Solving Transportation Models with Spreadsheet Software

Dr. Steven C. Ross, Management Information Systems, Western Washington University
Dr. Richard L. Schwaller, Management Science, Montana State University

Abstract

This paper describes the solution of transportation models on personal computers (PCs) using spreadsheet and other readily-available software. We choose PCs because of their commonness in university colleges of business and in organizations of all size. The models discussed include a simple transportation model and more complex models involving quantity discounts and multiple brands. The techniques are applicable to other linear programming situations. To facilitate their use, detailed descriptions of both model formulation and the use of the optimizing software are provided.

Introduction

This paper describes the solution of transportation models on personal computers (PCs) using spreadsheet and other readily-available software. Although the specific models discussed have appeared elsewhere in the management science literature, the linear programming (LP) formulations of the models and the discussion of these techniques with personal computers is uncommon.

We choose PCs because of their commonness in university colleges of business and in organizations of all size. One of the forces resisting the greater adoption of management science models by practitioners is that such models tend to require sophisticated computer systems for computation. Models that run on PCs, however, are more readily accessible and therefore more likely to be used in an organization. A similar dynamic exists in universities: often the mainframes are "too much hassle" for use of such models whereas a PC-based system might not be.

We use spreadsheet software for similar reasons. Formulating input for many mainframe and even many PC-based software packages is an intimidating task. Although the input is economical in terms of data items, it fails to have an easily-understood structure. In a spreadsheet, however, the user is presented with a visual display of the relationship between various factors, a reasonably direct input method, and the ability to further analyze a model once the optimal solution has been found. Entry of the model is also facilitated by spreadsheet capabilities such as pointing to cells during formula entry and the ability to copy formulas to create multiple similar entries.

The examples presented in this paper include a simple transportation model, a more complex transportation model that includes quantity discounts, and a multiple-brands problem. The techniques are applicable to other LP situations.

This paper is oriented toward both education and practice. Where PCs and spreadsheet software are readily available, and users or students have some elementary training in their use, the techniques we illustrate can be adopted with minimal time and cost. Educators will find a technique that many of them can readily adapt to their courses. Practicing managers will find a technique that they can use in their office, freeing them from the need to find a computer system containing LP software.

We hope that readers of this paper will make use of the techniques discussed in university courses and professional seminars. To facilitate their use, detailed descriptions of model formulation and use of the optimizing software are provided. We assume that readers of this paper are familiar with the use of PCs in general and spreadsheet software in particular. Lotus 1-2-3 Release 2.01 is the specific package discussed here. The optimization software used is What's-*Best!* Academic Version 1.2 published by Holden-Day (Savage, 1986). This is available for approximately $30 to students. The professional version, a product of General Optimization, costs approximately $1000.

Spreadsheets as a Vehicle for Formulating and Analyzing Optimization Problems

Spreadsheets are a very useful vehicle for formulating and analyzing optimization problems for several reasons. First, they are a familiar "programming medium" for many managers and students (probably the *most* familiar programming medium). Second, once the solution is determined, spreadsheets provide an ideal vehicle for further analysis. Further analysis can include analyzing dual values that are part of the model solution, graphics portraying the solution and alternatives, and analysis of competing sub-optimal solutions.

The What's*Best!* Program

What's*Best!* is a *terminate and stay resident* program. This places it in the same category as desk-top aids such as SideKick and keyboard enhancers such as SuperKey. When the program is first executed, before entering the spreadsheet software, it occupies a portion of the random access memory and then returns control to the DOS prompt. What's*Best!* is called from within the spreadsheet program to formulate and solve an LP problem.

Use of Spreadsheet Features to Identify Parameters

What's*Best!* makes rather clever use of spreadsheet features to mark cells as continuous variables, 0-1 integer variables, the objective function to maximize or minimize, and the nature of the constraints. Continuous variables are marked as unprotected cells. 0-1 integers are formatted to + (bar graph) format. The cell containing the objective function is named either WBMAX or WBMIN. Constraints are marked with symbols (<, =, and >). The program examines constraints by computing slack values: a negative slack value indicates a violated constraint and therefore an infeasible solution. All other cells must also return a positive value unless they are included in a range whose name starts with WBFREE (e.g., WBFREE1, WBFREE2, etc.).

Specifying the Model

What's*Best!* simplifies the concepts as much as possible for the spreadsheet user. Simple models are formulated and described using an ABC mnemonic:

A -- Adjustable Cells.
B -- Best (Objective Function).
C -- Constraints.

These three facets of the formulation are identified to What's*Best!* using the *Program to Spreadsheet Connection* (PrtSc) key. Press the PrtSc key (*not* Shift-PrtSc), and the What's*Best!* menu appears on the spreadsheet (see Figure 1). Press F3 to mark a cell or range of cells as adjustable; or F4 to mark cells as fixed. The normal state of cells is fixed (protected). Indicate 0-1 integer adjustable cells with F8. To mark the objective function, move the cell pointer to that cell, then press PrtSc followed by either F5 (maximize) or F6 (minimize). Constraints require that a formula be placed in one cell and the constant (right-hand side value) be placed two cells to the right. Place the cell pointer between the two entries, press PrtSc, then one of <, >, or =. (Not all options are displayed on the pop-up menu.) Once ABC are completed, press PrtSc followed by F1 to compute the solution. The program saves the file (as WBTO.WK1), exits Lotus, computes the optimal solution, re-enters Lotus, and retrieves the

FIGURE 1 The What'sBest! Pop-up Menu

What'sBest! Commands			
Optimize	F1	F2	
Adjustable Cell	F3	F4	Fixed Cell
Maximize	F5	F6	Minimize
Dual Value	F7	F8	Integer
Extension	F9	F10	Options
Less or Equal	<	>	Greater or Equal

optimized worksheet (named WBFR.WK1).

Formulas and Functions

What's*Best!* places certain restrictions on the formulas and spreadsheet functions used in the model. (The terms *formula* and *function* are used differently in spreadsheet and mathematical contexts. In spreadsheet terms, a formula is a mathematical expression or function entered into a cell, a function is a sub-program that returns a result.) All formulas depending directly or indirectly on adjustable cells must be linear:

- Adjustable cells can be multiplied or divided by constants (cells with fixed values).
- Formulas can contain addition, subtraction, or the @SUM function.
- Formulas that do not depend on adjustable cells can be formed by addition, subtraction, multiplication, division, or the @SUM function.

Other functions are not supported by What's*Best!*. Some functions, such as @MAX, can be used, but What's*Best!* does not recalculate the function as it computes the optimal solution. The value of the function at the time the optimization routine is called remains fixed. Other functions, such as @IF, usually cause the program to fail.

The Simple Transportation Model

Formulation

The simple transportation model assumes that a product is transported from a number of sources to a number of destinations and that there is a shipping cost per unit associated with each source-destination pair. Each source has a fixed number of units available for shipment and there is a fixed demand for the product at each of the destinations. The number of units sent from a given source to a given destination is called the shipping assignment for that pair. The goal of the problem is to determine the shipping assignments that will minimize the total cost of transportation.

Several algorithms for approximating or solving the problem are found in Management Science and Production and Operations Management texts. Some of these are the northwest corner rule, Vogel's approximation, the stepping-stone method, and the MODI method (Eppen & Gould, 1984). In this paper, however, we do not focus upon such methods but rather formulate the problem as a linear program on a spreadsheet and find the solution using a spreadsheet add-in program.

The LP formulation for the simple transportation model can be expressed as follows: Choose values of X_{ij} to minimize

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} X_{ij} \qquad (1)$$

where m is the number of sources, n is the number of destinations, c_{ij} is the per unit cost of shipping the product from source i to destination j, and X_{ij} is the number of units sent from source i to destination j. The constraints are:

$$\sum_{j=1}^{n} X_{ij} \begin{cases} \leq a_i & \text{if } \sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \\ = a_i & \text{otherwise} \end{cases} \qquad (i=1,2,\ldots,m) \qquad (2)$$

$$\sum_{i=1}^{m} X_{ij} \begin{cases} \leq b_j & \text{if } \sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \\ = b_j & \text{otherwise} \end{cases} \qquad (j=1,2,\ldots,n) \qquad (3)$$

where there are a_i units available at source i and b_j units ordered from destination j.

Specific Problem

To illustrate the simple formulation, we consider the situation portrayed in Table 1. Here we have two sources, two destinations, and a shipping cost for each source-destination pair.

TABLE 1 The Simple Transportation Problem

Supply		Demand		Shipping Cost/Unit	
Source 1	600	Dest. 1	400	From S1 to D1	13
Source 2	400	Dest. 2	500	From S1 to D2	13
				From S2 to D1	14
				From S2 to D2	11

Spreadsheet Entry

Entry of this problem into a spreadsheet model is straightforward (see Figure 2). Entries in the range B4..B7 are cost coefficients (c_{ij}) from Table 1. The range C4..C7 contains the variables (X_{ij}). The variables are initially set to 0 (zero), but will be manipulated by What's*Best!* for the solution. The objective function is entered in cell D5: +C4*B4+C5*B5+C6*B6+C7*B7.

Constraints are entered in the lower half of the spreadsheet. (This is purely cosmetic; the spatial arrangement is of no consequence to What's*Best!*) The supply values for sources 1 and 2 (a_i) are entered in D11 and D12. The demand values for destinations 1 and 2 (b_j) are entered in D14 and D15. Formulas for total quantity shipped or received (ΣX) are entered in column B. For instance, +C4+C5 is entered in B11 and +C4+C6 in B14.

FIGURE 2 Initial Spreadsheet Entries

	A	B	C	D
1	SIMPLE MODEL, NO QUANTITY DISCOUNTS			
2				
3	Source, Dest.	Cost	Qty.	TOTAL
4	1,1	13	0	COST
5	1,2	13	0	0
6	2,1	14	0	
7	2,2	11	0	
8				
9	Constraints			
10	Source	Qty.		Supply
11	1	0		600
12	2	0		400
13	Dest.	Qty.		Demand
14	1	0		400
15	2	0		500

Setup with What'sBest!

Once the initial entries are in place, What's*Best!* will assist in identifying variables, objective function, and constraints. We next do the following:

- **Adjustable**: Adjustable cells are indicated by moving the cell pointer to C4, pressing PrtSc followed by F3, highlighting the range C4..C7, and terminating the process by pressing Enter. What's*Best!* removes cell protection from these cells.
- **Best**: Move the cell pointer to the cell containing the objective function (D5 in this case), press the PrtSc key, and indicate that the current cell is to be minimized by pressing the F6 function key. What's*Best!* names the cell WBMIN.
- **Constraints**: Identify supply constraints by moving the cell pointer to C11, then pressing PrtSc followed by <. What's*Best!* enters the < symbol in C11 and a formula, +D11-B11, in E11. When this formula has a value ≥ 0, the constraint is satisfied. The process is repeated in C12.
- Finally, identify demand constraints by moving the cell pointer to C14, then pressing PrtSc followed by =. (For this illustration, in which total supply exceeds total demand, we specify demand as a constraint that must be met exactly, no more and no less.) What's*Best!* enters the = symbol in C14 and formulas in E14 and F14: +B14-D14 in E14 and -E14 in F14. Only when both formulas have the value 0 is the equality constraint satisfied. The process is repeated in C15.
- After adding the label "Slack," the spreadsheet now looks like that illustrated in Figure 3.

FIGURE 3 After What's*Best!* Setup

	A	B	C	D	E	F
1	SIMPLE MODEL, NO QUANTITY DISCOUNTS					
2						
3	Source, Dest.	Cost	Qty.	TOTAL		
4	1,1	13	0	COST		
5	1,2	13	0	0		
6	2,1	14	0			
7	2,2	11	0			
8						
9	Constraints					
10	Source	Qty.		Supply	Slack	
11	1	0	<	600	600	
12	2	0	<	400	400	
13	Dest.	Qty.		Demand		
14	1	0	=	400	-400	400
15	2	0	=	500	-500	500

FIGURE 4 The What's*Best!* Solution

	A	B	C	D	E	F
1	SIMPLE MODEL, NO QUANTITY DISCOUNTS					
2						
3	Source, Dest.	Cost	Qty.	TOTAL		
4	1,1	13	400	COST		
5	1,2	13	100	10900		
6	2,1	14	0			
7	2,2	11	400			
8						
9	Constraints					
10	Source	Qty.		Supply	Slack	
11	1	500	<	600	100	
12	2	400	<	400	0	
13	Dest.	Qty.		Demand		
14	1	400	=	400	0	0
15	2	500	=	500	0	0

Solution

Once the setup steps are completed, solution with What's*Best!* is readily accomplished. Press PrtSc, then F1. The file is saved and Lotus exited. Summary information is displayed during optimization. The process is quite fast. On a PC/XT the completed solution (Figure 4) appears less than one minute after pressing F1. On a 386 machine, this problem is solved in less than 10 seconds. As a validity check, this solution was also obtained with LINDO on a VAX computer.

Transportation Model with Quantity Discounts

Formulation

The simple model is now extended to the case where the unit shipping cost for each source-destination pair is a discrete function of the number of units shipped. We will assume that the quantities required or provided are integer valued. If the shipping cost functions associated with the pairs are non-decreasing, the problem is the transportation problem with quantity discounts (Beale, 1959). A description of the quantity discount problem and the LP formulation that provides the solution follows.

The parameters m, n, a_i, and b_j are as defined in the simple model described earlier. For the (i,j)th source-destination pair, the number of distinct shipping costs will be denoted by K(i,j). With k indexing the K(i,j) cases, X_{ijk} and c_{ijk} will denote the amount to be shipped and the corresponding unit shipping cost for the kth case. The values that the shipping quantity, Q, takes on along with the corresponding cost breaks at Q_{ijk} (the breakpoints), and the corresponding unit costs are shown in Table 2.

TABLE 2 Relationship among Quantity Shipped, Per Unit Shipping Cost, and Breakpoints

Value of Q	Unit cost	Interval
X_{ij1}	c_{ij1}	$0 \leq Q \leq Q_{ij1} - 1$
X_{ij2}	c_{ij2}	$Q_{ij1} \leq Q \leq Q_{ij2} - 1$
.	.	.
.	.	.
.	.	.
$X_{ijK(i,j)}$	$c_{ijK(i,j)}$	$Q_{ij[K(i,j)-1]} \leq Q$

The objective function to minimize is:

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K(i,j)} c_{ijk} X_{ijk} \qquad (4)$$

The first set of constraints, which depend on the a_i's, are the *supply constraints*:

$$\sum_{j=1}^{n} \sum_{k=1}^{K(i,j)} X_{ijk} \begin{cases} \leq a_i & \text{if } \sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \\ = a_i & \text{otherwise} \end{cases} \qquad (i=1,2,\ldots,m) \qquad (5)$$

The next constraints, which depend on the b_j's, are the *demand constraints*:

$$\sum_{i=1}^{m} \sum_{k=1}^{K(i,j)} X_{ijk} \begin{cases} \leq b_j & \text{if } \sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j \\ = b_j & \text{otherwise} \end{cases} \qquad (j=1,2,\ldots,n) \qquad (6)$$

The supply constraints (5) ensure that the amounts shipped from the sources will not exceed the demand when supply exceeds the demand. The demand constraints (6) ensure that when demand exceeds supply, the total amount received at the destinations will be the total supply available, although the total demand cannot be met. These constraints do not, however, ensure that X_{ijk} will lie within the interval bounded by the breakpoints shown in Table 2. When $k = 1$, this requirement is met by constraints of the form

$$X_{ij1} \leq Q_{ij1} - 1 \qquad (1 \leq i \leq m,\ 1 \leq j \leq n) \qquad (7)$$

For $k > 1$, constraints are needed that will ensure that when X_{ijk} is positive, X_{ijk} will lie within the interval that does not include zero and that X_{ijk} will equal zero when it does not lie within the interval. To meet this requirement, constraints will be included which together, satisfy inequalities of the form

$$Q_{ij(k-1)} Z_{ijk} \leq X_{ijk} \leq (Q_{ijk} - 1) Z_{ijk} \qquad (8)$$

where the Z_{ijk}'s are integer 0-1 variables. When Z_{ijk} is 0, the corresponding value of X_{ijk} is also 0. However, when Z_{ijk} is 1, the corresponding X_{ijk} will lie within the interval bounded by the breakpoints. Two constraints are entered in the formulation for each inequality of the form (8). These constraints are

$$X_{ijk} - Q_{ij(k-1)} Z_{ijk} \geq 0 \qquad (1 \leq i \leq m,\ 1 \leq j \leq n,\ 2 \leq k \leq K(i,j)) \qquad (9)$$

$$X_{ijk} - (Q_{ijk}-1)Z_{ijk} \leq 0 \qquad (1\leq i\leq m,\ 1\leq j\leq n,\ 2\leq k\leq K(i,j)-1) \qquad (10)$$

$$X_{ijk} - a_i Z_{ijk} \leq 0 \qquad (1\leq i\leq m,\ 1\leq j\leq n,\ k=K(i,j)) \qquad (11)$$

Constraint (9) checks for lower bounds, (10) checks for upper bounds. The last constraint (11) is needed to ensure that $X_{ijK(i,j)}$ will be 0 when $Z_{ijK(i,j)}$ is 0. The coefficient of $Z_{ijK(i,j)}$ could be any number that is at least as large as $\min\{a_i,b_j\}$.

Specific Problem

To illustrate the model with quantity discounts, we extend the situation of the previous problem. See Table 3. In addition to two sources and two destinations, we now have two or three shipping costs depending on quantity shipped.

TABLE 3 The Transportation Problem with Quantity Discounts

Supply		Demand		Shipping Cost/Unit (min. qty.)			
Source 1	600	Dest. 1	400	From S1 to D1	13 (0)	12 (200)	10 (300)
Source 2	400	Dest. 2	500	From S1 to D2	13 (0)	11 (150)	10 (250)
				From S2 to D1	14 (0)	10 (220)	9 (300)
				From S2 to D2	11 (0)	8 (200)	

Spreadsheet Entry

The spreadsheet developed for the first model serves as the basis for this model. Figure 5 contains the spreadsheet *after* What's*Best!* setup has been completed. In the upper portion, rows are added for each source, destination, quantity combination. Columns are added for the 0-1 variable (this column is titled "Used"), the upper limit of the cost bracket ("Brkpt."), and the extension ("Ext.").

Zeros are entered under Used in rows where k > 1. Formatting accomplished by What's*Best!* setup makes these appear as periods in Figure 5.

Under Brkpt., enter the minimum quantity for the next higher bracket (e.g., 200 is entered in E4). The entry for the maximum quantity of the lowest-cost/largest-quantity bracket is the quantity available at the source (e.g., +D18 is entered in E6).

Under Ext., enter the product of Cost and Qty. (e.g., +C4*B4 in F4).

The objective function, under TOTAL COST, is the sum of the extensions: @SUM(F4..F14).

Source and Destination constraints are the same as in the simple model.

Lower Bounds constraints must be specified for all quantity levels except the highest-cost/smallest-quantity brackets (which are forced to be greater than or equal to zero in any event). The formula in B24 is +C5-E4*D5, and the value 0 (zero) is entered in D24. These entries accomplish constraint (9). (Entries in columns C and E are accomplished during What's*Best!* setup.)

Upper Bounds constraints are specified for all quantity levels. The formula in B32 is +C4-(E4-1), and the value 0 (zero) is entered in D24; this corresponds to (7). The entry +C5-(E5-1)*D5 in B33 corresponds to (10). Formulas in the lowest-cost/largest quantity brackets (11) are slightly different: -+C6-E6*D6 is entered in B34.

Setup with What'sBest!

This generally follows the procedures for the simple model, with some changes to accommodate the 0-1 integers and the larger number of constraints:

- **Adjustable:** Cells in the range C4..D14 are identified as adjustable using the PrtSc/F3 combination.
- The 0-1 Integer adjustable cells (D4..D14) are identified with the PrtSc/F8 combination.
- **Best:** G5 is identified as the cell to be minimized with the PrtSc/F6 combination.
- **Constraints:** Supply and Demand constraints are identified as before.
- The first Lower Bounds constraint is identified by placing the cell pointer in C24, then using the PrtSc/> combination. What's*Best!* enters the formula for slack in E24. To save time, /Copy the symbol from C24 to the range C25..C30, and /Copy the formula from E24 to the range E25..E30.
- The first Upper Bounds constraint is identified by placing the cell pointer in C32, then using the PrtSc/< combination. What's*Best!* enters the formula for slack in E32. /Copy the symbol from C32 to the range C33..C42, and /Copy the formula from E32 to the range E33..E42. Because values in the range C32..C42 can be negative (when some quantity other than the upper limit is shipped), this range is named WBFREE1 to signal What's*Best!* that negative values are acceptable.

At this point, the spreadsheet looks like Figure 5.

Solution

As before, solution with What's*Best!* is readily, but not so quickly, accomplished. The solution is illustrated in Figure 6. Solution on a PC/XT required more than eight minutes. On the 386 machine, approximately 20 seconds were required. This solution was also verified with LINDO. In column D, + marks indicate the value 1 for the 0-1 integer variables. As you see, the 0-1 variables have the value 1 wherever the quantity shipped is greater than zero.

The Multiple Brands Problem

Thus far in this paper, we have discussed two transportation problems: one with quantity discounts and one without. When the formulation presented for the quantity discount model is used to solve the transportation problem, the solution will contain at most one positive value of X_{ijk} for any source-destination pair. With minor changes, the formulation can be used to solve other kinds of problems in which the solution can contain more than one positive value of X_{ijk} for a supply-demand pair. The multiple brands problem is one such case.

Formulation

In the multiple brands problem, a buyer (destination) can make purchases of K different brands of a product from a vendor (source). Each of the brands meets the product specifications of the buyer.

FIGURE 5 After What's*Best!* Setup

	A	B	C	D	E	F	G
1	MODEL WITH QUANTITY DISCOUNTS						
2							
3	Source, Dest., Level	Cost	Qty. Used		Brkpt.	Ext.	TOTAL
4	1,1,1	13	0		200	0	COST
5	1,1,2	12	0 .		300	0	0
6	1,1,3	10	0 .		600	0	
7	1,2,1	13	0		150	0	
8	1,2,2	11	0 .		250	0	
9	1,2,3	10	0 .		600	0	
10	2,1,1	14	0		220	0	
11	2,1,2	10	0 .		300	0	
12	2,1,3	9	0 .		400	0	
13	2,2,1	11	0		200	0	
14	2,2,2	8	0 .		400	0	
15							
16	Constraints						
17	Source	Qty.		Supply	Slack		
18	1	0	<	600	600		
19	2	0	<	400	400		
20	Dest.	Qty.		Demand			
21	1	0	=	400	-400	400	
22	2	0	=	500	-500	500	
23	Lower Bounds						
24	1,1,2	0	>	0	0		
25	1,1,3	0	>	0	0		
26	1,2,2	0	>	0	0		
27	1,2,3	0	>	0	0		
28	2,1,2	0	>	0	0		
29	2,1,3	0	>	0	0		
30	2,2,2	0	>	0	0		
31	Upper Bounds						
32	1,1,1	-199	<	0	199		
33	1,1,2	0	<	0	0		
34	1,1,3	0	<	0	0		
35	1,2,1	-149	<	0	149		
36	1,2,2	0	<	0	0		
37	1,2,3	0	<	0	0		
38	2,1,1	-219	<	0	219		
39	2,1,2	0	<	0	0		
40	2,1,3	0	<	0	0		
41	2,2,1	-199	<	0	199		
42	2,2,2	0	<	0	0		

FIGURE 6 The What's*Best!* Solution

	A	B	C	D	E	F	G
1	MODEL WITH QUANTITY DISCOUNTS						
2							
3	Source, Dest., Level	Cost	Qty. Used		Brkpt.	Ext.	TOTAL
4	1,1,1	13	0		200	0	COST
5	1,1,2	12	0 .		300	0	8450
6	1,1,3	10	400 +		600	4000	
7	1,2,1	13	0		150	0	
8	1,2,2	11	150 +		250	1650	
9	1,2,3	10	0 .		600	0	
10	2,1,1	14	0		220	0	
11	2,1,2	10	0 .		300	0	
12	2,1,3	9	0 .		400	0	
13	2,2,1	11	0		200	0	
14	2,2,2	8	350 +		400	2800	

There is a total cost per unit which is the sum of a purchase price and a shipping cost, and this cost varies with both the brand and the vendor-buyer pair. Consider the following situation:

The buyer is a government agency that intends to replace its fleet of vehicles at several locations. The buyer sets specifications for the vehicles desired and will accept any brand that meets these specifications. The vendors are dealers who submit bids that include the purchase price and the cost of delivery. Since a large number of vehicles are involved and they can be shipped most economically in trailers containing several vehicles, the bid per unit requires that a minimum number of units of a given brand be purchased and sent to a given location. Other limitations on the problem are the maximum number of vehicles the dealer has available and also the maximum number of each brand that can be sent to each location. The goal of the problem is to minimize the total cost of replenishing the fleet of vehicles.

The assumptions for this model are as follows:

- The buyer will accept any of the brands offered since each brand meets product specifications.
- There is a minimum and maximum amount of each brand available for shipment from each vendor to each buyer location.
- The demand of each buyer location will be at least as large as the minimum quantity of the brands available from the vendors.

Notation:

X_{ijk} = the number of units of brand k sent from vendor i to buyer location j.
Z_{ijk} = a 0-1 integer variable
c_{ijk} = the cost of providing buyer location j with one unit of brand k from vendor i.
M_{ijk} = the maximum number of units of brand k available for shipment from vendor i to buyer location j.
L_{ijk} = the minimum number of units of brand k available for shipment from vendor i to buyer location j.
a_{ik} = the number of units of brand k available from vendor i.
b_j = the number of units ordered by buyer location j.
$S(i)$ = a set listing the brands available from vendor i.
m = the number of vendors.
n = the number of buyer locations.

The LP formulation is as follows. Minimize

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k \in S(i)} c_{ijk} X_{ijk} \qquad (12)$$

subject to

$$\sum_{j=1}^{n} X_{ijk} \begin{cases} \leq a_{ik} & \text{if } \sum_{i=1}^{m} \sum_{k \in S(i)} a_{ik} > \sum_{j=1}^{n} b_j \\ = a_{ik} & \text{otherwise} \end{cases} \quad (i=1,2,\ldots,m,\ k \in S(i)) \qquad (13)$$

$$\sum_{i=1}^{m}\sum_{k\in S(i)} X_{ijk} \begin{cases} \leq b_j & \text{if } \sum_{i=1}^{m}\sum_{k\in S(i)} a_{ik} < \sum_{j=1}^{n} b_j \\ & (j=1,2,\ldots,n) \\ = b_j & \text{otherwise} \end{cases} \quad (14)$$

$$X_{ijk} - L_{ijk}Z_{ijk} \geq 0 \qquad (1\leq i\leq m,\ 1\leq j\leq n,\ k\in S(i)) \quad (15)$$

$$X_{ijk} - M_{ijk}Z_{ijk} \leq 0 \qquad (1\leq i\leq m,\ 1\leq j\leq n,\ k\in S(i)) \quad (16)$$

If $L_{ijk} = 0$, (15) and (16) can be simplified to

$$X_{ijk} \leq M_{ijk} \quad (17)$$

Specific Problem

A government agency maintains a fleet of vehicles in three different locations. There are 85 vehicles at location 1, 70 at location 2, and 120 at location 3. All the present vehicles are to be replaced with new ones that meet government specifications. The specifications are met by just three different brands. After advertising a call for bids, two dealers submit bids that are in the form of unit costs with delivery included and that apply only if the order size for the location is sufficiently large. There is also a maximum number of each brand available from each dealer and a maximum number that the dealer can send to a given location. Finally, the first dealer can provide vehicles of all three brands but the second dealer can only provide vehicles of the first two brands. The data are shown in Table 4.

Spreadsheet Entry

This problem is entered into a spreadsheet as illustrated in Figure 7. Supply and demand constraints are entered as before. For instance, the entry in B24 is +E5+E8+E11, adding quantities of Brand 1 shipped from Source 1 to all destinations.

For most vendor, buyer, brand combinations, both lower and upper bound constraints, as well as Zs, are needed. For instance, cell B34 contains the formula +E5-B5*F5 and B50 contains +E5-C5*F5. Where the minimum quantity is 0, neither lower bound constraints nor Zs are needed (rows 9, 18, 38, and 47). The upper bound constraints are less complex: B54 contains +E9-C9.

Setup with What's*Best!*:

- **Adjustable:** E5..F19.
- **Best:** Minimize D1.
- **Constraints:** in rows 24 through 64, as illustrated in Figure 7.

Solution. The solution is illustrated in Figure 8.

TABLE 4 The Multiple Brands Problem

Vendor, Brand, Supply			Demand at Buyer Location	
Vendor 1,	Brand 1	55	Location 1	85
	Brand 2	80	Location 2	70
	Brand 3	60	Location 3	120
Vendor 2,	Brand 1	65		
	Brand 2	75		

Vendor, Buyer Location, Brand	Minimum	Maximum	Cost (in $000)
Vendor 1, Location 1, Brand 1	15	27	9
1, 1, 2	5	20	11
1, 1, 3	10	30	15
1, 2, 1	7	18	12
1, 2, 2	0	20	9
1, 2, 3	20	35	11
1, 3, 1	12	30	10
1, 3, 2	10	40	12
1, 3, 3	5	10	8
2, 1, 1	20	35	10
2, 1, 2	14	30	8
2, 2, 1	5	10	12
2, 2, 2	10	30	10
2, 3, 1	0	20	11
2, 3, 2	15	40	10

Additional Features of What's*Best!*

In this paper we have demonstrated the use of What's*Best!* to solve transportation problems. What's*Best!* provides other features useful in these and other LP situations. These are mentioned here, but not discussed or demonstrated, due to space limits. The program can calculate **Dual Values** and the **Upper and Lower Limits** of the effective range. It can display a graph of the objective function value during optimization. The **Extension Feature** aids in the entry of the objective function. The formulation can be output in a form readable by LINDO. The number of iterations can be limited. And, the program can warn the user of multiple optima and indicate adjustable values that would be non-zero at those optima.

Conclusion

The use of specialized add-in software such as the What's*Best!* program greatly extends the utility of commonly used tools such as spreadsheets and PCs. It is important that academicians and practitioners develop formulations such as those presented in this paper to realize the potential of the combination.

References

1 Beale, E.M.L., "An Algorithm for Solving the Transportation Problem When the Shipping Cost Over Each Route is Convex," *Naval Research Logistics Quarterly*, Vol. 6, No.1, pp. 43-56, 1959.

2 Eppen, G.D. and F.J. Gould, *Introductory Management Science*, Prentice-Hall, Englewood Cliffs, N.J., 1984.

3 Savage, S.L., *The ABC's of Optimization using* What's*Best!*, reprinted by Holden-Day, Oakland, Calif., 1986.

FIGURE 7 After What's*Best!* Setup

	A	B	C	D	E	F
1	MULTIPLE BRANDS PROBLEM			$0	TOTAL COST	
2						
3	TABLE OF COSTS				Qty.	
4	Vendor, Buyer, Brand	Minimum	Maximum	Cost	Ship. Z	
5	1,1,1	15	27	9	0 .	
6	1,1,2	5	20	11	0 .	
7	1,1,3	10	30	15	0 .	
8	1,2,1	7	18	12	0 .	
9	1,2,2	0	20	9	0	
10	1,2,3	20	35	11	0 .	
11	1,3,1	12	30	10	0 .	
12	1,3,2	10	40	12	0 .	
13	1,3,3	5	10	8	0 .	
14	2,1,1	20	35	10	0 .	
15	2,1,2	14	30	8	0 .	
16	2,2,1	5	10	12	0 .	
17	2,2,2	10	30	10	0 .	
18	2,3,1	0	20	11	0	
19	2,3,2	15	40	10	0 .	
20						
21						
22	CONSTRAINTS					
23	Supply: Source, Brand			Limit	Slack	
24	1,j,1	0	<	55	55	
25	1,j,2	0	<	80	80	
26	1,j,3	0	<	60	60	
27	2,j,1	0	<	65	65	
28	2,j,2	0	<	75	75	
29	Demand: Buyer Location					
30	i,1,k	0	=	85	-85	85
31	1,2,k	0	=	70	-70	70
32	i,3,k	0	=	120	-120	120
33	Lower Bounds					
34	1,1,1	0	>	0	0	
35	1,1,2	0	>	0	0	
36	1,1,3	0	>	0	0	
37	1,2,1	0	>	0	0	
38	1,2,2	no need				
39	1,2,3	0	>	0	0	
40	1,3,1	0	>	0	0	
41	1,3,2	0	>	0	0	
42	1,3,3	0	>	0	0	
43	2,1,1	0	>	0	0	
44	2,1,2	0	>	0	0	
45	2,2,1	0	>	0	0	
46	2,2,2	0	>	0	0	
47	2,3,1	no need				
48	2,3,2	0	>	0	0	
49	Upper Bounds					
50	1,1,1	0	<	0	0	
51	1,1,2	0	<	0	0	
52	1,1,3	0	<	0	0	
53	1,2,1	0	<	0	0	
54	1,2,2	-20	<	0	20	
55	1,2,3	0	<	0	0	
56	1,3,1	0	<	0	0	
57	1,3,2	0	<	0	0	
58	1,3,3	0	<	0	0	
59	2,1,1	0	<	0	0	
60	2,1,2	0	<	0	0	
61	2,2,1	0	<	0	0	
62	2,2,2	0	<	0	0	
63	2,3,1	-20	<	0	20	
64	2,3,2	0	<	0	0	

FIGURE 8 The What's*Best!* Solution

	A	B	C	D	E		F
1	MULTIPLE BRANDS PROBLEM			$2,740	TOTAL COST		
2							
3	TABLE OF COSTS				Qty.		
4	Vendor, Buyer, Brand	Minimum	Maximum	Cost	Ship.	Z	
5	1,1,1	15	27	9	25	+	
6	1,1,2	5	20	11	0	.	
7	1,1,3	10	30	15	0	.	
8	1,2,1	7	18	12	0	.	
9	1,2,2	0	20	9	20		
10	1,2,3	20	35	11	35	+	
11	1,3,1	12	30	10	30	+	
12	1,3,2	10	40	12	25	+	
13	1,3,3	5	10	8	10	+	
14	2,1,1	20	35	10	30	+	
15	2,1,2	14	30	8	30	+	
16	2,2,1	5	10	12	5	.	
17	2,2,2	10	30	10	10	+	
18	2,3,1	0	20	11	20		
19	2,3,2	15	40	10	35	+	