

An intelligent modeling system for generalized network flow problems: With application to planning for multinational firms

Richard D. McBride and Daniel E. O’Leary
*School of Business, University of Southern California,
Los Angeles, CA 90089-1421, USA*

E-mail: mcbride@sba2.usc.edu; oleary@rcf.usc.edu

This paper presents a discussion of the **Generalized Network System (GNS)**, a system that captures knowledge about generalized network flow problems, in order to help users formulate, solve and interpret generalized network flow problems. Although previous researchers have built intelligent systems that incorporate knowledge about linear programming, this system includes more specific knowledge about generalized network flow problems. GNS is illustrated using it in a setting that requires international financial and production planning. In particular, these multinational planners need to be able to plan and replan rapidly. In addition, they need to be able to model complex events and organization structures. For example, multinational planners need to be able to plan for production in multiple countries and repatriatization of funds. GNS allows users to meet these needs.

1. Introduction

Within the artificial intelligence (AI) community there is the view that (Winston [21, p. 2]) “One central goal of artificial intelligence is to make computers more useful.” Although operations research (OR) tools can be used to guide the user to better solutions than other approaches, researchers have unfortunately found few users of the portfolio of OR techniques, including mathematical programming (e.g., [11]).

As a result, there has been substantial interest in the relationship between AI and OR, in particular, mathematical programming (MP). Researchers are interested in the question, “Can AI make OR useful (and used)?” Accordingly, a number of special issues of journals, such as *Annals of Operations Research*, have focused on the integration of AI and MP. National and international meetings (e.g., American Association for Artificial Intelligence, 1992) have had special sessions and workshops relating to the topic.

Research integrating AI and MP has concentrated on building knowledge of MP into systems to solve MP problems. Generally, this has meant knowledge about *linear programming problems* and their structures. Similarly, the system discussed in this paper uses knowledge about *generalized network flow* problems in order to exploit our knowledge of the underlying structure of those problems to facilitate problem formulation and solution. It does this in the context of an example of financial planning for a multinational firm. Concentrating on specific knowledge about generalized network flow problems is consistent with previous literature integrating AI and MP, as discussed below. Further, focusing on specific structure has enabled the solution of generalized network flow problems using algorithms that are much faster than solving those problems as general linear programming problems (e.g., [5]).

1.1. Application: Planning for multinational firms

Multinational firms need to be able to plan where to produce goods and when to repatriate funds earned in other countries. Multinational firms need to be able to plan rapidly because currency values can fluctuate rapidly. In addition, multinational firms face a complex assortment of changing currency and production situations. As a result, users need robust and flexible planning tools to help them adapt to numerous situations.

Firms do not have time to have a planner contact an OR expert, meet for lunch, talk about the planning problem, have the OR expert work for the next six months constructing a model, and then have the planner ask the OR expert to solve the problem one day and then the next day solve another version of the problem. Instead, there is a need for a tool that allows firms to rapidly model complex integrated multinational corporate structures and international currency situations.

1.2. This paper

As a result, this paper presents an intelligent system called the **Generalized Network System (GNS)**, that can be used to formulate, solve and interpret generalized network flow problems. Because the system is easy to use and facilitates real-time response, it can facilitate planning and replanning. In the example presented in this paper, GNS helps the user generate an integrated production and currency valuation model of the firm. The system then formulates that model as a generalized network flow problem. GNS tests the feasibility of the model and then solves the model, and presents the results to the user. The system can be used for replanning, since parameters can be easily changed and nodes and arcs can be easily added or deleted.

This paper proceeds as follows. Section 2 briefly reviews some of the previous literature on integrating AI and MP. Section 3 provides a brief background of the use of generalized network models for planning. Section 4 reviews the knowledge embedded in GNS. Section 5 illustrates the use of the system. Section 6 briefly summarizes the paper and discusses some extensions.

2. Previous research integrating mathematical programming and artificial intelligence

Researchers have developed integrated AI and MP systems for general analysis of linear programs, within the context of a number of business situations.

2.1. General systems

The literature contains three forms of general integration of AI and MP, including systems that: formulate problems for solution by a linear programming algorithm; determine which algorithm should be used to solve a particular problem; and interpret and debug linear programming models. These systems contain substantial knowledge about linear programming models.

A sequence of papers, including Murphy and Stohr [18] and Ma et al. [16], described an intelligent system, called LPFORM, that formulated linear programs for users. LPFORM supports three classes of inputs: row orientation, activity orientation and transportation structures used to define different objects of interests, within the context of linear programming.

Schittkowski [19] developed EMP, which has the ability to choose a suitable linear or nonlinear programming algorithm to solve the program. EMP then writes a FOTRAN program to solve the particular problem under consideration. That program is then executed and the results are stored in a database for further analysis.

Perhaps the first research that built knowledge into mathematical programming systems was Greenberg's [12, 13] work on debugging and explaining linear programs. Greenberg's [13, p. 333] ANALYZE is an "English language discourse model to explain linear programming models and their solutions". ANALYZE detects redundant constraints, infeasibility and other issues.

2.2. Applications

The primary focus of applications to date has been on production planning, with little emphasis on financial planning or international financial planning problems. Instead, researchers have primarily focused on solving production-based problems. Binbasioglu and Jarke [5] developed a model that helps the user formulate a linear programming example from scratch to solve a production scheduling problem. Lee and Lee [15] developed a system designed to consider short and long term effects of the sales mix. The system solves a linear programming problem and then analyzes the output to determine if the solution satisfies the long-run interests of the company. Takahara et al. [20] developed a platform for constructing intelligent systems using a hierarchical connection of a numeric solver and a nonnumeric evaluator, which evaluates the solution heuristically. They illustrate their system using an example drawn from the job assignment problem.

There has been limited research on integrating mathematical programming and artificial intelligence to solve financial problems. Dempster and Ireland [9,10] developed a decision support system that integrates stochastic programming into an intelligent system for supporting borrowing planning for a public utility. Once output is received from the programming model, it is refined by the system based on rules developed jointly with an experienced corporate treasurer and professional underwriters. Back [1,2] and Back and Back [3] describe an expert system for assisting in the development of financial statements, with particular interest in understanding the impact of income smoothing. A mathematical program, embedded in an expert system, generates the recommended solution to the user.

2.3. Contribution of our system

Previous researchers have focused virtually exclusively on linear programming problems. As a result, the system discussed in this paper is concerned with using knowledge about generalized network flow problems to build an intelligent system to help formulate, ensure feasibility and interpret results. It does this in the context of international financial planning, an application area that also has received limited attention in the integration of AI and MP.

3. Generalized network flow problems: Models of multinational firms

This section summarizes some of the relevant previous research on network models and their use in planning for multinational firms. Then, the basic generalized network flow model used in this paper is generated.

3.1. Generalized networks

Although the linear programming algorithms can be used to solve virtually any linear program, there are computational advantages to developing algorithms for specific types of problems. A particularly computationally efficient algorithm has been developed for generalized network flow problems by Brown and McBride [5]. In addition, McBride [17] developed a computationally efficient approach for the solution of embedded networks. Those algorithms can solve even very large problems rapidly. Solution of those generalized network flow problems is speeded up by exploiting the unique structure of the problems.

3.2. Models of multinational firms and mathematical formulation

Perhaps the first network models for multinational firms were developed by Crum [6], who modeled the process of planning cash conversion by multinational corporations as a generalized network flow problem. The formulation presented in

this section comes from Crum et al. [7]. This example problem is chosen because it illustrates the importance of having tools available to easily formulate and solve generalized network flow problems and to illustrate system use.

Let x_{ij} be the arc flow from node i to node j . Let L_{ij} be the lower bound on the flow on that arc. Let U_{ij} be the upper bound on the flow on that arc. Let c_{ij} be the cost of shipping one unit on arc (i, j) . Let p_{ij} be the multiplier on the flow of one unit on arc (i, j) . Thus, $p_{ij} * x_{ij}$ is the amount of flow entering node j from node i . Let b_i be the supply or demand at node i , where supply is denoted as a positive quantity and demand is a negative quantity.

Each arc in a generalized network represents a variable and each node represents an equation. Let \mathbf{A} be the set of feasible arcs in a network. The general mathematical program can be written as

$$\begin{aligned} & \text{minimize} && \sum_{(i,j) \in \mathbf{A}} c_{ij} * x_{ij} \\ & \text{subject to} && \sum_{(i,j) \in \mathbf{A}} x_{ij} - \sum_{(i,j) \in \mathbf{A}} p_{ij} * x_{ij} = b_i, \quad i = 1, \dots, m, \\ & && 0 \leq L_{ij} \leq x_{ij} \leq U_{ij}, \quad (i, j) \in \mathbf{A}. \end{aligned}$$

Different nodes would be used to represent currency and production resources in both the parent and each of the subsidiaries. Arcs would be used to represent direct transfer of cash and product to different nodes.

3.3. Availability of data

The model assumes that up to four primary types of conversion data and processes are available. First, there can be linear data that relates to the cost of converting foreign currency to foreign product. Second, there can be linear data that relates to the cost of converting domestic currency to domestic product. Third, there can be linear data that relates conversion of foreign currency to domestic currency (or conversely). Fourth, there can be conversion of foreign product to domestic product (or conversely).

Associated with each type of conversion there is a per unit cost. In addition, there can be either or both lower and upper bounds on the amount of the conversions.

3.4. Need for GNS in multinational corporations

However, the use of generalized networks to solve multinational planning problems has been limited by the availability of approaches to facilitate the formulation of problems. Generally, in order to use the approaches of Brown and McBride [5] and McBride [17] requires user familiarity with the basic algorithms and substantial problem formulation work.

In order to use generalized network algorithms the user could be required to develop large numbers of constraints. However, few potential multinational corporate planners are operations research experts or have the time to generate such programs. Further, if any changes occurred then replanning would be required and the formulation would need to be changed.

One approach to circumvent some of these difficulties is to provide the multinational planner with the assistance of an operations research expert. However, this would not mitigate the time issue. Even an operations research expert could require substantial time to formulate problems for generalized network algorithms. Further, there would need to be substantial feedback from the planner to ensure the model captured the reality being modeled.

As a result, we can anticipate that if a multinational corporate planner was to employ a generalized network approach, then an alternative vehicle would be required to facilitate formulation, feasibility testing and solution generation. GNS is proposed as a tool to rapidly generate generalized network models of complex integrated multinational corporate structures and international currency situations.

4. GNS's knowledge of network problems

GNS has substantial knowledge about generalized network flow problems. By understanding the nature of generalized network flow problems, GNS is able to facilitate the planning process by anticipating what the user wants to do and what the user can do within the context of generalized network flow problems.

GNS solicits information from the user directly, using no mathematical formulations. GNS generates a generalized network, tests feasibility, solves that generalized network and provides the user with some interpretation of that network, using sensitivity analysis and dual variables.

4.1. Structuring of the problem by GNS

GNS requires only minimal information from the user to construct a network representation of their planning problem. GNS assumes that the firm will group its different types of operations and different locations in different node sets or groups. As a result, GNS needs to know the number of sets that will be used. For example, one node set could represent the set of currencies that the firm generates, while another node set could represent the different production locations that ultimately convert product to cash and conversely.

GNS also needs to know how many nodes are in each set. Given this information and the number of node sets, GNS will automatically generate all the nodes and all the directional arc sets between all nodes in a given set. Arcs establish direct relationships between different locations and different types of operations, such as cash and production. Alternatively, the user has the option of developing and describing each node and arc individually.

Finally, GNS needs to know what the user wishes to call the different node and arc sets. In order to limit potential intimidation by the use of variables, such as " x_{ij} ", GNS allows the user to employ English language names for the variables. As a result, the system can employ names such as "cash" or "excess_cash_foreign". This helps the user formulate the problem because there is a direct tie to their world. It also helps the user understand the output since the variable labels directly correspond to real world objects.

4.2. Default parameters

Although GNS attempts to anticipate the user's needs, it is difficult to decide, a priori, default coefficient values. However, GNS does provide default values. For example, multipliers are set with a value of 1, costs are set with a value of 0, lower bounds are set at 0 and upper bounds are set at infinity (in the current model infinity equals 10,000). Each of these default parameters could be changed using another screen, not discussed here.

4.3. Formulation of problems

As noted by Iijima [14, p. 293], "automatic modeling is one of the key topics in model base management in DSS". GNS has substantial model formulation capabilities. GNS takes the information provided to it regarding the different subsidiaries and different types of operations, the direction of flows and parameter inputs. Using this information, the system formulates a network problem that is first analyzed for feasibility and then solved. The user does not need to formulate the problem, choose a solver or analyze its feasibility, the system does all the necessary formulation for them.

4.4. Need for feasibility knowledge

Murphy and Stohr [18] have noted the importance of ensuring that the formulated model is feasible. Since the user is not an expert at formulating the model and since the user probably cannot understand the impact of all the interacting constraints, the models that they formulate may not be feasible. Further, users can make errors when constructing a network model. For example, users may develop node sets and not develop corresponding arcs, or may develop only partial arc sets. In addition, users may build erroneous parameters into the system, where lower bounds exceed upper bounds. As a result, GNS automatically does a number of feasibility tests. Those tests include the following:

- (1) Nodes with supply or demand but no incident arcs.
- (2) Nodes with exiting arc capacity less than supply.
- (3) Nodes with entering arc capacity less than demand.

- (4) Nodes with arcs only entering, but no demand. (This could result if the user did not create a connecting arc set. This may not cause an infeasibility, so users are only warned and asked if the system should continue to the solver after the remaining checks are performed.)
- (5) Nodes with arcs only exiting, but no supply. This is similar to the previous check. If the system finds errors, then it provides user-friendly messages to the user indicating the problem nodes and arcs, and what is wrong.

4.5. *Solving and interpreting knowledge*

GNS takes the resulting, apparently feasible network and solves it using the algorithm in Brown and McBride [5]. The solution is presented to the user. In addition, GNS presents some interpretation of the solution. In particular, GNS provides sensitivity analysis and the dual variables.

4.6. *Using previous cases*

GNS allows the user to make use of previous cases and templates that have been created. This allows the generation and capture of a general corporate model that can be used over and over. Alternatively, generic templates, such as node sets for cash and node sets for production can be generated (as in the following section) and reused.

Previous cases allow the user to perform a type of case-based reasoning. Users start with previous cases and reason about the situation facing them. The system facilitates this process by providing the ability for the user to employ templates and previous examples.

5. **Use of the GNS**

GNS is illustrated using an example problem. Selected screens from the system are presented to indicate how the system would be used to solve the example problem.

5.1. *Example problem*

In order to illustrate GNS consistent with the literature on international financial planning, the example problem presented in Crum et al. [7] will be solved. In that problem, there is a single parent and a subsidiary. There is also a product that is converted into cash. Further, foreign currency can be converted into United States (U.S.) currency. The example is summarized in figure 1, where nodes i , g , and k illustrate currency, and nodes j and h illustrate product. Nodes i and j are at the subsidiary, and nodes g , h , and k are at the parent.

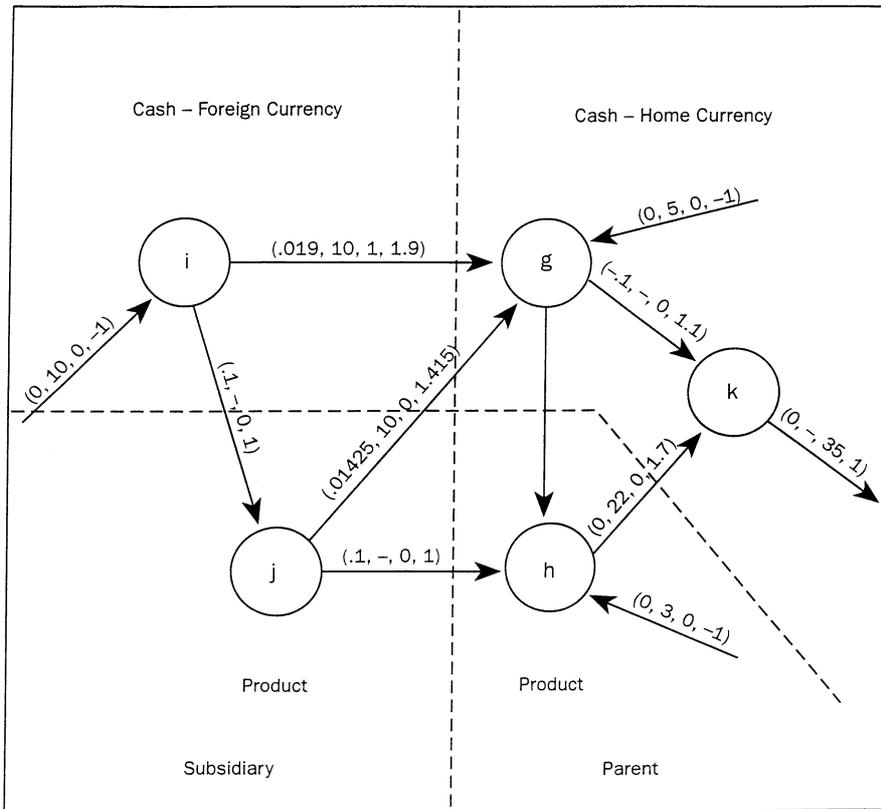


Figure 1. Multinational planning example. All costs in terms of home currency. Legend (cost, upper bound, lower bound, multiplier). When - is used for the upper bound, the arc flow is not bounded. Source: Crum et al. [7, p. 181].

Associated with each arc (i, j) is a 4-tuple $(c_{ij}, U_{ij}, L_{ij}, p_{ij})$. c_{ij} is used to represent the cost of a unit of flow. For example, if one unit of product is sent from node j to node h , the cost is (0.1).

U_{jh} is used to capture the upper bounds of the amount of those flows, and L_{jh} is used to capture the amount of the lower bound. The term $U_{ij} = "-"$ is used to indicate that there is no upper bound.

The arc multiplier (p_{ig}) is used to capture the conversion process. For example, if three units of foreign currency are shipped from the subsidiary to the parent ($i \rightarrow g$), then 5.7 units of U.S. currency arrive at node g . The multiplier represents the net exchange rate.

There is a supply of cash (10 units) in the foreign currency at the subsidiary, node i . There is also a supply of cash (5 units) in U.S. currency, at node g . There is a supply of product at node h . Node j has neither an excess nor deficit inventory situation. Node k has a demand for up to 35 units of U.S. currency. Thus, there are

supplies and demands for multiple commodities, and allocation decisions must be made.

The arcs indicate the feasible ways that cash or product can be transferred from one node to another node. If there is no arc, then that indicates that there is no direct transfer between the nodes. Since there is an arc from node i to node g , this means that foreign currency can be translated into U.S. currency. Similarly, an arc from node j to node h means that materials can be transferred from subsidiary to parent.

There are three paths through which foreign cash can be translated into product in the U.S. First, foreign currency can be used to generate product in the foreign country, which is then shipped to the parent ($i \rightarrow j \rightarrow h$). Second, foreign currency can be used to generate product in the foreign currency, which is then sold in the U.S. and then transferred to product in the U.S. ($i \rightarrow j \rightarrow g \rightarrow h$). Third, foreign currency can be transferred to the U.S. and then transformed into product ($i \rightarrow g \rightarrow h$).

5.2. Solving the problem

Solving the example is facilitated using GNS. This section will summarize the solution process used to solve the example in figure 1.

First, the user employs an existing model or template via the “file” command or the user must generate the node sets (figure 2) using the “create” command. For illustration purposes, it is assumed that the user will “create” the model, so in this situation there is no case-based reasoning from the system. In this example, there are two sets, “cash” and “product”. In its current configuration, the system allows up to 20 different node sets, but that can be changed using another screen. The names i , g and k are given to maintain consistency with the example in Crum et al. [7]; however, the system will generate names if the user wishes to default.

Second, the arc sets must be generated using the “create” command. The system generates arcs between each subset of nodes that the user specifies so that the user’s work is minimized. In addition, the system allows the user to name the subsets of arcs. In figure 3, the subset “make_product_home” is being developed. Those arcs consist of all the pairs cash (2–2), that is, node g , and product (2–2), that is, node h . Thus, in this case the arc set consists of the arc from node g to node h , where the product is made in the U.S. The system is currently configured for twenty such arc sets, but that bound can be easily changed using another screen.

Third, data contained in the tuple $(c_{ij}, U_{ij}, L_{ij}, p_{ij})$ must be entered into GNS. This is illustrated in figure 4. Once the data is entered then, using the same “data” commands, the data can be edited. Alternatively, the system can draw the data from an existing database.

Fourth, the system checks for feasibility and then solves the problem using the “solve” command. The solution provides the flow and reduced costs on those arcs, as seen in figure 5. In addition, as seen in figure 6, additional solution information is provided.

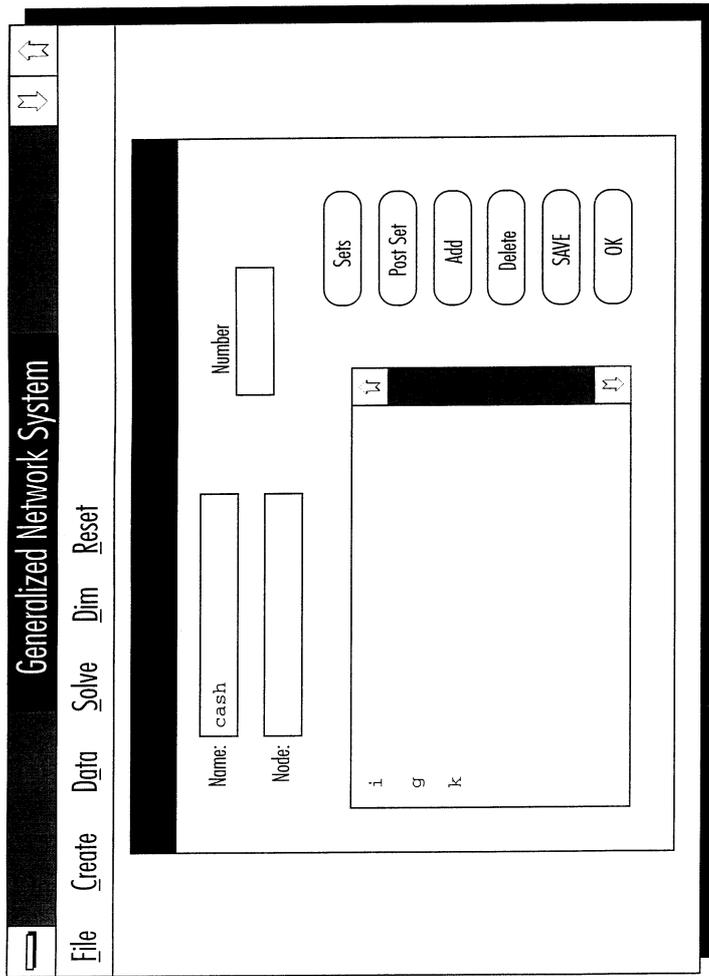


Figure 2.

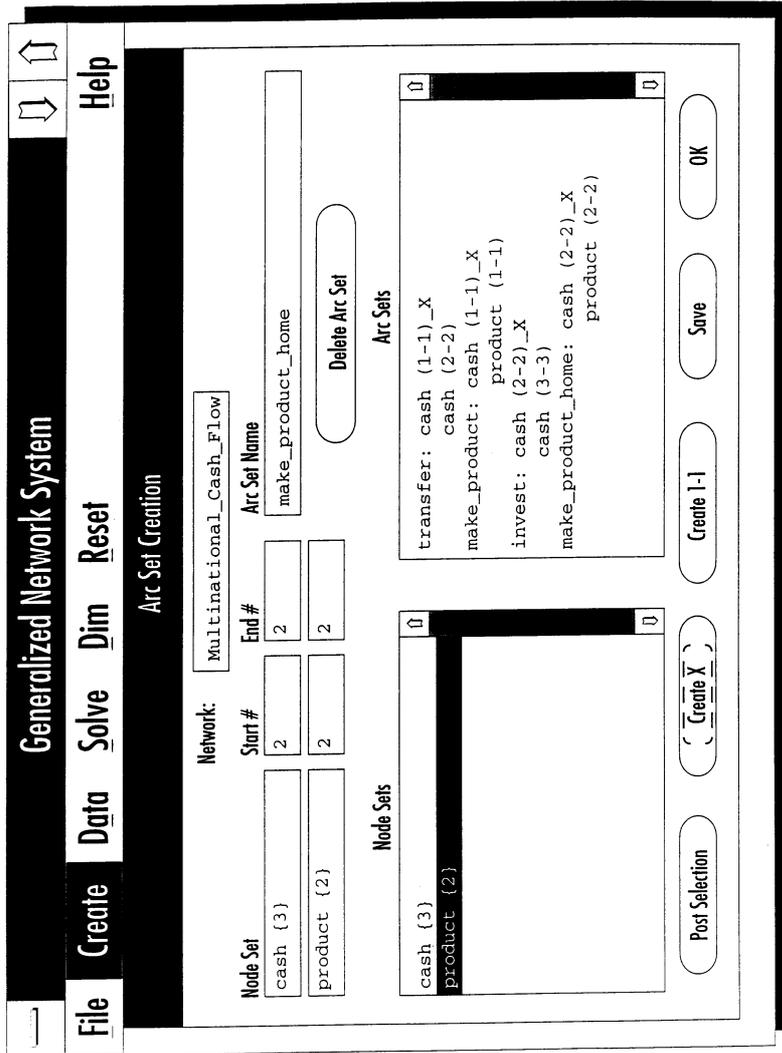


Figure 3.

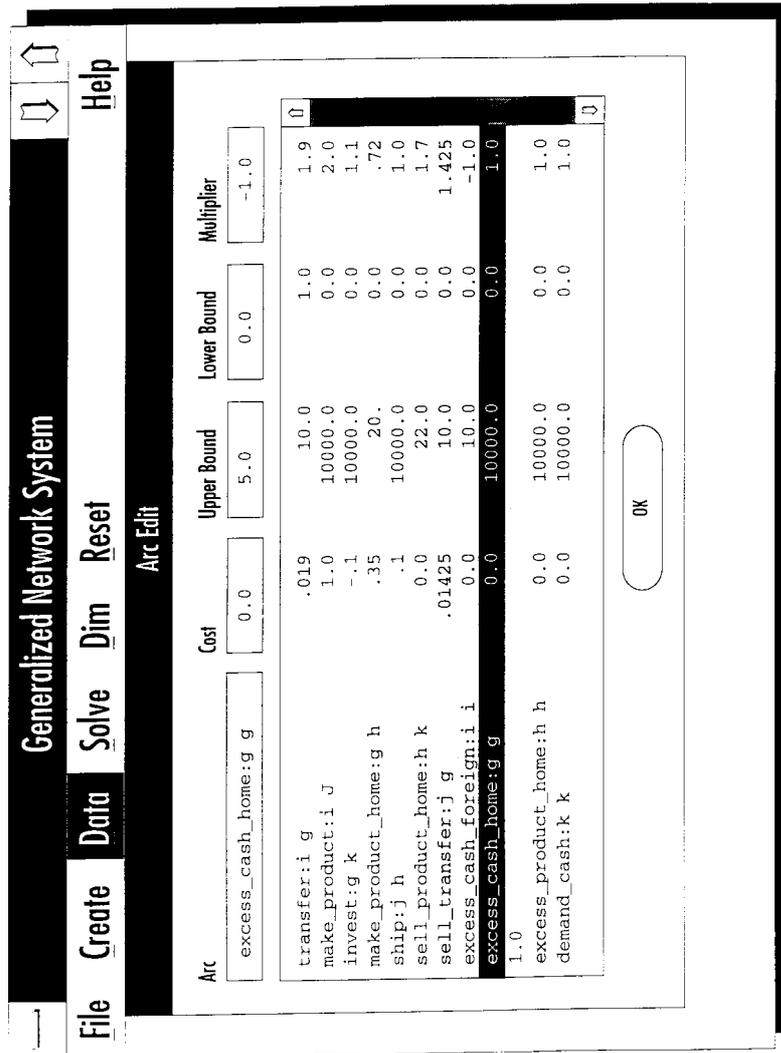


Figure 4.

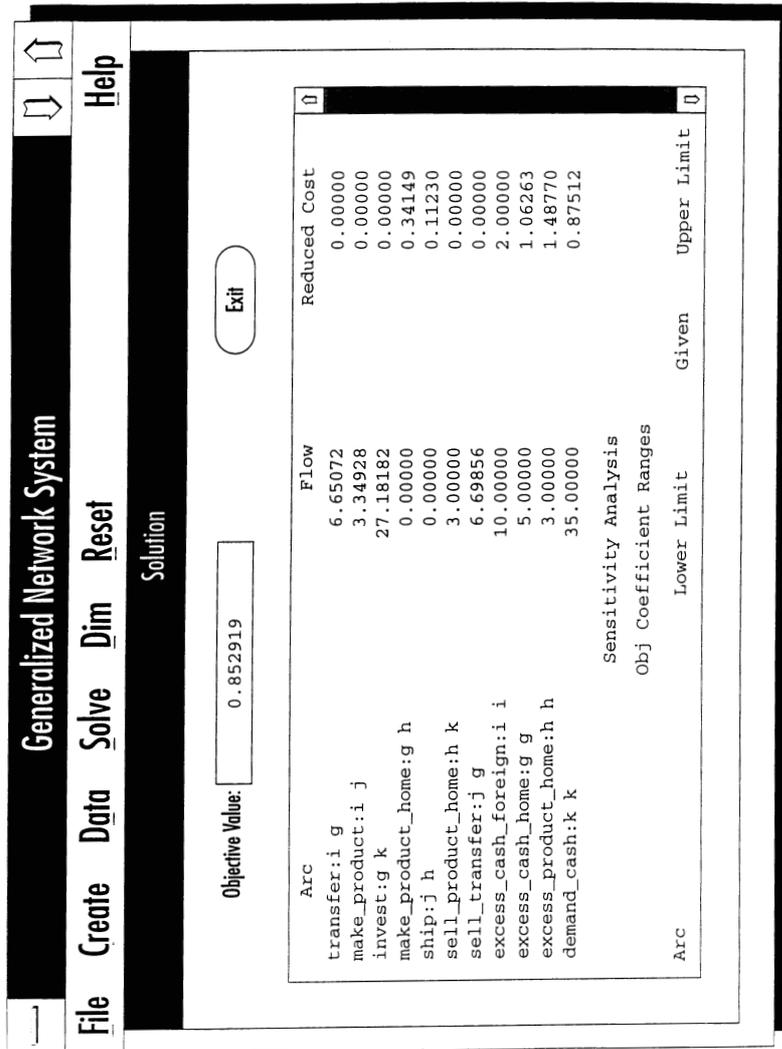


Figure 5.

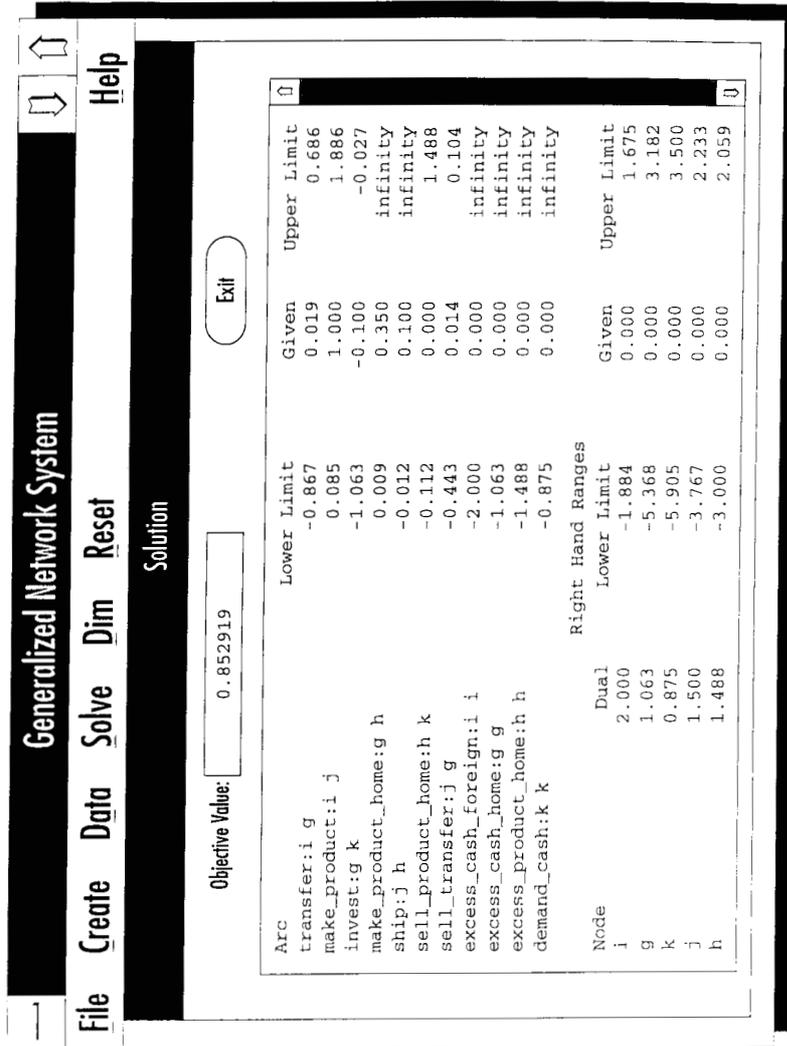


Figure 6.

5.3. *Use of GNS in multinational corporate settings*

GNS could be used in a variety of corporate settings. In the case of multinational planning, generally a model would be developed with substantial care to capture the parent and each of the subsidiaries. In some cases, the subsidiaries also could be U.S. firms. The model could then be used to study issues such as interstate transfers and accordingly account for different state tax rates.

In some cases, users will make use of previously established cases, stored as templates, that capture particular settings of interest to the planner, e.g., potential acquisitions or financial instruments. Previous cases can be reused and integrated with other cases to facilitate planning. In other cases, the structural model would be stable, but the parameters would be subject to fluctuation. As a result, as parameters change, such as exchange rates, the model could be quickly up-dated and resolved in order to facilitate replanning.

5.4. *Computing environment*

GNS was developed for a personal computer environment. The system was developed for a Microsoft "Windows" environment. GNS provides a user-friendly, mouse-based environment for supporting multinational planners.

6. **Summary and extensions**

This section provides a brief summary of the paper and discusses some extensions.

6.1. *Summary*

This paper has presented an intelligent system that can be used to assist in the planning process by facilitating the use of a generalized network in a user-friendly manner. GNS uses substantial knowledge of generalized network flow problems to facilitate processes of formulating, solving and presenting planning problems. Using node and arc sets, the system provides an intelligent way to easily generate problems. Variables are named so that the user knows what they represent. Using a range of feasibility checks, GNS ensures the problem is feasible before it uses the Brown and McBride [5] algorithm to solve the problem. Finally, the system solves and presents the solution for the user.

Multinational planners face the problem of integrating production and cash generation from multiple subsidiaries. Those problems can be couched as generalized network flow problems. GNS guides the user through the process of formulating and solving complex systems in a timely manner. GNS permits the user to reuse templates representing different organizational structures.

6.2. Extensions

There are a number of extensions. First, the results can be extended to other financial planning situations. For example, Crum et al. [8] develop a network model for the analysis of integrated working capital management. That problem considers balancing activities from three relatively separate activities, marketing, production and finance. Integration of those activities requires simultaneous consideration of intertemporal dynamics, such as short-run and long-run implications, and cross sectional relationships between the functional areas. Crum et al. [8] use networks to model temporal and cross sectional relationships.

Second, the current model could be used to study a wide range of problems in more detail. For example, different states and countries have different tax rates. The impact of subsidiaries incurring those different rates could be studied.

Third, the current model uses a basic generalized network model. However, this could be extended to more general forms, such as generalized network models with side constraints. In that case, the algorithm of McBride [17] could be used to solve the problem.

References

- [1] B. Back, *An Expert System for Financial Statements Planning*, Abo Academy Press, Abo, Finland, 1991.
- [2] B. Back, Assisting inexperienced accountants in developing financial statements, *International Journal of Intelligent Systems in Accounting, Finance and Management* 1(1992)155–162.
- [3] B. Back and R. Back, Financial planning in the presence of tax constraints, *Expert Systems with Applications*, Unpublished Paper, 1992.
- [4] M. Binbasioglu and M. Jarke, Domain specific DSS tools for knowledge-based model building, *Decision Support Systems* 2(1986)213–223.
- [5] G. Brown and R. McBride, Solving generalized networks, *Management Science* 30 (December 1984).
- [6] R. Crum, Cash allocation in the multinational firm: A constrained generalized network approach, Unpublished Ph.D. Dissertation, University of Texas, 1974.
- [7] R. Crum, D. Klingman and L. Tavis, Strategic management of multinational companies: Network-based planning systems, *Applications of Management Science* 3(1983)177–201.
- [8] R. Crum, D. Klingman and L. Tavis, An operational approach to integrated working capital planning, *Journal of Economics and Business* 35(1983)343–378.
- [9] M. Dempster and A. Ireland, Object oriented model integration in MIDAS, *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences*, R. Blanning and D. King, eds., 1989.
- [10] M. Dempster and A. Ireland, A financial expert decision support system, in: *Mathematical Models for Decision Support*, NATO ASI Series, Vol. F48, G. Mitra, ed., Springer, Berlin, 1988.
- [11] F. Fabozzi and J. Valente, Mathematical programming in American companies: A sample survey, *Interfaces* 7(1976).
- [12] H. Greenberg, A functional description of ANALYZE: A computer assisted analysis system for linear programming models, *ACM Transactions on Mathematical Software* 9(1983)18–56.
- [13] H. Greenberg, A natural language discourse model to explain linear programming models and solutions, *Decision Support Systems* 3(1987)333–342.

- [14] J. Iijima, Automatic model building and solving for optimization problems, *Decision Support Systems* 18(1996)293–300.
- [15] J. Lee and H. Lee, Interaction of strategic planning and short term planning: An intelligent DSS by the post model analysis approach, *Decision Support Systems* 3(1987)141–154.
- [16] P. Ma, F. Murphy and E. Stohr, Representing knowledge about linear programming formulations, *Annals of Operations Research* 21(1989)149–172.
- [17] R. McBride, Solving embedded generalized network problems, *European Journal of Operational Research* 21(1985)82–92.
- [18] F. Murphy and E. Stohr, An intelligent system for formulating linear programs, *Decision Support Systems* 2(1986)39–47.
- [19] K. Schittkowski, EMP: A software system for mathematical programming, Unpublished Paper, Mathematisches Institut, Universität Bayreuth, Bayreuth, Germany, 1985.
- [20] Y. Takahara, N. Shiba and H. Tanaka, An implementation of unified programming on actDSS, *Decision Support Systems* 18(1996)273–292.
- [21] P. Winston, *Artificial Intelligence*, Addison-Wesley. Reading, MA, 1984.