Understanding and Overcoming Difficulties with Building Mathematical Models in Engineering: Using Visualization to Aide in Optimization Courses.

Rachael Kenney, Nelson Uhan, Ji Soo Yi, Sung-Hee Kim, Mohan Gopaladesikan, Aiman Shamsul and Amit Hundia

Introduction

Operations research—and in particular, optimization—is one of the key courses in many universities' engineering curricula. An optimization model (or mathematical program) is a mathematical representation of a decision-making problem, consisting of variables that reflect the decisions to be made, and an objective function to be minimized or maximized, subject to a set of mathematical constraints on the variables. Formulating a valid optimization model from a verbal description of a decision-making, problem is perhaps the most important skill taught in an optimization course aimed at undergraduate students, since excellent modeling skills are vital to putting optimization techniques into practice. However, though undergraduate engineering students have been engaging in modeling activities (i.e., mathematical "word" or "story" problems) since elementary school, many students find it difficult to learn how to build good optimization models. Many educators in operations research anecdotally report this phenomenon (e.g., Sokol 2005), but little work has been done on systematically understanding why optimization modeling is such a difficult skill to learn and how such insights can lead to effective modeling pedagogies. By effectively teaching optimization modeling skills, we can provide our students with a powerful set of tools that can help solve important, complex problems in engineering, mathematics, and management.

The objective of this study is to help undergraduate engineering students overcome their difficulties in optimization modeling by

- determining and understanding commonly made mistakes in optimization modeling;
- developing a visual, web-based environment that teaches students to formulate valid and tractable optimization models; and
- evaluating the effectiveness of the developed visual, web-based environment on learning modeling in optimization.

This preliminary proposal is intended to share work completed on the first two objectives and to generate discussions to help us better conceptualize the next stages of our project.

Literature Review

A modeling approach to teaching in engineering or mathematics puts the focus in problem solving on creating a system of relationships that is generalizable and reusable (Doerr & English, 2003). Contemporary approaches to solving mathematical story problems have emphasized the need for a proper conceptual understanding of the problem. However, the factors that inhibit such conceptual understanding are quite complex. Lucangelli, Tressoldi, and Cendron (1998) suggest that problem solving with modeling problems is more difficult than solving algorithms because it requires (a) comprehension of the text, (b) ability to visualize the data provided, (c) capacity to recognize the underlying structure, (d) ability to correctly sequence solution activities, and (e) ability to evaluate the procedures used. These skills are especially important when solving college-level word problems in engineering where the problem complexity is often increased, contributing to learners' difficulties with problem solving (Jonassen, 2000).

The ability to translate from one representation of a mathematical problem to another is critical to the problem solving process (Janvier, 1987). However, it has been shown that even after several years of schooling in algebra or calculus, students often cannot engage successfully in this translation (Clement, Lochhead, & Monk, 1981; Clement, 1982; Janvier, 1987; Arcavi, 1994). Learners need a way of "developing a cognitive representation of information in the story" (Jonassen, 2000, p. 79). That is, in order to be successful, problem solvers must have an accurate mental representation of the pattern of information indicated by the story problem (Hayes & Simon, 1976; Riley & Greeno, 1988; Jonassen, 2000).

Researchers have discussed the role of visual diagrams as a conceptual tool that promote students' construction of flexible and applicable concept images that allow for flexible problem solving and connection making (e.g. Dreyfus, 1994; Koedinger, 1994; Larkin & Simon, 1987). According to Dreyfus (1994), computer-designed diagrams can be thought of as cognitive tools that make it possible to represent mathematics with an amount of visual structure that we cannot readily achieve with any other medium. Koedinger (1994) has identified emergent properties of diagrams that that make them superior to a linear representation of information for many learning and reasoning activities. For example, they provide the potential for students to recognize relationships that may have otherwise gone unnoticed in a verbal or symbolic representation. This supports earlier findings from Larkin and Simon (1987) who identified a diagrams' superiority to verbal problem descriptions due to their usefulness for grouping together all useful information and for supporting a large number of perceptual inferences. Koedinger (1994) suggests that students are more practiced in relying on perceptual inferences than the corresponding symbolic inferences, making the former often seem easier for the learner.

Methodology and Preliminary Findings

One end-goal for our study is to develop a visualization tool that can aid in modeling. Before fully developing this tool, we first need to better understand students' experiences and practices when solving optimization modeling problems (specifically linear programming problems) and to identify common errors that a visualization tool could help correct. The following sections outline our procedures and findings for the first three phases of our work.

Taxonomy of Optimization Modeling Word Problems

As a first step, we looked at five optimization textbooks (Hillier and Lieberman 1995, 2001; Rardin, 1997; Srinivasan, 2007, Winston, 1994) to determine their categorizations of different linear programming models. After comparing these categorizations, we developed a preliminary, unified taxonomy of word problems, based on the types of constraints a problem requires (i.e., the *constraint patterns*). Then, we tested the validity of this taxonomy by solving approximately 35 word problems from the different textbooks and examining how each problem fit into our taxonomy. Throughout this process, we discovered that some constraint patterns needed to be more specific, and so we revised our taxonomy accordingly.

Our current version of the constraint pattern taxonomy consists of five categories: (a) *composition constraints* (indicated by terms such as "meets", "has only", and "more than"); (b) *balance constraints* (e.g. "Each *A* requires *x* number of…"); (c) *ratio constraints* (often includes a mixture of *A* and *B*); (d) *pattern-covering constraints* ("*x* people work this type of shift"); and (e) *time-based constraints* (e.g. investment problems). By identifying and categorizing the different types of constraints, we propose that we may be able to develop a more generalizable method for formulating mathematical models across all problem types.

Taxonomy of Common Student Errors

To identify students' common errors and difficulties with modeling problems, we analyzed three sets of quizzes (one question each) from two sections of an optimization course and three sets of similar word problems, given to students on their final exams in three different semesters. We first studied each response and recorded the specific errors each response contained, keeping track of similar errors between students. We then categorized these errors broadly, depending on where they appeared in the model: the decision variables, the objective function, or the constraints. During this process, we also kept track of summary statistics for each type of error.

The analysis of students' responses showed mistakes on 84% of the 374 total responses analyzed. We identified five categories for the taxonomy of mistakes: (a) *mathematical notation errors* indicate errors associated with, for example, missing or having too many summation signs, or reversing indices; (b) *comparison errors* indicate mistakes in the direction of the inequality sign; (c) *flow errors* usually occur when multiple statements relate to one constraint but students forget to take that into consideration; (d) *missing information errors* suggest a student ignored the type of constraint construction, such as a profit function equation where you need the revenue and cost equations; and (e) *decision variable errors* included missing decision variables from the objective function or constraints, replacing a decision variable with some other variable, or using incorrect parameters. We found that the majority of mistakes fit into either the mathematical notation error (25.40%) or decision variable error (20.32%) categories. Comparison errors were found least (4.55%). Our current data on student errors comes from students' work on only one type of constraint pattern – composition constraints. As our work continues, we will collect and analyze data related to other types of constraint patterns.

Development of a Visualization Tool

Based on these taxonomies, we designed a preliminary visualization scheme that could help students gain a better conceptual understanding of optimization modeling problems and that could diminish the types of mistakes typically made on these problems. We have considered several visualization types, including node-link diagrams, tables, and timeline diagrams, by solving different word problems using these visualizations. After reviewing a number of different types of problems and student work, we found that node-link diagrams provided a possible basis for a robust visualization scheme to represent the conceptual ideas in a wide range of word problems. We are currently developing a prototype of an interactive visualization web tool (shown in Figure 1) based on our investigation of students work on a composition constraint pattern problem. The tool is intended to guide the students by letting them interact with the question (given in written form at the top) by allowing them to form node-link diagrams that represent their conceptual understanding of the problem. This is to help students understand the flow of the problem and identify the constraints available in the question.

At this point, our data collection has only included quantitative data from textbooks and student work samples. This data has provided us with an informative view of students' experiences in modeling in linear programming, but it is incomplete. In the next stages of our project, we will conduct qualitative interviews to further understand difficulties in modeling, begin to test our prototype of the visualization tool with students, and expand the tool to include additional feedback capabilities and to handle several different constraint pattern problems.

Questions for Consideration

Our team would be interested in discussing the following questions during the conference:

1. What approaches should we use to investigate the underlying causes of the mistakes that many students make in order to inform the design of our visualization tool?
2. How can we most effectively study the usefulness of the tool with students?
3. How can we ensure that the skills that the students learn through our tool (if any) are generalizable beyond the types of problems for which the tool is designed?
4. What are best practices for incorporating an interactive learning tool into a traditional lecture-driven course? What would make such a tool appealing to other instructors?

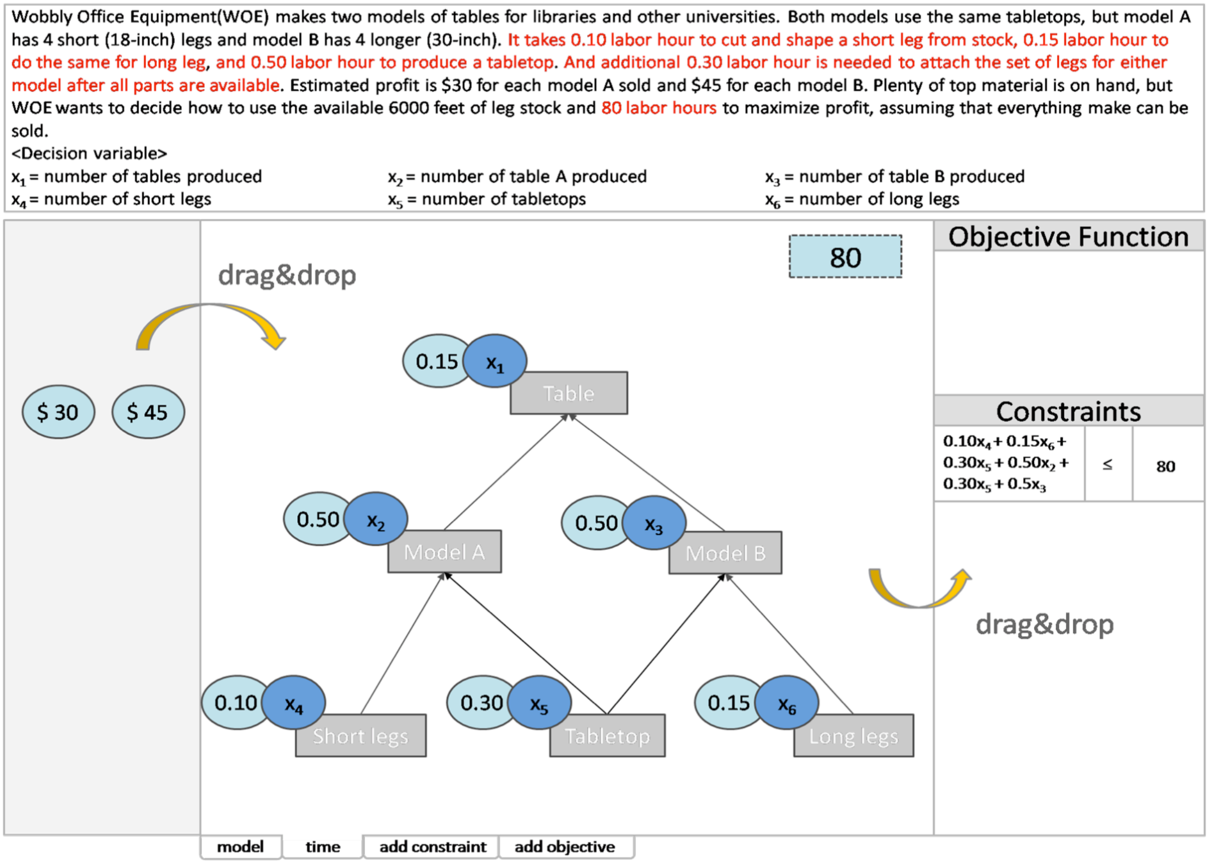

Figure 1. Screenshot of the visualization tool prototype

References

Arcavi, A. (1994). Symbol sense: Informal sense-making in formal mathematics. For the Learning of Mathematics, 14, 24–35.

Clement, J. (1982). Algebra word problem solutions: Thought processes underlying a common misconception. Journal for Research in Mathematics Education, 13, 16–30.

Clement, J., Lochhead, J., & Monk, G. (1981). Translation difficulties in learning mathematics. American Mathematical Monthly, 88, 286–290.

Doerr, H. M., & English, L. D. (2003). A modeling perspective on students' mathematical reasoning about data. Journal for Research in Mathematics Education, 34, 110–136.

Dreyfus, T. (1994). The role of cognitive tools in mathematics education. In R. Biehler, R. Scholz, R. Strasser, & B. Winkelmann (Eds.) Didactics of mathematics as a scientific discipline. Dordrecht: Kluwer.

Hayes, J. R., & Simon, H. A. (1976). The understanding process: problem isomorphs. Cognitive Psychology, 8, 165–190.

Hillier, F. S., Lieberman, G. J. (1995). *Introduction to Mathematical Programming*. 2nd edition. McGraw-Hill.

Hillier, F. S., Lieberman, G. J. (2001) *Introduction to Operations Research*. 7th edition. McGraw-Hill.

Janvier, C. (1987). Translation processes in mathematics education. In C. Janvier (Ed.) Problems of representation in the teaching and learning of mathematics, (pp. 27–32). Hillsdale, NJ: Lawrence Erlbaum.

Jonassen, D. H. (2000). Toward a design theory of problem solving. Educational Technology, Research and Development, 48, 63–85.

Koedinger, K. R. (1994). Emergent properties and structural constraints: advantages of diagrammatic representations for reasoning and learning. In B. Chandrasekaran, & H. Simon (Eds.) Reasoning with diagrammatic representations, (pp. 151–156). Menlo Park, CA: AAAI Press.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 11, 65–69.

Lucangelli, D., Tressoldi, P. E., & Cendron, M. (1998). Cognitive and metacognitive abilities involved in the solution of mathematical word problems: Validation of a comprehensive model. Contemporary Educational Psychology, 23, 257–275.

Rardin, R. L. (1997). *Optimization in Operations Research*. Prentice Hall, Upper Saddle River, NJ.

Riley, M. S., & Greeno, J. G. (1988). Developmental analysis of understanding language about quantities and solving problems. Cognition and Instruction, 5, 49–101.

Sokol, J. (2005). Teaching O. R. modeling. OR/MS Today, December 2005.

Srinivasan, G. (2007). *Operations Research: Principles and Applications*. Prentice Hall of India.

Winston, W. L. (1994) *Operations Research: Applications and Algorithms*. 3rd edition. Dux bury Press.