A Generalized Network Modeling System for Scheduling

Richard D. McBride

Daniel E. O'Leary

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

Abstract

NETSYS is an intelligent system for creating and solving generalized network models with substantial knowledge of generalized networks. It uses that knowledge to create networks and to investigate their feasibility.

This system uses a very powerful and easy-to-use user interface. A user of NETSYS would first create node sets and then arc sets by taking cross products and using one-to-one mappings with subsets of those sets. Default data, associated with the nodes and arcs at the time of creation, can be easily edited before problem solution. It is not required for the user to learn the syntax of a modeling language before this system can be used.

1. Introduction

NETSYS is a computer-based system for formulating, solving and interpreting generalized network problems. NETSYS functions as an intelligent assistant that operates between a generalized network algorithm and the user. NETSYS is an intelligent system that has substantial knowledge about generalized network problems and scheduling.

Without a system like NETSYS, at one extreme the user would have to create a large set of constraints. This could be a very time and resource consuming task. However, by anticipating a network structure and exploiting that network structure, NETSYS greatly facilitates the building of models.

Alternatively, the user could employ a general purpose modeling system. However, this would require that the user understand and know how to use the language. Unlike such general purpose modeling systems, NETSYS is very easy to learn and operate.

1.1 Generalized Network Problems

Many realistic financial problems can be formulated as network problems. For example, McBride et al. [6] developed a model for cash management.

Included in this class of network models are transportation, shortest path, assignment and transshipment models. Pure network models are generalized network models with arc multipliers restricted to having a value of one. When multipliers are permitted to take a value different than one, the class of models is greatly enlarged. When multipliers are viewed as simply modifying the flow through the network then generalized networks can be used to represent situations such as interest rates. When multipliers are interpreted as transforming one good to another then it is possible to model such processes as production. The generality of this class of problems is very helpful since a number of very efficient algorithms are available to solve generalized network problems (e.g., Brown and McBride [3]).

1.2 General Purpose Modeling Systems

In order to solve mathematical programming problems, general purpose modeling systems have been developed, including GAMS (Bishop and Meeraus [2]), Magic (Williams [7]), XML (Fourer [4]), Best-Plan [1] and NETWORLD (Geoffrion [5]). Associated with each of these systems is a modeling language. Unfortunately, as with all such languages, they are difficult to learn and remember. Most modeling languages have well-developed syntaxes that must be learned before the language can be used. This is a serious potential drawback for any users.

1.3 NETSYS

On the other hand, NETSYS is an easy to use system that takes advantage of the special interface techniques of Microsoft Windows. This approach permits users to quickly build and solve generalized network models on micro computers. The richness of this interface allows it to replace the usual modeling language syntax. Network models are built by first creating node sets and then taking cross products and one-to-one mappings with subsets of the node sets to create arc sets. The structure of the network can then be built very quickly.

As the nodes and arcs are built, necessary data for the network is solicited from the user. After the structure and data are developed the model is presented to the solver. NETSYS does a number of feasibility checks to ensure that a feasible model is developed. When the model is feasible, the solution is presented to the user and written to the user's disk for future reference. A complete sensitivity analysis also is provided to the user.

Once a model is built it can be reused. NETSYS allows the user the capability of either using that model independently or as a subset in another model.

1.4 This Paper

This paper proceeds as follows. Section 1 has provided an introduction to the paper. Section 2 discusses NETSYS's knowledge of generalized network problems and how that knowledge helps it formulate, solve and assist in the interpretation of the network problem. Section 3 briefly summarizes an example that is used to demonstrate NETSYS throughout the remainder of the paper. Section 4 outlines the basic structure of NETSYS. Each element of that structure is discussed in more detail in sections 5 through 9. Section 5 demonstrates how NETSYS can assist the user in the formulation of the nodes and arcs that constitute the network characterization of problems. Section 6 discusses the data editing capabilities of NETSYS. Section 7 summarizes the model solution and infeasibility checks. Sections 8 and 9 discuss other details of NETSYS's operation. Section 10 analyzes some potential extensions of NETSYS. Section 11 provides a brief summary of the paper.

2. NETSYS's Knowledge of Network Problems

NETSYS has substantial amounts of knowledge about network flow problems. NETSYS uses that knowledge to anticipate what the user wants in a generalized network model. Thus, NETSYS has some intelligence that we would expect of an operations research expert.

NETSYS assumes that nodes can be grouped into node sets, where each of the nodes in those sets is similar, for example, sources or destinations. Once NETSYS knows what the basic groups are, it can rapidly elicit all the nodes, requiring only that the user specify the group name and the number of nodes in the group. The expert also could develop the nodes individually.

NETSYS also allows the user to rapidly define the arcs between the node sets or subsets of the node sets. The arcs that NETSYS defines are directional. NETSYS allows the definition of the arcs with little more than the click of a mouse. Default parameters are provided, e.g. multipliers are set with a value of one.

NETSYS uses its knowledge of generalized network problems to test feasibility before solutions. Once those tests have been done, NETSYS solves the problem and does a sensitivity analysis.

3. Example Problem

A bicycle manufacturing company has two assembly plants and four dealerships located in a particular state. The company wants to determine which plants should assemble the bicycles for each dealer in order to maximize overall profit for the coming summer. The following information has been obtained:

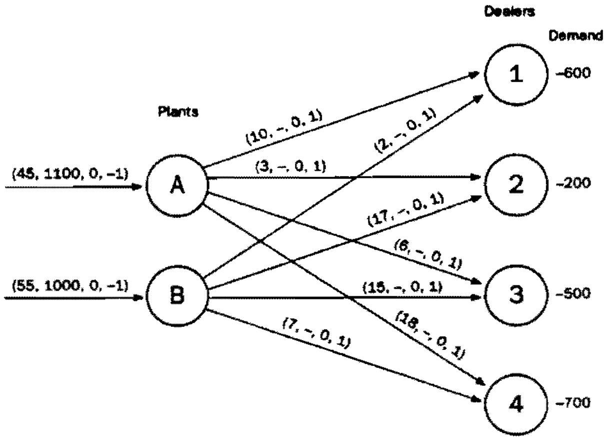
Assembly Plant	A	B
Capacity(supply)	1100	1000
Assembly Cost/Bicycle	45	55

Dealer	1	2	3	4
Requirements(demand)	600	200	500	700

Assembly Plant	Shipping Cost to Dealer			
	1	2	3	4
A	10	3	6	10
B	2	17	15	7

The problem facing the user is to determine the optimal assembly and distribution of the bicycles. Since plant capacity is greater than demand, this problem can be modelled as a transportation problem with singleton arcs entering the two plant nodes representing the indefinite supply at each plant, illustrated in figure 1.

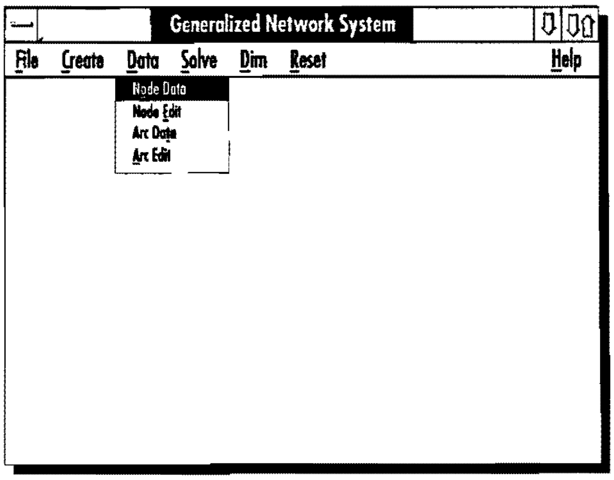
Figure 1



4. System Menu Structure

There are seven basic menus/options in NETSYS. Figure 2 shows the system menu bar with the 'Data' menu dropped down. The user makes a menu selection by placing the cursor on the menu name and clicking the mouse button. The user then makes his choice by selecting the appropriate option.

Figure 2



The 'File' menu has the following selections:

'New' - configure system to build a new model.
'Open' - allows a previously created model and its data to be read from a file.
'Save' - save model and its data for future use or editing.
'Save As...' - save current model in a new file.

When the user first enters NETSYS the system is configured to build a new model.

The 'Create' menu consists of the following selections:

'Node Sets' - create node sets.
'Networks' - create arc sets.

The 'Data' menu has the following selections:

'Node Data' - edit node values by node set.
'Node Edit' - edit node values with global edit.
'Arc Data' - edit arc values by arc set.
'Arc Edit' - edit arc values with global edit.

Selecting the 'Solve' option in the main menu bar causes the system to attempt solution of the current working model. The infeasibility checks described later in the paper are applied to the model to help the user debug the model. The solution and sensitivity analysis are written to a file for and displayed on the screen.

Selecting the 'Dim' option allows the user to change the dimensions of the arrays supporting NETSYS. In particular, the user can adjust the system dimensions to make better use of memory.

Selecting the 'Reset' option allows the user to reset some or all of the values associated with nodes and arcs back to the default values.

The 'Help' menu provides screens explaining how to use the menu selections for the 'Create', 'Data' and 'Dim' menus. A 'General' screen is also provided to help give an overall view of the system. The system is completely self contained. Reading the help screens allows a user to quickly understand how to use system.

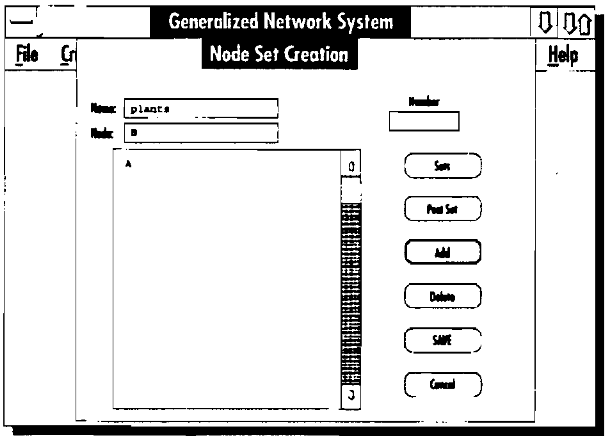
5. Model Creation ('Create')

The creation of models requires that the user specify node and arc information.

5.1 Node Set Creation

NETSYS has knowledge about the generic nature of network flow problems. It assumes that the user will employ node sets to represent groups of nodes with similar characteristics. In the example there are two such sets of nodes: plants and dealers (figure 3). In a general transshipment problem there would be three sets: sources, destinations and transhipment points.

Figure 3



Name: plants

Node: B

A

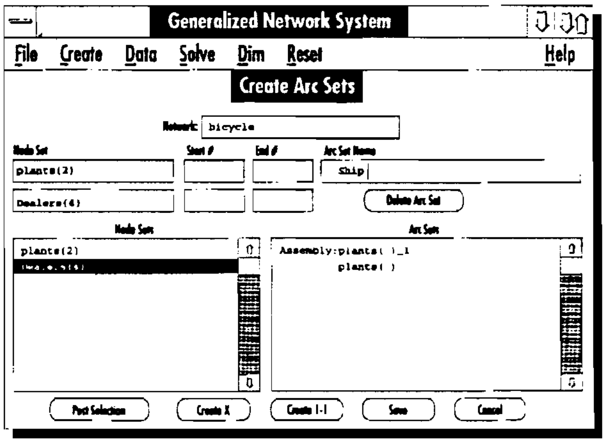
Once NETSYS knows the different node groups, it solicits information about individual nodes within the groups. Users either name the nodes individually or have NETSYS name them. In the example, the plants are named individually by the user (A and B), while the dealers have been named by the system (1, ..., 4), after the name of the node set.

5.2 Arc Set Creation

After the nodes are created, the arcs between them are developed (figure 4). NETSYS's knowledge of generalized networks includes two different ways of generating arcs: cross product and one-to-one relationships. In the example, the cross product is used to generate an arc from each plant to each dealer to represent the process of distribution. The one-to-one mapping is used to generate the singleton arcs at the plant nodes, used to represent the process of 'Assembly.' In addition, these operators can be used for sequentially adjacent subsets of the nodes in a given node set (e.g., for dealers 2 through 4).

NETSYS also allows the possibility of the user naming the arc sets. In figure 4 the set of arcs from the plants to the dealers is called 'ship.' This feature can facilitate model building, debugging and interpretation.

Figure 4



Network: bicycle

Node Set: plants(2)

Dealers(4)

Arc Set Name: Ship

Node Sets:
plants(2)
Dealers(4)

Arc Sets:
Assembly:plants( )_1
plants( )

6. Data Editing ('Data')

The value of a node determines whether there is supply (positive value) at the node or demand (negative value) at the node. Since initially NETSYS does not know whether the user wants a node set to have supply or demand (or what those values might be) the value of each node is set to zero.

The values on the arcs indicate the flow capacity and the value of the flow. NETSYS anticipates that the user will have a lower bound on the flows in the network of zero. In addition, NETSYS assumes that the existence of an upper bound that is arbitrarily set at 10,000. Further, NETSYS initially assumes an objective function coefficient of 0 and a multiplier value of 1.

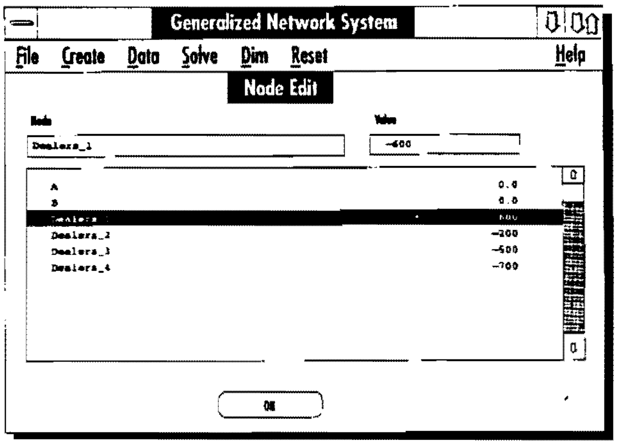
Since NETSYS assumes these parameters, it is necessary that the user be able to change these assumptions so that a reasonable model can be built. That process is accomplished via data editing.

6.1 Editing Node and Arc Data

Figure 5 illustrates the edit process for nodes. Using the example, since the demand for Dealer_1 is 600, the value is set at -600. The list of the parameters in figure 5 is a complete set for the example.

Associated with each arc are four values: Cost, Upper Bound, Lower Bound and multiplier. Figures 6 and 7 illustrate editing the parameters for the arcs. The list in figure 6 is complete for the example.

Figure 5

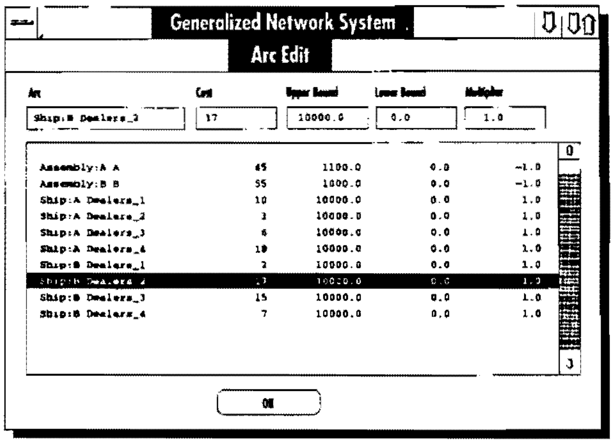


Node: Dealers_1

Value: -600

A	0.0
B	0.0
Dealers_1	600
Dealers_2	-200
Dealers_3	-500
Dealers_4	-700

Figure 6



Arc	Cost	Upper Bound	Lower Bound	Multiplier
Ship:B Dealers_2	17	10000.0	0.0	1.0

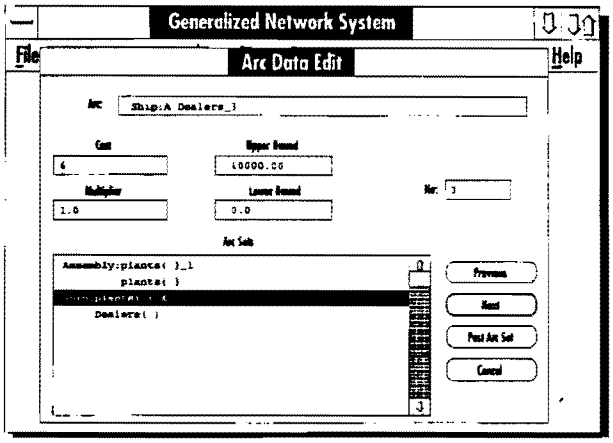
Assembly:A A	45	1100.0	0.0	-1.0
Assembly:B B	55	1000.0	0.0	-1.0
Ship:A Dealers_1	10	10000.0	0.0	1.0
Ship:A Dealers_2	3	10000.0	0.0	1.0
Ship:A Dealers_3	6	10000.0	0.0	1.0
Ship:A Dealers_4	18	10000.0	0.0	1.0
Ship:B Dealers_1	2	10000.0	0.0	1.0
Ship:B Dealers_2	17	10000.0	0.0	1.0
Ship:B Dealers_3	15	10000.0	0.0	1.0
Ship:B Dealers_4	7	10000.0	0.0	1.0

7. Model Solution and Infeasibility Checks ('Solve')

When the 'Solve' option on the menu bar is selected NETSYS builds a generalized network problem using the given structure and data. It then performs infeasibility checks to help the user debug the model and its data. All of these infeasibility checks cannot guarantee model feasibility. However, the checks can help the user to quickly find oversights in building the model and many structural problems.

Before the infeasibility checks are performed, all arcs with non-zero lower bounds are transformed to arcs with zero lower bounds. This is done by inducing a demand to the "tail" node of the arc equal to the lower bound and giving a supply to the "head" node of the arc equal to the lower bound. The upper bound is then reduced by an amount equal to the lower bound.

Figure 7



Arc: Ship:A Dealers_3

Cost: 6

Upper Bound: 10000.00

Multiplier: 1.0

Lower Bound: 0.0

Nr: 3

Arc Sets

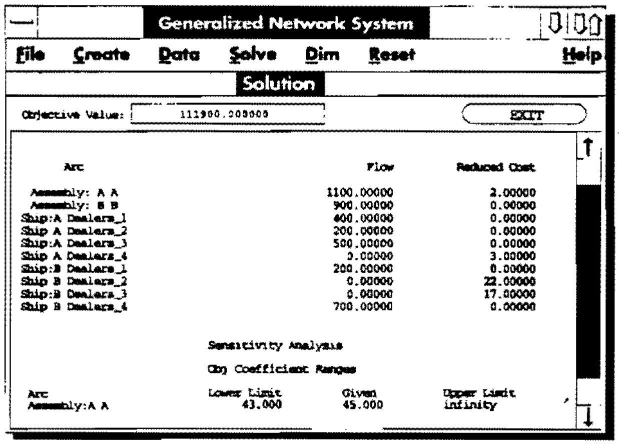
```
Assembly:plants( )_1
        plants( )
Ship:plants( ) ( )
      Dealers( )
```

The infeasibility checks exploit a knowledge of the structure of feasible network models. The six infeasibility checks are performed on the transformed problem by searching for the following conditions:

1. A node with supply or demand but no incident arcs;
2. A node with arcs only entering but no demand. This may not cause an infeasibility so is only warned and asked if the system should continue to the solver after all of the remaining checks are performed. However, this could result if the user failed to create a connecting arc set.
3. A node with arcs only exiting but no supply. This is similar to case 2.
4. A node with leaving arc capacity less than the supply at the node.
5. A node with entering arc capacity less than the demand at the node.
6. The supply does not equal the demand when the problem is a pure network problem (i.e. all multipliers equal one and there are no singleton arcs).

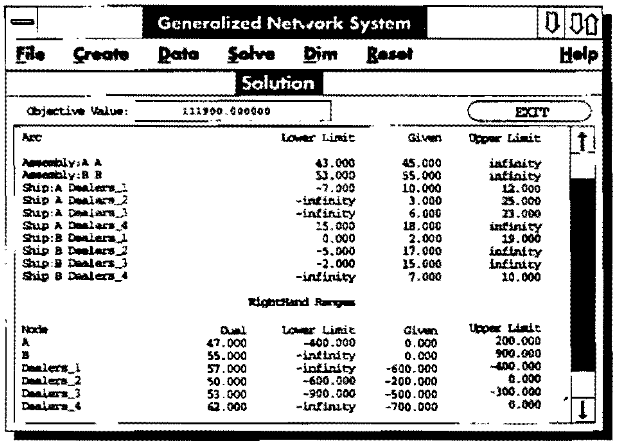
If the system passes all of the infeasibility checks, it attempts solution. When a feasible solution is obtained the solution and sensitivity analysis are written to the screen and to a disk file for future reference. The solution and sensitivity analysis for the example bicycle problem are shown in Figure 8 and 9.

Figure 8


Generalized Network System
File Create Data Solve Dim Reset Help
Solution
EXIT

```
Objective Value:   111900.000000

     Arc                                  Flow      Reduced Cost

  Assembly: A A                      1100.00000         2.00000
  Assembly: B B                       900.00000         0.00000
 Ship:A Dealers_1                     400.00000         0.00000
 Ship A Dealers_2                     200.00000         0.00000
 Ship:A Dealers_3                     500.00000         0.00000
 Ship A Dealers_4                       0.00000         3.00000
 Ship:B Dealers_1                     200.00000         0.00000
 Ship B Dealers_2                       0.00000        22.00000
 Ship:B Dealers_3                       0.00000        17.00000
 Ship B Dealers_4                     700.00000         0.00000

                      Sensitivity Analysis

                      Obj Coefficient Ranges

  Arc                 Lower Limit       Given        Upper Limit
  Assembly:A A            43.000        45.000       infinity
```

Figure 9


Generalized Network System
File Create Data Solve Dim Reset Help
Solution
EXIT

```
Objective Value:   111900.000000

Arc                                  Lower Limit      Given   Upper Limit

Assembly:A A                              43.000     45.000      infinity
Assembly:B B                              53.000     55.000      infinity
Ship:A Dealers_1                          -7.000     10.000        12.000
Ship A Dealers_2                       -infinity      3.000        25.000
Ship:A Dealers_3                       -infinity      6.000        23.000
Ship A Dealers_4                          15.000     18.000      infinity
Ship:B Dealers_1                           0.000      2.000        19.000
Ship B Dealers_2                          -5.000     17.000      infinity
Ship:B Dealers_3                          -2.000     15.000      infinity
Ship B Dealers_4                       -infinity      7.000        10.000

                                 RightHand Ranges

Node                      Dual       Lower Limit      Given   Upper Limit
A                       47.000         -400.000      0.000       200.000
B                       55.000        -infinity      0.000       900.000
Dealers_1               57.000        -infinity   -600.000      -400.000
Dealers_2               50.000         -600.000   -200.000         0.000
Dealers_3               53.000         -900.000   -500.000      -300.000
Dealers_4               62.000        -infinity   -700.000         0.000
```

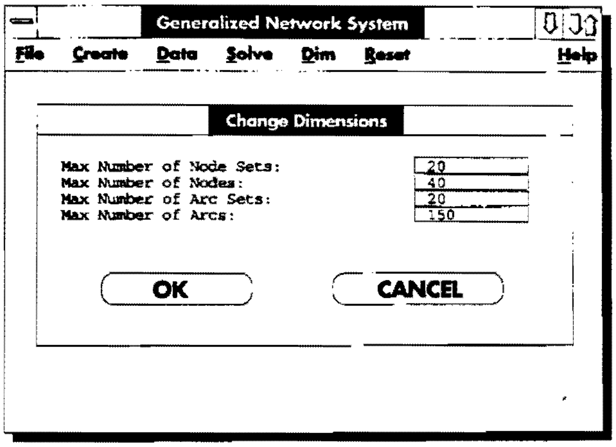
8. Changing Size of Supporting Arrays ('Dim')

This window allows the user to change the size of the supporting arrays in NETSYS. This permits adjusting memory requirements. When memory is not a problem it is not necessary to change the dimensions. Instead, the dimensions would be set large enough for the largest problem to be solved.

The 'Change Dimensions' screen is shown in Figure 10. It indicates that the user need only work with the following four parameters:

Max Number of Node Sets - This is the maximum number of node sets that can be created.
Max Number of Nodes - When the number of nodes in each set are added together this sum cannot exceed this value.
Max Number of Arc Sets - This is the maximum number of arc sets that can be created.
Max Number of Arcs - The total number of arcs in all arc sets cannot exceed this value.

Figure 10


Generalized Network System
File Create Data Solve Dim Reset Help
Change Dimensions
OK
CANCEL

```
Max Number of Node Sets:    20
Max Number of Nodes:        40
Max Number of Arc Sets:     20
Max Number of Arcs:         150
```

Once a model is created with its data, it can be saved and used as a building block for another model. The node and arc set editing features built into the system facilitate this. The 'Reset Data to Default Values' screen shown in Figure 11 permits the user to set any or all of the model data back to its default values.

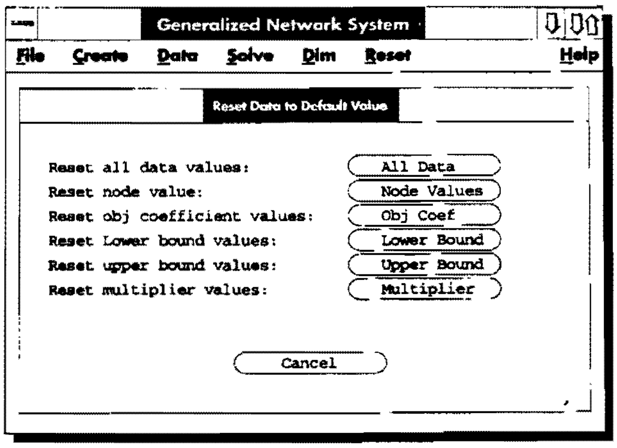
9. Adjusting Model Coefficient Value ('Reset')

NETSYS's anticipations can be easily changed to meet the needs of the user. The assumed coefficients can be altered by using the 'Reset' option. If the user wants to have a common parameter that is different than NETSYS's original expectations, then this option can be used in the initial development of the model. Alternatively, the user can revise the coefficients of an existing model. This option is illustrated in figure 11.

10. Extensions

There are a number of extensions that can be made to the basic NETSYS model. First, since the system described in this paper is very general, it can benefit from the specificity of a specific situation, e.g., a specific firm or decision making problem. For example, for a given decision situation, the node sets and the corresponding arcs might be stable so that the user only needs to make slight perturbations in the model. Further, the coefficients for a given decision problem may be stable, for example, upper and lower bounds may be well-established. In these situations, those node sets, arc sets and values can be integrated into the system as NETSYS's expectations.

Figure 11



Second, additional domain intelligence could be embedded into the system. For example, CASH MANAGER (McBride et al. [6]) included substantial cash management expertise. This would take the form of being able to anticipate more about the models being formulated. For example, if we assume a transhipment problem then we know that the values associated with the transhipment nodes will be 0. In addition, in the case of models for problems like the current example where one of the node sets produce and then export to the other nodes, then the multiplier will be -1 for that node set. Further, a library of subnetworks used for specific problems could be generated and stored for further use.

Third, the system could be extended to more general network problems. For example, the system could be extended to include side constraints or variables.

11. Summary

NETSYS is a user friendly, knowledge-based assistant for the solution of generalized network problems. By anticipating the user's needs, NETSYS solicits only minimal amounts of information. The system employs substantial amounts of knowledge about generalized network problems and their formulation, solution and interpretation. Feasibility analysis is employed to ensure that the user develops a model that can be solved. This is in contrast to many other 'intelligent' systems in the literature which assume that feasibility is not an issue.

NETSYS uses a Windows interface substitutes for a modeling language. Users work with visual displays on the screen and use a mouse to manipulate those screens. Accordingly, users do not get bogged down in learning another language, that must be understood and remembered in order to solve problems.

References

[1] Best-Plan profit optimization system, Rosetta Systems, 17101 Columbia Drive, Castro Valley, CA 94552.

[2] Bishop, J. and A. Meeraus, (1982) "On The Development of a General Algebraic Modeling system in A Strategic Planning Environment," Math. Prog. Study 20, pp 1-29.

[3] Brown G. and R. McBride, (1984) "Solving Generalized Networks," Management Science, Volume 30, No. 12.

[4] Fourer, R., (1983) "Modeling Languages Versus Matrix Generators for Linear Programming," ACM Transactions on Math. Software 9 (2), pp 143-183.

[5] Geoffrion, A. M., (1988) "The Concept of A Solver Shell: Gennet as an Example," Informal Note, AGSM, UCLA.

[6] McBride, R., D. O'Leary and G. Widmeyer, (1989) "A Knowledge-Based System for Cash Management Using Management Science Expertise," Annals of Operations Research, Volume 21, pp. 301-316.

[7] Williams, H. P., (1985) "Model Building in Linear and Integer Programming," in Computational Mathematical Programming by K. Schittkowski, Nato ASI Series, Vol F15, Springer, Berlin.