一致).

(7) 其他函数

```
@IN(set _ name ,set _ element)
```

如果数据集 set _ name 中包含元素 set _ element 则返回 1,否则返回 0.

```
@SIZE(set _ name)
```

返回数据集 set _ name 中包含元素的个数.

```
@WARN('text',condition)
```

如果条件 condition 成立,求解模型时将显示 'text' 信息.

```
@WRAP(I,N)
```

当 I 位于区间 $[1,N]$ 内时返回 I;否则,从 N 中减去 I 直到 I 位于区间 $[1,N]$,再返回 I (即返回 $J=I-K * N$,其中 J 位于区间 $[1,N]$,K 为整数). 此函数对 $N < 1$ 无定义.

```
@USER
```

该函数是允许用户自己编写的函数,该用户函数应当用 C 语言或 FORTRAN 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(类似于 C 语言中的 main(argc, argv)的编写格式). 在 LINGO 中调用@USER 时则直接指定对应的参数(类似于 C 语言中的 main(argc, argv)的执行格式). (LINGO 学生版中不提供此功能)

[1] Wiuston W L. Operations research: applications and algorithms. third edition, Duxbury press. 1994.