Art of Formulating LPs and IPs from Real Life Problems

Md. Hasib Uddin Molla and M. Babul Hasan[*]

Deepartment of Mathematics, Dhaka University, Dhaka-1000, Bangladesh

Abstract

Formulation of LPs and IPs is a technique to convert real life decision problems into a mathematical model. This model consists of a linear objective function and a set of linear constraints expressed in the form of a system of equations or inequalities. In this paper, we present formulation from real life problem as an art. We discuss formulation through real life example and solve them using computer techniques AMPL and LINDO.

I. Introduction

In practical life, we have to make decision at every steps. While decision making we seek to answer the question 'What is best?'. Always we want the best output with limited resources. This is called optimization. Linear programming (LP) is part of a very important area of mathematics called "optimization techniques". Optimization problems arise in all branches of Economics, Finance, Chemistry, Materials Science, Astronomy, Physics, Structural and Molecular Biology, Engineering, Computer Science and Medicine. LP was developed as a discipline during 2nd world war when the British Military asked scientists to plan expenditures and returns in order to reduce costs to the army and increase losses to the enemy. The development of LP was first started by George B. Dantzig[1,2] by devising the simplex method in 1947. Though the founder of LP is Leonid Kantorovich a Russian mathematician. Before using optimization technique in real life it is required to formulate LP or IP from real life problems. In this section, we will discuss some relevant definitions about LP and IP.

Linear Programming (LP)

Linear programming[3] is a technique for optimizing linear objective function, subject to linear equality and linear inequality constraints where all the variables must be non-negative. The objective function may be profit, loss, cost, production capacity or any other measure of effectiveness which is to be maximized or minimized.

In matrix vector notation an LP problem can be expressed as:

Optimize: $Z = cx$ (1)

subject to: $Ax \ (\leq, =, \geq) \ b$ (2)

$x_j \geq 0,$ where $j=1, 2 \dots n$ (3)

where A is an $(m \times n)$ matrix, x is an $(n \times 1)$ column vector, b is an $(m \times 1)$ column vector and c is a $(1 \times n)$ row vector.

*Author for correspondence, *e-mail :* mbabulhasan@yahoo.com

Integer Programming (IP)

Integer programming[4] problem is a special type of linear programming problem where some or all of the variables are required to be integers while in linear programming problem variables are only non-negative real numbers.

In matrix vector notation an IP problem can be expressed as:

Optimize: $Z = cx$

subject to: $Ax \ (\leq, =, \geq) \ b$

x_j integer (for some or all $j=1, 2 \dots$ n) (4)

where A is an $(m \times n)$ matrix, x is an $(n \times 1)$ column vector, b is an $(m \times 1)$ column vector and c is a $(1 \times n)$ row vector.

Preliminaries of LP and IP

Decision Variables: The variables that must be decided by the model are called decision variables.

Objective function: The linear function which is to be optimized is called objective function of the linear programming problem (LPP).

Constraints: The set of equations or inequalities which allows decision variables to take on certain values & exclude others is called the constraints of the linear programming problem.

Pure IP: An IP in which all variables are required to be integers is called a pure integer programming problem.

Mixed IP: An IP in which only some of the variables are required to be integers is called a mixed integer programming problem.

Binary IP: An IP in which all the variables must equal 0 or 1 is called binary integer programming problem or 0-1 IP.

Paper Outline

The rest of the paper is organised as follows. In Section 2, we discuss general procedures of LP and IP formulations. In Section 3, we formulate LP and IP from real life decision problem. In Section 4, we find solution of the resulting LP and IP using computer techniques AMPL[5] and LINDO and present a comparison[6] between them. We then conclude the paper at Section 5.

II. Formulation

In this section, we will discuss about some basics of formulation and strategy of formulation from real life decision problems.There are three basic steps in formulation[7]. They are: 1. Identifying the *unknown variables* 2. Identifying all the restrictions or *constraints* and 3. Identifying the *objective function.*

Strategy during formulation

To formulate a given decision problem in mathematical form, we should understand the problemby carefully reading and re-reading the problem statement. Because it is important to understanding the problem for finding out the decision variables, objective function and constraints.

First, we have to identify what type of decision problem is this? There are several types of decision problems[3,4,8], such as Diet problem, Working–Scheduling problem, Capital Budgeting problem, Short-Term Financial Planning problem, Blending problem, Production Process problem, Multi-period Decision problem, Multi-period Financial Planning problem etc.

To identify the decision variables, we have to identify what must be decided by the model, what is the model going to tell the decision maker about the decision problem being faced. Decision variables are sometime called activity levels, since they represent the level at which certain activities are performed. What are the activities in the problem? In what units should they be measured? Answers to these questions can usually be obtained by studying the language management uses in describing the problem situation.

Once the decision variables have been identified, including units of measure, it is often very helpful to tabulate the data for a problem, again being careful to identify the units in which the data is given. This means pulling the numerical values out of the verbal discussion, and presenting them in a table or set of tables so that their meaning and purpose can be clearly identified. Sometimes it is very helpfull if we make a set of tables instead of making a single table by dividing the problems into smaller parts. Now, we are ready to begin work on the objective function and the constraints. This is where the algebraic translating or manipulating work occurs. It is frequently helpful to break a long complex expression up into simpler component parts. For example, if a profit function for the objective function needs to be developed in terms of the difference between a revenue function and a cost function, write down the revenue function and cost function separately first, then subtract them algebraically. If the cost function has several component parts, write down each part first, then sum them up to get total cost. In other cases, a proportionality constraint may be most naturally stated as a ratio between two expressions, which is not in standard LP form. Continue this process and write down the fraction first as it is stated verbally, then perform the required algebraic manipulations necessary to write it in the standard LP form. Finally we have to identify wheather the decision variables are integer or not.

There is no general rule that can be used to all types of LP model formulation but the general procedures are almost the same. That is, we have to find decision variables, define objective functions, identify constraints and set restriction to the variables in all case. But while doing this, thinking isn't the same in all types of problems. But we can apply the techniques obtained from one problem to another problem. Understanding the problem scenario is the main challenge in formulation. It requires a deeper understanding of what we want to do when we build models[4].

III. Real Life Example

In this section, we will formulate a model from real life problem[8].

Lizzie's Cheese Production Problem: Lizzie's Dairy produces cream cheese and cottage cheese. Milk and cream are blended to produce these two products. Both high-fat and low-fat milk can be used to produce cream cheese and cottage cheese. High-fat milk is 60% fat; low-fat milk is 30% fat. The milk used to produce cream cheese must average at least 50% fat and that for cottage cheese, at least 35% fat. At least 40% (by weight) of the inputs to cream cheese and at least 20% (by weight) of the inputs to cottage cheese must be cream. Both cottage cheese and cream cheese are produced by putting milk and cream through the cheese machine. It costs 40¢ to process 1 lb of inputs into a pound of cream cheese. It costs 40¢ to produce 1 lb of cottage cheese, but every pound of input for cottage cheese yields 0.9 lb of cottage cheese and 0.1 lb of waste. Cream can be produced by evaporating high-fat and low-fat milk. It costs 40¢ to evaporate 1 lb of high-fat milk. Each pound of high-fat milk that is evaporated yields 0.6 lb of cream. It costs 40¢ to evaporate 1 lb of low-fat milk. Each pound of low-fat milk that is evaporated yields 0.3 lb of cream. Each day, up to 3,000 lb of input may be sent through the cheese machine. Each day, at least 1,000 lb of cottage cheese and 1,000 lb of cream cheese must be produced. Up to 1,500 lb of cream cheese and 2,000 lb of cottage cheese can be sold each day. Cottage cheese is sold for $1.20/lb and cream cheese for $1.50/lb. High-fat milk is purchased for 80¢/lb and low-fat milk for 40¢/lb. The evaporator can process at most 2,000 lb of milk daily. Formulate an LP that can be used to maximize Lizzie's daily profit.

Formulation

First we will see what we are going to do here.

Two types of product are produced here:

1. Cream Cheese
2. Cottage Cheese

Two types of ingredient that are blended to make these two products are:

1. Milk
2. Cream

There are two types of milk available:

1. High fat milk

2. Low fat milk

Cream can be produced by evaporating each type of milk running through evaporator.

Key: The given problem is a *Production process problem*, where production of cream from milk affects the production of cheese in later stage. Here milk and cream are *blended* in some desired proportion to produce cheese. We will assume that the amount of cream produced from milk every day is fully used to produce cheese.

Here we have to make following three types of decisions

1. How much each type of milk should be used to produce cream by evaporating?

2. How much cream should be used to produce each type of cheese?

3. How much each type of milk should be used to produce each type of cheese?

Before defining the decision variable we first define the products and the inputs as:

Table 1. Defining products and inputs

Products	Cream cheese (1)	Cottage cheese (2)
Inputs/ Milk	High Fat Milk (1)	Low Fat Milk (2)

Now we will define the decision variables as:

MC_i = amount of milk (in lb) i used daily for producing cream (i = 1, 2)

C_i = amount of cream (in lb) used daily for producing product j(j = 1, 2)

$M_{i\ i}$ = amount of milk (in lb) i used daily for producing product j(i= 1, 2 ; j = 1, 2)

Here all of the variables are non- negative real numbers. Now we will draw a figure according to our production model that will show us the daily production process in details.
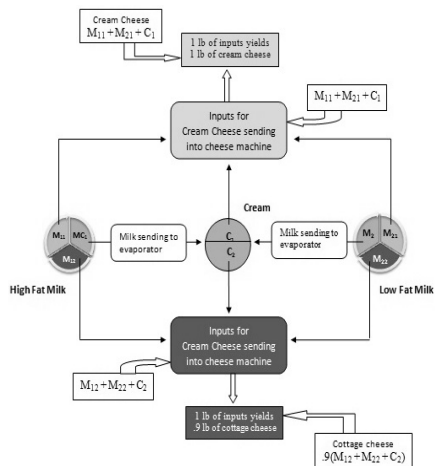

Fig. 1. Production process of two types of cheese

From the above figure we see that:

Total amount of high fat milk

$= (M_{11} + M_{12} + MC_1)$ lb

Total amount of low fat milk

$= (M_{21} + M_{22} + MC_2)$ lb

Total amount of cream cheese produced

$= (M_{11} + M_{21} + C_1)$ lb

Total amount of cottage cheese produced

$= .9 (M_{12} + M_{22} + C_2)$ lb

Our objective is to maximize daily profit. Now we will look for the information in the given problem and gradually build a model.

Both types of milk contain fat. Fat belongs to each type of milk and price of milk per pound are given in the following table:

Table 2. Price and amount of fat in milk

	Hight Fat Milk	Low Fat Milk
Fat	60%	30%
Price	80 ¢	40 ¢

There are minimum requirements of fat & creams in both of the products are given in the following table:

Table 3. Minimum requirement of fat and cream in cheese

	Cream Cheese	Cottage Cheese
Fat	50%	35%
Cream	40%	20%

Minimum requirements of fat & cream must be met in both types of cheese. So the constraints corresponding to fat are:

$.6M_{11} + .3 M_{21} \geq .5(M_{11} + M_{21})$

$.6M_{12} + .3 M_{22} \geq .35(M_{12} + M_{22})$

And the constraints corresponding to cream are:

$C_1 \geq .4(M_{11} + M_{21} + C_1)$

$C_2 \geq .2(M_{12} + M_{22} + C_2)$

Costs for purchasing milk

$.8(M_{11} + M_{12} + MC_1) + .8(M_{21} + M_{22} + MC_2)$

Machine costs for producing per pound of the products, selling price per pound and the output corresponding to 1 pound of inputs for the products are also given in the following table:

Table 4. Cost, selling price and output rate of cheese

	Cream Cheese	Cottage Cheese
Costs	40 ¢	40 ¢
Price	$ 1.50	$ 1.20
Output	1 lb	.9 lb

Machine costs $= .4\{(M_{11} + M_{21} + C_1)$

$$+ .9(M_{12} + M_{22} + C_2)\}$$

Total revenue $= 1.5 (M_{11} + M_{21} + C_1)$

$$+(1.2)(.9)(M_{12} + M_{22} + C_2)$$

Cream can be produced only by evaporating both types of milk. Cost for evaporating per pound of each type of milk and amount of cream that yield in each type of milk are given in the following table:

Table 5. Costs for evaporating and output rate of cream

	High Fat Milk	Low Fat Milk
Costs	40 ¢	40 ¢
Cream	.6 lb	.3 lb

Amount of cream produced from milk by evaporating is equal to the amount of cream used to produce cheese. So the constraint is

$$.6MC_1 + .3MC_2 = C_1 + C_2$$

Costs for evaporating milk $= .4(MC_1 + MC_2)$

Minimum requirements for daily production and maximum market demands for the products are given in the following table:

Table 6. Minimum daily production and demand

	Cream Cheese	Cottage Cheese
Production	1000	1000
Demand	1500	2000

Constraints corresponding to minimum daily production are:

$$M_{11} + M_{21} + C_1 \geq 1000$$

$$.9(M_{12} + M_{22} + C_2) \geq 1000$$

Constraints corresponding to maximum market demand are:

$$M_{11} + M_{21} + C_1 \leq 1500$$

$$.9(M_{12} + M_{22} + C_2) \leq 2000$$

Maximum daily capacity of processing inputs for cheese machine and evaporator are given in the following table:

Table 7. Maximum capacity of machine

	Cheese Machine	Evaporator
Capacity	3000 lb	2000 lb

Constraint corresponding to the maximum capacity of cheese machine

$$(M_{11} + M_{21} + C_1) + (M_{12} + M_{22} + C_2) \leq 3000$$

Constraint corresponding to the maximum capacity of evaporator

$$MC_1 + MC_2 \leq 2000$$

Now,

Total Cost = Costs for purchasing milk + Machine costs + Costs for evaporating milk

$= .8 (M_{11} + M_{12} + MC_1)$

$$+ .4 (M_{21} + M_{22} + MC_2)$$

$$+ .4 \{(M_{11} + M_{21} + C_1)$$

$$+ .9(M_{12} + M_{22} + C_2)\} + .4(MC_1 + MC_2)$$

$=1.2M_{11} + 1.16M_{12} + .8M_{21} + .76M_{22}$

$$+1.2MC_1 + .8MC_2 + .4C_1 + .36C_2$$

Total revenue $= 1.5 (M_{11} + M_{21} + C_1)$

$$+(1.2)(.9)(M_{12} + M_{22} + C_2)$$

$=1.5M_{11} + 1.08M_{12} + 1.5M_{21} + 1.08M_{22}$

$$+1.5C_1 + 1.08C_2$$

Total profit = Total revenue – Total Cost

$= .3M_{11} - .08M_{12} + .7M_{21} + .32M_{22}$

$$-1.2MC_1 - .8MC_2 + 1.1C_1 + .72C_2$$

This is our objective function.

Now rewriting the constraints with the objective function we get the following LP model:

Max $Z = .3M_{11} - .08M_{12} + .7M_{21}$

$+.32M_{22} - 1.2MC_1 - .8MC_2 + 1.1C_1 + .72C_2$

Subject to,

$$.1M_{11} - .2M_{21} \geq 0$$
$$.25M_{12} - .05M_{22} \geq 0$$
$$.6C_1 - .4M_{11} - .4M_{21} \geq 0$$
$$.8C_2 - .2M_{12} - .2M_{22} \geq 0$$
$$-C_1 - C_2 + .6MC_1 + .3MC_2 = 0$$
$$M_{11} + M_{21} + C_1 \geq 1000$$
$$.9 M_{12} + .9M_{22} + .9C_2 \geq 1000$$
$$M_{11} + M_{21} + C_1 \leq 1500$$
$$.9M_{12} + .9M_{22} + .9C_2 \leq 2000$$
$$M_{11} + M_{21} + C_1 + M_{12} + M_{22} + C_2 \leq 3000$$
$$MC_1 + MC_2 \leq 2000$$
$$M_{11}, M_{12}, M_{21}, M_{22}, MC_1, MC_2, C_1, C_2 \geq 0$$

IV. Solution Techniques

After formulating LPs & IPs from real life decision problems it requires to solve them before using the results into real life. There are several computer techniques available. In this paper, we consider AMPL and LINDO.

AMPL

AMPL (A Mathematical Programming Language) is a comprehensive and powerful algebraic modeling language for linear and nonlinear optimization problems, in discrete

or continuous variables. AMPL lets us use common notation and familiar concepts to formulate optimization models and examine solutions, while the computer manages communication with an appropriate solver such as MINOS, CPLEX, lp_solve etc.

AMPL needs a model file, which describes variables, objectives and relationships without referring to specific data. AMPL also needs a data file that specifies parameters. AMPL put the model and data files into an intermediate file that can be read by a solver. The solver actually finds an optimal solution to the problem by reading in the intermediate file produced by AMPL and applying an appropriate algorithm. The solver outputs the solution as a text file.

LINDO

LINDO (Linear, INteractive, and Discrete Optimizer) is a convenient, but powerful tool[9] for solving linear, integer, and quadratic programming problems. LINDO software is designed to be easy to learn and to use, especially for small problems. There are three basic styles of using the LINDO software. For small to medium sized problems, LINDO is simple to use interactively from the keyboard. Entering a model is quite easy to do. It's also possible to use LINDO with externally created files, which contain scripts of commands and input data, to produce files for reporting purposes. Finally, custom-created subroutines may be linked directly with LINDO to form an integrated program containing both our code and the LINDO optimization libraries.

Now, we will solve[5,9,10] the resulting LP model using AMPL and LINDO.

AMPL model file

```
set product;        #two types of cheese produced
set milk;           #two types of milk available

param fat{milk};                   #amount of fat at each type milk
paramleast_fat{product};           #fat requirement at every product
paramleast_cream{product};         #cream requirement at every product
parammaccost{product};             #machine cost per lb of input
paramevap_cost{milk};              #evaporation cost per lb of milk
param waste{product};              #waste per lb of input
paramevap_rate{milk};              #cream per lb of milk when evaporate
paramevap_capacity;                #maximum daily evaporation capacity
parammachine_capacity;             #maximum capacity for cheese machine
paramllimit{product};              #minimum cheese production limit
paramulimit{product};              #minimum cheese production limit
parammcost{milk};                  #price of both types of milk
param price{product};              #selling price of cheese

varcrp{product}>=0;                  #cream per product
varmevap{milk}>=0;                   #milk for evaporator
var amount{milk,product}>=0;         #milk per product

maximize Profit:
sum{m in milk,p in product}price[p]*amount[m,p]*(1-waste[p])
         +sum{p in product}price[p]*crp[p]*(1-waste[p])
-(sum{m in milk,p in product}mcost[m]*amount[m,p]
     +sum{m in milk}mcost[m]*mevap[m]+sum{m inmilk}evap_cost[m]*mevap[m]
      +sum{m in milk,p in product}maccost[p]*amount[m,p]*(1-waste[p])
           +sum{p in product}maccost[p]*crp[p]*(1-waste[p]));
subject to leastfat{p in product}:sum{m in milk}fat[m]*amount[m,p]
>=sum{m in milk}least_fat[p]*amount[m,p];
subject to leastcream{p in product}:crp[p]
>=sum{m in milk}least_cream[p]*amount[m,p]+least_cream[p]*crp[p];
subject to tcream:
sum{m in milk}evap_rate[m]*mevap[m]=sum{p in product}crp[p];
subject to machinecapacity:sum{m in milk,p in product}amount[m,p]
                          +sum{p in product}crp[p]<=machine_capacity;
```

```
subject to evapcapacity:sum{m in milk}mevap[m]<=evap_capacity;
subject to upperlimit{p in product}:
            (sum{m in milk}amount[m,p]+crp[p])*(1-waste[p])<=ulimit[p];
subject to lowerlimit{p in product}:
            (sum{m in milk}amount[m,p]+crp[p])*(1-waste[p])>=llimit[p];
```

AMPL data file

```
set product:=cream_chcottage_ch;
set milk:=high_fatlow_fat;

param     :fat evap_rateevap_costmcost:=
high_fat  .6       .6         .4           .8
low_fat   .3       .3         .4           .4;

param     :least_fatleast_creammaccost waste llimitulimit  price:=
cream_ch      .5           .4       .4      0   1000   1500     1.50
cottage_ch    .35          .2       .4     .1   1000   2000     1.20;
paramevap_capacity:=2000;
parammachine_capacity:=3000;
```

AMPL answer

```
objective -159.2592593

ampl: display crp;
crp [*] :=
cottage_ch  222.222
cream_ch  400;
ampl: display mevap;
mevap [*] :=
high_fat  1037.04
low_fat     0;

ampl: display amount;
amount :=
high_fatcottage_ch   148.148
high_fatcream_ch      400
low_fatcottage_ch    740.741
low_fatcream_ch      200;
```

So, 222.222 lb and 400 lb cream, 148.148 lb and 400 lb high fat milk , 740.741 lb and 200 lb low fat milk needs daily for producing cottage cheese and cream cheese respectively. Also 1037.04 lb high fat milk needs to producing cream through evaporator.

LINDO solver file

```
MAX .3M11-.08M12+.7M21+.32M22-1.2MC1-.8MC2+1.1C1+.72C2
ST
.1M11-.2M21>0
.25M12-.05M22>0
.6C1-.4M11-.4M21>0
.8C2-.2M12-.2M22>0
-C1-C2+.6MC1+.3MC2=0
M11+M21+C1>1000
.9M12+.9M22+.9C2>1000
M11+M21+C1<1500
.9M12+.9M22+.9C2<2000
M11+M21+C1+M12+M22+C2<3000
```

```
MC1+MC2<2000
END
```
LINDO answer
```
OBJECTIVE FUNCTION VALUE
  1)    -159.2593
 VARIABLE        VALUE          REDUCED COST
      M11        400.000000          0.000000
      M12        148.148148          0.000000
      M21        200.000000          0.000000
      M22        740.740723          0.000000
      MC1       1037.036987          0.000000
      MC2          0.000000          0.200000
       C1        400.000000          0.000000
       C2        222.222229          0.000000
```

So, 222.222 lb and 400 lb cream, 148.148 lb and 400 lb high fat milk , 740.741 lb and 200 lb low fat milk needs daily for producing cottage cheese and cream cheese respectively. And 1037.04 lb high fat milk needs to producing cream through evaporator.

Comparison

In the previous section, we solved LP & IP model with AMPL and LINDO. Now we will present some

comparison[1] between them.

1. LINDO allows us to enter a model in a straight forward algebraic way which AMPL doesn't. But this is very useful for solving small model.

2. With AMPL using display command we can see the resulting value of same kind of decision variable separately which is very useful for further analysis. But LINDO doesn't have this feature.

3. With AMPL using expand command we can see the whole LP problem in terms of their original name of decision variables. But with LINDO it is not possible.

4. AMPL is very useful when we want to use the model and `data` file for similar kind of problems with different parameters. But LINDO is well behind in this section.

5. There are different solver for AMPL which are used for different kinds of optimization problem. But, LINDO doesn't need any other solver.

Both of the AMPL and LINDO are very useful computer techniques for solving LP & IP models. But AMPL's flexibility and convenience render it ideal for rapid prototyping and model development, while its speed and control options make it an especially efficient choice for repeated production runs.

V. Conclusion

In this paper, we have discussed about the basics of LP and IP and detail techniques of formulating LP & IP models from real life decision problems. We have represented formulating LP & IP model from real life problem as an art not just science. For most of the real life models with large number of decision variables & constraints it is quiet hard to solve by hand calculations. So, we have used AMPL & LINDO for solving resulting LP and IP problems. Also we have presented a comparison between those two computer techniques. Both of them are very user friendly and useful.

References

1. Dantzig, G.B., 2002. Linear Programming, *Operations Research*, **50** (1), 42–47.

2. Dantzig, G.B., 1947. Maximization of a linear function of variables subject to linear inequalities, Published pp. 339–347 in T.C. Koopmans (ed.): Activity Analysis of Production and Allocation, New York-London 1951 (Wiley & Chapman-Hall).

3. Eiselt, H. A. and Sandblom, C.-L., 2007. Linear Programming and its Applications, Springer.

4. Taha, Hamdy A., 2007. Operations Research: An introduction, 8th Ed. Pearson Princeton hall.

5. Fourer, Robert, Gay, David M. and Kernighan, Brian W., 2003. AMPL: A Modeling Language for Mathematical Programming, 2nd Ed. Thomson/Brooks/ Cole.

6. Braitsch, Raymond J., 1972. A computer Comparison of four Quadratic Programming Algorithms, *Management Science*, **18** (11).

7. Hossein, Arsham, Deterministic Modeling: Linear ptimization with Applications, http://home.ubalt.edu/ ntsbarsh/opre640a/ partVIII.html [accessed 03/02/2012].

8. Winston, Wayne L. and Goldberg, Jeffrey B., 2004. Operations Research: Applications and Algorithms, 4th Ed. Thomson Brooks/Cole.

9. LINDO User's Manual, 2003. LINDO SYSTEMS INC, Chicago.

10. Schrage, Linus E., 1997. Optimization modeling with LINDO, 5th Ed. Duxbury Press.