International Journal of Science and Mathematics Education (2020) 18:635–655
https://doi.org/10.1007/s10763-019-09980-5

Linear Programming Models: Identifying Common Errors in Engineering Students' Work with Complex Word Problems

Rachael Kenney[1] · Tuyin An[2] · Sung-Hee Kim[3] · Nelson A. Uhan[4] · Ji Soo Yi[5] · Aiman Shamsul[6]

Received: 30 October 2018 / Accepted: 9 April 2019 / Published online: 20 May 2019
© Ministry of Science and Technology, Taiwan 2019

Abstract

In linear programming, many students find it difficult to translate a verbal description of a problem into a valid mathematical model. To better understand this, we examine the existing characteristics of college engineering students' errors across linear programming (LP) problems. We examined textbooks to identify the types of problems typically found in introductory linear programming courses. We then developed a comprehensive set of tasks and analyzed students' work to create a taxonomy of the errors and issues that students exhibited. From our findings, we define four categories of identified error types: (1) decision variable errors, (2) variable relationship errors, (3) notation errors, and (4) form errors. This study contributes to the research by investigating students' work in an area of undergraduate mathematics that has not been heavily explored before. Findings suggest specific areas of focus for future work in helping students develop their understanding of linear programming models and mathematical modeling in word problems in general.

Keywords Engineering · Error analysis · Linear programming · Mathematization · Word problems

✉ Rachael Kenney
rhkenney@purdue.edu

[1] Department of Mathematics, Purdue University, 150 N. University St, West Lafayette, IN 47906, USA

[2] Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA, USA

[3] Department of Industrial ICT Engineering, Dong-eui University, Busan, South Korea

[4] Mathematics Department, United States Naval Academy, Annapolis, MD, USA

[5] IT and Mobile Communications, Samsung Electronics Co., Ltd, Suwon, South Korea

[6] Treasury, Bank Negara, Kuala Lumpur, Malaysia

With increased attention given to STEM education, it is important for us to not lose sight of the "M" and the roles it plays in "S," "T," and "E." The ability to mathematize a situation that has been presented in a realistic context or word problem in science, technology, or engineering is often critical but is a skill that involves mathematical insight and experience (Niss, 2010). Many students struggle with representing contextual word problems mathematically, which limits their ability to make use of mathematics to solve important problems in their fields of study (Lucangeli, Tressoldi, & Cendron, 1998). Yet, the mathematizing of phenomena is one of the most powerful uses of mathematics. This is especially true in the field of engineering, where the construction of mathematical models is central to the work of engineers (Gainsburg, 2006; Perlow & Bailyn, 1997). In practice, computer software packages are often available that can produce solutions to mathematical models of word problems, which reduces the need to memorize solution algorithms. However, algorithms, no matter how sophisticated or automated, are useless without accurate models to solve. Thus, our interests turn to the practices of engineering students to investigate issues encountered when constructing mathematical models from complex word problems.

One of the key components of the undergraduate engineering curriculum, and one that is essential to areas of mathematics, the natural sciences, and business, is *operations research (OR)*—the discipline of mathematical decision-making. College students are often introduced to this discipline through a course in mathematical optimization, also known as mathematical programming. An *optimization model* (or mathematical program) is a symbolic representation of a decision-making word problem. It consists of *decision 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. In this paper, we focus on one type of optimization problem: *linear programming (LP)*, which is concerned with the optimization of a linear objective function subject to one or more linear constraints. A typical introductory undergraduate optimization course focuses on LP modeling, since it provides a foundation for more sophisticated modeling frameworks. Such courses also cover algorithms (e.g. the simplex method) that solve LP models.

Figure 1 shows an LP problem used in this study. LP problems are more complex than typical algebra word problems encountered in earlier undergraduate mathematics courses. A correct model for this example requires three decision variables, an objective function, and six constraint equations. This is typical of the complexity of LP problems. The increased number of variables and complexity can contribute to complications with these problems (dos Santos & Brodlie, 2004; Jonassen, 2000). Students can struggle with deciding what to include (Wang & Brooks, 2007) and how to mathematically represent the chosen items.

To better understand how students reason with LP models, it is important to first understand the existing characteristics of different student errors across LP problems. Studying errors in mathematics can be a powerful tool and can serve as "springboards for inquiry" by unpacking questions surrounding the cause and setting of errors (Borasi, 1994). Considering errors created by students can serve as a starting point for remediation and reflection (Borasi, 1994; Kaczmarczyk, Petrick, East, & Herman, 2010).

In this study, we examine work on LP word problems where students are asked to only provide the model (not a numeric solution for the model). We address the following questions: (a) *What issues and errors do engineering students exhibit as*

Q1: Feeding Cows Farmer Jane wants to make sure her cows are fed a daily diet that contains at least 22 grams (g) of fat, 38 g of carbohydrates, and 7 g of protein. In addition, she wants each cow's daily diet to contain at most 4 g of sodium and 5 g of potassium. She has access to 3 different foods – Food X, Food Y, and Food Z – that she can mix to create a blend for her cows. Food X contains 7 g of fat, 10 g of carbohydrates, 3 g of protein, 1 g of sodium, and 4 g of potassium, and costs $0.25 per ounce. Food Y contains 12 g of fat, 12 g of carbohydrates, 1 g of protein, 1 g of sodium, and 2 g of potassium, and costs $0.30 per ounce. Food Z contains 8 g of fat, 12 g of carbohydrates, 3 g of protein, 5 g of sodium, and 5 g of potassium, and costs $0.20 per ounce. Formulate a linear program that determines a blend of Food X, Food Y and Food Z that meets one cow's nutritional requirements at minimum cost. Use the decision variables:

x = ounces of Food X in the blend,
y = ounces of Food Y in the blend,
z = ounces of Food Z in the blend.

Solution: Minimize $0.25x + 0.3y + 0.2z$ (total cost) subject to
$7x + 12y + 8z \geq 22$ (fat)
$10x + 12y + 12z \geq 38$ (carbohydrates)
$3x + y + 3z \geq 7$ (protein)
$x + y + 5z \leq 4$ (sodium)
$4x + 2y + 5z \leq 5$ (potassium)
$x, y, z \geq 0$

Fig. 1 Example of a linear programming modeling word problem used in this study, labeled as Quiz 1—feeding cows problem

they build symbolic models for typical LP problems? and (b) *What relationships, if any, exist between the characteristics of the LP problems and the errors found in students' work?* We present our findings as taxonomies developed from careful evaluation of OR textbooks and engineering students' work on course quizzes. Our goal is to provide an overview of observable issues in students' written work, setting a foundation for further investigations of student reasoning and difficulties with mathematizing complicated word problems in advanced mathematics.

Background

The LP problems shared below mimic traditional textbook word problems. We use the term "model" in this paper as an equivalent term to symbolic or algebraic representation to be consistent with terminology in OR texts; these typically refer to the "linear programming model" (Hillier & Lieberman, 2014; Rardin, 2016; Winston, 2003).

Difficulties with Creating Mathematical Representations in Word Problems

Mathematizing a given problem allows a problem solver to apply mathematical tools to think more precisely or quantitatively about the ideas. The steps involved in this process can include (a) taking a real-life problem and posing a well-defined question about it, (b) creating a mathematical representation, (c) solving and/or analyzing the mathematics, and (d) translating that solution/analysis back into the real world (Panaoura, Gagatsis, & Demetriou, 2009; Sokol, 2005). The traditional mathematics curriculum that focuses heavily on symbolic manipulation prepares students primarily for step (c)—solving the mathematical model. However, university students often have access to software that can compute solutions to optimization models like LP models, and the calculation procedures needed are often elementary and unchallenging for engineers (Gainsburg, 2006). For our research, we are interested in the act of translating from a word statement to a symbolic mathematical model. We focus our attention primarily on step (b) above: taking a verbal description of a problem and creating a

mathematical representation (e.g. an LP model). This is an important step in using mathematics to answer real-world problems, but it is one of the most difficult (Abrams, 2001; Niss, 2010). The act of translating from verbal to symbolic representations often puts the learner on unfamiliar ground (Abrams, 2001; Niss, 2010; Sokol, 2005).

A mathematical representation results from the translation of a problem or idea into symbols. Symbols are what make it possible for a person doing mathematics to communicate, manipulate, and reflect upon abstract mathematical concepts. Many difficulties in mathematics can be attributed to students' problems with manipulating and understanding algebraic symbols (Driscoll, 1999; Gray & Tall, 1994; Kinzel, 1999; Stacey & Macgregor, 1999). Lucangeli et al. (1998) suggest that factors that contribute to students' difficulties are complex due to the number of components involved, including:

- the comprehension of the text,
- the ability to visualize the data provided,
- the capacity to recognize the underlying structure,
- the ability to correctly sequence solution activities, and
- the ability to evaluate the procedures used

The ability to transition from one representation to another is critical to this process (Gagatsis & Shiakalli, 2004). Previous studies have suggested that even after several years of schooling in algebra or calculus, students struggle to engage successfully in this transition (Arcavi, 2005; Stacey & MacGregor, 1999). It requires one to be able to organize and restructure concepts from the information into usable algebraic notation (Driscoll, 1999; Stacey & MacGregor, 1999) and to be able to recognize patterns and organize them in a way that can be communicated symbolically (Driscoll, 1999).

There is a clear interplay between symbolic and natural language in mathematical word problem solving. Students need to decode the language and the context of the question and represent words with mathematical symbols. It is possible, however, for one person's interpretation of words and symbols to differ from another's based on prior experiences (Kenney, 2015; Kinzel, 1999). In LP programming, the potential for errors in translating from words to symbols is compounded by the fact that students have little prior experience with problems at this complexity level. Examining patterns in errors allows us to develop common language to use and reflect on student understanding in this context (Priem, 2010).

Usefulness of Error Analysis

Error analysis can provide rich information about students' learning processes and practical help for teachers regarding individualizing their own instruction (Ashlock, 2010; Borasi, 1994; Kaczmarczyk et al., 2010). Particularly, a systematic error analysis can reveal how far a learner has progressed in their way of reasoning about a mathematical topic, and errors can be seen as possibly necessary steps in the development of mature concepts (Smith, DiSessa, & Roschelle, 1993). Instead of considering errors as things that should be avoided and overcome in student work, we accept them in this study as a normal part of developing conceptual understanding. Recognizing common issues can help us make informed decisions for designing explorations or interventions.

Analyses of errors related to mathematical word problems can be found in the literature; however, much of the error analysis has concentrated on secondary mathematics (Kingsdorf & Krawec, 2014; Stacey & MacGregor, 1999; Wiens, 2007) or on only a specific error type (Weinberg, 2007; Yazdani, 2008). Exceptions include research in physics and computer science that have created a "concept inventory" to identify specific areas of student difficulties (Danielsiek, Paul, & Vahrenhold, 2012; Goldman, Gross, Heeren, Herman, Kaczmarczyk, Loui, & Zilles, 2008; Kaczmarczyk et al., 2010). These researchers used a Delphi study and follow-up interviews to collect their data. While we do not follow the same methodology, the intentions are the same—to begin to identify core components of LP modeling that are prone to student error.

One important influence for our work comes from Sebrechts, Enright, Bennett, and Martin (1996), who looked at university students' work on algebra word problems. They characterized problems by their major attributes (e.g. number of givens, representation structure, and linguistic characteristics) and then described solutions to the problems in terms of students' strategies and errors, creating an error taxonomy. They identified general errors such as computation errors, weighting errors, and inability to carry out a constructed plan for solving. They found that students had struggled with problems that required variables explicitly in their solution and that, when they could, students tended to use representations that did not require variables explicitly. In this study, we extend their research to more complex problems where the "answer" must always involve variables (recall that we are only looking at the models created, not the numeric solutions to the model). We are interested in the specific errors that persist in engineering students' work at this level.

Theoretical Perspective

Our investigation is situated in a conception-based, constructivist perspective where mathematical learning is viewed as a process by which learners develop conceptions to organize their experiential worlds (Simon, Tzur, Heinz, & Kinzel, 2004; von Glasersfeld, 1995). We operate under the assumption that learners only have access to mathematics that is created through their own activity. Students' experiences with symbols and models are unique for each learner and influence the development of conceptual understanding of mathematizing word problems.

Mathematical learning can be seen as taking place as learners cycle through a series of actions (physical and mental), reflections, and abstractions in a way that allows them to develop sophisticated cognitive images of phenomena (Hershkowitz, Schwarz, & Dreyfus, 2001; Simon et al., 2004). We see learning as occurring in this manner regardless of instructional setting. Even when teachers tell students directly how to understand a given mathematical situation or construct a mathematical model, students make their own sense of what the teacher says and shows and can only do so in terms of their existing ways of knowing (Cobb, Yackel, & Wood, 1992). This suggests that to be successful with problems such as LP models, students must be able to identify relevant facts, connect them to their existing knowledge as it relates to the problem domain, elicit symbolic models with which they are familiar, and reflect on the usefulness of such models for producing a solution to the problem (Lesh & Doerr, 2003; Panaoura et al., 2009; Sokol, 2005).

When a student constructs a mathematical representation of a problem, they can be seen as demonstrating their initial conceptual structure of the problem (Cifarelli, 1993). The initial structure, characterized by the created representation, is crucial for later development of abstract structures. It is these initial structures, evident in solutions on written course quizzes, that we focus on in this study. We analyze these as initial representations that allow us access, to some extent, to engineering students' initial reasoning with LP models.

Methods

The work reported in this paper is part of a larger project for developing web-based interactive tools to help students learn LP modeling. Here, we work to identify the most important errors created by students as a starting point for remediation and reflection. We focus on students' written work to identify common errors that an interactive tool could be designed to help address and generate an overview of the kinds of errors that students make. We divide our work for this paper into two phases as described below, following methods for developing taxonomies similar to those described in Sebrechts et al. (1996).

Phase 1: Classifying LP Problems and Creating Quizzes

We began by identifying the types of LP modeling tasks typically encountered in an introductory optimization course. We examined LP problems in three commonly used optimization textbooks (Hillier & Lieberman, 2014; Rardin, 2016; Winston, 2003). These textbooks primarily classify LP models based on their area of application. For example, Rardin (2016) and Winston (2003) classify LP models according to different classic practical applications (e.g. financial portfolio optimization, blending in chemical processes, and inventory management) and provide examples for students to follow. Hillier and Lieberman (2014) also provide examples of applications of LP modeling but do not attempt to classify models into general categories.

Classifying by Constraint Type After examining the textbooks and problems, we classified LP modeling word problems based on the type(s) of constraints required for the model rather than their area of application. We chose this method of classification because (a) it is possible that student errors could be found to be consistent with one constraint type over another within the same problem and (b) it is possible that errors in students' models could stem from issues with the underlying mathematical or conceptual relationships, rather than the problem context. Following Sebrechts et al. (1996), we classified the LP problems in a way that could aid our understanding of how problem design may influence problem-solving processes. Our five resulting constraint types include *composition*, *balance*, *ratio*, *pattern covering*, and *non-negativity*. Table 1 shows characteristics and examples of each type.

Note that these constraint types are not defined by the superficial appearance of the mathematical equality or inequality in the constraints but by the type of conceptual relationship between decision variables required by the context of the given word problem. For instance, consider the example for composition constraints in Table 1:

Table 1 Constraint types in linear programming modeling tasks

Constraint type	Characteristics	Example
Composition	• Used when a quantity is compared with a fixed value • Typically involves an inequality to bound a linear combination of quantities • Indicated by terms such as "meets," "has available," and "more/less than"	"Manufacturing a unit of Product A requires 2 hours of labor, and Product B, 3 hours. There are 200 hours of labor available." $2A + 3B \leq 200$
Balance	• Used to enforce conservation of a quantity in a system that depends on the decision variables • Most often involves an equality • Indicated by phrases such as each x requires y number of …	"Each unit of Product A produced is sold as-is, or used to produce Product B. Each unit of Product B requires 2 units of Product A." $A_{\text{produced}} = A_{\text{sold}} + 2B_{\text{produced}}$
Ratio	• Used to enforce a defined proportional requirement between quantities depending on the decision variables • Often required in problems that involve blending of products	"A mixture used to make pralines must contain at least 20% nuts." $N \geq 0.2P$
Pattern covering	• Used when decision variables represent quantities of a particular pattern of objects • Often arise when a problem involves complicated patterns, such as workforce shift scheduling problems	"There must be at least 12 postal workers working on Fridays." $\sum_{\text{shift } i \text{ includes Friday}} p_i \geq 12$
Non-negativity	• Used in most models to indicate that only positive values make sense in the problem context	$x_1, x_2, x_3 \geq 0$

manufacturing a unit of product A uses 2 h of labor, and product B, 3 h. Suppose A and B are decision variables that represent how many units of products A and B are manufactured. Then the total amount of labor required to manufacture these products can be understood from direct interpretation of the problem and can be expressed in terms of the decision variables: $2A + 3B$. This constraint is used to bind a linear combination of existing variables. Since there are 200 h of labor available, our model should constrain the values so that $2A + 3B \leq 200$. On the other hand, consider the example given for ratio constraints, which states that the mixture used to make pralines (P) must contain at least 20% nuts (N). Our model should constrain the values of N and P so that $N \geq 0.2P$, which we can rewrite as $0.2P - N \leq 0$. Superficially, this looks like the example for the composition constraints. However, the ratio constraint represents a multiplicative or proportional relationship among the quantities. Thus, the conceptual relationships embodied by these two constraints are different.

Balance constraints impose a conservation of quantities of variables across a linear equation. Modelers need to properly determine which variables should be included in each constraint. Pattern-covering constraints, on the other hand, are needed when decision variables represent quantities of a particular *pattern* of objects instead of quantities of the objects themselves. As a result, these constraints require a different type of conceptual thinking than composition constraints. For example, suppose we want to assign workers to shifts, and there must be at least 12 workers on Thursday and

Friday. However, they can only work in the following shifts: (1) M-W-F, (2) M-T-W, (3) T-W-TH(R). A first inclination might be to define decision variables y_M, y_T, y_W, y_R, and y_F to represent the number of workers each day. However, it is impossible to write *linear* relationships that ensure that these variables' values are consistent with the possible shifts available. As an alternative, we can define a decision variable x_{MWF} to represent the number of workers who work the M-W-F shift and define x_{MTW} and x_{TWR} similarly. Then we can write the constraint as $x_{MWF} + x_{TWR} \geq 12$. In this case, determining variables to include in the left-hand side of the inequality requires an additional conceptual hurdle of understanding the working shift patterns and how to incorporate them in the definition of the decision variables.

Designing Quizzes for Data Collection Categorizing constraints allows us to design quizzes covering different problem types and conceptual hurdles that students may encounter. One of the authors served as the optimization course instructor. He created five quizzes for the study that were reviewed and approved by the research team. Quiz questions (Q2–Q5) are provided below in Figs. 2, 3, 4, and 5 (quiz 1 (Q1) was given above in Fig. 1).

We chose to provide the students with the names and definitions of all decision variables on the quiz tasks. We know that students can struggle greatly with defining variables, and we were afraid that this difficulty would prohibit us from looking in depth at other issues that students face. Thus, we decided to reduce this problem area and direct our attention on the translation of words into mathematical models. Table 2 shows solutions to all five quiz questions and the distribution of constraint types that were needed in the solutions. Note that a single LP problem can contain more than one type of constraints, and we classify these problems based on the combination of the constraint types required.

Phase 2: Building a Taxonomy of Student Errors

Data Collection We collected data from 66 students enrolled in an undergraduate engineering optimization course who were being introduced to this topic for the first time. All students enrolled in the course were invited to participate by answering five extra credit quiz questions (Q1–Q5) given one at a time over the course of the semester (note that students were given an option to do the extra credit but disallow its use in research if they wished). The research team designed quiz problems to reflect one or

Q2: College Fund The Smiths have twins who will begin college in 4 years. They want to design an investment strategy over the next 4 years for a college fund. At the beginning of each year, they plan to put $25,000 into the fund, and any money in the fund can be invested in a CD that returns 4% after 1 year, and another CD that returns 9% after 2 years. This year, they have an opportunity to make an investment that would return 17% after 4 years. All funds must be invested at any time. Formulate an LP that determines an investment strategy that maximizes the value of the college fund, using the following decision variables:

$x_{1,2}$ = amount invested at the beginning of year 1 that becomes available at the beginning of year 2
$x_{1,3}$ = amount invested at the beginning of year 1 that becomes available at the beginning of year 3
$x_{1,5}$ = amount invested at the beginning of year 1 that becomes available at the beginning of year 5
$x_{2,3}$ = amount invested at the beginning of year 2 that becomes available at the beginning of year 3
$x_{2,4}$ = amount invested at the beginning of year 2 that becomes available at the beginning of year 4
$x_{3,4}$ = amount invested at the beginning of year 3 that becomes available at the beginning of year 4
$x_{3,5}$ = amount invested at the beginning of year 3 that becomes available at the beginning of year 5
$x_{4,5}$ = amount invested at the beginning of year 4 that becomes available at the beginning of year 5

Fig. 2 Quiz 2—college fund investment problem

Q3: Candy Factory In your candy business, you can produce two types of candies: Primal Pralines and Dual Doves, using different mixtures of sugar, nuts, and chocolate. You have 100 ounces sugar, 20 ounces nuts, and 30 ounces chocolate in inventory. The mixture for Primal Pralines must contain at least 20% nuts. The mixture for Dual Doves must contain at least 10% nuts and 10% chocolate. Each ounce of Primal Pralines can be sold for 25 cents, and each ounce of Dual Doves can be sold for 20 cents. Since your candy is so delicious, you can always sell all the candy you produce. Formulate an LP that determines a production plan that maximizes revenue, using the decision variables:

p_1 = amount of sugar used to produce Primal Pralines, in ounces
p_2 = amount of nuts used to produce Primal Pralines, in ounces
p_3 = amount of chocolate used to produce Primal Pralines, in ounces
d_1 = amount of sugar used to produce Dual Doves, in ounces
d_2 = amount of nuts used to produce Dual Doves, in ounces
d_3 = amount of chocolate used to produce Dual Doves, in ounces

Fig. 3 Quiz 3—candy factory problem

two of the constraint types identified in phase 1, covering all constraint types with the five quizzes. Question difficulty levels matched problems in course homework assignments. We limited the scope of our quiz questions to those that provide numerical parameters (e.g. "You have 100 ounces of sugar, 20 ounces of nuts …") instead of abstract symbolic parameters (e.g. "You have a_i units of ingredient i"). While student errors resulting from confusion between symbols that represent parameters and symbols that represent decision variables are interesting, we decided in this study to begin our focus on errors that students committed on easier LP problems.

Quiz responses were collected and graded by teaching assistants and then de-identified for anonymity by the research team. On average, 33.6 (30–37) out of 66 students submitted responses for each extra credit quiz. For the five quizzes, 168 total quizzes were submitted and 74 responses received full credit (i.e. made no errors) or requested to be excluded from the research, leaving 91 responses to be coded. Because a single quiz was often coded with multiple errors, we chose the coded errors as our unit of analysis instead of individual quizzes. As shown in Table 3, we identified 184 total coded errors across all five quizzes.

One limitation of our data collection was that we did not interview students as they worked on the quiz questions. Due to the classroom structure, quizzes were completed during large-lecture sessions; thus, it was unreasonable for researchers to talk to students while they worked. Analysis of the quizzes was not completed until after the course ended, and we deemed it too late to invite participants back to recall the ideas they had portrayed.

Data Analysis To begin the data analysis, 50 incorrect responses were randomly selected (10 from each quiz set). Following the methods in Sebrechts et al. (1996), an initial set of codes was developed using these samples by two members of the research team. This initial list was expanded and modified through discussion by the whole research team.

Q4: Nurse Shift Scheduling The Simplexville Hospital is trying to determine the schedule for nurses in the emergency room for the next few weeks. The hospital uses the following work shifts:

Shift	*Days of week covered by shift*
1	Sunday, Monday, Wednesday, Thursday
2	Monday, Tuesday, Thursday, Friday
3	Tuesday, Wednesday, Friday, Saturday
4	Wednesday, Thursday, Saturday, Sunday

10 nurses are needed on Monday, Tuesday, Wednesday, and Thursday; 12 nurses are needed on Friday and Saturday; and 8 nurses are needed on Sunday. Formulate an LP that determines a staffing plan for the hospital that minimizes the total number of nurses needed, using the following decision variables:

x_i = number of nurses assigned to shift i for $i = 1, 2, 3, 4$.

Fig. 4 Quiz 4—nurse shift scheduling problem

Q5: Perfect Stack: Perfect Stack builds two types of palettes: standard and long. Palettes are assembled from three components: standard separators, long separators, and cross pieces. Each standard palette consists of 3 standard separators and 10 cross pieces, and takes 0.25 hours to assemble. Each long palette consists of 3 long separators and 18 cross pieces, and takes 0.30 hours to assemble. Components need to be fabricated before being used to assemble a palette: it takes 0.005 hours to fabricate a standard separator, 0.007 hours to fabricate a long separator, and 0.002 hours to fabricate a cross piece. Perfect Stack has 200 hours of assembly time and 40 hours of fabrication time available. Standard palettes are sold at a profit of $5 each, and long palettes are sold at a profit of $7 each. Assume all palettes produced are sold. Write an LP that determines a production plan that maximizes profit, using the following decision variables:

x_1 = number of standard palettes assembled and sold,
x_2 = number of long palettes assembled and sold,
y_1 = number of standard separators fabricated,
y_2 = number of long separators fabricated,
y_3 = number of cross pieces fabricated.

Fig. 5 Quiz 5—perfect stack manufacturing company problem

We agreed that code definitions should obey the following rules: (a) they should be mutually exclusive and occur more than three times across responses, (b) similarities or differences between certain definitions should be clarified to avoid possible confusions, and (c) each definition should contain an example selected from the error pool to help coders understand its application. Our initial taxonomy included eight code categories with 25 subcategories. After a preliminary independent coding session to test codes with the data, three coders from the team came together to review and revise codes. Several codes were identified as redundant or combinable with others. Others were relevant to only three or fewer cases and were removed. This process was repeated one additional time, and a reduced set of codes was established with four major categories and 12 subcategories (shown in the next section). The three coders then coded all 91 quizzes and met to discuss discrepancies. Initial inter-rater reliability was approximately 60%. All discrepancies in the data were resolved so that all three coders agreed on all codes assigned, with a final inter-rater reliability of 100%.

Findings

Our analysis resulted in a taxonomy of errors (Table 4) that contains four primary categories—(a) *decision variable errors (DV)*, (b) *variable relationship errors (VR)*, (c) *notation errors (N)*, *and* (d) *form errors (F)*—with 12 total subcodes. Examples from student work are provided below to illustrate and provide a more detailed explanation of each code, followed by a discussion of how these codes help us think about student understanding.

Decision Variable Errors

Decision variables in an LP model represent the decisions to be made or the unknowns (e.g. number of tabletops required, number of workers assigned to one shift). Quiz problems used in this study included between three and seven decision variables. Decision variables were provided and explicitly defined for students on all five quizzes.

We defined decision variable errors as those instances where students introduced their own new variables that were either disruptive and unnecessary (DV.1) or not correctly connected to the given variables (DV.2). In the former case, students often

Table 2 Quiz solutions and constraint types

Quiz	Solution	Constraint type
Q1: feeding cows	Objective: minimize $0.25x + 0.3y + 0.2z$ (total cost)	
	subject to	
	$7x + 12y + 8z \geq 22$ (fat)	Composition
	$10x + 12y + 12z \geq 38$ (carbohydrates)	Composition
	$3x + y + 3z \geq 7$ (protein)	Composition
	$x + y + 5z \leq 4$ (sodium)	Composition
	$4x + 2y + 5z \leq 5$ (potassium)	Composition
	$x, y, z \geq 0$	Non-negativity
Q2: college fund	Objective: maximize $1.17x_{1,5} + 1.09x_{3,5} + 1.04x_{4,5}$ (total fund value)	
	subject to	
	$x_{1,5} + x_{1,3} + x_{1,2} = 25{,}000$ (cash flow year 1)	Balance
	$x_{2,4} + x_{2,3} = 25{,}000 + 1.04x_{1,2}$ (cash flow year 2)	Balance
	$x_{3,5} + x_{3,4} = 25{,}000 + 1.09x_{1,3} + 1.04x_{2,3}$ (cash flow year 3)	Balance
	$x_{4,5} = 25{,}000 + 1.09x_{2,4} + 1.04x_{3,4}$ (cash flow year 4)	Balance
	$x_{1,2}, x_{1,3}, x_{1,5}, x_{2,3}, x_{2,4}, x_{3,4}, x_{3,5}, x_{4,5} \geq 0$	Non-negativity
Q3: candy factory	Objective: maximize $0.25(p_1 + p_2 + p_3) + 0.2(d_1 + d_2 + d_3)$ (total revenue)	
	subject to	
	$p_2 \geq 0.2(p_1 + p_2 + p_3)$ (at least 20% nuts for PP)	Ratio
	$d_2 \geq 0.1(d_1 + d_2 + d_3)$ (at least 10% nuts for DD)	Ratio
	$d_3 \geq 0.1(d_1 + d_2 + d_3)$ (at least 10% chocolate for DD)	Ratio
	$p_1 + d_1 \leq 100$ (sugar capacity)	Composition
	$p_2 + d_2 \leq 20$ (nut capacity)	Composition
	$p_3 + d_3 \leq 30$ (chocolate capacity)	Composition
	$p_1, p_2, p_3, d_1, d_2, d_3 \geq 0$	Non-negativity
Q4: nurse shift scheduling	Objective: minimize $x_1 + x_2 + x_3 + x_4$ (total number of nurses needed)	
	subject to	
	$x_1 + x_4 \geq 8$ (nurses needed on Sunday)	Pattern covering
	$x_1 + x_2 \geq 10$ (nurses needed on Monday)	Pattern covering
	$x_2 + x_3 \geq 10$ (nurses needed on Tuesday)	Pattern covering
	$x_1 + x_3 + x_4 \geq 10$ (nurses needed on Wednesday)	Pattern covering
	$x_1 + x_2 + x_4 \geq 10$ (nurses needed on Thursday)	Pattern covering
	$x_2 + x_3 \geq 12$ (nurses needed on Friday)	Pattern covering
	$x_3 + x_4 \geq 12$ (nurses needed on Saturday)	Pattern covering
	$x_1, x_2, x_3, x_4 \geq 0$	Non-negativity
Q5: perfect stack building company	Objective: maximize $5x_1 + 7x_2$ (total profit)	
	subject to	
	$3x_1 = y_1$	Balance
	$3x_2 = y_2$	Balance
	$10x_1 + 18x_2 = y_3$	Balance
	$0.25x_1 + 0.30x_2 \leq 200$ (assembly time)	Composition
	$0.005y_1 + 0.007y_2 + 0.002y_3 \leq 40$ (fabrication time)	Composition
	$x_1, x_2, y_1, y_2, y_3 \geq 0$	Non-negativity

Table 3 Distribution of codes across quizzes

	Q1	Q2	Q3	Q4	Q5	Totals
Quizzes taken	35	37	33	33	30	168
Quizzes coded	15	33	17	9	17	91
Errors coded	17	99	31	12	25	184

ignored the given defined variables and created ones that held meaning to them. However, the introduction of these new variables always resulted in an incorrect model for the problem. We labeled this introduction as disruptive to the modeling process. A DV.1 code was seen to trump most other errors in a problem because the disruptive variables were interpreted as the cause of many other mistakes. We coded 20 such errors, predominantly in the objective functions.

One interpretation for why some students may have felt they needed new variables was that there were not enough given variables to "cover" all of the values given in the word problem. For example, quiz 1 (Q1) involves creating a daily diet for cattle that contains certain amounts of fat, carbohydrates, protein, etc. The quiz contains the line "Food X contains 7 g of fat, 10 g of carbohydrates, 3 g of protein, 1 g of sodium, and 4 g of potassium, and costs 0.25 per ounce," and the problem instructs students to use "x" as "ounces of Food X in the blend," "y" as "ounces of Food Y", etc. A correct model relates the amount of, for example, fat in Foods X, Y, and Z to the total amount of fat allotted (e.g. $7x + 12y + 8z \geq 22$). One student, however, defined five new variables (x_1 = grams of fat, x_2 = grams of carbs, and so on) and wrote the constraint: $7x_1 + 10x_2 + 3x_3 + 1x_4 + 4x_5 = 0.25x$. This student produced a literal translation of the text to equation (7 g of fat plus 10 g of carbs …) and needed additional variables to complete their model.

Student work was coded DV.2 if they introduced new variables that were useful to the task but were redundant and not explicitly connected back to the given decision

Table 4 Taxonomy of errors in linear programming modeling word problems

Code categories	Subcategories
Decision variable errors	DV.1. Introduce and incorrectly use redundant/disruptive variables DV.2. Introduce useful auxiliary decision variables without explicit connection to existing decision variables
Variable relationship errors	VR.1. Trivial relationships VR.2. Missing or incorrect coefficients VR.3. Proportional reasoning error VR.4. Missing/additional variables with partially correct relationship VR.5. Grouped by the wrong characteristics
Notation errors	N.1. Isolated subscripts N.2. Incorrect use of summation symbol or for statement N.3. Direction of the inequality N.4. Incorrect relational symbol
Form errors	F.1. Missing non-negativity constraints

variables. Work surrounding this code (found eight times) was usually partially correct; however, we had to make some assumptions about what the new variables represented. This code was primarily assigned in Q3, the candy factory problem. Here, seven students combined the variables that represented separate candy ingredients into a single variable to represent the whole recipe. Figure 6 shows an example of this. If one assumes that this student meant that $x_1 = p_1 + p_2 + p_3$, then the objective function is correct. Without further details from the students, it is difficult to determine their intended meaning, especially when most students who provided this representation wrote their constraint functions using the given "p_i" variables provided instead of their new variables (as seen here). This results in a mathematically incorrect model because the objective function contains variables that are never constrained.

Variable Relationship Errors

Variable relationship errors were found in both the objective and constraint functions in students' responses. These errors are connected to the ways in which students make sense of the given information in the problem and represent it using the given decision variables in the model. We identified five categories of incorrect relationships among the variables and the data (e.g. coefficients) given in the problems. This category of errors was identified more often than any others and was most frequent in the balance constraints found in Q2 and Q5.

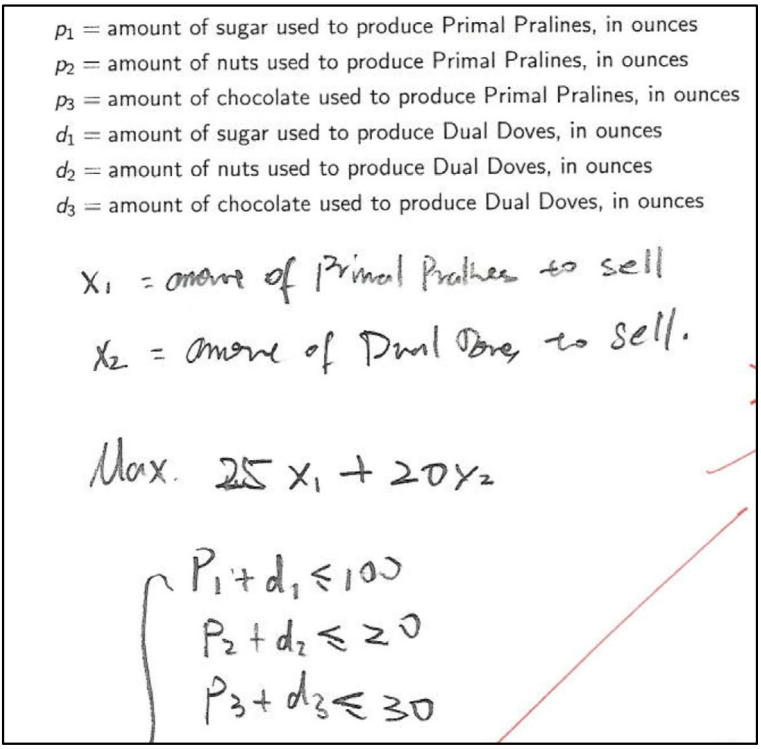

Fig. 6 A student's introduction of unnecessary or unclear variables on Q3

Trivial relationships errors (VR.1) were identified six times but represented a continuing difficulty that even advanced engineering students retain with interpreting mathematical symbols in context. In these cases, students did not represent relationships among the given variables but instead related every variable to a numeric value. For example, given the statement on Q3, "The mixture used to make Primal Pralines must contain at least 20% nuts," and the decision variable "p_2 = amount of nuts used to produce Primal Pralines," multiple students wrote the constraint: $p_2 \geq 1/5$. Such statements do not make sense when translated into words (p_2 is greater than 1/5 of …?), but, for the student, it may represent what they were told in the problem, i.e., that the amount of nuts must be more than some number. A similar example was found on Q5, the perfect stack problem, where students modeled the statement "it takes 0.005 hours to fabricate a standard separator" as $y_1 = 0.005$. This equation ignores the fact that y_1 stands for the "number of standard palates fabricated" and not the time of fabrication. It is common for students to change the meaning of a given defined variable to suit the statements they are trying to create. It is also possible that some students believe that a variable can have more than one meaning.

Code VR.2 (missing or incorrect coefficients) was applied when students had a partially correct answer but had not included the correct coefficients, and VR.4 (missing or additional variables) indicated that they did not have all of the correct variable terms. These two codes had a higher occurrence than any others (20 and 40, respectively) and were found predominantly on Q2, the college fund problem. Q2 asked students to formulate a model for an investment strategy to maximize a college fund where money could be invested for 1, 2, or 4 years for the next 4 years. For a decision variable $x_{i,j}$, i is the year the investment is made and j is the year that the investment becomes available to withdraw. Investments yielded different interest rates depending on the length of investment. The correct objective function for this problem is to maximize the total fund value: $1.17x_{1,5} + 1.09x_{3,5} + 1.04x_{4,5}$. Figure 7 illustrates an error committed by several students who included all decision variables in the objective function instead of only those that dealt with the end of the 4-year investment (thus, we coded this as VR.4 because there were more variables than needed). It is possible that these students did not recognize or understand that $x_{3,5}$ includes money made from investments $x_{1,3}$ and $x_{2,3}$ because the money that became available in year 3 was reinvested. In the constraint equations, many students were again coded with a VR.4 error because they had correct relationships on the left side of all equations but were missing information on the right that accounted for reinvestments from previous years (e.g. $x_{2,3} + x_{2,4} = 25{,}000$ in Fig. 7 should be $x_{2,3} + x_{2,4} = 25{,}000 + 1.04x_{1,2}$). It is likely that context played a role in the high number of errors on this problem, though difficulty with double-subscript notation may also be a factor. Additionally, many students used interest rates as coefficients (0.04 instead of 1.04) without adding the invested amount to the interest (coded as VR.2).

Proportional reasoning errors (VR.3) represent one of the larger and more interesting categories of codes (identified 19 times) in student work. This category represents cases where students made a "reversal error" and transposed the coefficients in a problem so that they were associated with the wrong variables. This error was found 15 times in the balance constraints in Q5, which required students to relate the number of palettes produced with components needed to

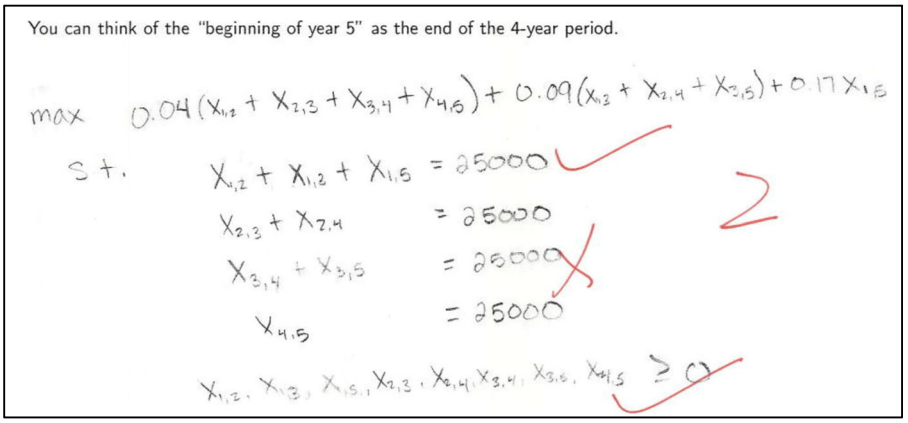

Fig. 7 Example of variable relationship errors VR.2 and VR.4 on Q2

make a palette. The correct constraints include $y_1 = 3x_1$ and $y_2 = 3x_2$, which show that the number of standard and long separators needed is 3 times the number of standard and long palettes, and $y_3 = 10x_1 + 18x_2$, which denotes that the number of crosspieces needed is 10 times the number of standard palettes plus 18 times the number of long palettes. However, 10 out of 15 students placed the coefficients on the wrong variables in the balance constraints: $x_1 = 3y_1 + 10y_3$ and $x_2 = 3y_2 + 18y_3$, incorrectly reversing the desired relationships. Proportional reasoning errors were also found four times in the ratio constraints on Q3. Instead of the correct constraint $p_2 \geq 0.2(p_1 + p_2 + p_3)$, which represents the fact that the mixture $p_1 + p_2 + p_3$ must contain at least 20% nuts (p_2), all four students coded with this error wrote $0.2p_2 \geq (p_1 + p_2 + p_3)$, reversing the ratio.

Code VR.5 (grouping by the wrong characteristics) represents cases where we felt we could make a claim about how or why students had incorrect constraints. We note here that, without talking to the students, we cannot be sure that these interpretations are correct. They are best guesses as to what contributed to the confusion for students. We saw evidence that students grouped decision variables by the wrong characteristics. This was identified 19 times across three different quizzes. For example, on Q1, students organized constraints by type of food rather than different nutrients (Fig. 8), and in Q2, students grouped values in the model by the length of investments rather than the year invested (Fig. 9).

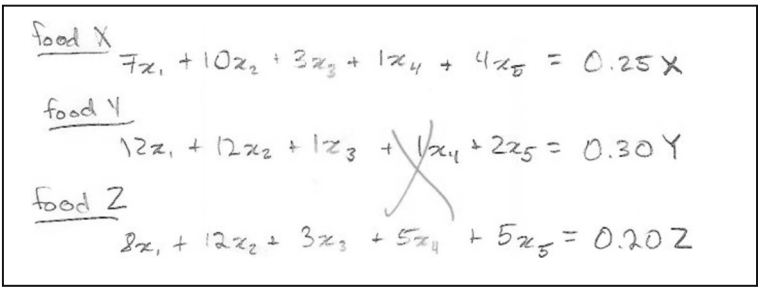

Fig. 8 Example of student grouping equations by the wrong characteristics in Q1

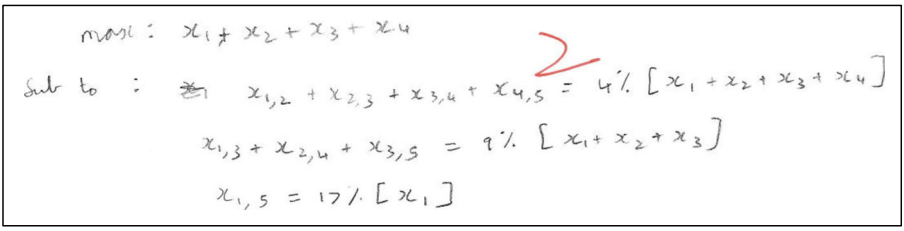

Fig. 9 Example of student grouping equations by the wrong characteristics in Q2

Notation Errors

Notation errors are specifically linked to difficulties with particular mathematical symbols. We identified errors related to subscripts (N.1), the summation symbol and *for* statements (N.2), and the inequality and/or equal signs (N.3 and N.4) in some way on all five quizzes. For example, on Q2, 11 out of 37 students demonstrated an incorrect use of the summation symbol and *for* statement (N.2). In this investment task, the given variable $x_{i,j}$ did not include all i and j values (e.g. there was no option to invest in year 1 for 3 years, so $x_{1,4}$ is not included in the model). Most students, however, wrote the constraint: $\sum_{j=2}^{5} x_{i,j} = 25,000$ for $i = 1, \ldots, 4$. The use of summation statements was common in class, and a sum of terms was required in this problem (e.g. $x_{1,2} + x_{1,3} + x_{1,5} = 25{,}000$). However, students seemed to apply the notation here without knowing its full meaning in this application.

Notation errors involving relation symbols (N.3 and N.4) also frequented student work. Students used an equal sign when an inequality was needed, or vice versa, or used the wrong inequality direction. It is reasonable to assume that the students in this study have a grasp of the meanings of an inequality and an equality, and so we consider these as potential careless errors or a misreading of the problem. For example, on Q3, the first three constraints involve an "at least" relationship, while the last two involve "at most." Some failed to notice this change. Though minor, we feel that notation errors point to issues in students' concern with communicating effectively using the language of mathematics.

Form Errors

On every quiz, some students failed to record the non-negativity constraints (e.g. x, y, $z \geq 0$). Though we recognize that this is likely due to oversight and not conceptual difficulties, we felt it was important to record these instances, which we found 19 times on 91 quizzes. These constraints are critical when applying algorithms to solve the LP problems. As teachers, we hope that students recognize the role they play in the model.

General Patterns Across Quizzes and Constraint Types

Analysis across the five quizzes suggests that difficulty levels vary among quizzes. For instance, all but four quizzes taken for Q2 had at least one error (33/37 = 89.2%). This is followed in rank by Q5 (17/30 = 56.7%), Q3 (17/33 = 51.5%), Q1 (15/35 = 42.8%), and Q4 (9/33 = 27.2%). As mentioned above, we suggest that the problem context and the

double-subscript notation are possible reasons for students' difficulty on Q2. This quiz was also responsible for the high occurrence of the error code VR.4, which represents 40 out of the 184 total codes (21.7% of all codes). DV.1 and VR.2 tied for the second highest occurrence (20/184 = 10.9%), while N.1 and N.3 had the fewest occurrences (4/184 = 0.02%). The overall occurrence for each error type is shown in Fig. 10.

We also looked across the quizzes for a relationship between constraint types and errors coded. Figure 11 shows percentages of distributions across the objective function and five types of constraints when we group the errors into their larger categories of decision variables, variable relationship, and notation errors (form errors are excluded here because they only represent non-negativity constraints). We found, for example, that 19 of the 28 (71.4%) decision variable errors coded were identified in objective functions. The prevalence of variable relationship errors in the objective function can be mostly attributed to Q2, where codes VR.2 and VR.4 had high occurrences. More interesting, perhaps, is the high percentage of both variable relationship and notation errors in the balance constraints that were found in both Q2 and Q5. The proportional reasoning error (VR.3) from Q5 contributes to the variable relationship errors here, while summation notation accounts for most notation errors.

Discussion

This paper presents our work to understand the different types of common experiences and errors that engineering students encounter in complex word problems in their curricula. This is a necessary first step in our larger research agenda for developing learning tools to help students better understand mathematical modeling. We have analyzed engineering students' work on five questions that represent typical problems encountered in an introductory course. We identified and categorized four different types of observable errors. This allows

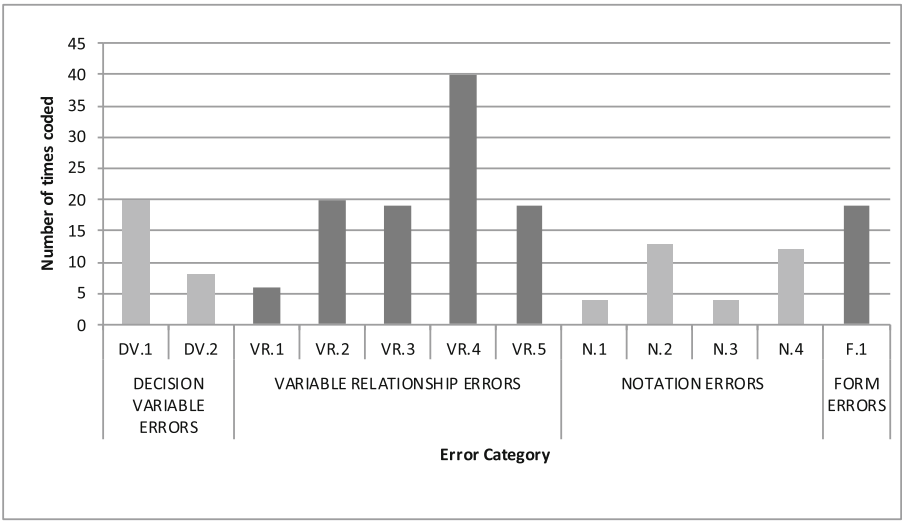

Fig. 10 The number of occurrences for each coded error type

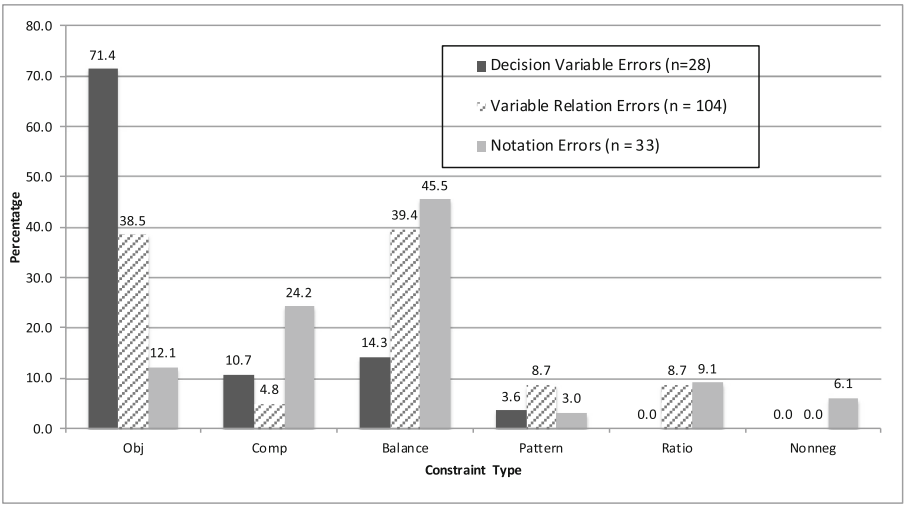

Fig. 11 Percentage of errors across constraint types

us some insight into the mathematical conceptions available to students upon which they build an understanding of modeling.

The *decision variable errors* that we identified fell into two categories: those that were unnecessary or undefined but still resulted in a semi-correct model and those that were disruptive to the modeling process. The latter had greater prevalence in students' work and are disconcerting. On all quizzes, students were given defined decision variables. This was done to narrow the scope of potential errors. However, in 20 cases across the quizzes, students ignored the given variables and created their own. We know that giving students a tool does not mean they will know how to use it. The symbols only develop meaning when used and interpreted by the student (Kinzel, 1999; von Glasersfeld, 1995), and students' previous experiences allow them to elicit symbols to use with which they are already familiar (Lesh & Doerr, 2003; Stacey & Macgregor, 1999). However, because decision variables are not typically given in practice, students need to learn how to choose useful variables and use them effectively in their models. Our findings suggest a need for instructors to draw explicit attention to the role that these variables play in models and in mathematical communication.

Some *notation* and *form errors* might be attributed to carelessness, though errors involving the summation symbol and accompanying *for* statements point to greater issues of understanding with these signs. The course instructor used summations often in examples, so it is natural that students would try to imitate him in their own work if they saw connections to prior experiences from class lectures. However, we saw evidence of these symbols being employed without reflection on what the mathematical statement meant to the given problem. The example of nearly one-third of the students writing $\sum_{j=2}^{5} x_{i,j} = 25,000$ for $i = 1, \ldots, 4$ on Q2 when values like $x_{1,4}$ were absent from the model is intriguing and warrants further investigation into students reasoning with these symbols.

Variable relationship errors were the most common across the quizzes. We might attribute some VR.2 errors (solutions that were missing some needed coefficients) and

VR.4 errors (solutions that failed to include some variables in the relationship) to potential confusion with the context of the investment problem in Q2 where the majority of these errors were identified. Coupling these errors with trivial relationships (VR.1), however, suggests that building clear relationships among multiple variables was not yet a strong component of some students' conceptions of these problems. They relied on familiar basic relationships between variables and numeric values, reducing the complexity of the problem.

In a similar fashion, grouping errors (VR.5) also suggest a reduction in complexity where students seemed to rely more on a direct translation of the problem into symbols. Future work should investigate whether, given the opportunity to reflect on the model by actually solving it, students would recognize this grouping error in their work. We suspect that this error could be related to tendencies for word problems in many traditional K-12 textbooks to lend themselves to direct translations, making this technique for creating a model one which is a familiar part of students' existing schemes.

Proportional reasoning errors (VR.3) resided with balance and ratio constraint functions. These types of constraints are used to enforce conservation of a quantity in a system and to enforce a defined proportional requirement between quantities, respectively. Thus, they are likely locations for such errors to occur. More importantly, these constraints are clear areas where instructors can pay attention to students' engagement in this type of error and take steps to help correct/prevent it. While previous literature on "reversal" errors makes our finding of such errors unsurprising (e.g. see Kim, Phang, An, Yi, Kenney, & Uhan, 2014; Cohen & Kanim, 2005; Weinberg, 2009), they still give us great concern. This error does not seem context-specific but instead represents a conceptual difficulty where students may understand the problem but cannot translate to a correct mathematical model. They demonstrate a tendency to reinterpret the given variables from, for example, x = *the number of* palettes to a static interpretation of just x = palettes, so that "3x" represents "3 palettes." The prevalence of this behavior with college engineering students (most of whom have already completed a calculus sequence at the university) suggests a need to identify ways of perturbing students' thinking in this area and help them adapt their current conceptions of proportional reasoning relationships.

In the large-lecture setting for the optimization course in which our participants were enrolled, the instructor tended to focus more on the creation of models rather than the solving of completed models (this part was more easily completed outside of class with software packages). This can put limits on the reflective behaviors that occur because models are not analyzed in relation to their solutions. Many of the errors identified here suggest that it may be beneficial for students to have ample opportunities to reflect on the structure of the models created and anticipate their usefulness for producing accurate solutions to a problem. As a member of the research team, the course instructor has subsequently made changes to his own teaching practices to include helping students learn to reflect on the reasonableness of the models they create.

These snapshots of students' work on classroom quizzes represent our work to understand what we consider initial representations of engineering students' cognitive images of models in LP problems. We believe that the common errors we have identified provide direction for addressing persistent student difficulties in LP modeling problems. Future work in this area should include interviews with students to unpack their thinking as they work through the model construction in complex word problems

like those analyzed here, and a focus on how students' conceptions of the models may change as they progress through the problem-solving stages.

Funding Information This material is based upon work supported by the Purdue Engineer of 2020 Seed Grant Program and the National Science Foundation (Grant No. 1044182).

References

Abrams, J. P. (2001). Teaching mathematical modeling and the skills of representation. In A. A. Cuoco (Ed.), *The roles of representation in school mathematics (2001 Yearbook)* (pp. 269–282). Reston, VA: NCTM.

Arcavi, A. (2005). Developing and using symbol sense in mathematics. *For the Learning of Mathematics, 25*(2), 42–47.

Ashlock, R. (2010). *Error patterns in computation: Using error patterns to help each student learn* (10th ed.). New York, NY: Pearson Education.

Borasi, R. (1994). Capitalizing on errors as "springboards for inquiry": A teaching experiment. *Journal for Research in Mathematics Education, 25*(2), 166–208.

Cifarelli, V. V. (1993). *Representation processes in mathematical problem solving*. Paper presented at the annual meeting of the American Educational Research Association, Atlanta, GA.

Cobb, P., Yackel, E., & Wood, T. (1992). A constructivist alternative to the representational view of mind in mathematics education. *Journal for Research in Mathematics Education, 23*(1), 2–33.

Cohen, E., & Kanim, S. E. (2005). Factors influencing the algebra "reversal error". *American Journal of Physics, 73*(11), 1072–1078. https://doi.org/10.1119/1.2063048.

Danielsiek, H., Paul, W., & Vahrenhold, J. (2012). *Detecting and understanding students' misconceptions related to algorithms and data structures*. Paper presented at the Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, Raleigh, NC.

dos Santos, S., & Brodlie, K. (2004). Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics, 28*(3), 311–325.

Driscoll, M. (1999). *Fostering algebraic thinking: A guide for teachers grades 6–10*. Portsmouth, NH: Heinemann.

Gagatsis, A., & Shiakalli, M. (2004). Ability to translate from one representation of the concept of function to another and mathematical problem solving. *Educational Psychology, 24*(5), 645–657.

Gainsburg, J. (2006). The mathematical modeling of structural engineers. *Mathematical Thinking and Learning, 8*(1), 3–36.

Goldman, K., Gross, P., Heeren, C., Herman, G., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2008). Identifying important and difficult concepts in introductory computing courses using a delphi process. *ACM SIGCSE Bulletin, 40*(1), 256–260.

Gray, E. M., & Tall, D. O. (1994). Duality, ambiguity, and flexibility: A "proceptual" view of simple arithmetic. *Journal for Research in Mathematics Education, 25*(2), 116–140.

Hershkowitz, R., Schwarz, B. B., & Dreyfus, T. (2001). Abstraction in context: Epistemic actions. *Journal for Research in Mathematics Education, 32*(2), 195–222.

Hillier, F. S., & Lieberman, G. J. (2014). *Introduction to operations research* (10th ed.). New York, NY: McGraw-Hill Education.

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development, 48*(4), 63–85.

Kaczmarczyk, L. C., Petrick, E. R., East, J. P. & Herman, G. L. (2010). *Identifying student misconceptions of programming*. Paper presented at the Proceedings of the 41st ACM Technical Symposium on Computer Science Education, Milwaukee, WI.

Kenney, R. H. (2015). Investigating a link between pre-calculus students' uses of graphing calculators and their understanding of mathematical symbols. *International Journal for Technology in Mathematics Education, 21*(4), 157–166.

Kim, S. H., Phang, D., An, T., Yi, J. S., Kenney, R. H., & Uhan, N. (2014). The role of visualization in breaking cognitive lock-up of mental models: Investigation through student-professor problems. *International Journal of Human-Computer Studies, 72*(1), 12–22.

Kingsdorf, S., & Krawec, J. (2014). Error analysis of mathematical word problem solving across students with and without learning disabilities. *Learning Disabilities Research & Practice, 29*(2), 66–74.

Kinzel, M. (1999). Understanding algebraic notation from the students' perspective. *Mathematics Teacher, 95*(5), 436–442.

Lesh, R., & Doerr, H. (2003). Foundations of a model and modeling perspective on mathematics teaching, learning, and problem solving. In R. Lesh & H. Doerr (Eds.), *Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching* (pp. 3–33). Mahweh, NJ: Erlbaum.

Lucangeli, 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.

Niss, M. (2010). Modeling a crucial aspect of students' mathematical modeling. In R. Lesh, P. L. Galbraith, C. R. Haines, & A. Hurford (Eds.), *Modeling students' mathematical modeling competencies* (pp. 43–59). New York, NY: Springer.

Panaoura, A., Gagatsis, A., & Demetriou, A. (2009). An intervention to the metacognitive performance: Self-regulation in mathematics and mathematical modeling. *Acta Didactica Universitatis Comenianae Mathematics, 9*, 63–79.

Perlow, L., & Bailyn, L. (1997). The senseless submergence of difference: Engineers, their work, and their careers. In S. R. Barley & J. E. Orr (Eds.), *Between craft and science: Technical work in US settings* (pp. 230–243). Ithaca, NY: ILR Press.

Priem, J. (2010). Fail better: Toward a taxonomy of e-learning error. *Journal of Educational Computing Research, 43*, 377–397.

Rardin, R. L. (2016). *Optimization in operations research* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Sebrechts, M. M., Enright, M., Bennett, R. E., & Martin, K. (1996). Using algebra word problems to assess quantitative ability: Attributes, strategies, and errors. *Cognition and Instruction, 14*(3), 285–343.

Simon, M., Tzur, R., Heinz, K., & Kinzel, M. (2004). Explicating a mechanism for conceptual learning: Elaborating the construct of reflective abstraction. *Journal for Research in Mathematics Education, 35*(5), 305–329.

Smith, J. P., DiSessa, A. A., & Roschelle, J. (1993). Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The Journal of the Learning Sciences, 3*(2), 115–163.

Sokol, J. (2005). Teaching OR modeling. *OR/MS Today, 32*(6), 12–13.

Stacey, K., & MacGregor, M. (1999). Learning the algebraic method of solving problems. *The Journal of Mathematical Behavior, 18*(2), 149–167.

von Glasersfeld, E. (1995). *Radical constructivism: A way of knowing and learning*. New York, NY: Routledge.

Wang, W., & Brooks, R. J. (2007). *Empirical investigations of conceptual modeling and the modeling process*. Paper presented at the 39th conference on Winter Simulation: 40 years! The Best Is Yet To Come, Washington, DC.

Weinberg, A. (2007). *New perspectives on the student-professor problem.* Paper presented at the 29th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education, Lake Tahoe, Nevada.

Weinberg, A. (2009). *Students' mental models for comparison word problems.* Paper presented at the 31st annual meeting of the North American Chaper of the International Group for the Psychology of Mathematics Education, Atlanta, GA.

Wiens, A. (2007). *An investigation into careless errors made by 7th grade mathematics students* (Master's thesis). University of Nebraska-Lincoln, Lincoln, United States. Retrieved from http://digitalcommons.unl.edu/mathmidsummative/32. Accessed 1 October 2018.

Winston, W. L. (2003). *Operations research: Applications and algorithms* (4th ed.). Belmont, CA: Brooks/Cole.

Yazdani, M. A. (2008). The limitations of direct sentence translation in algebraic modeling of word problems. *Journal of Mathematical Sciences and Mathematics Education, 3*(2), 56–61.