Operations Research and the Information Systems Curriculum

Greta Pangborn

gpangborn@smcvt.edu

Computer Science and Information Systems, Saint Michael's College
Colchester, VT 05439

Abstract

The current boom in information technology and the corresponding wealth of data available to management would seem to make quantitative data analysis techniques more essential than ever. This paper includes discussion of how a number of operations research techniques including mathematical programming, simulation, yield management, dynamic programming, network algorithms, and statistics might be incorporated into the information systems curriculum.

Keywords: operations research, mathematical programming, simulation, yield management, dynamic programming, statistics, information systems curriculum

1. INTRODUCTION

The executive summary of the IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Gorgone 2002) lists strong analytical and critical thinking skills as one of four key characteristics important for an IS professional. The authors propose to develop these characteristics through a curriculum that contains embedded problem solving as well as mathematics and statistics prerequisites. Operations Research techniques, namely analytical methods to inform decision making would seem to match the quantitative needs of IS students.

If students are going to feel comfortable applying mathematical skills after graduation they must gain experience using these skills in the context of computational information systems activities. Students must learn not only how to organize and efficiently access data but also how to analyze data. This can be accomplished through assignments related to topics such as mathematical programming, simulation, and statistics that emphasize problem formulation, modeling, and the use of heuristics.

2. MATHEMATICAL PROGRAMMING

Linear programming has its roots in military planning problems for World War II and involves the maximization or minimization of a linear objective function subject to a set of linear inequalities. Integer programming adds additional restrictions that some or all of the variables must take on only integer values, while nonlinear programming relaxes the linearity restrictions. (However, it should be noted that nonlinear programming problems are too challenging for algorithms to effectively handle the general case.) Current versions of Microsoft's Excel spreadsheet program include a Solver add-in with a simplex algorithm implementation for linear programs, branch-and-bound for integer programs, and a generalized reduced gradient algorithm for nonlinear problems (Microsoft Support Pages, 2004). This tool is particularly appealing since most students will have access to Excel in their future positions.

A reasonable introductory IS course assignment is to have students design a spreadsheet that will support efficient trial and error attempts for finding a good solution to a small integer programming

problem. The transportation problem attempts to minimize the cost of shipping (identical) goods from a set of n supply centers (each with a specified inventory) to a set of m demand centers (each with a specified demand level). The unit shipping cost from each supply center to each demand center is known, the total supply is greater than or equal to the total demand, and the goal is to determine the minimum cost shipping plan that fills the demands while respecting the supply constraints. The students should create spreadsheets that have all of the given information in a separate parameter area, a clearly marked area for the user to enter shipping numbers in, and formulas that automatically update the total cost and give the user an indication of how well demand is being met and what suppliers have left in stock.

Students should also be asked to articulate how they arrived at their final solution. What might constitute a good initial solution? What improvement strategy did they employ? This exercise leads naturally into a demonstration of the Solver software, since the students have completed 95% of the required work in their creation of a flexible spreadsheet design. The appendix shows how Solver may be applied to such a spreadsheet. This problem can also lead into an exploration of sensitivity analysis; i.e., how does the solution change when one of the objective function coefficients is modified or when one of the resource limits is changed.

A more complex scheduling formulation (but one that is still reasonably small) can be presented to show the difficulty of some of these problems and the limitations of the Excel Solver software. Such an example would be a machine scheduling problem with approximately 3 identical machines and 10 jobs that each require some number of contiguous minutes on any one of the machines. The corresponding objective function should maximize the minimum amount of reserve time available on the machines. Such a problem indicates the difficulty of these problems to students since it is unrealistically small but is still not solved instantly by the Excel Solver.

3. DYNAMIC PROGRAMMING

The basic idea of dynamic programming is to transform a complicated problem into a series of simple problems using a set of recursive equations. One particularly famous example is the knapsack problem where there is a bag with weight limit B and a set of n potential items to take on the trip (each with an associated weight and utility). The goal is to maximize the utility of the selected items while not exceeding the bag's weight limit. This problem is trivial (for bag sizes $0...B$) if there is only one item to consider; you take the item if it will fit in the bag and omit it if it does not. If you want to consider a second item you then need to decide if you are better off taking the second item along with the best solution for the first item at the reduced bag size or omitting the second item and leaving all of the space for the first item. Once you have the best solution for bag sizes $(0..B)$ for two items, you would take the third item into consideration in a similar manner, and so on. Dynamic programming gives yet another way to illustrate the power of recursion in an introductory programming class, though students are likely to find the knapsack problem more realistic if it is rephrased as an investment question.

There are also a number of dynamic programming algorithms used in bioinformatics that can interest students because of their obvious application and the clear challenges related to the size of the DNA and protein sequence datasets they must be applied to. The maximum subsequence search starts with a sequence and a score for each nucleotide or amino acid that appears in the sequence. The goal is to determine a maximum scoring subsequence of the original sequence, where the subsequence score is the sum of the scores of the nucleotides or amino acids that appear in that subsequence. Students are quick to see a brute-force approach looking at each possible subsequence that runs in either $O(n^3)$ or $O(n^2)$ time depending on the implementation. A simple dynamic program can be used to reduce this to $O(n)$ time. This dynamic program relies on the fact that the best subsequence ending at a spot j will consist of either just that element or an extension of the best sequence ending at spot (j-1).

4. SIMULATION

A computer simulation is an abstraction or imitation of a real system that is used as an inexpensive way to model the outcomes of potential decisions when the complexity of the system makes analytical analysis impractical. Simulation models generally fall into two categories: dynamic system simulations that capture the sequence of interactions among parts of a system over time and Monte Carlo simulations which are static and repeatedly sample from a given input distribution in order to create a characterization of the related output distribution.

The newsboy problem is a classic probabilistic inventory model that could be easily explored in an introductory programming course Monte Carlo simulation assignment. In the newsboy problem the goal is to determine the appropriate number of papers to buy from the distributor in the morning where the known information includes the cost of the papers, the price each paper may be sold for, the penalty for having too few papers to meet the demand, and the probability distribution of the demand. Depending on whether the members of the class have already taken a statistics course, the demand distribution could be either a simple discrete distribution given in tabular form or a more complicated distribution. (Similarly the expectations regarding output analysis could vary accordingly.) Such an application could also be appealing in a course that employed a language such as Visual Basic, where part of the assignment would focus on the design of a GUI for a decision support tool. New Excel add-ins have also made it possible to work with Monte Carlo simulations using spreadsheets as the computational engine (Evans 2000).

An extremely simple system simulation that demonstrates the impact of randomness in a system is a manufacturing line involving *n* sequential machines with identical processing time distributions and buffer space in front of each machine to store work in progress (WIP). For example, during one time unit the number of units processed by a given machine could be a random number between 1 and 10. The students should examine the buildup of WIP in the buffers as well as the amount of time machines are left idle. This analysis may again be completed in either a standard programming assignment (where it may easily be extended to a more complicated system) or in a spreadsheet assignment (Johnson 2002).

5. YIELD MANAGEMENT

Yield Management refers to a pricing model in which a company charges buyers different prices in accordance with the value they place on the good in question. A classic example of yield management is the airline industry's use of a two week window to differentiate between early booking leisure travelers who are generally thrifty and late booking business travelers who are willing to pay a premium. Yield management has had a significant impact on the bottom line for the airline industry and was described as the "single most important technical development in transportation management since the era of airline deregulation," by former American Airlines CEO Robert Crandall (Smith 1992). Characteristics that indicate a situation where yield management is likely to be useful include perishable goods or services which cannot be stored, uncertain demand, and the ability to differentiate among customer segments. Netessine and Shumsky provide a set of potential examples for students to work with (Netessine 2002). Students could also consider applying this idea to a new arena such as the movie theatre industry where online ticket purchases increase the potential for categorizing customers (Oberwetter 2001).

6. NETWORK PROBLEMS

A wide range of problems can be modeled using graphs consisting of a set of nodes and a set of edges that each connects a pair of nodes. A few examples include road planning (where nodes correspond to cities and edges to potential highways), professional relationships (where nodes correspond to people and edges correspond to working relationships), and mechanical systems (where nodes correspond to joints and edges to rods or springs). In many of these problems there is also cost information associated with each edge (or node). Graph problems can be an excellent tool for

demonstrating the advantages and disadvantages of different data structures (adjacency matrices versus linked lists) and run-time analysis.

One classic network example is the Minimum Spanning Tree (MST) problem which seeks to find a minimum cost subset of the edges that connects all of the nodes in the graph. This problem can be solved by a pair of greedy algorithms (Prim's and Kruskal's) which are reasonable to code in an introductory class and can be used to highlight some of the issues mentioned above. A typical example used to motivate the MST problem is the question of finding the least expensive way to connect a network of machines. A slightly more sophisticated case to consider would be the storage of DNA data where the nodes correspond to DNA sequence data and the cost of an edge corresponds to the number of differences between the sequences represented by its endpoints.

The Critical Path Method (CPM) is a project management technique that considers a graph with nodes corresponding to each of the tasks, directed edges between tasks where one task must be completed before the other can be started, and edge lengths corresponding to task completion times. The longest path from the starting task to the finishing task (dummy tasks each requiring 0 time) is the minimum amount of time that it will take to complete the project. This problem can be solved via a simple dynamic programming recursion in a programming course or using a spreadsheet (Seal 2001).

7. PROBABILITY AND STATISTICS

It is expected that most IS students will take a semester long course in statistics and probability, however, it is important that these ideas are reinforced in other courses. One way to do this is to integrate statistical analysis into other activities such as the simulation exercises above. There are also many examples from statistics and probability classes that may be adapted for introductory programming courses. One such example is the game of HOG, a dice game that was developed for the purposes of teaching probability and statistics (Feldman 2003). The basic rules of the

game are as follows: the players take turns rolling the dice, in a given round the players may roll as many dice as they would like, if a player rolls a 1 (on any of the dice) they receive a score of 0 for that round, otherwise their score is the sum of all of the dice, the first player to reach a score of 100 wins. The students need to determine what strategy they want to employ and to defend their strategy empirically. (Extra credit could be given to students who also justify their results analytically.)

8. FINAL COMMENTS

INFORMS, the Institute for Operations Research and the Management Sciences is currently in the midst of a campaign promoting the tagline "The Science of Better" (Horner 2003). They point out the increased importance of analytical decision making in the context of the current boom in information technology. Our students must be able to effectively use the large amounts of data and computing power at their disposal to make more informed decisions.

The quantitative decision support tools discussed in this paper may be integrated into any number of courses currently included in the Information Systems curriculum. These topics could also be combined into a quantitative decision analysis elective that would complement the curriculum.

9. ACKNOWLEDGEMENTS

10. REFERENCES

Evans, J.R, 2000, Spreadsheets as a Tool for Teaching Simulation, *INFORMS Transactions on Education*, Vol. 1, No. 1, http://ite.informs.org/Vol1No1/Evans/.

Feldman, L. and Morgan, F., 2003, The Pedagogy and Probability of the Dice Game HOG, Journal of Statistics Education Volume 11, Number 2, www.amstat.org/publications/jse/v11n2/feldman.html.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D. & Longenecker, H., 2002,

IS 2002: Model curriculum and guidelines for undergraduate programs in information systems. http://www.acm.org/education/is2002.pdf.

Horner, P., 2003, The Science of Better, *OR/MS Today*, Vol. 30, No. 6.

Johnson, A. and Drougas, 2002, A., Using Goldblatt's Game to Introduce Simulation in the Introductory Operations Management Course, *INFORMS Transactions on Education*, Vol. 3, No. 1.

Microsoft Support Pages, 2004, Solver Uses Generalized Reduced Gradient Algorithm,http://support.microsoft.com/default.aspx?scid=kb;en-us;82890.

Netessine, S. and Shumsky, R., 2002, Introduction to the Theory and Practice of Yield Management, *INFORMS Transactions on Education*, Vol. 3, No. 1, http://ite.informs.org/Vol3No1/NetessineShumsky/.

Oberwetter, R., 2001, Building Blockbuster Businesses, *OR/MS Today*, Vol. 28, No.3.

Seal, K.C., 2001, A Generalized PERT/CPM Implementation in A Spreadsheet , *INFORMS Transactions on Education*, Vol. 2, No. 1, http://ite.informs.org/Vol2No1/Seal/.

Smith, B.C., Leimkuler, J.F., and Darrow, R.M., 1992, Yield Management at American Airlines, *Interfaces*, Vol. 27, No. 1.

APPENDIX

The following screen snapshots indicate how to solve an instance of the transportation problem using the Excel Solver add-in. In this case there are four supply centers with respective supply quantities of 40, 70, 55, and 35 and six demand centers with respective demand quantities of 30, 45, 20, 15, 50, and 40. The "goods shipped" area provides a clearly marked rectangle in which the user can enter shipping solutions. The totals shipped from each supply center and to each demand center are automatically updated. The unit shipping costs are displayed just below, and the cost of the current solution is automatically calculated at the bottom.

This spreadsheet supports reasonably efficient testing of initial solution and improvement heuristics. This spreadsheet structure also makes it very easy to input the necessary information into the Excel Solver. The Solver Parameters window allows the user to enter the target cell (total cost), the objective (maximization), what values may be changed (the shipping quantities), and the constraints. It makes sense to introduce students to integrality constraints, even though this is redundant in the case of the transportation problem. Solver Options that should be discussed include the tolerance setting, non-negativity requirements, and the indication that this is a linear model.

GOODS SHIPPED

	D-1	D-2	D-3	D-4	D-5	D-6	SUM	Supply
S-1	30	0	0	0	10	0	40	40
S-2	0	45	10	15	0	0	70	70
S-3	0	0	10	0	40	5	55	55
S-4	0	0	0	0	0	35	35	35
SUM	30	45	20	15	50	40		
Demand	30	45	20	15	50	40		

UNIT SHIPPING COSTS

	D-1	D-2	D-3	D-4	D-5	D-6
S-1	75	65	175	90	110	150
S-2	90	30	45	50	105	130
S-3	40	55	35	80	70	75
S-4	95	150	100	115	55	55

TOTAL SHIPPING COSTS

	D-1	D-2	D-3	D-4	D-5	D-6	SUM
S-1	2250	0	0	0	1100	0	3350
S-2	0	1350	450	750	0	0	2550
S-3	0	0	350	0	2800	375	3525
S-4	0	0	0	0	0	1925	1925
					TOTAL COST		11350

Solver Parameters ? X

Set Target Cell: $I$23

Equal To: ○ Max   ● Min   ○ Value of: 0

By Changing Cells:

$C$3:$H$6

Subject to the Constraints:

$C$3:$H$6 = integer
$C$7:$H$7 >= $C$8:$H$8
$I$3:$I$6 <= $J$3:$J$6

Solve
Close
Guess
Options
Add
Change
Delete
Reset All
Help

Solver Options ? X

Max Time: 100   seconds

Iterations: 100

Precision: 0.000001

Tolerance: 0   %

Convergence: 0.0001

OK
Cancel
Load Model...
Save Model...
Help

☑ Assume Linear Model   ☐ Use Automatic Scaling
☑ Assume Non-Negative   ☐ Show Iteration Results

Estimates   Derivatives   Search
● Tangent   ● Forward   ● Newton
○ Quadratic   ○ Central   ○ Conjugate