MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0374719 5

ornl

ORNL/TM-12090

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

INTERLINE 5.0 — An Expanded Railroad Routing Model: Program Description, Methodology, and Revised User's Manual

P. E. Johnson
D. S. Joy
D. B. Clarke
J. M. Jacobi

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.
UCN-7969 (3 9-77)

Prepared for the
Transportation Management Division
Office of Environmental Restoration
and Waste Management
U.S. Department of Energy

(TTP Number: OR-139002(4))
Date: March 1993
Revision No.: 0

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-12090

Chemical Technology Division

INTERLINE 5.0— AN EXPANDED RAILROAD ROUTING MODEL: PROGRAM DESCRIPTION, METHODOLOGY, AND REVISED USER'S MANUAL

P. E. Johnson*
D. S. Joy
D. B. Clarke†
J. M. Jacobi†

*Computing Applications Division
†University of Tennessee, Knoxville, Transportation Center

Prepared for
Environmental Restoration and Waste Management
Transportation Management Division, EM-561
U.S. Department of Energy
Washington, DC 20585
TTP No. OR-139002(4)

Date Published—March 1993

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6285
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0374719 5

CONTENTS

ACKNOWLEDGMENTS vii

ABSTRACT ix

1. PROGRAM DESCRIPTION 1
   - 1.1 OPERATIONAL CHARACTERISTICS OF THE U.S. RAILROAD SYSTEM 2
   - 1.2 INTERLINE RAILROAD NETWORK 4
     - 1.2.1 INTERLINE Naming Conventions 5
     - 1.2.2 General Routing Considerations 8
       - 1.2.2.1 Rail Routing 9
       - 1.2.2.2 Alternative Routing 9
       - 1.2.2.3 Barge Routing 12
       - 1.2.2.4 Other Routing Considerations 13
       - 1.2.2.5 Population Density Estimates 16

2. METHODOLOGY 17
   - 2.1 ORGANIZATION OF THE NETWORK FOR ROUTING ANALYSIS 17
     - 2.1.1 Railroad Company Representation 17
     - 2.1.2 Transfer Connections 18
   - 2.2 BASIC ROUTING METHODOLOGY 21
     - 2.2.1 Routing Nomenclature 21
     - 2.2.2 The Routing Objective Function 21
     - 2.2.3 Shortest Path Algorithm Overview 23
     - 2.2.4 The INTERLINE Routing Algorithm 24
       - 2.2.4.1 Initialization Sequence 26
       - 2.2.4.2 Main Loop 26
     - 2.2.5 Example of a Route from an Origin Served by a Single Railroad 28
     - 2.2.6 Example of a Route from an Origin Served by Multiple Railroads 32
   - 2.3 PROGRAM OPTIONS 36
     - 2.3.1 Link Avoidance 36
     - 2.3.2 Node Avoidance 36
     - 2.3.3 Directed Routing 38
     - 2.3.4 Subnetwork Prohibition 38
     - 2.3.5 Alternative Routing 38
     - 2.3.6 Population Density 39

3. INTERLINE USER'S MANUAL 43
   - 3.1 INSTALLATION 43
   - 3.2 STARTING THE INTERLINE PROGRAM 44
   - 3.3 COMMANDS 46

3.3.1 Route 47
3.3.1.1 Alternative Ways to Define Origins/Destinations 50
3.3.2 Reroute 51
3.3.3 Retype 52
3.3.4 Options 52
3.3.4.1 Alternative Routing 55
3.3.4.2 Barge Routing 55
3.3.4.3 Population Density 57
3.3.5 Node 58
3.3.6 Link 61
3.3.7 Remove 63
3.3.8 Transfer 65
3.3.9 Help 67
3.3.10 Stop 69

REFERENCES 71

Appendix A. SUBNETWORK LISTING 73

Appendix B. ADVANCED FEATURES 81

LIST OF FIGURES

1. Rail route between Boston, Massachusetts, and Los Angeles, California 10
2. Alternative rail routes between Charleston, South Carolina, and Las Vegas, Nevada 11
3. Example of barge routes 14
4. Graphic illustration of (a) an undecomposed network and (b) a decomposed network 19
5. Interline transfer link model 20
6. Shortest path algorithm example: (a) base network, (b) step 1, (c) step 2, (d) step 3, (e) step 4, (f) step 5, (g) step 6, and (h) step 7 25
7. Example of network structure 29
8. Pivot through subnetwork 1 at (a) the start of and (b) the end of iteration 30
9. Pivot through subnetwork 2 at (a) the start of and (b) the end of iteration 31
10. Pivot through subnetwork 3 at (a) the start of and (b) the end of iteration 33
11. Pivot through subnetwork 4 at (a) the start of and (b) the end of iteration 34
12. Complete path tree 35
13. Completed path tree for a multiple railroad origin route 37
14. Network impedances after increasing impedances on the base route 40
15. Completed path tree—alternative routing 41
16. Path comparison—normal vs alternative routing 42
17. Modified RAILROAD.BAT file 45
18. Opening screen 46
19. Example of using the route command 48
20. List of node names in Wisconsin starting with the letter m 51
21. Use of the reroute command 53
22. Option command choices 54
23. Alternative routing example 56
24. Barge routing calculation 57
25. Population density table and RADTRAN input data 59
26. Node deletion and restoration 61
27. Link removal and restoration 63
28. Railroad removal and restoration 65
29. Changing transfer penalties 68
30. The HELP command 69

LIST OF TABLES

1. Node naming convention 6
2. Railroad naming convention 7
3. INTERLINE commands 46
A.1. Network 10 subnetworks 75
B.1. Mainline classification codes and default values 85
B.2. Route listing codes 87

ACKNOWLEDGMENTS

The INTERLINE routing model was originally developed over a decade ago at the Oak Ridge National Laboratory (ORNL) in support of the Waste Transportation Studies sponsored by the U.S. Department of Energy (DOE) through the Transportation Technology Center at Sandia National Laboratories and through the Transportation Operations and Traffic Management organization. Over the years, ORNL has received funding for further development and enhancement of the INTERLINE program from DOE's Office of Civilian Radioactive Waste Management (OCRWM) and the Office of Environmental Restoration and Waste Management (EM). During FY 1992, OCRWM provided funding for the development of this document. Publication of this document was funded by the Transportation Management Division (EM-561).

The authors wish to acknowledge the efforts of E.L. Hillsman and B.E. Peterson of ORNL for their efforts in designing and developing the methodology for the original version of the INTERLINE model. The authors also wish to thank L.M. Green, of TRW Environmental Safety Systems, the Civilian Radioactive Waste Management M&O contractor, for his efforts in reviewing this document.

ABSTRACT

A rail routing model, INTERLINE, has been developed at the Oak Ridge National Laboratory to investigate potential routes for transporting radioactive materials. In Version 5.0, the INTERLINE routing algorithms have been enhanced to include the ability to predict alternative routes, barge routes, and population statistics for any route.

The INTERLINE railroad network is essentially a computerized rail atlas describing the U.S. railroad system. All rail lines, with the exception of industrial spurs, are included in the network. Inland waterways and deep water routes along with their interchange points with the U.S. railroad system are also included. The network contains over 15,000 rail and barge segments (links) and over 13,000 stations, interchange points, ports, and other locations (nodes).

The INTERLINE model has been converted to operate on an IBM-compatible personal computer. At least a 286 computer with a hard disk containing approximately 6 MB of free space is recommended. Enhanced program performance will be obtained by using a random-access memory drive on a 386 or 486 computer.

INTERLINE 5.0— AN EXPANDED RAILROAD ROUTING MODEL: PROGRAM DESCRIPTION, METHODOLOGY, AND REVISED USER'S MANUAL

P. E. Johnson,* D. S. Joy, D. B. Clarke,† and J. M. Jacobi†

1. PROGRAM DESCRIPTION

The potential impacts associated with the transportation of hazardous materials are important to shippers, carriers, and the general public. This is particularly true for shipments of radioactive material. Shippers are primarily concerned with safety, security, promptness, and equipment requirements. The carriers are concerned with the impact that radioactive shipments may have on their operations—particularly if such materials are involved in an accident. The general public has also frequently expressed concerns regarding the safety of transporting radioactive and other hazardous materials through their communities. Since transportation routes are a central characteristic in most of these issues, the prediction of likely routes is the first step toward resolution of these issues.

A rail routing model, INTERLINE, has been developed at Oak Ridge National Laboratory (ORNL) for the express purpose of predicting likely rail routes. While this model was developed to investigate potential routes for transporting radioactive materials, the INTERLINE model can be used to estimate routes for any commodity. Model development was initiated in early 1980 when a shortest path algorithm was developed to predict routes on the U.S. railroad system.[1] The routing algorithm has subsequently been enhanced to include the ability to predict alternative routes which can then be evaluated as part of the overall risk assessment for establishment of a radioactive materials transportation system. Other recent enhancements include the ability to route shipments on the inland waterway system and the ability to calculate population statistics for any route generated by the INTERLINE model.

---

*Computing Applications Division

†University of Tennessee, Knoxville, Transportation Center

INTERLINE's basic function is straightforward. Given an origin and destination, the program will determine the set of nodes and links comprising the "optimal" route between the two. This optimal route is determined by using predefined objectives which can be modified by the user. Fundamentally, the program minimizes the length of the route subject to user-specified criteria. The basic output of the program is a listing of the points and segments comprising the selected route.

This report contains three sections. The first section presents a general overview of the routing capabilities of the INTERLINE model with a number of examples. A short discussion of the INTERLINE naming conventions for nodes and railroads is also included. Section 2 presents a description of the methodology used to calculate rail routes. (The casual reader may want to skip this section.) The routing methodology is presented at a simplified level using narrative and examples to illustrate the calculational techniques. Additional resource material for a more detailed derivation of the mathematical formulations is referenced as well. Section 3 contains the INTERLINE user's manual, which includes a detailed discussion of the INTERLINE routing commands. Numerous examples are included in this section to illustrate how these commands operate. Appendix A lists the 96 subnetworks contained in INTERLINE, while the more advanced commands that are included in the expert option, are discussed in Appendix B. Since the general user is not expected to need these commands, their description was moved to an appendix in order to simplify the user's manual section. The casual user needs only to read and understand the material in Sects. 1 and 3.

1.1 OPERATIONAL CHARACTERISTICS OF THE U.S. RAILROAD SYSTEM

Any mathematical model designed to predict rail transportation routes must be capable of simulating the operation of the U.S. railroad system. This system is composed of a large number of independent companies that compete economically while simultaneously cooperating through interchange agreements to efficiently move freight across the country. Each company generally owns its own network of rail lines. In some instances, a company may have operating rights on a rail line owned by another company; this concept is referred to as "trackage rights." For more information the reader is referred to ref. 2.

The overall network is, therefore, an aggregate of the interconnected track networks of hundreds of individual railroad systems. Virtually all railroads in the United States are controlled by private companies. Only a small amount of U.S. rail mileage is publicly owned, and most

of this is dedicated to passenger operation—specifically commuter service in major urban areas (see ref. 3). Private ownership of rail lines places constraints on the movement of commodities. No single railroad company is able to serve all possible origins and destinations within the country, and thus, at least two railroads must be used for most shipments.

The ability of railroad companies to freely exchange equipment between their networks is one feature of the U.S. rail system. Although any single company is limited to regional service, railroads cooperate by exchanging cars moving to points beyond their individual service areas. These exchanges occur at designated transfer, or interchange, points where the lines of two or more companies meet. Sometimes interchange is handled by a third party, such as a terminal railroad. Interchanges are the bridges that connect the networks of the individual railroads thereby forming a national network.

Many locations are served by more than one railroad, and the shipper may choose which railroad will transport the shipment. This choice is based on a number of factors including cost, quality of service, and the railroad's willingness to provide service. While the shipper is, in principle, free to specify the railroads to be used as well as the interchange points, most shippers generally negotiate with a single railroad company, and that railroad will arrange to have the shipment interchanged with the other carriers involved in completing the shipment.

There are two aspects of simulating railroad routing practices. The first is the route that traffic will take between points on a single railroad system. The routing algorithm is designed to preferentially route a shipment on the rail lines having the highest traffic volume. Frequently traveled routes are preferred because they are generally well maintained since the railroad depends on these lines for a major portion of its revenue. In addition, routing along the high-traffic lines usually replicates railroad operational procedures.

The second aspect of rail routing is the selection of the sequence of railroads between the origin and destination. A delay is often involved in transferring a shipment from one railroad to another. While there are some run-through interchanges where the trains are not disassembled, the majority of interchanges require the incoming train to be disassembled and the cars to be sorted according to the receiving railroad. After the cars have been transferred to the receiving railroad's yard, they will be resorted according to destination and assembled into outgoing trains. In order to provide efficient service, the railroads try to reduce the time delays associated with interchanges by minimizing the number of interchanges in a route. This is accomplished in the INTERLINE model by imposing a numerical penalty for each interchange, which increases the

apparent length of the route. Thus, when the model attempts to minimize the length of the route, it will also minimize the number of transfers.

A characteristic of routing on U.S. railroads is the desire of each individual company to maximize its portion of the shipment's associated revenue. This normally results in the originating railroad transporting the shipment as far as possible on its system before transferring the shipment to another railroad. This feature is represented in the INTERLINE model by the originating railroad weighting factor in the routing algorithms. In evaluating the length of the route, the model treats 1 mile of travel on the originating railroad as being "less" than 1 mile on other railroads. This numerical adjustment increases the originating railroad's portion of the route.

1.2 INTERLINE RAILROAD NETWORK

The INTERLINE railroad network is essentially a computerized rail atlas describing the U.S. railroad system. All rail lines with the exception of industrial spurs are included in the network. Inland waterways and deep water routes along with their interchange points with the U.S. railroad system are also included in the network. The network contains more than 15,000 rail and barge segments (links) and over 13,000 stations, interchange points, ports, and other locations (nodes).

The rail network is a modification of a Federal Railroad Administration (FRA) data base that was obtained by ORNL in 1977.[4] This network has been extensively revised and is continually updated to reflect abandonments, company mergers, short line spin-offs, and new rail-line construction. The network is composed of 93 separate subnetworks, where each subnetwork represents a separate railroad system. Additional subnetworks have been included to represent the waterway networks and the National Railroad Passenger Corporation, AMTRAK. The waterway network is divided into two components. All inland and intracoastal waterways are included in a single subnetwork known as the Barge/Intracoastal System. Deep water routes along the Pacific coast, the Atlantic coast, the Gulf of Mexico, the St. Lawrence Seaway, and the Great Lakes are included in the Merchant Marine subnetwork. In this subnetwork, the Panama Canal connects the Pacific coast deep water routes with those in the Gulf of Mexico.

The network includes several characteristics for each of the rail segments (links). The most important characteristic from a routing standpoint is the mainline classification (MLC), which is a measure of the traffic volume on a particular link. The routing algorithm utilizes the

MLC in order to replicate actual railroad routing practices. The mainline classifications used in the INTERLINE routing model are as follows:

A-mainline—more than 20 million gross ton miles per year;
B-mainline—between 5 and 20 million gross ton miles per year;
A-branch line—between 1 and 5 million gross ton miles per year; and
B-branch line—less than 1 million gross ton miles per year.

Railroad companies tend to concentrate traffic on certain lines. These lines will have higher traffic densities than the less frequently used lines. Hence, basing route selection on the MLC will replicate railroad routing practices.

In addition to the mainline classification, other link characteristics include distance, line ownership, geographical coordinates of the end points, and population density.

1.2.1 INTERLINE Naming Conventions

Railroad stations, interchange points, and other locations in the INTERLINE data base are represented as nodes. All of the nodes in the data base are numbered, and a large proportion of the nodes are also named; therefore, it is rather unusual for the general user to encounter origins or destinations that are not named. A sample listing of a number of nodes in California is shown in Table 1. This table shows the master network node number, the node location, and the state (e.g., node number 14666 corresponds to San Bernardino, California, and node number 14745 corresponds to San Diego, California).

In general, the node names are completely spelled out. When an abbreviation is required, no punctuation is included in the node name; however, a comma separates the node name and state. For example, Saint Louis, Missouri, is listed as ST LOUIS, MO. Some other examples of unusual spelling are O FALLON for O'Fallon, Illinois, and MC COOK for McCook, Nebraska, and McCook, Illinois.

A number of the nodes in the INTERLINE data base represent barge ports and commercial nuclear plants. Barge locations are indicated by including a semicolon after the city name followed by the words PORT OF or PORT. As noted in Table 1, node 17293 is the port of San Diego, and node 16869 is the port of San Francisco. These nodes can be used as origins or destinations of barge routes or as interchange points between the barge and rail networks.

Nuclear plants are denoted by the letters NP in the node name. For example, node 14711 SAN ONOFRE NP,CA in Table 1 represents the San Onofre nuclear plant in southern California. All nuclear plants located on navigable waterways are assigned a barge node. (It should be noted

that the definition of such nodes does not necessarily imply that barge shipments could actually be made from those locations.) The dock at the San Onofre plant is called SAN ONOFRE NP DOCK, CA, node number 17290.

Table 1. Node naming convention

Node number	Node name	
	Location	State
14666	SAN BERNARDINO	CA
14745	SAN DIEGO	CA
17293	SAN DIEGO; PORT OF	CA
14682	SAN DIMAS	CA
14677	SAN FRANCISCO	CA
16868	SAN FRANCISCO BAY	CA
16869	SAN FRANCISCO; PORT	CA
14629	SAN JACINTO	CA
14539	SAN JOSE	CA
14536	SAN JOSE YARD	CA
14448	SAN LEANDRO	CA
16302	SAN LUCAS	CA
16313	SAN LUIS OBISPO	CA
14711	SAN ONOFRE NP	CA
17290	SAN ONOFRE NP DOCK	CA
14451	SAN PABLO	CA
16864	SAN PABLO BAY	CA
14441	SAN RAFAEL	CA
14738	SAN YSIDRO	CA

At different times, the INTERLINE model will request that the user specify a node, for example, to define the origin or destination of a route. The node may be specified by its name or by its unique number. Using the node name is the simplest way to specify a node location. Both the location and state must be entered, for example, SAN DIEGO,CA; however, the entire name does not have to be entered. The user can enter the first several letters of the name, a comma, and the standard two-letter abbreviation of the state. The model will list all locations in

that state whose name starts with the characters entered. The user will then be given the opportunity to select the proper node from the list supplied. The list of nodes shown in Table 1 was generated by simply entering SAN,CA.

Railroad systems are also named and assigned a unique number. In the INTERLINE model, railroad names are abbreviated. The Association of American Railroads abbreviations are used in the model. Examples of the railroad naming convention are shown in Table 2. For instance, the Chicago & North Western railroad is abbreviated as CNW and is defined as railroad number 3. A complete list of railroad names, abbreviations, and numbers is included in Appendix A.

Table 2. Railroad naming convention

Railroad	Abbreviation	Railroad number
Atchison, Topeka & Santa Fe	ATSF	1
Burlington Northern	BN	2
Chicago & North Western	CNW	3
Consolidated Rail Corporation	CR	4
CSX Transportation	CSXT	5
Florida East Coast	FEC	6
Grand Trunk Western	GTW	7
Southern Pacific	SP	12
Montana Rail Link	MRL	59
Terminal Railroad Association of St. Louis	TRRA	80

A special nomenclature is used for some short line railroads, some terminal railroads, and waterways. A number of the smaller short line railroads have been combined into a single system that is assigned subnetwork number 92 and abbreviated as <C3>. The <C3> notation is derived from the Interstate Commerce Commission Class I, II, and III railroad terminology. The casual user will probably not encounter these short lines when running the INTERLINE model. In a number of urban areas, there are one or more switching or terminal railroad companies that provide switching service between major railroad systems. For convenience, a number of the smaller transfer companies have been combined into a single classification, <TR>, and assigned a subnetwork number of 93.

Subnetwork 94 represents the AMTRAK intercity passenger railroad system. This subnetwork should not be used for the routing of rail freight and is included in the data base only for completeness. Routes generated with the AMTRAK subnetwork may not replicate the actual path taken by specific AMTRAK trains in areas where several AMTRAK routes parallel and cross each other.

Inland and coastal waterways are divided into two separate systems in the INTERLINE data base. The inland and intracoastal waterways are included as the Barge/Intracoastal Waterway system, which is abbreviated as *BRG and is subnetwork number 95. Hence, when trying to route from a barge port, the user will see *BRG as one of the available carriers (provided the barge routing option has been activated). INTERLINE also has the ability to route a shipment on ocean going barges. These deep water routes provide barge service along the Pacific and Atlantic coasts, the Gulf of Mexico, the Panama Canal, and the Great Lakes. All deep water routes are combined into a single classification, Merchant Marine (subnetwork 96 and is abbreviated as *M-M).

1.2.2 General Routing Considerations

Rail routes are calculated by minimizing the total impedance between the origin and the destination. The impedance is defined as a function of distance, mainline classification, and number of railroads involved in making the shipment. The INTERLINE program identifies the set of links between the origin and destination that minimizes the impedance:

$$L = Min \left\{ \sum_i (\sigma_i f_i d_i) + \sum_n (T_n) \right\}, \qquad (1)$$

where

L = impedance for route;
σ_i = railroad factor for link i;
- σ_i = 0.8 for the originating railroad
- σ_i = 1.0 for all other railroads

f_i = mainline classification factor for link i;
- f_i = 1.0 for A-mainline
- f_i = 1.2 for B-mainline
- f_i = 1.9 for A-branchlines
- f_i = 4.0 for B-branchlines

d_i = distance along link i, miles;
T_n = transfer penalty factor at node n.

1.2.2.1 Rail Routing

The routing algorithm shown in Eq. (1) preferentially routes a shipment along the A- and B-mainlines, while minimizing interchanges between railroad companies. In general, shipments will utilize only the A- and B-branchlines as a connection between the mainline network and the origin or destination. Frequently, a number of railroads will provide service at the same location. Selection of an originating railroad has a major impact on the estimated route because the originating railroad will preferentially attempt to move the shipments on its own system before interchanging with another railroad in order to maximize its portion of the revenue.

An example of a rail route is shown in Fig. 1. In this example, a route is calculated between Boston, Massachusetts and Los Angeles, California. Conrail was selected as the originating railroad for the shipment in Boston, and the Atchison, Topeka and Santa Fe (ATSF) was selected as the terminating railroad in Los Angeles. The route travels on the Conrail system through central Massachusetts and New York, northwestern Pennsylvania, northern Ohio and Indiana, and into Illinois. The shipment was transferred from Conrail to the ATSF at Streator, Illinois. The ATSF portion of the route passes through Illinois, the southeast tip of Iowa, northern Missouri, southeastern Kansas, northwestern Oklahoma, the panhandle of Texas, northern New Mexico and Arizona, and finally southern California. This route is 3293 miles long.

1.2.2.2 Alternative Routing

The alternative routing capability is a useful tool for estimating a number of different routes between the same origin and destination. The alternative routing algorithms will always generate different routes. However, some of the alternative routes may display only minor differences. Hence, it is necessary to review the results of the alternative routing calculations and reject alternatives that do not display significant differences.

A number of alternative routes between Charleston, South Carolina, and Las Vegas, Nevada, are shown in Fig. 2. The base route, which is the normal route calculated by the INTERLINE model, travels on two major systems. This northernmost route originates on the Norfolk Southern (NS) system and travels to Kansas City, Missouri, by way of North Carolina, Tennessee, and Kentucky. At Kansas City, the base route transfers to the Union Pacific (UP) system and travels through Kansas, Nebraska, Wyoming, and Utah before reaching Las Vegas. The base route is 2844 miles long.

ORNL-DWG 93-5610

Fig. 1. Rail route between Boston, Massachusetts, and Los Angeles, California.

ORNL-DWG 93-5611

Fig. 2. Alternative rail routes between Charleston, South Carolina, and Las Vegas, Nevada.

The first alternative route travels on two major railroad systems and one terminal railroad. This route originates on the CSX Transportation (CSXT) system and passes through Georgia, central Tennessee, and western Kentucky before reaching St. Louis, where the route is then transferred via a terminal railroad to the UP system. The UP segments of both the base route and the first alternative route west of Kansas City are very similar, except for a deviation in the vicinity of Salt Lake City, Utah. This route is 2882 miles long, approximately 38 miles longer than the base route.

The second alternative route, which is the southernmost route, uses three major railroads to move the shipment from Charleston to Las Vegas. This route originates on the CSXT, transfers to the Southern Pacific (SP) in New Orleans, and makes a final transfer to the UP system at San Bernardino, California. This route is 3015 miles long.

The third alternative route, which is 3170 miles long, also utilizes three different major railroads; the NS, the ATSF, and the UP. Interchanges for this route take place at Kansas City, Missouri, and Daggett, California.

The fourth and final route shown in Fig. 2 is shorter than either the second or third alternative routes—2952 miles as compared with 3015 or 3170 miles. However, the fourth alternative route uses a significantly higher amount of B-mainline track than the other routes which run primarily along A-mainlines. Three major railroads and one terminal railroad are involved in the fourth alternative route. These are the CSXT, the SP, the UP, and a terminal railroad in St. Louis. Two transfers take place in the vicinity of St. Louis, Missouri (CSXT to the SP via a terminal railroad), and the final transfer to the UP system takes place in Provo, Utah.

1.2.2.3 Barge Routing

Barge routes may be calculated by using the inland waterway and deep water subnetworks that are included in the INTERLINE data base. If it is not possible to complete the entire route by water, the INTERLINE program will automatically determine a rail/barge interchange point.

The inland waterway and deep water networks are not automatically turned on when the model is first activated. Initially, when the INTERLINE routing model was created, only the railroad network was included in the data base. As the model went through several enhancements, it became apparent that there was a need to include the inland and coastal waterways and deep water routes. With the waterways option activated, it was noticed that some

routes between rail nodes actually utilized the waterway for part of the route. Hence, it was necessary to carefully review all rail routes to ensure that the waterways were not inadvertently included. In order to simplify operation, INTERLINE automatically activates only the railroad systems. When barge routes are desired, these networks must be activated manually.

An example of barge routing is shown in Fig. 3. Two routes are included in this example to illustrate different capabilities of the model. The first is an all water route between the port at Knoxville, Tennessee, and the port at New Orleans, Louisiana. The estimated route follows the Tennessee River, Tennessee-Tombigbee Waterway, Mobile River, and Gulf Intracoastal Waterway, for a distance of 1037 miles. While there are a number or railroad companies connecting these locations, the lower impedance associated with waterways and the high transfer penalties associated with barge-to-rail transfers resulted in the selection of a route that remained on the inland waterway system for the entire distance. If both end points of the route are on waterway networks, the entire route will remain on the waterway networks.

A second example, included in Fig. 3, is a route originating at the port in Knoxville, Tennessee, and terminating in Helena, Montana. Since Helena is not located on a navigable waterway, this route must transfer to the railroad network at some intermediate point. As shown in Fig. 3, the route follows the waterways from Knoxville, Tennessee, to St. Paul, Minnesota, where it transfers to the Burlington Northern (BN) railroad. The route then remains on the railroad system from St. Paul to Helena. The total distance of this route is 2572 miles: 1458 miles along the waterway system and 1114 miles on the BN and Montana Rail Link (MRL) railroads.

1.2.2.4 Other Routing Considerations

In general, the standard routing capabilities described above will be sufficient for most applications. However, the user may constrain a route from passing through a particular location or from using a particular railroad or waterway system.

It may be necessary to temporarily remove a link or node from the network to simulate the closure of a segment of a rail line. Such closures could be caused by a washout, flooding, or other natural phenomena.

Link removal may be simulated by defining a very large impedance for the link. The routing algorithm will then tend to select other links to connect the origin and destination.

ORNL-DWG 93-5612

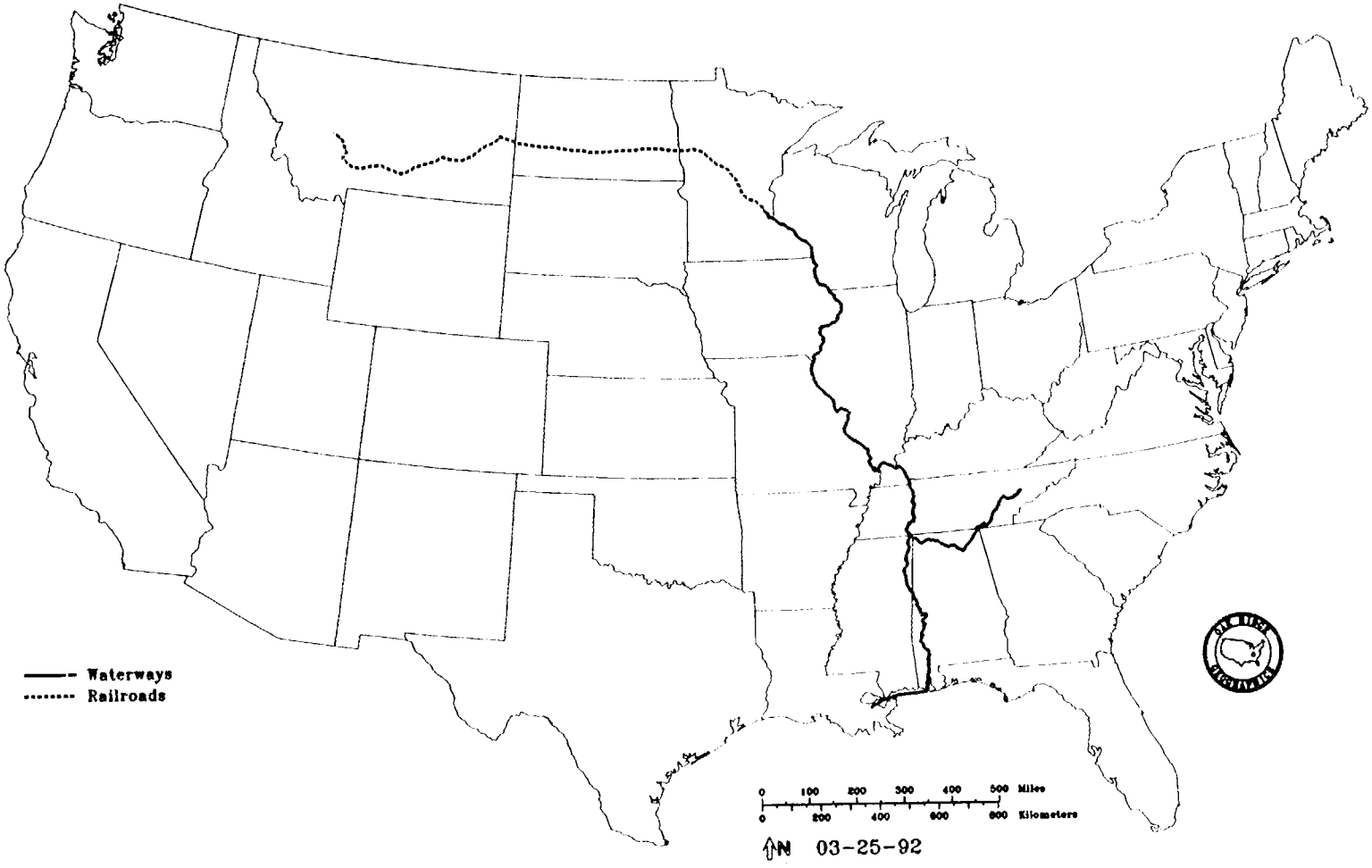


Fig. 3. Example of barge routes.

Placing a high impedance on a link does not guarantee that the "removed" link will not be used in a route. A "removed" link will still be included in the route if it is the only possible route. However, the route listing will contain a message indicating that a "removed" link was included in the route.

A special case of link removal is the program's ability to permit traffic to flow in only one direction on a link. This feature is useful in simulating the sharing of parallel rail lines by different railroad companies. In some instances, westbound traffic uses one line, and eastbound traffic uses a nearby parallel line.

The user may also ask the program to avoid a node or a series of nodes. Avoiding nodes is somewhat more complicated because more than one railroad may provide service to those nodes. In INTERLINE's internal structure, a node is defined as a combination of a location and a railroad. The user usually interprets a node as a spatial location (i.e., a location and all railroads entering or leaving that location) and assumes that once the node is removed it cannot be used. Hence, the use of all railroads passing through the node must be avoided.

A third routing constraint that can be imposed by the user is the removal of a railroad system. In this case, routes will not utilize any of the links on the removed system. Three subnetworks included in the INTERLINE data base, AMTRAK and the two waterway networks, are not activated when the model starts operation. The INTERLINE model was created to model freight movements throughout the country, and all railroad systems that transport freight are automatically activated at the start of operation. Section 3.3.7 in the User's Manual contains instructions for activating the AMTRAK and waterway subnetworks if needed for a particular application. Detailed instructions for removing or reactivating links, nodes, and railroad systems are included in the User's Manual (Sects. 3.3.5, 3.3.6, and 3.3.7).

Other routing constraints that can be imposed by the advanced user are altering the MLC factors, altering the originating railroad factor, altering the transfer penalties, and defining the sequence of railroads to be used. However, to use these capabilities effectively, the user needs detailed knowledge of the U.S. railroad system and the mathematical structure of the INTERLINE model. Many of these operations are considered advanced concepts and are discussed in Appendix B.

1.2.2.5 Population Density Estimates

INTERLINE can generate a table of population densities for any route calculated. The population data used for this calculation are based on the U.S. Bureau of Census block group data. The population density levels reported were designed to be comparable with the RADTRAN risk analysis code.[5] For details in obtaining population density data, please refer to the User's Manual (Sect. 3.3.4.3).

2. METHODOLOGY

A description of the shortest path techniques used to calculate routes in the INTERLINE model is presented in this chapter. A basic understanding of the algorithms performing the various routing calculations will help the user to more effectively and accurately use the program. The discussion is aimed at the general reader and uses narrative along with a number of simple examples to illustrate the calculational techniques. A number of references are included that will supply further information for the user interested in the mathematical formulations of the shortest path technique.

The discussion starts with a description of the U.S. railroad system and illustrates how the large network is subdivided into a number of subnetworks. A general overview of the shortest path algorithm is presented in Sect. 2.2.3. The next several sections show how the generalized shortest path algorithm has been modified in the INTERLINE model to handle decomposed networks. This technique is illustrated in Sect. 2.2.5 by showing a calculation for a route that spans four interconnected networks. More complicated examples are included in the later sections of this chapter. These examples include an origin with multiple railroads and alternative routing.

2.1 ORGANIZATION OF THE NETWORK FOR ROUTING ANALYSIS

In its basic structure, the railroad network model is essentially a data base. To be used for route analysis, this data base must be processed separated or decomposed, into a series of subnetworks representing individual railroad companies, which are connected at transfer locations. Finally, the network is placed in a form convenient for the routing algorithms.

2.1.1 Railroad Company Representation

The networks for the individual railroad companies are developed from the information in the network data base. This is done in a rather straightforward fashion by examining link ownership and usage rights and extracting all links which a carrier is authorized to use. This includes both links that the carrier owns and those links upon which the carrier has trackage rights.

From the link list, the set of nodes that the railroad's network serves is determined by scanning the end points for each link. The numbers assigned to these nodes are the numbers from the original FRA data base and are not in a particular sequence. (NOTE: In this document, the original node numbers will be referred to as FRA node numbers.) The routing algorithm requires that the nodes in a subnetwork be numbered in sequence. Hence, the nodes in a railroad subnetwork are automatically assigned a new sequential number, starting at 1.

Figure 4 graphically illustrates this decomposition process. Figure 4(a) shows the combined network where the dashed and solid lines indicate ownership by two different railroad companies. For this example, the nodes are arbitrarily numbered. Nodes 9946 and 11211 are served by links of both railroads (i.e., they are transfer points). Figure 4(b) shows the network decomposed into the two separate companies. Nodes in each company network have been assigned sequential numbers. The table accompanying each subnetwork shows the correspondence between the new sequential subnetwork node numbers and the original FRA node numbers.

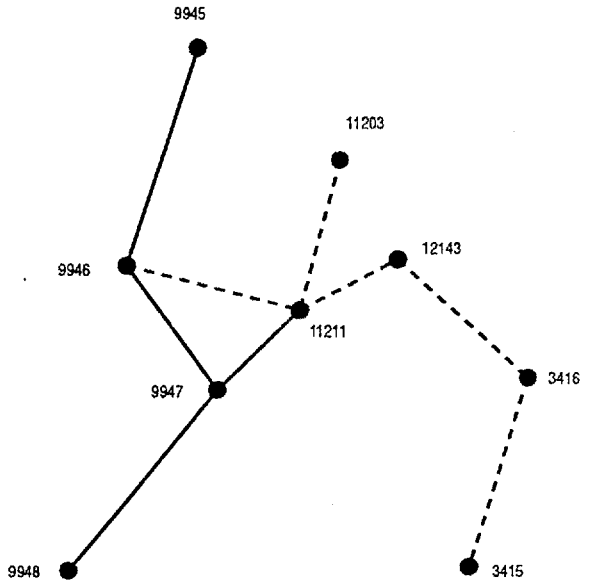
2.1.2 Transfer Connections

Interline transfer locations are incorporated into the network model as a special type of link. These links connect the individual railroad subnetworks and allow routes to cross subnetwork boundaries. Figure 5 shows transfer links installed to connect the two railroad company subnetworks shown in Fig. 4.

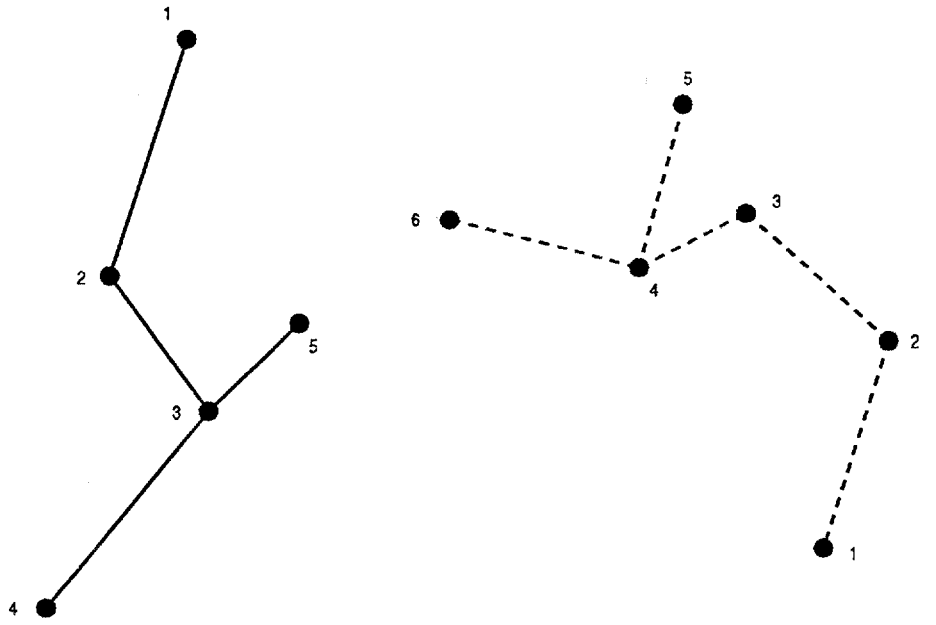
Transfer links have an associated value that represents the attractiveness of moving between two subnetworks at that location. A low-value transfer location is highly attractive. High value indicates that a transfer is discouraged. These values are set to reflect the individual railroad's preferences for interline activity. Note that transfer links are directional. Transfers from Company A to Company B may have a different value than transfers from B to A.

Once the network has been decomposed, it is automatically organized and processed to make it efficiently accessible for the routing software. The details of this process are not significant to the routing methodology and will not be discussed in this document.

ORNL-DWG-93-7639



(a)



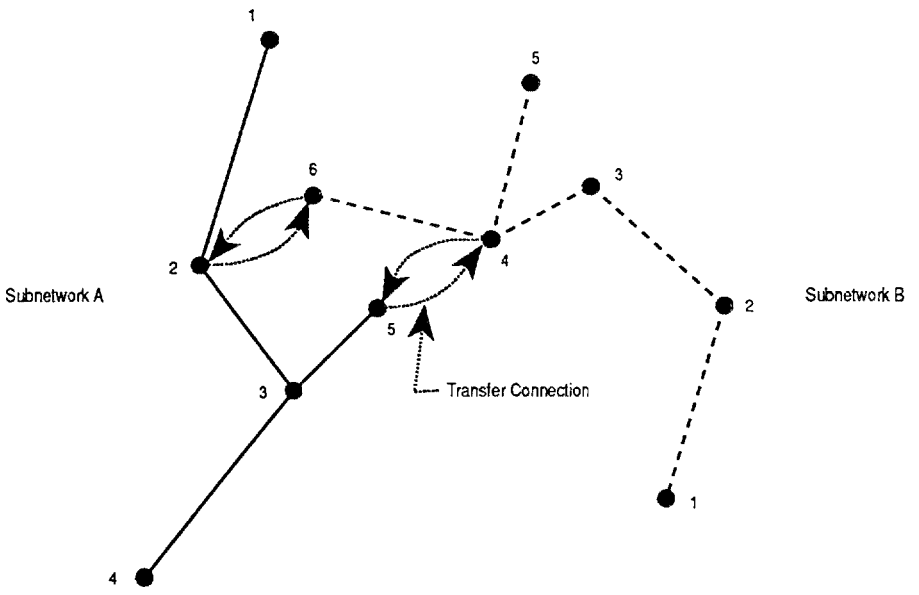
RR Node	FRA Node
1	9945
2	9946
3	9947
4	9948
5	11211

(b)

RR Node	FRA Node
1	3415
2	3416
3	12143
4	11211
5	11203
6	9946

Fig. 4. Graphic illustration of (a) an undecomposed network and (b) a decomposed network.

ORNL-DWG-93-7640



B RR Node	A RR Node	Impedance
4	5	80
6	2	40

A RR Node	B RR Node	Impedance
5	4	80
2	6	40

Fig. 5. Interline transfer link model.

2.2 BASIC ROUTING METHODOLOGY

This section presents the logic used in the INTERLINE model to estimate routes. The general procedure is based upon a mathematical technique called a shortest path algorithm. However, this algorithm has been modified to handle decomposed interconnected networks. The following subsections describe the principles employed by the algorithm. The user is referred to refs. 6 and 7 for a mathematical derivation of the shortest path algorithms.

2.2.1 Routing Nomenclature

To understand the subsequent material, the reader should be acquainted with some terms and concepts that will be employed. A discussion of these terms follows.

Simply stated, a route, or path, is a sequence of adjacent links. If a link is defined by its end points (nodes n_a and n_b), a path can be represented as the series of connected links (n_1, n_2), (n_2, n_3), (n_3, n_4), . . . (n_{k-1}, n_k). Node n_1 in the path is the origin, and node n_k is the destination. All other nodes represent connection points between links in the path. Each node in a typical path is unique (appears only one time), which means that the path does not cross itself.

Each link in a network has an associated impedance that is incurred for using the link in a path. This impedance value may be of interest to the analyst. Typical values could include distance, travel time, and monetary cost. Impedances may not assume negative values.

The cost of traveling between two nodes is the sum of the impedances of the links in a path connecting the nodes. Given the set of all possible paths between two nodes, the algorithm finds the path with the minimum impedance.

The INTERLINE routing algorithm (which is normally called the shortest path algorithm) takes a network that has a number of interconnections (like a web) and creates a minimum impedance tree, which is the collection of the minimum impedance paths from a single origin to all other nodes in a network. The term "tree" is an analogy. The origin node is the "root." Destination nodes are located on the leaves of the tree. The individual paths from the destination nodes join to form branches that lead to the root node.

2.2.2 The Routing Objective Function

The objective function is the mathematical expression that the routing algorithm uses to evaluate the minimum impedance or shortest path. INTERLINE produces routes consisting of line segments and transfer links according to the following objective:

$$L = Min \left\{ \sum_i (\sigma_i f_i d_i) + \sum_n (T_n) \right\}, \tag{2}$$

where

L = impedance for route;
σ_i = originating railroad factor for link i;
f_i = mainline classification (MLC) factor for link i;
d_i = distance along link i, miles;
T_n = transfer penalty factor at node n.

The distance of the link is the basic impedance value. In the real world, however, a mile of travel on a segment of well-maintained mainline track is more likely than a mile on lower quality branchlines, since railroads tend to concentrate traffic on the highest class portions of their networks. To reflect this behavior, the MLC factors are assigned values, the defaults used in the INTERLINE model are itemized below:

f_i = 1.0 for A-mainline,
f_i = 1.2 for B-mainline,
f_i = 1.9 for A-branchlines, and
f_i = 4.0 for B-branchlines.

The length of each line segment is multiplied by the appropriate MLC factor to determine the link impedance. A mile of B-branchline would have an impedance of 4, while the corresponding distance along an A-mainline segment would only have an impedance of 1. The reader will see that the model attempts, wherever possible, to maximize travel on the higher MLC lines. Only where the circuity of high MLC lines become excessive will a lesser MLC line be selected.

Waterway segments also have classification factors. The default values are as follows:

f_i = 0.7 for inland waterway segments, and
f_i = 0.6 for deep water route segments.

The originating railroad factor, σ_i in Eq. (2), reduces the apparent impedance of links in the originating railroad subnetwork to 80% of their initial value; that is, the default values are

σ_i = 0.8 for the originating railroad, and
σ_i = 1.0 for all other railroads.

The algorithm will then tend to increase the portion of the route using the originating railroad, since its impedance per mile of equivalent MLC class is lower than those of other railroad subnetworks.

The default values for the standard impedance value for a transfer is 300 units, equivalent to 300 miles of travel on an A-mainline. Transfers to terminal railroads receive an impedance of 151 units, so that these railroads may be used as connections between two Class I railroads without significant penalty. (The impedance of the two transfers involved will be 302 units, slightly more than associated with a direct Class I to Class I transfer.) Impedance factors for transfers between railroad companies under a common corporate umbrella have also been reduced to reflect operational coordination between the subsidiary companies.

Transfers between the railroads and the waterway networks have an impedance of 900 units. With a penalty of this magnitude, routes are unlikely to change modes unless the origin and destination are located on differing modes.

2.2.3 Shortest Path Algorithm Overview

In its basic form, the shortest path algorithm generates a path tree between an origin node and other nodes in a network. The logic behind this procedure is not complex, and a short discussion of it will prepare the user for the discussion of the complete routing algorithm in Sect. 2.2.4.

Within a subnetwork, each node has the following attributes that are used by the algorithm:

- a label that contains the lowest impedance currently computed for a path from the origin to that node;
- a status flag, which is TRUE when the label represents the impedance of the shortest path from the origin and FALSE otherwise; and
- a predecessor node in the path back towards the origin.

At the start of the algorithm, the program:

- sets the label of the origin node to 0,
- sets the status of the origin node to TRUE,
- places the source node in a pool of candidate nodes to be examined for expansion of the path tree on subsequent iterations of the algorithm,

- sets the labels of all other nodes to infinity, and
- sets the status of all other nodes to FALSE.

The main portion of the algorithm procedure is described below:

1. Examine the pool of candidate nodes. If the pool is empty, the tree is complete, so exit the procedure. Otherwise, pick the node with the lowest impedance, and call it the pivot node.

2. Make the pivot node's status TRUE to indicate that its path from the origin is the shortest; that is, no other path having a lower impedance exists to this node. Remove the pivot node from the pool of candidate nodes.

3. Perform the following steps for every node having a status of FALSE at the end of a link emanating from the pivot node:

    a. Compute a candidate impedance for the node by adding the impedance of the pivot and the impedance of the link.

    b. If the current impedance of the node is greater than the candidate impedance, set the node's impedance equal to the candidate impedance and designate the pivot node as the predecessor node.

    c. If the node is not in the candidate pool, add it.

4. Return to the first step.

Figure 6 provides a graphic demonstration of the shortest path algorithm. In Fig. 6(a), the links and nodes of an sample network are shown. The numbers beside the links represent the impedances. Figures 6(b)-6(h) show the successive stages of the algorithm for an origin at node 1. The node number within two concentric circles represents the pivot node in each step. A node's label is shown adjacent to its circled number. Labels in bold are permanent (status of TRUE), while other labels are temporary. Links in the path tree at each stage are shown as solid lines. Since the reader can directly visualize the route, predecessor node numbers are not displayed.

2.2.4 The INTERLINE Routing Algorithm

INTERLINE uses a modification of the shortest path algorithm to find the best route through a series of subnetworks. The algorithm must route within a subnetwork and decide how to move between subnetworks. To do this, a two-stage approach is used. The top-level

ORNL-DWG-93-7641

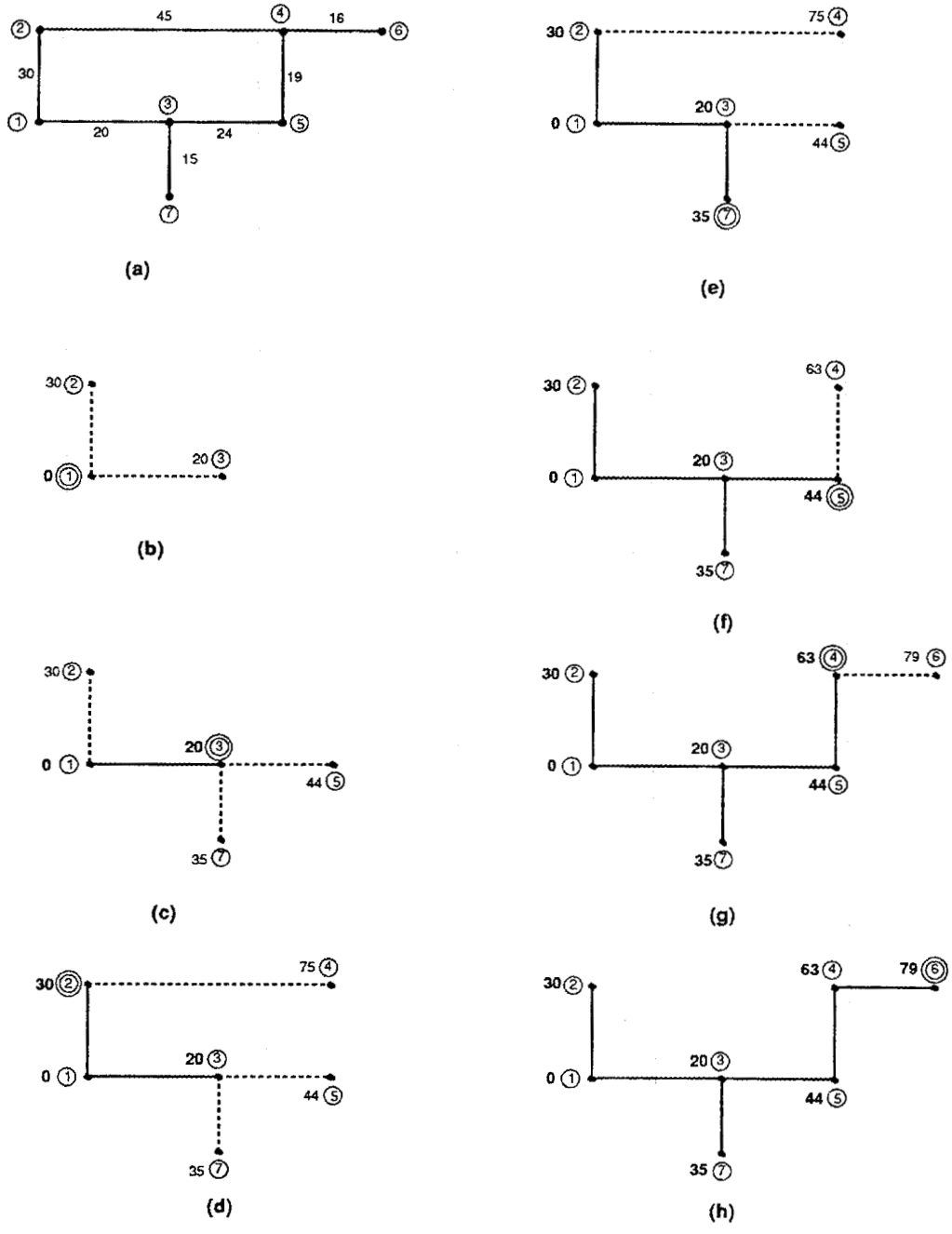


Fig. 6. Shortest path algorithm example: (a) base network, (b) step 1, (c) step 2, (d) step 3, (e) step 4, (f) step 5, (g) step 6, and (h) step 7.

procedure decides which subnetwork should be examined in detail to extend the path tree from the origin. The lower-level procedure then extends the tree within the selected subnetwork using a traditional shortest path approach. The behavior of the algorithm is described below.

Conceptually, the upper-level procedure treats subnetworks and transfer connections like nodes and links respectively. These entities are given additional attributes to reflect this, as explained below.

- Each subnetwork has a label that reflects the minimum impedance of all paths reaching the subnetwork from the origin node.
- Each transfer link has a label that is used in transferring label values for transfer nodes between subnetworks. This label should not be confused with the transfer impedance, which represents the actual impedance of a transfer operation.

For tracing the paths, individual nodes must have both a predecessor subnetwork identifier and a predecessor node number. A number alone will not uniquely identify a node, since all subnetworks have sequentially numbered nodes, starting at 1.

2.2.4.1 Initialization Sequence

To start the routing procedure, the following operations are performed.

- The labels of all nodes in all subnetworks are set to infinity.
- The status of all nodes in all subnetworks is set FALSE.
- The label of each subnetwork is set to infinity.
- The label of each transfer link is set to infinity.
- The label of the source node in the source subnetwork is set to 0.
- The status of the source node in the source subnetwork is set TRUE.
- The label of the source subnetwork is set to 0.

These actions will ensure that the main body of the algorithm will begin the route at the origin node in the origin subnetwork.

2.2.4.2 Main Loop

The body of the algorithm consists of the following steps.

1. Select the subnetwork having the lowest label, and treat this as the pivot. Initially, the origin railroad will be selected since its label is 0. The path tree will be extended within this pivot subnetwork.

2. Consider the following conditions:

   a. If the label of the pivot subnetwork is greater than or equal to the label of the destination node, then the route is complete, so stop.

   b. If no pivot subnetwork is selected, no direct path to the destination exists, so stop.

3. Examine all transfer connections into the pivot subnetwork. If the label of a transfer link is less than the label of its end node in the pivot subnetwork, do the following.

   a. Make the label of the transfer node in the pivot equal to the label of the transfer link. The path from the origin via the connecting railroad is shorter than the current path.

   b. Set the predecessor subnetwork of the transfer node in the pivot equal to the identifier of the subnetwork from which the transfer originates.

   c. Set the predecessor node of the transfer node in the pivot equal to the node in the subnetwork from which the transfer originates.

   d. Set the status of the transfer node to TRUE. This indicates that it should be included in the list of candidate pivot nodes in step 4 below.

4. Extend the path tree within the pivot subnetwork using a shortest path algorithm. The procedure starts with the first node in the candidate pool of the subnetwork. Link impedances in the originating subnetwork will receive the originating railroad factor.

5. Set the label of the pivot subnetwork to infinity so that it will not be reselected in the next iteration.

6. Examine all transfer connections **into** the pivot subnetwork again. Set the label of each transfer link equal to the label of its end node in the pivot subnetwork.

7. Examine all transfer connections **from** the pivot network to surrounding networks. Compute a candidate impedance for each transfer by adding the impedance of the end node in the pivot network to the impedance of the transfer link.

   a. If the candidate impedance is less than the transfer link's label, then the label is set to the candidate impedance.

   b. If the candidate impedance is less than the transfer subnetwork's label, the label is set to the candidate impedance.

8. Return to the first step.

The reader is urged to review this sequence of steps carefully because the logic may not be intuitively obvious.

Once path generation is complete, the route must be traced backward from the destination to the origin. Starting with the destination, the path is followed by examining the predecessor node and predecessor subnetwork for each node until the origin is reached.

2.2.5 Example of a Route from an Origin Served by a Single Railroad

To demonstrate the algorithm, the sample network of Fig. 7 is presented. The overall network consists of four subnetworks connected at transfer points. The circled numbers identify nodes within each subnetwork. Impedance values, which already include appropriate factors, are provided beside each link. Transfer links, indicated by dashed lines with arrowheads, have an impedance of five units.

The route of interest lies between node 1 in subnetwork 1 and node 2 in subnetwork 4. As the figure shows, there are a number of possible routes, and the shortest route is not immediately obvious.

Following the initialization procedures, the labels of subnetwork 1 and node 1 in subnetwork 1 are set to 0. Node 1 is placed in the candidate pivot pool for subnetwork 1. All other transfer labels, subnetwork labels, and node labels are set to infinity. To begin the first iteration, subnetwork 1 is selected as the pivot because it has the smallest label.

Figure 8(a) shows the initial conditions at the beginning of the pivot operation for subnetwork 1. Figure 8(b) shows the label values and path tree at the conclusion of step 7 of the algorithm. The path tree from the origin is indicated by solid links. Nodes 3 and 4 of subnetwork 1 connect to subnetwork 2. The minimum impedance path to subnetwork 2 is 19 units via node 4 in subnetwork 1. This value (19 units) is assigned to the subnetwork label for subnetwork 2. Likewise, nodes 4 and 5 of subnetwork 1 connect to subnetwork 3, and the minimum impedance to this network is also 19 units for a path passing through node 4 in subnetwork 1. The updated subnetwork labels indicate that either subnetwork 2 or subnetwork 3 could be selected for the next pivot because they have the same values. The selection of subnetwork 2 is arbitrary.

Figure 9 shows the start and end conditions for the pivot involving subnetwork 2. Note that the calculations continue from both nodes 1 and 5 of this network and the path extension

ORNL-DWG-93-7642

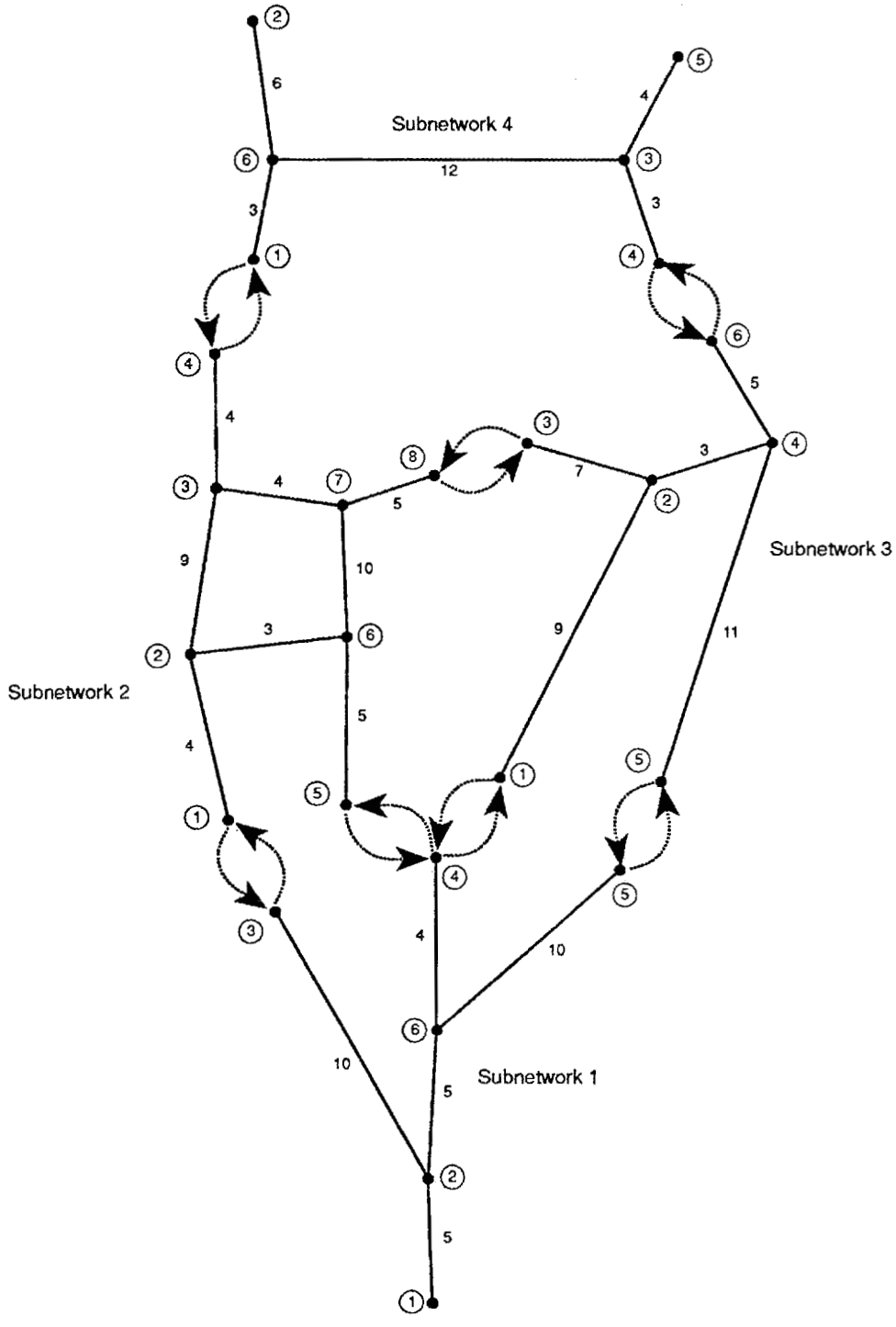


Fig. 7. Example of network structure.

ORNL-DWG-93-7643

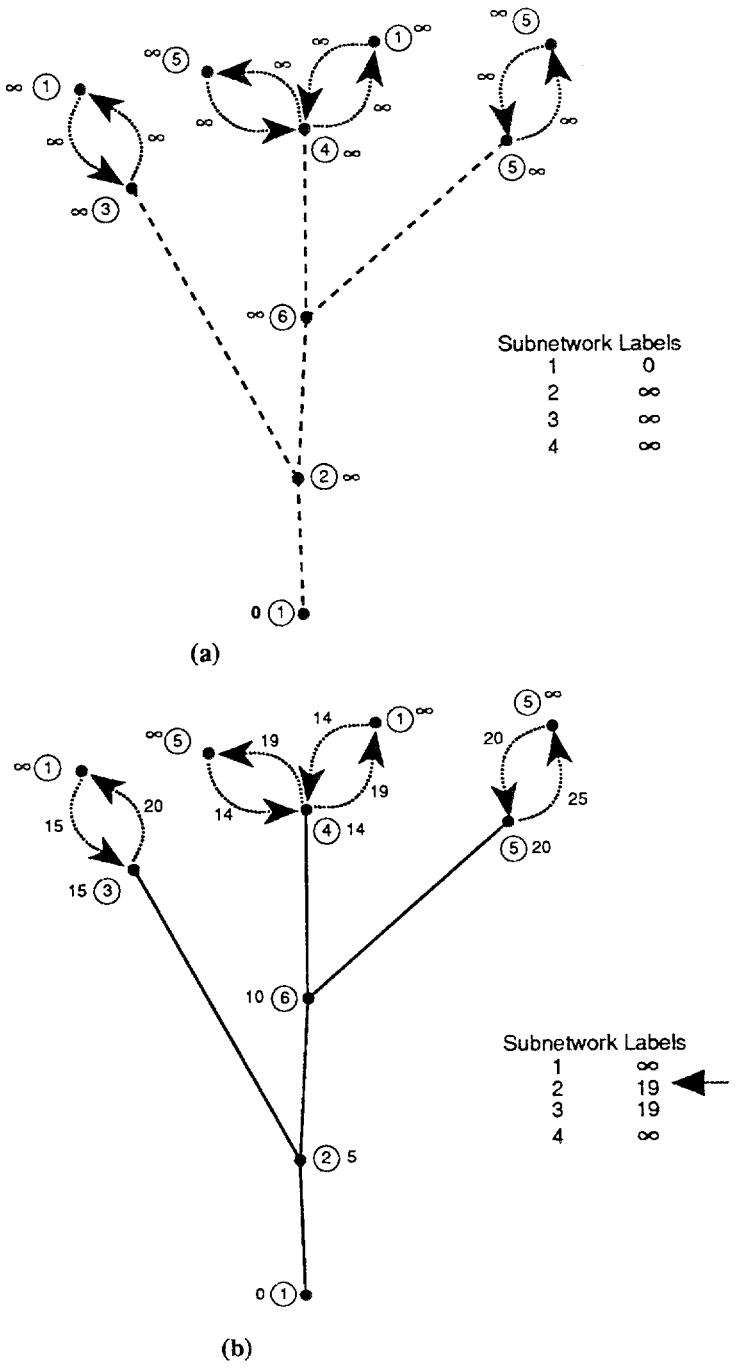


Fig. 8. Pivot through subnetwork 1 at (a) the start of and (b) the end of iteration.

ORNL-DWG-93-7644

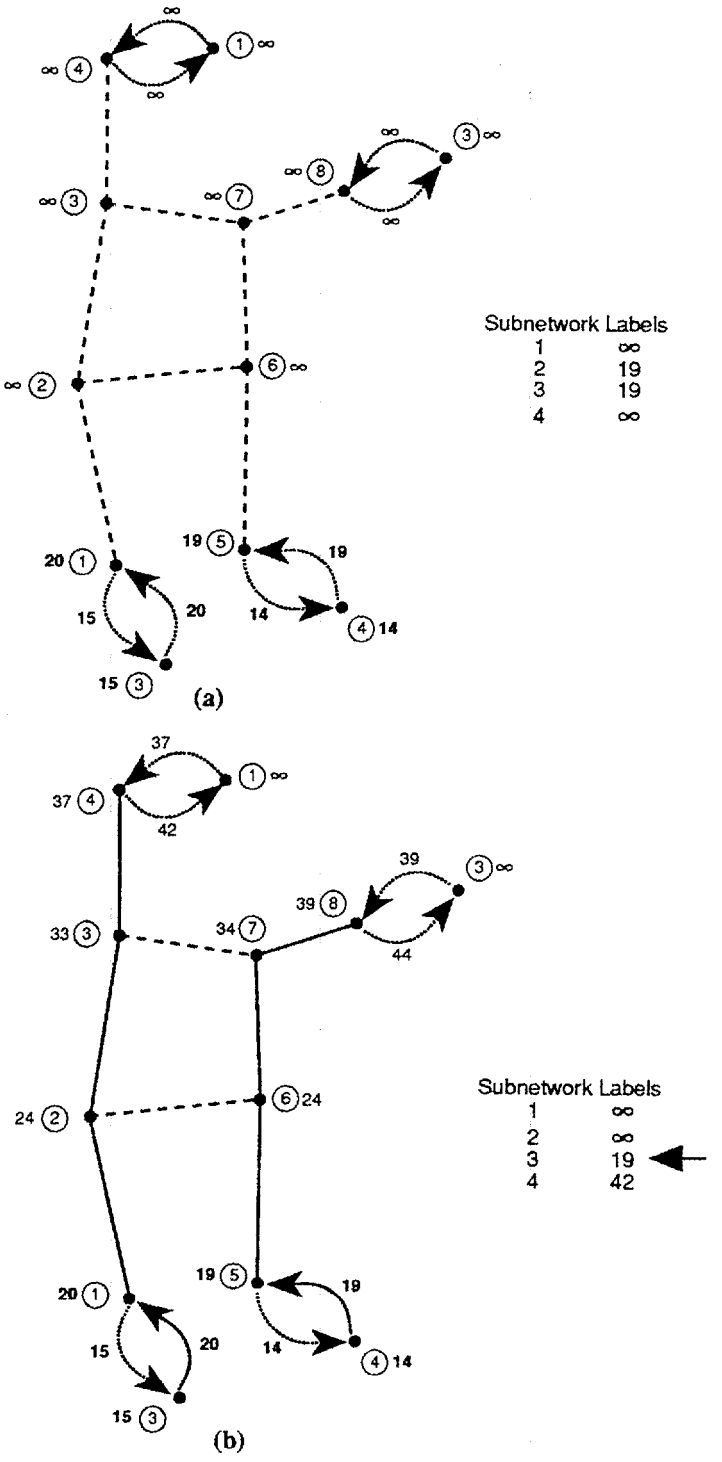


Fig. 9. Pivot through subnetwork 2 at (a) the start of and (b) the end of iteration.

generated **does not form a tree** within this subnetwork. The paths shown in Fig. 9(b) are an extension of the tree rooted at the origin in subnetwork 1. The reader should trace through the algorithm steps carefully to determine how the label values are derived. Subnetwork 3 will be the next pivot.

Figure 10 continues the process by expanding the path tree through subnetwork 3. Observe the following alteration of a tentative path. Node 3 receives a tentative label of 44 units in step 3 of the iteration through a transfer from node 8 of subnetwork 2. Step 4 finds a shorter path of 35 units via node 2. The transfer is not used in the tree.

Figure 11 completes the algorithm by examining subnetwork 4. The final impedance of the optimal (or base) path from the origin to node 2 is 51 units. No further iterations are required since the label of node 2 is smaller than the label of any subnetwork.

Figure 12 shows the complete path tree through the four subnetworks, along with the transfer links used. Dashed lines indicate links not included in the tree. As noted earlier, once the path generation is complete, the route is traced backward from the destination to the origin.

2.2.6 Example of a Route from an Origin Served by Multiple Railroads

Many points in the United States have multiple railroads providing service. When a user selects an origin node with more than one railroad providing service, the user has the option of selecting one or more of these railroads for originating the route. The INTERLINE model will build the shortest path from the origin node within each of the selected railroads. The resultant route will be the path with the minimum impedance.

The sample network of Fig. 7 will be used to demonstrate this feature of the algorithm. Impedance values for each link already include the appropriate factors.

The origin for this route will be the node served by three railroads. This point is represented as a node in three subnetworks—node 4 in subnetwork 1, node 5 in subnetwork 2, and node 1 in subnetwork 3. All three subnetworks will be considered as originating railroads. The destination for this example will be node 2 in subnetwork 4.

Following the initialization procedures, the labels of subnetworks 1, node 4; subnetwork 2, node 5; and subnetwork 3, node 1, are all set to 0. All other transfer labels, subnetwork labels, and node labels are set to infinity. To begin the first iteration, the model will arbitrarily select either subnetwork 1, 2, or 3, since all three have the identical subnetwork label of 0.

ORNL-DWG-93-7645

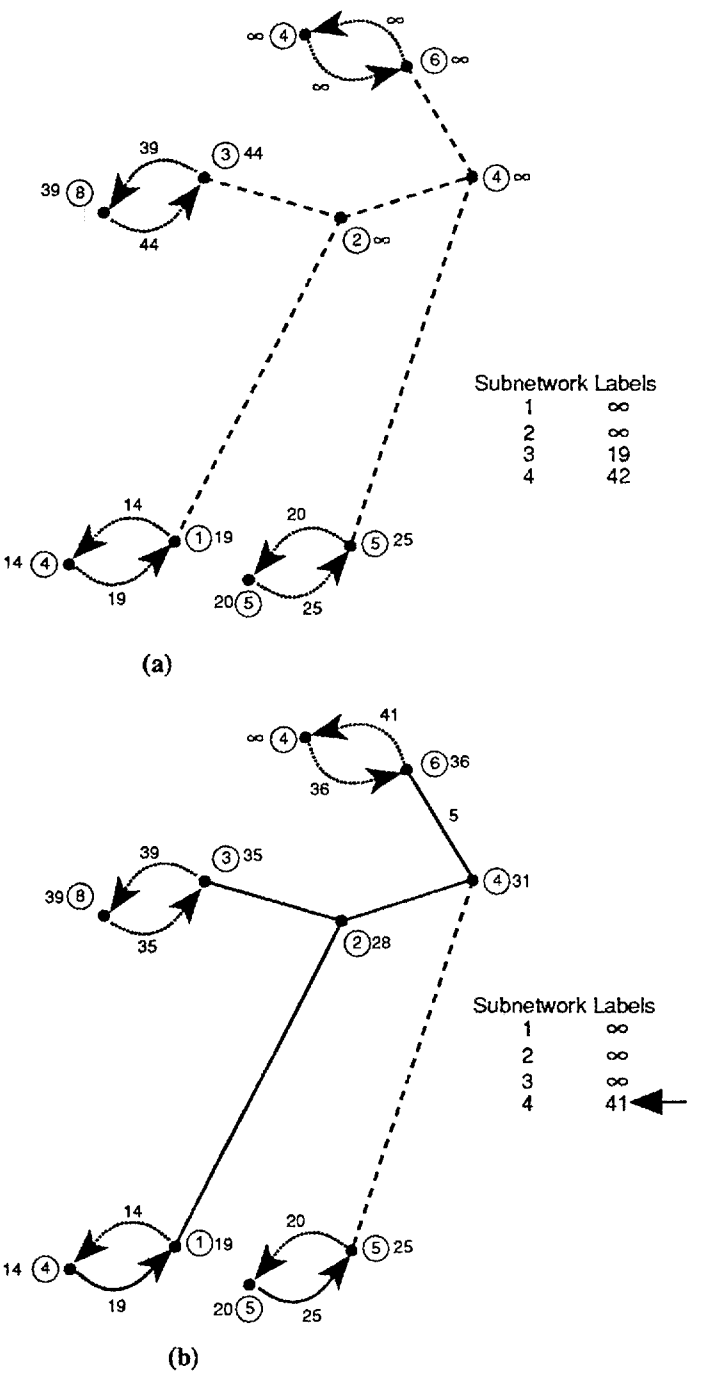


Fig. 10. Pivot through subnetwork 3 at (a) the start of and (b) the end of iteration.

ORNL-DWG-93-7646

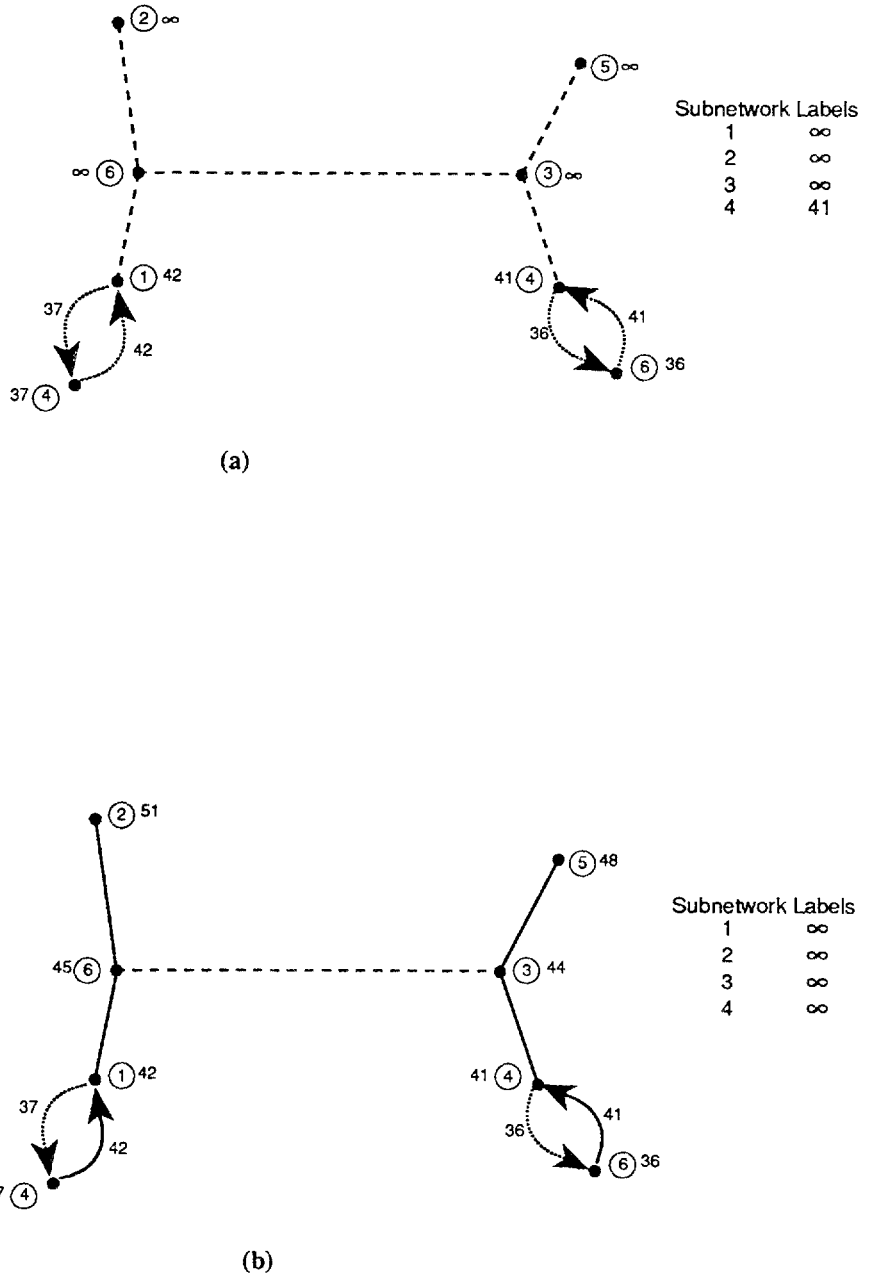


Fig. 11. Pivot through subnetwork 4 at (a) the start of and (b) the end of iteration.

ORNL-DWG-93-7647

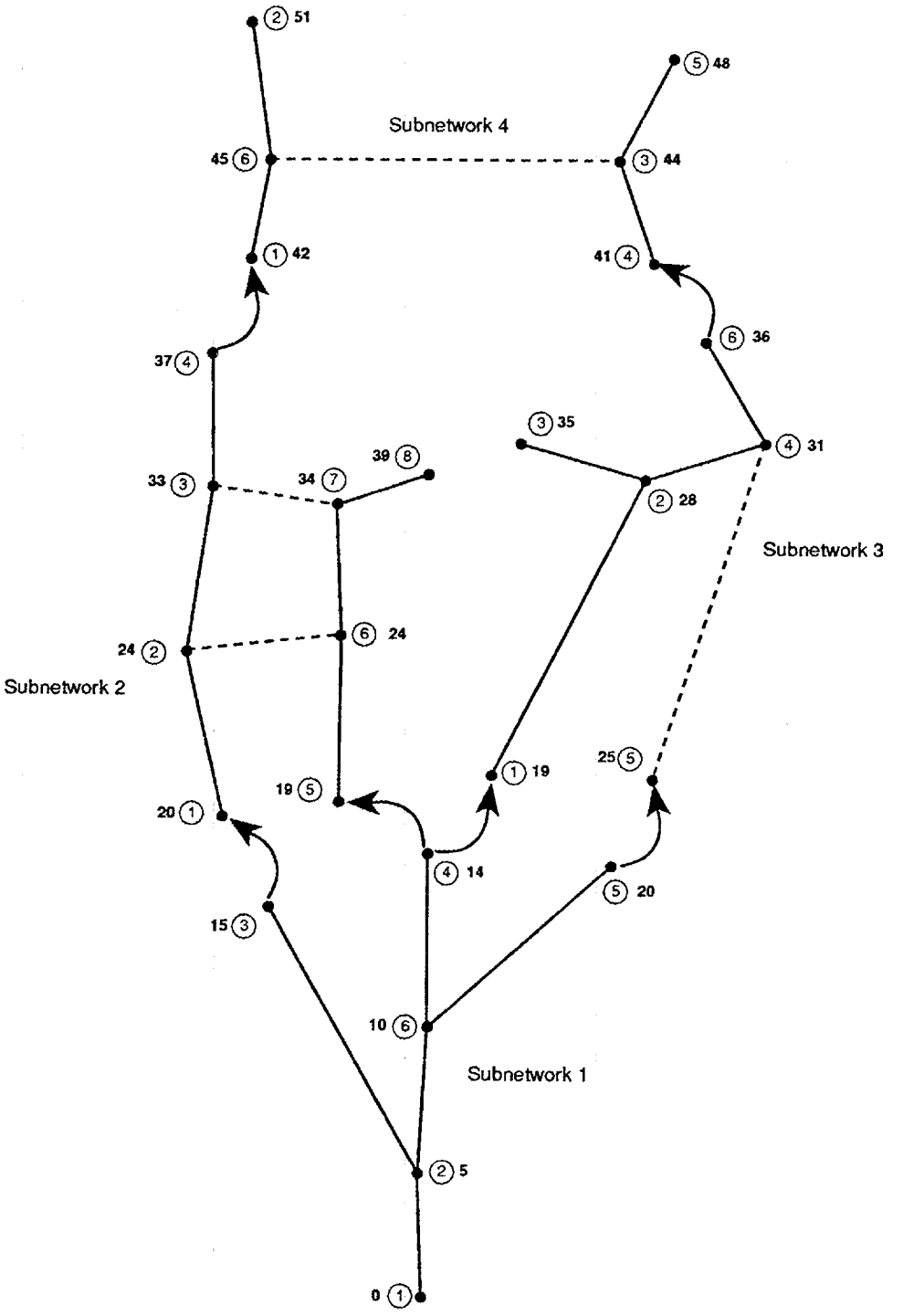


Fig. 12. Complete path tree.

As described in Sect. 2.2.5, the model will progress through the subnetworks until the shortest path has been determined. Figure 13 shows the complete path tree through the four subnetworks along with the transfer links used. Dashed lines indicate links not included in the tree. As seen on the figure, the origin that produces the minimal impedance route begins at node 5 of subnetwork 2. The impedance for this route is 32 units.

2.3 PROGRAM OPTIONS

INTERLINE has a number of options that can be used to influence the routing algorithm. The ability of the user to change the transfer impedances, line classification weights, and originating railroad factor has been previously mentioned. Program options allow the user to directly replace default values of these parameters with desired values. The program allows the user to make the route avoid selected links, nodes, or railroads and to calculate alternative routes. This section describes how these options are incorporated into the INTERLINE routing algorithms.

2.3.1 Link Avoidance

If the impedance of a link is increased by a very large amount, the algorithm will not use the link in a path if there is a feasible detour. When the user specifies that a particular link should not be included in a route, an impedance of 70,000 units is added to the original link impedance. As links are directional, the penalty is normally applied to both directions of travel along the link, although the user has the option to discourage travel in one direction only. Nodes at one or both ends of the high-impedance link may be included in the route, but the route will include the high-impedance link only if there is **no alternative path with a lower impedance.** A message will be printed if penalized links have been included in a route.

2.3.2 Node Avoidance

A route can be constrained from passing through a specified node if a high-impedance penalty is applied to all links that emanate from the node. When these constraints are applied, a 70,000-unit impedance is added to the original impedance of each emanating link. The high impedance associated with using the node will discourage its inclusion in the route provided that an alternative path exists around it. Node avoidance cannot be applied to the origin and/or destination of a route, because all links emanating from these nodes would be penalized, and one

ORNL-DWG-93-7648

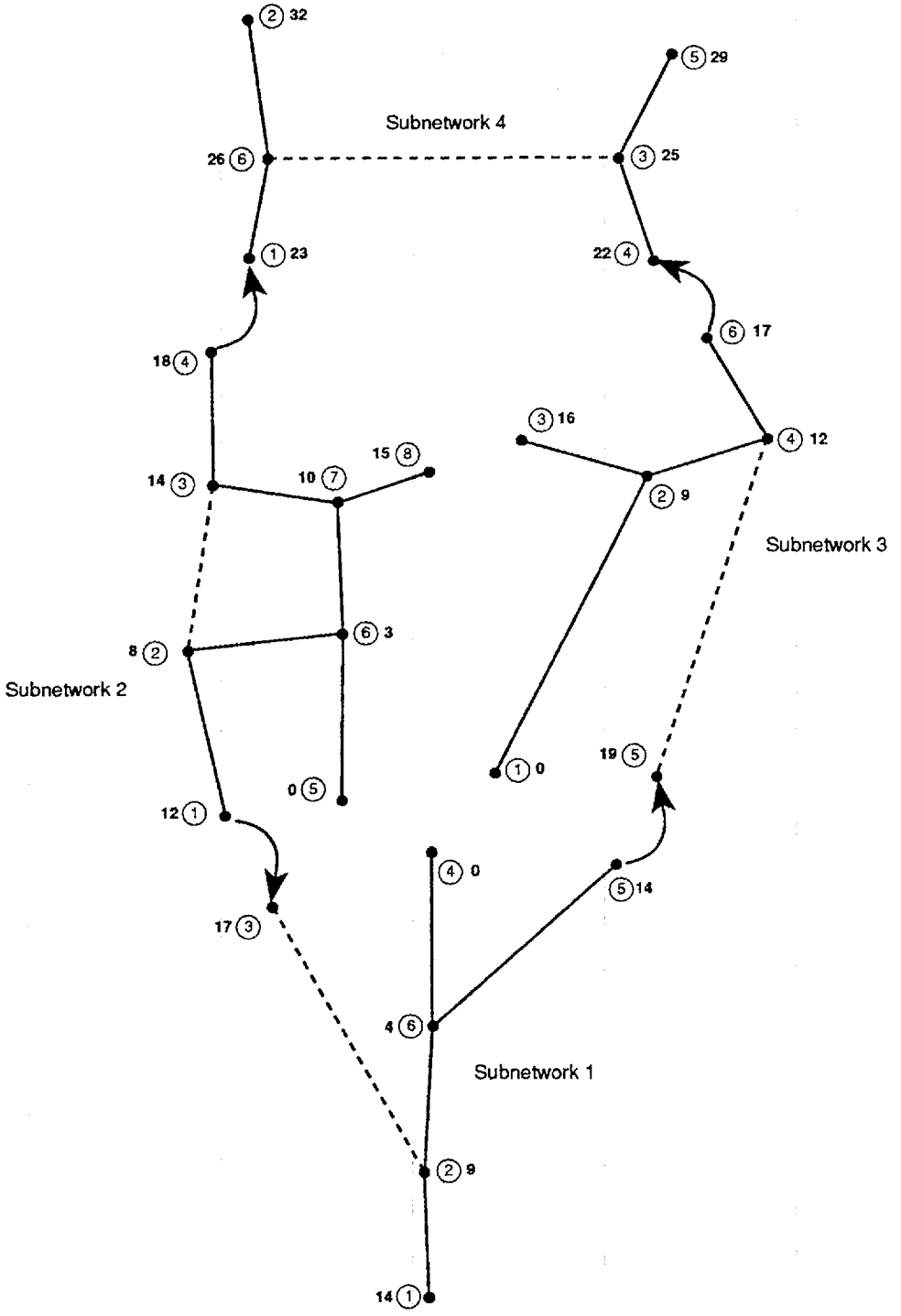


Fig. 13. Completed path tree for a multiple railroad origin route.

of the links is required to be part of the route. A message will be printed if the calculated route passes through a penalized node.

2.3.3 Directed Routing

In directed routing, the user specifies the sequence in which subnetworks will be examined for expansion of the path tree. The user may require, for example, that the route proceed from subnetwork A to subnetwork C to subnetwork D. Of course, these subnetworks must be interconnected such that at least one contiguous path exists between the origin and the destination.

Directed routing is straightforwardly implemented. Instead of examining labels to select a pivot subnetwork in the first step, the algorithm selects the next subnetwork from the user-provided list.

2.3.4 Subnetwork Prohibition

INTERLINE allows the user to specify subnetworks that the path tree will not traverse. Unlike link and node avoidance, where the tree will include the object if there is no alternative, this option completely prohibits the inclusion of the subnetwork in the tree.

Subnetwork prohibition is implemented as follows. Each subnetwork has a status flag. If the value of this flag is FALSE, which is the normal condition, the subnetwork is eligible to be a pivot candidate during the first step. If the status flag is TRUE, the subnetwork will never be considered as a pivot and, therefore, will never be included in the path tree. The user has the capability to set these subnetwork status flags.

2.3.5 Alternative Routing

Frequently, in addition to the optimal route, the user wishes to find a number of alternative routes that have a slightly higher impedance. INTERLINE offers this capability.

To develop alternative routes, INTERLINE first finds the optimal or base route using the user-specified criteria. The impedance of the links and transfers of the base route are then increased to encourage the algorithm to find an alternative route between the origin and destination. Normally, the program increases the impedance of the links in the base route by 20% and increases the base route transfers by 200%. The algorithm is then rerun. Due to the increased impedances of the original route, the second route will tend to use a different set of

links and different transfer points between railroad networks. To generate a third route, the impedances of the links and transfers used in the second route are also increased, and the procedure is repeated. Links that appear in both routes, the base route and first alternative route, would have had their impedances increased two times. The alternative routing algorithm will always produce a different route; however, in some cases there may be only minor differences between routes.

Figures 14-16 demonstrate the alternative routing process. The link and transfer impedances of the path of the base route, shown in Fig. 12, between node 1 in subnetwork 1 and node 2 in subnetwork 4 are increased by 1.2 and 2.0 respectively. Figure 14 shows both the resulting links and the transfer links with increased impedances. The resulting path tree, with associated impedances, is shown in Fig. 15. The alternative route reaches node 2 of subnetwork 4 with an impedance of 64.2 units, whereas the base route shown in Fig. 12 had an impedance of 51. The reader is encouraged to perform the detailed calculations, thus verifying the shortest path shown in Fig. 15. The alternative and base routes are compared in Fig. 16.

2.3.6 Population Density

The INTERLINE routing model includes the option of calculating a population density distribution for a route. A detailed description of the techniques used to generate the population density data is beyond the scope of this manual. The interested reader is referred to ref. 8 for additional details.

ORNL-DWG-93-7649

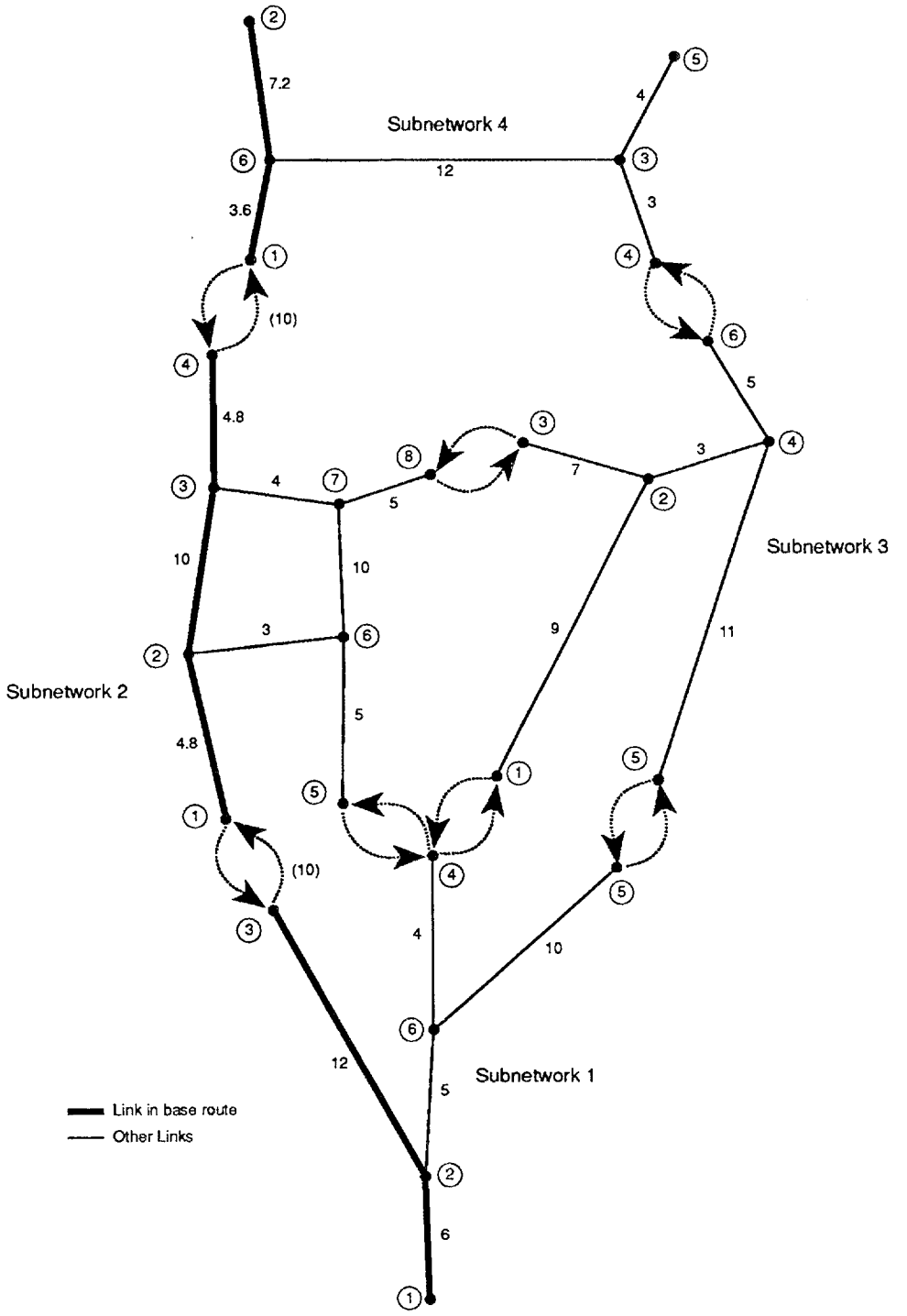


Fig. 14. Network impedances after increasing impedances on the base route.

ORNL-DWG-93-7650

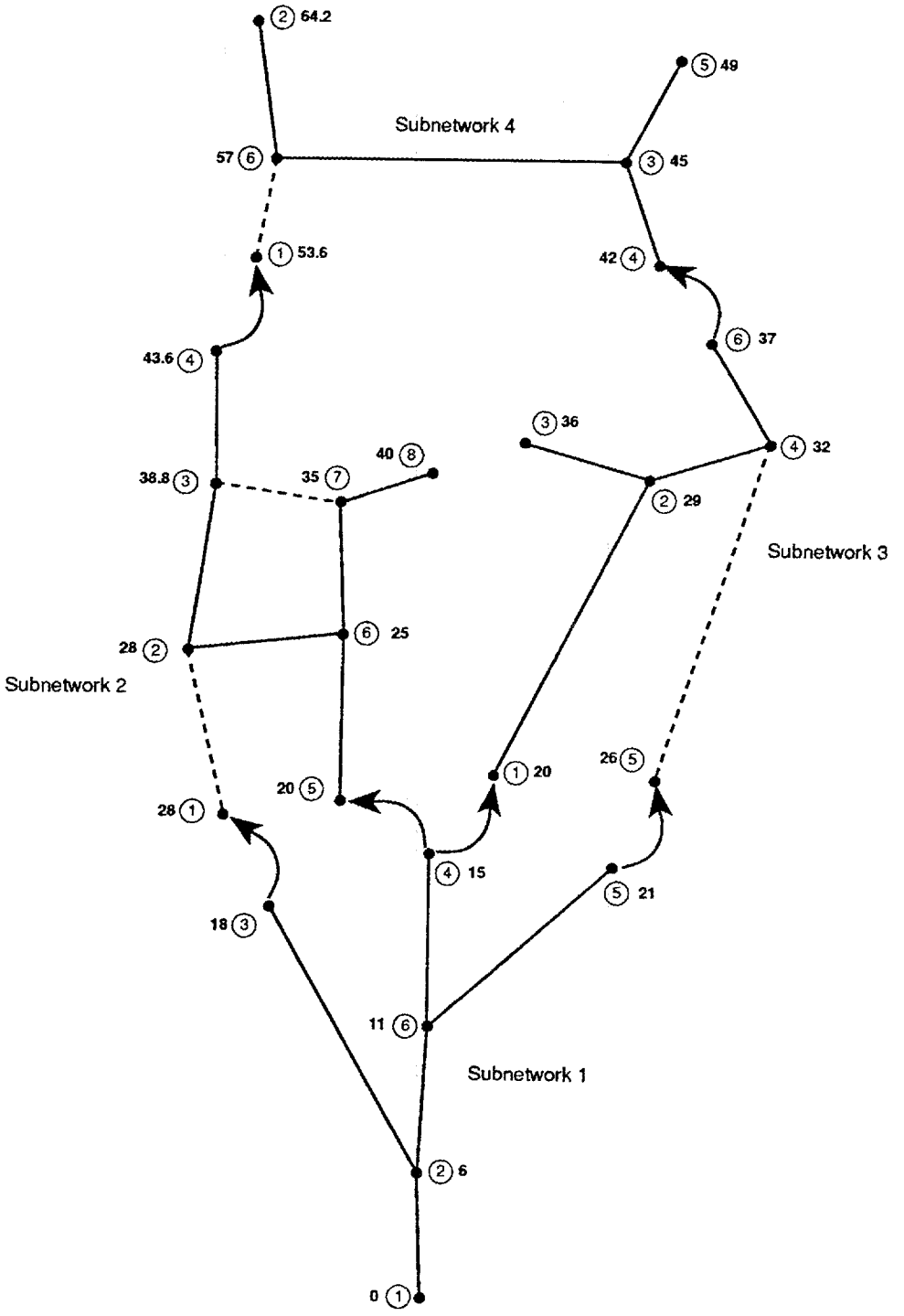


Fig. 15. Completed path tree—alternative routing.

ORNL-DWG-93-7651

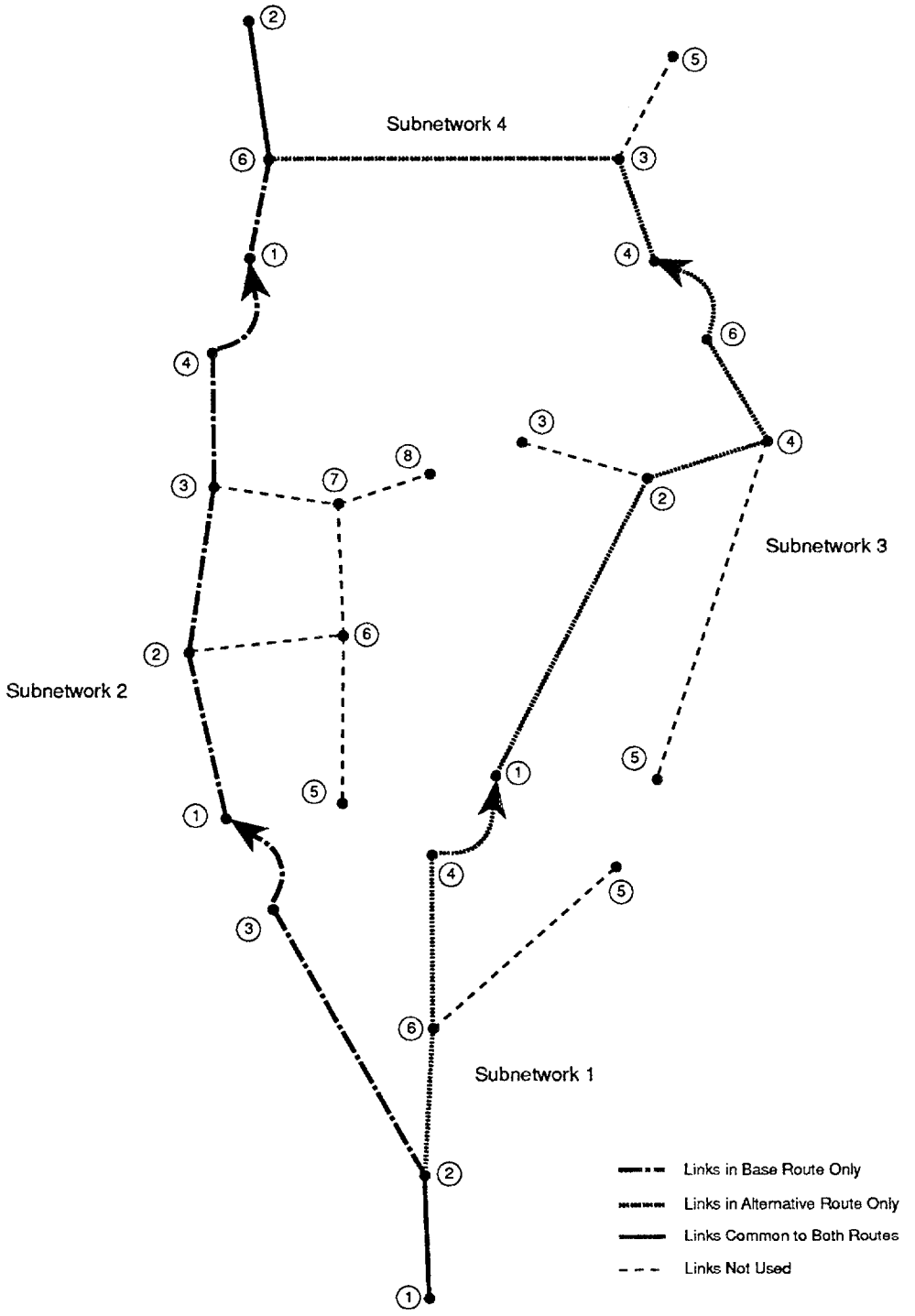


Fig. 16. Path comparison—normal vs alternative routing.

3. INTERLINE USER'S MANUAL

This section is designed to guide the user through the operation of the INTERLINE interactive rail routing model and explain the logic of the program operation. Unless there is a particular need, advanced features described in some sections may be skipped by the beginning user. The individual routing commands are discussed in separate sections in this chapter. All users should review Sects. 3.1 and 3.2. Beginning users will want to initially concentrate on the ROUTE command described in Sect. 3.3.1 and the STOP command described in Sect. 3.3.11. While the material in Sects. 3.3.2 and 3.3.3 is helpful at the beginning level, a detailed understanding of these commands is not essential. As the user becomes more familiar with the operation of the INTERLINE model, the material included in Sect. 3.3.4—the OPTION command—as well as the command's access to the alternative routing, barge routing, and population statistics capabilities (Sects. 3.3.4.1 through 3.3.4.3) would be the next logical extension of the user's capabilities. The advanced user who wishes to include numerous constraints on the routing calculations will need to be familiar with the commands that modify the basic INTERLINE data base. These commands are described in Sects. 3.3.5 through 3.3.8.

Many of the program prompts have a default answer that can be exercised by using a carriage return, **<cr>**. These defaults will be explained in the appropriate sections. Conventions described in the manual will assist the user in determining the sequence of commands needed to operate the INTERLINE program. Program prompts will be indicated by **BOLD UPPERCASE**. User responses will be printed in **bold lowercase**. The user may respond with either uppercase, lowercase, or a combination of the two. However, for clarity, lowercase will be used in this manual. Commentary pertaining to various command responses and options will be enclosed in brackets ([ ]). Additional information will be highlighted with ***bold italics***. This convention is used to distinguish user actions from program information.

3.1 INSTALLATION

The INTERLINE routing package, including the INTERLINE program and railroad/barge network, is contained on a single high-density diskette. A special installation program is included to assist the user in installing INTERLINE on an IBM-compatible personal computer. The procedure for installing the INTERLINE program on a hard disk is as follows:

1. Change the drive specification to the floppy disk drive containing the INTERLINE diskette (assumed to be drive A: in this example) by typing:

   C:\> a:<cr>.

2. Execute the installation routine by typing:

   A:\> install<cr>.

3. Answer the questions presented on the screen. The INSTALL program will ask for the designation of the hard drive (C:, D:, etc.). A new subdirectory **\RAILROAD** will automatically be created for the routing programs and data bases on the selected hard drive. The name of this subdirectory can be changed by responding to the appropriate question during the setup procedure.

The hard drive must have approximately 5.7 MB of free space to accommodate the INTERLINE rail routing model. Once installation is complete, unneeded files will be automatically deleted, and the RAILROAD subdirectory will occupy about 3.7 MB of space. During program operation, an additional 0.8 MB of space will be required.

It is recommended that the rail routing model be installed on at least a 286 IBM compatible computer with 640K of memory and a hard drive with 5.7 MB of free space. Calculational performance will be improved significantly if there is sufficient memory to define a 4-MB random-access memory (RAM) drive. (For information regarding the installation of a RAM drive, see the DOS manual.)

If a RAM drive is be used, the RAILROAD.BAT file in the \RAILROAD subdirectory must be modified as shown in Fig. 17. In this example, it is assumed that drive d: is the RAM drive.

3.2 STARTING THE INTERLINE PROGRAM

To begin using the INTERLINE routing model the user should follow the steps listed below.

1. Change directories to the \RAILROAD subdirectory.

   C:> cd \railroad <cr>

```
@echo off
echo INTERLINE/PC
echo Please wait; Your temporary work file is being created.
copy rrn10b.fil d:          [This line must be added to the file.]
copy interlin.exe d:        [This line must be added to the file.]
copy darrpdn.dat d:         [This line must be added to the file.]
copy rrn10bo.lls d:rrn10b.lls  [The d: must be added to place the destination
                            file on the correct drive.]
d:                          [This line must be added to the file.]
interlin
del rrn10b.lls
del interlin.exe            [This line must be added to the file.]
del darrpdn.dat             [This line must be added to the file.]
del rrn10b.fil              [This line must be added to the file.]
c:                          [This line must be added to the file.]
@echo on
```

Fig. 17. Modified RAILROAD.BAT file.

2.Execute the railroad batch file.

C:\RAILROAD> railroad <cr>*

The first program statement is as follows:

INTERLINE/PC
PLEASE WAIT; YOUR TEMPORARY WORK FILE IS BEING CREATED.
1 FILE(S) COPIED.

There will be a pause at this point while the computer makes a copy of the master data base. The duration of this pause is a function of the type of computer being used. A copy of the master data base is made so that the user can make temporary changes to the data base while performing certain routing operations without permanently altering the master data base. The opening screen is shown in Fig. 18. At this point, the impedance and transfer penalties in the routing model are automatically set to their default values by the program.

The program is now ready to accept commands defining the specific actions to be taken. The various commands are discussed in the following sections.

---

*In this manual it is assumed that the railroad program is installed on drive C. By typing **railroad**, a batch program is run which defines the necessary files and initiates program execution.

```
INTERLINE/PC
Please wait; Your temporary work file is being created.
        1 file(s) copied

    ++++++++++++++++++++++++++++++++++++++
    +            INTERLINE 5.00          +
    +             Developed at           +
    +    OAK RIDGE NATIONAL LABORATORY   +
    +          Revised  26 Feb 92        +
    ++++++++++++++++++++++++++++++++++++++

++ Using the complete network 10.00 (30 JA 92),  Revision  0 (30 JA 92)

If you need assistance at any prompt, type HELP

COMMAND?
```

Fig. 18. Opening screen.

3.3 COMMANDS

The program includes 11 commands that control the action of the INTERLINE interactive rail routing model. Each command will be explained, with examples, in subsequent sections. The commands (shown in Table 3) fit into three general categories: program operation, data base modification, and user assistance.

Table 3. INTERLINE commands

Section	Command
3.3.1	Route
3.3.2	Reroute
3.3.3	Retype
3.3.4	Options
3.3.5	Node
3.3.6	Link
3.3.7	Remove
3.3.8	Transfer
3.3.9	Help
3.3.10	Stop

Four commands, **route**, **reroute**, **retype**, and **option**, relate to program operation. The **option** command allows the user to choose different capabilities that can alter the program operation. The **node**, **link**, **remove**, and **transfer** commands allow the user to modify the network to include specific routing constraints. The **help** and **find** commands are used to obtain additional information or to define specific actions the user should perform. These commands can be used whenever needed. The **stop** command terminates the program and returns the user to the DOS prompt: **C:\RAILROAD>**.

The commands may be abbreviated by using only the first three characters of each command. For example, the **route** command may be specified as **rou** and the **reroute** command as **rer**. In this manual, all commands will be completely spelled out.

The first request for input displayed is

COMMAND?

At this point, the program is ready to accept instructions defining the actions to be taken. At any point in the program operation, the user may enter one or more carriage returns to return to the **COMMAND?** level. The user may then enter any of the commands shown in Table 3, or exit the program by typing **stop**.

3.3.1 Route

The **route** command generates a route from an origin to a destination and is the most frequently used command. The first prompt of the route command is

ENTER ORIGIN >>.

At this point, the user starts the process of selecting an origin, a railroad or railroads serving the origin, a destination, and a railroad serving the destination.

Using a node name is the simplest way to enter a node location. Both the location and state must be entered: for example, **Fargo,ND**. The entire name does not have to be entered. The user can enter the first several letters of the name, a comma, and the standard two-letter abbreviation of the state (e.g., **Far,ND**). The model will then list all locations in that state that start with the characters entered, and the user will be asked to select the proper node from the list supplied. In the example illustrated in Fig. 19, the origin was specified by entering **knox,tn**. The program asks the user to select the proper node by issuing the following prompt:

ENTER NODE NUMBER >>.

In this example, node **7286** was selected as the origin.

```
COMMAND?   route
   Initializing data base, please wait ...

ENTER ORIGIN    >>  knox,tn
++ MULTIPLE NODES WITH NAME:  tn  knox
       7286   KNOXVILLE           TN
      17159   KNOXVILLE; PORT OF  TN
 ENTER NODE NUMBER >>  7286
++   RAILROADS SERVING NODE  7286:
                5-CSXT      10-NS
 ENTER RAILROAD FOR THIS NODE  >>  5
      5  7286   CSXT  KNOXVILLE           TN       .0  1321I   SOURCE  1

++ Initialization complete     1 source,  1 source railroads

 ENTER DESTINATION  >>  milw,wi
++ MULTIPLE NODES WITH NAME:  wi  milw
       5841   MILWAUKEE           WI
      17274   MILWAUKEE; PORT OF  WI
 ENTER NODE NUMBER >>  5841
++   RAILROADS SERVING NODE  5841:
                3-CNW       11-SOO       86-WC        90-WSOR
 ENTER RAILROAD FOR THIS NODE  >>  soo
          11  5841    SOO   MILWAUKEE           WI    155I    SINK NUM  1

Routing ....................................................

   ROUTE FROM:  CSXT  7286-KNOXVILLE           TN     LENGTH:   756.9 MILES
           TO:  SOO   5841-MILWAUKEE           WI  POTENTIAL:  924.76


  MILEAGE SUMMARY BY RAILROAD           A-M   B-M   A-BR  B-BR  OTHER
                        CSXT  673.1    662.6  10.5    .0    .0     .0
                        SOO    83.8     79.8   4.0    .0    .0     .0
                              - - -  - - - - - - - - - - - - -
                       TOTAL   756.9    742.4  14.5    .0    .0     .0
   MILEAGE SUMMARY BY STATE
            78.0-IL    144.5-IN    209.0-KY    213.6-OH     73.0-TN
            38.8-WI


 Do you want to see a route listing (Y/N) ? yes

     RR      NODE             STATE  DIST
    CSXT  7286-KNOXVILLE       TN     0.
 Continue?
    CSXT  3231-COVINGTON       KY   282.
    CSXT  3228-CINCINNATI      OH   286.
    CSXT  3251-HAMILTON        OH   309.
    CSXT  3282-DAYTON          OH   345.
    CSXT  3374-LIMA            OH   413.
    CSXT  4070-GARY            IN   630.
    CSXT  4161-FRANKLIN PARK   IL   673.
   - - - - - - - - - - - - - - - - - - - - - - - - - TRANSFER
    SOO   4161-FRANKLIN PARK   IL   673.
    SOO   4216-DES PLAINES     IL   683.
    SOO   5841-MILWAUKEE       WI   757.


 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
```

Fig. 19. Example of using the route command.

If two or more railroads serve the selected origin, the program will list the railroads and will ask the user to select a railroad by printing

ENTER RAILROAD FOR THIS NODE > >.

The user may enter either the railroad number or abbreviated railroad name, and the program will proceed. The reader is referred to Table B.2 in Appendix B for a description of the different route listings. As shown in Fig. 19, two railroads, the CSXT (subnetwork 5) and the Norfolk Southern (subnetwork 10) provide service in Knoxville. In this example, railroad number 5 (the CSXT) was selected. Following the selection of an origin and railroad, the program will indicate that the initialization is complete.

The user may select more than one originating railroad by entering multiple railroad numbers and/or railroad names. (Multiple entries must be separated with a space: for example, **5 10**, for the railroads listed in Fig. 19.) The user also has the option of requesting all listed railroads by entering **all.** When more than one railroad is selected, the program evaluates routes originating on the different railroads and selects the railroad that produces the minimal impedance route. After the origin has been specified, the user is asked to identify the destination and destination railroad when the program issues the following prompt:

ENTER DESTINATION > >.

The procedure for selecting a destination and a destination railroad is identical to the procedure just described for the origin. However, since the program must have a single end point defined, the user does not have the option of selecting multiple railroads at the destination: that is, **a specific railroad must be selected**.

Once the origin and the destination have been selected, the routing calculation proceeds when the program responds by typing:

ROUTING ..... .

The dots following the routing statement indicate that the program is calculating the route. Once the route calculation has been completed, the program will print a route summary including the origin, the destination, the length of the route, the route potential (the value of the impedance of the route), a mileage summary by railroad and mainline classification, and a mileage summary by state (see Fig. 19). As seen in Fig. 19, the program will prompt the user:

DO YOU WANT TO SEE A ROUTE LISTING (Y/N) ?

[The default for this prompt is yes.]

The program will print a listing of the major nodes and transfer points along the route. The listing is interrupted periodically so the user can read before information scrolls off of the screen. The interrupts are denoted by the **CONTINUE?** printed in the lower left part of the listing. The user can continue with the route listing by entering **y** or **<cr>** or may terminate the listing by entering **n**. The next prompt asks:

DO YOU WANT TO SEE A MORE DETAILED ROUTE LISTING (Y/N) ?

[The default for this prompt is no.]

If the user types **y**, a more detailed listing will be generated. Usually, the initial route listing will be sufficient for most routing applications. As shown in Fig. 19, the program will next prompt the user:

DO YOU WANT A HARD COPY OF THIS ROUTE INFORMATION (Y/N) ?

[The default for this prompt is no.]

If the user requires a hard copy, the program will automatically send the data to the printer connected to the PC.

Occasionally, the desired origin or destination is not included in the list of nodes printed by the INTERLINE program. The user may enter a **<cr>** to return to the

ENTER ORIGIN >> or **ENTER DESTINATION >>**

prompt and try entering a single letter or two letters of the node name along with the state abbreviation. As shown in Fig. 20, the list produced by using a single letter will show only the first 20 nodes in that state that begin with the letter. If the desired node is in the list, the user may proceed as previously directed. If the user is still having difficulty defining the desired node, the railroad atlas[9] should be checked.

3.3.1.1 Alternative Ways to Define Origins/Destinations

As shown below, node names and railroads may be specified in several ways. These techniques will speed up the program operation for the experienced user who knows the proper node number and railroads providing service at specific nodes. These techniques are not recommended for the casual user.

1. Enter a node number (e.g., **7286**) rather than a node name. If multiple railroads are available at this node, the user will be asked to select a railroad.

```
ENTER ORIGIN     >>  m,wi
++ MULTIPLE NODES WITH NAME:  wi  m
       5914   MADISON              WI
       5809   MANITOWOC            WI
       5980   MARENGO              WI
       5676   MARINETTE            WI
       5770   MARKESAN             WI
       5726   MARSHFIELD           WI
       5915   MAZOMANIE            WI
      15247   MEDFORD              WI
       5981   MELLEN               WI
       5962   MENOMONIE JCT        WI
       5735   MERRILLAN            WI
       5913   MIDDLETOWN           WI
       5716   MILLSTON             WI
       5901   MILTON JCT           WI
       5841   MILWAUKEE            WI
      17274   MILWAUKEE; PORT OF   WI
       5894   MINNESOTA JCT        WI
       5969   MINONG               WI
      15242   MONROE               WI
       5823   MUKWONAGO            WI
 ENTER NODE NUMBER >>
```

Fig. 20. List of node names in Wisconsin starting with the letter m.

2. Enter a node number and railroad name (e.g., **7286 rcsxt**).*
3. Enter a node number and railroad number (e.g., **7286 r5**).**
4. Enter a node name, state, and railroad name (e.g., **knox,tn rcsxt**).**
5. Enter a node name, state, and railroad number (e.g., **knox,tn r5**).**
6. Enter a node and multiple railroad numbers (e.g., **knox,tn r5 10** or **7286 r5 10**).**
7. Enter a node and request all railroads (e.g., **knox,tn rall** or **7286 rall**).**

3.3.2 Reroute

The **reroute** command provides a convenient and efficient way to generate additional routes from a previously defined origin. From the user's standpoint, the **reroute** command simplifies data input, since only a new destination needs to be specified. From a computational standpoint, the **reroute** command uses the most recent routing calculation as the starting point for the calculation of the next route. Hence, the shortest path tree does not have to be

*The **r** in this example and those that follow instructs the program that railroad information follows.

In order to use some of these techniques, the user must know the number of the desired node and/or the railroad(s) serving the desired node. Remember, however, that the basic routing procedure outlined in Sect. 3.3.1 will **always work.

regenerated from the origin as would be required if the same route were to be generated using the **route** command.

In order for the **reroute** command to work, a few conditions need to be satisfied. First, a route must be generated with the **route** command. Then, the basic routing parameters (node or link deletion, option capabilities, etc.) must not have been changed since the last use of the **route** command. The user will be notified if one of the above conditions has been violated when the program issues the following statement:

NO PREVIOUS ROUTE GENERATED, USE THE ROUTE COMMAND.

An example of using the **reroute** command is shown in Fig. 21. The initial route from Fargo, North Dakota, to Flagstaff, Arizona, was generated using the **route** command. A second route, terminating in Billings, Montana, was calculated using the **reroute** command.

3.3.3 Retype

The **retype** command will simply generate a listing of the most recently calculated route. While this command is not used frequently, it is an efficient way of getting additional information about the last route. With the **retype** command, the user has all of the output options that were available when the route was originally generated, such as detailed route listing and hard copy.

3.3.4 Options

The **option** command is used to activate or deactivate specific program capabilities. The four capabilities which can be accessed by the **option** command are alternative routing, barge routing, population density, and expert.

The first three capabilities are used frequently and are discussed in the following sections. The **expert** option has been designed to allow the advanced user to change various weighting factors used in the routing calculations. Since the basic INTERLINE user will not normally a need these advanced features, the discussion of the expert option has been deferred to Appendix B.

When the user enters **option** at the command level, the program will list the status of the four capabilities and ask the user to specify the capability whose status should be modified (see Fig. 22). The letters T or F indicate whether the particular option is active or disabled respectively. The user selects a capability by entering the first letter of the desired capability, **a**, **b**, **p**, or **e**. As shown in Fig. 22, when a capability has been selected, the program will reprint

[Initial route calculation using the **route** command]

```
COMMAND?   route
   Initializing data base, please wait ...

ENTER ORIGIN    >>  farg,nd
      2 11132   BN    FARGO                ND       .0   754I   SOURCE  1

++ Initialization complete     1 source,  2 source railroads

 ENTER DESTINATION  >>  flag,az
          1 12959    ATSF  FLAGSTAFF           AZ    463I    SINK NUM  1

Routing ..............................................................
...............................

   ROUTE FROM:  BN   11132-FARGO               ND     LENGTH:  1860.9 MILES
          TO:  ATSF 12959-FLAGSTAFF            AZ  POTENTIAL:  2127.9


  MILEAGE SUMMARY BY RAILROAD           A-M   B-M   A-BR  B-BR  OTHER
                        ATSF  673.0    352.6 320.4    .0    .0     .0
                        BN   1187.9    310.0 877.9    .0    .0     .0
                             - - -  - - - - - - - - - - - - - -
                       TOTAL  1860.9    662.61198.3    .0    .0     .0
   MILEAGE SUMMARY BY STATE
          170.0-AZ    386.8-CO     76.0-IA    288.0-MN    430.4-NE
          487.2-NM     10.5-ND     12.0-SD


 Do you want to see a route listing (Y/N) ? no
 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
```

[Second route from Fargo calculated using the **reroute** command]

```
COMMAND?   reroute

 ENTER DESTINATION  >>  billings,mt
++   RAILROADS SERVING NODE 13207:
                2-BN        59-MRL
 ENTER RAILROAD FOR THIS NODE  >>  bn
          2 13207    BN    BILLINGS            MT   1209I    SINK NUM  2

Routing

   ROUTE FROM:  BN   11132-FARGO               ND     LENGTH:   630.6 MILES
          TO:  BN   13207-BILLINGS             MT  POTENTIAL:  504.48


  MILEAGE SUMMARY BY RAILROAD           A-M   B-M   A-BR  B-BR  OTHER
                        BN    630.6    630.6    .0    .0    .0     .0
                             - - -  - - - - - - - - - - - - - -
                       TOTAL   630.6    630.6    .0    .0    .0     .0
   MILEAGE SUMMARY BY STATE
          265.0-MT    365.6-ND


 Do you want to see a route listing (Y/N) ? no
 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
```

Fig. 21. Use of the reroute command.

```
COMMAND?   option

Options:
  Alternative routing-F
  Barge routing-F
  Population density-F
  Expert-F
Enter first letter of option to change > a

Alternative routing has been activated

Options:
  Alternative routing-T
  Barge routing-F
  Population density-F
  Expert-F
Enter first letter of option to change >
```

Fig. 22. Option command choices.

the list of capabilities and their updated status. In Fig. 22, the second listing shows that the alternative routing capability has been activated.

Selecting a capability under the **option** command is analogous to flipping a light switch. If the selected capability is not active, it will be activated. If the capability is active at the time of selection, it will be disabled. A message will always be printed indicating which capability was selected and whether it was activated or disabled.

The option command is terminated when the user responds to the prompt:

ENTER FIRST LETTER OF OPTION TO CHANGE >

with a **<cr>**. (An additional **<cr>** may be required before the program returns to the **COMMAND?** prompt.)

It is recommended that the barge option be disabled if it is not needed for the next route calculation. If the barge option is active, barge segments may appear in routes that the user anticipated to include only rail segments. Since it may not be immediately obvious that barge segments are included, the user could incorporate erroneous results in the project on which he or she is working. The alternative routing and population density capabilities can be left in the active state even if they are not needed for the next route calculation. These capabilities will not affect route calculations and will simply require the user to answer additional questions regarding whether alternative routes or population density tables are needed for the current route.

3.3.4.1 Alternative Routing

Once the alternative routing capability has been activated (Sect. 3.3.4), the user must calculate a base route using the **route** command. This step defines the origin, destination, and any special routing constraints. The first listing in Fig. 23 shows the calculation of a base route from Fargo, North Dakota, to Flagstaff, Arizona. Please note that the program identifies this as the base route by printing

BASE ROUTE FOR ALTERNATIVE ROUTING.

At the end of the route listing, the following question asks the user whether an alternative route between this origin and destination is desired:

DO YOU WANT TO CALCULATE AN ALTERNATIVE ROUTE (Y/N) ?

[The default for this prompt is no.]

If an alternative route is desired, the user responds with a **y**. If an alternative route is not needed, the user must respond with either a **n** or a **<cr>**, and the program will return the user to the **COMMAND?** prompt.

The user may request as many alternative routes as desired. However, after the third or fourth alternative route, there may be only slight differences between new alternative routes and previously calculated alternative routes.

3.3.4.2 Barge Routing

Once the barge routing capability has been activated through the **option** command (Sect. 3.3.4), the inland waterway and deep water networks (*BRG and *M-M) are activated allowing the user to calculate barge routes or rail/barge intermodal routes. As shown in Fig. 24, barge routes are calculated with the **route** command. In order to generate barge routes, the user must designate a port and one of the waterway networks as the origin and/or destination of the route. In Fig. 24, the port of St. Paul in Minnesota was selected as the origin, while the port of St. Louis, Missouri, was selected as the destination. In general, only one of the waterway networks is available; however, at a number of the major ports (e.g., New Orleans), both the inland waterway and deep water networks are available. At such locations, the user will be asked to select the proper network. Intermodal routes can be calculated by selecting a rail node as the origin or destination and a port as the other end point of the route.

[Base Route]

```
COMMAND?   route
   Initializing data base, please wait ...

ENTER ORIGIN    >>  farg,nd
     2 11132   BN    FARGO                ND        .0   7541   SOURCE  1

++ Initialization complete     1 source,  2 source railroads

 ENTER DESTINATION  >>  flag,az
          1 12959    ATSF  FLAGSTAFF           AZ    4631    SINK NUM  1

Routing ..............................................................
...............................

   ROUTE FROM:  BN   11132-FARGO               ND     LENGTH:  1860.9 MILES
           TO:  ATSF 12959-FLAGSTAFF           AZ  POTENTIAL:  2127.9


 Base route for alternative routing


  MILEAGE SUMMARY BY RAILROAD           A-M   B-M   A-BR  B-BR  OTHER
                       ATSF  673.0    352.6 320.4    .0    .0     .0
                       BN   1187.9    310.0 877.9    .0    .0     .0
                            - - -  - - - - - - - - - - - - - -
                      TOTAL  1860.9    662.61198.3    .0    .0     .0
   MILEAGE SUMMARY BY STATE
           170.0-AZ    386.8-CO     76.0-IA    288.0-MN    430.4-NE
           487.2-NM     10.5-ND     12.0-SD


 Do you want to see a route listing (Y/N) ? no
 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
 Do you want to calculate an alternative route (Y/N) ? yes
```

[Alternative Route]

```
Routing ..............................................................
..................................

   ROUTE FROM:  BN   11132-FARGO               ND     LENGTH:  1945.9 MILES
           TO:  ATSF 12959-FLAGSTAFF           AZ  POTENTIAL:  2341.9


 Alternative route number  1
 Weighting factors:  for links  .200  for transfers  2.000


  MILEAGE SUMMARY BY RAILROAD           A-M   B-M   A-BR  B-BR  OTHER
                       ATSF 1207.9   1207.9    .0    .0    .0     .0
                       BN    738.0    258.1 479.9    .0    .0     .0
                            - - -  - - - - - - - - - - - - - -
                      TOTAL  1945.9   1466.0 479.9    .0    .0     .0
   MILEAGE SUMMARY BY STATE
           170.0-AZ     76.0-IA    289.4-KS    288.0-MN    107.1-MO
           223.4-NE    433.6-NM     10.5-ND    118.0-OK     12.0-SD
           217.9-TX


 Do you want to see a route listing (Y/N) ? no
 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
 Do you want to calculate another alternative route (Y/N) ? no
```

Fig. 23. Alternative routing example.

```
COMMAND?   route
   Initializing data base, please wait ...

ENTER ORIGIN    >>  st paul,mn
++ MULTIPLE NODES WITH NAME:  mn  st paul
       9830   ST PAUL              MN
       9785   ST PAUL PARK         MN
      17123   ST PAUL; PORT OF     MN
 ENTER NODE NUMBER >>  17123
     95 17123   *BRG  ST PAUL; PORT OF    MN       .0   5541   SOURCE  1

++ Initialization complete     1 source,  1 source railroads

 ENTER DESTINATION  >>  st louis,mo
++ MULTIPLE NODES WITH NAME:  mo  st louis
      10858   ST LOUIS             MO
      17138   ST LOUIS; PORT OF    MO
 ENTER NODE NUMBER >>  17138
          95 17138    *BRG  ST LOUIS; PORT OF   MO    568I    SINK NUM  1

Routing .

   ROUTE FROM:  *BRG 17123-ST PAUL; PORT OF    MN     LENGTH:   626.6 MILES
           TO:  *BRG 17138-ST LOUIS; PORT OF   MO  POTENTIAL:  350.90


  MILEAGE SUMMARY BY RAILROAD            A-M   B-M   A-BR  B-BR  OTHER
                        *BRG  626.6       .0    .0    .0    .0  626.6
                              - - - - - - - - - - - - - - - - - - -
                       TOTAL  626.6       .0    .0    .0    .0  626.6
   MILEAGE SUMMARY BY STATE
           626.6-*W


 Do you want to see a route listing (Y/N) ? yes

     RR      NODE              STATE  DIST
   *BRG 17123-ST PAUL; PORT OFMN     0.
 Continue?
   *BRG 17132-DAVENPORT; PORT IA   340.
   *BRG 16648-MISS/MISSOURI JCMO   613.
   *BRG 17138-ST LOUIS; PORT OMO   627.


 Do you want to see a more detailed route listing (Y/N) ? no
 Do you want a hard copy of this route information (Y/N) ? no
```

Fig. 24. Barge routing calculation.

3.3.4.3 Population Density

Activating the population density capability via the **option** command allows the user the opportunity to request a population density table for the most recently calculated route. As shown in earlier examples, the user is asked a number of questions after the route has been calculated. When the population density capability is activated, an additional question is included in this list:

DO YOU WANT TO SEE A POPULATION DENSITY LISTING (Y/N) ?

[The default for this prompt is no.]

If a population density table is desired, the user must respond to the above question with a **y**. A response of either **n** or **<cr>** will suppress the generation of the population density table.

A population density table is shown for a route between Fargo, North Dakota, and Flagstaff, Arizona, in Fig. 25. The basic table includes 12 population density categories ranging from 0 to over 9996 people per square mile. The entries in the table show the distance the route travels in each category. The population density data are generated for the entire route and also for each state the route passes through. For example, in Fig. 25, the route between Fargo and Flagstaff is 1860.9 miles long. Of this route, 661 miles travels through areas with a population density between 0 and 5 people per square mile, and 66.6 miles of the route passes through areas with a population density between 5.0 and 22.7 people per square mile. This route also travels 170 miles within the state of Arizona. Over half of this distance, 103.6 miles, has a population density between 0 and 5 people per square mile.

After the basic population density table has been listed, the user will be asked whether the population density data should be converted into a form usable in the RADTRAN risk analysis code when the following question is printed:

DO YOU WANT RADTRAN INPUT DATA (Y/N) ?

[The default for this prompt is no.]

The user must respond with a **y** if the RADTRAN data are desired or with an **n** or a **<cr>** if these data are not needed. An example of RADTRAN input data for the same route is shown in Fig. 25. RADTRAN data consist of route distances in three population density zones: rural, suburban, and urban. The distance traveled within each population density zone is given in both miles and kilometers.

3.3.5 Node

As discussed in Sect. 2.3.2, a route can be constrained from passing through a specified node if a high impedance penalty is applied to all links that emanate from the node. This is accomplished in the INTERLINE model by adding a 70,000-unit impedance to the original link impedance for each link emanating from the node. The high impedance will discourage the inclusion of the node in the route, provided an alternative path exists around the node. Putting a large penalty on the links emanating from a node is called removing the node from the network.

```
   POPULATION DENSITY FROM:  BN   11132-FARGO              ND
                      TO:  ATSF 12959-FLAGSTAFF            AZ

          ------------------ MILEAGE WITHIN DENSITY LEVELS ------------------
                <0.0   5.0  22.7  59.7   139   326   821  1861  3326  5815
St Miles    0  -5.0 -22.7 -59.7  -139  -326  -821 -1861 -3326 -5815 -9996 >9996
-------------------------------------------------------------------------------

AZ 170.0   .0 103.6  47.7   2.5   3.8   3.0   4.4   3.0   2.0    .0    .0    .0
CO 386.8   .0 204.8  71.0  38.2  13.3   9.6  10.4  11.4  15.6  12.5    .0    .0
IA  76.0   .0   4.0  29.1  30.6   4.8   5.0   1.0   1.0    .5    .0    .0    .0
MN 288.0   .0  32.7 150.9  58.0  23.0  13.6   2.7   1.8   2.5   1.9    .9    .0
NE 430.4   .0  97.3 165.9  93.4  34.7  16.7   8.9   5.7   4.5   3.3    .0    .0
NM 487.2   .0 210.1 156.7  47.9  19.5  17.1  15.9  12.7   4.9   2.5    .0    .0
ND  10.5   .0   8.5    .0    .0    .0    .0    .0    .4    .4   1.2    .0    .0
SD  12.0   .0    .0   5.2   3.7   3.1    .0    .0    .0    .0    .0    .0    .0

Totals
  1860.9   .0 661.0 626.6 274.2 102.2  64.9  43.3  35.9  30.4  21.5    .9    .0
Percentages
           .0  35.5  33.7  14.7   5.5   3.5   2.3   1.9   1.6   1.2    .0    .0
Basis: 1980 Census data

Do you want RADTRAN input data (Y/N) ? yes

   RADTRAN Input Data      Rural Suburban    Urban

   Weighted Population
        People/sq. mi.      19.1    956.2   4704.2
        People/sq. km.       7.4    369.2   1816.3

   Distance                                        Total
        Miles             1664.0    174.5     22.4  1860.9
        Kilometers        2677.9    280.8     36.0  2994.7

   Basis (people/sq. mi.)   <139 139-3326    >3326   1980 Census data


 Do you want a hard copy of this route information (Y/N) ? no
```

Fig. 25. Population density table and RADTRAN input data.

The **node** command is an advanced but powerful command, and its use requires a knowledge of the structure of the INTERLINE network. The **node** command and its companion command **node 1** are used to remove and reinstall nodes respectively. Node removal applies only to the current INTERLINE session. When a session is ended, and the program is restarted at a later time, all nodes are active.

The user needs to be careful in defining nodes to be removed from routing consideration. First, neither the origin nor the destination should be removed since the route must leave or enter these nodes. Secondly, many nodes are served by more than one railroad. The **node** command will request that the user specify a location and a railroad or a number of railroads at that location. Those combinations of locations and railroads will be removed. Consider the Knoxville

location that has been used in a number of examples in this manual. Two railroads provide service at Knoxville, the Norfolk Southern and the CSXT. If the Knoxville node served by the NS railroad is specified to be removed, routes can still pass through Knoxville on the CSXT railroad. From a program standpoint, there are two Knoxville nodes, one on the NS railroad and the other on the CSXT railroad.

A number of node removal activities are summarized in Fig. 26. The first example illustrates the complete removal of Knoxville, Tennessee, by removing both railroads. In this case, both railroads were selected by the **all** response to the question

ENTER RAILROADS FOR THIS NODE > >.

The node removal process can be terminated by answering the

ENTER NODE TO DELETE > >

prompt with a **<cr>**. This will return the user to the **COMMAND?** prompt.

If the user tries to remove a node that has already been deleted, the program will issue the following statement:

**** NO CHANGE (NODE ALREADY DELETED).**

The second example in Fig. 26 shows how a deleted node can be reactivated. The procedure to reactivate a node is identical to that for removing a node except that the command for reactivating is **node 1**. After **node 1** has been entered, the program will prompt the user to determine whether nodes will be deleted or restored by printing:

DO YOU WANT TO RESTORE NODES (Y/N) ?

[The default for this prompt is yes.]

A **y** or **<cr>** answer will activate the node restoration logic. An **n** answer will result in all subsequent nodes being deleted.

In Fig. 26, the nodes in Knoxville that were deleted in the first example are reactivated. Note that in this example the node name was specified by the node number **(7286)** rather than **knox,tn**.

The **node 1** command can be terminated by answering the

ENTER NODE TO RESTORE > >

prompt with a **<cr>**. This will return the user to the **COMMAND?** prompt.

[Removing nodes]

```
COMMAND?    node

  ENTER NODE TO  DELETE  >>      knox,tn
++ MULTIPLE NODES WITH NAME:  tn  knox
       7286   KNOXVILLE           TN
      17159   KNOXVILLE; PORT OF  TN
 ENTER NODE NUMBER >>  7286
++   RAILROADS SERVING NODE  7286:
                 5-CSXT      10-NS
 ENTER RAILROAD FOR THIS NODE  >>  all
     5R  CSXT    7286-KNOXVILLE          TN   1321I   DELETION    1
    10R  NS      7286-KNOXVILLE          TN    847I   DELETION    2

  ENTER NODE TO  DELETE  >>
++     2 nodes deleted
```

[Reactivating nodes]

```
COMMAND?    node 1
 Do you want to restore nodes (Y/N) ? yes

  ENTER NODE TO RESTORE  >>     7286
++   RAILROADS SERVING NODE  7286:
                 5-CSXT      10-NS
 ENTER RAILROAD FOR THIS NODE  >>  all
     5R  CSXT    7286-KNOXVILLE          TN   1321I   DELETION   -1
    10R  NS      7286-KNOXVILLE          TN    847I   DELETION   -2

  ENTER NODE TO RESTORE  >>
++    -2 nodes deleted
```

Fig. 26. Node deletion and restoration.

3.3.6 Link

The **link** command is a very powerful command because it allows the user to remove a link or to constrain movement in one direction along a link. Like the **node** command, the **link** command is an advanced command, and its use requires a knowledge of the structure and contents of the INTERLINE data base. Links are normally defined as two-way, meaning that traffic may move in either direction along the track. A link is removed by adding a large impedance (70,000 units) to the original link impedance. With this high impedance, the routing calculations will select an alternative path around the removed link if such a detour exists. Nodes at one or both ends of the removed link may be included in the route since other links enter and leave these nodes. The route will include a removed link only if there is no alternative path. If this occurs, a message will be printed stating that a removed link is included in the route.

The **link** command permits the user to remove a link from the network or to restore a previously deleted link or links to the network. As seen in Fig. 27, the program requests the user to

ENTER LINK > >.

The user has three options at this point: (1) enter a node name, (2) enter a link number, or (3) enter a **<cr>**. All links are numbered, and if the user happens to know the appropriate link number, it can be used at this prompt. Entering a **<cr>** will terminate the link command and return the user to the **COMMAND?** prompt.

If the node is served by more than one railroad, the user will be prompted to select the appropriate railroad. The user may select more than one railroad by entering multiple railroad identifications or select all railroads be entering **all**. In the link deletion example shown in Fig. 27, the Knoxville, Tennessee, node and the NS railroad were selected. Next, the program starts to list the links leaving and entering Knoxville on the NS tracks. For example, the first link goes from Knoxville to node number 7275. Note that the user will encounter nonnamed nodes when using the **link** command, and a detailed knowledge of the network is needed to fully utilize this capability. At the end of the first line is the question **OK?** The user responds with a **y** or a **<cr>** if this link is to be deleted or an **n** if the link is not to be deleted. Deleting this link will prevent a route from using this link between Knoxville and node 7275. A route could still use this link between node 7275 and Knoxville for travel in the opposite direction because, at this point, the link has been deleted in only one direction. Note that the seventh link (6769) is the reverse of the link discussed above. Both this link and the previously discussed link must be deleted to block the use of the link in either direction.

Blocked links may be activated by entering the node name or link number followed by a **p-1** at the **ENTER LINK > >** prompt. In Fig. 27, the link reactivation example specified link number 6769 by entering

6769 p-1.

Then the user will be asked to select the links to be activated as they are listed by the program by responding to the question **OK?**

Deletion of links is not permanent and applies only to the current session of INTERLINE. When the model is restarted, all links are activated and any links deleted during previous sessions are lost. The actual deletion is recorded in the temporary network file that is created at the start

of a session. When the program is terminated, this temporary file is deleted. If the user has the need to retain deletions, etc., please refer to the discussion in Sect. B.4 of Appendix B.

[Deleting links]

```
COMMAND?    link

  ENTER LINK  >>    knox,tn
++ MULTIPLE NODES WITH NAME:  tn  knox
       7286   KNOXVILLE           TN
      17159   KNOXVILLE; PORT OF  TN
 ENTER NODE NUMBER >>  7286
++   RAILROADS SERVING NODE  7286:
                5-CSXT      10-NS
 ENTER RAILROAD FOR THIS NODE  >>  ns
   6769K  NS   KNOXVILLE  ->   7275      TN   14.4 NEW=DEL   OK? y
   6764K  NS   KNOXVILLE  -> ROCKFORD    TN   11.4 NEW=DEL   OK? y
   6758K  NS   KNOXVILLE  -> CORRYTON    TN   38.0 NEW=DEL   OK? y
   6727K  NS   KNOXVILLE  -> LENOIR CIT  TN   14.0 NEW=DEL   OK? y
   6762K  NS   KNOXVILLE  -> MARBLEDALE  TN   20.0 NEW=DEL   OK? y
   6760K  NS   KNOXVILLE  -> JEFFERSON   TN   18.0 NEW=DEL   OK? y
   6769K  NS        7275  -> KNOXVILLE   TN   14.4 NEW=DEL   OK? y
   6764K  NS   ROCKFORD   -> KNOXVILLE   TN   11.4 NEW=DEL   OK? y
   6758K  NS   CORRYTON   -> KNOXVILLE   TN   38.0 NEW=DEL   OK? y
   6727K  NS   LENOIR CIT -> KNOXVILLE   TN   14.0 NEW=DEL   OK? y
   6762K  NS   MARBLEDALE -> KNOXVILLE   TN   20.0 NEW=DEL   OK? y
   6760K  NS   JEFFERSON  -> KNOXVILLE   TN   18.0 NEW=DEL   OK? y

  ENTER LINK  >>
++     12 link impedances changed
```

[Restoring links]

```
COMMAND?    link

  ENTER LINK  >>    6769 p-1
   6769K  NS        7275  -> KNOXVILLE  TN  DEL   NEW= 14.4 OK? y
   6769K  NS   KNOXVILLE  ->   7275     TN  DEL   NEW= 14.4 OK? y

  ENTER LINK  >>
++      2 link impedances changed
```

Fig. 27. Link removal and restoration.

3.3.7 Remove

This advanced command permits the user to remove or restore subnetworks from or to the set of networks that can be used to select a route. If a railroad has been removed, a route will not use any links in that subnetwork. The user should not remove subnetworks 92 and/or

93. Each of these subnetworks contains a number of individual small Class III and terminal railroads. Removing these subnetworks may affect the route selection process. Care also needs to be used when deleting railroads because the user could inadvertently define a disconnected network. In that case, the route selection process will fail.

Previously removed railroads can be restored or reactivated with the **remove** command. Any railroad removed at the end of a session will automatically be activated when the INTERLINE model is run another time. As discussed in Sect. 1.2.2.4, the inland waterway and deep water networks (*BRG and *M-M) and the AMTRAK (AMTK) network are not activated at the start of an INTERLINE session. The easiest way to activate or deactivate the two waterway networks is to use the **option** command discussed in Sect. 3.3.4. If the AMTK network is needed for some reason, the only way it can be activated is with the **remove** command.

An example of the **remove** command is shown in Fig. 28. The first prompt is

ENTER RRs TO REMOVE > >.

The user must enter the name or number of the railroad to be removed or a list of railroads to be removed. The first example in Fig. 28 uses a list of railroad numbers to remove the CSXT; NS; Aberdeen, Carolina & Western (ACWR); and Bangor & Aroostook (BAR) railroads. The second section of Fig. 28 shows how the **all** response to the prompt can be used to produce a listing of all deleted railroads. The four railroads deleted in the first example appear on the list along with the default deletions of AMTK, *BRG, and *M-M.

Deleted railroads may be activated with the **remove** command as shown in the third example of Fig. 28. In this example, the railroad numbers are entered with a minus sign (**-5 -10 -15 -20**) to indicate that these railroads are to be activated.

Abbreviated railroad names may also be used in the **remove** command. An example of using railroad names is also included in Fig. 28. For removal, the railroad names can be simply listed (**bn up** to remove the Burlington Northern and Union Pacific railroads). If railroad names are to be used to reactivate deleted railroads, they must be followed with a space and an asterisk (**bn * up** * in the final example in Fig. 28). **Railroad names and numbers may not be mixed on the same line.**

The **remove** command may be terminated by entering a **<cr>** at the following prompt:

ENTER RRs TO REMOVE > >.

[Using railroad numbers to remove railroads]

```
COMMAND?    remove
 ENTER RRs TO REMOVE  >>   5 10 15 20
                       5  CSXT
                      10  NS
                      15  ACWR
                      20  BAR
```

[Listing deleted railroads]

```
 ENTER RRs TO REMOVE  >>   all
++ Removed RRs are:
                       5  CSXT
                      10  NS
                      15  ACWR
                      20  BAR
                      94  AMTK
                      95  *BRG
                      96  *M-M
```

[Using railroad numbers to activate deleted railroads]

```
 ENTER RRs TO REMOVE  >>   -5 -10 -15 -20
                      -5  CSXT
                     -10  NS
                     -15  ACWR
                     -20  BAR
```

[Using railroad names to delete railroads]

```
 ENTER RRs TO REMOVE  >>   bn up
                       2  BN
                      14  UP
```

[Using railroad names to activate railroads]

```
 ENTER RRs TO REMOVE  >>   bn * up *
                      -2  BN
                     -14  UP
```

Fig. 28. Railroad removal and restoration.

3.3.8 Transfer

The **transfer** command, another advanced command, permits the user to modify or list the transfer penalty between the subnetworks in the INTERLINE model. The standard impedance value for a transfer is 300 units, equivalent to 300 miles of travel on an A-mainline. Transfers to terminal railroads receive an impedance of 151 units, so that these railroads may be used as

connections between two Class I railroads without significant penalty. (The impedance of the two transfers involved will be 302 units, slightly more than a direct Class I to Class I transfer.) Transfers between railroad companies under a common corporate umbrella are also lowered to values that are based upon expert judgement reflecting actual operational coordination between the subsidiary companies. Transfers between the railroads and the waterway network have an impedance of 900 units.

Unlike the other commands, the data describing transfer changes must be entered in specific columns. After the user has entered the **transfer** command, the program will respond with the prompt for data:

***--RR1-- --RR2-- --PENALTY-- ---NODE--------**

ENTER POINT > >.

The railroad names or numbers must be entered in the spaces indicated by the --RR1-- and the --RR2--. The new transfer penalty must be entered in the space indicated by the --PENALTY--, while the node name or number must be entered under the ---NODE--------- heading. The first railroad identification, RR1, is required. This information can be entered as the abbreviated railroad name or as the railroad number. If nothing is entered for the second railroad (RR2), the program will assume that the new transfer penalty will be assigned to all of the interchanges involving the first railroad, that is, all interchanges between the first railroad and all other railroads and waterway networks. When the second railroad is entered, only the transfer penalty for interchanges between the two railroads is changed.

If the transfer penalty field is left blank, the program will list all of the transfer points requested in the data line and the current value of their transfer penalty. The node field is used to indicate that the transfer penalties at a specific node are to be changed. When the node field is left blank, all of the transfers between the specified railroads will be changed.

The transfer penalty is a directional parameter. The transfer penalty can vary with the direction of the transfer. For example, it is possible to assign the transfer penalty of 500 units for freight being transferred from the UP to the BN railroad while the transfer penalty for freight moving from the BN to the UP can have a transfer penalty of only 100 units. The user has complete flexibility in assigning transfer penalties with this command. If the new transfer penalty is to apply to both directions, an asterisk must be placed under the asterisk in the heading. If the transfer penalty is to be directional, nothing appears under the asterisk in the heading. The program assumes the direction to be from RR1 to RR2.

Sample data input for the **transfer** command are illustrated in four examples in Fig. 29. The first example defines the transfer penalty between the Burlington Northern railroad and the Wisconsin Central railroad at four nodes to be 1000 units. The asterisk indicates that the transfer penalty applies in both directions. After the data were entered, the program lists all of the transfer points between the BN and WC railroads and includes the new transfer penalties.

The second example is a request for a listing of the transfers and their transfer penalties at Minneapolis, Minnesota. The Minneapolis node was specified by entering the node number (**9829**) under the NODE heading. The node could also have been defined as **minn,mn**, and the program would prompt the user to select the proper node if more than one match is found. The listing of transfer points was specified by leaving the PENALTY field blank.

The third example shows how to specify a transfer penalty of 750 units for transfers from the Burlington Northern railroad at St. Paul, Minnesota. In this example, the name of the second railroad is blank; hence, the new transfer penalty applies to all railroads that can interchange freight with the BN at this node. The new transfer penalties are directional (no asterisk included before the RR1 name) and apply only to freight moving from the BN to another railroad. In this case, the node was specified by entering the node name, **st paul,mn**.

The final example shows another summary of transfer penalties that were changed in this set of examples. Since a transfer penalty was not supplied, a summary will be printed. In addition, no node was supplied, so the summary will include all interchanges between the WC and the BN railroads. Since an asterisk was included before the RR1 name, the transfer penalties are shown for both directions. Note the directional dependence of the transfers at St. Paul, Minnesota.

The user can return to the **COMMAND?** level by entering a **<cr>** after the

ENTER POINT > >

prompt. All changes in the transfer penalties apply only to the current INTERLINE session. When the program is started at a later time, all transfer penalties will be reset to their default value.

3.3.9 Help

The user may enter **help** at any point in the program for assistance in making the appropriate responses. The program will print a brief explanation of the input required and then proceed with the program operation. The **help** command is illustrated in Fig. 30. In the first

```
COMMAND?   transfer
```

[Example 1, Setting two way transfer penalty]

```
                    *--RR1-- --RR2-- --PENALTY--   ----NODE----------
 ENTER POINT  >>  *  bn       wc       1000
      2 86   4217   BN   WC    CHICAGO                IL     1000.0
      2 86   9527   BN   WC    SUPERIOR               WI     1000.0
      2 86   9829   BN   WC    MINNEAPOLIS            MN     1000.0
      2 86   9830   BN   WC    ST PAUL                MN     1000.0
     86  2   4217   WC   BN    CHICAGO                IL     1000.0
     86  2   9527   WC   BN    SUPERIOR               WI     1000.0
     86  2   9829   WC   BN    MINNEAPOLIS            MN     1000.0
     86  2   9830   WC   BN    ST PAUL                MN     1000.0
      2 86 *    0*  BN   WC    **    8 CHANGES MADE  **   1000.0
```

[Example 2, Summary of interchanges and transfer penalties at a node]

```
                    *--RR1-- --RR2-- --PENALTY--   ----NODE----------
 ENTER POINT  >>  *  bn       wc                      9829
      2 86   9829   BN   WC    MINNEAPOLIS            MN     1000.0
     86  2   9829   WC   BN    MINNEAPOLIS            MN     1000.0
```

[Example 3, Setting one-way transfer penalty at a node]

```
                    *--RR1-- --RR2-- --PENALTY--   ----NODE----------
 ENTER POINT  >>     bn                750          st paul,mn
++ MULTIPLE NODES WITH NAME:  mn  st paul
       9830   ST PAUL              MN
       9785   ST PAUL PARK         MN
      17123   ST PAUL; PORT OF     MN
 ENTER NODE NUMBER >>  9830
      2  3   9830   BN   CNW   ST PAUL                MN      750.0
      2 11   9830   BN   SOO   ST PAUL                MN      750.0
      2 86   9830   BN   WC    ST PAUL                MN      750.0
      2 93   9830   BN   <TR>  ST PAUL                MN      750.0
      2 95   9830   BN   *BRG  ST PAUL                MN      750.0
      2 97 * 9830*  BN   -     **    5 CHANGES MADE  **    750.0
```

[Summary of interchange points and transfer penalties]

```
                    *--RR1-- --RR2-- --PENALTY--   ----NODE----------
 ENTER POINT  >>  *  wc       bn
     86  2   4217   WC   BN    CHICAGO                IL     1000.0
     86  2   9527   WC   BN    SUPERIOR               WI     1000.0
     86  2   9829   WC   BN    MINNEAPOLIS            MN     1000.0
     86  2   9830   WC   BN    ST PAUL                MN     1000.0
      2 86   4217   BN   WC    CHICAGO                IL     1000.0
      2 86   9527   BN   WC    SUPERIOR               WI     1000.0
      2 86   9829   BN   WC    MINNEAPOLIS            MN     1000.0
      2 86   9830   BN   WC    ST PAUL                MN      750.0
```

Fig. 29. Changing transfer penalties.

```
COMMAND?   help
    Available commands are:
      ROUte - Find routes.          STOp - End program.
      REROute - from same spot.     OPTion - Program options.
      NODe n - Remove nodes.        HELP keyword - Lists HELP file for
      LINk - Alter links.             a procedure.  HELP can be entered
      REMove - Remove RRs.            during procedures if you need
      TRAnsfer - Change or list       more information.
        transfer impedances.
    A null line will terminate all procedures.

COMMAND?   reroute

 ENTER DESTINATION  >>  help
    Type a node number; or a node name, comma, and state.  You may fol-
      low with the RR: R(name/num).  A null line will terminate the des-
      tination list and return you to command level.

 ENTER DESTINATION  >>
```

Fig. 30. The HELP command.

example in Fig. 30, **help** was entered at the **COMMAND?** prompt. The program responded with a brief description of the commands and then repeated the **COMMAND?** prompt. The second example uses the **help** command at the **ENTER DESTINATION > >** prompt in the **reroute** command. Again a brief explanation of the needed input is printed.

For more detailed information about the commands, the user is referred to the appropriate sections of the User's Manual.

3.3.10 Stop

This command terminates program operation and returns the user to the directory level. The program will display the following message:

NORMAL END OF PROGRAM.

At any point in the program operation, the user may enter a **<cr>** (one or more times) to return to the **COMMAND?** level. At that time, the user may exit the program by typing **stop** or re-enter any of the program commands and begin again.

REFERENCES

1. Peterson, B. E. *INTERLINE, A Railroad Routing Model: Program Description and User's Manual*, ORNL/TM-8944, November 1985.

2. Armstrong, J. H. *The Railroad-What It Is, What It Does*, Omaha, NE: Simmons-Boardman, 1977.

3. *Railroad Facts*, 1989 Edition, Washington, DC: Association of American Railroads, 1989.

4. Infomatics Inc., "FRA Integrated Rail Network Study," prepared for the Federal Railroad Administration, December 1979.

5. Newhauser, K. S.; Kanipe, F. L. *RADTRAN 4: Volume 3, User Guide*, SAND89-2370, January 1992.

6. Even, S. *Graph Algorithms*, Rockville, MD: Computer Science Press, 1979.

7. Reingold, E. M.; Nieverelt, J.; Deo, N. *Combinatorial Algorithms: Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1977.

8. Durfee, R. C.; Coleman, P. R. *Population Distribution Analyses for Nuclear Power Plant Siting*, NUREG/CR-3056, December 1983.

9. *Handy Railroad Atlas of the United States*, Chicago, IL: Rand McNally & Company, 1988.

Appendix A. SUBNETWORK LISTING

Appendix A. SUBNETWORK LISTING

Table A.1 lists the 96 subnetworks that are contained in Network 10 of the INTERLINE model. The first 14 subnetworks comprise the major U.S. railroad systems, listed in alphabetical order. Some of the subnetworks are combinations of several individual companies that essentially operate as a single corporate entity (e.g., Southern Pacific). For these companies, the subsidiary railroad companies are included in parentheses. Subnetworks 15 through 91 represent regional and longer shortline railroad companies, again listed in alphabetical order of their abbreviation. A number of the smaller shortline railroads are combined into a generic subnetwork (number 92). Likewise, the smaller terminal railroads are also grouped into another generic subnetwork (number 93). The National Railroad Passenger Corporation, more commonly known as AMTRAK, is subnetwork 94.

Inland and coastal waterways are divided into two separate systems in the INTERLINE data base. The inland waterways and intracoastal waterways are included as the Barge/Intracoastal Waterway system, which is abbreviated *BRG and is subnetwork 95. Deep water routes providing barge service along the Pacific and Atlantic coasts, the Gulf of Mexico, the Panama Canal, and the Great Lakes are combined into a single classification, Merchant Marine (subnetwork 96, which is abbreviated as *M-M).

Over 500 railroads are represented in Network 10. It should be noted that new versions of the INTERLINE network will be issued periodically to reflect future changes in the U.S. railroad system.

Table A.1. Network 10 subnetworks

Subnetwork number	Subnetwork abbreviation	Railroad name (and subsidiaries)
1	ATSF	Atchison, Topeka & Santa Fe Railway
2	BN	Burlington Northern Railroad
3	CNW	Chicago & North Western Transportation Company (Western Railroad Properties)
4	CR	Conrail - Consolidated Rail Corporation (Monongahela)
5	CSXT	CSX Transportation, Inc. (Baltimore & Ohio Chicago Terminal Railroad; Richmond, Fredericksburg and Potomac Railway)

Table A.1. (continued)

Subnetwork number	Subnetwork abbreviation	Railroad name (and subsidiaries)
6	FEC	Florida East Coast Railway
7	GTW	Grand Trunk Western Railroad
8	IC	Illinois Central Railroad
9	KCS	Kansas City Southern Railway (Arkansas Western Railway; Fort Smith & Van Buren Railway; Louisiana & Arkansas Railway)
10	NS	Norfolk Southern Railway
11	SOO	Soo Line Railroad
12	SP	Southern Pacific Transportation Company (Denver and Rio Grande Western Railroad; Northwestern Pacific Railroad; Southern Pacific Chicago & St Louis; St Louis Southwestern Railway)
13	ST	Springfield Terminal Railway (Boston & Maine; Maine Central Railroad; Portland Terminal)
14	UP	Union Pacific Railroad (Doniphan, Kensett & Searcy; Missouri Pacific Railroad)
15	ACWR	Aberdeen, Carolina & Western Railway
16	ALY	Allegheny Railroad
17	AM	Arkansas & Missouri Railroad
18	ARZC	Arizona & California Railroad
19	AUNW	Austin & Northwestern Railroad (Texas-New Mexico Division)
20	BAR	Bangor & Aroostook Railroad
21	BCLR	Bay Colony Railroad
22	BLE	Bessemer & Lake Erie Railroad (Union Railroad)
23	BPRR	Buffalo & Pittsburgh Railroad
24	BRC	Belt Railway Company of Chicago
25	CAGY	Columbus & Greenville Railway
26	CIM	Chicago & Illinois Midland Railway
27	CMGN	Central Michigan Railway
28	CC	Chicago, Central & Pacific Railroad (Cedar River Railroad)
29	CN	Canadian National Railways (Central Vermont Railway; Duluth, Winnipeg & Pacific Railway)
30	CP	CP Rail (D & H Corporation)
31	CSS	Chicago, Southshore & South Bend Railroad
32	DM	Detroit & Mackinac Railway

Table A.1. (continued)

Subnetwork number	Subnetwork abbreviation	Railroad name (and subsidiaries)
33	DME	Dakota, Minnesota & Eastern Railroad
34	DMIR	Duluth, Missabe & Iron Range Railway
35	DMVW	Dakota, Missouri Valley & Western Railroad
36	DSRC	Dakota Southern Railway
37	EJE	Elgin, Joliet & Eastern Railway
38	ELS	Escanaba & Lake Superior Railroad
39	EUKA	Eureka Southern Railroad
40	FRVR	Fox River Valley Railroad
41	GBW	Green Bay Western Railroad
42	GC	Georgia Central Railway
43	GNBC	Grainbelt Corporation
44	GWR	Great Western Railway
45	GWWR	Gateway Western Railway
46	HESR	Huron & Eastern Railway
47	IAIS	Iowa Interstate Railroad
48	IANR	Iowa Northern Railway
49	IHB	Indiana Harbor Belt Railroad
50	IHRC	Indiana Hi-Rail Corporation
51	INOH	Indiana & Ohio Railroad (Indiana & Ohio Central Railroad; Indiana & Ohio Eastern Railroad; Indiana & Ohio Railway)
52	INRD	Indiana Rail Road
53	KBSR	Kankakee, Beaverville & Southern Railroad
54	KCTV	KCT Railway
55	KRR	Kiamichi Railroad (Chaparral Railroad)
56	KSW	Kansas Southwestern Railway
57	KYLE	Kyle Railroad
58	LI	Long Island Rail Road
59	MRL	Montana Rail Link
60	MSRC	MidSouth Rail Corporation (MidLouisiana Rail; SouthRail)
61	NERR	Nashville & Eastern Railroad
62	NYSW	New York, Susquehanna & Western Railway (Cooperstown & Charlotte Valley Railway; Central New York Railroad; Staten Island Railway)

Table A.1. (continued)

Subnetwork number	Subnetwork abbreviation	Railroad name (and subsidiaries)
63	PAL	Paducah & Louisville Railway
64	PLE	Pittsburgh & Lake Erie Railroad (Youngstown & Southern Railway)
65	PW	Providence & Worcester Railroad
66	RBMN	Reading, Blue Mountain & Northern Railroad
67	RJCR	R J Corman Railroad (Cleveland Line; Memphis Line)
68	RRVW	Red River Valley & Western Railroad
69	RSR	Rochester & Southern Railroad
70	SCRF	South Carolina Central Railroad (Carolina Piedmont Division; Georgia & Alabama Division; Georgia Great Southern Division; Georgia Southwestern Division)
71	SDIY	San Diego & Imperial Valley Railroad
72	SEKR	Southeast Kansas Railroad
73	SKOL	South Kansas & Oklahoma Railroad (Osage Railroad)
74	SLR	St Lawrence & Atlantic Railroad
75	SW	Southwestern Railroad
76	TCWR	Twin Cities & Western Railroad
77	TM	Texas Mexican Railway
78	TNER	Mid-Michigan Railroad (Texas Northeastern Division)
79	TPW	Toledo, Peoria & Western Railway
80	TRRA	Terminal Railroad Association of St Louis
81	TSBY	Tuscola & Saginaw Bay Railway
82	TXOR	Texas & Oklahoma Railroad
83	USG	United States Government
84	UTAH	Utah Railway
85	VTR	Vermont Railway (Clarendon & Pittsford Railroad)
86	WC	Wisconsin Central Limited
87	WCRC	Washington Central Railroad
88	WE	Wheeling & Lake Erie Railway
89	WICT	Wisconsin and Calumet Railroad
90	WSOR	Wisconsin Southern Railroad
91	WSS	Winston-Salem Southbound Railway (High Point, Thomasville & Denton Railroad)

Table A.1. (continued)

Subnetwork number	Subnetwork abbreviation	Railroad name (and subsidiaries)
92	<C3>	Other Class III [Shortline] Railroads
93	<TR>	Terminal Railroads
94	AMTK	AMTRAK - National Railroad Passenger Corporation
95	*BRG	Barge/Intracoastal Waterways
96	*M-M	Merchant Marine

Appendix B. ADVANCED FEATURES

Appendix B. ADVANCED FEATURES

Several advanced features of INTERLINE are not discussed in Sect. 3 of the User's Manual primarily because the normal use of the model does not require a knowledge of these aspects of the program. This appendix will discuss the use and operations of these more complex features.

The **expert** option enables the user to change the originating railroad advantage, change the mainline classification factors, change the alternative routing penalty factors on previously used links and transfers, perform directed routing, identify the subnetworks being examined during routing, and output a map file of calculated routes. Only the **option** and **route** commands are affected by the use of the **expert** option, and they will be discussed in that order. A discussion of how to specify different types of route listings is also included in this appendix.

B.1 EXPERT CAPABILITY WITHIN THE OPTION COMMAND

The expert capability is activated by typing an **e** at the **option** command prompt (see Fig. 22 in Sect. 3.3.4). The expert capability opens access to another capability—the map file capability. This capability is discussed in Sect. B.1.1.

B.1.1 Map File

When the map file capability is activated, the user has the option of generating a sequential list of node and link numbers for a route in a data file, INTERLIN.OUT. The INTERLIN.OUT file is always created when the program is run. If the map file option is never activated, the file will not contain any data. This data file is used with an auxiliary ORNL program (not included in the INTERLINE distribution diskette) to generate maps of routes.

Since the general user would not need the map file, **it is recommended that the map file capability not be activated.**

When the map file capability has been activated, an additional question is posed to the user after the question regarding a hard copy of the route. The prompt asks:

DO YOU WANT TO WRITE THIS ROUTE TO THE MAP FILE (Y/N) ?

[The default for this prompt is no.]

If the user types a **y**, this route will be written to the map file.

A number of routes can be placed in the INTERLIN.OUT file. However, if the INTERLINE session is terminated and restarted at a later time, any new route information will overwrite data previously contained in the INTERLIN.OUT file.

B.1.2 Alternative Routing with the Expert Capability

When the expert capability has been activated, additional prompts will appear when the alternative routing capability is activated. These additional prompts allow the user to change the default values for the penalties placed on previously used links and transfers. After the user has activated both expert and alternative routing, in that order, the program will prompt:

CURRENT PENALTY ON PREVIOUSLY USED LINKS IS 20%
CURRENT MULTIPLICATIVE PENALTY ON PREVIOUSLY USED TRANSFERS IS 2.0
DO YOU WISH TO CHANGE THESE VALUES (Y/N) ?

[The default for this prompt is no.]

If the user responds affirmatively to this prompt, the following two prompts will appear:

ENTER NEW LINK PENALTY >
ENTER NEW TRANSFER PENALTY >

The user should enter a value at both of these prompts. If no value is entered, the program will assign a value of 0. If both values are set to 0, the alternative routing capability is effectively disabled, since no penalties will be assigned to previously used links and transfer points.

B.2 EXPERT CAPABILITY WITHIN THE ROUTE COMMAND

The expert capability, when activated, has several effects upon the **route** command. These will be discussed in the order that they appear when the user issues the **route** command.

B.2.1 Originating Railroad Advantage and Mainline Penalties

When the expert capability is activated, the **route** command will issue two additional prompts regarding the originating railroad advantage and the mainline classification penalties. The prompts appear as follows:

ENTER ORIGINATING RR ADVANTAGE FACTOR > >
ENTER MLC PENALTY FACTORS
---1- ---2- ---3- ---4- ---5- ---6- ---7- ---8- ---9-

At the first prompt the user can enter a new originating railroad advantage. If a zero or a **<cr>** is entered, the default value of 0.8 will be used. When a new value is entered, that value will be used for all subsequent routes until the originating railroad advantage is changed again or the

program session is terminated. The mainline classification penalty factors can be changed by entering new values after this prompt. New values should be entered under the appropriate dashed areas. If a zero or a <cr> is entered below any of the categories, the default value for that item will remain in effect. Table B.1 identifies each of the categories and lists the default values.

Please note that the initial default values are always in effect whenever the INTERLINE model is started.

Table B.1. Mainline classification codes and default values

Class	Mainline classification	Default value
1	A-mainline	1.0
2	B-mainline	1.2
3	A-branchline	1.9
4	B-branchline	4.0
5	Not used	500.0
6	Inland waterways	0.7
7	Deep water routes	0.6
8	Not used	500.0
9	Not used	500.0

B.2.2 Directed Routing

One of the features of the INTERLINE model is the ability to let the user designate the sequence of railroads to be used in the route. If the expert capability has been activated, the program will issue the following prompt after the origin and destination has been entered:

ENTER RR SEQUENCE >>.

At this prompt, the user can enter either railroad abbreviations or the subnetwork numbers (at this prompt, abbreviations with numbers may not be mixed) in the sequence in which the user wishes the route to proceed. If the user responds with a <cr>, the model will calculate the route in the normal fashion.

The directed routing feature is useful if a particular sequence of railroads is desired for a route. For example, assume the origin of a route is a node served by Conrail, the destination

is on the Union Pacific system, and the user wants the route to travel on the Conrail, Chicago and North Western, and Union Pacific railroads. The directed routing feature can be used to specify this sequence of railroads. With this feature, it is the user's responsibility to ensure that the railroads specified define a connected network between the origin and destination. If a connected network does not exist, the program cannot find a route.

Please note that if the user issues a **reroute** command after performing a directed route, the railroad sequence prompt is not repeated and **the previous sequence of railroads provided by the user will be used**. The **route** command must be used instead of the **reroute** command to remove a previously defined directed routing railroad sequence.

B.2.3 Subnetwork Progression

With the expert capability activated, the INTERLINE model will identify the subnetworks that are being processed by the shortest path algorithm. During normal operation, the program will indicate that the calculations are proceeding by printing

ROUTING ..... .

When a dot appears, a subnetwork has been analyzed by the shortest path algorithm. If the expert capability is activated, the program will print the subnetwork number that is being examined. When the shortest path algorithm has finished with that subnetwork, the dot will appear. This feature has no other purpose than to provide the user with the assurance that the program is generating a route.

B.3 ADDITIONAL ROUTE LISTING OPTIONS

The INTERLINE model issues several prompts regarding route listings during the operation of the **route**, **reroute**, and **retype** commands. These prompts are as follows:

DO YOU WANT TO SEE A ROUTE LISTING (Y/N) ?

DO YOU WANT TO SEE A MORE DETAILED ROUTE LISTING (Y/N) ?

DO YOU WANT A HARD COPY OF THIS ROUTE INFORMATION (Y/N) ?

Under most circumstances, these listings should provide sufficient route information to most users. The first prompt lists the origin, destination, major cities (with population over 50,000 people), and transfer locations along the route. The detailed route listing and the hard copy prompts list additional nodes, which are locations of importance for railroads, within the route.

Additional route listing information can be obtained at these prompts. To access the other route listing options, the user should respond to these prompts with a y#, where the # represents either a positive or negative number from 1 to 9. Providing a code at any of these prompts overrides the default for that prompt. Table B.2 lists the functions of the various codes.

As identified in Table B.2, codes 2 and 3 are the default values for the prompts for route listings in the model. Code 1 provides the minimum amount of information about the route. If the detailed (Code 3) listing does not provide sufficient information, the user may wish to use code 4. This code will identify all nodes within the route which have named nodes. Code 5 will list all nodes including both named and numbered nodes. The final two codes are similar—they provide information about the links in the route. This information may provide some useful information to a specialized user, such as identifying which portion of a route is B-mainline vs A-mainline. The user should be aware that the use of codes 4 through 9 may generate many pages of output.

Table B.2. Route listing codes

Code[a]	Function
1	Lists only origin, destination, and any transfer nodes.
2	Lists major cities (cities with population exceeding 50,000), in addition to items for Code 1. (This code is used for the standard route listing.)
3	Lists important railroad locations (locations where railroad lines cross or divide to form two lines), in addition to items for Code 2. (This code is used for the detailed and hard copy listing.)
4	Lists all named nodes in route.
5	Lists all nodes (including those nodes without names) in route.
6	Lists all named nodes and provides information on individual links (distance, mainline classification, coded traffic density, and owner of segment if railroad has trackage rights on this link).
9	Lists all nodes and provides information on individual links (as listed for Code 6).

[a]If the value 7 or 8 is entered, it is interpreted as a 6. When the code number is entered as a positive number, the listing will be continuous. If the code number is negative, the program will issue the **CONTINUE?** prompt sequence. All listings intended to be viewed on the screen should be entered as a negative value.

B.4 ADVANCED CONCEPTS FOR NODE AND LINK DELETIONS

The node and link blocking features of the INTERLINE model were discussed in Sects. 3.3.5 and 3.3.6. It was noted in those sections that the blocking of nodes and links applies

only to the current session of the model. Instances may arise when a user may wish to have a set of nodes and links blocked for multiple sessions. It can be very time-consuming to enter the same blockages every time the model is started. If the user wishes to use the same modified network for multiple sessions, the procedure outlined in this section should be followed.

At the start of an INTERLINE session, the batch file (discussed in Sect. 3.1) copies the original network file (RRN10BO.LLS) to a temporary network file (RRN10B.LLS). At the conclusions of the batch file, the temporary network file is deleted. If the user wishes to use a modified network file that contains node and link blockages, the user should perform these steps:

1. Copy the RRN10BO.LLS file to the RRN10B.LLS file.

2. Edit the RAILROAD.BAT file, and add **rem** and a space to the beginning of the line that copies the network file to the temporary network file and to the beginning of the line that deletes the temporary network file. The addition of **rem** to the beginning of a line causes that line to be treated as a remark.

3. Start the INTERLINE model by typing **railroad**, as usual, and make the changes to the network using the **node** or **link** command. The user can exit and restart the program, and all node and link blockages will still be in effect.

4. Delete the RRN10B.LLS file and remove the remarked lines (see Step 2 above) in the RAILROAD.BAT file when finished using this customized version.

The user should note that the INTERLINE program needs to find the RRN10B.LLS file in order to operate. The RRN10BO.LLS is the master network file and contains some links that are blocked due to operational patterns of some railroads. If the master network file is modified, the INTERLINE model should be reinstalled from the distribution diskette.

INTERNAL DISTRIBUTION

1. J.M. Begovich
2. M.S. Bronzini
3. A.G. Croff
4. R.M. Davis
5. R.C. Durfee
6. W. Fulkerson
7. L.D. Gasaway
8. R.K. Genung
9. R.D. Halverson
10. R.B. Honea
11-20. P.E. Johnson
21-25. D.S. Joy
26. S.B. Ludwig
27. A.P. Malinauskas
28. D.H. Pike
29. R.B. Pope
30. R.R. Rawl
31. W.J. Reich
32. M.K. Savage
33. R.M. Walker
34. G.E. Whitesides
35. Central Research Library
36. ORNL Y-12 Technical Library, Document Reference Section
37. ORNL Patent Section
38-39. Laboratory Records
40. Laboratory Records—RC

EXTERNAL DISTRIBUTION

41-43. R. Amato, Bettis Atomic Power Laboratory, MS 32B, Room 211, P.O. Box 79, Pittsburgh, PA 15122
44. J. Arthur III, DOE WIPP Joint Integration Office, 1 Americas Parkway, Albuquerque, NM 87110
45. B. Biwer, Argonne National Laboratories, EID/900, 9600 South Cass Ave., Argonne, IL 60439
46. L.G. Blalock, Transportation Management Division, EM-561, U.S. Department of Energy, Washington, DC 20585
47-76. J.W. Cashwell, Sandia National Laboratories, Division 6641, Albuquerque, NM 87123
77. D.B. Clarke, University of Tennessee Transportation Center, 10521 Research Dr., Suite 200, Knoxville, TN 37932
78. S. Conner, NUS Corporation, 900 Trail Ridge Rd., Aiken, SC 29803
79. R. Church, Department of Geography, University of California, Santa Barbara, Santa Barbara, CA 93106-4060
80. R.J. Davis, NUS Corporation, 910 Clopper Road, Gaithersburg, MD 20878
81. L.G. Ellis, Analysas Corp., 151 Lafayette Dr., Suite 110, Oak Ridge, TN 37830
82. R. English, Pittsburgh Naval Reactors, P.O. Box 109, West Mifflin, PA 15122
83. S. Everett, S.M. Stoller Corporation, 1717 Louisiana Ave NE, Albuquerque, NM 87110
84-87. L.M. Green, CRWMS M&O, 2650 Park Tower Dr., Vienna, VA 22189
88. C.R. Hamberger, SAIC, P.O. Box 2501, Oak Ridge, TN 37831
89. J.M. Jacobi, 3700 Southland Ave., Apt. D-10, Knoxville, TN 37919

90. R. Johnson, CRWMS M&O, 2650 Park Tower Dr., Vienna, VA 22189
91. M. Lauer, Dames and Moore, 5400 Glenwood Ave., Suite 205, Raleigh, NC 27612
92. M. Lazaro, Argonne National Laboratories, EID/900, 9600 South Cass Ave., Argonne, IL 60439
93. P.B. Lester, Oak Ridge Field Office, U.S. Department of Energy, P.O. Box 2001, Oak Ridge, TN 37831-8620
94. J. Lockwood, Proteus Corporation, 10010 Indian School Road NE, Albuquerque, NM 87112
95. M&O Technical Information Center, CRWMS M&O, 2650 Park Tower Dr., Vienna, VA 22189
96-99. J.T. Mayer, CRWMS M&O, 2650 Park Tower DR., Vienna, VA 22189
100. M.F. Moerchen, SAIC, P.O. Box 2501, Oak Ridge, TN 37831
101. T. Molina, Sandia National Laboratories, Division 6641, Albuquerque, NM 87123
102-111. R. Pineda, Albuquerque Field Office, U.S. Department of Energy, Albuquerque, NM 87115
112. T. Poasko, Argonne National Laboratories, EID/900, 9600 South Cass Ave., Argonne, IL 60439
113. J.H. Portsmouth, Westinghouse Hanford Company, Richland, WA 99352
114. M.D. Ruska, EG&G Idaho, Inc., P.O. Box 1625, Idaho Falls, ID 83415-4113
115. A.L. Sandoval, Proteus Corporation, 10010 Indian School Road NE, Albuquerque, NM 87112
116. R.P. Sandoval, Sandia National Laboratories, Division 6641, Albuquerque, NM 87123
117. W. Schofield, EBASCO Environmental, Plaza Center Bldg., 10900 N.E. 8th Street, Bellevue, WA 98004-4405
118. D.J. Size, Albuquerque Field Office, U.S. Department of Energy, Albuquerque, NM 87115
119. G.L. Sullivan, System Management Services, Inc., 685 Emery Valley Road, Oak Ridge, TN 37831
120. B.R. Teer, CRWMS M&O, 2650 Park Tower Dr., Vienna, VA 22189
121-125. T.M. Thomas, Transportation Management Division, EM-561, U.S. Department of Energy, Washington, DC 20585
126. Transportation Management Division Library, EM-561, U.S. Department of Energy, Washington, DC 20585
127. G.W. Wecksung, Proteus Corporation, 10010 Indian School Road NE, Albuquerque, NM 87112
128. S. Woolfolk, SAIC, 14062 Denver West Parkway, Suite 200, Golden, CO 80401

129. Office of Assistant Manager, Energy Research and Development, DOE-OR, P.O. Box 2001, Oak Ridge, TN 37831

130-131. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831