

## Invited Review

---

# The use of mathematical programming with artificial intelligence and expert systems

Richard D. McBride and Daniel E. O'Leary

*School of Business, University of Southern California, Los Angeles, CA 90089-1421, USA*

**Abstract:** Researchers have developed artificially intelligent (AI) and expert systems (ES) to assist in the formulation, solution and interpretation of generic mathematical programs (MP). In addition, researchers also have built domain-specific systems either modeled around a mathematical program or which include a mathematical program module. In these systems, the specificity of the domain allows researchers to extend the interpretation or formulation beyond that available from the generic set of assumptions about mathematical programming. Further, researchers have begun to investigate the use of mathematical program formulations of expert systems. The purpose of their research has been to, e.g., understand the complexity of the expert systems and also to examine the feasibility of mathematical programming as an alternative solution methodology for those expert systems. This paper surveys and extends some of that literature that integrates AI/ES and MP, and elicits some of the current research issues of concern.

**Keywords:** Expert systems; Mathematical programming; Artificial Intelligence

### 1. Introduction

The purpose of this paper is to investigate the interface between Mathematical Programming (MP) and Expert Systems (ES) and Artificial Intelligence (AI). There are at least four facets of that interface. First, AI/ES can be used generally to facilitate the use of MP. For example, AI/ES can be used to formulate and interpret MP. Second, MP can be used to ensure that AI/ES systems generate good solutions. MP approaches generate optimal solutions that might be used to solve parts of problems faced by

general AI/ES problems. Third, in some cases AI/ES can be formulated as MP. This may have benefits such as understanding about the complexity of the AI/ES system or getting solutions faster. Fourth, it may be that formulation of parts of MP algorithms as ES/AI problems could facilitate the computational quality of MP algorithms.

#### 1.1. Coupling ES/AI and MP

Winston (1984, p. 2) noted that "one central goal of Artificial Intelligence is to make computers more useful". In that sense, there is substantial need for artificial intelligence in the use of mathematical programming. Often the formulation and interpretation of mathematical programs is too time consuming or too costly to impact

*Correspondence to:* D.E. O'Leary, School of Business, University of Southern California, Los Angeles, CA 90089-1421, USA.

decision making. Further, the use of much MP software requires substantial expertise in operations research and the particular software. Thus, often times operations research just is not used (e.g., Fabozzi and Valente, 1976). Given this perspective there is substantial need for AI in operational research to make mathematical programming 'useful'.

This can be done by using AI/ES to formulate, solve and analyze MP, in order to facilitate use of MP. Both domain-independent and domain dependent approaches are discussed in this paper. In addition, AI/ES can be used in other facets of using MP models, such as model management.

### 1.2. Using MP in AI/ES models

Not only can AI/ES assist in the use of MP models, but the reverse can also be true. AI/ES models typically employ heuristic approaches to solve problems. However, in some cases various subproblems might be effectively and efficiently solved as MP problems. This would guarantee optimal solutions to those portions for which optimal solutions could be developed.

### 1.3. Formulating ES/AI as MP

There is another side to the relationship between AI/ES and MP. MP has been established for a number of years. As a result, many facets of the complexity of particular problems have been established. For example, it has been established that the traveling salesman problem is a very complex problem (NP-complete). Thus, if it could be shown that an AI/ES issue is equivalent to a traveling salesman problem then that would indicate that it is an equally complex problem.

Further, many special formulations of MP problems have been made to solve specific problems, for example, shortest path problems or minimum cutset problems. Thus, if there is a mapping that allows us to view AI/ES as mathematical programs, then we can make use of those formulations to solve analogous problems.

In addition, assuming that we can establish a mapping between MP and ES then those situations when it may be preferable to solve the problem from an MP perspective or an ES perspective could be explored.

### 1.4. Formulating MP solution algorithms with AI/ES

Probably the most neglected aspect of coupling AI/ES into or with MP is integrating AI/ES into making MP algorithms more effective. When researchers first were unsuccessful in their ability to duplicate the results of Karmarkar's well-known algorithm for linear programs, it was rumored that part of the success of Karmarkar's algorithm was Karmarkar. It was rumored that Karmarkar 'assisted' the algorithm with his expertise at critical points in the solution process. Whether or not this was true is not the issue to be discussed here. However, it does raise the question: (How) can expertise be integrated into algorithms to assist the algorithms? If so, then AI/ES can be used to improve existing algorithms. Since little has been developed on this question, it will not receive further discussion in this paper.

### 1.5. Plan of this paper

This paper proceeds in the following manner. Section 2 investigates some of the issues involved with coupling of AI/ES and MP. In addition, that section provides a brief review of some AI concepts.

Section 3 discusses intelligent systems using MP that are domain independent. Some research issues are discussed as a summary for that section. Section 4 examines the implications of building intelligence into domain-dependent application system that either are built around mathematical programming systems or mathematical programs that are built into AI/ES. This section examines some of the implications of computerizing the knowledge of the operational research expert and those implications of using a mathematical programming approach as opposed to a heuristic approach. It also discusses the unique aspects of development and implementation required for establishing the example systems with the embedded mathematical programs.

Section 5 discusses MP representations of AI/ES, and their use to better understand AI/ES applications and the potential use of those MP formulations rather than AI/ES approaches to solve those problems. Section 6 provides a summary of the paper.

## 2. Coupling AI/ES and MP

Although most of the research done to date has focused on expert systems and MP this paper does not limit itself to ES. Instead, a broad-based approach is used, focusing on a number of components of AI. The purpose of this section is to briefly review some key concepts in AI/ES and the nature of coupling AI/ES and MP.

### 2.1. Background: AI and ES

Newell and Simon (1972, p.6) have defined AI as "...the part of computer science devoted to getting computers (or other devices) to perform tasks requiring intelligence". Two areas of AI appear to be of substantial use in coupling with MP, expert systems and case-based reasoning.

ES are a branch of AI that have received substantial attention. There are a number of definitions of ES. The most narrow of those definitions focus on the need for the system to be a rule-based system ('If *A*, then *B*'), based on knowledge gathered from an expert, that functions at the level an expert would function. Other definitions allow alternative forms of knowledge representation (e.g., frames), other sources of knowledge acquisition (e.g., text books), and other levels of performance (e.g., 'satisfactory').

Case-based reasoning (CBR) involves the process of making decisions based on specific examples of what has occurred in the past, rather than a set of rules. Previous cases or plans are stored for use in solving future problems. In addition, means of adapting previous decision making problems are saved. By making previous solutions available to decision makers, the decision maker can anticipate variables of concern and alternative solutions. In addition, past mistakes can be avoided, while short-cuts can be made available. As noted by Hammond [1988, p.17], the ideas behind case-based planning rise out of the simple principle:

If it worked, use it again, and a corollary;  
if it works, don't worry about it.

The refinements of the basic idea come out of a second, equally simple principle:

If it didn't work, remember not to do it again, to which is added:  
If it doesn't work, fix it.

### 2.2. Insight, not numbers

MP, in general, and linear and integer programming, in particular, have been some of the most successful operations research methods to solve a wide range of constrained optimization problems. Although much of the research and many of the headlines are aimed at generating faster algorithms, as noted by Geoffrion (1976), "the purpose of mathematical programming is insight, not numbers".

In order to build such insight into the systems that use MP, there have been at least two basic trends. First, some researchers have investigated AI/ES approaches to have the system formulate, solve, debug and interpret output from the system, while assuming relatively generic domains (domain independent). Second, other researchers have made more detailed assumptions about the domain in which the application is based (domain dependent). These assumptions generally allow the researcher to develop systems that formulate programs based on general inputs from the user and they allow the researcher to be more specific in the investigation of output and interpretation of the meaning of that output.

### 2.3. Shallow versus deeply coupled systems

Kitzmiller and Kowalik (1987) distinguish between shallow and deeply coupled systems. A coupled system is any system linking both numeric and symbolic processing. Such a distinction may be useful in the analysis of coupling of AI/ES and MP. In the case of shallow systems, the MP is treated as a 'black box'. The AI/ES portions of the system have little knowledge about what goes on in the use of MP. On the other hand, in deeply coupled systems, the AI/ES portion of the system has extensive knowledge about MP. For example, it may be able to formulate or interpret the system.

As will be seen later in the paper, most of the systems developed so far coupling AI/ES and MP have been deeply coupled. Since most of these systems are research systems designed to explore the relationship between AI/ES and MP this is not unexpected. In those systems cost and development time typically are not a factor. However, this seems to ignore the potentially cost effective approach of shallow coupling. As a re-

sult, it is probably not unusual that the one shallow coupled system found in the literature was a system developed for an actual business setting.

Deeply coupled systems likely are more robust (Kitzmilller and Kowalik, 1987), with respect to the use of the system's results. This occurs because the AI/ES portion is aware of the limitations of the use of MP and what MP requires. Unfortunately, these systems are not often 'user-proofed' with respect to those limitations of MP. If the system formulates, solves and interprets the solution, the user may not know if the problem is an appropriate use of the system.

#### 2.4. Other implications of coupling AI/ES and MP

There are at least two implications of coupling a mathematical program in an AI/ES system. First, coupling MP and AI/ES indicates that the expertise of the operation research analyst can be captured in a computer program. This suggests that the design, formulation, interpretation and management of those MP can be formulated as a program. Initially, it was unclear as to the ability of developers to accomplish that task. As seen in Sections 3 and 4, there should be no question regarding that task. This also indicates that there probably is no more need to develop prototypes simply to test the ability to develop programs of this sort.

Second, building MP components into an AI/ES indicates that the developer expects an advantage with MP. Those advantages could include that an analytic approach rather than a heuristic approach yields a faster solution time or better solutions. Whether or not those advantages can be realized is an empirical issue, and is discussed later in the paper.

#### 2.5. Development methodologies for coupled AI/ES and MP

A prototyping approach typically is promulgated for expert systems (e.g., Hayes-Roth et al., 1983) However, researchers recently have suggested the use of more traditional software engineering processes (Bull et al., 1987). The development of coupled MP and AI/ES systems has not pointed in any new directions or in any one

direction. Most of the systems appear to have been developed using a prototyping approach, although that is not clear. In addition, the possible existence of unique development methodologies deriving from such coupled hybrid systems also is not clear.

### 3. Domain-independent analysis of mathematical programs

The most general approach to deriving insight from a mathematical program is to make no assumptions about the specific domain, in effect, draw only on the knowledge of mathematical programming and mathematical programming technical knowledge for system intelligence. Research in this area can focus on how to accomplish particular tasks in this process, what tasks can or should be automated, and to what extent these efforts can remain domain-independent.

Systems have been developed with intelligence to

- choose which algorithm is required to solve the particular problem and solve the problem,
- formulate problems as mathematical programs, and
- interpret and debug problems.

#### 3.1. Algorithm choice and solution

Schittkowski (1985, p.2) developed a system that the author called "...the first implementation towards an expert system for mathematical programming". Various options are available in the system for the formulation of various linear and nonlinear functions. The intelligence of the system comes from its ability to choose a suitable linear or nonlinear mathematical programming algorithm to solve the program. The system then writes a FORTRAN source program that will solve the problem. That program is then executed and the numerical results are stored in a database, available for further processing, retrieval or modification.

#### 3.2. Problem formulation

AI/ES can be used to assist in the formulation of MP. The research that has been done to-date

has focused on general aspects of the formulation process. For example, there must be a process to generate variables and constraints in an intelligent manner in order to facilitate problem formation. As will be seen later, if additional assumptions are made about the domain, then greater model specificity can be obtained in the formulation process, since the domain dictates certain problem features. There have been at least two approaches toward the development of intelligent systems designed to assist users in the general formulation of MP.

### *LPFORM*

In a sequence of papers, Murphy and Stohr (1986), Stohr (1988), Murphy, Stohr and Ma (1988), Murphy, Stohr and Asthana (1989), and Ma, Murphy and Stohr (1989a,b) describe an intelligent system, LPFORM, designed to formulate linear programming problems. As will be seen later in this paper, LPFORM also facilitates domain-dependent problem formulation. As noted in Stohr (1988) there are three major components to that system.

First, LPFORM supports three generic classes of inputs: row orientation ('the total tons of product produced in each factory has to be less than certain limits'), activity orientation ('we have buying and selling activities in each of our warehouses') and transportation structures. These classes of inputs are used to define different objects of interest.

Second, LPFORM allows the definition of problems using icons to represent objects being modeled, such as warehouses, transportation flows, etc. This object-oriented approach to representing knowledge allows hierarchical arrangement of objects so that lower level objects inherit properties of higher level objects. Once an object is defined and related to other objects, inheritance limits the number of properties that must be defined at the level of individual objects. This hierarchical structure also is used to structure the graphics used in LPFORM. An algebraic representation is generated as an artifact of the formulation of the problem in terms of its objects.

Third, LPFORM also provides for a relational database for data required in the problem. Further, LPFORM allows the storage and recall of model components that have been stored previously. Users can recall all models using a given

resource or all with a common activity. There is more discussion on LPFORM in Section 4.

### *NETSYS*

McBride and O'Leary (1993) discuss another system aimed at the generic formulation of generalized network models, a class of mathematical programming problems. Many problems can be formulated as generalized networks: transportation, shortest path, assignment and transshipment models.

Generalized network models can represent machine efficiencies, sewage treatment, and many other processes. When network multipliers are interpreted as transforming one good to another, then generalized networks can be used to model manufacturing, blending, production, and a broad range of other processes.

Since the scope of the system is limited to network problems, this allows the system to perform formulation, data editing, etc., based on the assumption of a generalized network structure. The structure assumption also allows the system to perform a number of infeasibility checks, and present information on those infeasibilities, so that the user can address those problems.

NETSYS also uses an object-oriented approach that is facilitated by the development environment, Microsoft's Windows. The system uses network models, built by creating node sets, and taking cross products and 1-1 mappings with subsets of the node sets to create the arc sets. This approach requires minimal user effort to build substantial models.

The system does a minor amount of interpretation of the solution. The system also provides a sensitivity analysis of the problem.

### *3.3. Model interpretation and debugging*

In one of the first systems to integrate AI/ES and MP, Greenberg (1983) described a system called ANALYZE. That system was designed for the linear programming expert, to assist in the investigation of solutions of linear programs. The system has the ability to investigate issues such as feasibility, redundancy and sensitivity analysis at a general level.

Greenberg (1983, 1985) describes a later version of ANALYZE as forming (Greenberg, 1983,

p.333) "...one fundamental component of an intelligent mathematical programming system". In that revised version, an English language discourse version of the earlier system is developed and extended. Greenberg's (1983) system is designed for use after the system has been formulated, however, it can be used during model implementation. As a result, it might be used in conjunction with those systems discussed above for formulation.

Such a system is dependent on knowing the structure of the constraints to ensure that it can debug and interpret the system. For example, it needs to know the type of equations that it can expect. Given those expectations and a user supplied set of tables that correspond to those equations and variables, the system uses that structure to analyze and 'interpret' the resulting tableau.

### 3.4. Implementation characteristics

The implementation of these systems has certain generic concerns including: the use of graphics and knowledge representation.

#### *Graphic interfaces*

One of the most comprehensive discussions of the use of graphics in any of the above systems is Ma, Murphy and Stohr (1989a). The primary focus of that graphic interface is to provide the user with a different representation that allows them to depict their problems in a graphic, rather than mathematical form. Important empirical issues are 'does it make a difference' and 'what are the best approaches'. Ma (1988) has initiated additional research in this area.

#### *Knowledge representation*

Consistent with the Kitzmiller and Kowalik (1987), the dominant form of knowledge representation appears to be objects. Generally the user chooses or generates an object that meets their needs and the system generates equations that correspond to that object. In addition to their correspondence to graphic structures, it seems that the hierarchical nature of objects is a very convenient approach to store knowledge for such systems. Further, recently developed programming vehicles, such as WINDOWS, are object-oriented, furthering the ability to develop such systems.

### 3.5. Model management

An important issue is the reusability of previous model parts or templates. Generally, this is referred to as model management.

An approach that employs AI for model management would likely use a case-based approach to capture and retain knowledge about previous models that have been generated. That knowledge could be used to assist in generating new models or modifying existing models.

In a domain-independent environment, there are a number of characteristics that can be used to structure the cases. These include: who developed the application, when it was developed, who was the client of the application, size of the model, and many other characteristics. In addition, hierarchical relationships within a given characteristic can also be pursued.

The use of case-based reasoning to manage models has received limited attention. As a result, additional research is needed to focus on what are the appropriate characteristics used to capture previous models, how well the case-based approach works in the choice of previous MP models, and a variety of other systems issues.

This model management issue has behavioral and design aspects that are facilitated by investigation of a case-based approach. From a behavioral perspective, to what extent would users and designers employ parts of previous systems? What do expert model designers do? From a design perspective, what is the best way to store, recall and use previous model templates? In addition, what characteristics distinguish those templates that receive frequent re-use - are they domain-dependent? Further, how do we store parts of models for further use? Whenever there are more than a few core models or core users such issues become very important.

### 3.6. Selected additional research issues

There are additional research issues in both the behavioral analysis of the integration of AI/ES and MP and in the design of these systems. This section summarizes some of those issues.

Although the goal of these systems is to help people solve problems, little, if any empirical work has been done to examine the extent to

which the models developed aid in problem solution. Related work on the impact of using domain dependent systems is discussed in the next section.

Virtually all of the systems discussed in this section at some point began to draw on domain knowledge. Focusing only on MP structures is a focus primarily on syntax (e.g., Greenberg, 1987). Generally, the introduction of semantics occurs with the investigation of a specific domain. Unfortunately, there is little research that studies 'how far' we can go in the development of these systems without assuming domain knowledge. Murphy, Stohr and Ma (1988) is an exception. In that paper, there are some formal results that indicate what can be accomplished with syntactical knowledge and what requires semantic knowledge. Although the development of systems provides an empirical frontier of the ability to model between semantic and syntactical, such theoretical results provide important guidelines to future system development efforts.

#### 4. Domain-specific analysis of mathematical programs

There have been a number of domain-specific systems that have integrated MP and AI/ES. In most situations, it appears that the focus is the MP and the AI/ES has been designed to facilitate the development or interpretation or use of the system. In at least one case discussed below, the AI/ES was the focus, but MP was used to provide a better solution to one component of the AI/ES system.

Domain knowledge can facilitate the integration of AI/ES in MP. The domain provides the opportunity for increased specificity in the analysis of the input or output of the program. For example, if it is known what the output variables represent, then it is possible to develop rules to analyze the output, including the solution and the dual variables, just as a human analyst might do.

##### 4.1. Applications

There have been a number of domain-specific applications that employ at least a linear programming module. These applications range from production models to finance models to strategy

models to personnel models. It appears that the most research in this area has been done in production and related models. Some of the applications discussed in this section are domain-based applications of some of the literature discussed in the previous section on domain-independent models.

Most of the models in this section appear to be deeply coupled models. That is, most of these models are intelligent formulators and interpreters of MP for domain-specific problems.

##### *Production management*

Binbasioglu and Jarke (1986, p.215) made heavy use of domain specific knowledge to generate a prototype system for the "...conceptual development and symbolic (as contrasted to numeric) formulation of the model" for production management problems. The model combines the use of syntactic knowledge about linear programming and semantic knowledge about the production management. The system is a prototype that has not been deployed for actual use. The focus of the prototype was to bypass the potentially lengthy and ambiguous development process typically involved in operational research projects. As a result, the system was designed so that the manager could *formulate* a model directly.

Knowledge in the system is represented as frames. Accordingly, the system allows for unique characterization of individual products and resources, and hierarchies in the relationships of machines used in the production of different products.

The system has a linear programming knowledge base and an application knowledge base. The linear programming knowledge uses two different knowledge bases. One contains knowledge about naming and selecting variables, problem types and other issues. Another contains knowledge about obtaining and interpreting parameter values. The system's knowledge about the domain covers a large range of problems, including the allocation of resources to products, relative composition and contribution of products, etc.

##### *Widget production management*

In the discussion of LPFORM, a number of different models are discussed to exemplify the concepts and gain additional meaning through semantic knowledge. Stohr (1988) discusses a pro-

duction management problem that produces widgets and then ships them to warehouses. Murphy et al. (1988) discuss a problem of shipping grain, Ma et al. (1989a) investigate capital capacity, and Ma et al. (1989b) investigate a problem of transporting energy.

In order to formulate the problem, the user needs to specify the basic structure of the problem, using the graphic interface. In the widget example, the user specifies a relationship between Factories and Warehouses and between Raw Materials and Widgets. At the basis of these applications is a 'definition of production' activity provided to the system. This definition includes a declaration of inputs and outputs, upper bounds lower bounds, coefficient definitions, activity name, and other information. In the case of widget production that means defining raw materials as inputs, widgets as outputs, activity name as 'product mix'. Then the user needs to specify the members of the previously established sets Factories, Warehouses, Widgets and Raw Materials.

#### *Production scheduling via transportation model*

Using the version of the transportation model developed by Bowman (1956) for production scheduling, O'Leary (1986) designed an expert system to analyze the output from that mathematical program. That analysis exploited three aspects of the formulation. First, the transportation problem provides knowledge that the right-hand side quantities are sources and requirements, while the solutions to the problem define the amount of a shipment from a source to a destination. Second, the context of production scheduling further defines those constants as forecasted sales and necessary production. Third, the further detail of production scheduling establishes the sources of production and the costs as inventory, regular time wages and overtime wages. All of this domain information provides additional insights over and above the notion that the problem is a generic linear program. Thus, this information can be used to *interpret* the MP solutions, using a rule-based ES.

#### *Sales mix planning and interface to strategic planning*

Lee and Lee (1987) develop a system that analyzes the results from a short-term product mix model with a rule-based system to determine

if the short-term results and long-term objectives are congruent. The output of the analysis by the rules can include a set of changes that should be made to the linear programming model to have consistency in the short- and long-term goals.

Often linear programming is criticized for its focus only on the long run or only on a given objective. This type of application can be used to mitigate many of those criticisms, by introducing additional analysis into the results to answer some of those criticisms.

#### *Intelligent aggregate planning*

Lee and Kang (1988) developed a rule-based system to assist in the consideration of additional qualitative factors in aggregate production planning problems (e.g., Holt, Modigliani, Muth and Simon, 1960). In particular, issues such as employee morale and customer goodwill are not directly included in such models. Thus, a rule-based system was developed to facilitate consideration of those factors within the context of the MP.

The rule-based system evaluates the minimum cost solution in terms of its impact on employee morale and customer goodwill. If any of the qualitative goals related to these issues is unsatisfactory, then trade-offs can be made via new or different constraints for the MP.

Additional extensions and discussions of related models are provided by Lee (1991). For example, similar approaches are discussed for scheduling crude oil delivery. In addition, an approach is discussed for formulating such MP (Lee et al., 1989).

#### *Aggregate production planning*

Donnelly et al. (1991) develop a system, AG-PLAN-LP, that combines a mathematical programming model with heuristics to generate a production plan. Holt, Modigliani, Muth and Simon (1960) formulated the production planning problem as a quadratic program. Although this means that an optimal solution can be developed for the formulated problem, the disadvantages of using such an approach include that management may not be comfortable with the mathematics and the model requires a cost approximation, which may not accurately reflect the actual process.

In a test of the quality of using an ES/MP approach, Donnelly et al. (1991) compared an expert system based entirely on heuristics to AGPLAN-LP. The traditional expert system has roughly 500 rules distributed in four modules (data acquisition, data modification, initial production plan and final production plan). AGPLAN-LP was created by removing the 25 rules in the initial production plan module, and substituting them with 40 rules and a link to a linear programming model.

The authors do a detailed analysis of the quality of the two systems. Apparently, not only was AGPLAN-LP able to generate more inexpensive solutions than the expert system, but it also was able to generate solutions that were more acceptable to the planner. They also found that AGPLAN-LP had more acceptable solutions than those generated solely by a linear program.

#### *Cash management*

In a sequence of papers, McBride, O'Leary and Widmeyer (1989a,b, 1990), discuss an intelligent system for using linear programming to solve cash management problems. Like the Binbasioglu and Jarke (1986) system, CASHMANAGER is designed for a domain user not the operation research expert. Knowledge in the system is stored in frames and rules.

With CASHMANAGER, the user generates a problem by choosing from the set of available financial instruments represented as templates, with corresponding sets of equations. The user must only specify time frame over which the system is operative and the costs and returns associated with the instrument. The system was developed to formulate and solve the problem of choosing the optimal portfolio. In addition, the system then can assist the user in understanding the results, e.g., by analyzing the dual variables.

CASHMANAGER does more than just a conceptual formulation of the problem: it also investigates the formulation for feasibility. In addition, since the system is aimed at relatively naive users, the user is integrated into the process of making sure that the system is feasible. Infeasibilities are displayed to the user in a manner that relates to the instruments and the corresponding dollar flows.

The system assumes an underlying generalized network structure to the cash management prob-

lem. That approach allows the generation of integer solutions in a timely manner.

#### *Debt advisory system*

Dempster and Ireland (1988, 1989) have developed a system called MIDAS. The system is an intelligent debt management decision support system, which supports planning for a Canadian utility. As noted by Dempster and Ireland (1988, p.415), the conceptual design of the system "... provides for selected models appropriate to the decision process; configures and solves the models on user request; explains, interprets and refines their results interactively and assists the user in evaluating alternative borrowing plans".

In terms of the specific mathematical programming module this system is relatively unique, since it focuses on a stochastic programming model. Once output is received from the model it is refined by the system based on rules developed jointly with an experience corporate treasurer and professional underwriters.

The knowledge in MIDAS falls into four main categories: corporate knowledge (goals, planning constraints, evaluation criteria); financial/economic knowledge (debt sources, types of debt available, future rate expectations); modeling knowledge (variables and their explanations, mathematical functions); and system and input/output objects (to handle I/O functions). As with a number of other systems, the frames allow for a parsimonious way to store knowledge, because of the hierarchical relationships that can be established.

#### *Manpower allocation*

Berger and Huttinger (1986) describe an application that is a combination of linear programming and expert systems to solve manpower allocation issues. The ES is used to formulate the MP.

That system has three distinct modules. The first module ranks tasks in terms of cost-benefit ratios, where cost is measured in person-years and benefit is measured as the fraction of an organization's objective that is accomplished by the task. The expert system uses rules and heuristics to convert the input allocations to a set of constraints. Then the linear programming module allocates manpower, subject to the constraint set generated by the ES.

### *Developing financial statements*

Back (1992) developed an expert system that integrated a rule-based approach and a MP approach to assist in the development of financial statements. In Finland the development of financial statements is a complex process with many interacting rules and choices to be made. It generally is so complex that even experts are likely to make the wrong decision. Thus, potentially this system can provide the user with the appropriate decision support to help generate the financial statements.

Back (1992) is one of the few researchers to do behavioral experiments to determine whether the use of the system that she developed actually made a difference. Her analysis found that the use of the system led to correct answers in 94 out of 96 questions whereas traditional support, using a text book to determine if answers were correct, led to only 53 correct answers out of 120 possible problems. The system makes a difference.

### *Auditing judgement*

Kelly et al. (1987), Willingham and Ribar (1988) and Ribar (1988) describe one of the largest financial-based expert systems developed to date. The system was designed to assist public accounting auditors in their analysis of the collectability of loans. As developed by Peat, Marwick, Main and Co., 'Loan Probe' has over 3000 rules. As noted in Willingham and Ribar (1988), it also has an embedded integer programming module.

The integer programming module was developed in an iterative manner. As noted in Willingham and Ribar (1988, p. 183),

"Initially, we attempted to solve this problem by investigating the methods used by practitioners. This attempt resulted in a large, that is, approximately 1000 rule subsystem which generally yielded only a good first approximation. This approach was eventually abandoned and the problem reformulated as an integer programming problem".

The user does not know that the system contains an integer programming module. Instead, the results from the programming module are directly integrated into the system.

This system is different from the others discussed in this section since the focus is not on using an ES to either formulate or interpret the results of the MP. Instead, MP is used to solve a

subproblem in and ES, which then feeds into the rest of the system. This system's use of MP is probably best described as shallow, yet very cost effective. It is likely that more applications of this type will be generated if developers of ES do not limit themselves to a particular type of problem solution approach, whether that is rules or linear programming.

### *4.2. Advantages and disadvantages*

There are some potential advantages and disadvantages of using a coupled AI/ES and MP approach in application domains.

#### *Replacing the operational research analyst*

Coupling a mathematical programming module in an AI/ES replaces some of the role that the operational research analyst might play. Instead of the analyst gathering information from the user, formulating the problem, solving the problem and interpreting the solution, systems have been developed to perform those tasks. Thus, the time and the cost required to develop a model, within the domain of the system, and a solution likely will be substantially smaller.

However, there also may be some drawbacks. Because it is a computer system, it is likely to be locked into a limited number of views of the world. Typically, such systems are rigid and seldom are able to provide unique or creative solutions to unusual problems. The environment may change causing a change in the optimal decision process, yet current generation AI and expert systems rarely are designed to change in response to those environmental changes. The systems discussed apparently do not have any such sensing mechanisms to note when there is a change in the environment. Thus, an important research issue is how to develop systems that know when they are likely to not be useful and how to develop systems that change in response to environmental changes.

#### *Optimality*

The use of optimization models, with rule-based approaches, has certain advantages and disadvantages. In the case of linear programs, algorithms typically rapidly find 'optimal solu-

tions'. However, in the case of integer or nonlinear programs, depending on the processing capabilities, optimal solutions may take substantial amounts of time and intermediate solutions may not be feasible.

Further, although MP provides an optimal solution, it may suffer from being a single dimensional optimality. It is only optimal for the given objective function, and in the case of integer solutions, heuristics aimed to move it slightly away from that solution may provide substantially worse solutions.

Further, the solution derived by an optimization model is 'optimal' for the world for which it has been constructed. This is a limited view of the world, where precision at the 12th decimal place is important. Thus, systems like those of Lee and Kang (1988) that assist in the analysis of qualitative factors might be extended to account for other 'realism factors.' Additionally, it could be important to study the use of multiple criterion decision making (MCDM) MP and AI/ES.

#### *Feasibility*

The use of mathematical programs introduces the problem of feasibility. For example, one potential problem with the Donnelly et al. (1991) approach is that the system just stops whenever there is an infeasible solution to the linear program. As noted by other researchers (e.g., Murphy and Stohr, 1986), this is not a desirable approach for supporting decision making. Thus, research needs to be done on how users react to infeasibilities and different system responses, how relatively naive users can be asked to fix such infeasibilities and what intelligence the system needs to attack those infeasibilities.

#### *Explanation*

As noted in Swartout (1981, p.815), "to be acceptable, expert programs must be able to explain what they do and justify their actions in terms understandable to the users". With MP there generally is no 'explanation' as to 'why' the model chose certain results. Even when an explanation can be generated it is limited to expression in terms of the MP structure, e.g., constraints. Although Kosy and Wise (1984) provide a means of interpreting some fundamental accounting equations, the use of MP leads to substantially more difficult explanation situations.

ES have a tradition of 'explanation' with solution. If MP are to be coupled with AI/ES then in order to maintain that tradition, research needs to be done in the area of explanation of why one MP formulation was arrived at and why one solution is preferred over another. Some research has been done by Greenberg (1983, 1985, 1987); however, additional research is necessary.

#### *4.3. Additional issues in systems design and development: Knowledge acquisition, system development, testing*

Coupling of AI/ES and MP leads us to question whether or not there are any unique requirements or approaches for the design and development of such coupled systems.

#### *Knowledge acquisition*

There has been little investigation into the necessity of alternative forms of knowledge acquisition for systems that are either shallowly or deeply coupled AI/ES and MP systems. However, there has been some initial research into knowledge acquisition from experts for generic analytical representation (Fischhoff, 1989). Unfortunately, that study did not deal explicitly with the case of MP. As a result, knowledge acquisition in coupled systems of the type discussed in this paper remains a research area.

#### *Development*

Piaget (1973) noted that "all mathematical ideas begin by a qualitative construction before acquiring a metrical character". Piaget also noted that by moving too rapidly from the qualitative to the quantitative can impact the ability to understand the mathematics underlying a given process. In terms of systems development this can mean that unless there is an initial explicit effort to include a mathematical program in the overall system, that the mathematical programming nature of subsystems is likely to be iteratively discovered. As noted above, Loan Probe was developed in concert with those notions. Two rule-based versions were developed before the mathematical programming-based version was developed.

#### *Testing*

There are at least two basic testing questions that need to be addressed. First, does the ap-

proach with a coupled AI/ES and MP system do as well as or better than only an AI/ES approach or only an MP approach. This is the critical problem of establishing a bench mark discussed by, e.g., Hayes-Roth (1989) and others. Second, what unique approaches must be used in the verification and validation of coupled AI/ES and MP systems (Cohen and Howe, 1989).

In the first case where there are MP and non-MP versions of the system, as in the case of the Loan Probe or the aggregate production planning problem, previous versions offer a unique benchmarks to verify and validate the output from the coupled system. For example, solutions provided by non-MP versions can be used to test MP versions, where ideally, later versions with the MP are at least as good, based on a single criteria, as the heuristic-based system.

In the second case, there are substantial bodies of literature in the areas of verification and validation of AI/ES (e.g., O'Leary, 1987). Those efforts usually are aimed at the overall performance of the system or take advantage of unique knowledge base structures.

However, structural analysis of knowledge bases rarely considers more than a single form of knowledge representation at a time. Although there have been investigations into verification procedures for different representations (e.g., rules), there has been little research into hybrid types of systems such as the coupled AI/ES and MP systems discussed in this paper. Systems that couple MP and AI/ES add additional complexity to the testing process, since they include multiple forms of knowledge. This coupled form of knowledge appears to have received only limited research to this point (e.g., O'Leary, 1988), and is an area where research is ongoing.

## 5. Using MP instead of an AI/ES formulation

Probably the most recent use of MP and AI/ES is to use MP as an alternative formulation to the AI/ES formulation typically either for solution purposes or to better understand characteristics of the AI/ES formulation (e.g., Jersolow, 1988). In these cases the MP is formed either based on a network representation of the knowledge base or as a set of logical propositions.

### 5.1. Network representations

Many forms of knowledge representation are either formulated as a networks or can be represented as networks. For example, the knowledge representation format, semantic networks, are network or tree representations of knowledge (e.g., Rich, 1983); frames are linked in either a network or tree format (e.g., Winston, 1984); influence diagrams (e.g., Howard, 1989) are Bayesian network representations of knowledge; and rules can be represented as a network (e.g., O'Leary, 1990). For example the rule 'if  $a$ , then  $b$ ' can be viewed as a variable  $x_{a,b}$ .

Since optimization problems in networks can be represented as MP, this indicates that optimization issues in the areas of knowledge bases can be investigated using MP formulations. This allows the introduction into AI/ES, of a wide range of problems that have already been solved in MP. For example, Glover and Greenberg (1988) and O'Leary (1990) develop MP formulations for verification of rule-based knowledge bases, e.g., detecting circular reasoning.

MP formulations also can be used to study characteristics of problems in knowledge bases. For example, O'Leary (1990) formulates the problem of ordering rules in a knowledge base as a traveling salesman problem. This suggests that level of complexity of a problem in knowledge basis can be found by finding equivalent MP problems.

### 5.2. Logic operators

Dhar and Ranganathan (1990) view knowledge as a set of logical propositions. Since Dantzig (1963) showed how logical propositions can be modeled in an integer programming approach, the extension from knowledge bases to MP is clear. For example, the clause ' $x_1$  or not  $x_2$  or  $x_3$ ' can be written as the following inequality:

$$x_1 + (1 - x_2) + x_3 \geq 1,$$

where the values of true and false are denoted as 1 and 0. This approach is discussed in more detail in Hooker (1988). Dhar and Ranganathan (1990) provide an empirical comparison between the expert system approach and the integer programming approach.

Dhar and Ranganathan (1990) found that using an integer programming approach yielded unpredictable decision times, varying from a few minutes to a few days. In contrast, the expert system's time was less volatile. Consistently the solution time required between one and two hours on a SUN-3. In addition, in contrast to the integer programming approach, if the expert system could not find a solution, it generated a partial solution, if there was one. The integer programming approach consistently was unable to find feasible solutions after many hours of running time. This has two difficulties: length of time and the impact of not have any feasible solution after all that time. This can prove additionally inconvenient when alternative plans or approaches need to be established.

The integer programming approach typically generated solutions that were judged at least partially inappropriate by a human expert for three primary reasons: single objective limitations, compiled knowledge limitations and global optimization limitations.

The objective function in a linear program has a single goal. However, as researchers in multiple criteria decision making argue, human experts try to respond to multiple goals. In addition, single objectives, by their very nature of excluding other objectives, sometimes lead to unsatisfactory solutions. Some steps can be taken to account for multiple objectives, such as objectives as additional constraints.

Compiled knowledge refers to the sensitivity of MP to cost coefficients. As noted by Dhar and Ranganathan (1990, p.332), "the basic problem with the cost coefficients is that they incorporate a lot of compiled knowledge about preferences, flexibility and trading criteria which makes the behavior of the system somewhat unpredictable". Further, in some cases, knowledge may be difficult to express in the structure of constraints. For examples, anomalies in processes that are used in some situations and not in others can be difficult to model.

Global optimization limitations refers to the basic tendency of MP to move to the lower bounds or upper bounds on the variables and the general inability to explain why a particular solution was chosen.

Thus, in a comparison of MP and AI/ES it appears in somecases the local optimality ap-

proach of ES is better received by users of the system and by those for on whom implementation of the system depends. Although MP provides a global optimum, that optimum often is for a different representation of the problem than the one for which an ES/AI approach could be formulated.

## 6. Summary

This paper has surveyed some of the coupled AI/ES and MP systems. The general capabilities of these systems range from formulating mathematical programming problems from limited raw data, to interpreting output from the systems, to extrapolating changes in the mathematical programs based on the output to deciding how to solve these systems.

This approach to decision making has some apparent advantages. To a certain extent, these systems are aimed at replacing operational research analysts, in order to improve the cost and time required to develop a solution to a decision problem that includes a constrained optimization problem. Further, mathematical programs exploit the decision making technology to the advantage of the user of such systems.

However, this approach is not without limitation. Such systems may lock decision makers into decision models that are inappropriate. Similarly, systems that include a mathematical programming module are likely to furnish few explanations for why a given solution was pursued.

The development of systems that include such a module may be part of a gradual movement from the qualitative to the quantitative that occurs with understanding a process. However, by integrating multiple forms of knowledge processing (numeric and symbolic) into a given system, testing of the resulting system may be complicated further. Finally, implementation of such systems is facilitated by software that allows the user to access such models outside the basic software.

## Acknowledgement

The authors would like to acknowledge the comments of Bruce Murtagh and the referees on an earlier version of this paper.

## References

- Asthana, A. (1989), "LPGRAPH: An iconic language for formulating linear programs", Unpublished Ph.D. Dissertation, Stern School of Business, New York University.
- Back, B. (1992), 'Assisting inexperienced accountants in developing financial statements', *International Journal of Intelligent Systems in Accounting, Finance, and Management* 1/3, 155-162.
- Berger, M., and Huttinger, J. (1986), "The Expert System - Linear Programming approach (ESLP)", paper presented at the ORSA/TIMS meeting, October, 1986.
- Binbasioglu, M., and Jarka, M. (1986), "Domain specific DSS tools for knowledge-based model building", *Decision Support Systems* 2/3, 213-223.
- Bowman, E. (1956), "Production scheduling by the transportation method of linear programming", *Operations Research* 4, 100-103.
- Bull, M., Duda, R., Port, D., and Reiter, J. (1987), "Applying software engineering principles to knowledge-base development", in: *Proceedings of the First Annual Conference on Expert Systems in Business*, Learned Information, NJ, 27-38.
- Cohen, P., and Howe, A. (1989), "Toward AI research methodology: Three case studies in evaluation", *IEEE Transactions on Systems, Man, and Cybernetics* 19/3, 634-646.
- Dantzig, G. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ.
- Dempster, M., and Ireland, A. (1988) "A financial expert Decision Support System", in: G. Mitra (ed.), *Mathematical Models for Decision Support*, NATO ASI Series, Vol. F8, Springer-Verlag, Berlin.
- Dempster, M., and Ireland, A. (1989), "Object-oriented model integration in MIDAS", *Proceedings of the Twenty-Second Annual International Conference on System Sciences*, Volume III, 612-619.
- Dhar, V., and Ranganathan, N. (1990), "Integer programming vs. Expert Systems: An experimental comparison", *Communications of the ACM* 33/3, 323-336.
- Donnelly, R., Saniga, E., Chester, D., and Pohlen, M. (1991), "Expert Systems in aggregate production planning", Unpublished Paper.
- Fabozzi, F., and Valente, J. (1976) "Mathematical Programming in American companies: A sample survey", *Interfaces* 7, 93-98.
- Fischhoff, B. (1989), "Eliciting knowledge for analytical representation", *IEEE Transactions on Systems, Man, and Cybernetics* 19/3, 448-461.
- Geoffrion, A. (1976), "The purpose of mathematical programming is insight, not numbers", *Interfaces* 7, 81-92.
- Glover, F., and Greenberg, H. (1988), "Logical testing for rule-based management", *Annals of Operations Research* 12, 199-216.
- Greenberg, H. (1983), "A functional description of ANALYZE: A computer assisted analysis system for linear programming models", *ACM Transactions on Mathematical Software*, 9, 18-56.
- Greenberg, H. (1985), "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. (1987), "A natural language discourse model to explain linear programming models and solutions", *Decision Support Systems* 3, 333-342.
- Hammond, K., "Case-based planning: Viewing planning as a memory task", in: J. Kolodner, *Case-Based Reasoning, Proceedings of a Workshop*, Morgan Kaufman, Palo Alto, CA.
- Hayes-Roth, F. (1989), "Towards benchmarks for knowledge systems and their implications for data engineering", *IEEE Transactions on Knowledge and Data Engineering* 1/1, 101-10.
- Hayes-Roth, F., Waterman, D., and Lenat, D. (1983), *Building Expert Systems*, Addison-Wesley, Reading, MA.
- Holt, C., Modigliani, F., Muth, J., and Simon, H. (1960), *Planning Production Inventories and Work Force*, Prentice-Hall, Englewood Cliffs, NJ.
- Hooker, J. (1988), "A quantitative approach to logical inference", *Decision Support Systems* 4/1, 45-96.
- Howard, R. (1989), "Knowledge maps", *Management Science*, 35/8, 903-921.
- Jeroslow, R. (1988), "Alternative formulations of mixed integer programs", *Annals of Operations Research* 12, 241-276.
- Kelly, K., Ribar, G., and Willingham, J. (1987), "Interim report on the development of an Expert System for the auditor's loan loss evaluation", in: *Proceedings of the Touche Ross / University of Kansas Audit Symposium 1987*, 167-188.
- Kitzmler, C., and Kowalik, J. (1987), "Coupling symbolic and numeric computing in knowledge-based systems", *AI Magazine* Summer, 87-90.
- Kosy, D., and Wise, B. (1984), "Self-explanatory financial planning models", in: *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, 1984, 176-181.
- Lee, J., (1991), "Integration of Artificial Intelligence with optimization", Unpublished paper, Department of Management Science, Korea Advanced Institute of Science and Technology, Seoul, Korea.
- Lee, J., and Kang, B. (1988), "Intelligent production planning analysis using the post model analysis approach", in: E. Turban and P. Watkins, *Applied Expert Systems*, North-Holland, Amsterdam, 87-106.
- Lee, J., and Lee, H. (1987), "Interaction of strategic planning and short-term planning: An intelligent DSS by the post-model analysis approach", *Decision Support Systems* 3, 141-154.
- Lee, J., Chu, S., Shim, S., and Kim, M. (1989), "A knowledge-based formulation of linear programming models using UNIK-OPT", *2nd International Workshop on Artificial Intelligence in Economics and Management*, North Holland.
- Ma, P. (1988), "An intelligent approach towards formulating linear programs", Unpublished Ph. D. Dissertation, New York University.
- Ma, P., Murphy, F., and Stohr, E. (1989a), "A graphics interface for linear programming", Unpublished Paper, Center for Research on Information Systems, New York University, February, 1989.
- Ma, P., Murphy, F., and Stohr, E. (1989b), "Representing

- knowledge about linear programming formulations", *Annals of Operations Research* 21, 149-172.
- McBride, R. (1988), "NETSYS - A generalized network modeling system", Working Paper, University of Southern California, School of Business, August, 1988.
- McBride, R., and O'Leary, D. (1993), "A modeling system for financial applications", *Advances in Mathematical Programming and Financial Planning* 3, 119-136.
- McBride, R., O'Leary, D., and Widmeyer, G. (1989a) "A system for supporting cash management decisions", in: *Proceedings DSS-89*, Sponsored by The Institute of Management Sciences, San Diego, CA, June, 1989.
- McBride, R., O'Leary, D., and Widmeyer, G. (1989b) "A knowledge-based system for cash management with management science expertise", *Annals of Operations Research* 301-316.
- McBride, R., O'Leary, D., and Widmeyer, G. (1990), "CASH MANAGER: A knowledge-based DSS for cash management", *Advances in Mathematical Programming and Financial Planning* 2, 89-106.
- Murphy, F., and Stohr, E. (1986), "An intelligent system for formulating linear programs", *Decision Support Systems* 2/1, 39-47.
- Murphy, F., Stohr, E., and Asthana, A. (1989), "Representation schemes for mathematical programming models", Unpublished Paper, Center for Research on Information Systems, New York University, September, 1989.
- Murphy, F., Stohr, E., and Ma, P. (1988), "Composition rules for building linear programming models from component models", Unpublished Paper, Center for Research on Information Systems, New York University, July, 1988.
- Newell, A., and Simon, H. (1972) *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.
- O'Leary, D. (1986), "Expert Systems in mathematical programming", Presented at the Symposium for Artificial Intelligence in Engineering, October, 1985; published in: B. Silverman and W. Hutzler (eds.), *Artificial Intelligence for Military Applications*, Operations Research Society of America.
- O'Leary, D. (1987), "Validation of Expert Systems", *Decision Sciences* 18/3, 468-486.
- O'Leary, D. (1988) "The use and validation of optimization methods in Expert Systems", Unpublished paper, presented at EURO IX, TIMS XXVI, Paris, July 6-8, 1988.
- O'Leary, D. (1990), "Measuring and managing complexity in knowledge-based systems: A network and mathematical programming approach", Presented at the New York ORSA/TIMS Meeting, October 1989; published in: D. Brown and C. White (eds.), *Artificial Intelligence and Operations Research*, Sponsored by ORSA/TIMS, Kluwer Academic Publishers, Dordrecht.
- Piaget, J. (1973), *To Understand is to Invent*, Grossman, New York.
- Ribar, G. (1988), "Development of an Expert System", *Expert Systems Review* 1/3, 3-8.
- Rich, E. (1983), *Artificial Intelligence*, McGraw-Hill, New York.
- Swartout, W. (1981), "Explaining and justifying expert consulting programs", in: *Proceedings of International Joint Conference on Artificial Intelligence*, 815-823.
- Schittkowski, K. (1985), "EMP: A software system for mathematical programming", Unpublished Paper, Mathematisches Institut, Universität Bayreuth, Bayreuth, Germany.
- Stohr, E. (1988), "Automated support for formulating linear programs", Unpublished Paper, Center for Research on Information Systems, New York University, February, 1988.
- Willingham, J., and Ribar, G. (1988), "Development of an expert audit system for loan loss evaluation", in: *Auditor Productivity in the Year 2000, 1987 Proceedings of the Arthur Young Professor's Roundtable*, Arthur Young, Reston, VA.
- Winston, P. (1984), *Artificial Intelligence*, Addison-Wesley, Reading, MA.