IMPOSING STRUCTURE ON LINEAR PROGRAMMING PROBLEMS: AN EMPIRICAL ANALYSIS OF EXPERT AND NOVICE MODELS

Wanda Orlikowski
and
Vasant Dhar

July 1986

Center for Research on Information Systems
Information Systems Department
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #122

GBA #86-59

This paper appears in the **Proceedings of the National Conference on Artificial Intelligence**, Philadelphia, Pennsylvania, August 1986 (AAAI-86).

Table of Contents

ABSTRACT

Research on expert-novice differences falls into two complementary classes. The first assumes that novice skills are a subset of those of the expert, represented by the same vocabulary of concepts. The second approach emphasizes novices' misconceptions and the different meanings they tend to attribute to concepts. Our evidence, based on observations of problem solving behavior of experts and novices in the area of mathematical programming, reveals both type of differences: while novices are to some extent underdeveloped experts, they also attribute different meanings to concepts. The research suggests that experts' concepts can be characterized as being more *differentiated* than those of novices, where the differentiation enables experts to categorize problem descriptions accurately into standard archetypes and facilitates attribution of correct meanings to problem features. Our results are based on twenty-five protocols obtained from experts and novices attempting to structure problem descriptions into mathematical programming models. We have developed a model of knowledge in the LP domain that accommodates a continuum of expertise ranging from that of the expert who has a highly specialized vocabulary of LP concepts to that of a novice whose vocabulary might be limited to high school algebra. We discuss the normative implications of this model for pedagogical strategies employed by instructors, textbooks and intelligent tutoring systems.

1. Introduction

Analytical modeling techniques constitute an important component of the curriculum of Operations Research, Industrial Engineering and Management schools. In particular, mathematical programming models such as linear programming (LP) have proved useful in solving many real-world problems. However, structuring open-ended problem descriptions into formal LP models is not a straightforward task. We have found that despite having taken courses in linear programming, students are often unable to frame even relatively simple problem descriptions into appropriate mathematical programming models. We conjecture that one important reason for this situation is an overly normative orientation in instruction which arises out of pragmatic considerations – an instructor with limited contact hours may be unable to take cognizance of students' "naive conceptions" of the material, and may focus only on presenting the "correct" modeling formalisms. On the other hand, a good *tutor* is sensitive to the student's conceptualization of a problem, in detecting the lack of congruence between the student's conceptual system and the "correct one" (assumed to be the tutor's), and in eliminating the mismatch between the two. This requires knowledge about the domain (a model of *expertise*), knowledge about the novice (a *novice model*), and *tutoring strategies* that help "remodel" the novice.

A pragmatic long term goal of this research is to develop an instructional system in the domain of Mathematical Programming that will effect the novice-expert transition. Achieving this overall objective requires developing the three knowledge components mentioned above. In this paper, we have attempted to understand the first two. We have developed an abstract theoretical model of knowledge that expresses "levels of expertise" in the domain of mathematical programming. It casts novice and expert models in terms of a common set of "concepts" where a more elaborate *differentiation* of a concept is associated with more expertise in that part of the problem domain covered by the concept. Differentiation is a construct that is used widely by developmental psychologists and historians of science for contrasting progressive "conceptual systems" (Wiser and Carey, 1983). For example, in the history of science, Kuhn (1977) provides, among other cases, the shift from Aristotle's conception of velocity to Newton's, which differentiated it into instantaneous and average velocities. This differentiation was a necessity in Newtonian physics whereas the more diffused notion of velocity was adequate, albeit limiting, in Aristotelian physics. We have found that expressing knowledge in terms of two types of differentiation, namely *structural* and *semantic* differentiation, provides a good theoretical foundation for contrasting expert and novice behavior in the mathematical programming domain. Structural differentiation involves elaboration of LP concepts or generic terms into fine grained ones. Semantic differentiation involves attributing different meanings to features of a problem description during the process of formulation. We illustrate these two types of differentiation in section 3 following analysis of the data.

Our model is based on the results of an empirical investigation of problem solving processes of experts and novices solving linear programming problems. Before describing the model, we discuss prior research that has motivated it. Section 3 contains the results of the study and the differentiated model of knowledge that explains expert/novice differences. Pedagogical implications of this view of expert/novice differences are discussed in section 4.

2. Prior Research

Prior research into expert-novice differences has been characterized by two complementary approaches. The first focuses on the differences among experts and novices in terms of their relative abilities to categorize problem descriptions into standard abstractions (Chi et al. 1981; Larkin et al. 1980; Larkin 1983; Wiedenbeck 1985.). Within this view, experts are characterized as employing abstract schematic representations that enable mapping problem descriptions into a "deep structure", while novices, lacking such abstractions, typically fail to progress beyond the superficial problem features (Chi et al. 1981). Other research has found that experts "automate" the simple aspects of the problem solving procedures they employ, while novices have certain difficulties performing the simplest stereotypical procedures (Wiedenbeck 1985; Hinsley 1983).

The second approach focuses on the *misconceptions* or errors made by novices in attempting to solve some problems. The premise of this approach is that novice and expert models of a domain are fundamentally different in that they *interpret* the same terms in the domain very differently, not unlike pre and post paradigmatic situations in the history of science (Clement 1983; diSessa 1983; McCloskey 1983; Wiser & Carey 1983). McCloskey (1983;pp.318-319) provides evidence that novices frequently misinterpret terms taught to them, distorting them to fit their naive theories about the domain. He suggests that these naive theories are strongly held and might not be easily remodeled by mere presentation of expert concepts and strategies. Wiser and Carey (1983) suggest that the *shift* from novice to expert might be characterized along the lines of the scientific paradigm shift (Kuhn 1977). The dimensions suggested by them for contrasting the conceptual differences between novice and expert are *differentiation* -- where the previously diffused understanding of a phenomenon is replaced by multiple finer-grained or more accurate concepts, *coalescence* -- where prior understanding of a phenomenon is recognized as involving redundant categories which are collapsed into a single category in the revised theoretical framework, and the *shift* from *property* to *relation* -- where a phenomenon that was previously viewed in isolation becomes related to others in the domain.

While it is not clear whether the novice-expert shift can be characterized in the same way as a paradigmatic shift in the history of science, we have found the notion of differentiation of concepts among novices and experts to be a useful theoretical construct along which to express and contrast expertise. Structural and semantic differentiation allow us to interpret the behavioral differences between experts and novices observed in both types of abovementioned studies. We have found the other concepts of coalescence and property-relation shift less useful in classifying mathematical programming knowledge although they may be useful in highlighting novice-expert differences in other domains.

3. Experimental Results

Two experts and three novices were chosen to participate in the experiment. The experts were a professor and a graduate student, both with considerable knowledge and experience in management science. The novices consisted of graduate students who had successfully completed one or more courses in introductory mathematical programming. Five simple[1] problem scenarios selected from a fundamental mathematical programming textbook (Wagner, 1975) were chosen. These problems represent a reasonable spectrum of LP problem types: one fluid blending problem, one feed-mix problem, one dynamic programming problem and two transportation problems (the Appendix shows a blending and a transportation problem). A total of twenty-five observations were obtained. The subjects were asked to formulate each of the problems into LP models and were specifically requested not to try and solve the problems, since formulations, expressed in some suitable form, can be solved using standard procedures such as the simplex algorithm.

Fundamentally, formulation involves *imposing a formal structure* on a problem. This requires strategies for factoring out the complexity into manageable components. Our observations reveal three types of effective strategies used by experts for handling complexity, contrasted with the methods employed by novices that preclude a coherent structuring of a problem description. In this study, not a single one of the fifteen novice formulations was correct. The errors were not trivial, but revealed a lack of knowledge about certain important concepts, as well as a diffused understanding of certain specialized LP concepts.

The results can be summarized as follows: first, our findings support those of Chi et. al (1981), Larkin (1983) and others where experts were found to employ *categorization* of problems into standard types based on underlying principles of the domain (1981:p.150). The complementary finding that novices tend to handle problems according to the entities present in the problem

[1] We classify problems as simple if they correspond to standard types such as the transportation problem or blending problem. Complex real-world problems typically involve combinations of these standard types.

description (1981:p.150) was also confirmed in the novice protocols. Secondly, the choice of an appropriate *decision variable* by experts was found to be central in reducing problem complexity. When appropriately subscripted, a decision variable *relates* the various problems dimensions thereby enabling a "holistic" approach toward structuring the problem. In contrast, novices employed several erroneous complexity-reduction strategies, partly in order to compensate for poor choices of decision variables. The third interesting finding was the detection among experts of a form of *dimensional analysis* which was not part of the novices' repertoire. It involves simple manipulations of the *units* associated with the variables (i.e. $, $/lb) in a way that influences the actual construction of algebraic expressions. In effect, the "semantic" information in the units is used as a means of shaping or validating expressions in the formulation. This aspect of expertise has not been stressed in the literature, but we find it particularly relevant to the formulation of models in this domain.

3.1. Imposing Structure Via Categorization

Chi et al. (1981;p.150) suggest that experts begin problem solving by first attempting to categorize the problem from a brief analysis of the problem statement. This problem analysis typically yields category names that serve as labels to access appropriate internal schemata. To illustrate categorization, consider Exhibit 1 line 18 where the expert begins the formulation by stating "**It's an allocation problem**" which he refines subsequently to "**a blending problem**", followed by a description of the "deep structure" of the blending problem (Exhibit 1, line 24):

> **"You could liken these variables (amounts of exposure) to amounts of ingredients where each ingredient (each advertising medium) supplies a certain amount (media effectiveness) of a certain thing."**

The parenthesized phrases correspond to terms in the actual problem description. As we can see, the expert attempts to map the surface features of the problem description into the structure of the identified category. In effect, a recognition of the "blending problem" initiates a search for problem dimensions, namely, ingredients to be blended (in this case, media) into the product (in this case, exposures of audiences). These findings are similar to those from the experiments of Simon and Hayes (1976) on problem isomorphs.

In contrast, novice statements such as (Exhibit1, line 1):

> **"It's an advertising problem"**

suggest the lack of any such categorization, and hence the inability to impose structure on the problem description. This finding was common to all novice cases. The lack of a schema within which to interpret the problem also appears to increase the chance of the novice forming *misconceptions* about the problem features, that is, of attributing incorrect meanings to them. For

Novice 1, problem 1

1. Its an advertising problem.
2. Identify the audiences. Three audiences X1,X2,X3.
3. OK, so we have our specification, which is the level of exposure per audience type.
4. Audiences and media are all I've got.
5. I'm calling audiences X1,X2,X3.
6. I'm calling media Y1,Y2,Y3.
7. I've got some coefficient for each.
8. So, A1Y1, A2Y2, A3Y3. If I add all these together, I guess I'll get some sort of exposure for each of these groups. I'll call that an exposure level.
9. So we have exposure level 1, level 2, level 3 for X1, X2 and X3.
10. Somehow, the exposure levels.....exposure level for X1 plus exposure level for X2 plus exposure level for X3 has to be equal or less than the total money...
11. Hmm, if I had only one audience, say X, and I took care of that first...

Novice 2, Problem 4

12. Alright, so we've got a cost function.
13. We have to maximize amounts available from each company, Company A = 275,000, ...
14. So, we have to minimize amounts at each airport. You want to start with airport 4, and see how you can meet its requirements from company B, then see where the next lower price is, and so on...

Novice 3, Problem 4

15. OK, company A gives 10, B gives 7, OK I have the cost data.
16. So, how much should I buy from...whether I should buy from company A, or B, or C at the different locations.
17. If I say I want 110,000 gallons at airport 1, and I pick company B, then B can provide just fine. Company A... can provide 275,000 to say airport... fine. So, X from A, Y from B, Z from C.

Expert 1, Problem 1

18. It is an allocating problem... blending problem.
19. Allocating dollars, so...looks like the decision variables are going to $ to teens, $ to married, etc.
20. So, let us try and formulate the problem in terms of the decision variables. Two requirements, each row is an audience category in the tableau...
21. OK, so there's a coefficient here -- the effectiveness for each of the variables on each of these audiences (i.e. $A_{i,j}$).
22. Coefficients could be $/number of people, ok, the product should be $.
23. OK, so here's the formulation (Shows the formulation as "Minimize Cx" such that Ax > B).
24. Its a blending problem -- you could liken those variables to amounts of ingredients where each ingredient supplies a certain amount of a certain thing to the product.

Expert 1, Problem 4

25. OK, so some of the decision variables are the amount of oil bought from A, B, C. So, $\Sigma X_{A,j} < 275$ etc. Upper bounds on these variables.
26. This is a transportation problem.
27. OK, so we've got to change $X_{A,j}$ to $X_{i,j}$; index i is vendors, j is indexing airports. So now, constraints on availability from vendor i to be added to the general transportation problem.
28. So it is a minimize $\Sigma C_{A,j}$...oops $\Sigma C_{i,j} X_{i,j}$ with these constraints -- classic transportation problem.

Novice 2, Problem 1

29. So, we'll start by creating a chart with the clients on one axis and the different kinds of exposure on the other. Then in the boxes I'd put the score that each vehicle has for each target group.
30. Now, let us see...you'll have to express the desired level of exposure for each audience. So, you'll have to express that somehow, probably in percentages, that is, the score is a percentage of the exposure you want for the group.

EXHIBIT 1: Excerpts of Expert and Novice Protocols for Problems 1 and 4.

example, in Exhibit 1, line 10, the novice actually interprets problem data that should be in the objective function (i.e. minimize cost) as belonging to the right hand side of a constraint.[2]

3.2. Imposing Structure Via "Holistic Reduction"

In mapping surface problem descriptions into categorizations, experts invariably express the problem in terms of subscripted variables which capture the various dimensions of the problem (such as buyers and suppliers in problem 4) simultaneously. In this way, the problem is reduced "holistically" in that recognizing the right subscripts imposes significant constraints on the overall problem structure – to the point that surface features become "irrelevant" and can be added or deleted without affecting the complexity of the problem description. For example, in problem 4, once the expert establishes the variables $\Sigma X_{i,j}$ as representing the amount of oil from i to customer j, the actual number of vendors and customers in the problem become irrelevant to the abstract formulation since the constraint

$$\forall_i,\ \Sigma X_{i,j} \geq B_i$$

captures the abstract relationship (the minimum amounts of oil to be supplied) between the problem dimensions (in this case, suppliers and buyers). Adding more suppliers or buyers does little to affect the complexity of the problem.

In contrast, superficial features of the problem add considerable complexity for the novice. Consider the following segment:

"Let's say X = teenagers, Y = married couples, Z = geriatric group"

which illustrates that what is lacking is an appreciation that the three elements in the problem statement represent *one dimension* of the problem for which a single dimensional variable is more appropriate. In this case, adding more suppliers and buyers leads to more variables and relationships among them, which tends to add complexity for the novice. A related observation was that novices attempted to remove the complexity they could not deal with by artificially "simplifying" the problem by reducing the number of elements in a dimension to one (this amounts to removing the need for a subscript, Exhibit 1, line 11):

"If I had only one audience, say X, and I took care of that first..."

Another example of this type of sequential simplification was a transformation of an inequality into an equation thereby constraining the problem drastically and introducing a sub-optimal solution, as illustrated in the following excerpt from problem 4 (Exhibit 1, line 17):

"If I say I want 110,000 gallons at airport 1, and I pick company B, then B can provide just fine."

A third type of simplification leading to sub-optimal solutions was that of sequential

[2] Actually, this is sometimes a legitimate strategy employed by experts in multiobjective programming problems, but it is highly unlikely that the novice was thinking in these terms.

decomposition, an attempt to break the problem down into isolated parts and then to handle each part separately (Exhibit 1, line 14):

> **"You want to start with airport 4 and see how you can meet its requirements from company B, then see where the next lower price is, an so on."**

Finally, a fourth simplification strategy is not to introduce a decision variable at all, but to try and solve the problem arithmetically – a phenomenon similar to one observed by Matz (1983). In line 29 for example, the novice correctly identifies the cost coefficient, but in line 30, she attempts to insert it into the formulation in place of the decision variable. The upshot of this is that the novice can neither arrive at the formulation because no variables are introduced, nor solve the problem because the algorithm required to obtain an optimal solution is too complex procedurally.

3.3. Imposing Structure Via Dimensional Analysis

A final method in the expert's problem structuring repertoire is one of ensuring consistent dimensionality among the units in the algebraic expressions. Casting the problem features into appropriate units not only makes explicit what the decision variables and coefficients stand for, but also constrains how they combine (i.e., multiply, add) with other variables. For example, in problem 1, the expert deliberates on the units of "advertising effectiveness", pointing out that if this is specified in terms of $/person, then the units of the decision variable must be "# of people" (since these are to be multiplied, and the units to be maximized are $). On the other hand, an "advertising effectiveness" without units would require a decision variable to be expressed in terms of dollars. In effect, the choice of units – an area in which there can be considerable discretion – constrains what the units of other variables and coefficients can be, which also clarifies their problem-specific interpretations.

In contrast, in all fifteen cases, novices were unable to arrive at a consistent interpretation of units. A typical example was as follows (Exhibit 1, line 10):

> **"Somehow, the exposure levels...exposure level for Xl, plus exposure level for X2, plus exposure level for X3 has to be equal or less than the total money."**

To summarize, ensuring consistent units, appropriate subscripting, and rapid categorization enable an expert to impose a correct structure on a problem description. In contrast, the lack of an appreciation of standard archetypes, simplistic complexity reduction strategies and a disregard for units makes it virtually impossible for the novices to arrive at an appropriate formulation. In the following subsection, we summarize the findings in terms of a model of expertise that makes some of the distinctions explicit.

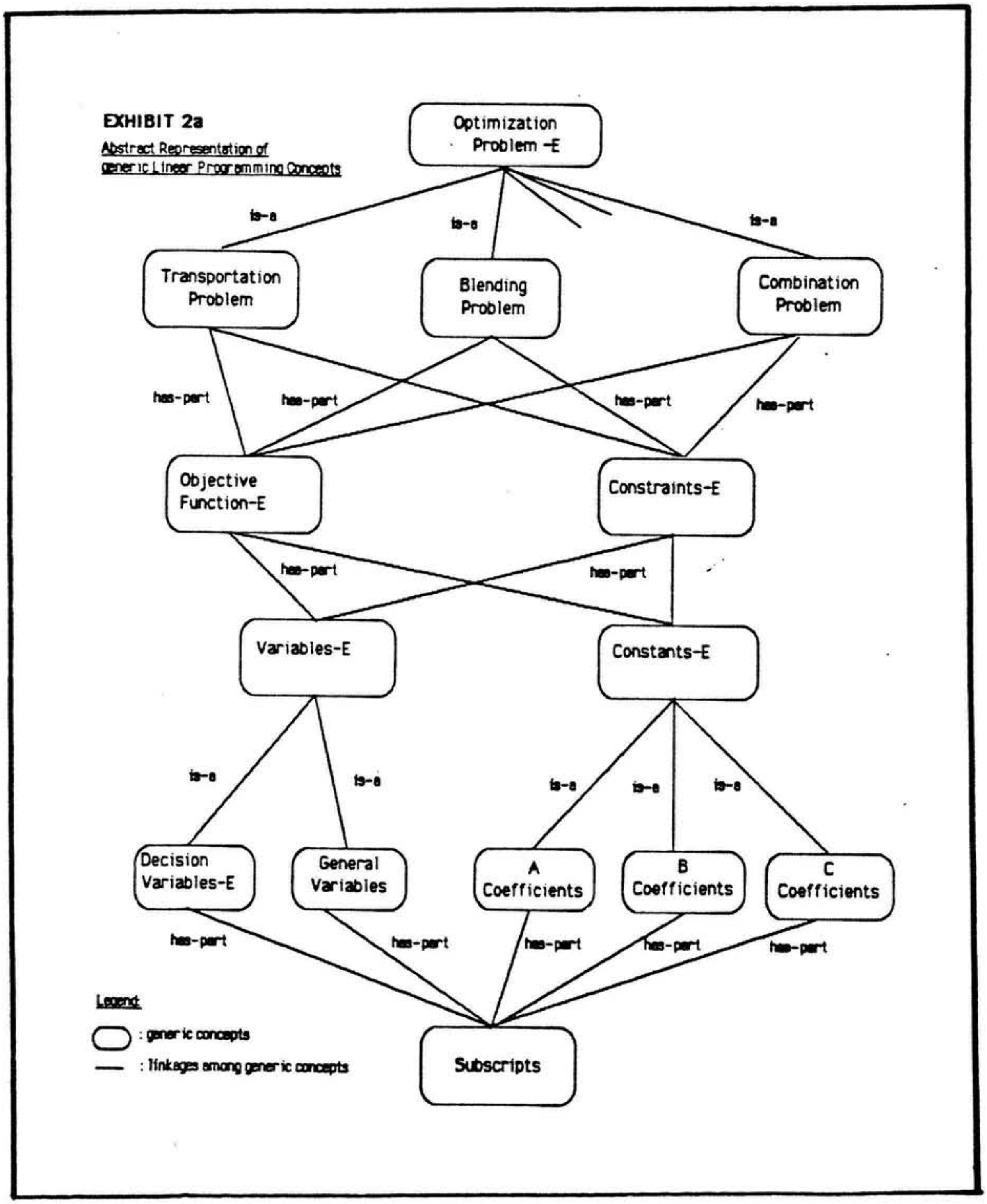
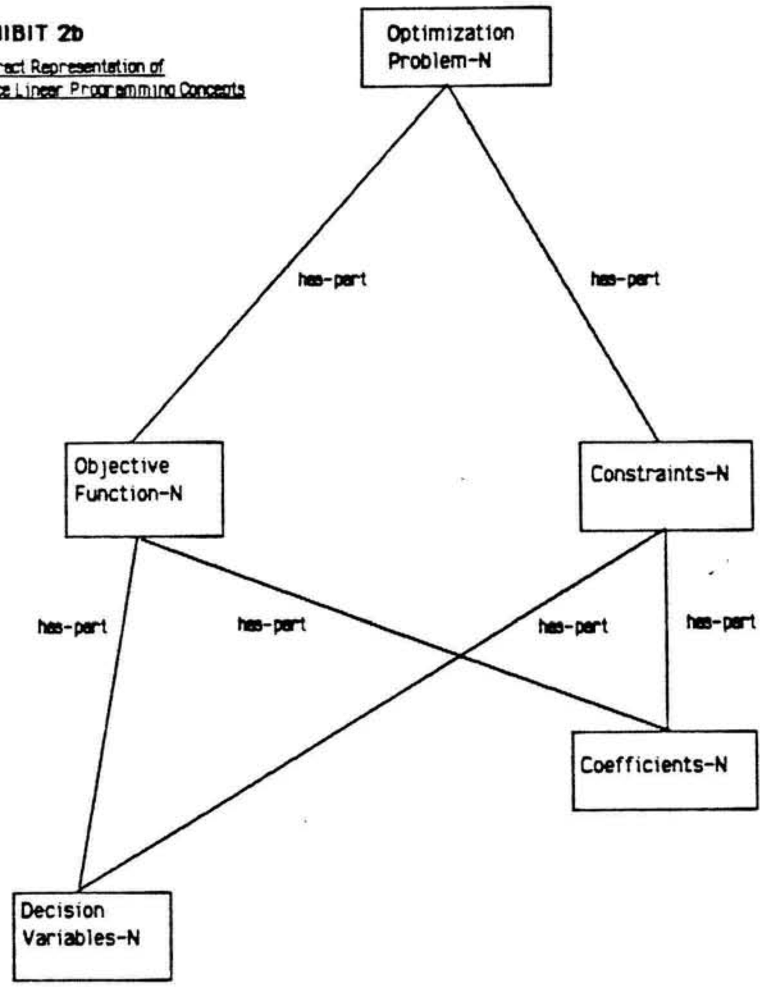


EXHIBIT 2a
Abstract Representation of generic Linear Programming Concepts
Optimization Problem -E
is-a
is-a
is-a
Transportation Problem
Blending Problem
Combination Problem
has-part
has-part
has-part
has-part
Objective Function-E
Constraints-E
has-part
has-part
Variables-E
Constants-E
is-a
is-a
is-a
is-a
is-a
Decision Variables-E
General Variables
A Coefficients
B Coefficients
C Coefficients
has-part
has-part
has-part
has-part
has-part
Subscripts
Legend:
: generic concepts
: linkages among generic concepts

EXHIBIT 2b

Abstract Representation of
novice Linear Programming Concepts



Legend:

☐ : novice concepts

— : linkages among novice concepts

3.4. Summary: A Differentiated Model of Knowledge

It appears that the observed differences among novices and experts can be explained in terms of the two types of differentiation mentioned at the outset of this paper. The first, *structural differentiation* refers to the elaboration or decomposition of LP concepts into finer grained ones. Based on our results, two levels of differentiation of terms are apparent among experts that are absent among novices. These are illustrated in Exhibit 2a and 2b. The first level of differentiation is at the level of the "optimization problem" itself. As we discussed in the previous subsection, experts try to characterize descriptions into one among several standard types. Novices, lacking these differentiations, must rely on a more bottom-up strategy for arriving at the objective function and constraints which, taken together, represent a formulation.[3] Another level of differentiation occurs at the level of variables and constants. Experts have a precise conception of decision variables and constants. The decision variable is what must be computed to maximize or minimize some objective function. When appropriately subscripted, it relates the problem dimensions. Similarly, constants are of three types: the C matrix -- which is a vector of multipliers used in the objective function, the A matrix -- a vector multiplier on the left hand side of constraint inequalities, and the B matrix -- a vector reflecting certain constraint levels on the right hand side. Experts search for problem features to match these vectors. To a novice, however, a decision variable is not differentiated from a variable, and a coefficient is nothing more than some constant. Fundamentally, what is lacking is a clear concept of vectors and vector multiplication. As a consequence, the novice has little chance of synthesizing an appropriate formulation in terms of scalars unless enough of them are introduced in order to capture the dimensionality of the problem.

The second type of differentiation, which we term *semantic differentiation*, refers to the process of associating the problem features (some of which may be implicit in the problem description) with the appropriate semantic labels. For experts, categorization appears to serve as an anchor around which problem features can be interpreted, whereas novices are more susceptible to interpreting problem features incorrectly. For example, it is inappropriate for the "total cost" to appear as a constraint in the blending problem because the *objective* for such a problem is cost minimization. However, lacking general knowledge about this class of problems, the novice interprets the objective as a constraint. In addition, inappropriate interpretations of problem data appear to result from an inadequate understanding of concepts such as decision variables and coefficients. In effect, although novices might *refer* to decision variables, coefficients, objective functions, etc., the

[3] It should be noted that we distinguish between novice and expert concepts (i.e. decision-variable-E versus decision-variable-N) because the problem features that are associated with these may differ among novices and experts. Because of this, instantiations of the schematized concepts in Exhibit 2 for novices and experts solving the same problem can contain different problem features.

meanings attached to these can be inappropriate (misconceptions) resulting in incorrect interpretations of problem features. In the following section, we examine the implications of these findings for instruction in LP.

4. Discussion

The underlying assumption of early normatively oriented CAI (Computer-Aided Instruction) systems was that the novice would assimilate the material presented by somehow eliminating previous conceptions or biases if these existed, and that the correct conceptual categories and meanings would be established. More recent tutoring systems, that maintain explicit student models, have typically adopted one of two approaches for representing knowledge about the student. In the first approach, a student model is synthesized by comparing the student's behavior to that of the expert model. This student modeling approach has been termed an *overlay* model (Goldstein, 1983), as the student's knowledge is represented entirely in terms of the expert's. GUIDON (Clancey, 1981) uses such an overlay model. Novices and experts are assumed to share the same underlying structural and semantic models, with the novice model assumed to be a subset of the expert model. A different approach to the modeling of student knowledge is to recognize that the student's knowledge is not a subset of the expert's, but a *perturbation* or deviation from the normalcy of the expert's knowledge, that is, in terms of bugs (Burton, 1983; Sleeman, 1983). Such systems attempt to identify the student's mistakes and classify them according to the assumed misconception underlying the mistakes.

The results of our study suggest that LP tutoring requires both approaches to modeling expert and novice knowledge. It is based on the observation that the LP domain models of novices and experts are distinct, and in particular can reflect significant conceptual differences in (a) the absence of certain concepts from novice models, and (b) the meanings attached to existing concepts and relationships. From a tutoring standpoint, the challenge is one of differentiating novices' notions of LP concepts from those of high school algebra, and establishing precise meanings of these concepts (i.e. variables, vectors and scalars).

Current introductory textbooks and course instruction do not seem to guarantee the development of adequate differentiations (as our novices' protocols and a perusal of basic LP textbooks has revealed). Based on our results we offer two recommendations for improving LP tutoring. First, we hypothesize that the differentiation of LP problems into categories helps substantially in reducing misconceptions by decreasing the novices' sensitivity to superficial problem features. For this reason, we suggest that tutoring media foster the development of explicit problem schemata. Second, we suggest that a sharper distinction be made between the concepts of algebra and LP.

Much of the lack of semantic differentiation we detected, could be attributable to an inadequate transition from algebraic to LP domains, where terms and operations assume specialized meanings. For example, strategies such as sequential substitution of variables for solving simultaneous equations cannot be used for solving LP problems. These simplification strategies are a consequence of novices' inability to deal with the problem dimensions simultaneously, due to their diffuse understanding of simultaneous optimization problems, and in particular, the concept of multi-dimensional decision variables and vector operations on them. This contributes to their being overwhelmed by the multi-dimensionality in even simple LP problems. Based on this evidence, it appears that textbooks that present LP without first establishing the foundations of the concepts and mechanics of linear algebra are unlikely to help novices in differentiating the meanings of the specialized LP concepts from those of high school algebra. As the next step in this research, we are about to conduct an in-depth empirical investigation of how exactly experts and novices conceptualize terms that appear in LP. Following this, we shall study the student/human-tutor interactions in the LP domain in order to understand how good tutors resolve novice misconceptions. Modeling this interaction will provide us with a sound base for constructing an intelligent tutoring system in the LP domain.

REFERENCES

Burton R.R. Diagnosing Bugs in Simple Procedural Skills, *Intelligent Tutoring Systems*, Sleeman D. & Brown J.S. (eds.), Academic Press, London, 1983, pp. 157-183.

Chi, Michelene T.H., Feltovich Paul J. & Glaser Robert, Categorization and Representation of Physics Problems by Experts and Novices, *Cognitive Science*, Vol. 5, June 1981, pp. 121-152.

Clancey, William J., Methodology For Building An Intelligent Tutoring System, Report #STAN-CS-81-894, Dept. of Computer Science, Stanford University, Stanford CA, October 1981.

Clement, J., Students' Preconceptions in Introductory Mechanics, *American Journal of Physics*, 50, 1982.

diSessa, A., Phenomenolgy and the Evolution of Intuition in *Mental Models*, Gentner D.R. & Stevens (eds.), LEA, 1983,

Goldstein I.P., The Genetic Graph: A Representation for the Evolution of Procedural Knowledge, *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983.pp. 51-77.

Hinsley, D.A., Hayes, J.R., & Simon, H.A., From Words to Equations: Meaning and Representation in Algebra Word Problems, in *Cognitive Processes in Comprehension*, Carpenter and Just (eds), Erlbaum, 1978.

Kuhn, Thomas., *The Structure of Scientific Revolutions*, Chicago University Press, 1977.

Larkin, J., Problem Representation in Physics, in *Mental Models*, Gentner D.R. & Stevens (eds.), LEA, 1983.

Larkin J.H., McDermott J., Simon D.P. & Smith H.A., Expert and Novice Performance in Solving Physics Problems, *Science*, Vol. 208, 1980, pp. 1335-1342.

Matz M., Towards a Generative Theory of High School Algebra Errors, *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983. pp. 25-50.

McCloskey M., Naive Theories of Motion, in *Mental Models*, Gentner D.R. & Stevens (eds.), LEA, 1983.

Simon H., and Hayes J.R., The Understanding Process: Problem Isomorphs, *Cognitive Psychology*, 1976,8, pp. 165-190.

Sleeman D.H., Assessing Aspects of Competence in Basic Algebra, *Intelligent Tutoring Systems*, Sleeman and Brown (eds), Academic Press, 1983. pp. 185-199.

Wagner H.M., *Fundamentals of Operations Research*, Addison-Wesley, 1975.

Weidenbeck Susan., Novice/Expert Differences in Programming Skills, *International Journal of Man-Machine Studies*, 1985.

Wiser M., and Carey S., When Heat and Temperature Were One, in *Mental Models*, Gentner D.R. & Stevens (eds.), LEA, 1983.

APPENDIX:

Problem 1

An account executive, Lotta Billings of the Flag-Pole Advertising Co. has announced that she can optimally allocate her clients' advertising dollars by means of linear programming. Her approach is to identify the various audiences the client wants addressed, such as teenagers, young married couples, the geriatric group, etc. The client specifies a desired level of exposure for each audience. There are various advertising vehicles (e.g. Magazines, TV spot commercial, color ads in a Sunday newspaper etc.). Each is scored for its effectiveness in each of the identified audience categories. Her clients' objective is to minimize the total advertising expenditure while still meeting the desired levels of product exposure.

Problem 4

The purchasing agent of the Fly-by-Night Airline must decide on the amounts of jet fuel to buy from three possible vendors. The airline refuels its aircraft regularly at the four airports it serves. The oil companies have said that they can furnish up to the following amounts of jet fuel during the coming month: 275,000 gallons from Oil Company A; 555,000 gallons from Oil Company B; and 660,000 gallons from Oil Company C. The required amount of jet fuel is: 110,000 gallons at Airport 1; 220,000 gallons at Airport 2; 330,000 gallons at Airport 3; and 440,000 gallons at Airport 4. When transportation costs are added to the bid price per gallon supplied, the combined costs per gallon for jet fuel from each vendor furnishing a specific airport is shown in the following table:

	Company A	Company B	Company C
Airport 1	10	7	8
Airport 2	10	11	14
Airport 3	9	12	4
Airport 4	11	13	9