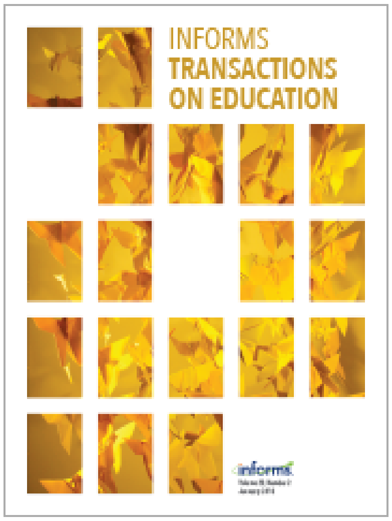

INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Translation Approach To Teaching Linear Program Formulation

Scott P. Stevens, Susan W. Palocsay,

A Translation Approach To Teaching Linear Program Formulation

Scott P. Stevens And Susan W. Palocsay
Information Technology & Management Science Program
James Madison University, MSC 0202
Harrisonburg, VA 22807
stevensp@jmu.edu
palocssw@jmu.edu

Abstract

While there have been many improvements in the teaching of operations research/management science (OR/MS) in recent years, students continue to have great difficulty with the process of constructing a linear programming model. We propose addressing this issue with a translation approach that breaks the process down into a series of small, well-defined steps. The underlying idea is to develop a representation of the problem in terms of measurable quantities that can then be translated, via application of an explicit rule, into the proper algebraic form and/or equivalent spreadsheet formulas. As background, related research on word problem solving from cognitive psychology and mathematical education is reviewed in the paper. We illustrate the translation approach with several examples and explain how using it can improve students' modeling skills. Preliminary data on the effectiveness of the methodology for undergraduate business students in an introductory MS course is also presented.

Editor's note: This is a pdf copy of an html document which resides at http://ite.pubs.informs.org/Vo4No3/StevensPalocsay/ (Volume 4, Number 3, May 2004)

1. Introduction

Students without a strong mathematical background frequently have difficulty formulating linear programs (LPs). They read the word problem quickly and then grope for the appropriate algebraic expressions. These expressions often mimic those appearing in text or in-class examples, even when such examples are irrelevant to the current problem. When finished, the student may be unable to explain what the program's variables represent, the meaning of its constraints, or the significance or reasonableness of any solutions proposed for it. This haphazard approach is common even among otherwise good students, and overcoming it demands a more intuitive way of thinking about formulation. It also requires introducing the tools to apply that new perspective.

Spreadsheet applications can be used to address some of these problems and are now widely employed in teaching LP formulation and solution. In its normal view a spreadsheet will display numbers, not formulas. Most students think more clearly about specific examples than general formulas, so this numeric display can help students to identify the correct relationships among the important elements of the problem. Unfortunately, spreadsheets are not a panacea. Students may inadvertently create nonlinear programs. They may identify as changing cells (decision variables) quantities that are outside of their direct control. If the trial values used for the decision variables give reasonable values for the other quantities in the problem, students will often assume that an erroneous formulation is correct. Direct spreadsheet formulation does not really eliminate the need for more traditional formulation skills - it recasts them as debugging skills. Of particular concern is that many students seem unable to generate an adequate spreadsheet representation of a problem without a teacher-provided template. Yet such templates are unlikely to be available in real-world applications.

In response to these concerns, we propose an approach of teaching LP formulation as a formal act of translation. The approach provides students with a consistent, step-by-step formulation method that emphasizes understandable concepts over mathematical hieroglyphics. Classroom testing suggests that the translation approach discourages many of the most common and glaring errors seen in student formulations. This linguistic orientation also dovetails naturally with Excel program representation.

2. Student Difficulties with Word Problem Solving

The cognitive processes associated with "thinking" have been studied extensively by psychologists, with an objective of understanding how humans reason and solve problems. Mathematics has been a domain of particular interest to cognitive scientists because it provides a rich set of problems that have standard, unambiguous representations and well-defined solution procedures. Mayer (1977) describes several early studies of the psychology of thinking in mathematics that have some relevance, either directly or indirectly, to LP model formulation. An experiment by Paige and Simon (1966) found evidence that students who can correctly solve algebra word problems are better able to translate problem information into an integrated diagram, in contrast to non-solvers who produced a series of diagrams with each corresponding to a fragment of the problem. In experiments on deductive reasoning using algebra word problems by Suppes, Loftus, and Jerman (1969) and Loftus and Suppes (1972), students' performance decreased under certain conditions: as the number of computational operations increased, when unit conversions were required, if problem sentence structure was complex, and when the problem differed from the previous one.

In one of many early efforts reported by Mayer (1977) to develop computer simulations of human problem solving behavior, Bobrow (1968) wrote a program he called STUDENT to solve algebra word problems. Most of the computer processing in STUDENT was associated with "direct translation", the phase that required changing English into a set of algebraic equations to which computational algorithms could be applied. Hinsley, Hayes, and Simon (1977) describe the implementation of the STUDENT translation process in more detail, explaining that the basic process is sometimes augmented by heuristics that make changes to the processing when the text indicates a standard algebra problem type such as an "age" or "work" problem. This corresponds to the introduction of semantic knowledge in the form of schema to the solution approach, which Hinsley, Hayes, and Simon hypothesized is an important aspect of algebra problem solving. Their experiments showed that people have two ways of attacking algebra word problems, depending on whether or not they recognize a known type early in the formulation. When encountering a familiar problem type, subjects invoked empirical rules based on information retrieved from memory that were specifically useful in setting up and solving that type. But if they did not recognize a problem type, they applied a general line-by-line procedure similar to the STUDENT direct translation process.

In more recent cognitive science studies, mathematics has been used as an application area for evaluating various general theories of problem solving, including the relationship between conceptual and procedural knowledge (Byrnes and Wasik, 1991), the integration of examples and procedures in problem solving (Reed and Bolstad, 1991), the ability of students to appropriately match examples to test problems (Reed, Willis, and Guarino, 1994), the acquisition of procedural knowledge from example problems (Anderson and Fincham, 1994), and the evolution of the solution process as people learn to skip steps (Blessing and Anderson, 1996). While the implications of this research are not yet clear with respect to improving mathematical instruction, further development of these theoretical models may eventually provide additional insight into sources of student errors in word problem solving. Another approach in cognitive science research has been to build computer-based tutoring tools to improve instruction in word problem solving as well as explore proposed theoretical models (e.g., Looi and Tan, 1998; Singley, et al., 1991). Wheeler and Regian (1999) present the results of a study where the use of an algebra word problem solving tutor improved the performance of students on both abstract and concrete reasoning tests but showed that students experienced considerable more difficulty on the abstract reasoning test.

Mathematical word problems have also been studied by psychologists interested in reading comprehension (see e.g., Cummins, et al., 1988; LeBlanc and Weber-Russell, 1996; Nathan, Kintsch, and Young, 1992). Kintsch (1998) gives a detailed overview of text comprehension theories from the perspective of solving arithmetic and algebra word problems. He identifies two crucial steps in the process of solving word problems where students have the most difficulty:

1. formulation of a mathematical model of the problem, and
2. determining whether or not they formulated a correct model.

He stresses the importance of providing help with the formulation step because students often "jump to equations without a clear understanding of the problem structure" (Kintsch, 1998). He also notes that an important characteristic of experienced algebra problem solvers is that they have learned to provide their own feedback by checking the reasonableness of their answers, whereas students without this ability frequently generate irrational values without noticing that their solution is incorrect.

The difficulty of word problem solving is also well known among educators in mathematics. As the emphasis on applications of mathematics increased, they observed a clear distinction between math computation skills and higher-level problem solving abilities and began to investigate possible causes (Cloer, 1981). Their research indicates that the modeling process is very complex, with algebra word problems having a significant cognitive component that requires construction and manipulation of representations of the problem elements (Singley, et al., 1989). Students begin to exhibit difficulties with this cognitive process as early as the sixth grade, and these difficulties often persist even after many years of mathematical study.

Booth (1984) investigated the underlying causes of a selected set of algebraic errors with high incidence among secondary school students and found these students resist the interpretation of letters as symbols representing the *numerical value* of an object (versus representing the object itself), and often do not understand the process of writing a symbolic mathematical statement (focusing instead on the numerical answer). Sharma (1987) describes three stages in the application of mathematics to real-world problems: translating the problem situation into a formal mathematical model, solving the mathematical problem, and interpreting the solution for the original situation. The most frequent error identified in the translation stage is referred to as the "reversal of variables". When asked to mathematically represent the relation that there is one professor for every six students, students write 6S = P instead of S = 6P (with S defined as the number of students and P as the number of professors). Clement, Lochhead, and Monk (1981) gave this problem to two samples of first-year college students; only 63% of the calculus-level engineering students and 43% of the non-science majors were able to answer it correctly. A significant majority of the wrong answers were associated with the students' tendency to reverse variables, and further study indicated that this tendency appeared to result from two different types of mistakes. Some students were using "word order matching," where they made a direct mapping from the English words to the algebraic symbols. Other students applied a "static-comparative" method, understanding that the student population was larger than the professor population but incorrectly believing that the larger coefficient should go with the variable for the larger population. In either case, the variables are treated as labels rather than numbers. Students who did the problem correctly showed better comprehension of the static comparison of the two groups, seeing the number S as being larger than the number P so that P must be multiplied by 6 to be equal to S.

More recently, Verschaffel, Greer, and de Corte (2000) performed a detailed analysis of a phenomenon called the "suspension of sense-making" (Schoenfeld, 1991) that frequently occurs when elementary school children are solving arithmetic word problems. In essence, they ignore the real world aspects of the problem context and treat word problems as puzzles that require the selection and application of some operation(s) to be solved. Consequently, the authors propose the adoption of a "modeling perspective" in the mathematics educational environment where word problems are presented as exercises in modeling realistic problem situations. A study by Lithner (2003) examined how college calculus students apply simplified strategies that result in answering questions with a suspension of sense-making. He characterizes their reasoning as superficial, based on identifying similarities and mimicking procedures instead of acquiring an understanding of the underlying, fundamental mathematical concepts.

3. The Linear Programming Model Formulation Process

Early books on linear programming devoted most of their attention to theoretical and algorithmic topics, with discussion of the process of developing an LP model primarily based on a few applications such as diet, production-scheduling, and transportation problems (Gass, 1964). Dantzig's (1963) chapter on formulating an LP model describes the "linear programming approach" as one of decomposing a system into a number of black box activities where the focus is on the rates of flow into and out of the activities. He outlines five steps in the model-building process:

1. define the activities;
2. define the items consumed or produced by the activities;
3. determine the input-output coefficients;
4. determine the net inputs or outputs of the system; and
5. determine the material balance equations.

He notes that repetition of these steps may be required in cases where the formulation of the balance constraints leads to the identification of additional activities. To support the model-building process, Dantzig graphically depicts the input and output items for each activity in a black box diagram before constructing the coefficient matrix. This activity analysis approach has been further developed and implemented in several computer-based modeling systems, including the LP-FORM system prototyped by Ma, Murphy, and Stohr (1996). Krishnan, Li, and Steier (1992) also use the activity analysis perspective in the development of a mathematical model formulation system based on artificial intelligence concepts. However, the potential usefulness of these systems in teaching LP modeling has not yet been explored.

With the rapid advancements in computer technology over the recent past, it has become increasingly viable to develop graphical interfaces to assist in the formulation of mathematical models. A decision support system with a graphical interface for network modeling was designed by Steiger, Sharda, and Leclaire (1993) to provide a non-mathematical environment for constructing models with NETFORM (Glover, Klingman, and McMillan, 1977) notation. In an overview of visualization techniques for optimization, Jones (1994 and 1996) provides a comprehensive survey of relevant literature on different representation styles for formulation, some of which are now available in commercial software products. From a teaching perspective, graphical representations have also been proposed as visual aids for helping students to translate LP word problems into the appropriate mathematical (algebraic) form. Evans and Camm (1990) advocate presenting semantic networks constructed with nodes and links as an intermediate step between the verbal problem statement and the LP model. Another example is influence diagrams, which are used as a guide in the development of algebraic LP model formulations by Meredith, Shafer, and Turban (2002). To date, however, there has been no formal research into the pedagogical effectiveness of using visualization-based approaches to teach model formulation.

OR/MS textbooks have evolved to support the teaching of LP modeling primarily by expanding discussion of the general steps in formulating an LP model from Dantzig and providing more detailed explanation of the development of example models. The steps are typically summarized as

1. understand the problem;
2. describe the objective in words;
3. describe each constraint in words;
4. define the decision variables;
5. express the objective in terms of the decision variables;
6. express each constraint in terms of the decision variables (Anderson, Sweeney, and Williams,; Moore and Weatherford, (2001).

In some texts, these steps are modified to outline a procedure for formulating an LP model directly onto a spreadsheet (Hillier and Hillier, 2003; Winston and Albright, 2001). There are also guidelines for implementing LP models in a spreadsheet, including suggestions for designing models that give a clear presentation of the problem situation while also being reliable and easily modifiable (Conway and Ragsdale, 1997; Winston and Albright, 2001). Example problems are usually presented as instances of general applications of LP, such as make-buy decisions, blending problems, production and inventory planning, multiperiod cash flow problems, and transportation routing. This approach has the benefit of defining schema that show students the structural features usually associated with a particular type of application.

Murphy and Panchanadam (1999) conducted an interesting set of experiments to test the value of using schemas in teaching undergraduate business students how to build LP models. Performance of students was evaluated under three different teaching approaches: an instance-based framework, where students were taught to derive models from individual example problems with no categorization of problem types; a "schema-later" framework, where categorization and identification of structural similarities by the instructor occurred after presentation of multiple example problems; and a "schema-earlier" framework where the order of presentation was reversed (i.e., schema presented before examples). The researchers hypothesized that students in the two schema classes would be better at categorizing LP problems, retrieving and using a similar example from taught material, defining decision variables, and building the algebraic LP than those students in the instance class. Using diet and transportation test problems, they found knowledge of schema later did not help students in using earlier examples, identifying the decision variables, or formulating the correct model. When the schema was presented earlier, students were better able to construct analogies to both categories and examples, but this schema knowledge did not improve performance as measured by number of models that were correctly formulated. They conclude that schemas do provide students with a conceptual framework that improves their understanding at the start of the modeling process and give them a tool for organizing and accessing the material. But unfortunately, presentation of this conceptual framework does not appear to significantly decrease modeling errors, based on the results from their study. They cite two possible reasons for this:

1. students have difficulty in verbally recognizing model constraints, and
2. students have difficulty translating a verbal concept into the corresponding algebraic expression.

Both the translation and interpretation stages of LP modeling require an understanding of the meaning of an equation and an operational process for converting a word problem into a mathematical model and stating its solution in words. Based on our experience, we believe that these translation difficulties are the primary cause of our students' struggles when attempting to construct their own LP models. Grossman (2002) has identified three "keys" that students need if they are to effectively use management science: model creation, spreadsheet engineering, and basic analytics. The skill of modeling a problem from scratch is essential for them to address business situations they will encounter in their professional work. They also need to be able to correctly develop a computer-based representation of the problem, and to apply basic analytical skills to systematically evaluate the final model. We believe that the translation approach described in this paper can be useful to instructors who want to help their students improve their LP modeling skills, algebraically and/or in the spreadsheet environment. It offers additional structure to the model formulation process, thereby reducing what appear to our students to be the mystical aspects of such general steps as "write the objective as a linear function of the decision variables." Its intrinsic focus on meaning leads to improvements in both spreadsheet implementation and post-optimality analysis. As a result, students respond more positively to OR/MS and leave the course with a higher level of confidence in their own ability to perform quantitative analysis.

4. The Translation Approach: An Overview

To translate from one language to another, one begins by understanding the text to be translated, then generates a meaningful (albeit foreign) text. Successful translation means that these two texts must express a common meaning, although this correspondence is rarely at a word level. We exploit the parallel between this process and the process of LP formulation to improve students' formulation skills and program comprehension.

The translation approach has two stages: rephrasing the word problem and constructing the mathematical program. In the first stage we focus on semantic content only, restating the original problem in terms of measurable quantities. This corresponds to the general steps (1) through (4) listed above. We essentially use a re-ordered version of the "helpful questions" posed by Camm and Evans (2000) to help identify the major components of the LP model:

1. What is the objective to be maximized or minimized?
2. What limitations or requirements restrict the values of the decision variables?

3. What am I trying to decide?

However, our approach differs in that the answers are formally expressed as measurable quantities. Because no numbers or variables are involved at this stage, students can and must focus on meaning and "math phobic" reactions are minimized. The quantities identified are natural choices for calculated cells in an Excel representation, and the relationships identified among them form the constraints of the program. We note that it is often appropriate to use graphical diagrams in this stage, to show structural relationships for more complex problems. In the second stage, a simple procedure is used to transform each measurable quantity into proper mathematical form. This stage corresponds to the general steps (5) and (6) but augments them with a rule that generalizes the mechanics for building mathematical expressions in the model.

In the following sections of the paper, we detail both the steps in applying the translation approach and the rationale for these steps, using several example problems to discuss the process from the perspective of the student. The development of this approach is based on our own work in modeling real world problems (e.g., Bopp, et al., 1996) as well as our teaching experience. A summary of the steps in the form of a class handout that can be used by the instructor is provided as an appendix to the paper.

5. Measurable Quantities and Categories of Constraints

A *measurable quantity* is any physically meaningful quantity that can be identified as "the number of *units of something*". The number of hours spent doing a job or the number employees working a given shift are examples of measurable quantities. Sometimes an important quantity is best thought of as a percentage of another measurable quantity, such as "20% of total revenue". In those instances, we can broaden the term "measurable quantity" to mean "measurable quantity or percentage thereof".

Measurable quantities are natural building blocks for LPs. The objective function of an LP represents a measurable quantity, as does each decision variable in its formulation. Each constraint in an LP represents a relation between two measurable quantities, asserting either that they are equal, that the first is no greater than the second, or that the first is no less than the second. We associate these three mathematical possibilities with three semantic categories of constraints: limited resource, minimum performance, and conservation.

Constraint Type	English Translation	Measurable Quantity Translation
Limited Resource[1]	"You can't use more than you have."	*# of units used ≤ # of units available*
Minimum Performance	"You must meet the quota."	*# of units obtained ≥ # of units required*
Conservation	"The books have to balance."	*# of units "coming in" = # of units "going out"*

Figure 1: Categories of Constraints.

Each constraint category has an equivalent in common English. These English forms help students to identify and correctly frame the problem's requirements. The result is the *measurable quantity representation* of the problem. We present the following example to illustrate this process.

5.1. Example 1

A bookstore receives 40 paperback copies and 65 hardback copies of a new novel from its shipping warehouse, but due to preorders taken from customers, it needs at least 80 copies of each type before the store opens tomorrow. The bookstore *must* have these 160 books, but would like to have as many copies of the book available for sale tomorrow as possible. Given the short time horizon, the shipping warehouse does its best. It has facilities to rush deliver up to ten boxes of books. The "paperback boxes" each hold 6 copies of the paperback. The warehouse also has 7 "hardback boxes" on hand, but these boxes actually contain 5 hardbacks and 2 paperbacks each. What should the warehouse ship in order to best address the store's needs?

The student begins in normal English, summarizing the problem as: "maximize books in the store tomorrow"; "can't ship too many hardback boxes", "can't ship more than 10 boxes" and "store has to have at least 80 of each book type". Most students will be able to reach this point with relative ease. If not, students are encouraged to look for key "constraint words" in the problem description, such as *at most, at least, no more than, no less than, up to, exactly, required, limited, restricted, must, cannot, budget,* and *demand*. The objective is relatively easy to identify; it is the quantity to be made as large as possible or as small as possible. The student then notes that the first two of the restrictions identified are limited resource constraints, while the last implies two separate minimum performance constraints. Applying the definitions from Figure 1, the student expresses the problem in terms of measurable quantities as shown in Figure 2.

Maximize	# of books in store, total
subject to	# of hardback boxes shipped ≤ # of hardback boxes available
	# of boxes shipped ≤ # of boxes of shipping capacity
	# of hardback copies in store ≥ # of hardback copies required
	# of paperback copies in store ≥ # of paperback copies required

Figure 2: Measurable quantity representation for Example 1.

Notice that this representation involves none of the numbers in the problem, nor knowledge of the problem's decision variables. It is an easily understandable conceptual representation, highlighting similarities among constraints (such as the last two constraints here) and emphasizing important properties (such as the fact that the two sides of a constraint must be expressed in a common unit). The language of the constraint templates can and should be adapted to reflect common English usage (e.g., "# of boxes shipped" instead of "# of boxes used").

Once the student has created this measurable quantity representation of the problem, he or she usually has a good idea of the decision variables - they are the directly controlled quantities in the problem. When difficulties arise, students may examine the measurable quantities appearing in the measurable quantity representation and ask, "What do the values of these quantities depend upon?" By recursively asking this question, they finally arrive at a point where the answer is simply, "Nothing. I get to pick the value." At that point, they have identified a decision variable. In our example, one concludes that warehouse management controls how many of each kind of box to ship. Decision variable definitions are written (like everything else) as measurable quantities, and thus must include units: P = # of paperback boxes to ship; H = # of hardback boxes to ship. (We have found that using mnemonic variable names, rather than the more traditional x and y, can aid students in their formulations.)

Students now apply a simple test to determine if all of the necessary decision variables have been found. They imagine that they are given numeric values for all of their decision variables, as well as the information in the problem. They then ask themselves if they could determine the values of all of the measurable quantities in the representation. If the answer is yes, the variables are sufficient for the problem. Most students quickly see, in our example, that knowing the number of boxes of each type shipped to the store will allow us to determine how many hardback boxes were shipped, the total number of boxes shipped, and the number of each type of book that the bookstore finally receives. The other quantities, such as the number of each type of book required, are given in the problem. There is no need for the student to write the appropriate algebra or spreadsheet formulas for these expressions at this time.

5.2. Expressing Measurable Quantities Mathematically

We turn now to the second stage of the translation approach: converting each measurable quantity into the appropriate linear expression and/or spreadsheet formula. While Hillier and Hillier (2003) propose a similar categorization of constraints, their presentation does not articulate a general procedure for conversion to a mathematical form. We have found it effective to implement this process as presented in Figure 3. The mathematically literate are well aware that the coefficient of a variable in a linear expression is just the partial derivative of that expression with respect to the variable in question. However, with or without this understanding, determining the numerical value of the coefficient in a particular problem instance requires a procedure to operationalize the concept.

To translate a measurable quantity in the program into the equivalent mathematical expression, follow this procedure:

1. Select one of the decision variables; let's call it X.
   a. State, in good English, what is means for X to increase by 1.
   b. Now assume that the change you identified in Step 1 does happen, but that all other decision variables remain unchanged. By how much does the measurable quantity (that you are trying to translate) increase? The answer will be a number, and this number is the coefficient of X in the mathematical representation of the measurable quantity.
2. Repeat Step 1 for each decision variable in the problem. Add the results.
3. Imagine setting all of the decision variables equal to zero. What would this mean in terms of the original problem? If this situation arose, what value should the measurable quantity take on? The answer to this question is the constant term. Add it to your result from Step 2.

Figure 3: Mathematically Expressing a Measurable Quantity: the Coefficient Rule

To demonstrate the use of the Coefficient Rule, we return to the bookstore problem.

5.3. Example 1 (continued)

The bookstore problem has two decision variables, P and H, so students know that every measurable quantity in the problem can be written as __P + __H + __. They must simply fill in the blanks with the appropriate numbers. Informally, the question becomes "if the variable goes up by one, by how much does the measurable quantity go up?" For example, consider the objective, # of books in store, total: __ P + __ H + __. If P increases by 1, then the warehouse shipped one more box of paperbacks. The measurable quantity - the total number of books obtained - increases by 6. On the other hand, if H increases by 1, then one more hardback box is shipped and the bookstore gains five hardbacks and two paperbacks. The measurable quantity is # of books in store, total, and this increases by 5 + 2 = 7 as a result of the additional hardback box. Thus, the coefficient of H is 7. To finish, the constant term is found by setting P and H equal to zero, which means that the warehouse does not ship at all. Then how many books does the bookstore have, total? Only what they already have on hand: 65 + 40 = 105 books. Hence, the objective is $6P + 7H + 105$. Applying the same techniques to the rest of the measurable quantities, one at a time, yields the LP formulation shown in Figure 4. The program is completed by adding any appropriate nonnegativity constraints ($P \geq 0, H \geq 0$).

Maximize

$$\overbrace{6P + 7H + 105}^{\text{\# of books shipped, total}}$$

subject to

$$\overbrace{1H}^{\text{\# of hardback boxes shipped}} \leq \overbrace{7}^{\text{\# of hardback boxes available}}$$

$$\overbrace{1P + 1H}^{\text{\# of boxes shipped}} \leq \overbrace{10}^{\text{\# of boxes of shipping capacity}}$$

$$\overbrace{5H + 65}^{\text{\# of hardback copies store obtains}} \geq \overbrace{80}^{\text{\# of hardback copies required}}$$

$$\overbrace{6P + 2H + 40}^{\text{\# of paperback copies store obtains}} \geq \overbrace{80}^{\text{\# of paperback copies required}}$$

and all variables ≥ 0

Figure 4: Mathematical formulation for Example 1.

Experience has shown that students who approach the bookstore problem without using this procedure will commonly write the left-hand-side of the paperback constraint as $6P + 2P + 40$, since the two copies from the "hardback box" are actually paperback books. With our procedure, they ask, "if H goes up by 1, how much does the # of paperbacks delivered go up?" Since H is the number of hardback boxes, one more box gives 2 more paperbacks, and the term $2H$ in the linear expression. The error is thus avoided.

We have seen improvements in students' ability to develop an LP spreadsheet model when they express the problem initially in terms of measurable quantities, whether or not they write an algebraic model. The measurable quantity analysis suggests to the student a sensible template for entering the program into Excel. The decision variables (changing cells) and the objective (target cell) have been identified, and each program requirement has been explicitly stated as a direct comparison between two measurable quantities. It would then be reasonable to have the values of these two quantities be reflected by two proximate cells in the spreadsheet. The application of the Coefficient Rule from Figure 3 in this context identifies the proper multiplier for each changing cell in the constraint and objective function formulas. Figure 5 shows one possible arrangement of the model elements, completed with the optimal solution to demonstrate the appropriate calculations. We follow a convention of red borders for changing cells, gray cells for data provided in the problem, yellow for the objective value, and white for computed values.

Decisions							
Number of hardback boxes to ship		5					
Number of paperback boxes to ship		5					
Data							
	Hardbacks	Paperbacks					
Contents of a hardback box	5	2					
Contents of a paperback box	0	6					
Requirements							
Boxes	*Shipped*		*Allowed*				
Hardback boxes	5	≤	7				
Total number of boxes	10	≤	10				
Books	*In stock*		*Rushed*		*Received*		*Required*
Total hardbacks	65	+	25	=	90	≥	80
Total paperbacks	40	+	40	=	80	≥	80
Results							
Total books received	170						

Figure 5: An Excel representation for Example 1.

5.4. Addressing Common Formulation Errors: Recipe Errors and Missing Constraints

The pervasive error noted by Sharma (1987), the "reversal of variables" is a special case of a more general error which we call "the recipe error". As an example, suppose that each Quality Control team consists of a statistician, two inspectors, and two workers. Let T be the number of teams and S, I, and W represent the total number of statisticians, inspectors and workers on these teams, respectively. Students commonly write $T = S + 2I + 2W$. But the Coefficient Rule in Figure 3 shows this to be absurd. The constraint is measured in teams, so the term $2W$, for example, says that obtaining one additional worker increases the number of teams by two. Most students think of $2W$ as meaning "two workers". A student using the Coefficient Rule sees it properly as "two (of the measurable quantity - teams) - for every worker". The Coefficient Rule thus indirectly includes an analysis of units, showing that the coefficient of W in this constraint would have to be measured in teams/worker, not in workers/team. The correct formulation, of course, is $S = T$, $I = 2T$, $W = 2T$. The first constraint, in measurable quantities, says *"# of statisticians involved = # of statisticians on QC teams "*, and the other two are similar.

The recipe error often arises in problems that include decision variables whose values are entirely determined by the value of other variables, as in the QC team example. The inclusion of such dependent variables is not a mistake; indeed, "extra" variables may improve program readability and clarify its structure, as in many multiperiod planning problems. When using such variables, though, students often omit needed program constraints. To address this problem, we introduce the notion of "extra", or *auxiliary* variables as defined in Figure 6.

> **Definition:** (*Independent and auxiliary variables*) Partition the decision variables into two subsets, **I** and **A** . This partition must meet two requirements:
>
> - The value of any variable in **A** can always be determined by knowing only the values of the variables in **I** and the information provided in the problem.
> - The value of any variable in **I** cannot in general be determined by knowing only the values of the remaining variables in **I** and the information provided in the problem.
>
> Given such a partition, we call the elements of **I** the *independent variables* and the elements of **A** the *auxiliary variables.*

Figure 6: A Partition of the Decision Variables

It's natural to view the independent variables as the decisions directly made and the auxiliaries as the unavoidable results of those decisions, but this is not strictly necessary. One would, for example, probably consider *# of dollars of change received from the cashier* as being an auxiliary to *# of dollars tendered* and *# of dollars spent on purchases*. It would be equally correct to view *# of dollars tendered* as auxiliary to the other two, and the choice does not affect our discussion. We simply need some partition of the variables into "essential information" and "derived information". Once the partition into independent and auxiliary variables has been carried out, correct formulation is assisted by invoking the Auxiliary Rule given in Figure 7.

> For each auxiliary variable, write one equality (=) constraint defining that auxiliary in terms of the other variables.

Figure 7: Constraints for Auxiliary Variables: the Auxiliary Rule

This constraint can often be sensibly viewed as a conservation constraint, but it may also be defined as a fourth category of constraint, the auxiliary variable definition. We use the next problem to demonstrate the incorporation of the Auxiliary Rule into the translation approach.

5.5. Example 2

(From Winston and Albright, 2001) Rylon Corporation manufactures Brute and Channelle perfumes from a raw material costing \$3/pound. Processing 1 pound of the material takes 1 hour and yields 3 ounces of Regular Brute and 4 ounces of Regular Chanelle. This perfume can be sold directly or can be further processed to make luxury versions of the perfumes. One ounce of Regular Brute can be sold for \$7, but 3 hours of processing will convert it into one ounce of Luxury Brute, which sells for \$18. One ounce of Regular Chanelle can be sold for \$6, but 2 hours of processing will convert it into one ounce of Luxury Chanelle, which sells for \$14. Rylon has available 4000 pounds of raw material, 6000 hours of processing time, and will sell everything it produces. Determine how Rylon can maximize its profit.

Using the templates from Figure 1, students immediately see the two limited resource constraints (raw materials and time). The goal of maximizing profit is also clear to them. Many see the problem as involving seven variables: *RBM, RCM, RBS, RCS, LBS, LCS* and *RAW*. The first two are the number of ounces of Regular Brute and Regular Channelle initially made, respectively. The next four are the number of ounces of Regular Brute, Regular Channelle, Luxury Brute, and Luxury Chanelle sold, respectively. The last, of course, is the number of pounds of raw materials used. While the materials constraint is easy enough now ($RAW \leq 4000$), students are often flummoxed as to what to do next. They may, for example, write a time constraint like $RBM + RCM + 3LBS + 2LCS \leq 6000$. If they write any other constraints involving *RAW* at all, they are commonly recipe errors, such as $RAW = 3RBM + 4RCM$. Auxiliary variables can help to cut through this confusion.

As soon as the value of *RAW* is known, students know how many ounces of Regular Brute and Regular Channelle are made. These quantities are auxiliary, then. There must be an equation defining each of them which depends on *RAW* alone. The translation procedure from Figure 3 gives the correct forms as $RBM = 3RAW$ and $RCM = 4RAW$. Now we sell some Regular Brute and Regular Channelle. But as soon as we know how much we sell, we also know how much Luxury Brute and Luxury Channelle will be made and sold. That is, *LBS* and *LCS* are auxiliary. This being the case, we need an equality constraint to define them. Since whatever Regular Brute is not sold will be made into Luxury Brute, $LBS = RBM - RBS$. Similarly, $LCS = RCM - RCS$. These constraints can be viewed as auxiliary variable definitions, of course, but they can also be seen from the perspective of conservation constraints. The total Brute made ("total in") will have to be equal to the total Brute used ("total out"). Introducing the concept of auxiliary variables acts as a safety net: it lets students know that such a constraint must exist, even if they haven't yet identified it.

From this point, completing the program is straightforward. Time is used only in processing raw materials or making luxury perfumes, so $RAW + 3LBS + 2LCS \leq 6000$. Money is made only on the sale of perfume, so the objective is $7RBS + 6RCS + 18LBS + 14LCS - 3RAW$. In each case, the coefficients are obtained by the Coefficient Rule. The complete formulation is shown below in Figure 8.

The use of auxiliary variables is always optional, and the problem above could be formulated with six, five, or even four decision variables. Auxiliary variables give the student more latitude in the particular way that they frame the problem, and can help to simplify the solution of a complex problem by recording the values of intermediate quantities of relevance. Few people would choose to manage their checkbooks by using only the decision variables of deposits and withdrawals, while disdaining the auxiliary variable of bank balance.

Maximize
$$\underbrace{\text{\# of \$ of net profit}}_{7RBS + 6RCS + 18LES + 14\ LCS - 3\ RAW}$$

subject to

$$\underbrace{\text{\# of pounds of raw material used}}_{RAW} \leq \underbrace{\text{\# of pounds of raw material available}}_{4000}$$

$$\underbrace{\text{\# of hours of processing time used}}_{1\ RAW + 3\ LBS + 2\ LCS} \leq \underbrace{\text{\# of hours of processing time available}}_{6000}$$

$$\underbrace{\text{\# of ounces of Regular Brute made}}_{RBM} = \underbrace{\text{\# of ounces of Regular Brute made}}_{3\ RAW}$$

$$\underbrace{\text{\# of ounces Regular Chanelle made}}_{RCM} = \underbrace{\text{\# of ounces of Regular Chanelle made}}_{4\ RAW}$$

$$\underbrace{\text{\# of ounces of Luxury Brute sold}} = \underbrace{\text{\# of ounces of Regular Brute not sold}}_{RBM - RBS}$$

$$\underbrace{\text{\# of ounces of Luxury Chanelle sold}}_{LCS} = \underbrace{\text{\# of ounces of Regular Chanelle not sold}}_{RCM - RCS}$$

and all variables ≥ 0

Figure 8: Measurable Quantity and LP Formulation for Example 2.

The detail in which we have analyzed the preceding examples may lead the reader to believe that the translation approach is much more time consuming than other methods of formulation. We have not found this to be the case. Once students become familiar with the technique, the time required to formulate seems to be about the same as with other techniques. The increase that we have observed has been in formulation quality, rather than formulation time. We close with a multiperiod example, presenting the translation approach solution with limited commentary.

5.6. Example 3

Bel Aire Pillow Company needs to manufacture 12,000 pillows in their factory next week. They have two kinds of employees: line workers (who do the actual work) and supervisors (who oversee the line workers). Bel Aire currently has 120 line workers and 20 supervisors. Each line worker can make 300 pillows per week. Supervisors do not engage in actual pillow-making.

A recent study done on Bel Aire's operation concluded that, for proper supervision, there must be at least 1 supervisor for every 5 line workers. The study also suggested that Bel Aire might be able to cut its costs by doing some or all of the following:

- hiring new employees as line workers (the training cost is $100/worker)

- firing some current line workers (the fired worker is given $240 severance pay)
- promoting some current line workers to supervisors (the training cost is $200/worker)
- demoting some current supervisors to line workers (no cost associated with this)

Pay for a line worker is $400/week. Pay for a supervisor is $700/week. All hiring and firing, and training occur before the upcoming work week, and the costs of those activities will be charged against the upcoming week. Bel Aire wishes to perform the appropriate hiring, firing, promotions and demotions that will minimize its operations cost for next week.

Students easily identify the objective and understand that Bel Aire's requirements for next week are that they must have sufficient production capacity to make the required pillows and no more lineworkers than can be supervised - a minimum performance constraint and a limited resource constraint, respectively. These things depend on how many people Bel Aire *PROMotes*, *DEMotes*, *HIREs*, and *FIREs*. This list of variables is sufficient for the problem, but many students will define additional (auxiliary) variables for the number of each kind of employee that will be on staff next week. Let *SUP2* be the number of supervisors working next week and *LINE2* be the number of lineworkers working next week. The measurable quantity representation of this problem and its corresponding linear program appear in Figure 9.

Many students not using the translation approach will omit the last two constraints. Those using the translation approach know that, since two auxiliary variables were used in the formulation, two equality constraints must appear in the program to define them. The Coefficient Rule given in Figure 3 does the rest.

Minimize
$$\overbrace{100\ HIRE + 240\ FIRE + 200\ PROM + 400\ LINE2 + 700\ SUP2}^{\text{\# of \$ spent next week}}$$
subject to
$$\overbrace{300\ LINE2}^{\text{\# of pillows of production capacity next week}} \geq \overbrace{12{,}000}^{\text{\# pillows needed next week}}$$
$$\overbrace{LINE2}^{\text{\# of line workers next week}} \leq \overbrace{5\ SUP2}^{\text{\# of line workers that can be supervised}}$$
$$\overbrace{SUP2}^{\text{\# of supervisors next week}} = \overbrace{20 + PROM - DEM}^{\text{\# of supervisors next week}}$$
$$\overbrace{LINE2}^{\text{\# of line workers next week}} = \overbrace{120 + HIRE - FIRE - PROM + DEM}^{\text{\# of line workers next week}}$$
and all variables ≥ 0

Figure 9: Measurable Quantity and LP Formulation for Example 3.

6. Preliminary Results

Result of experimental studies conducted to date have been encouraging. During each Spring and Fall academic session from Fall 2001 until Spring 2003, each student in the introductory management science for business course at our institution was evaluated with a set of seven questions testing basic LP formulation skills. These questions were administered at the end of the semester and were given both to students who had been taught using the translation approach to formulation (n = 548) and to those who were taught with more traditional approaches (n = 771). Students of the translation approach did better on six of the seven questions, and worse on the remaining question. As can be seen in Figure 10, the results were statistically significant (with $\alpha = 0.1$) in five of the seven cases.

Focus of "Word Problem" Question (Formulation)	% Correct (Translation Approach)	% Correct (Traditional Approach)	Difference	p-value
Identification of constraint words	75.9%	74.8%	1.1%	0.656
English interpretation of coefficient	79.2%	73.2%	6.0%	0.012
Recipe error (variable reversal)	54.0%	42.9%	11.1%	0.00007
Identifying decision variables	76.6%	55.6%	21.0%	4×10^{-15}
Formulating simple linear expression	97.3%	97.0%	0.2%	0.793
Finding coefficients for simple linear expression	84.7%	81.1%	3.6%	0.089
Formulating percentage constraint	84.1%	93.3%	-9.1%	1×10^{-7}

Figure 10: Formulation Performance of Introductory Management Science Students.

The fifth question was an extremely easy formulation question that was intended to verify minimal competency, and about 97% of all students answered it correctly. The translation approach students show marked superiority over their traditional counterparts on most of the other questions, with the notable exception of generating a percentage constraint. Investigation revealed that one of the translation approach teachers, in reworking course materials, had inadvertently removed all homework problems and in-class examples involving percentage constraints. The students with markedly poor performance on the percentage constraint question were generally in these classes. Students of other translation approach teachers who had not made this omission showed no significant difference in performance on this question from those students taught with the traditional approach.

The data also suggest that the translation approach may help students to better deal with questions of post-optimality analysis. At the end of the Fall 2002 and Spring 2003 sessions, all students in the introductory management science course at our institution were asked seven questions dealing with post-optimality analysis of a linear program. Again, both students taught with the translation approach (n = 316) and those taught with traditional approaches (n = 268) were included. The difference in performance levels between the two groups was significant for all seven questions at the 10% level of significance. The results are reported in Figure 11.

Focus of "Word Problem" Question (Post-Optimality)	% Correct (Translation Approach)	% Correct (Traditional Approach)	Difference	p-value
RHS ranging (outside range)	65.2%	29.5%	35.7%	0
Shadow price (inside range)	63.9%	31.7%	32.2%	9E-15
OFCR (objective change)	48.4%	26.1%	22.3%	3E-08
OFCR (schedule change)	57.6%	44.0%	13.6%	0.0011
Shadow price = 0 for nonbinding	50.3%	63.1%	-12.7%	0.002
Calculation of slack	93.4%	89.6%	3.8%	0.0986
OFCR (outside range)	66.1%	59.0%	7.2%	0.073

Figure 11: Post-Optimality Performance of Introductory Management Science Students.

Students taught with the translation approach methodology substantially outperformed students taught with the traditional approach on these questions, with the exception of the question 5. We believe that the tight linkage between English and mathematics intrinsic in the translation approach may help to avoid some of the obstacles discussed earlier, particularly the "suspension of sense-making" identified by Schoenfeld. While our study did not explicitly control for differences in teachers, we did ask each tested student additional questions concerning linear programming but not requiring "word problem" skills. Students were given the mathematical formulation of a linear program without an English context, then asked questions about solving it graphically, identifying binding and redundant constraints, and so on. Since the problem does not involve semantic content, we would expect no consistent differences in performance between the translation approach and traditional approach students. The results - 57.1% success for students of translation approach teachers and 54.5% success for students of traditional approach teachers - give no reason to suspect that either the teachers or the students in the translation approach classes were inherently better than their traditional counterparts (p = 0.351).

7. Conclusions

Students are generally more effective when using approaches that allow them to proceed with their work in sequential stages. The translation approach begins with the conceptual process of developing a measurable quantity representation of the problem, avoiding formulas and numbers and focusing student attention on what, precisely, they are trying to say. Vague notions of constraints can be refined by using the constraint templates for limited resource, minimum performance, and conservation constraints. The notion of auxiliary variables can help them to find implicit constraints. The results of this analysis can be used as the springboard to Excel template design.

The translation stage of the approach, as embodied in the Coefficient Rule, allows students to proceed in easy steps. It focuses on only a single measurable quantity (a "bracket") at a time. It creates the linear expression for that quantity without reference to the rest of the program. Students are "filling in the blanks" in expressions like $\_\_\ P + H +$ by repeatedly asking the question "If *this variable* goes up by 1, how much does the measurable quantity go up?" This approach helps to avoid many of the common formulation errors that are so tempting to the mathematically unsophisticated, including the recipe error and the omission of equality constraints needed when auxiliary variables are used.

By focusing on the measurable quantity representation of the program, the translation approach helps students to bring their (nonmathematical) critical thinking skills to bear on problems of mathematical modeling.

Students taught with the translation approach have significantly outperformed their fellow students in a number of key formulation skills. They also seem quicker to recognize that algebraic relations carry comprehensible (and often important) information. Such development is important if introductory management science courses are to be considered as critical components of business core curricula at both the undergraduate and graduate levels. Faculty who are interested in teaching the translation approach and participating in future experiments to evaluate student formulation performance are encouraged to contact us.

References

Anderson, D. R., D. J. Sweeney, and T. A.Williams. (2003), *An Introduction to Management Science: Quantitative Approaches to Decision Making*, Thomson Learning/South-Western Publishing: Mason, OH.

Anderson, J. R. and J. M. Fincham. (1994), "Acquisition of Procedural Skills From Examples," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 20, No. 6, pp. 1322-1340.

Blessing, S. B. and J. R. Anderson. (1996), "How People Learn to Skip Steps," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 22, No. 3, pp. 576-598.

Bobrow, D. G. (1968), "Natural Language Input for a Computer Problem Solving System," in *Semantic Information Processing*, M. Minsky (Ed.), pp. 132-215, MIT Press: Cambridge, MA.

Booth, L. (1984), "Misconceptions Leading to Error in Elementary Algebra," *Journal of Structural Learning*, Vol. 8, pp. 125-138.

Bopp, A., V. Kannan, S. Palocsay, S. Stevens. (1996), "An Optimization Model for Planning Natural Gas Purchases, Transportation, Storage, and Deliverability," *Omega, International Journal of Management Science*, Vol. 24, No. 5, pp. 511-522.

Byrnes, J. P. and B. A. Wasik. (1991), "Role of Conceptual Knowledge in Mathematical Procedural Learning," *Developmental Psychology*, Vol. 27, No. 5, pp. 777-786.

Camm, J. D. and J. R. Evans. (2000), *Management Science and Decision Technology*, Thomson Learning/South-Western Publishing: Cincinnati, OH.

Clement, J., J. Lochhead, and G. S. Monk. (1981), "Translation Difficulties in Learning Mathematics," *American Mathematical Monthly,* Vol. 88, No. 4, pp. 286-290.

Cloer, T., Jr. (1981), "Factors Affecting Comprehension of Math Word Problems - A Review of the Research," *ERIC Document Reproduction Service*, No. ED209655.

Conway, D.G. and C. T. Ragsdale. (1997), "Modeling Optimization Problems in the Unstructured World of Spreadsheets," *Omega, International Journal of Management Science,* Vol. 25, No. 3, pp. 313-322.

Cummins, D. D., W. Kintsch, K. Reusser, and R. Weimer. (1988), "The Role of Understanding in Solving Word Problems," *Cognitive Psychology*, Vol. 20, pp. 405-438.

Dantzig, G. B. (1963), *Linear Programming and Extensions*, Princeton University Press: Princeton, NJ.

Evans, J. R. and J. D. Camm. (1990), "Using Pictorial Representations in Teaching Linear Programming Modeling," *IIE Transactions*, Vol. 22, No. 2, pp. 191-195.

Gass, S. I. (1964), *Linear Programming: Methods and Applications*, McGraw-Hill Book Company: New York, NY.

Glover, F., J. D. Klingman, and C. McMillan. (1977), "The NETFORM Concept: A More Effective Model Form and Solution Procedure for Large Scale Nonlinear Problems," *Annual Proceedings of the ACM*, October 16-19, pp. 283-289.

Grossman, T. A. (2002), "The Keys to the Vault," *OR/MS Today,* Vol. 29, No. 6, pp. 12-13.

Hillier, F. S. and M. S. Hillier. (2003), *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, McGraw-Hill/Irwin: Boston, MA.

Hinsley, D. A., J. R. Hayes, and H. A. Simon. (1977), "From Words to Equations Meaning and Representation in Algebra Word Problems," in *Cognitive Processes in Comprehension*, M. A. Just and P. A. Carpenter (Eds.), pp. 89-106, Laurence Erlbaum Associations: Hillsdale, NJ.

Jones, C. V. (1994), "Visualization and Optimization," *INFORMS Journal on Computing*, Vol. 6, No. 3, pp. 221-257.

Jones, C. V. (1996), *Visualization and Optimization*, Kluwer Academic Publishers: Boston, MA.

Kintsch, W. (1998), *Comprehension, A Paradigm for Cognition*, p. 364, Cambridge University Press: Cambridge, MA.

Krishnan, R., X. Li, and D. Steir. (1992), "A Knowledge-Based Mathematical Model Formulation System," *Communications of the ACM*, Vol. 35, No. 9, pp. 138-146.

LeBlanc, M. D. and S. Weber-Russell. (1996), "Text Integration and Mathematical Connections: A Computer Model of Arithmetic Word Problem Solving," *Cognitive Science*, Vol. 20, pp. 357-407.

Lithner, J. (2003), "Students' Mathematical Reasoning in University Textbook Exercises," *Educational Studies in Mathematics*, Vol. 52, pp. 29-55.

Loftus, E. F. and P. Suppes. (1972), "Structural Variables that Determine Problem-solving Difficulty in Computer-Assisted Instruction," *Journal of Educational Psychology*, Vol. 63, pp. 531-542.

Looi, C. and B. Tan. (1998), "A Cognitive-Apprenticeship-Based Environment for Learning Word Problem Solving," *Journal of Computers in Mathematics and Science Teaching*, Vol. 17, No. 4, pp. 339-354.

Ma, P., F. H. Murphy, and E. A. Stohr. (1996), "An Implementation of LPFORM," *INFORMS Journal on Computing*, Vol. 8, No. 4, pp. 383-401.

Mayer, R. E. (1977), *Thinking and Problem Solving: An Introduction to Human Cognition and Learning*, Scott, Foresman and Co: Glenview, IL.

Meredith, J. R., S. M. Shafer, and E. Turban. (2002), *Quantitative Business Modeling*, Thomson Learning/South-Western Publishing: Mason, OH.

Moore, J. H. and L. R. Weatherford. (2001), *Decision Modeling with Microsoft ® Excel*, Prentice-Hall, Inc: Upper Saddle River, NJ.

Murphy, F. H. and V. Panchanadam. (1999), "Using Analogical Reasoning and Schema Formation to Improve the Success in Formulation Linear Programming Models," *Operations Research*, Vol. 47, No. 5, pp. 663-674.

Nathan, M. J., W. Kintsch, and E. Young. (1992), "A Theory of Algebra-Word-Problem Comprehension and Its Implications for the Design of Learning Environments," *Cognition and Instruction*, Vol. 9, No. 4, pp. 329-389.

Paige, J. M. and H. A. Simon. (1966), "Cognitive Processes in Solving Algebra Word Problems," in *Problem Solving: Research, Method and Theory*, B. Kleinmuntz (Ed.), Wiley: New York, NY.

Ragsdale, C. T. (2001), *Spreadsheet Modeling and Decision Analysis*, Thomson Learning/South-Western Publishing: Cincinnati, OH.

Reed, S. K. and C. A. Bolstad. (1991), "Use of Examples and Procedures in Problem Solving," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 17, No. 4, pp. 753-766.

Reed, S. K., D. Willis, and J. Guarino. (1994), "Selecting Examples for Solving Word Problems," *Journal of Educational Psychology*, Vol. 86, No. 3, pp. 380-388.

Schoenfeld, A. (1991), "On Mathematics as Sense-Making: An Informal Attack on the Unfortunate Divorce of Formal and Informal Mathematics," in *Informal Reasoning and Education*, J. Voss, D. Perkins, and J. Segal (Eds.), pp. 311-344, Lawrence Erlbaum Associates: Hillsdale, NJ.

Sharma, M. C. (1987), "Problems in Algebra: Part One (Why Some Students Have Difficulty in Learning Algebra)," in *Math Notebook, Information for Teachers/Parents of Children with Learning Problems in Mathematics,* Vol. 5, ERIC Document Reproduction Service, No. ED342600.

Singley, M. K., J. R. Anderson, J. S. Gevins, and D. Hoffman. (1989), "The Algebra Word Problem Tutor," *Proceedings of the 4 th International Conference on Artificial Intelligence and Education, pp. 267-275,* Amsterdam, Netherlands.

Steiger, D., R. Sharda, and B. Leclaire. (1993), "Graphical Interfaces for Network Modeling: A Model Management System Perspective," *INFORMS Journal on Computing*, Vol. 5, No. 3, pp. 275-281.

Suppes, P., E. F. Loftus, and M. Jerman. (1969), "Problem-solving on a Computer-based Teletype," *Educational Studies in Mathematics*, Vol. 2, pp. 1-15.

Verschaffel, L. B. Greer, and E. de Corte. (2000), *Making Sense of Word Problems*, Swets and Zeitlinger: Lisse, The Netherlands.

Wheeler, J. L. and J. W. Regian. (1999), "The Use of a Cognitive Tutoring System in the Improvement of the Abstract Reasoning Component of Word Problem Solving," *Computers in Human Behaviour*, Vol. 15, pp. 243-254.

Winston, W. L. and S. C. Albright. (2001), *Practical Management Science*, p. 150, Thomson Learning/Duxbury: Pacific Grove, CA.

Appendix

Linear Program Model Formulation Steps

1. **Read and understand the problem.** Ignore the numbers in the problem and concentrate on verbally identifying the parts of the linear program:
   - Objective (overall goal)
   - Constraints (limitations and requirements)
   - Decision variables (quantities under direct control)

   Drawing a picture, making a table, or finding a trial solution may help you to better understand the problem structure.

2. **Write the measurable quantity formulation of the linear programming model.** A *measurable quantity* is the "*# of units of something*." Translate your English statement of the objective into a measurable quantity and define each decision variable as a measurable quantity. Rewrite each constraint as a relation between two measurable quantities, based on one of the following categories:

Constraint Type	English Translation	Measurable Quantity Translation
Limited Resource	"You can't use more than you have."	*# of units used ≤ # of units available*
Minimum Performance	"You must meet the quota."	*# of units obtained ≥ # of units required*
Conservation	"The books have to balance."	*# of units "coming in" = # of units "going out"*

3.**Translate the measurable quantity formulation into a mathematical model.** Apply the Coefficient Rule described below to each measurable quantity in the problem, using a "fill in the blanks" approach.

Coefficient Rule

To translate a measurable quantity in the program into the equivalent mathematical expression, follow this procedure:

1. Select one of the decision variables; let's call it X.
   a. State, in good English, what is means for X to increase by 1.
   b. Now assume that the change you identified in Step 1 *does* happen, but that all other decision variables remain unchanged. By how much does the measurable quantity (that you are trying to translate) increase? The answer will be a number, and this number is the coefficient of X in the mathematical representation of the measurable quantity.
2. Repeat Step 1 for each decision variable in the problem. Add the results.

Imagine setting all of the decision variables equal to zero. What would this mean in terms of the original problem? If this situation arose, what value *should* the measurable quantity take on? The answer to this question is the *constant term*. Add it to your result from Step 2.

4. **Add an equality constraint for each auxiliary variable.** If you used any variables whose values are entirely determined by the other variables in the problem, your final program must include one equality constraint (=) for each such auxiliary variable, defining how it is calculated from those other variables.

5. **Validate your model**, by plugging in numbers to make sure your mathematical expressions are correct.