CASH MANAGER:

A KNOWLEDGE-BASED DECISION SUPPORT SYSTEM FOR CASH MANAGEMENT

Richard D. McBride, Daniel E. O'Leary, and George R. Widmeyer

ABSTRACT

CASH MANAGER is a knowledge-based decision support system (DSS) designed to help the user solve cash management problems. The system is designed to support decisions made by cash managers who are not operations research experts but desire to employ an operations research tool in order to "optimize" their decisions. CASH MANAGER can formulate a cash management problem as an embedded network problem, solve that network problem, ensure feasibility of the network, interpret the output of the solution, and recommend alternative courses of action.

Advances in Mathematical Programming and Financial Planning,
Volume 2, pages 89–106.

ISBN: 0-89232-815-0

I. INTRODUCTION

Surveys of the use of operations research (OR) have found that it is not used for a number of reasons, including a lack of understanding by management of the techniques and a lack of qualified personnel (e.g., Fabozzi and Valente, 1976). Organizations do not have access to the appropriate OR expertise. Since for the decision maker gaining access to OR expertise is not easily remedied, the primary alternative is to employ an OR expert.

Unfortunately, in the traditional OR approach there is a large time lag between the user's recognition of the need to solve a problem and the time when the OR expert can help the user formulate a problem and interpret the solution. That is, the solutions may not be available in real time if the OR techniques must be implemented through an intermediary human OR expert for each application.

In addition, OR techniques may not be used because of the time and difficulty associated with the inputting the data before the OR software can be used to solve the problem.

As a result of these difficulties, OR techniques frequently are not used in those situations where they could assist the decision maker. An alternative to the traditional approach is to develop a system with the built-in expertise of a human OR expert to formulate the problem, input the data, and analyze an OR solution. Such a system would allow the decision maker to take advantage of the power of OR techniques in real time.

A. Cash Management Problem

The cash management problem is choosing from among a portfolio of potential investments or financial instruments, in such a way as to maximize cash flow at the end of a specified period of time. In it this type of problem, OR techniques may be used to chose between alternative investments. However, the nature of the cash management problem requires that decisions need to be made in a timely manner.

B. CASH MANAGER

This paper describes a knowledge-based DSS (decision support system), CASH MANAGER, designed to interact with the cash management expert to support the decision-making process. CASH MANAGER includes expertise from the OR analyst and expertise from the cash management expert. From the OR analyst, it includes the knowledge of how to formulate an embedded network problem (a network linear program with side constraints), how to solve such a problem, and how to interpet the output from the network algorithm. From the cash management expert, it includes knowledge of a set of financial instruments and an approach that is congruent with the approach that cash managers use to solve cash management problems, outside the system.

C. Plan of this Paper

This paper proceeds as follows. Section II reviews the previous research in knowledge-based, OR-based DSS. Section III reviews the previous research in generalized networks and cash management systems. Section IV examines the financial instruments in CASH MANAGER and their network representation. Section V analyzes the artificial intelligence foundations of CASH MANAGER. Section VI discusses the operation, the output analysis, and the implementation of CASH MANAGER. Finally, Section VII summarizes the paper.

II. DECISION SUPPORT SYSTEMS, EXPERT SYSTEMS, AND MATHEMATICAL PROGRAMMING

A. Decision Support Systems and Knowledge-Based Expert Systems

DSS's support decision makers (Keen and Scott Morton, 1978) by providing them with a portfolio of computer-based tools to aid analysis of decision making. Typically, these tools include graphics, statistical analysis tools, OR tools, and other decision aids.

Recently, extensive work has been done in the area of knowledge-based expert systems (ES's) to support decision making. ESs are a branch of artificial intelligence. An ES is loosely defined as a computer program that performs a task in a specific subject domain as well as a human expert would perform the same task. However, based on current technology, ES's do not replace decision makers, but instead support them. In addition, such tools are often considered a part of a DSS. For example, the well-known system MYCIN is referred to as a DSS in Keen and Scott Morton (1978).

Generally, an ES can employ knowledge about the particular application (domain knowledge) by using multiple formats to represent that knowledge (e.g., Barr and Feigenbaum, 1981). Procedural knowledge is knowledge that relates to the way information should be processed. Rule-based knowledge representation generally takes the form "If [condition], then [consequence]." Frame-based knowledge representation uses a "frame" to capture the characteristics associated with a given entity. Characteristics define the knowledge that is of interest. This representation is analogous to the notion of a "template" organization of knowledge.

B. Query Systems and Solver Systems for Math Programs

Currently, OR experts use computer-based *query systems* to analyze general math programs (MPs). These systems (Greenberg, 1983) allow the user to retrieve data, test alternative solutions for a particular decision, and eliminate redundant constraints. These query systems function as DSS's, by supporting the intelligent and knowledgeable user.

Two linear programming query systems have been given substantial attention in the literature: Analyze (Greenberg, 1983) and Peruse (Kurator and O'Neill, 1980). These systems are designed for the analyst who is an "expert" in linear programming, allowing the user to analyze the solution of a linear programming problem using a wide range of commands. However, these systems do not contain any domain-specific knowledge of the particular application. In addition, the OR expert may have little domain-specific knowledge.

OR experts also can use a *solver system*. Schittkowski (1985) developed a user-friendly system that selects a suitable algorithm to solve the problem and puts the data in a format suitable for that algorithm. However, the user must enter constraints and the objective function. Such a system is designed to assist the OR expert in solving a particular problem.

C. Knowledge-Based Expert Systems in Mathematical Programming

Recently, extensive work has evolved in the area of knowledge-based ES's. Knowledge-based ES's have been suggested for interfacing databases and mathematical programming (Minker, 1981, and Greenberg, 1985) by developing a general system to analyze the output of an arbitrary mathematical program.

However, Tomlin (1981) argues that it may be a difficult task to provide a general mathematical programming analysis package. Tomlin suggests that such an approach may not be cost-beneficial. He (p. 444) also raises the question, "When is it simpler (and perhaps safer) for the user to specify all the logical relationships in his program?" That is, it may be more efficient to include domain-specific knowledge into the system. This is typical of human endeavor, where there are, e.g., "OR experts" and "computer experts."

Tomlin's questioning is congruent with a suggestion by builders of ESs. McDermott (1984) and others have indicated that ESs should be developed for self-contained and specific problems. A general mathematical programming problem output analyzer may not have enough specificity.

A few systems have been designed or developed for the formulation and interpetation of specific OR-based problems, without assuming that the user has any OR knowledge. Binbasioglu and Jarke (1986) developed a Prolog-based model that formulated and solved production-based linear programming problems. The model incorporated substantial production management information into the model. O'Leary (1986) designed a system that exploited specificity and knowledge in production scheduling applications.

III. APPROACHES TO CASH MANAGEMENT

A. Linear Programming Formulations of Cash Management Systems

The cash management problem was first modeled as a linear program by Robichek et al. (1965). That linear programming model was later generalized by Mao (1968), Orgler (1969), Pogue and Bussard (1972), and O'Leary and O'Leary (1982) to allow for the possibility of unequal time periods, additional deterministic constraints, chance constraints, and multiple criteria. Maier and Vander Weide (1978) developed a version of the linear programming model that was more "user oriented" than other versions. Their approach focused on making the system accessible to nontechnical users, by allowing easier input of the data and output reports that the user could understand.

B. Network Formulations of Cash Management Systems

Network formulations of the cash management problem have also been investigated. Srinivasan (1974) used the transshipment model to solve the problem. Golden et al. (1979) modeled cash flow problems using networks.

Embedded generalized networks have found use in conceptualizing a broad range of applications. Crum et al. (1983a) designed a model for cash management of multinational companies. Crum et al. (1983b) designed a model for meeting working capital needs using a generalized network approach. Both of these models were aimed at making broad-based decisions, e.g., production and marketing expenditures.

However, generalized networks provide more than a conceptual framework for such problems. There are very efficient algorithms to solve generalized network problems (Brown and McBride, 1984, and McBride, 1985). Thus, formulating problems as generalized networks also provides an efficient implementation format for such applications.

C. Limitations of Previous Models

Barbosa and Hirko (1980) suggested that there are a number of advantages to including OR algorithms in DSS. Unfortunately, the above approaches still require OR expert intervention to allow the use of the OR tool.

Although a DSS framework for investigating the cash management problem was presented in Srinivasan and Kim (1986), they did not implement a cash management DSS. Further, although there have been some efforts to integrate general financial intelligence into computer programs (e.g., Lee, 1984, 1985), little software has been developed to support cash management and other financial decision makers directly.

IV. FINANCIAL INSTRUMENTS AS NETWORKS

The linear programming formulation of the cash management problem discussed in Maier and Vander Weide (1978) is amenable to a generalized network approach. This network approach is based on representing the potential investment choices (financial instruments) as subnetworks that are all combined to form a large network of financial instruments linked over time. The user then faces the problem of deciding what investments and borrowings to make on each of the days of the overall time period of concern.

A. Short-Term Financial Instruments

The most common financial instruments used for short-term financing include money market securities, certificates of deposit (CDs), banker's acceptances, commercial paper, money market funds, and daily repurchase agreements. In addition, coupon-bearing and installment financing instruments also are used in short-term financing. For example, installment financing agreements can be constructed to pay for equipment or merchandise.

Money market securities are highly marketable securities that have short maturity terms. They usually involve little or no risk of default. All money market securities pay interest to their investors by selling at a discount from their face (maturity) value. Short-term U.S. Treasury bills are an example of a money market security.

A CD is a receipt from a federally insured commercial bank for a deposit of $100,000 or more, with the provision that the deposit will not be withdrawn before its maturity date. CDs provide a daily interest return.

A banker's acceptance is a written promise by a bank to repay borrowed funds plus interest that is issued to a borrower. Other banks accept this written promise if they make the loan to the firm. If a bank wants to withdraw the money it has invested in a loan before the loan expires it sells the written promise to another investor. Banker's acceptances may be resold to any number of investors. However, the bank that originated the banker's acceptance retains liability for the loan should the borrower default.

Commercial paper is short-term promissory notes issued by blue-chip corporations. The maturities vary from 5 to 270 days. Paper pays no interest since it is bought at a discount from face (maturity) value. A daily repurchase agreement (daily repo) is an investment for one day only. Short-term coupon-bearing instruments are similar to bonds, where several interest payments are made at regular time intervals prior to redemption.

These instruments can be organized conceptually as borrowing instruments—commercial paper, banker's acceptances, coupon-bearing instruments, and installment financing—and investing instruments—money market securities, CDs, commercial paper, banker's acceptances, daily repos, and coupon-bearing instruments. Instruments appear in both categories since for some users the same instrument will have different purposes.

B. Network Representation of Instruments

Each of the above instruments can be modeled easily using generalized network modeling concepts and techniques coupled with side constraints. Let points in time be represented by nodes and investments and returns be represented by arcs, as in Figure 1.

Node i represents the start and node j the end of the time period. Arc (i,j) represents an investment with return r. If $X_{i,j}$ is invested at time i, then $(1 + r)X_{i,j}$ arrives at time j. Upper and lower bounds can be imposed upon the flow to limit the amount invested. The quantity $(1 + r)$ is called the multiplier for arc (i,j). The multiplier modifies the flow leaving node i to get the flow arriving at node j. If $(1 + r) > 1$, then more arrives at node j (i.e., $r > 0$). If $(1 + r) < 1$, then less arrives at node j. The bounds apply to $X_{i,j}$ and not to $(1 + r)X_{i,j}$.

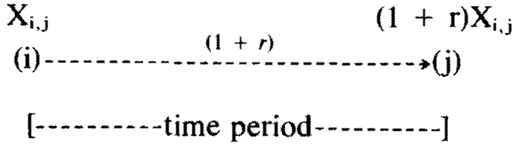

Figure 1. Network structure of investments.

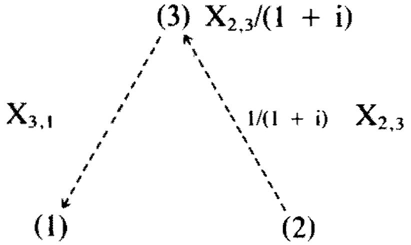

Figure 2. Short-term loan.

In Figure 2, $X_{3,1}$ flows into node 1 as a short-term loan, and $X_{2,3}$ represents repayment at time 2. Node 3 here represents the flow to the lender at time 2. The amount arriving at node 3 is $X_{2,3}/(1 + i)$. The amount lost between nodes 2 and 3 obtained by using a multiplier less then 1 represents the interest paid on the loan. The transaction in Figure 2 can be modeled without node 3, as illustrated in Figure 3.

In Figure 4 a complete short-term financial planning problem is illustrated. There is a shortfall of $-A_1$ in the first time period. In addition, the company receives the amount A_2 at the beginning of time period 2. Arc (2,1) represents a loan at interest rate i for the time period and arcs (1,3) and (2,3) represent two investment opportunities. The flow on the arc (3,3) (the singleton arc pointing out of node 3) represents the total cash at the end of the second time period. We maximize this flow.

Figures 1 and 2 can be used to model market securities, CDs, banker's acceptances, commercial paper, and daily repurchase agreements. The model in Figure 1 is used for investment, and the model in Figure 2 is used for borrowing. The main difference

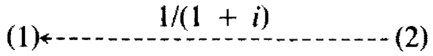

Figure 3. Alternative model of short-term loan.

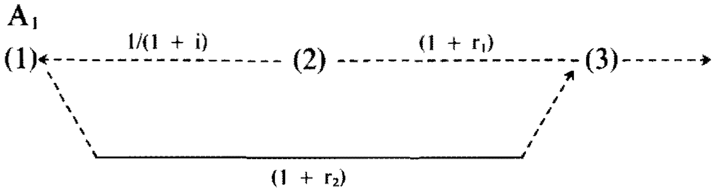

Figure 4. Simple financial planning model.

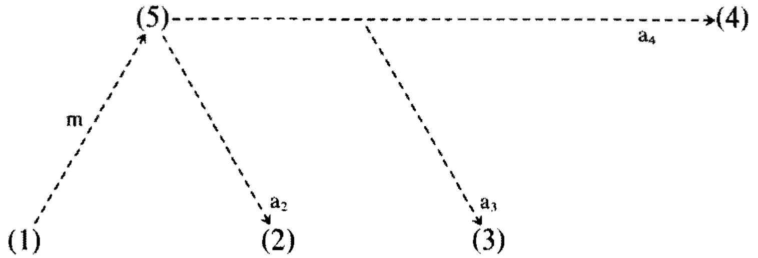

Figure 5. Coupon-bearing and installment financing network.

between these instruments is the return computation. Mathematically, there is no difference between buying an instrument at a discount or receiving interest.

Coupon-bearing and installment financing instruments can be modeled using a network like that shown in Figure 5. Suppose that $X_{1,5}$ is invested at time 1 and that $A = m\ X_{1,5}$ will be payed back at times 2, 3, and 4. In a coupon-bearing instrument an interest payment is made at times 2 and 3, and at time 4 an interest payment along with the principal will be repayed. In an installment financing instrument the flow is $X_{5,2} = X_{5,3} = X_{5,4}$.

V. CASH MANAGER

CASH MANAGER is a user-friendly system that allows a non-OR expert to develop an OR network model and solve it using an OR tool: generalized networks. CASH MANAGER uses a set of menus to elicit information about a portfolio of investment opportunities from the user. The user chooses the set of instruments to be compared and supplies the parameters that explicitly define particular instruments (e.g., days the instrument is to be used). The system formulates the problem as a network using the instrument templates, and then it solves and interprets the overall network that relates those instruments.

A. Implementation and Operation

CASH MANAGER is written in C language. The user interface is developed using Microsoft's Windows Tool Kit, in a mouse-based environment. The system is designed to run on an IBM AT that is equipped with enhanced graphics.

The system makes use of menus to guide the user through the system. Initially, the user faces four major categories: file, edit, build, and solve.

Under "file," the user has the option to start a *new* cash management problem, *open* an existing file, *append* one file onto another, or *save* a file. If the user wishes to edit an existing file, then the instruments that are in the given file are listed. From this point in the menu the user can change the parameters associated with the individual instruments in the particular file under investigation.

If the user wishes to add a new instrument to the portfolio then "build" is accessed. The options include discount investments, general investments, simple loans, installment loans, and general loans. (These options include all of the special cases discussed in Section IV.) At each option, the program solicits from the user the necessary parameter information for the particular applications of the instruments. When the user wishes to obtain the optimal solution to the portfolio of instruments that has been chosen, the user accesses "solve." Under this option, the system ascertains if there have been any "general" investments or loans added to the problem. If none have been added then a generalized network algorithm is used. Otherwise the system uses an embedded network algorithm.

B. Infeasibility Checks

The formulation of a cash flow network can sometimes inadvertently lead to some infeasibilities. As a result, the system is designed to perform two sets of feasibility checks. The first set examines the feasibility of the network before the solution, during the formulation process. The second set occurs during the solution process.

In each case, the error messages are presented to the user in a manner that is easily understood. For example, rather than "arcs entering a node," the term "cash inflows" is used.

Prior to Solution Process

CASH MANAGER performs infeasibility checks on the network before it is solved. When any of the following conditions occur the solution step is not taken.

1. a node with supply or demand but with no incident arcs
2. a node with only arcs entering but no demand
3. a node with only arcs leaving but no supply
4. a node with leaving capacity less than entering supply
5. a node with entering capacity less than leaving demand
6. an arc with zero upper capacity
7. an arc with a negative multiplier

As Part of the Solution Process

Feasibility analysis also can be integrated into the solution process. A singleton arc with a zero profit can be used to account for excess investable funds. An infeasible surplus flow will only occur on that arc if there is no other opportunity to disburse the funds.

Alternatively, a singleton arc with a cost much greater than the cost on any other arc can be used to capture those situations where the network has an infeasibility deficit. Activity on this arc will occur only if there are insufficient funds to meet the needs of the problem. Such an infeasibility deficit occurs when maturing instrument inflows cannot meet demands. These situations are summarized in Figure 6.

Analysis on final linear programming tableau is used to determine how to reduce the infeasibility surplus or deficit. These infeasibilities can be reduced by increasing the capacity of selected instruments. Thus, the system can make recommendations as to which instruments should be increased. As a result, the feasibility process can be used to provide additional feedback to the user about the limitations of the current portfolio of instruments.

C. Analysis of Output

Since, as noted by Geoffrion (1976, p. 81), "the purpose of mathematical programming is insight, not numbers," one of the main concerns of CASH MANAGER is expert analysis of the output.

Figure 6. Infeasibility analysis.

Presentation of Solution

First, CASH MANAGER provides its user with the optimal set of investment instruments, the timing of the investments, and the investment amounts.

Second, the solution is presented in a manner that summarizes the results on a day-by-day basis, for each of those days for which a decision must be made. Throughout, the results are presented in a manner that a cash manager can understand.

Sensitivity Analysis

Second, CASH MANAGER also does a sensitivity analysis of the network to provide further analysis of the output. Sensitivity analysis is done on the portfolio cash inflows and outflows, which are supplies and demands at nodes. The dual variable is used to get the rate of change of the total profit, and sensitivity analysis is used to compute the upper and lower limits of the supply and demand values that apply to the current dual.

D. Cash Management Heuristics and CASH MANAGER

Systems may force the user to relinquish certain problem-solving heuristics in order to adopt the system. However, CASH MANAGER allows human cash managers to continue to employ many of the same heuristics that they have used in the past.

For example, human cash managers may try to invest in only certain high-grade instruments. However, if there are an insufficient number of instruments then they will need to invest in other available instruments. CASH MANAGEMENT uses an analysis of infeasibilities that encourages cash managers to employ the same strategy when they use the system.

As another example, there may be a great deal of uncertainty in the arrival dates of cash inflows or the dates when cash outflows will be required. Indeed, there may be a number of alternatives. As a result, cash managers often must find new investment portfolios, because the flows have changed. CASH MANAGER allows the user to find new alternatives rapidly.

VI. ARTIFICIAL INTELLIGENCE BASIS OF CASH MANAGER

CASH MANAGER was developed using a first-principles approach to structure the knowledge about the financial instruments. The system does not use the traditional rule-based approach to capture knowledge about the financial instruments. Instead, the system uses a frame-based (template-based) approach, which is congruent with human models of professional judgment. In addition, the system is designed to acquire new knowledge about financial instruments from the cash managers. Further, the system includes other heuristics used by cash managers.

A. First Principles

CASH MANAGER is one of the few systems designed using a first-principles approach. Systems that reason from first principles use the structure and function of the devices they are examining (Davis, 1983). Previously first principles were used in, e.g., electronic trouble-shooting, where—roughly speaking—structure is the information that would remain after removing all the textural annotation (functional information) from a schematic (Davis, 1984).

That same interpetation holds in the analysis of financial instruments. The structural organization is the network that derives from the flow of funds, over time and to particular sources (e.g., lenders and investors). The two basic functions are investing and lending. The system formulates the instruments as networks. However, that process is done without the user having to provide that formulation.

There is no presumption that all characteristics of financial instruments can be developed a priori. Instead, CASH MANAGER was developed in concert with the philosophy that change is inevitable. Thus, it is necessary to have the system acquire knowledge about arbitrary types of instruments.

B. Knowledge Acquisition

Knowledge is acquired by CASH MANAGER by putting information about other financial instruments into the system. This is done either by the user or by an OR analyst. The user can add knowledge to the system by defining custom instruments. Then the custom instrument can be used as any other instrument. The information requested for use of these custom instruments includes the name of the instrument, the rate of return (or interest on the loan), the limit of the amount of the investment or loan, the cash inflow and outflow days, and the proportion of the flows associated with each of the days.

In some instances, the custom instrument capacity of the system may not allow the user to develop an instrument that meets the user's needs. In that case, the OR analyst can add new frames to represent alternative financial instruments. CASH MANAGER was not designed to be 100% free of the OR analyst but to allow the user access to embedded generalized network-modeling techniques in a timely manner.

C. Knowledge Representation

CASH MANAGER uses frames (templates) to capture each of the generic types of financial instruments in the system. The contents of the frame vary, depending on the type of instrument. The frames for generic instruments define a means of storing information that can be used to generate a network for a particular instrument.

CASH MANAGER also uses procedural knowledge. The system knows that it has to assemble a set of instruments from which to choose. This process is transparent to the user, who does not need to understand networks or even how the system uses them.

Finally, the system also knows how to perform a sensitivity and infeasibility analysis of the output. This results in suggestions to the user about other potential opportunities for investment and loans.

D. Human Information Processing

Research on human information processing suggests a number of propositions about professional judgment including the following (Gibbins 1984):

1. Experience produces structured judgment guides (templates).
2. Templates are maintained in long-term memory.
3. Judgment begins with a search for a template.
4. Template selection depends on circumstantial fit.
5. Personal characteristics affect template selection.

The frame/template form of knowledge representation used by CASH MANAGER to represent the different instruments is consistent with this theory of professional judgment.

VII. CONCLUSION

CASH MANAGER is a user-friendly, knowledge-based DSS designed for use by a corporate cash manager. The system solicits minimal amounts of information from the user and employs that information in the context of an embedded network, to help the user choose between alternative financial instruments.

The system employs OR expertise and cash management knowledge. Network-modeling expertise is used to take information about a portfolio of financial instruments and develop a network that represents that portfolio. Feasibility analysis is employed to ensure that the user inputs the parameters in an appropriate manner and to guide the user to the addition of other instruments to the portfolio. Sensitivity analysis is used to help the user understand the implications of the output. The system couches its cash management knowledge of financial instruments as network templates (frames).

Finally, CASH MANAGER is not limited to a small set of financial instruments. Instead, the system is designed to acquire information about new financial instruments as the need arises.

REFERENCES

Barbosa, L.C., and Hirko, R.G., "Integration of Algorithmic Aids into Decision Support Systems," *MIS Quarterly*, March 1980.

Barr, A., and Feigenbaum, E.A., *The Handbook of Artificial Intelligence*. Heuris Tech Press, CA: Stanford, 1981.

Binbasioglu, M., and Jarke, M., "Domain Specific DSS Tools for Knowledge-Based Model Building," *Proceedings of the Nineteenth Annual Hawaii International Conference on System Sciences*, 1986, pp. 503–514.

Brown, G.G., and McBride, R., "Solving Generalized Networks," *Management Science* 30, 1984, 1497–1523.

Crum, R.L., Klingman, D.D., and Travis, L.A., "An Operational Approach to an Integrated Working Capital Planning," *Journal of Economics and Business* 35, 1983a, 345–378.

Crum, R.L., Klingman, D., and Travis, L., "Strategic Management of Multinational Companies: Network-Based Planning Systems," *Applications of Management Science* 3, 1983b, 177–201.

Davis, R., "Reasoning from First Principles in Electronic Troubleshooting," *International Journal of Man-Machine Studies* 19, 1983, 403–423.

Davis, R., "Diagnostic Reasoning Based on Structure and Behavior," *Artificial Intelligence* 24, 1984, 347–410.

Fabozzi, F.F., and Valente, J., "Mathematical Programming in American Companies: A Sample Survey," *Interfaces* 7, 1976, 93–98.

Geoffrion, A.M., "The Purpose of Mathematical Programming Is Insight, Not Numbers," *Interfaces* 7, 1976, 81–92.

Gibbins, M., "Propositions about the Psychology of Professional Judgment in Public Accounting," *Journal of Accounting Research* 22, 1984, 103–125.

Golden, B., Liberatore, M., and C. Lieberman, C., "Models and Solution Techniques for Cash Management," *Computers & Operations Research* Vol. 6, 1979, 13–20.

Greenberg, H.J., "A Functional Description of Analyze: A Computer-Assisted Analysis System for Linear Programming Models," *ACM Transactions on Mathematical Software* 9, 1983, 18–56.

Greenberg, H.J., "The Fifth Generation of Mathematical Programming Systems: Towards an Intelligent MPS." Unpublished paper presented at the TIMS College of Practice of Management Science, 1985.

Greenberg, H.J., and Maybee, J.S., *Computer-Assisted Analysis and Model Simplification*. New York: Academic Press, 1981.

Keen, P.G.W., and Scott Morton, M.S., *Decision Support Systems*, Reading, MA: Addison-Wesley, 1978.

Kurator, W.G., and R.P. O'Neill, "PERUSE: An Interactive System for Mathematical Programs," *ACM Transactions on Mathematical Software* 6, 1980, 489–509.

Lee, R., "Information System Semantics (a Logic-Based Approach)," *Journal of Management Information Systems* 1(2), 1984, 18–44.

Lee, R., "Candid Description of Commercial and Financial Concepts: A Formal Semantics Approach to Knowledge Representation." Working Paper 84-85-3-3, Department of General Business, University of Texas, 1985.

Maier, S., and Vander Weide, J.H., "A Practical Approach to Short-Run Financial Planning," *Financial Management*, Winter 1978, 10–16.

Mao, J.C., "Application of Linear Programming to the Short-Term Financing Decision," *The Engineering Economist*, July 1968, 221–241.

McBride, R.D., "Solving Embedded Generalized Network Problems," *European Journal of Operational Research* 21, 1985, 82–92.

McDermott, J., "Background, Theory and Implementation of Expert Systems." Unpublished paper presented at the CPMS Seminar on Expert Systems, Pittsburgh, PA, December 1984.

Minker, J., "Logical Inference as an Aid to Analysis in Large Databases," pp. 405–414, in *Computer-Assisted Analysis and Model Simplification* (H.J. Greenberg and J.S. Maybee, eds.). New York: Academic Press, 1981.

O'Leary, D., "Expert Systems in Mathematical Programming," pp. 137–147 in *Artificial Intelligence for Military Applications* (Silverman, B.G. and Hutzler, W.P., eds.). Operations Research Society of America, 1986.

O'Leary, D., and O'Leary, J., "A Mathematical Programming Approach to the Hospital Cash Management Problem and Extensions," *Proceedings of Fifteenth Annual Hawaii International Conference on System Sciences*, 1982.

Orgler, Y.E., "An Unequal Period Model for Cash Management Decisions," *Management Science*, October 1969, 77–92.

Pogue, G.A., and Bussard, R.N., "A Linear Programming Model for Short-Term Financial Planning under Uncertainty," *Sloan Management Review*, Spring 1972, 69–98.

Robichek, A., Teichroew, D., and Jones, J., "Optimal Short-Term Financing Decisions," *Management Science*, September 1965, 1–36.

Schittkowski, K., "EMP: A Software System for Mathematical Programming." Working Paper, Mathematics Institute, Bayreuth University, 1985.

Srvinivasan, V. "A Transshipment Model for Cash Management Decisions," *Management Science*, June 1974, 1350–1363.

Srinivasin, V., and Kim, Y., "Decision Support for Integrated Cash Management," *Decision Support Systems* 2, 1986, 347–363.

Tomlin, J.A., "Comments on 'Logical Inference as an Aid to Analysis in Large Databases,' " pp. 443–453, in *Computer-Assisted Analysis and Model Simplification* (H.J. Greenberg and J.S. Maybee, eds.). New York: Academic Press, 1981.