JITTA
JOURNAL OF INFORMATION TECHNOLOGY THEORY AND APPLICATION

GOAL-PROGRAMMING TECHNIQUES TO SUPPORT END-USER DECISION MAKING

JEFF BUTTERFIELD, Kentucky University
Email: Jeff.Butterfield@wku.edu

SCOTT BUTTERFIELD, Georgia State University

JOSEPH GEIGER, University of Idaho

ABSTRACT

End users are frequently challenged with decision making where the goals, objectives, and priorities of management and entire functional areas are in conflict or defy standard quantifiable assessment (i.e., return on investment, payback period, etc.). In addition, objectives, priorities, and resources are constantly changing as corporate politics, staff turnover, or market conditions drive a firm in new directions. End users require, therefore, a straightforward capability of displaying resource or other constraints and the relative *priorities of initiatives and projects in such a way that the manager can strive towards several objectives* simultaneously.

This paper discusses a well-established modeling technique, Goal Programming *and shows how this once involved analysis technique has been simplified with the advent of powerful desktop hardware and software. GP models can now be developed on personal computers and used by managers and senior staff to simulate, in a matter of a few minutes, any scenario, which represents the relative priorities of initiatives and projects within defined resource or other constraints.*

Butterfield, J., S. Butterfield, J. Geiger, "Goal-Programming Techniques to Support End-User Decision Making", *The Journal of Information Technology Theory and Application (JITTA),* 1:1, 1999, 23-32.

INTRODUCTION

A popular view of the modern organization is one of being a consumer, overseer and purveyor of information and its associated products (Feldman and March, 1981). This information is often the basis for the management decisions that further drive the organization. Modern technology has been developed to better enable firms to acquire, store, and process this information as well as to aid management in making decisions (Keen and Morton, 1977). Decision Support Systems (DSS), as they are commonly known, are designed to manage the data and to present it in such a way as to allow managers to exercise their insight and expertise. Various models are used in this process depending on the type of decision being made. DSSs have been widely deployed to support knowledge workers in a variety of areas including finance, logistics and production (Holsapple and Whinston, 1987).

While often the product of traditional systems-design approaches, personal decision-support systems have also been a popular context for end-user computing (EUC). Since computers appeared on the corporate desktops, end-users have been developing their own individual or departmental applications. By 1983, it was reported that growth in end-user development was growing by 50 to 90 percent per year and the trend continues as easier-to-use applications become more common (Mayo, 1986). Some have suggested that this trend is due, in part, to dissatisfaction with centrally-developed applications. It is estimated that less than half of all systems developed by MIS departments provide support for decision making activities (Sumner and Klepper, 1987). Often, decision-making tools are not requested as they often are assigned a low priority by the central MIS infrastructure.

Improvements in hardware and software technology have made this dependence on traditional MIS design less significant. One particular set of tools that has become available to most personal computer users are the "Solver" or linear programming modules that are now included with most popular spreadsheet packages. One possible application of these tools is the formulation of Goal Programming (GP) models to help with complex (i.e., multi-criteria) decisions. Although GP is a well-developed technique that has been used for many years, the recent availability of powerful desktop computer hardware and software has made it potentially useful to a variety of users.

Consider a manager who must simultaneously balance the following multiple (and often conflicting) objectives: new product development, harvesting the existing product line, maintaining stable or growing profits and market share, retaining a conservative financing strategy, and remaining within current operating budgets. These complex problems are common and their solutions are often elusive. This manager must devise a series of decisions and functional-level agreements in such a way as to maximize the chances that the approach will succeed. The task must be done carefully, not only to avoid personal and professional embarrassment but also to avoid having the problem blossom into an unworkable large number of options and solutions.

This paper explores the use of Goal Programming as a tool to aid end users who are faced with complex decisions and may not enjoy much organizational computing support. The obvious advantages to the end user of using Goal Programming are twofold: (1) Goal Programming is well-suited to problems that require balancing trade-offs and costs between competing alternatives (as is common in organizational settings), (2) the decision models may be easily formulated by end-users using common spreadsheet packages. No software coding is required and the decision model may be readily adapted to different decision situations. Once developed, the end user is only required to enter the constraints and relative priorities in a series of tables easily formatted on a spreadsheet. This paper examines complex decisions of the type noted above and provides an example of how Goal Programming can be used as decision support tool.

DECISION MODELING WITH COMPLEX CRITERIA

Organizational decision-making often involves the evaluation of various courses of action where multiple objectives must be met. These objectives are often in conflict and various constraints (e.g., financial) will most often dictate that one objective be preferred to another. Such organizational dilemmas are examples of Multi-Criteria Decision Making (Keeney and Raiffa, 1976).

Keen (1987), in discussing such complex problems suggests that, "multi-criteria decisions pose dilemmas or even crises of judgment: ethical choices, trade-offs between cost and service, conflicts of preferences, and ‘political?problems are obvious examples. The multi-criteria problem is at the core of Decision Support."

Multi-Criteria decision problems may be formulated as:

Maximize: $f(x)$

Subject to: x ? S

Where: $f(x)$ is the set of objectives that must be simultaneously maximized

x is the set of decision variables, and

S is the set of feasible alternatives

Goal Programming translates the multi-criteria problem formulated above into a series of objectives approximated by a table of relative priorities. The relative priorities are displayed as "penalties". In practical application the penalty may be thought of as some cost that must be paid if an objective is not met. The cost may be financial or, more commonly, a less tangible personal or political one (e.g., angering one's supervisor). The numerical values in the table represent the penalty for not achieving the objective. In the example used in this paper, penalties are assigned for not appropriating resources for specific program budgets. The numerical value of a penalty increases as the target budget is missed by + or - 5% or (< 5% > 5%). Any reasonable range (10%, 100%, etc) may be used or the range can be broken into smaller segments. The numerical value of the penalties may also be any range. The example in the paper used 0 ?100 penalty points to create a tableau of relative priorities which effectively differentiates between the various competing objectives.

When provided this information, PC-based solvers, optimizers, etc. can run a goal program solution in a minute or so for most applications. The algorithms weigh each constraint against the relative priorities of the projects and reduce the solution space to a feasible solution. Although not necessarily optimal, the solutions maximize the values associated with the organization’s goals while minimizing penalties or associated costs.

The priorities attached to each objective are considered to be relatively preemptive ?meaning there is a bias towards satisfying higher priority goals but not to the entire exclusion of lower priority goals. For example, if the three highest priority projects add up to more than the total budget constraint, GP will select two and fill in with lower priority projects until exhausting the available budget.

A general formulation of a goal-programming model with (relative) preemptive weights is shown below:

Let:

n = number of objectives considered

x_i = value of the ith decision variable in the problem

d_i^+ = amount by which objective i is exceeded

d_i^- = amount by which objective i is underachieved

P_i = priority factor for the objective having the ith priority

(Note: objective with highest priority has factor P_1)

$P_i >>> P_{i+1}$ such that there is no number $k>0$ such that $nP_{i+1}?P_i$. P_i is infinitely larger than P_{i+1}

The priority factors are then included in the function with the appropriate deviational variables.

z is the objective function

$x_1, x_2, \ldots, x_n$ are the n decision variables

$c_1, c_2, \ldots, c_n$ are the coefficients of the decision variables in the objective function

$ai_1, ai_2, \ldots, ai_n$ are the coefficients of the decision variables in the ith constraint

b_i are the right-hand-side constraints of the ith constraint (i = 1, 2,...,m)

$$\text{Minimize: } z = \Sigma \sum P_k w_i^{+}{}_{,k} d_i^{+} + \Sigma \sum P_s w_i^{-}{}_{,s} d_i^{-}$$

$$n$$

$$\text{Subject To: } \Sigma\, m_{ij} x_j - d_i^{+} + d_i^{-} = g_i \quad i = 1, 2, \ldots, p$$

$$j=1$$

$$n$$

$$\Sigma\, a_{ij} x_j \ ?\, b_i \quad i = p+1, \ldots, p+m$$

$$j=1$$

$$x_j, d_i^{+}, d_i^{-} \ ?\ 0 \quad j = 1, \ldots, n;\ i = 1, \ldots, p$$

GOAL PROGRAMMING PROBLEM FORMULATION

A variety of scholarly texts present the goal-programming methodology in sufficient detail that one could write their own GP program and tailor it to their specific organizational support system (see: Lee, 1972; Ignizio, 1985). However, most modern spreadsheet packages (e.g., Lotus 1-2-3, Excel and Quattro Pro) include suitable optimizing tools. Such programs have the advantage of being readily available, user friendly, and readily usable by those who are already familiar with basic spreadsheet manipulation. Decision problems may be quickly set up with these available tools thereby allowing management to focus on developing alternatives, assigning priorities, and coding the values of penalties. It is not the goal of this report to provide a tutorial on using particular spreadsheets as each of the popular tools implements the tools in a different way. The reader is encouraged to take advantage of the more comprehensive resources that address this topic (e.g., Underdahl, 1994; Person, 1996; Habraken, 1998).

Tools of this sort will generally work best when three objectives are met during the problem-formulation stage:

(1) There should be one broad organizational goal guiding the selection between alternatives (e.g., cost minimization, profit maximization, etc.)

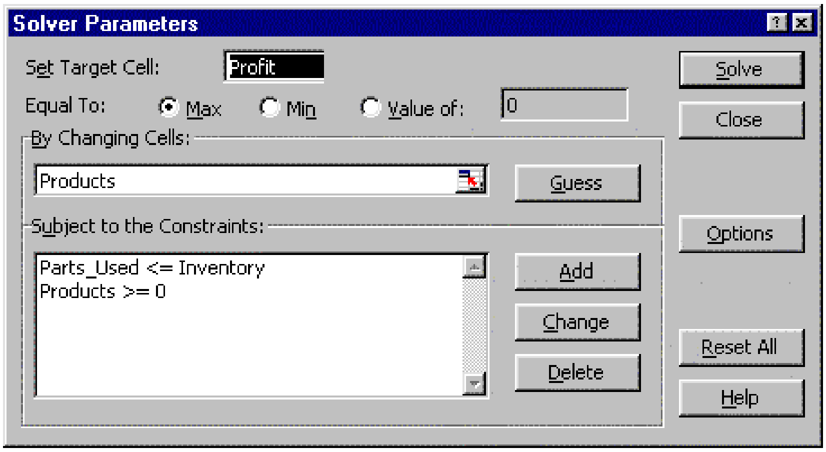
(2) Constraints should be defined as inequalities (e.g., raw materials <= inventory)

(3) The problem should have input values that directly or indirectly affect both the constraints and values being optimized.

The basic objective of most such spreadsheet tools is to find a solution for the user that satisfies the given constraints while maximizing (or minimizing) the problem objective. Most such models will be made up of *parameters* (fixed numbers or values associated with the problem), *decision variables* (variable input values that may be under the control of the decision maker), and *objective functions* (the quantity that the decision maker wants to maximize or minimize).

Additionally, most decision models will make use of *constraints,* which are relationships such as allowing production to proceed assuming that MATERIALS INVENTORY > 0. Constraints are made up of a *reference*, a *relation*, and an *expression*. The reference will typically be a cell reference such as G13 in the spreadsheet (e.g., G13<C17). The relation is any valid logical operation such as <, >, =, <=, =>, or <>. The expression may be one of the following:

- A numeric constant (e.g., 10)
- A cell reference in the spreadsheet (e.g., C8)
- A range of cells (e.g., C8:C10)
- A formula (e.g., E5/F9 + 3)



Set Target Cell: Profit
Equal To: ● Max ○ Min ○ Value of: 0
By Changing Cells: Products
Subject to the Constraints:
Parts_Used <= Inventory
Products >= 0

Figure 1. Example of Parameter Setting

The analysis tool works to identify solutions that satisfy the given constraints. A constraint is considered satisfied if the condition that it specifies is *true* (or falls within some small tolerance). An example of how parameters are set is shown in Figure 1.

Many complex business problems involve the allocation of scarce resources. These resources may be anything that the manager has to make decisions about and often include items such as money, time, human resources, materials, etc. These resources become the decision variables for the spreadsheet model and the constraints define their limits or how they might be used. The solution is expressed as the allocation of resources that will maximize or minimize some objective (e.g., profit or cost) while meeting the constraints.

To better illustrate how an end-user might use such tools to model and solve a complex problem we present an example of a City Administrator that must develop a detailed municipal budget that will prove acceptable to a contentious City Council.

GOAL PROGRAMMING EXAMPLE

In this particular problem a City Administrator is charged with developing an annual budget for a small municipality. The City Administrator is relatively competent with common computing applications such as word-processing and spreadsheets, but has no in-house computer support staff to call upon to develop custom budget-planning software. In this case, the City Administrator has worked with local IS students to develop a custom Goal-Programming model to identify an optimal proposal that she hopes will be well received by the local politicians. A team project of this sort is wholly suitable for undergraduate business majors. The example described is based on an actual project, but the names and context have been fictionalized.

As suggested by Wildavsky (1979) and Axelrod (1988) most municipal budgets consist of three major elements: (1) the Base Budget (the current ongoing authorized expenditure base), (2) Current Service Increments (additions to base budget for unforeseen increases due to inflation legislative mandates, workload increases, etc.), and (3) Program Enhancements (new programs or significant upgrades to existing programs). Typically, a fixed revenue ceiling is developed and used as the main constraint when allocating funds to competing programs, departments or programs. This is not dissimilar to corporate budgets that must be planned in light of forecast revenues.

Mid-City is a small town with a population of approximately 20,000 people. The municipal budget is made up of 17 expenditure budget categories and is shown in Table 1. Also shown is the list of requests from various city administrators each requesting funding for their department or area. This requested budget of $24.4 million exceeds the anticipated revenues by nearly one million dollars. The job of the City Administrator is to create a proposed budget that does not exceed tax revenues, does not starve any particular department or program, and that does not run afoul of the Mayor or other powerful City Councilors. Although this example is set in a government forum, managers in all areas of business face similar scenarios (Geiger and Pendegraft, 1995).

In the case of Mid-City, the two dominant politicians are Mayor Smith and Councilman Jones. To better understand the dilemma that the City Administrator faces one must first understand their political priorities.

Mayor Smith was born into a prominent Boston family. She is married to a professor of political science at a University that is located just outside of Mid-City. Smith is active in the arts, environmental and controlled growth movements and serves as the chair of her family's philanthropic foundation that supports a variety of social causes. She is considered an outspoken leader in many of the town's liberal causes. As mayor, Smith also directly oversees the City Administrator and makes recommendations for pay raises and such.

Councilman Jones was born on a nearby farm that his family has owned for nearly a century. Semi-retired, he also owns a large home in the city. Jones is an avid supporter of economic growth and upgrading the transportation options in the city. Jones has served on several Chamber of Commerce committees and enjoys the support of area business leaders and other politically powerful members of the community. He has opposed (with some success) all moves to significantly increase revenues from local property and sales taxes and user fees to fund new programs or liberal causes.

Table 1 - Sample Municipal Budget Problem

Agency Name		**Base Bdgt.**	**Cur. Services Increment**		**Prog. Enhancements**		**Requests**
1.	General	1,714,276	50,000	3%	33,000	2%	1,797,276
2.	Law Enforcements	2,710,510	54,000	4%	27,000	1%	2,791,510
3.	Public Works	347,736	10,500	3%	0	0%	358,236
4.	Other Departments	515,920	30,000	6%	30,000	6%	575,920
5.	Parks-Recreation	624,831	25,000	4%	25,000	4%	674,831
6.	Library Fund	230,066	-230,066	-100%	0	0%	0
7.	Art commission	30,018	600	2%	3,000	10%	33,618
8.	Streets	2,883,555	-550,000	-17%	350,000	8%	2,683,555
9.	Airport Funds	39,375	2,000	5%	2,000	5%	43,375
10.	911 Services	132,628	13,263	10%	6,631	5%	152,522
11.	Bond-Interests	1,063,909	-168,194	-16%	0	0%	895,715
12.	HUD Fund	50,688	0	0%	782,336	1500%	833,024
13.	Water-Sewer	4,660,940	93,218	2%	1,398,280	30%	6,152,438
14.	Sanitation Fund	3,199,107	63,982	2%	0	0%	3,263,089
15.	Parking Fund	109,759	2,000	2%	1,000	1%	112,759
16.	LID Guaranty	0	0	N/A	93,806	N/A	93,806
17.	Water Reserves	1,729.425	0	-	2,305,003	133%	4,034,478
Total Budgets		20,042,793	-603,697	-	5,057,056	-	24,496,152

Table 2. Decision Maker Priorities

	Budget Item	**Mayor Smith**	**Councilman Jones**
1.	General	High	Low
2.	Law Enforcements	Low-Mod	High
3.	Public Works	Low-Mod	High
4.	Other Departments	Mod-High	Low
5.	Parks-Recreation	High	Low-Mod
6.	Library Fund	High	Low
7.	Mid-City Art commission	High	Low
8.	Streets Operation	Mod	High
9.	Airport Funding	Low	High
10.	911 Services	High	Low
11.	Bond-Interests Payments	High	High
12.	HUD Fund	High	Mod
13.	Watre-Sewer	Mod-High	Low
14.	Sanitation Fund	Mod-High	Low
15.	Parking Fund	Low	High
16.	LID Guaranty	High	High
17.	Water/Sewer Reserves	High	High

Other issues are important to understanding the initial operating budget requests and potential reactions by the two council leaders include: Councilman Jones is increasingly concerned about rising crime in the schools; Smith and Jones have both voted in favor of youth recreation programs; Mayor Smith lost a battle to keep the library under control of the city (it was transferred to the county last year), but wants a continuing role for the city in library matters; Mayor Smith views strong support for the local airport as contrary to her controlled growth aspirations; Both support mandatory payments of bond, interest, reserve and sinking funds as fiscally responsible. Both favor Local Improvement Districts (LID), but for different reasons. Mayor Smith can use LIDs for tight control over specific projects while Councilman Jones can avoid charging developers while spreading the costs among the local residents.

Considering the reactions to various budget items is a first step in developing the penalty matrices for the goal program. The City Administrator uses her experience with the two politicians to create the priorities given in Table 2.

These differences in preferences are used in the development of penalty matrices. For these matrices, the decision-maker must assign some level of ‘pain?to the deviation from each objective. The penalty matrix is a special type of constraint that accommodates the real or political costs of certain decisions. In the above example, the mayor may desire a funding level of $120,00 for the Arts Commission. An arbitrary scale of 0 to 100 is chosen with 0 representing ‘no-pain?and 100 representing ‘severe-pain? Alternative scales such as 0 - 10 could just as easily be used. Thus, the mayor’s desires might translate as:

$P_1^+ = 0$, reflecting no pain for funding over $120,000, and

$P_1^- = 50$, reflecting considerable pain for funding less than $120,000

In this case, the penalty for a funding level less than $120,000 is a constant. However, in many cases the decision-maker’s pain may be non-linear. A small deviation from the desired state may be easily tolerated, while a larger deviation may be less so. We could assume that the mayor is willing to accept $110,000 for the Arts Commission without serious complaint, but might be prepared to fight vigorously if the cut is greater than $10,000. In this case the mayor’s numbers might look like the following:

$P_1^+ = 0$, reflecting no pain for funding over $120,000, and

$P_{1,1}^- = 20$, reflecting only a moderate unwillingness to accept $10,000 less, and

$P_{1,2}^- = 100$ reflecting a strong unwillingness to accept less than $110,000.

The priorities and penalties associated with Mayor Smith and Councilman Jones?preferences are presented in Table 2.

The programs and associated numbers represent some level of pain that would be felt if the budget deviated by the given percentage from the amount requested. Note that the penalties are set for both positive and negative adjustments to each budget item.

Table 3. Penalty Matrix

PROGRAM	<-5%	-5%	5%	>5%
Gen. Govt'	100	100	100	100
Enforcement	100	50	20	50
Engineering	100	100	30	100
Parks/Rec	100	100	10	50
Library	100	100	50	100
Arts Comm	100	100	20	50
Street Ops	20	0	100	100
Airport	40	10	100	100
911 Service	100	100	20	50

With the data set up in a spreadsheet model, the solution may be run by the Solver package. Table 3 first shows the initial estimates provided to the City Administrator ("First Pass"). These were used as the preliminary budget proposal that was submitted to the City Council.

As with many organizational decisions, budget building is characterized by a series of hearings, negotiation sessions, and eventual compromises and/or specific victories. Through each round of negotiations the Mayor and Council had to adjust their priorities in light of available financial resources. After each meeting, the City Administrator made appropriate changes to the penalty matrix and the subsequent 'passes' in Table 3 show the Solver's updated budget recommendations.

The budget actually adopted by the City Council is shown adjacent to the fourth recommendation of the Goal Program. The amounts actually adopted were very close to the budget numbers that the Solver was able to calculate. Additionally, the satisfaction by the decision-makers was higher than it might have been if the process used had not recognized and attempted to reconcile the various individual preferences.

Table 4 - Budget Negotiations

							Actual	Variance
Agency		Initial	First	Second	Third	Fourth	Council	From Act.
Name	Base Bdgt	Request	Pass	Pass	Pass	Pass	Allocation	Allocation
General Government	1,714,276	1,797,276	1,714,276	1,714,276	1,628,562	1,628,562	1,646,855	-18,293
Enforcement & Prot.	2,710,510	2,791,510	2,791,510	2,791,510	2,787,460	2,787,460	2,722,204	65,256
Engr. & Pub Wks.	347,736	358,236	358,236	358,236	340,324	340,324	339,923	401
Other Departments	515,920	575,920	515,920	515,920	515,920	515,920	564,578	-48,658
Parks and Rec.	624,831	674,831	674,831	674,831	672,331	672,331	654,373	17,958
Library Fund	230,066	0	0	0	0	0	0	0
Arts Commission	30,018	33,618	33,618	33,618	33,438	33,438	32,431	1,007
Str. Ops & Cap. Con.	2,883,555	2,683,555	1,989,736	1,955,341	2,135,168	2,097,761	2,191,212	-93,451
Airport Funds	39,375	43,375	0	0	0	37,406	31,500	5,906
911 Services	132,628	152,522	152,522	152,522	152,522	152,522	145,794	6,728
Bond and Interest	1,063,909	895,715	895,715	895,715	895,715	895,715	895,715	0
HUD fund	50,688	833,024	833,024	833,024	833,024	833,024	833,024	0
Water and Sewer	4,660,940	6,152,438	6,152,438	6,152,438	6,152,438	6,152,438	6,124,610	27,828
Sanitation Fund	3,199,107	3,263,089	3,259,890	3,263,089	3,199,107	3,199,107	3,177,144	21,963
Parking Fund	109,759	112,759	0	109,759	104,271	104,271	90,916	13,355
LID Guaranty	0	93,806	93,806	93,806	93,806	93,806	93,806	0
Water & Sewer Res.	1,729,475	4,034,478	4,034,478	4,034,478	4,034,478	4,034,478	4,034,478	0
Total Budgets	20,042,793	24,496,152	23,500,000	23,578,563	23,578,564	23,578,563	23,578,563	

CONCLUSION

Managers and advisory staff are often confronted with decision making between conflicting priorities and among projects or activities that cannot be measured in normal economic terms such as return on investment. Computer support for such decisions is too often not the focus of traditional MIS development. This paper has shown how a PC-based algorithm (Goal Programming) can simplify and formulate the conflicting priorities and frame them within operating or policy constraints while providing feasible solutions in a manner of minutes.

This is possible because Goal Programming:

(1) Explicitly considers multiple objectives ?even when they are conflicting

(2) Requires only a simple measurement of deviation from desired objectives

(3) Requires only a relative penalty for deviating from the desired objectives.

(4) Creates a feasible solution that is biased towards the highest priorities but does not ignore lower priority activities when budget constraints make it impossible to fund all high priority items.

(5) Requires only the entry of penalties figures into a table to re-run the program with a new set of relative priorities or "what-if" priority scenarios.

The potential applications of the approach described in the paper are many and varied. In this example, a city budget director can continuously monitor the impact of the changing results of public policy and priority debates by updating the penalty table throughout the budget-building period. Each time a potential consensus is reached, budget allocations consistent with the consensus can be determined and displayed. For longer range planning, sets of priorities can be simulated and provided to decision-makers.

In private industry, ongoing discussions on new product development, maximizing the return of the current product line, expanding market share, functional level resource allocations, and competing sets of capital projects, etc. can be simulated using estimates of relative priorities. Given the short time required for each iteration, the algorithm can even be run during meetings for "*what if*" discussions and for increasing the quality of priority debates.

REFERENCES

Cats-Baril, W.L., & Huber, G.P. (1987). "Decision Support Systems for Ill-Structured Problems: An Empirical Study", Decision Sciences, 350-372.

Feldman, M.S., & March, J.G., (1981). "Information in Organizations as Signal and Symbol", Administrative Science Quarterly, 171-186.

Geiger, J.J., & Pendegraft, N. (1995). "A P.C.-based Goal Programming Application for Priority Based Resource Allocation", Proceedings of the 23 Annual International Business Schools Computing Association, Kalamazoo, MI. 172-176.

Habraken, J., (1996). "Using Lotus SmartSuite Millennium Edition", Boston: Que.

Holsapple, C.W., & Whinston, A.B. (1987)."Knowledge-Based Organizations", The Information Society, 77-90.

Ignizio, J.P. (1985). "Introduction to Linear Goal Programming". Beverly Hills, CA: Sage Publications.

Keen, P. (1987). "Decision Support Systems: The Next Decade", Decision Support Systems, 253-265.

Keen, P., & Morton, M. (1977). "Decision Support Systems: An Organizational Perspective", Reading, MA: Addison-Wesley Publishing.

Keeney, R.L., & Raiffa, H. (1976). "Decisions with Multiple Objectives". New York: John-Wiley.

Lee, S.M. (1972). "Goal Programming for Decision Analysis". Philadelphia: Auerbach Publishers.

Mayo, D. (1986). "Can End-User Computing Be Controlled?" The Internal Auditor, 24-27.

Person, R., (1996). Special Edition Using Microsoft Excel 97, Boston: Que.

Sumner, M., & Sitek, J. (1987). "The Impact of Information Systems Strategy on End-User Computing", Journal of Systems Management, 12-17.

Underdahl, B., (1994). Special Edition Using Quattro Pro 6 for Windows, Boston: Que.

Wood, J.R.G., & Wood-Harper, A.T. (1993). "Information Technology in Support of Individual Decision Making", Journal of Information Systems, 85-102.

AUTHORS

Jeff Butterfield is an associate professor of information systems at Western Kentucky University. Dr. Butterfield has worked for several years as an electronic engineer and manufacturing manager before entering academia. He teaches courses on information technology, telecommunications and project management. His research interests include computer-based training and testing, improving the design process and improving the productivity of technical professionals.

Scott Butterfield is currently a Ph.D. candidate at Georgia State University, in the Department of Accounting. Previous education includes an MS in Accounting (Taxation) from San Diego State University, and a BA in Accounting from Weber State University. Mr. Butterfield has worked with Coopers & Lybrand as a tax consultant, specializing in corporations, multi-state and international taxation, and high net-worth individuals. His research interests include tax planning, compliance, and policy, as well as accounting information systems and computer-based problem solving.

Joseph Geiger is a professor of business at the University of Idaho. Dr. Geiger worked in private industry for several years at Hughes Aircraft and IBM before becoming a university administrator. He has served as the vice president of administration and finance at two universities before becoming a full time professor of finance and management. His publication interests are in applied corporate finance and strategic management.