Development of a Systematic Problem Solving Course: An Alternative to the Use of Case Studies

Karl A. Smith
Alan Wassyng
Anthony M. Starfield

Department of Civil and Mineral Engineering
University of Minnesota
Minneapolis, Minnesota

Summary

A major problem facing undergraduate engineering education is how to teach concepts, techniques and problem solving simultaneously. Periodic use of case studies added to engineering courses has been suggested as one means of addressing this problem. This paper describes an alternative approach which relies on structured tutorials to introduce new problems and on specially prepared interactive computer packages to aid in the solution of slightly open-ended problems. This approach achieves most of the benefits resulting from the use of case studies but does not require as large an investment of student time.

Introduction

Engineering educators are beginning to realize that the engineering curriculum has perhaps swung too far towards analysis, leading to a separation of theory and design and producing students who understand mathematical techniques but not how to use them in an engineering context. Recent findings in cognitive science have also highlighted some of these problems[1], including: (1) in physics and other sciences, even students who do well on textbook problems often, cannot apply the laws and formulae they have been drilled on to interpret actual physical events, and (2) the mathematical problem-solving skills of American students lag far behind their calculation abilities.

These trends and resultant concerns have created a renewed interest in the case studies approach to engineering education[2]. There are, however, some problems with the conventional use of case studies:

(1) Unless well designed, cases do not reinforce the link between analysis and synthesis. We are looking for an approach in which the analysis is incorporated and seen to be incorporated in the design procedure.

(2) Students can get caught up in details and spend undue time and attention on calculation procedures. While it is agreed that at some stage students should learn to cope with complex detail, complexity can detract from the pedagogic value of the exercise.

(3) Preparing case studies requires a great deal of work from the teacher and, even if a packaged study from the case study library[3] is used, the details must be mastered by the instructor for it to be effectively used.

(4) Case studies place a large demand on student time and although they involve important processes for the students to learn, students may be able to master these same processes in a more time efficient manner.

This paper describes several years of innovative change in a course in systems analysis for mineral engineers where the major objective was to teach students to solve practical problems using techniques of operations research. We will describe and discuss what we learned from teaching the course in different ways. In particular we will show how the microcomputer was used in meeting some of the objectives and in solving some of the difficulties that arose.

Background

The systems analysis course for mineral engineers involves the application of four operations research techniques: decision theory, linear programming, network analysis and simulation. It has undergone many changes since its inception. Initially, systems analysis was taught in a traditional manner with lectures to introduce new concepts followed by individual homework problems and then more lectures and more individual homework problems. We were, however, dissatisfied with the outcome of this instructional method; students were not learning quite what we thought we taught them, nor could they apply what they learned to practical problems. An interesting discussion of this problem of students' misconceptions appears in some recent literature[4,5,6].

First Iteration

The objective of our first set of changes was to expose students to complex, realistic problems and, at the same time, to the use of computer packages. Each operations research technique was first introduced in a series of formal lectures followed by a simple homework assignment which the student was expected to solve manually. The purpose of this assignment was to reinforce understanding of the technique. Next, the

CH1950-5/83/0000-0042 $00.75 © 1983 IEEE

student, working alone, was required to solve that same assignment using a specially prepared (by the faculty) computer package on a time-sharing mini-computer. This enabled the student (a) to check his own manual solution of the problem, and (b) to gain confidence in the use of the computer package. Finally, the student was required to use that same computer package, often iteratively, to solve a large and complex problem. Realistic, industry related problems were prepared which required formulation, solution and interpretation skills.

The faculty found the tasks of developing realistic 'mini' cases and marking homework very onerous indeed. The students appreciated the faculty effort and also worked very hard, but complained about the overload. They felt, and faculty agreed, that too much time went into formulating the problems and preparing input data for the computer and too little time was left for manipulating the data, exploring different strategies, interpreting results and writing reports. Students learned a lot, but at a high cost in both their own and faculty time, and faculty were generally disappointed with the standard of the student reports.

Second Iteration

In the second year of innovation, three new changes were introduced:

(1) The class was divided into groups of two to four students who worked on all assignments as a team effort. (The application of teamwork to engineering teaching is discussed in Smith, Johnson & Johnson[7].) Individual accountability was ensured by randomly calling on students to present their groups' solution and by giving individual exams.

(2) Instead of introducing each operation research technique in formal lectures, students were primed to 'discover' the necessity for each technique in structured tutorials. A simple but interesting problem was given in class prior to formal instruction on the technique. Students tried to find a solution to the problem ab initio, discussing it in their groups and prompted by faculty using a Socratic approach. (The application of structured tutorials to engineering teaching is discussed by Wilson[8].) The following problem, entitled 'Food for Thought', is an example of this approach and was used to introduce critical path analysis:

FOOD FOR THOUGHT

Suppose you are planning a dinner for two. Your menu consists of a very special soup and a baked chicken entree. The soup must be boiled for 35 minutes and you must allow 15 minutes to serve and consume it. The chicken dish requires a fair amount of preparation: you have to boil rice (for 30 minutes), lightly fry the chicken (for 15 minutes), and then put the rice and chicken in a baking dish in the oven for 15 minutes. It takes 5 minutes to prepare a sauce in the frying pan and 15 minutes to boil peas. [You only have two pots and one frying pan] You have bought a good white wine, allow 5 minutes to uncork it (very carefully) and let it stand for 30 minutes before serving it. Allow 25 minutes to serve and demolish the entree and wine. Finally, you have a good Danish ice cream that must stand for 5 minutes before serving and requires 10 minutes to serve and eat.

How quickly can you prepare and consume the meal?

Students were generally able to find an answer to problems such as 'Food for Thought' in twenty to thirty minutes, each group developing its own algorithm for keeping track of time and the sequence of activities. They were then asked whether their own algorithms would enable them to analyse much larger problems of the same kind. Invariably their answer was negative and they then watched wide-eyed while the lecturer went ahead to formally develop the standard method of solution.

(3) The 'mini' cases were replaced by simpler problems that were only slightly open-ended. These problems had a practical flavor, but were not as complicated as those used the previous year and not nearly as complicated as a typical case study. By carefully choosing the way in which the problem was kept open-ended, it was possible to have students argue constructively in their groups about problem formulation, objectives and method of solution. Keeping the problems small led to a better understanding of the problem itself and of the limitations, assumptions and implementation of the operations research techniques. The following problem on resource allocation, entitled 'Ore Delivery', is an example of a slightly open-ended problem that was used in the linear programming section of the course.

ORE DELIVERY

A mine has 3 ore dumps and 2 crushing mills. The cost (in $/ton) of transporting ore from each dump to each mill is given in Table 1.

The cost of operating mill 1 is $100 per ton and mill 2 is $120 per ton. The crushed ore is required at 2 destinations with the additional transportation costs shown in Table 2.

Table 1

Dump to Mill
Transportation Costs

Dump	Mill 1	Mill 2	Available (1000 tons)
1	50	80	50
2	55	70	60
3	--	50	30
Capacity (1000 tons)	50	50	

Table 2

Mill to Customer
Transportation Costs

Mill	Destination 1	Destination 2
1	80	70
2	75	85
Amount Req'd (1000 tons)	30	65

(a) Find the optimal solution to minimize the cost of meeting the requirements at the destinations.
(b) Would it pay to increase the capacity of Mill 1 to 60,000 tons?
(c) If Destination 1 doubled its order to 60,000 tons, how would you increase the capacity of 1the mills to supply the additional tonnage?

It should be noted that parts (b) and (c) convert the assignment from a straightforward exercise to an open-ended problem. Students have to think pragmatically about the constraints in part (c).

This version of the course was a lot more fun to teach than the previous one and student response was favorable. There was a marked increase in student interest as a result of introducing techniques via structured tutorials; the students felt as though they had a personal stake in the techniques. The students especially liked the teamwork, and since cooperation is an important part of an engineer's professional activity is was interesting to see it work in an instructional setting. The students' discussion and level of reasoning on the assignments showed us that they were getting all the benefits of the open ended problems. It was stimulating to see students uncovering aspects of problems and algorithms that students in previous classes had overlooked. As testimony to the effectiveness of teamwork and the structured tutorial approach, we were able to teach considerably more than in the previous year without any students complaining about overload.

Current developments

The most recent version of this course involved the conversion of all the computer packages to run on microcomputers. Most computer oriented courses are open to two complaints: (1) There is not enough access to the computer(s), and (2) Many students are reluctant to use computers. By using microcomputer workstations, we have provided increased access to computers. The computers themselves are easier to use than their larger counterparts, and the students feel less threatened by these small, independent machines. The applications packages were also rewritten to take advantage of the cheap color graphics available on most microcomputers. There is no doubt that color graphics can be used as an effective aid in the presentation of solutions, verification of input data, and in the actual solution procedure.

Perhaps the most surprising result, however, has been the very marked improvement in the quality of student reports. The use of the full screen editing facilities has led to more careful report planning, a clearer line of logic, and improved grammar and presentation.

Finally, we observed that the students had a feeling that they were keeping abreast of modern technology by using these workstations. This fact alone seemed to motivate them to spend extra time on their assignments.

Conclusions

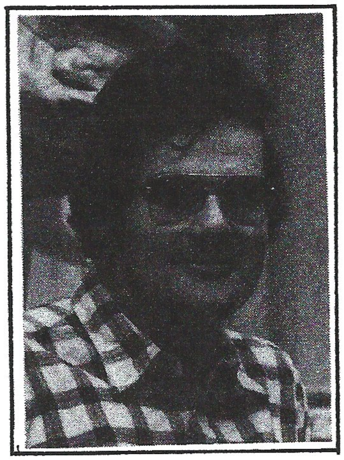
Some of the more important lessons (as realized by both instructors and students) learned during the process of reorganizing the course were:

(1) Structured tutorials and small group discussions were an extremely effective means of introducing new concepts.
(2) Student interest was stimulated by starting each topic with a simple and challenging problem.
(3) Lectures were efficient and effective for transmitting the formal details of each operations research technique.
(4) The computer lab was very popular and was tremendously helpful to students in solving the major problems; students responded imaginatively to user friendly programs and computer graphics.
(5) Teamwork (operating under the cooperative goal structure where interdependence and individual accountability are both stressed) was essential for providing a means whereby the students could learn from each other and manage the major problems without being overwhelmed.
(6) Problems need not be too complex to have practical significance.

While it is the combination of structured tutorials, focused lectures, microcomputer workstations, teamwork and open-ended problems that led to the success of this course, the last conclusion is probably the most important in addressing the question of case studies. As an alternative to case studies, problems of intermediate length and complexity can be used to help students develop engineering problem-solving skills. Even slightly open-ended problems are challenging to the students and illustrate important points concerning the solution of real, practical problems. Added complexity can expand the time commitment of both students and faculty and can lead to students being excessively absorbed in details without furthering their problem-solving skills.

References

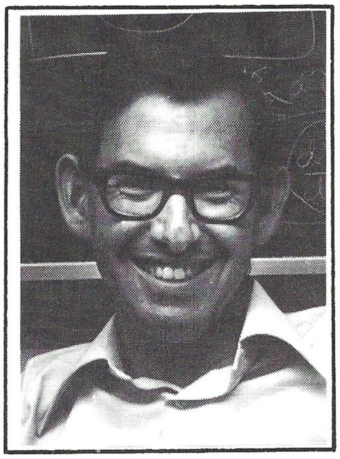
1. Resnick, L.B. Mathematics and science learning: A new conception. Science, 1983, 220, 477-478.
2. Henderson, J.M., Bellman, L.E. and Furman, B.J. A case for teaching engineering with cases. Engineering Education, 1983, , 288-292.
3. Fuchs, H.O. (Committee Chairman). Engineering Cases. Washington: American Society of Engineering Education, 1976.
4. Clement, J. Students' perconceptions in introductory mechanics. American Journal of Physics, 1982, 50(1), 66-71.
5. Lockhead, J. On learning to balance perceptions with conceptions: A dialog between to science students. In Lockhead, J. and Clement, J. (Eds.), Cognitive Process Instruction, Philadelphia, PA: Franklin Institute Press, 1979.
6. Lockhead, J. Beyond Emile: Misconceptions for education in the twenty- first century. Paper presented at American Educational Research Association annual meeting in Montreal, Quebec, April 11-15, 1983.
7. Smith, K.A., Johnson, D.W. and Johnson, R.T. Structuring learning goals to meet the goals of engineering education. Engineering Education, 1981, 72(3), 221-226.
8. Wilson, D.B. A report on the use of structured tutorials as a means to creative thinking. International Journal of Mathematics Education-Science and Technology, 1978, 9(4), 423-431.



Alan Wassyng

Alan Wassyng is an Assistant Professor in the Departmentof Civil and Mineral engineering at the University of Minnesota where he is responsible for running the Department's computing laboratory and for teaching courses in the application of numerical mathematics. He was instrumental in gaining a contract with IBM to supply applications packages to IBM, in return for which IBM has "made available" an IBM 4341 with 10 terminals, 15 IBM personal computers and 3 IBM Instruments Series 9000 computers. Most of his current activities involve utilization of personal computers in engineering education and practice.

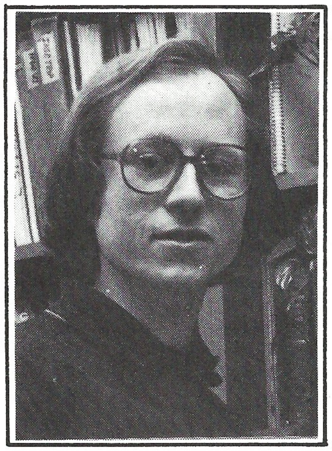
He taught numerical analysis and computer applications at the University of the Witwatersrand, Johannesburg, where he obtained B.Sc, B.Sc(Hons), M.Sc and Ph.D degrees.



Anthony M. Starfield

Professor Starfield received his Bachelors degree in Applied Mathematics and his Ph.D. degree in Mining Engineering from the University of Witwatersrand. He was Professor and IBM head of the Department of Applied Mathematics at the University of Witwatersrand and is currently a Professor in the Department of Civil and Mineral Engineering at the University of Minnesota.

His research interests are in numerical methods and computational approaches for a variety of problems, the central theme being an orientation towards decision making or design. He has supervised 13 graduate theses and has authored over 30 papers on underground heat transfer, computer methods in rock mechanics and ecological modelling. He is the co-author of two books on applied mathematics.



Karl A. Smith

Karl A. Smith is an Assistant Professor of Mineral Engineering at the University of Minnesota. He teaches undergraduate courses in mineral processing, extractive metallurgy and mine systems analysis. His research is in the areas of mineral processing, extractive metallurgy and engineering education. He has extensive experience using cooperative learning procedures and controversy in the classroom.

Professor Smith is a member of AIME, ASEE, AERA, AAAS, Phi Kappa Phi, and Alpha Sigma Mu. His education consists of a B.S. and M.S. degree in Metallurgical Engineering from Michigan Technological University and a Ph.D. degree in Educational Psychology from the University of Minnesota.