INTERMODAL COST ANALYSIS SOFTWARE USER'S MANUAL

MBTC FR-1100-2

Bonnie S. Boardman and Eric M. Malstrom

The work in MBTC project 1100 is documented in three parts:

1100-1 Final Report
1100-2 Reference [2], Software User's Manual
1100-3 Reference [3], Cost Analysis Tables

INTERMODAL COST ANALYSIS SOFTWARE
USER'S MANUAL*

Bonnie S. Boardman
Visiting Assistant Professor

Eric M. Malstrom
Professor and Head

Department of Industrial Engineering
University of Arkansas
Fayetteville, Arkansas 72701

***The development and distribution of the software described in this document was funded by grants from the Mack-Blackwell Transportation Center at the University of Arkansas.**

INTRODUCTION

This document presents installation procedures and operating instructions for intermodal transportation analysis software. The software enables the user to determine the least cost combination of transportation modes or the shortest throughput route between a given shipment origin and destination. The user may define both the origin and destination as well as the shipping network that is analyzed. The software accommodates truck, rail, barge and air transportation.

The software is described in more detail in two technical reports available from the Mack-Blackwell Transportation Center (MBTC) at the University of Arkansas. Interested readers are urged to obtain and consult references [2 and 20] which are available at no cost from the MBTC.

SOFTWARE OPERATION AND ARCHITECTURE

The main components of the developed Decision Support System (DSS) are the user interface, the database, and the k-shortest path algorithm. The DSS was designed so that through the interface the user can edit the database or run the k-shortest path algorithm using the appropriate network specifications. The components of the DSS and the flow of data between its components are illustrated in Figure 1. The individual components of the DSS are described separately in the sections that follow.

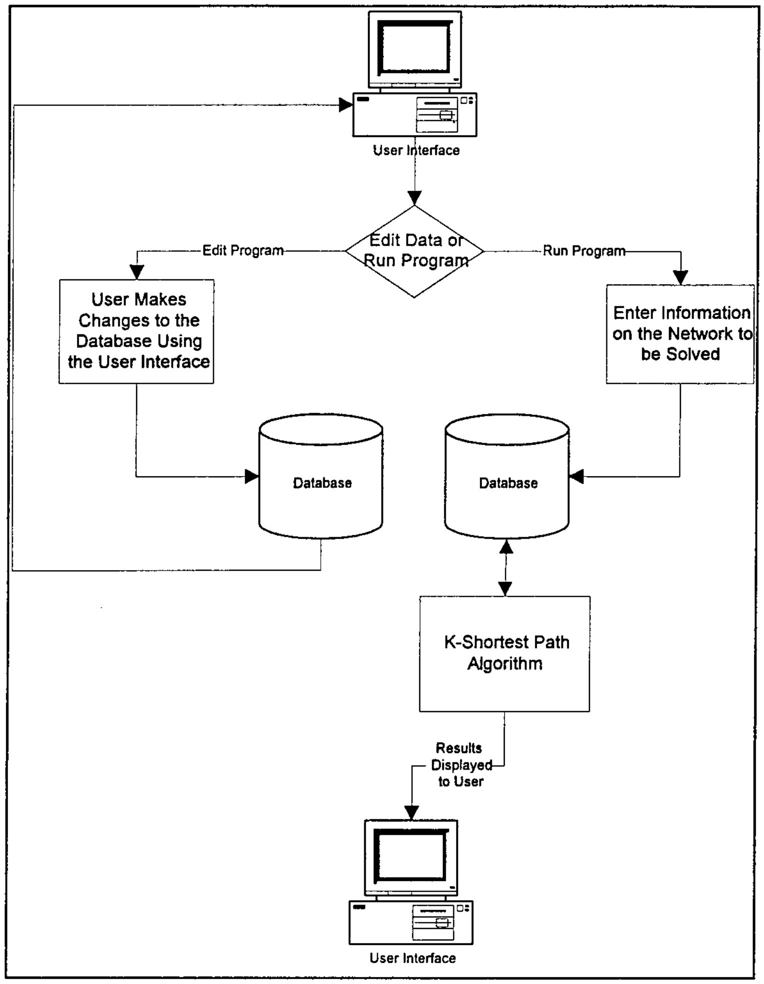

Figure 1: Data Flow for the DSS

K-SHORTEST PATH ALGORITHM

The Double-Sweep Method is used to solve the k-shortest paths through the network. Software to implement this network analysis tool was developed in Visual Basic. The architecture of the computerized version of the algorithm is described in the subsections that follow. A complete listing of the Visual Basic code can be found in the Appendix of reference [20].

Predecessor FORTRAN programs

Douglas R. Shier, who developed the Double-Sweep method [17], also wrote a FORTRAN V program that implemented the algorithm [18]. Philips and Garcia-Diaz [15] included the program, which was written in an updated version of FORTRAN in their textbook, along with a description of a slight code change which, excluded cyclic paths. A cyclic path is one in which a single node is visited more that once in a given path. For example, if the network did not exclude cyclic paths, a path might include visiting a given city twice. This is obviously unrealistic. Once a city has been visited, there is no reason to return to that city within in the same trip. Therefore, cyclic paths are excluded from those presented to the user as viable alternatives. The code provided in the literature sources cited in this section was the basis for the Visual Basic code written for this research.

For many reasons, the previously developed FORTRAN programs were not acceptable for use in the DSS. The original FORTRAN programs were both written in a time period corresponding to both card input and very limited computer storage capacities. The main difference between these predecessor programs and the described Visual Basic version is the increased size capabilities and the ease of interaction with the user. The developed Visual Basic algorithm has the ability to process networks with up to 1000 nodes and 5000 arcs. In the Visual Basic software, the user can specify the number of nodes, the number of arcs, the number of shortest paths that are calculated, the weight of the shipment, the source node, and the destination node. This can all be accomplished through the provided user interface at a fraction of the computational time previously required. There is no need for the user to have any programming knowledge to accomplish this network analysis procedure.

Another limitation of the FORTRAN programs is the fact that transfer costs and times could not be included. These were not available in either of the original programs. The FORTRAN programs assumed a constant length of each arc in the network. In the Visual Basic software, the user specifies the use of either transportation time or transportation cost as the arc length. The user may also change the times and/or costs associated with any particular arc length, again without needing any programming knowledge.

A description of the Visual Basic version of this algorithm along with the interaction of the code with the Access databases is presented in the subsections that follow.

Description of Visual Basic Software

The code for the Double-Sweep algorithm was written in a module that is used within a Visual Basic user interface. The module includes four subprograms or procedures, which interact with each other to complete the entire k-shortest path problem, including data input and output of the solution. Figures 2 and 3 illustrate the flow of data to and from the Visual Basic k-shortest paths module. Figure 2 shows the data flow within the module. Figure 3 shows the data flow between the components of the module and a database which has been constructed using Microsoft Access.

Module definition

Within Visual Basic, a module (or BAS file) is a collection of procedures and declarations that are stored together and recognized throughout an application. The module created for this application contains variable declarations and four general procedures. Because all of the code required to run the k-shortest path algorithm is stored in the module and is separate from the rest of the Visual Basic application, it can be inserted into any future Visual Basic programs without any reprogramming.

Procedure definition

A procedure is a section of code executed as a single unit that tells an application how to perform a specific task. Procedures are like subprograms. They divide complex applications into more manageable parts. For the k-shortest path algorithm needed for this application, four procedures consisting of about 350 lines of code were written which contain all of the code necessary to run the algorithm.

Procedure Descriptions

The first procedure in the module is the "subinput" procedure. This is the procedure that is called by the Visual Basic shell and starts the k-shortest path calculation procedure. The purpose of the "subinput" procedure is to read the network description into the format used by the algorithm. The number of nodes and arcs is read and then the distance of each arc is calculated. The arc distances are calculated based upon the information provided by the user, i. e., the origin, the destination, the weight and the objective, to either minimize time or minimize cost. All of the necessary data is read in from Access databases and then put into the appropriate arrays in the specified order.

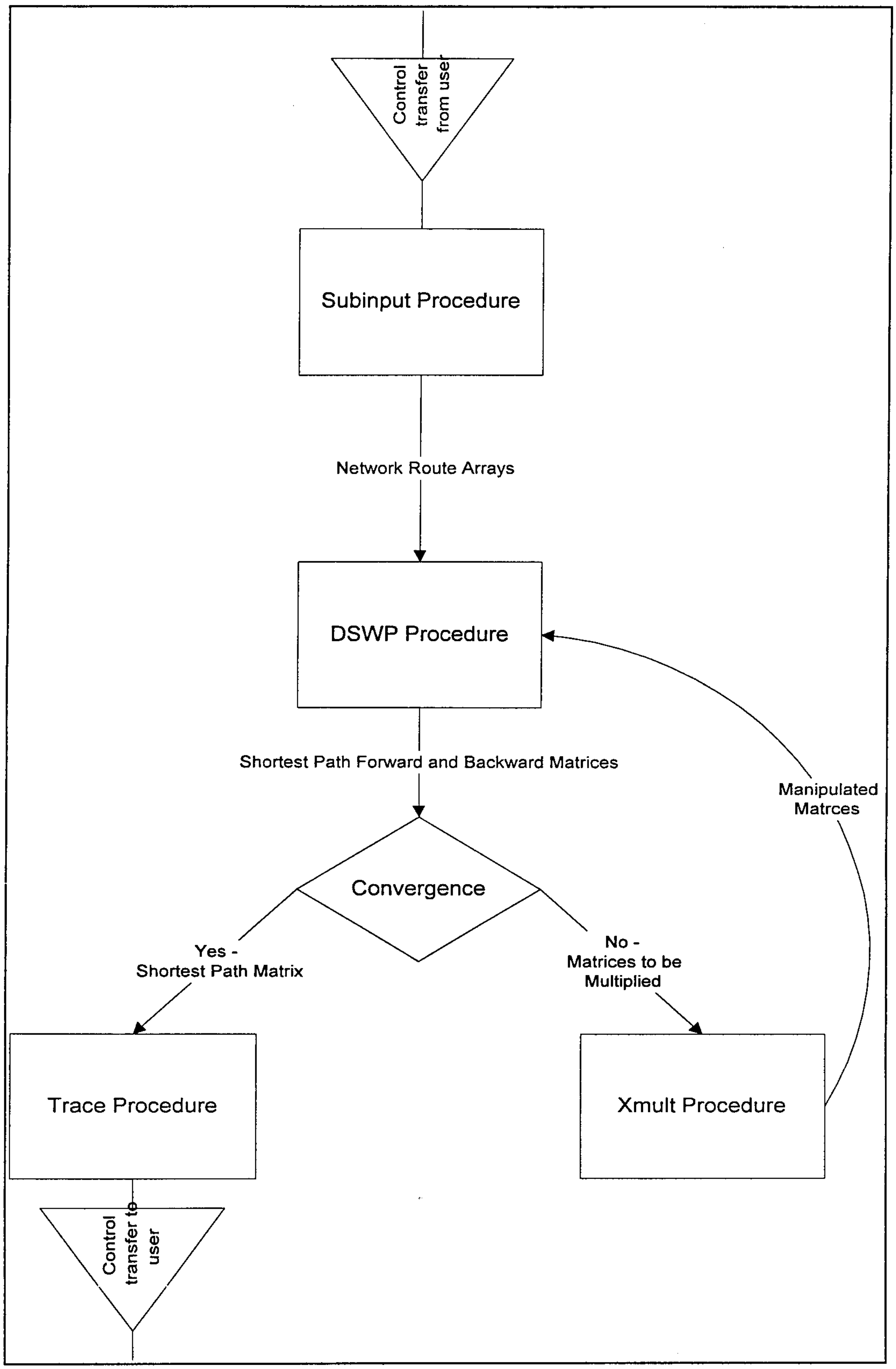

Figure 2: Data Flow Within the Visual Basic k-Shortest Paths Module

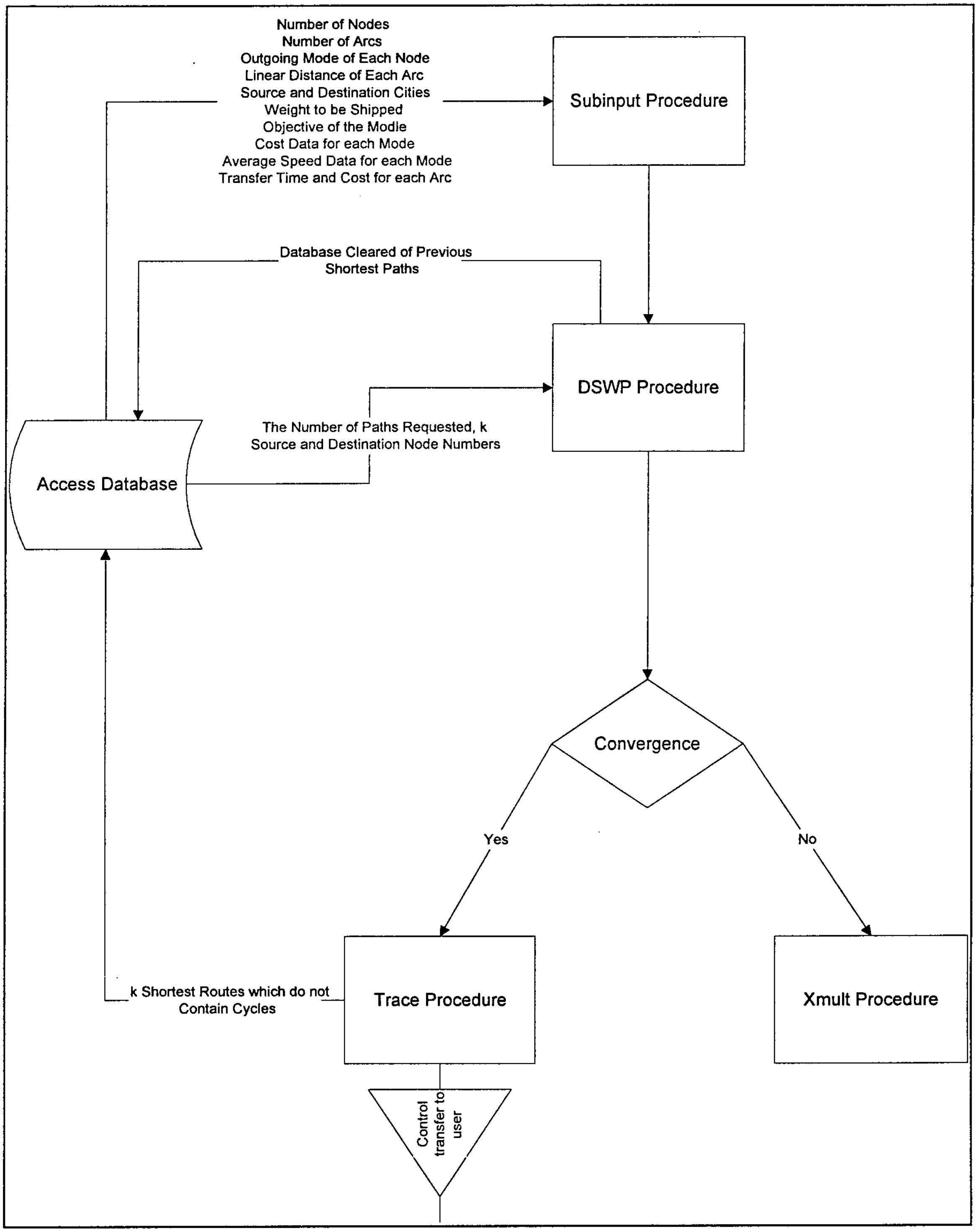

Figure 3: Data Flow Between Access Database and Visual Basic k-Shortest Paths Module

The double sweep ("DSWP") procedure is used to determine the k shortest paths. It is called by the "subinput" procedure once the needed variables and arrays have been created. The procedure calculates the k shortest path lengths (k is defined by the user), from the source node to all other nodes in the network. The procedure first forms the initial approximation matrix **X** and then modifies **X** through matrix manipulation. A forward sweep and a backward sweep are performed. The "DSWP" procedure then tests for convergence of the resulting matrices. Once convergence has been obtained, the "trace" procedure is called to identify the specific paths of the shortest routes identified in this procedure.

The "xmult" procedure takes values passed from the "DSWP" procedure and performs the necessary matrix manipulation and updates the **X** matrix. The procedure multiples the matrix **X** by the Ith column of the lower (or upper depending on the sweep being performed) portion of the arc length matrix. If improvements can be made to the **X** matrix, it is updated and the procedure passes a value to the "DSWP" procedure to let it know that an improvement has been made and convergence has not been met. Once the "xmult" procedure has been executed, it passes the updated variables and arrays back to the "DSWP" procedure and waits to be called again by "DSWP".

The "trace" procedure traces all the paths from the source node to the destination node corresponding to the shortest path lengths once convergence has occurred in the "DSWP" procedure. Each path is examined by starting at the destination node and working backwards, until the source node is reached. The "trace" procedure disregards all paths that contain cycles. Therefore, even though the "DSWP" might find 20 shortest paths from a given source, the "trace" procedure will only enumerate those distinct shortest paths which do not contain cycles.

Interaction with other procedures in the module

Figure 2 illustrates how the procedures within the k- Shortest Paths Module interact with one another. The Visual Basic shell calls the "subinput" procedure after the user has entered all of the starting information into the system. Once the procedure has been called, it reads in the appropriate data as described above and arranges the data into arrays that are used by the other procedures to calculate the k-shortest paths. After all of the information has been put into the appropriate arrays, the "DSWP" procedure is called to start the calculations.

The "DSWP" procedure calls the "xmult" procedure to perform the actual matrix manipulation in each sweep of the algorithm. The "DSWP" procedure sets the arrays and matrices up to be manipulated and then passes these values to "xmult". Once the "xmult" procedure has performed the matrix manipulation, the values are passed back to "DSWP" and execution continues. Once convergence has been obtained, the "DSWP" procedure calls the "trace" procedure to find the paths.

The "trace" procedure is called by the "DSWP" procedure once all of the requested shortest path lengths have been found. Once the trace procedure has written all possible distinct paths to the database, control is passed from the module procedures back to the user. The user once again controls of functioning of the program, by the buttons that he/she selects. The module is not called upon until the user selects another set of input values.

Module Interaction with Access

Figure 3 illustrates how the procedures interact with the developed Access database files. The "subinput" procedure has the most interactions with the database files of all of the procedures. Five databases are accessed in order to provide the procedure with all of the required information. Each of the following databases is described in detail later in this manual. The "nodes" database provides the number of nodes in the network as well as the mode of transportation outgoing from each node. The "arcs" database provides the number of arcs as well as the linear distance (in miles) of each arc. The "input" database provides the source city, the size of the shipment and the objective as provided by the user. The "rates" database provides the cost and the speed associated with each mode of transportation for use in the arc length calculations. Finally, the "xfer" database provides the transfer time and cost for each arc for use in the arc length calculations.

The "DSWP" procedure has limited interaction with Access. The value for k, the number of paths to be found, is extracted from the "input" database. The node numbers for the source and destination nodes are also required. These cities are stored in the "input" database and their corresponding node numbers are stored in the "nodes" database. Also accomplished in this procedure, is the clearing of the "trace" database of all results from prior software use, so that the results of two different DSS runs are not confused with one another.

The only interaction that the "trace" procedure has with Access is in the output of the paths. Once a complete path has been traced from the destination back to the source node, it is stored in the "trace" database. The number of nodes in the path, the path length, and the each node number in the path, ordered from destination to source, are stored in the database.

SOFTWARE INSTALLATION PROCEDURE

The developed software is a decision support system (DSS) which was written to run under the Windows 95 operating system. Using this system the computer in which the software is installed should meet the following specifications:

- Pentium 90 MHz platform
- VGA or better resolution monitor
- Windows 95 operating system
- [illegible] of hard disk space
- [illegible] of RAM

Three disks have been created in order to make installation procedures more user-friendly. These disks contain all of the files necessary to run the program. Two of the disks are installation disks for the program. The disk labeled Routes contains a database file on which the program runs and a Windows file which the user's computer may or may not need.

The database file named "routes.mdb" which is on the Routes disk needs to be copied to the C:\MyDocuments hard disk on the user's computer. This database defines a shipping network focused around the Mississippi River basin. Alternately, the user may create a different shipping network. The database file must be developed in Microsoft Access. The file should be named 'routes' and saved in the Microsoft Access format. This files should be saved in the 'My Documents' directory on the C:\ hard drive of the computer. Instructions on how to create an alternate database file are presented in a later section of this manual.

The other file on the Routes disk is named "Comctl32.ocx". The user should first check to see if this file exists in the C:\Windows\System hard disk directory. If not, the file should be copied from the third installation disk to the C:\Windows\System directory using Windows Explorer. The two remaining disks are labeled Shipdisk1 and Shipdisk2. They contain the files necessary to install the software. Installation instructions follow.

To install the software, insert the installation disk labeled 'Shipdisk1' into the 3.5" floppy drive (disk drive or A:\ drive) of the computer. Computers running under a Windows 95 platform will require users to left click with the mouse on the 'Start' button in the lower left of the display screen. From there, select the 'Settings' option. Under this option choose the 'Control Panel' option by left clicking on it while it is highlighted.

The user should then choose the option of 'Add/Remove Programs' by double left clicking on the option with the mouse. Once this has been done, the user should choose to left click on the 'Install' button on the 'Add/Remove Programs' screen. At the next screen, the user should insure that 'Shipdisk1' is in the floppy drive. The user should then left click on the 'Next' button on the bottom of the 'Install Program from Floppy Disk or CD-ROM' screen. The computer will automatically find the correct setup file. The user will then left click on the 'Finish' button to the bottom of the 'Run Installation Program' screen.

The computer will install 'Shipdisk1' and then request the user to enter 'Shipdisk2'. The computer will install 'Shipdisk2' and then inform the user when the installation program is complete and whether or not it was successful. The program will make its own icon on the Desktop of the user's computer. The software can be executed by choosing the 'Start' button in the lower left of the display screen. The option 'Programs' should be chosen. Here the user will find the program 'ShipCost'. To run the program, left click on 'ShipCost' while it is highlighted.

EXAMPLE CREATION OF A SHIPPING NETWORK

The developed software operates in conjunction with database tables that must be created and stored within Microsoft Access. These tables must be created *only* if the user wishes to create a unique shipping network. Otherwise, the database supplied with the software installation disks should be used. In order to further explain the makeup and creation of the tables for a new database, this section of the user's manual steps the reader through the entire process of creating each of the tables for a small network. For the purposes of this example, a network will be created which includes three cities. Each city will have two available modes of transportation.

The cities are labeled simply City A, City B, and City C. The two modes of transportation included in the example network are rail and barge. For the three cities, each with two modes of transportation, a total of twelve nodes will need to be created. These twelve nodes include three starting nodes (one for each city), three ending nodes, (again one for each city) and six city nodes (two for each city) one for each mode of transportation per city.

There are a total of thirty-six arcs in the network. Two arcs leave each of the starting nodes, and five arcs leave each of the city nodes. Four of these arcs go to the other cities, one for each transportation mode. The fifth arc goes to the city's end node. There are two separate arcs between two city node pairs. For example, between the two nodes labeled City A - Barge and City B - Rail there are two arcs, one going from City A to City B, and one pointing in the opposite direction. This notation infers that it is possible to travel by either rail or barge between Cities A and B. This allows the user flexibility in editing the network at a later time. A graphical representation of the nodes and arcs in the network is presented in Figure 4.

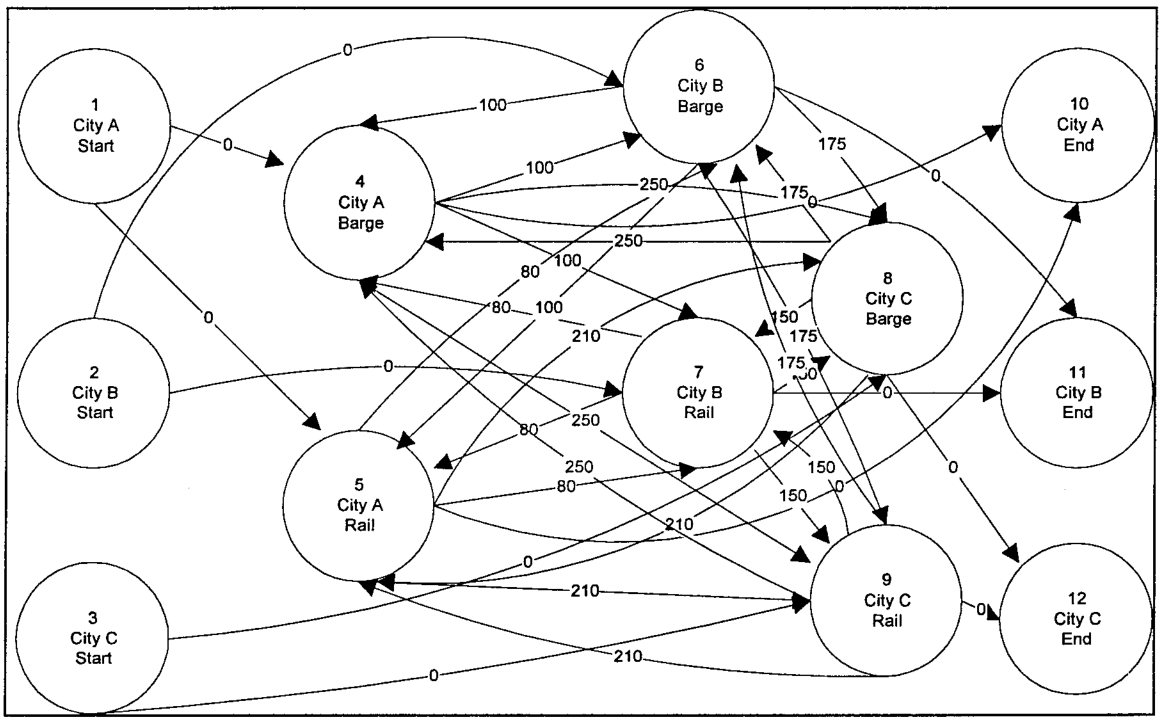

Figure 4: Graphical Representation of the Example Network

DATABASE CREATION

In order to create a database, Microsoft Access is started, and the New Database option is chosen from the File Menu. The database name, "routes", is entered into the File Name box, and the Create button is chosen, as shown in Figure 5. After the database has been created, a dialog box appears requesting the type of database object to be created. In this case a table needs to be created, so the table tab is selected and the New option button is chosen, as shown in Figure 6.

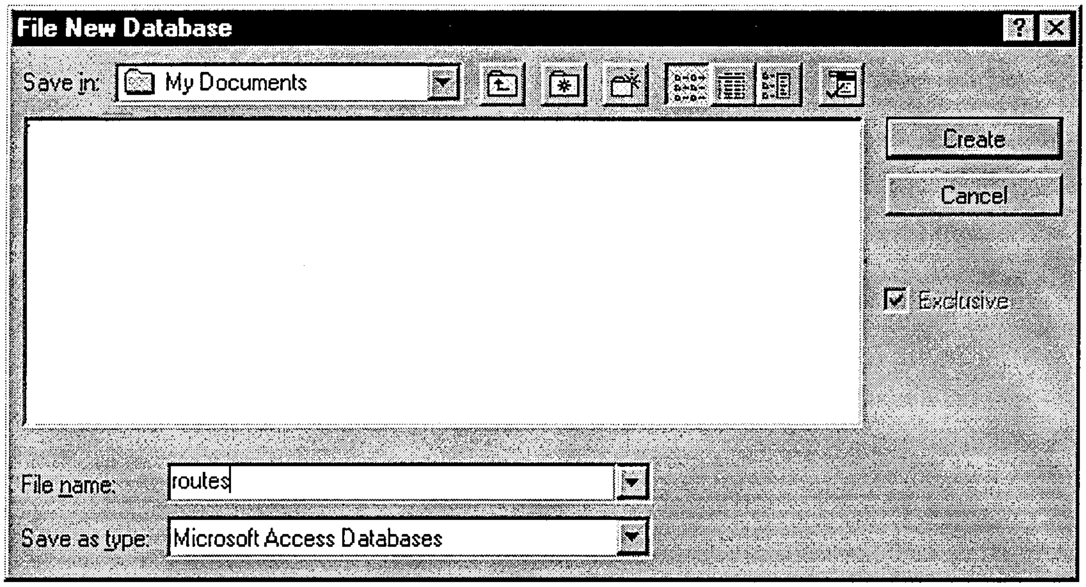

Figure 5: Access Create Database Screen

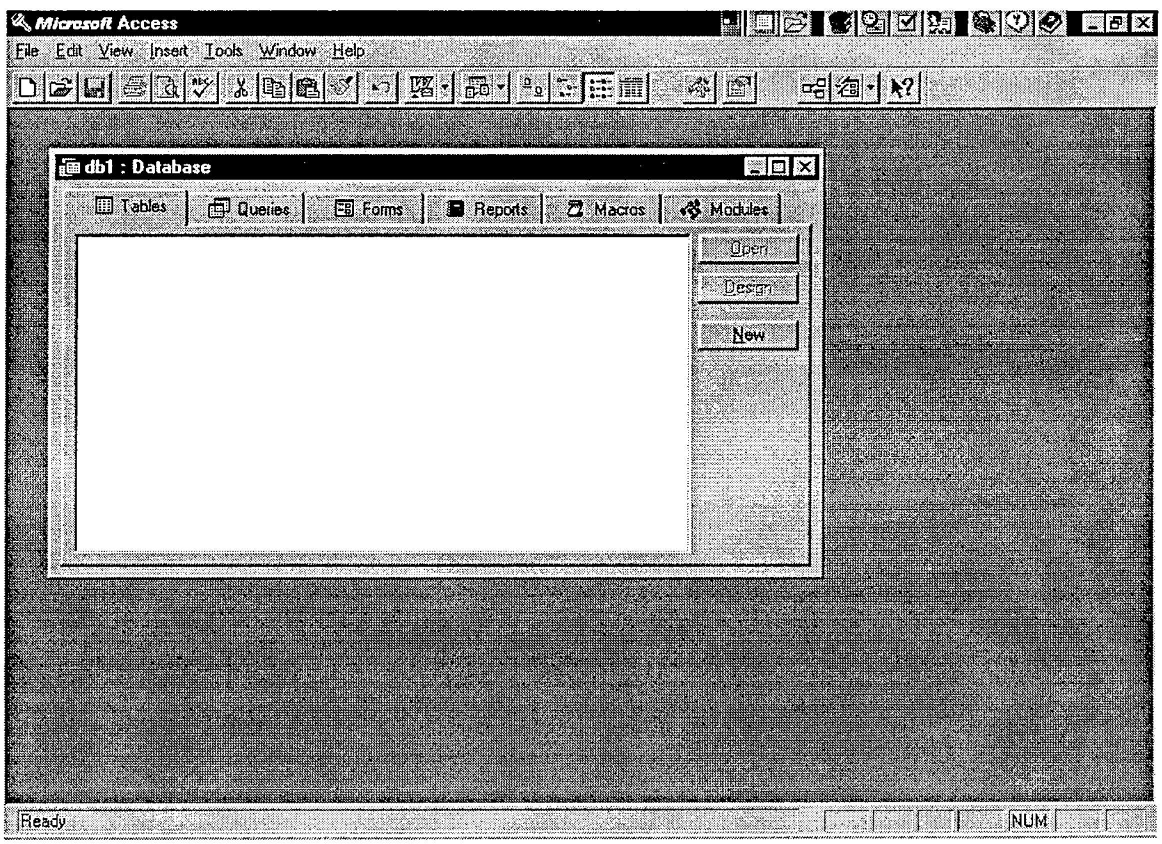

Figure 6: Access Create Table Window

Nodes Table

The first database table that will need to be created is the "nodes" table. This table defines the nodes that make up the network. After the New button is chosen in Figure 6, an option box appears asking the user to choose a viewing option for the new table, the user should choose the Design View option. This option allows the user to define the fields of the nodes table. The information that needs to be entered for the table includes the field name, data type and a description of the field. The information which needs to be entered to define the nodes table is illustrated in Figure 7. The table should then be saved with the name "nodes."

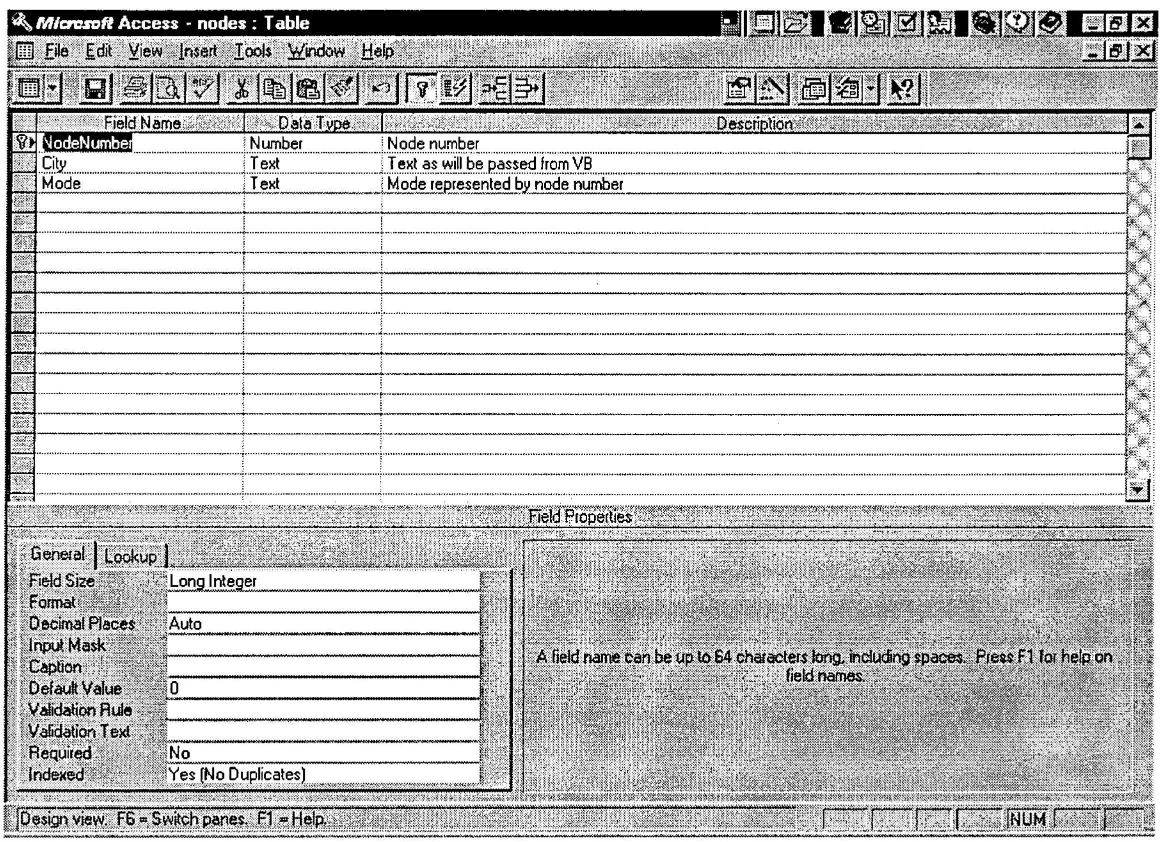

Figure 7: Access Creation of Nodes Table

After the table has been defined and saved, the appropriate data can be created to describe the nodes that make up the network. To enter data into the newly created database table, the user will need to switch to the datasheet view of the table. This can be accomplished by selecting Datasheet from the View menu in Microsoft Access. Once the datasheet view, which has a spreadsheet look, has been chosen, the user fills in the data fields for each of the twelve nodes. From Figure 4 the twelve nodes are described and with the node number and the mode which represents the node. For example, from Figure 4 it can be seen that node 4 represents City A and barge transportation. This information is entered for each of the nodes in the network. The completed nodes table for the example network is presented Table 1.

NodeNumber	City	Mode
1	City A	start
2	City B	start
3	City C	start
4	City A	barge
5	City A	rail
6	City B	barge
7	City B	rail
8	City C	barge
9	City C	rail
10	City A	end
11	City B	end
12	City C	end

Table 1: Nodes Table for the Example Network

Arcs Table

The “arcs” table can be created in the same way in which the “nodes” table was created. Following the same steps of creating a new table and then defining and saving the table, which were outlined above, the “arcs” table is created, and the data is entered for each of the 36 arcs shown in Figure 4. The arcs are the lines between the nodes in Figure 4. The numbers on the arcs represent the distance (in miles) between the two nodes that are connected by the arc. One entry needs to be made for each of these connecting arcs. For example, from Figure 4 the arc that goes from node 4 to node 6 has a length of 100 miles. The SourceNode entry for this arc would be 4, the DestNode entry would be 6 and the Distance entry would be 100. The entire example "arcs" table is illustrated in Table 2.

SourceNode	DestNode	Distance
1	4	0
1	5	0
2	6	0
2	7	0
3	8	0
3	9	0
4	6	100
4	7	100
4	8	250
4	9	250
4	10	0
5	6	80
5	7	80
5	8	210
5	9	210
5	10	0
6	4	100
6	5	100
6	8	175
6	9	175
6	11	0
7	4	80
7	5	80
7	8	150
7	9	150
7	11	0
8	4	250
8	5	210
8	6	175
8	7	150
8	12	0
9	4	250
9	5	210
9	6	175
9	7	150
9	12	0

Table 2: Arcs Table for the Example Network

Rates Table

The "rates" table is created, defined, and saved in the same way as the two previously described tables. The "rates" table has a total of four entries, one for each of the modes, one for start nodes and another for the end nodes. The start and end nodes need to be included in the rates table so that the k-shortest path algorithm will know what values to assign to arcs leaving these nodes. The k-shortest path algorithm is used by the software to determine the least cost or least throughput time path between a given origin and destination in the network. The "rates" table for the example network is illustrated in Table 3. The values shown in Table 3 are arbitrary values. The rates (rate) are given in dollars per container-mile; average speeds (avgspd) are given in miles per hour.

mode	rate	avgspd
Rail	0.351	37
Barge	0.0079	5
start	0	0
end	0	0

Table 3: Rates Table for the Example Network

Xfer Table

This table defines both the cost and time to transfer freight between the transportation modes present in any city in the network. The "xfer" table for the example network will consist of two entries for each node in the network, excluding the starting nodes. There are no entries for the starting nodes because no arcs enter those nodes; therefore no transfer will ever be necessary at those nodes. One entry will provide the cost and time to transfer a shipment from barge to the mode of transportation that the node represents. The other entry will provide the cost and time to transfer a shipment from rail to that same node. If the modes of transportation for two nodes are the same, the transfer cost and transfer time is zero. Similarly, the transfer cost and time for a shipment entering a destination node are zero as well. As an example, from Figure 1 it can be seen that between node 4 and node 6 the transfer cost and time would both be equal to zero because both nodes have the same transportation mode (barge). However, a transfer between node 4 and node 7 would involve a transfer time and cost to account for the transfer of a material from barge to rail. The "xfer" table for the example problem is presented in Table 4. The third column in the table gives the transfer cost in \$/container, the fourth column gives the transfer time in hours/container.

4	Barge	0	0
4	Rail	40	1.5
5	Barge	55	2
5	Rail	0	0
6	Barge	0	0
6	Rail	45	1.25
7	Barge	50	2.5
7	Rail	0	0
8	Barge	0	0
8	Rail	45	1.25
9	Barge	55	2.25
9	Rail	0	0
10	Barge	0	0
10	Rail	0	0
11	Barge	0	0
11	Rail	0	0
12	Barge	0	0
12	Rail	0	0

Table 4: Xfer Table for the Example Network

Input and Trace Tables

The "input" table and the "trace" tables need only to be created, defined, and saved in Microsoft Access just as all of the other tables were. These tables are automatically populated by the Visual Basic user interface as needed. The tables and the fields need merely to be defined for the user interface to function properly. The fields that need to be defined for each table are shown in Tables 5 and 6. Table 5 provides the k-shortest path algorithm with the source city (NS), the destination city (NF), the number of units to be shipped (Units), the number of container necessary for the shipment (Containers), the unit cost of the cargo (Unit Cost), whether the user wants to minimize cost or time (Objective), the source mode (Mode) and destination mode (ModeFrom), and the number of paths to be found (k). Table 6 is populated by the k-shortest paths algorithm with the information about the shortest paths. This table is a shorthand form for the information that is provided to the user in the shortest path screens of the user interface.

NS	NF	Units	Container	Unit Cost	Objective	Mode	ModeFrom	k
City A	City C	100	100	$100.00				10

Table5: Input Table

ID	length	p1	p2	p3	p4	p5	p6
1	337.716	23	5	2	17	0	0
2	418.1372	23	6	3	17	0	0
3	424.848	23	5	11	14	2	17
4	425.0896	23	6	9	3	17	0
5	427.716	23	6	2	17	0	0
6	487.1646	23	6	12	3	17	0
7	493.1372	23	5	3	17	0	0
8	499.6184	23	6	9	2	17	0
9	500.0896	23	5	9	3	17	0
10	505.0422	23	6	16	3	17	0
11	507.5252	23	6	16	12	3	17
12	509.2302	23	5	11	3	17	0
13	509.2302	23	5	11	3	17	0
14	512.716	23	4	2	17	0	0
15	514.848	23	6	11	14	2	17
16	528.1372	23	4	3	17	0	0
17	535.0896	23	4	9	3	17	0
18	536.678	23	5	14	3	17	0

Table6: Trace Array

EXECUTING THE DEVELOPED SOFTWARE

Before the user can utilize the developed software, the software must be installed as previously discussed. After the software has been installed, the user must activate the software by double-clicking the left mouse button on the 'ShipCost' icon located on the desktop of the computer.

Startup and Input Screens

Once the software has been executed, the screen that greets the user is the startup screen, which is shown in Figure 8. This screen gives the user three options: 'Run Program', 'Update Database Information', or 'Exit the program'. The user selects one of these options by using the left mouse button to click on the appropriate option.

If the user wishes to analyze the least cost or least time alternative between an origin-destination pair, then the 'Run Program' option should be selected. The 'Input Data' screen, Figure 9, will next appear. This screen allows the user to enter all of the necessary information needed by the program to execute the appropriate shortest path calculations. Several pieces of information need to be input by the user on this screen. The screen prompts the user to enter the origin city, the destination city, the number of units to be shipped, the number of units that fit into one container, the unit cost to the producer, the annual percentage to be used in calculating the inventory carrying cost, the objective (to minimize cost or time), and the number of alternatives to display to the user.

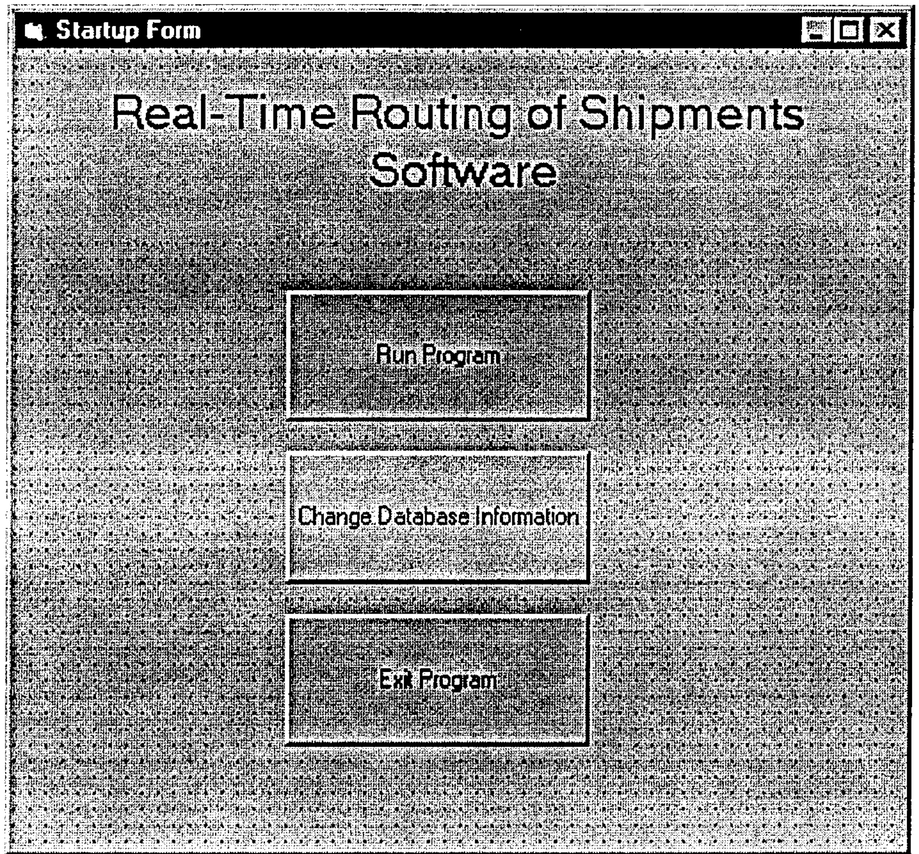

Figure 8: Startup Screen

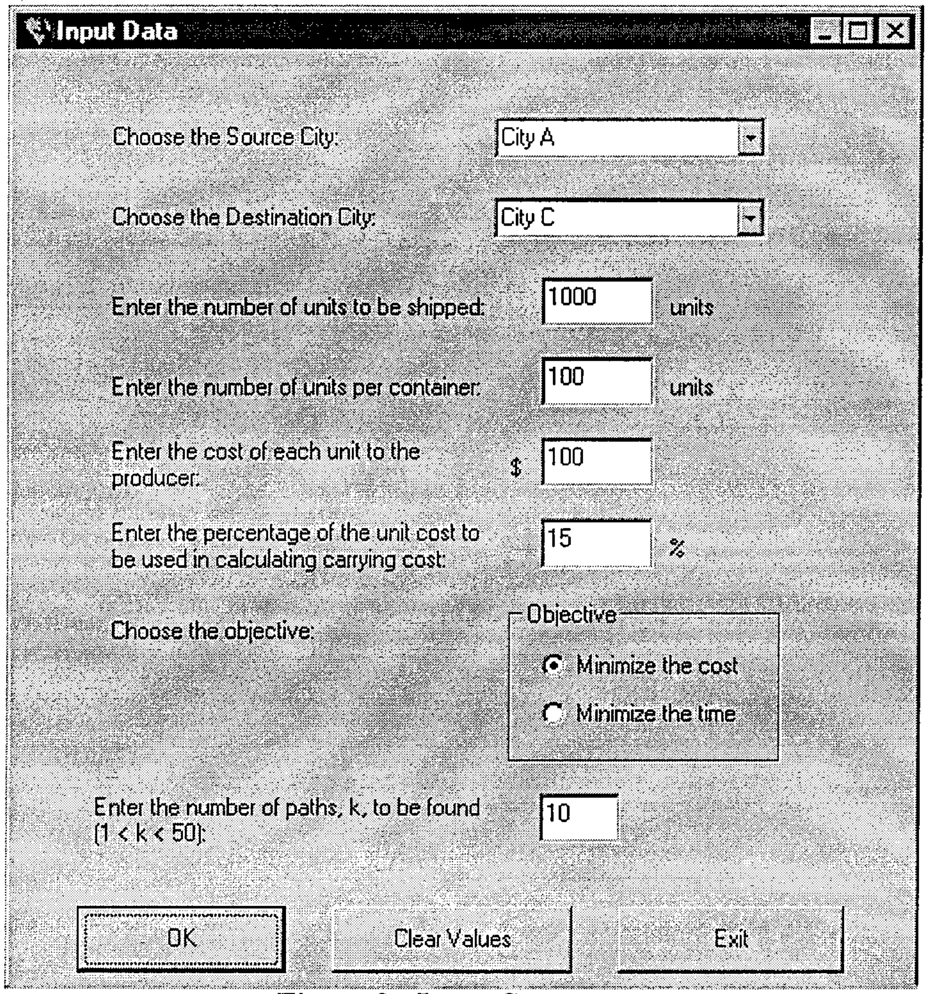

Figure 9: Input Screen

The user must select the source and destination cities from pull down menus. These pull down menus read the information stored in the database 'routes' and list only those cities that are defined in the database. The user must then enter the number of units to be shipped, the number of units that will fit into none container, and the unit cost of each unit. If there are different types of units being shipped in one container, then the unit cost should be an average of the different unit costs for the types of units being shipped. The user must also enter the annual percentage of the unit cost the user customarily uses in calculating inventory or holding costs for the units being shipped. This information is used by the software to balance increased inventory carrying costs associated with longer shipment throughput times against lower transportation cost rates associated with slower modes of transportation.

Again, if there is more than one type of unit being shipped, an average should be taken and entered into the appropriate space on the 'Input Data' screen. The user can then choose to either minimize cost or time, the appropriate option is chosen by left clicking on the circle by the option he or she wishes to analyze. Finally, the user must enter the number of alternatives to be displayed.

Once all values have been entered, the user should click on the 'OK' button at the bottom of the screen. The user can choose to clear the values that he or she entered by selecting the 'Clear Values' button at the bottom of the 'Input Data' screen. The computer will begin processing the alternatives. While the alternatives are being processed a 'Time Left' screen will be shown. The user may choose to stop the analysis at any time.

Output Screens

Once the shortest path algorithm is completed, screens containing the cost and time of the k shortest paths will appear in increasing order, as shown in Figure 10. Notice that only five alternatives are shown on this screen at a given time. If the user requested that the program report more than five alternatives, a 'See Next 5 Best Solutions' button will appear at the bottom of the 'K-alternative' screen. The number of paths which appear are those that do not contain cycles. For example, if 20 paths were requested, only 12 may appear in the output screens, the remaining 8 paths contained cycles and are therefore not presented as viable options.

The user can examine any of the alternatives by left clicking on the radio button next to the path, (see Figure 10), he or she wishes to examine. The user would then left click on the 'Examine a Specific Path' button.

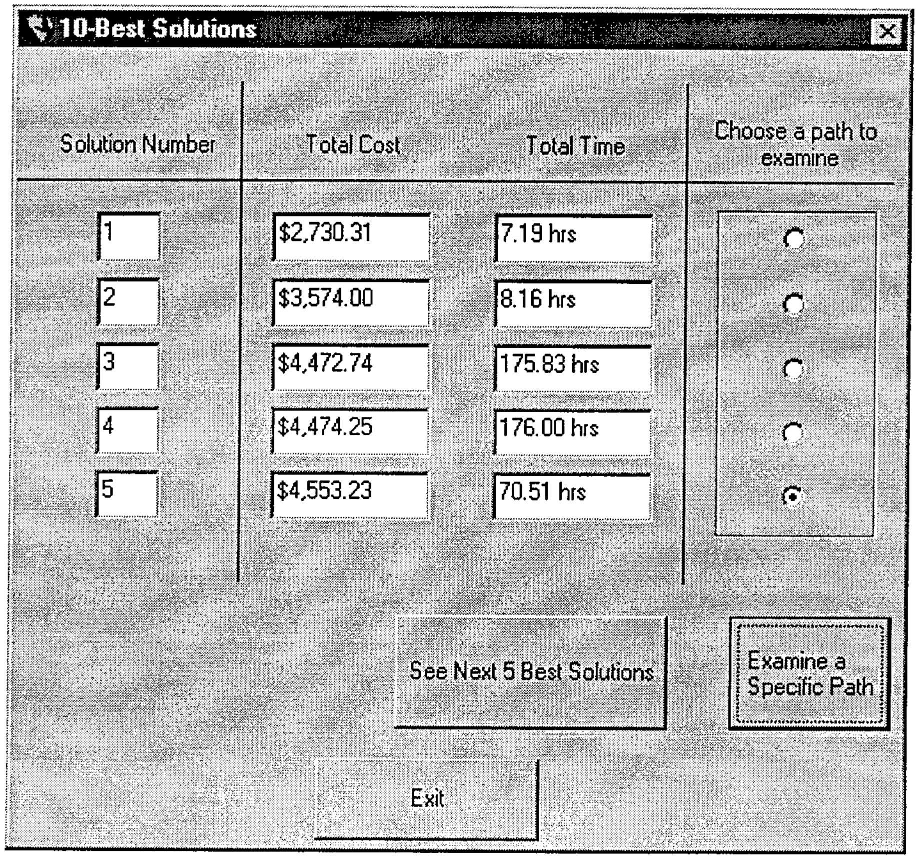

Figure 10: K-Alternatives Screen

The specific path will then be displayed on the 'Path Description' screen. An example of this screen is shown in Figure 11. The total cost and total time for alternative is shown at the top of the screen. The cost and time for each portion of the path is shown next to each route. Figure 11 shows that the best route for this example is represented by a barge route between the source city , City A, and an intermediate city, City B, and then a rail route from City B to the destination city, City C.

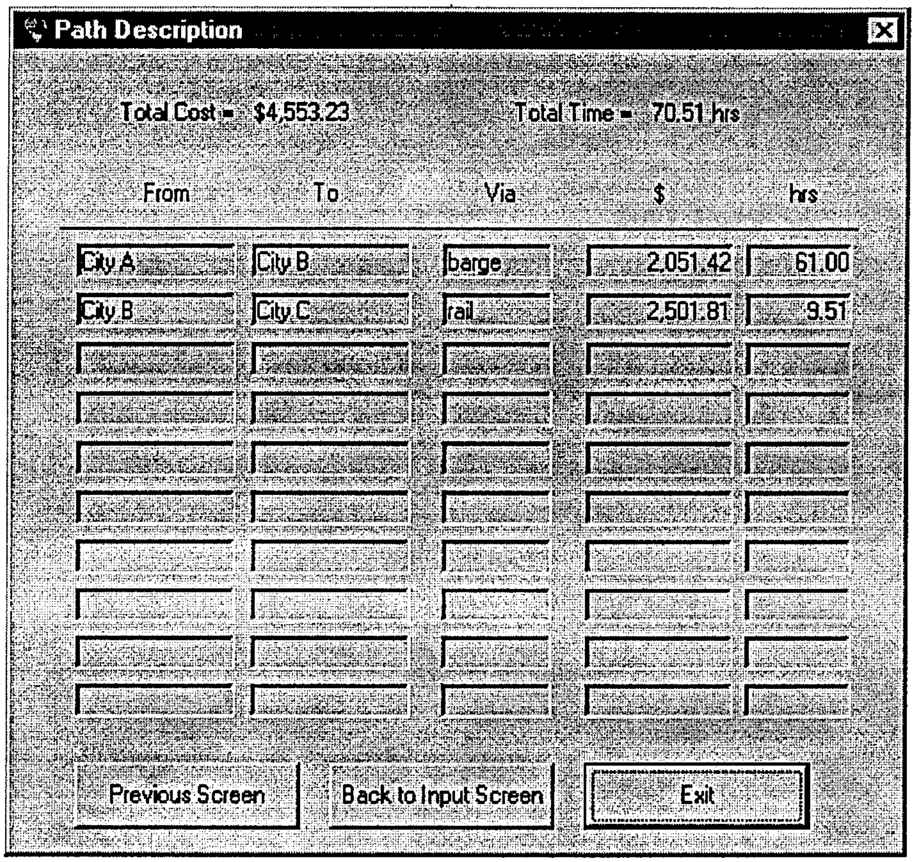

Figure 11: Path Description Screen

From here, the user can choose to Exit, go back to the 'K-alternatives' screen (Figure 10), or go back to the 'Input Data' Screen (Figure 9). The appropriate option can be chosen by using the left mouse button to click on the corresponding button at the bottom of the 'Path Description' screen.

EDITING DATABASE TABLES

Creation of the "routes" database in Microsoft Access has been previously described. If the "routes.mdb" Microsoft Access file already exists, then the user can edit portions of the file . The user can do this at he 'Startup' screen (Figure 8). The user chooses the 'Change Database Information' button from the screen. Once this is done, the 'Change Database' screen appears, as shown in Figure 12.

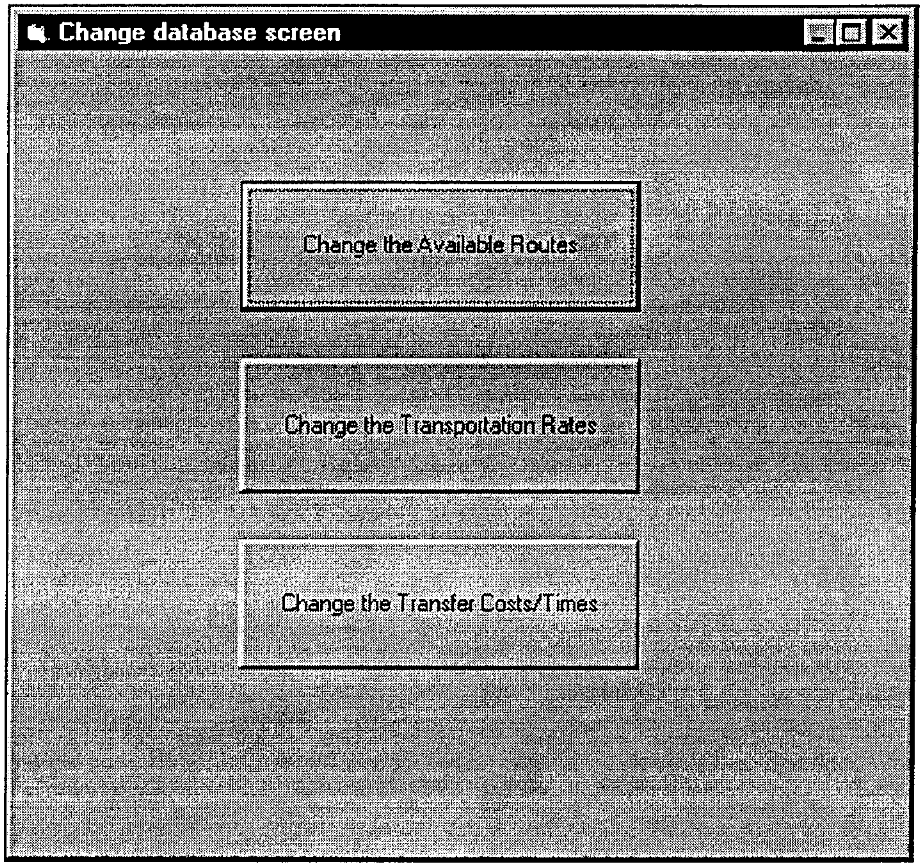

Figure 12: Change Database Screen

There are many reasons that user may want to edit the network information. If specific routes along the network become unavailable for travel, they can be taken out of the network. If transportation costs change for a specific mode, they can be changed or if transfer times for a certain mode of transportation increase or decrease, those figures can also be updated by the user.

If the user selects the first option, 'Change the Available Routes', he/she is allowed to edit the distance between two cities for a specific mode of transportation, Figure 13. This may become warranted if, for instance, there is a flood of the river, thereby impeding the flow of barge traffic between City B and City C. The user could change the distance in the database to a very large number, thereby effectively removing this route from the network for all intensive purposes.

The source and destination cities available to change are read directly form the cities in the 'arcs' database. The user must first choose the source city (origin) before the program will allow the user to change the destination city. Once both cities are chosen, the user can choose the mode of transportation to change. Only the modes of transportation available between the source and destination city will be listed. The modes of transportation are read directly from the 'arcs' database. Once the desired mode is selected, the current mileage listed in the database will appear. The user can then type the new mileage information in the space provided.

To update the database, the user should click on the 'Make Changes to Database' button. To cancel changes, the user should select the 'Return to Start' button.

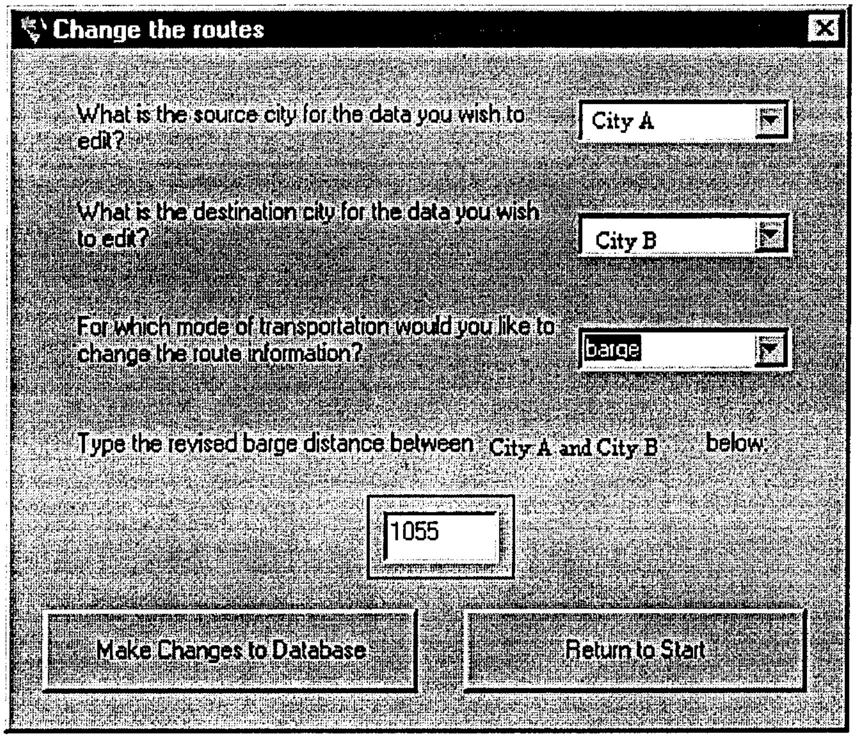

Figure 13: Change the Available Routes Screen

If the user wants to edit the cost or speed for a specific mode of transportation he/she would choose the 'Change the Transportation Rates' button on the 'Change Database' screen (Figure 12.). This may become necessary if the transportation cost for a mode of transportation increases or decreases. For example, if there were to be a railroad strike, the price of shipping goods via track may be temporarily increased. The user could edit the "rates" table to reflect this change. Figure 14 shows the 'Change the Transportation Rates' screen.

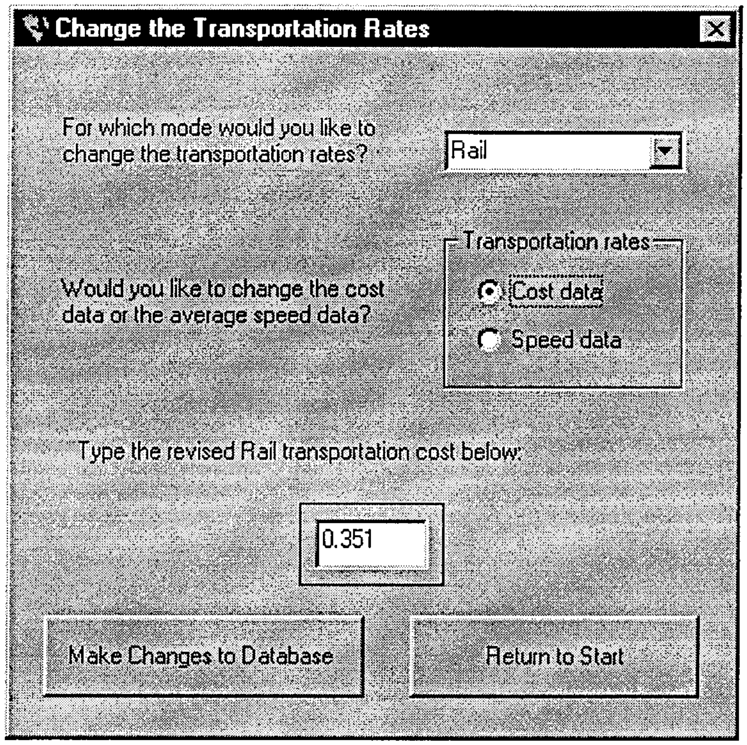

Figure 14: Change the Transportation Rates Screen

The user must select the mode of transportation to be changed form the pull-down menu at the top of the screen. The use must then choose whether the cost or speed data needs to me changed for the chosen mode of transportation by clicking the appropriate rate button. The current data reported in the rates array will be listed in the white box. The user can alter this information by typing new data in the space provided. To permanently change the database, the user should click on the 'Make Changes to Database' button. To cancel the changes the user should click on the 'Return to Start' button.

The 'Change the Transfer Costs/Times' button on the 'Change Database' screen allows the user to do just that. The user is given the opportunity to edit the transfer cost and or time between two modes of transportation for a city. This may become useful if, for example, the port at City A gets new equipment, which allows for less expensive servicing of barge to rail transfers. Figure 15 shows the change the transfer costs/times screen.

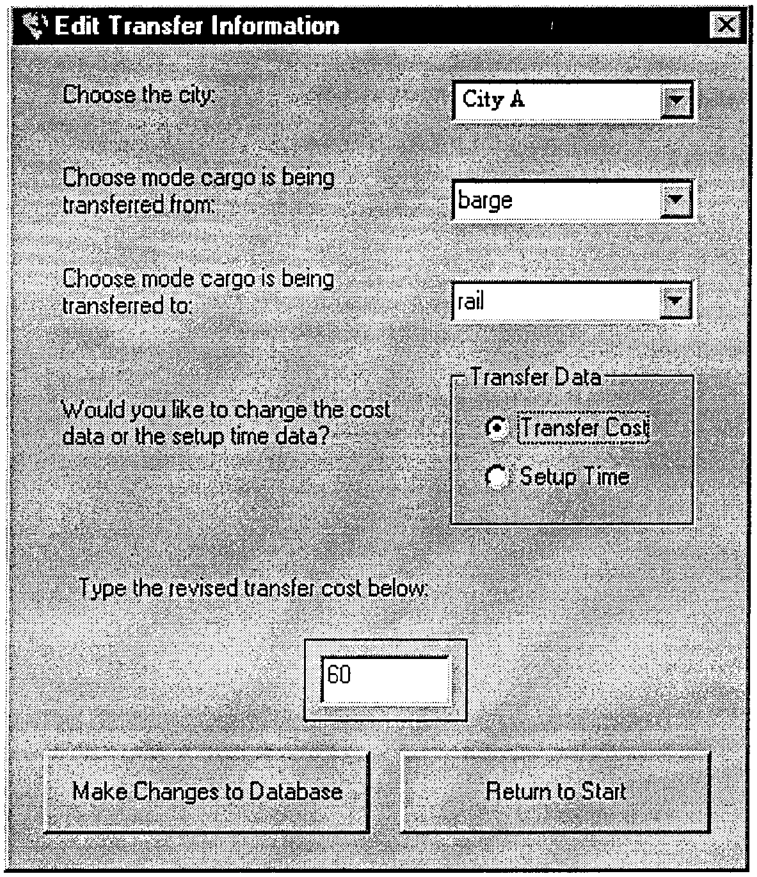

Figure 15: Change the Transfer Costs/Time Screen

The user must choose the city at which the changes in transfer information need to be made. The available cities are read directly from the 'nodes' database. The user must then select what two modes of transportation he or she wishes to change. The modes of transportation available to change at a particular city are read directly from the 'nodes' database. The user must choose to change the transfer cost or transfer time between the modes of transportation. The current information stored in the 'transfer' database is listed in the box provided. The user may change the information by typing the new information in the space provided. To permanently change the database, the user should click on the 'Make Changes to Database' button. To cancel the changes the user should click on the 'Return to Start' button.

SUMMARY

This software described in this document was developed in order to help shippers and transportation providers determine the best combination of modes and routes for a specific cargo shipment. Questions about the software or suggestions for its use or improvement are welcomed and may be directed to the point of contact listed below.

Eric M. Malstrom,
Professor and Head
Department of Industrial Engineering
University of Arkansas
Bell 4207
Fayetteville, AR 72701
emm@engr.uark.edu

Requests for additional copies of the software or this user's manual can be directed to the Mack-Blackwell Transportation Center at the address below.

Lyn Gattis
Mack-Black Transportation Center
University of Arkansas
Bell 4190
Fayetteville, AR 72701

REFERENCES

1. Benjamin, J., "An Analysis of Mode Choice for Shippers in a Constrained Network with Applications to a Just-In-Time Inventory", *Transportation Research B,* Vol. 24, 1990.

2. Boardman, B.S. and E.M. Malstrom, "Real-Time Routing of Shipments Considering Transfer Costs and Shipment Characteristics", Research Report FR-1050, Mack Blackwell Transportation Center, University of Arkansas, 1997.

3. Broadstreet, F.V., "Intermodal Lane Profitability: A Decision Support System", MSIE Thesis, Department of Industrial Engineering, University of Arkansas, 1996.

4. Burke, Jack, "New Wave in Intermodal: Fighting for Traffic", *Traffic World*, Vol. 242, No. 6, 1995.

5. Chew, Sze H., "A Methodology for Comparative True Cost Assessment of Transportation Modes", MSIE Thesis, Department of Industrial Engineering, University of Arkansas, May 1995.

6. Jara-Diaz, S.R., "The Estimation of Transport Cost Functions: A Methodological Review", *Transport Reviews*, Vol. 2, 1982.

7. Kondratowica, L.J., "Simulation Methodology for Intermodal Freight Transportation Terminals", *Simulation*, Vol. 55, No. 1, 1990.

8. Korpela, J. and M. Tuominen, "Analytic Approach to the Analysis and Selection of Transportation Services", *Proceedings of the IEEE International Engineering Management Conference,* Dayton, Ohio, October 1994.

9. Locklin, Phillips D., *Economics of Transportation*, Richard D. Irwin Inc., Homewood, Illinois, 1996.

10. *MapInfo*, Desktop Mapping Software, MapInfo Corporation, Troy, New York 1994.

11. Mittra, S.S., *Decision Support Systems Tolls and Techniques*, John Wiley and Sons, New York, 1986.

12. Moore, J.L., E.M. Malstrom, R.G. Kasilingam, and D.P. Butler, "Computer Assisted Cost Assessment of Intermodal Transportation Linkages, Phase II", Research Report FR-1036, Mack Blackwell Transportation Center, University of Arkansas, 1996.

13. Muller, G., *Intermodal Freight Transportation*, 2nd edition, ENO Foundation for Transportation, 1989.

14. Olsen, D.L. and J.F Courtney Jr., *Decision Support Models and Expert Systems*, MacMillan, NewYork, 1992.

15. Philips, D.T. and A. Garcia-Diaz, *Fundamental of Network Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

16. Revis, J.S. and C. Tarnoff, *The Nation's Public Works: Report on Intermodal Transportation*, National Council on Public Works Improvement, Washington, DC., 1987.

17. Shier, D.R., "Computational Experience With an Algorithm for Finding the k Shortest Paths in a Network", *Journal of Research of the National Bureau of Standards -B. Mathematical Sciences*, Vol. 78B, No. 3, 1974.

18. Shier, D.R., "Iterative Methods for Determining the K Shortest Paths in a Network", *Networks*, Vol. 6, No. 3, 1976.

19. Sparkman, D.L., "Here's an Idea: Intermodal Internet", *Transport Topics*, No. 3121, May 29, 1995.

20. Trusty, K. and E.M. Malstrom, "A Feasibility Assessment of Truck-Barge Intermodal Freight Transportation", Research Report FR-1079, Mack Blackwell Transportation Center, University of Arkansas, 1998.

21. US Department of Transportation, *Federal State and Local Transportation Financial Statistics: Fiscal Years 1981-1991,* Office of Economics, Washington D.C., 1993.