1976

Experiments with Computer Aided Self-Paced Instruction for Mathematical Programming Education

Thomas Ho

A. Ravindran

Arthur Sinensky

Report Number:

76-198

EXPERIMENTS WITH COMPUTER AIDED SELF-PACED INSTRUCTION FOR MATHEMATICAL PROGRAMMING EDUCATION

Thomas Ho*, A. Ravindran**, and Arthur Sinensky***

CSD-TR 198
May 1976

* Thomas Ho, Department of Computer Science, Purdue University, West Lafayette, Indiana 47907.

** A. Ravindran, School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907.

*** Arthur Sinensky, 345 - 8th Avenue, New York, New York 10001.

"Experiments with Computer Aided Self-Paced Instruction for Mathematical Programming Education"

by

A. Ravindran

School of Industrial Engineering

Arthur Sinensky

Department of Computer Science

and

Thomas Ho

Department of Computer Science

Purdue University

West Lafayette, Indiana 47907

May 1976

ABSTRACT

In this paper the authors discuss their experiences in converting a lecture-oriented mathematical programming course to the Self-Paced Instructional (PSI) format. This is an elective course for students from all the departments (Engineering, Science, and Management). We will discuss the different strategies including the development of conversational computer programs, which were employed to implement the PSI concept in this course so that its educational objectives are fully met. We will also discuss our experiences with the PSI system, its pros and cons, and the students' response to self-paced learning.

Introduction

The personalized self-paced method of instruction (PSI) has been utilized in engineering education for a number of years now. However the PSI system has rarely been attempted in higher level courses beyond the basic mechanics taught at the freshman level. This paper describes the successful implementation of the self-paced format at Purdue's School of Industrial Engineering for the past two years. The self-paced approach was applied in the operations research area to teach mathematical programming. To the best of our knowledge, this is one of the first successful attempts to use PSI in the mathematical programming education. In this article we will describe the different strategies employed to implement the PSI format to the mathematical programming course, the development of conversational computer programs to teach the mathematical programming algorithms, and the students' reactions to the PSI system.

Course Background

For several years, we at Purdue have been teaching a course on linear programming for undergraduate and graduate students. This is an elective course for all industrial engineering students. In addition to students from other branches of engineering, the course attracts students from industrial management, agricultural economics, mathematics, computer science and statistics. Because of this, the composition of the class and the interests of the students vary widely. The course is offered three times a year and attracts 100 to 150 students.

The basic objective of this course is to introduce the students to mathematical modeling (in particular linear programming models) for solving real-world problems. The major portion of the course deals with solution techniques for these models. From past experience in teaching this course, it has been observed that the non-engineering students prefer to see more emphasis on linear programming theory and computational methods. In contrast, the engineering and business students want more emphasis on case studies describing numerous applications of linear programming. They even express varied interests in the linear programming topics to be discussed in class. For instance, the mathematics students want the inclusion of advanced topics like game theory, and decomposition methods. The computer science students like to see the computer implementation of linear programming algorithms. The civil engineers show more interest in topics like network analysis, transportation problems, and critical path methods. This poses enormous problems in structuring and teaching this course. In previous years, a middle-of-the-road approach was taken, so that it would hopefully satisfy most of the students' interests.

Use of a Personalized Self-Paced Format

It was felt that the needs of the students could be best served if the course was restructured under a self-paced format. Because of the diversified interests found among the students, a personalized self-paced instructional system was developed for this course.

In the PSI system, the subject matter for this course is divided into some basic units, and some optional units. Class lectures are given for the basic units only. The students elect the optional units they want to learn according to their interests, and prepare for them on their own time (off class hours). For this sake, one-third of the scheduled class hours are cancelled to provide time for the self-study.

The basic units are so chosen that they provide the students a minimal amount of knowledge on the fundamentals, which is expected of anyone who has had a course in linear programming. The optional units are selected to provide diversification, and meet the varied and special interests of the students. For instance, during the summers of '74 and '75 when the PSI system was tried for this course, the following basic units and optional units were offered to the students:

I. Basic Units

1. Formulation of linear programs
2. The simplex method
3. Duality theory and the dual simplex method
4. Sensitivity analysis

II. Optional Units

1. Computer implementation of the simplex method
2. Transportation and assignment problems
3. Project networks and PERT/CPM
4. Game theory
5. Individual project
6. Parametric programming

7. Linear programs with lower and upper bounds
8. Decomposition theory
9. Integer programming - Branch & Bound Algorithm

For each of the units (basic and optional), the students are provided with study guides. These contain the objectives of the unit, topical outline, reading materials, and sample problems. Examinations on various units are given throughout the semester. Though excellence is required to pass a unit, the students can repeat the unit examinations any number of times without penalty. For a minimal passing grade in the course, the students have to complete all the basic units. Assignment of higher grades is a function of the number of optional units they pass.

Computer Aids in the PSI System

From the initial experience in teaching this course in PSI format, it was found that considerable time was spent with the students by both the professor and the teaching assistants in helping them learn the optional units for which there are no formal lectures. This increased the manpower needs of this course. It was felt that this could be reduced to a great extent by using the computer as a teaching aid for self-paced learning. The computer-aided teaching can help them learn the various linear programming techniques and their applications at their own time (on/off class hours), and at their own pace.

With the help of a teaching grant from the Purdue Parents' Association, interactive computer programs have been developed in such a way that they will illustrate and test the students' understanding of the subject matter. The students solve some problems on various linear programming techniques in a conversational mode by answering a series of questions. The programs are designed so that the computer does all the time consuming arithmetic calculations, while the "thinking" is done by the students. The student essentially directs the computer to calculate whatever quantities are needed for solving the problem. Based on these, he/she instructs the computer to proceed step by step in the required manner to arrive at the optimal solution to the problem. We have provided in the program a verification routine so that all the steps and instructions given by the students are checked by the computer for correctness.

The arithmetic calculations in linear programming techniques generally consume more than 90% of the total time in solving a problem. Since this time is eliminated by the computer, the students are able to use their study time more effectively. They can solve a variety of problems to test their understanding. A series of illustrative problems emphasizing various concepts and practical difficulties encountered in solving linear programs are also available to the students. Thus a student gets a better and a more complete exposure to the subject matter itself.

Description of the Interactive Computer Program

Conversational Features

The program is completely conversational. All input and output is handled via remote terminals. Thus, the user constantly interacts with the system. The system always directs the user to the next step of the simplex algorithm by asking a question related to some aspect of the algorithm. At all times, the user's response is checked for correctness. If the user should respond incorrectly to one of the system's queries, an immediate feedback will be sent to the user with an explanation of why the response is incorrect and how to go about finding the correct answer.

Error Detection for Inappropriate Student Responses

Each student response to a system question is checked for correctness. If the student's answer is correct, the next question is asked by the system. However, if the response is not correct, feedback explaining the error is given at the terminal. In some cases, the same question is re-asked and the student is given further opportunity to respond correctly. In other instances, the system supplies the correct answer and then proceeds to the next question.

Error checking is accomplished by having the system solve the given linear program step-by-step as the user solves the same problem. For instance, before asking the user to supply the index of the pivot column, the system will have determined that information. Then the user will be asked to supply the same. After the user enters the response, it is a simple matter to check the response against the result already determined by the system. If the user's response is correct, the next question is printed on the terminal. If the response is incorrect, a message to that effect is printed with a brief explanation of why the response is not correct.

For example, on line (A) of the enclosed example, the student is asked, "IS THE PROBLEM IN STANDARD FORM?". In this case, the problem is not in standard form. But the student has responded, "YES". Thus, the feedback given is "INCORRECT RESPONSE - THE PROBLEM IS NOT IN STANDARD FORM, NOT ALL OF THE CONSTRAINTS ARE EQUALITIES". Then the next question is generated.

User Assistance

In addition to the brief explanations provided when the user gives an incorrect response, another form of explanatory assistance is offered by the system. At several points, the student is asked to name the next step of the algorithm. The possible responses and their meanings are:

RELPROFIT - print the relative profit vector

RATIOS - calculate ratios of right-hand sides divided by the corresponding entry of the pivot column

PIVOT - perform a pivot operation

In any instance where one of the above responses is the correct one, the user may type "HELP" if he is uncertain of which choice is the correct one. After "HELP" is typed, a message indicating which is the correct choice and why it is correct will be provided. A possible extension would be to allow the user to ask for "HELP" at any point during the execution of the program. This should be included in the next version of the system.

An example of the use of this command is given in line (B) of the accompanying example. The question asked is, "WHAT IS THE NEXT STEP?". In this case, the correct response would be "RELPROFIT", since the user must look at the relative profit vector. The user has typed "HELP" and thus receives an appropriate explanation.

Implementation Description

Programming Language Requirements

The system was programmed in FORTRAN-IV on the CDC 6500. A language with better character string handling facilities would have been a better choice. However, the constraining factor was the interactive field length restriction on the Purdue system. Interactive jobs are limited to a maximum field length of 21K. The object code generated by FORTRAN was able to fit into this space and still allow some room for extensions. On an IBM system with time-sharing, APL would be a better choice.

Algorithm Description

At present three mathematical programming algorithms have been programmed in the interactive system. These include the primal simplex and the dual simplex algorithms to solve linear programs, and the branch and bound algorithm for solving integer programming problems. The program logic is based on the description of the algorithms given in the text by Phillips, Ravindran, and Solberg [1]. Students are asked to input a problem of their own choosing. This allows for great flexibility in the instructional problems chosen rather than limiting the selection to just a few. The instructor may also suggest some problems which demonstrate the basics of the algorithm. After mastering the basics, the students can experiment with problems of their own choosing.

Concluding Remarks

The PSI system is currently being well received by the students. The authors have evaluated the courses by several measures: comparison with test results in other years, student election of more advanced courses in these areas, student acceptance as indicated on written course critiques, etc. In all cases, the self-paced method of instruction was judged in general to be superior for this particular course. This is not to say that the usual problems of some students falling behind schedule did not exist, because they did. Effort is expended during the semester to help student motivation. Also, the PSI system consumes more faculty time initially in the planning of the units, and preparation of study guides and tests.

Considerable time is still being spent on an individual basis with some students in helping them learn the optional units for which there are no formal lectures. To offset this the author has obtained another research grant to support the preparation of video cassette tapes for the optional units. The School of Industrial Engineering has recently acquired a T.V. monitor and a cassette playback unit. These video tapes can be played on these units by the students at their own time. We hope to have these tapes prepared by the end of the year for students' use.

Acknowledgements

The support for developing the interactive computer programs came from an innovative teaching grant awarded to Professor Ravindran from the Purdue Parents' Association. Preparation of the video cassette tapes for T.V. viewing will be supported by a faculty grant awarded to Professor Ravindran for Instructional Development and Innovation.

Reference:

[1] Phillips, D. T., A. Ravindran, and J. Solberg, Operations Research: Principles and Practice, John Wiley & Sons, New York, 1976.

APPENDIX

The solution to the following linear program by the interactive simplex algorithm is shown as an illustration:

$$\begin{aligned} \text{Maximize} \quad & z = 10x_1 + 6x_2 + 4x_3 \\ \text{Subject to} \quad & x_1 + x_2 + x_3 \le 100 \\ & 10x_1 + 4x_2 + 5x_3 \le 600 \\ & 2x_1 + 2x_2 + 6x_3 \le 300 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

```
LINEAR PROGRAMMING SELF-PACED INSTRUCTION


ENTER THE NUMBER OF CONSTRAINTS FOLLOWED BY
A BLANK FOLLOWED BY THE NUMBER OF VARIABLES

: 3 3

ENTER THE LINEAR PROGRAM

IS THE OBJECTIVE FUNCTION TO BE MINIMIZED OR MAXIMIZED?

: MAX

ENTER THE C VECTOR

: 10 6 4

ENTER THE A MATRIX, ROWWISE

: 1 1 1
: 10 4 5
: 2 2 6

ENTER THE B VECTOR

: 100 600 300

ENTER THE CONSTRAINT TYPES:
   < FOR LESS THAN OR EQUAL TO,
   > FOR GREATER THAN OR EQUAL TO,
   = FOR EQUALITY CONSTRAINTS

: < < <

(A) IS THE PROBLEM IN STANDARD FORM ?

: YES

INCORRECT RESPONSE -
THE PROBLEM IS NOT IN STANDARD FORM


NOT ALL OF THE CONSTRAINTS ARE EQUALITIES


CHECK THE CONSTRAINTS TO SEE IF THERE ARE ANY
NEGATIVE RIGHT-HAND SIDES.  ALL SUCH CONSTRAINTS
NEED TO BE MULTIPLIED BY -1.
WHICH CONSTRAINTS NEED TO BE MULTIPLIED BY -1 ?
TYPE 0 (ZERO) IF NONE

: 0

FOR EACH CONSTRAINT, TYPE A +1, -1, OR 0
DEPENDING ON WHETHER A SLACK VARIABLE IS
NEEDED, A SURPLUS VARIABLE IS NEEDED, OR
NEITHER ONE IS NEEDED
```

```
: 1 1 1

THE SLACK AND SURPLUS VARIABLES HAVE BEEN INSERTED

ARE ARTIFICIAL VARIABLES NEEDED?

: YES

INCORRECT RESPONSE -
NO ARTIFICIAL VARIABLES ARE NEEDED BECAUSE
THERE EXISTS AN INITIAL FEASIBLE BASIS

THE CURRENT TABLEAU IS:

COST     10       6       4       0       0       0
         X1       X2      X3      X4      X5      X6      B
BASIS
X4      1.00    1.00    1.00    1.00     .00     .00100.00
X5     10.00    4.00    5.00     .00    1.00     .00600.00
X6      2.00    2.00    6.00     .00     .00    1.00300.00

WHAT ARE THE VALUES OF THE BASIC VARIABLES
IN THE ORDER IN WHICH THEY APPEAR IN THE TABLEAU

: 100 600 300

WHAT IS C(B), THE VECTOR OF COST COEFFICIENTS
OF THE BASIC VARIABLES IN CORRESPONDING ORDER ?

: 0 0 0

THE VALUE OF THE OBJECTIVE FUNCTION =             .00

(B) WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: HELP

THE NEXT STEP OF THE SIMPLEX ALGORITHM IS TO
LOOK AT THE RELATIVE PROFIT VECTOR AND
DETERMINE WHETHER THE CURRENT BASIC FEASIBLE SOLUTION
IS OPTIMAL.  DETERMINE THE  PIVOT COLUMN AND
HENCE THE VARIABLE WHICH ENTERS THE BASIS
BY CHOOSING THE VARIABLE WITH THE LARGEST RELATIVE
PROFIT COEFFICIENT.  IN ORDER TO SEE THE RELATIVE
PROFIT VECTOR, TYPE RELPROFIT.

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: RELPROFIT

THE RELATIVE PROFIT VECTOR IS:
      10.00        6.00        4.00         .00         .00         .00
```

```
IS THE TABLEAU OPTIMAL ?

: YES

INCORRECT RESPONSE -
FOR A MAXIMIZATION PROBLEM, THE TABLEAU IS OPTIMAL
WHEN ALL THE RELATIVE PROFIT COEFFICIENTS ARE
NON-POSITIVE

SELECT THE PIVOT COLUMN

: 3

INCORRECT RESPONSE
FOR A MAXIMIZATION PROBLEM, THE VARIABLE WITH THE
LARGEST RELATIVE PROFIT COEFFICIENT SHOULD ENTER
THE BASIS

SELECT THE PIVOT COLUMN

: 1

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: HELP

THE NEXT STEP OF THE SIMPLEX ALGORITHM
 IS TO DETERMINE THE PIVOT ROW AND
 HENCE THE VARIABLE WHICH LEAVES THE BASIS
 USE THE MINIMUM RATIO RULE TO DETERMINE
 THE BASIC VARIABLE TO LEAVE THE BASIS
 TO CALCULATE RATIOS TYPE RATIOS

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: RATIOS

FOR EACH ROW, TYPE A 1 IF A RATIO NEEDS TO BE CALCULATED
OTHERWISE TYPE A 0

: 0 0 0

THE RESPONSE FOR ROW 1 IS INCORRECT

THE RESPONSE FOR ROW 2 IS INCORRECT

THE RESPONSE FOR ROW 3 IS INCORRECT
```

```
THE CORRECT RESPONSE IS:    1    1    1

THE RATIOS ARE
ROW NO.        RATIO

   1         100.00
   2          50.00
   3         150.00

CAN A PIVOT OPERATION BE PERFORMED ?

: NO

INCORRECT RESPONSE -
A PIVOT OPERATION CAN BE PERFORMED SINCE AT
LEAST ONE RATIO HAS BEEN CALCULATED


SELECT THE PIVOT ROW

: 3

INCORRECT RESPONSE -
THE ROW WITH THE SMALLEST RATIO IS THE PIVOT ROW


SELECT THE PIVOT ROW

: 2

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: HELP


NOW THAT THE PIVOT ELEMENT HAS BEEN
DETERMINED, A PIVOT OPERATION CAN BE
PERFORMED.  TYPE PIVOT TO PERFORM THE
PIVOT OPERATION


WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: PIVOT

THE CURRENT TABLEAU IS:
```

COST	10	5	4	0	0	0	
	X1	X2	X3	X4	X5	X6	B
BASIS							
X4	.00	.60	.50	1.00	-.10	.00	40.00
X1	1.00	.40	.50	.00	.10	.00	60.00
X6	.00	1.20	5.00	.00	-.20	1.00	190.00

```
WHAT ARE THE VALUES OF THE BASIC VARIABLES
IN THE ORDER IN WHICH THEY APPEAR IN THE TABLEAU

: 40 60 150

WHAT IS C(B), THE VECTOR OF COST COEFFICIENTS
OF THE BASIC VARIABLES IN CORRESPONDING ORDER ?

: 6 10 0

THE VALUE OF THE OBJECTIVE FUNCTION =          600.00

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: RELPROFIT

THE RELATIVE PROFIT VECTOR IS:
       .00        2.00       -1.00         .00       -1.00         .00

IS THE TABLEAU OPTIMAL ?

: NO

SELECT THE PIVOT COLUMN

: 2

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP

: RATIOS

FOR EACH ROW, TYPE A 1 IF A RATIO NEEDS TO BE CALCULATED
OTHERWISE TYPE A 0

: 1 1 1

THE RATIOS ARE
ROW NO.      RATIO

  1          66.67
  2         150.00
  3         150.00

CAN A PIVOT OPERATION BE PERFORMED ?

: YES

SELECT THE PIVOT ROW

: 1

WHAT IS THE NEXT STEP ?
CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
IF YOU DONT KNOW, TYPE HELP
```

```
: PIVOT

 THE CURRENT TABLEAU IS:

COST     10       6        4        0                  0
         X1       X2       X3       X4       X5       X6      B
BASIS
X2       .00     1.00      .33     1.67     -.17      .00 66.67
X1      1.00      .00      .17     -.67      .17      .00 33.33
X6       .00      .00     4.00    -2.00     -.00     1.00100.00

 WHAT ARE THE VALUES OF THE BASIC VARIABLES
 IN THE ORDER IN WHICH THEY APPEAR IN THE TABLEAU

: 66.67 33.33 100.00

 WHAT IS C(B), THE VECTOR OF COST COEFFICIENTS
 OF THE BASIC VARIABLES IN CORRESPONDING ORDER ?

: 6 10 0

 THE VALUE OF THE OBJECTIVE FUNCTION =          733.33

 WHAT IS THE NEXT STEP ?
 CHOOSE FROM PIVOT, RATIOS, OR RELPROFIT
 IF YOU DONT KNOW, TYPE HELP

: RELPROFIT

 THE RELATIVE PROFIT VECTOR IS:
          .00           .00         -2.67         -3.33          -.67          .00

 IS THE TABLEAU OPTIMAL ?

: YES


 DO YOU WANT TO BEGIN A NEW PROBLEM?

: NO
+++LOG

TCB L024 11.44.51. 10/07/75.
ESTIMATED SESSION COST $    .17
PLEASE TURN OFF TERMINAL. THX.
```