



3 4456 0374726 6

ornl

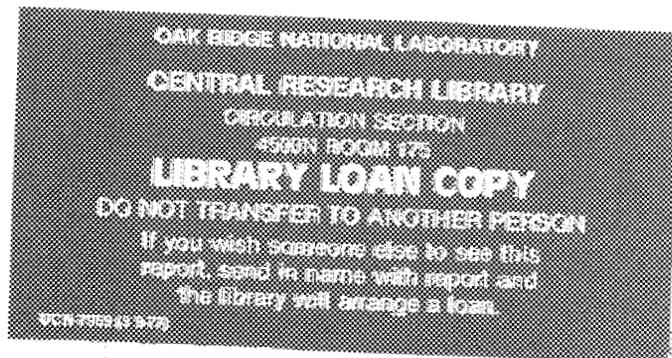
ORNL/TM-12124

**OAK RIDGE
NATIONAL
LABORATORY**

MARTIN MARIETTA

HIGHWAY 3.1 — An Enhanced Highway Routing Model: Program Description, Methodology, and Revised User's Manual

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



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

Chemical Technology Division

HIGHWAY 3.1—
AN ENHANCED HIGHWAY 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



CONTENTS

ACKNOWLEDGMENTS	ix
ABSTRACT	xi
1. PROGRAM DESCRIPTION	1
1.1 DEVELOPMENT OF THE ROUTING MODEL	2
1.2 HIGHWAY DATA BASE	3
1.3 NAMING CONVENTIONS EMPLOYED	3
1.4 GENERAL ROUTING CONSIDERATIONS	5
1.4.1 Setting Routing Criteria	6
1.4.2 Toll Bias	9
1.4.3 Number of Drivers	9
1.4.4 Routing Constraints	9
1.4.5 Routes for Highway Route-Controlled Quantities of Radioactive Materials (HM-164 Routes)	16
1.4.6 Alternative Routing	17
2. METHODOLOGY	21
2.1 ORGANIZATION OF THE NETWORK FOR ROUTING ANALYSIS	21
2.1.1 State Highway Network Representation	21
2.1.2 Border Crossings	22
2.2 THE HIGHWAY ROUTING ALGORITHM	24
2.2.1 Routing Nomenclature	24
2.2.2 The Routing Objective Function	25
2.2.3 Shortest Path Algorithm Overview	25
2.2.4 The HIGHWAY Routing Algorithm	28
2.2.4.1 Initialization Sequence	28
2.2.4.2 Main Loop	29
2.2.5 Sample Route	30
2.3 PROGRAM OPTIONS	34
2.3.1 Link Avoidance	34
2.3.2 Node Avoidance	34
2.3.3 Route Weighting by Highway Sign Type	38
2.3.4 Route Weighting by Highway Lane Type	38
2.3.5 State Prohibition	39
2.3.6 Weighting by Other Link Attributes	39
2.3.7 HM-164 Routing	40
2.3.8 Alternative Routing	40
2.3.9 Population Density	41
3. HIGHWAY PROGRAM USER'S MANUAL	45
3.1 COMMANDS	47
3.1.1 Help	48

3.1.2	Route	48
3.1.2.1	Changing the Default Routing Parameters	52
3.1.3	Reroute	54
3.1.4	Retype	57
3.1.5	Options	57
3.1.5.1	AL (Alternative Route) Code	58
3.1.5.2	DE (Detailed Route Listing) Code	59
3.1.5.3	DR (Number of Drivers) Code	62
3.1.5.4	HM (HM-164 Routing Constraint) Code	62
3.1.5.5	OP (Parameter Values) Code	63
3.1.5.6	PD (Population Density) Code	64
3.1.5.7	PW (Preferred Route Weight) Code	67
3.1.5.8	SH (Shipment Parameters) Code	68
3.1.5.9	RE (Return) Code	70
3.1.6	Advanced Routing Commands	70
3.1.6.1	Remove	72
3.1.6.2	D Node	74
3.1.6.3	D Link	78
3.1.6.4	Reset	82
3.1.6.5	Links	82
3.1.7	Stop	84
REFERENCES		85
Appendix. SPECIALIZED FEATURES		87

LIST OF FIGURES

1.	Highway and intersection naming conventions	4
2.	Comparison of commercial vs shortest route between Morris, Illinois, and Richland, Washington	8
3.	Commercial route between Manchester, Iowa, and Dalhart, Texas	10
4.	Route between Manchester, Iowa, and Dalhart, Texas, that bypasses Missouri	12
5.	Route between Manchester, Iowa, and Dalhart, Texas, that bypasses Wichita, Kansas	13
6.	Route between Manchester, Iowa, and Dalhart, Texas, that bypasses urban areas with populations greater than 100,000	14
7.	Route between Manchester, Iowa, and Dalhart, Texas, that bypasses I-35 between Des Moines, Iowa, and the Missouri state line	15
8.	Route between Manchester, Iowa, and Dalhart, Texas, that maximizes the use of Interstate highways	18
9.	Alternative routes between Marshall, Michigan, and Phoenix, Arizona	19
10.	Graphic illustration of (a) an undecomposed network, (b) a border node creation, and (c) a decomposed network	23
11.	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	27
12.	Example of network structure	31
13.	Pivot through state 1 at (a) the start and (b) the end of iteration	32
14.	Pivot through state 2 at (a) the start and (b) the end of iteration	33
15.	Pivot through state 3 at (a) the start and (b) the end of iteration	35
16.	Pivot through state 4 at (a) the start and (b) the end of iteration	36
17.	Complete path tree	37
18.	Network impedance after increasing impedances on the base route	42
19.	Completed path tree—alternative routing	43
20.	Path comparison—normal vs alternative routing	44
21.	Sample of opening screen	46
22.	Toll bias penalty prompt	46
23.	Command list	48
24.	Sample output generated by using the route command	49
25.	Example of changing routing parameters	54
26.	Use of the reroute command	56
27.	Alternative routing example	60
28.	Examples of standard and detailed route listings	61
29.	Use of the dr code	62
30.	Output for op code—current default values of routing parameters	63
31.	Output for op code—routing parameters for calculating HM-164 routes	65
32.	Routing calculation including population density table	66
33.	Subcommands functions	69
34.	Use of check and set subcommands	70
35.	Example of advanced command list	71
36.	Example of remove command functions	73
37.	Sample of link deletion notice	74

38.	Use of the d node command to remove a single node	75
39.	Use of the d node command to remove a range of nodes	76
40.	Use of the d node command to delete both a range of nodes and a specific node	77
41.	Use of the d node command to reinsert nodes into the data base	78
42.	Example of removing links	80
43.	Example of link insertion	81
44.	Example of link and node tracking	83
45.	The stop command	84
A.1.	Expanded op code report including specialized features	90
A.2.	Example of message alerting user of deleted nodes or links	91
A.3.	Route summary showing route impedace	92

LIST OF TABLES

1.	HIGHWAY node naming convention	6
2.	Highway usage for various routes	16
3.	Comparison of alternative routes	20
4.	Sign types and default weights	38
5.	Lane types and default weights	39
6.	Link flags	40
7.	HIGHWAY commands	47
8.	Option codes	57
9.	Advanced commands	71
A.1.	Specialized option codes	89

ACKNOWLEDGMENTS

The HIGHWAY 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 HIGHWAY 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 would like 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

The HIGHWAY program provides a flexible tool for predicting highway routes for transporting radioactive materials in the United States. The HIGHWAY data base is essentially a computerized road atlas that currently describes over 240,000 miles of highways. Complete descriptions of all Interstate System and most U.S. highways (except those that parallel a nearby Interstate highway) are included in the data base. Many of the principal state highways and a number of local and county highways are also identified. The data base also includes locations of nuclear facilities and major airports.

Several different types of routes may be calculated, depending on a set of user-supplied constraints. Routes are calculated by minimizing the total impedance between the origin and the destination. Basically, the impedance is defined as a function of distance and driving time along a particular highway segment. Several routing constraints can be imposed during the calculations. One of the special features of the HIGHWAY model is its ability to calculate routes that maximize use of Interstate System highways. This feature allows the user to predict routes for shipments of radioactive materials that conform to the U.S. Department of Transportation routing regulations. Other features of the model include the ability to predict routes that bypass a specific state, city, town, or highway segment.

Two special features have been incorporated in HIGHWAY, version 3.1. The first is the ability to automatically calculate alternative routes. Frequently, there are a number of routes between the source and destination that vary slightly in distance and estimated driving time. Most routing models will produce only a single route. With the alternative routing feature, the HIGHWAY program offers a selection of different but nearly equal routes. The second special feature is the capability to calculate route-specific population density statistics. The population density distribution is calculated for each highway segment in the route and is reported on a state-by-state basis. The population data used for this calculation are based on the U.S. Bureau of Census block group data.

**HIGHWAY 3.1—
AN ENHANCED HIGHWAY 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.

In response to these needs, a highway routing model (HIGHWAY) has been developed at Oak Ridge National Laboratory (ORNL). The HIGHWAY data base is essentially a computerized road atlas that currently describes over 240,000 miles of roadways. In addition to the basic routing capabilities, the model contains a number of specially designed features that constrain route selection. This document includes a discussion of the routing capabilities included in the HIGHWAY model.

This report contains three sections. The first section presents a general overview of the routing capabilities of the HIGHWAY model with several examples. A short discussion of the HIGHWAY naming conventions for nodes and highway segments is also included. Section 2 presents a description of the methodology used to calculate highway routes. (The casual reader may want to skip this section.) The routing methodology is presented at a simplified level using

*Computing Applications Division

†University of Tennessee, Knoxville, Transportation Center

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 HIGHWAY User's Manual, which includes a detailed discussion of the HIGHWAY routing commands. Numerous examples are included in this section to illustrate how these commands operate. A few of the more specialized commands are discussed in the appendix to this report. Since the general user is not expected to need these commands, their description has been included in an appendix in order to simplify the manual. The casual user needs only to read and understand the material in Sects. 1 and 3.

1.1 DEVELOPMENT OF THE ROUTING MODEL

ORNL has developed a highway routing model in support of the Waste Transportation Studies sponsored by the U.S. Department of Energy (DOE). Initially, a commercially available routing model and a highway data base (COMPU.MAP) were purchased from Logistics Systems, Inc. (Logistics Systems, Inc., was purchased by Rand McNally-TDM, Inc., in 1988.) The computer programs associated with COMPU.MAP were completely rewritten at ORNL. The resulting model, HIGHWAY, provides a flexible tool to be used in predicting highway routes for transporting radioactive materials. The proprietary COMPU.MAP highway data base was retained as part of the HIGHWAY model; however, it has been carefully checked for accuracy and substantially modified to include locations of specific interest to DOE and its contractors.

A number of program and data base enhancements have been made since the publication of the most recent HIGHWAY User's Manual in 1983.¹ The internal structure of the program has been changed to include more detailed prompting statements, which are designed to minimize the casual user's need to refer to the manual. The user's manual portion of the report contains a detailed description of all program commands of interest to non-ORNL users. Version 3.1 of this software contains two major routing enhancements. The first is the ability to automatically calculate alternative routes between any two locations. The second major enhancement is the inclusion of population density information for the highways included in the data base. This information allows the user to generate route-specific population distributions for use in risk analysis codes such as RADTRAN.²

The data base has been enhanced by adding the locations of commercial nuclear power plants and almost 900 major commercial and military airports. The data base is constantly being revised as new roads and locations are added to meet program needs. The user must be aware

of the dynamic nature of the data base. As the data base changes, it is possible that the user will not be able to reproduce all of the examples highlighted in this report.

1.2 HIGHWAY DATA BASE

The HIGHWAY data base is essentially a computerized road atlas that currently contains more than 20,000 highway segments (links) and 13,000 intersections (nodes), which describe over 240,000 miles of roadways in the continental United States. The entire Interstate System and all U.S. highways, except those that parallel a nearby Interstate highway, are included in the data base. Most principal state highways and a number of local and county highways are also identified. Data for each highway segment include highway designation; distance between end points; estimated average driving speed; whether a toll is charged; whether the link is located in an urbanized area containing over 100,000 people; whether the link is part of the routing network for route-controlled quantities of radioactive material; whether trucks are permitted to use the link; whether a tunnel is located on the link; whether the link represents a ferry crossing; and geographical coordinates for the end points.

1.3 NAMING CONVENTIONS EMPLOYED

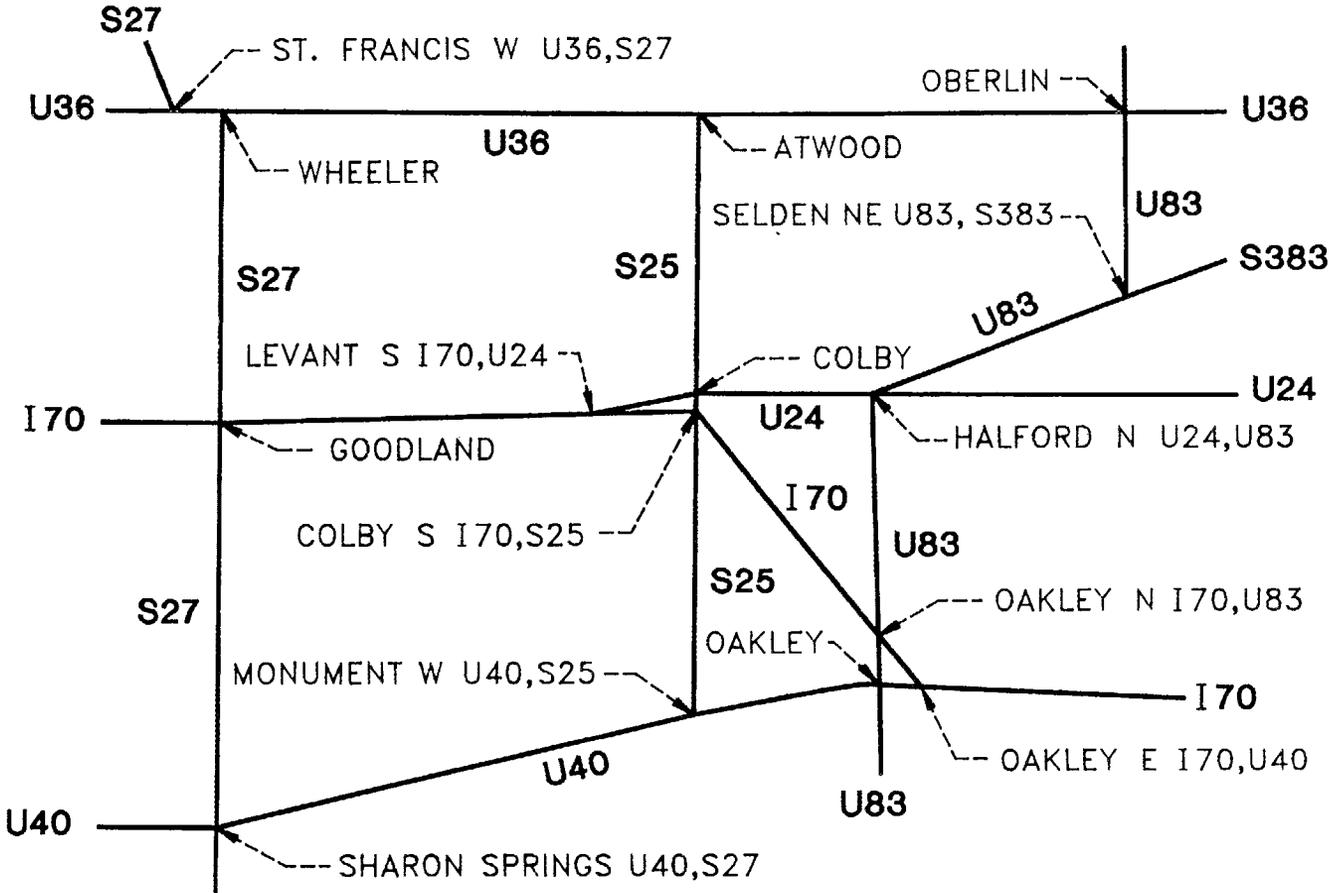
Naming conventions for the HIGHWAY routing model use the first letter of the highway name to identify the road as follows:

- I = Interstate highway (e.g., I40, I75);
- U = U.S. highway (e.g., U66, U45);
- S = state highway (e.g., S147, S62);
- T = turnpikes that are not part of the Interstate System (e.g., TCIM);
- C = county roads (e.g., C162, C1042), and
- L = local roads (e.g., LOCL).

Names of toll roads include a \$ sign (e.g., I35\$), and a # sign is used to indicate that a toll bridge is included in a highway segment.

All highway intersections (nodes) are named. Any name that is not followed by a directional and/or intersection modifier is located at or near the center of a city or town. Names of intersections that are located in the vicinity of a town are followed by a directional modifier (e.g., N, S, W, NW) and generally also include an intersection description (e.g., I35, U64).

Figure 1 illustrates the highway and intersection naming conventions used in the HIGHWAY program. This example shows a portion of the highway network in northwestern



4

Fig. 1. Highway and intersection naming conventions.

Kansas. Interstate 70 (I70) is the main east-west highway, and there is a junction with U.S. 24 (U24) south of the town of Levant, Kansas. This intersection is labeled **Levant S I70,U24** and shows the nearest town (Levant), location of the intersection (south of the town), and the intersection description as Interstate 70 and U.S. Highway 24. It should be noted that there are two junctions in the vicinity of Colby, Kansas. The junction of state highway 25 and U.S. 24 is in the town of Colby, and the junction of state highway 25 and I70 is south of Colby (**Colby S I70,S25**).

A listing of sample HIGHWAY node names is included in Table 1. In general, node names are completely spelled out. When an abbreviation is required, no punctuation is included. The MC CONNELL AFB entry in the table is the node representing the McConnell Air Force Base.

A number of the nodes in the HIGHWAY data base represent airports and commercial nuclear plants. Airports are identified by their three- or four-character representation and by inclusion of the word airport in the node name. The airport serving Manhattan, Kansas, is included in Table 1 as MHK AIRPORT, KS. Nuclear plants are denoted by the letters NP in the name; for example, the Wolf Creek nuclear plant in Kansas is identified as WOLF CREEK NP, KS.

At different times, the HIGHWAY model will request that the user specify a node (e.g., the origin of a route). When a node is specified, both the location and the state must be entered (MANHATTAN,KS). 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 program will then list all nodes in that state which start with the letters provided, and the user will be requested to select the proper node from the list. The list of nodes shown in Table 1 was generated by simply entering M,KS.

1.4 GENERAL ROUTING CONSIDERATIONS

This section discusses the various types of routing constraints that can be used to calculate routes. An understanding of these constraints will provide a better understanding of how the HIGHWAY model determines routes.

Table 1. HIGHWAY node naming convention

Location	Direction modifier	Intersection	State
MADISON		S57 S99	KS
MANHATTAN			KS
MANHATTAN	S	I70 S177	KS
MANKATÓ	E	U36 S14	KS
MARION	E	U56 U77	KS
MARYSVILLE			KS
MARYSVILLE	W	U36 U77	KS
MC CONNELL AFB	W	I135 S15	KS
MC PHERSON			KS
MC PHERSON	E	I135 U56	KS
MEADE	E	U160 U54	KS
MEDICINE LODGE		U160 U281	KS
MHK AIRPORT			KS
MILLER	W	I335 U56	KS
MINNEAPOLIS	E	U81 S106	KS
MINNEOLA			KS
MOLINE	E	U160 S99	KS
MONTROSE	W	U36 S14	KS
MONUMENT	W	U40 S25	KS
MORAN			KS
MOREHEAD	S	U160 U169	KS
MULLINVILLE			KS

1.4.1 Setting Routing Criteria

Several different types of routes may be calculated, and each type can be constrained to follow certain criteria. Routes are calculated by minimizing the total impedance between the origin and the destination. Basically, the impedance is defined as a function of distance and driving time along a particular segment, and the program calculates the set of links between the origin and the destination that minimizes these factors via the Eq. (1):

$$L = \text{Min} \sum_i (\alpha D_i + \beta T_i) , \quad (1)$$

where

- L = total impedance of a route;
- α = distance bias;
- D_i = distance of segment i, miles;
- β = time bias;
- T_i = time required to travel along segment i, minutes.

Setting the values of the time and distance biases defines a particular routing criterion. A number of different types of routes can be predicted. For example, if $\alpha = 1.00$ and $\beta = 0.0$, the shortest possible route will be calculated (giving the entire weight to distance). Setting $\alpha = 0.0$ and $\beta = 1.0$, the most rapid route will be estimated (giving the entire weight to time). A compromise between these extremes is the "commercial" route, where α and β are defined to be 0.3 and 0.7 respectively. This criterion was used to generate the HIGHWAY-predicted routes that were verified through comparison with actual trucking routes used for a number of DOE shipments. The user can select any combination of distance and time biases by simply defining the value of β . The distance bias is then automatically set by Eq. (2):

$$\alpha = 1.0 - \beta , \quad (2)$$

where $0 \leq \beta \leq 1$.

Theoretically, the criteria outlined above could produce different routes between any two points. However, in actual practice, two or, occasionally, all three of the criteria will produce identical routes. Figure 2 shows both the shortest and the commercial routes between Morris, Illinois, and Richland, Washington. (In this example, the quickest and the commercial routing criteria produced identical routes.) The commercial route follows I-80 and I-84 across Iowa, Nebraska, Wyoming, Utah, Idaho, and Oregon for a distance of 1938 miles and an estimated driving time of 39 h and 21 min. The shortest route follows a more northerly path through Minnesota, South Dakota, North Dakota, Montana, and Idaho, covering a distance of 1884 miles. While the latter route is 54 miles shorter than the commercial route, the estimated driving time is 43 h and 42 min, or approximately 4.5 h longer than the commercial route. The commercial

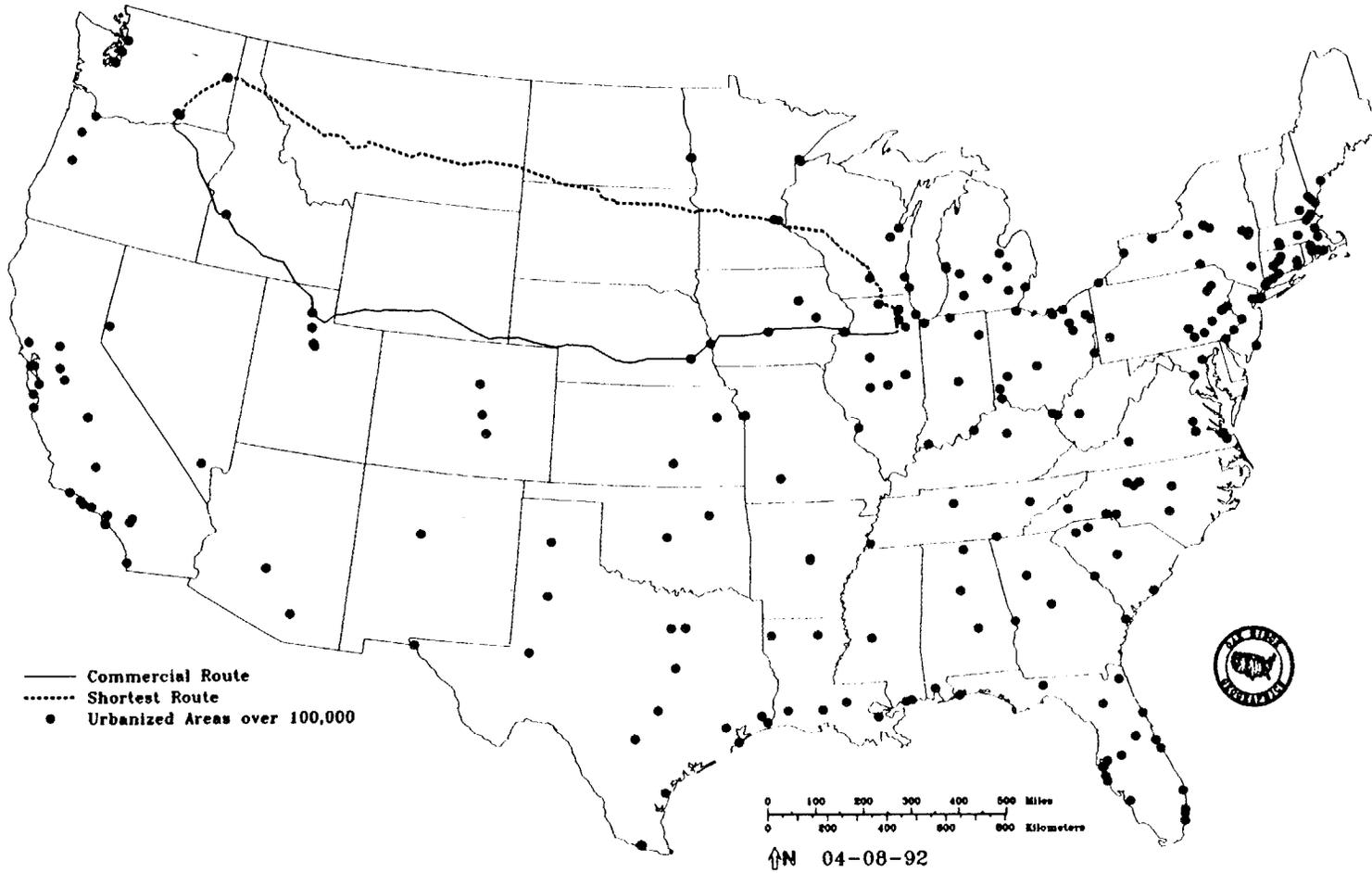


Fig. 2. Comparison of commercial vs shortest route between Morris, Illinois, and Richland, Washington.

route, with its compromise between the shortest time and distance, closely approximates the routes that are most likely to be used commercially.

1.4.2 Toll Bias

The use of toll roads or toll bridges can be controlled by the toll bias. Routes calculated using the toll bias will tend to avoid all toll roads. On the other hand, removing the toll bias does not place any special penalty on the use of toll roads, and, in such cases, toll roads and other highways are treated equally. At the start of the program, the user is asked whether routes should be calculated using a toll bias.

1.4.3 Number of Drivers

The estimated time of travel is reported for each route. Normally, times are based on a two-driver team, where a given shipment is assumed to move for 4 h and then stop for a 0.5-h break. This cycle is repeated until the destination is reached.

The user also has the option of specifying a single driver. In this case, the shipment is assumed to move for 5 h, stop for a 0.5-h break, then move for an additional 5 h, after which an 8-h break is taken. This pattern is repeated until the destination is reached. The number of drivers does not change the estimated route, but it does affect the time required to make a particular shipment.

1.4.4 Routing Constraints

In addition to the shortest, quickest, and commercial route types already discussed, several additional constraints can be imposed during the calculations. An entire state can be bypassed, if necessary, to accommodate local legislation or specific carrier operating rights. It is also possible to bypass a specific city or highway intersection. Specific highway segments may be avoided by simply blocking transport along that particular road. Any combination of these constraints can be imposed at the same time.

A number of examples showing the effects of various routing constraints are presented in this section. These examples are based on a route between Manchester, Iowa, and Dalhart, Texas. The standard route, shown in Fig. 3, was calculated using the commercial routing criterion discussed in Sect. 1.4.1. This route is 869 miles long, and the expected driving time for a two-driver team is 19 h and 20 min. The route primarily follows U.S. and state highways

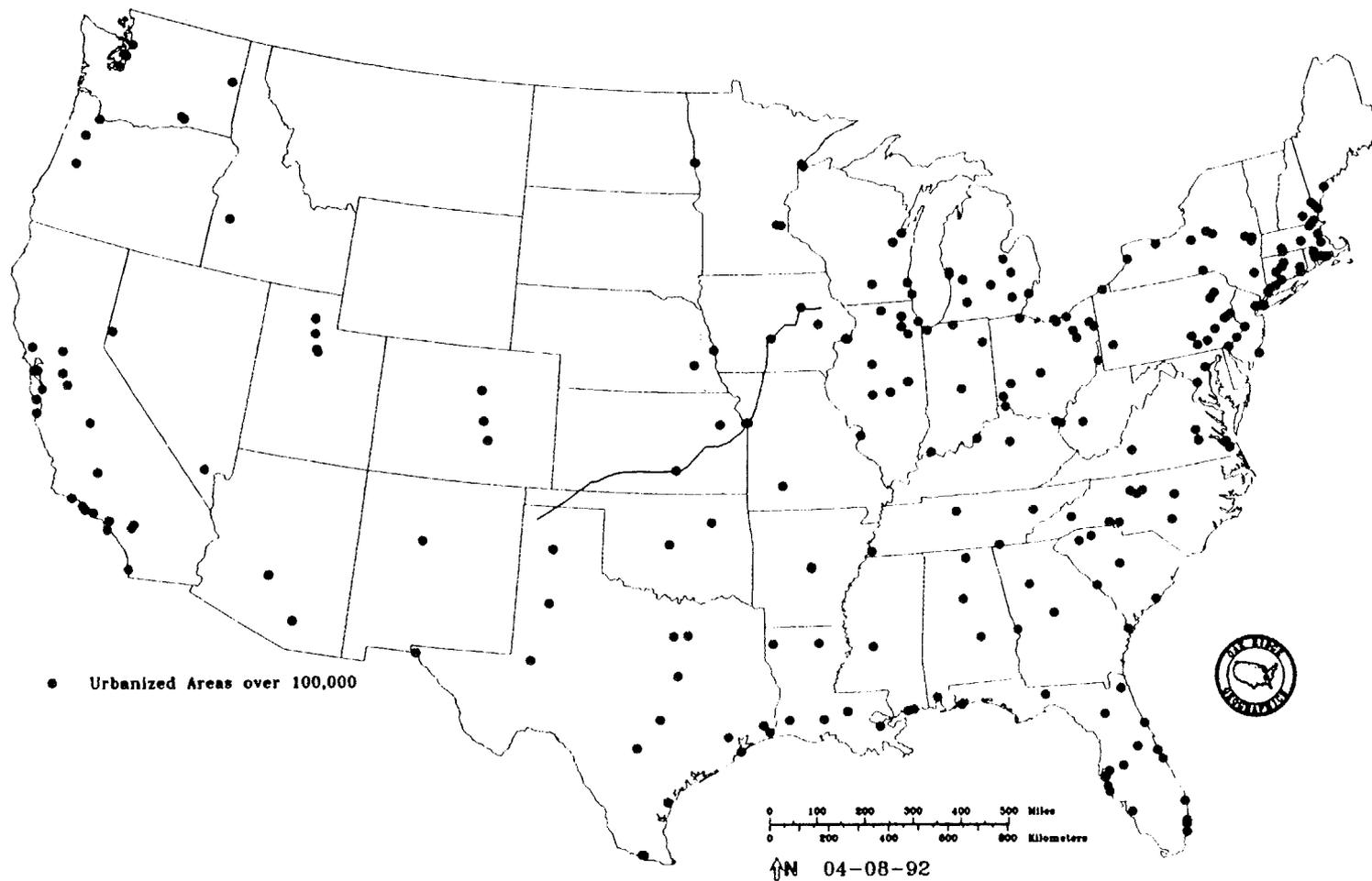


Fig. 3. Commercial route between Manchester, Iowa, and Dalhart, Texas.

between Manchester and Des Moines and Interstate highways from Des Moines to Wichita, Kansas. West of Wichita, the route again uses U.S. highways.

An example of bypassing an entire state is shown in Fig. 4. In this case, the route between Manchester and Dalhart was constrained to bypass the state of Missouri. The commercial routing criterion was also used to generate this route, which runs in a general west/southwesterly direction through Iowa and southeastern Nebraska, diagonally across Kansas and the panhandle of Oklahoma, and into Texas. The route bypassing Missouri is 879 miles long with an estimated driving time of 20 h and 16 min. While this route is only 10 miles longer than the commercial route illustrated in Fig. 3, the driving time is approximately 1 h longer because the route primarily follows state and U.S. highways with a reduced average speed.

An example of bypassing a specific city or town is shown in Fig. 5. In this case, the route between Manchester and Dalhart was constrained to bypass Wichita, Kansas. Between Manchester and Emporia, Kansas, the route is identical to the commercial route (Fig. 3), but at this point the constrained route passes to the north and west of Wichita before rejoining the original route at Meade, Kansas. The route bypassing Wichita is 10 miles shorter than the commercial route; however, due to the increased use of U.S. and state highways, the expected driving time is almost 0.5 h longer.

Blocking transportation through a specific city or town can be extended to include as many cities as needed. For the example shown in Fig. 6, the route was constrained to bypass all cities having populations greater than 100,000. This route passes between Waterloo and Cedar Rapids, east of Des Moines, between Topeka and Kansas City, and north of Wichita. The total length of this route is 911 miles, and the expected driving time is 21 h 45 min.

In addition to blocking transport in cities, specific highway segments can also be bypassed. The route shown in Fig. 7 was constrained from using I-35 between Des Moines, Iowa, and the Iowa/Missouri state line. Under these conditions, the route would head south from Des Moines on U.S. 65 through Indianola, Iowa, to Princeton, Missouri, then southwest on U.S. 136 to Bethany, Missouri. The remainder of the route between Bethany and Dalhart, Texas, is identical to the corresponding portion of the commercial route illustrated in Fig. 3.

It is interesting to look at the effect of various constraints on the type of highways used in the various routes. The fraction of each route along Interstate, U.S., and state highways is shown in Table 2. As a result of the existing highway network between Manchester and Dalhart, the Interstate system is normally used for approximately 47% of the route. In general, placing

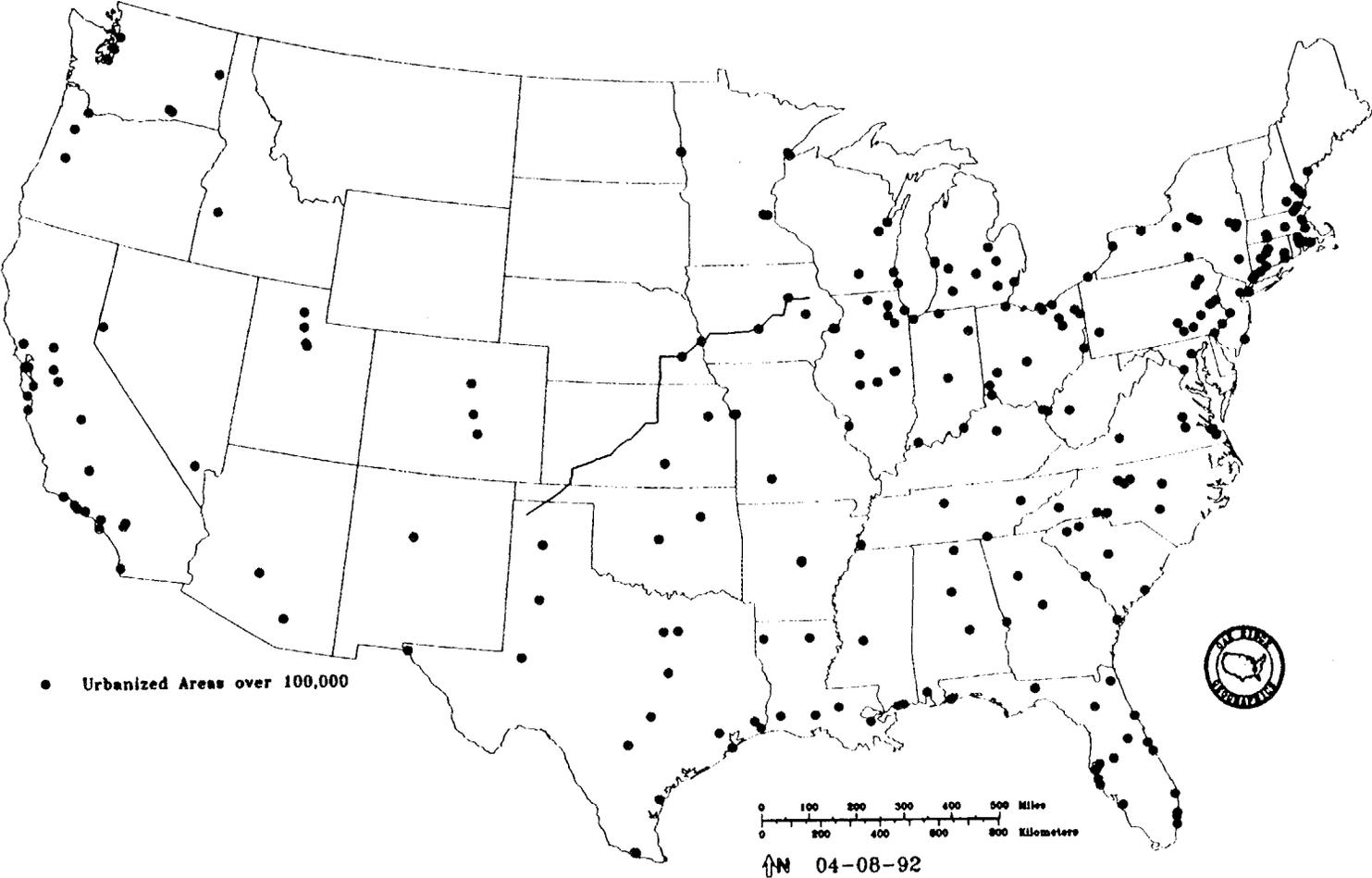


Fig. 4. Route between Manchester, Iowa, and Dalhart, Texas, that bypasses Missouri.

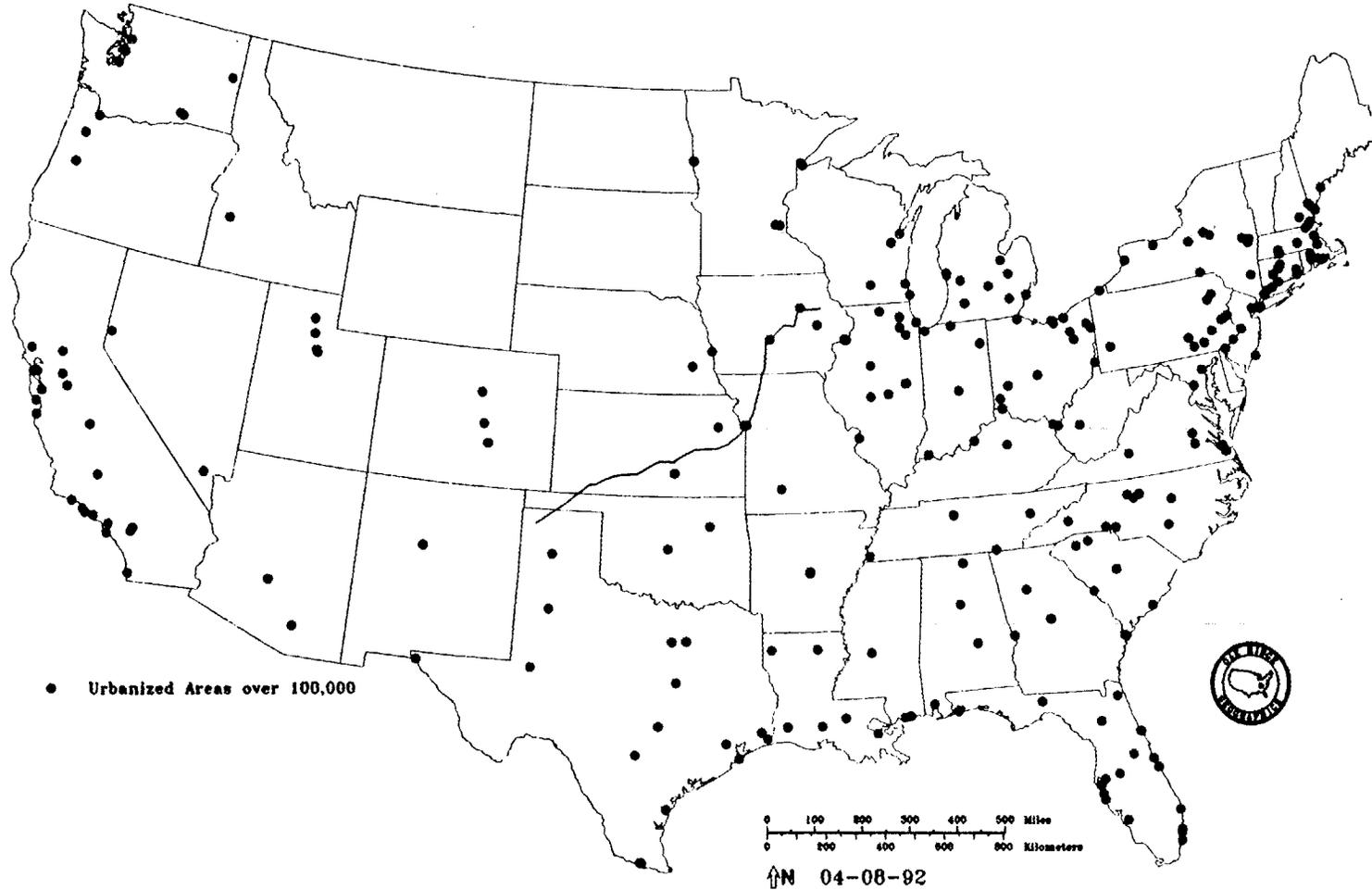


Fig. 5. Route between Manchester, Iowa, and Dalhart, Texas, that bypasses Wichita, Kansas.

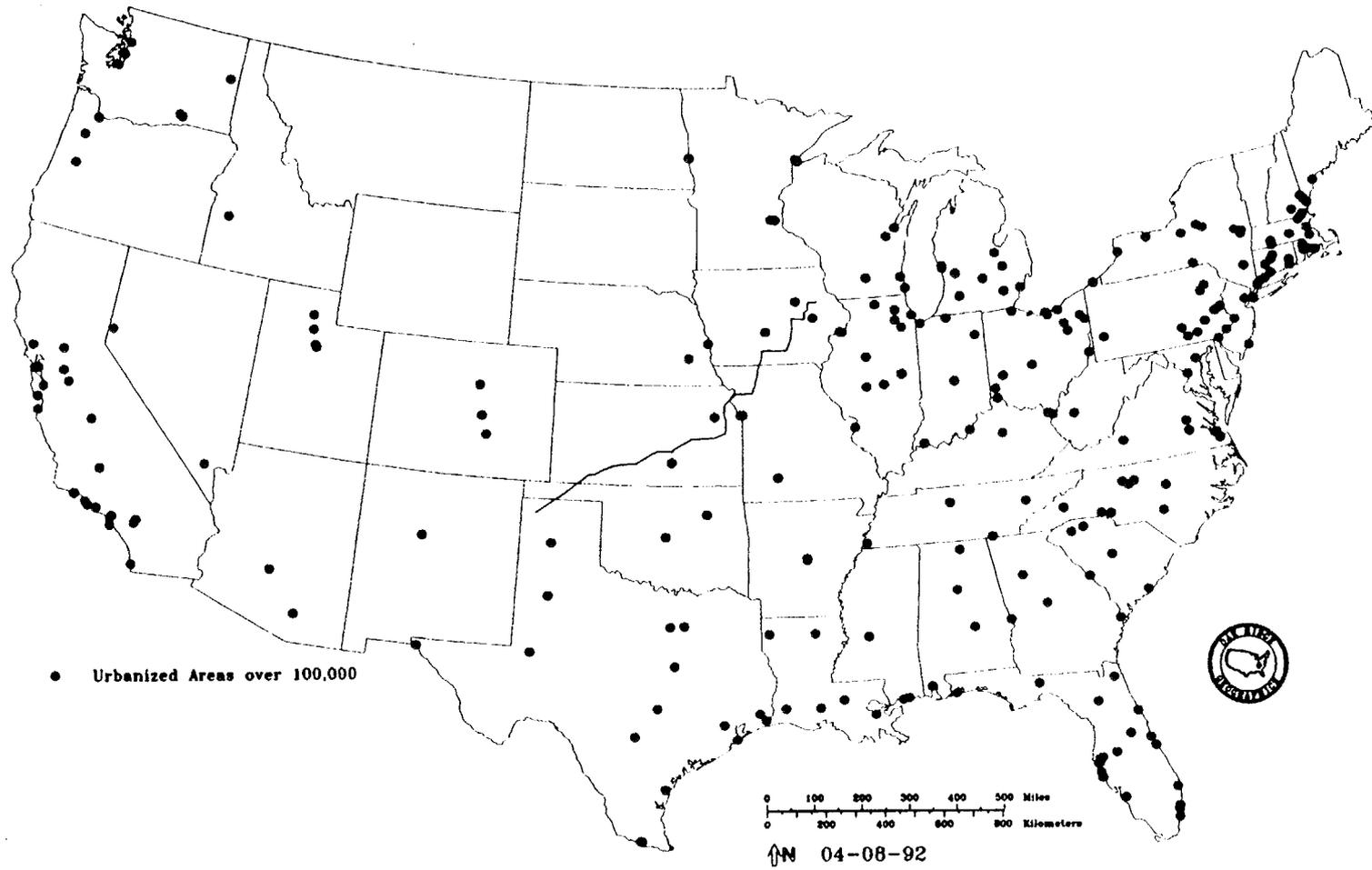


Fig. 6. Route between Manchester, Iowa, and Dalhart, Texas, that bypasses urban areas with populations greater than 100,000.

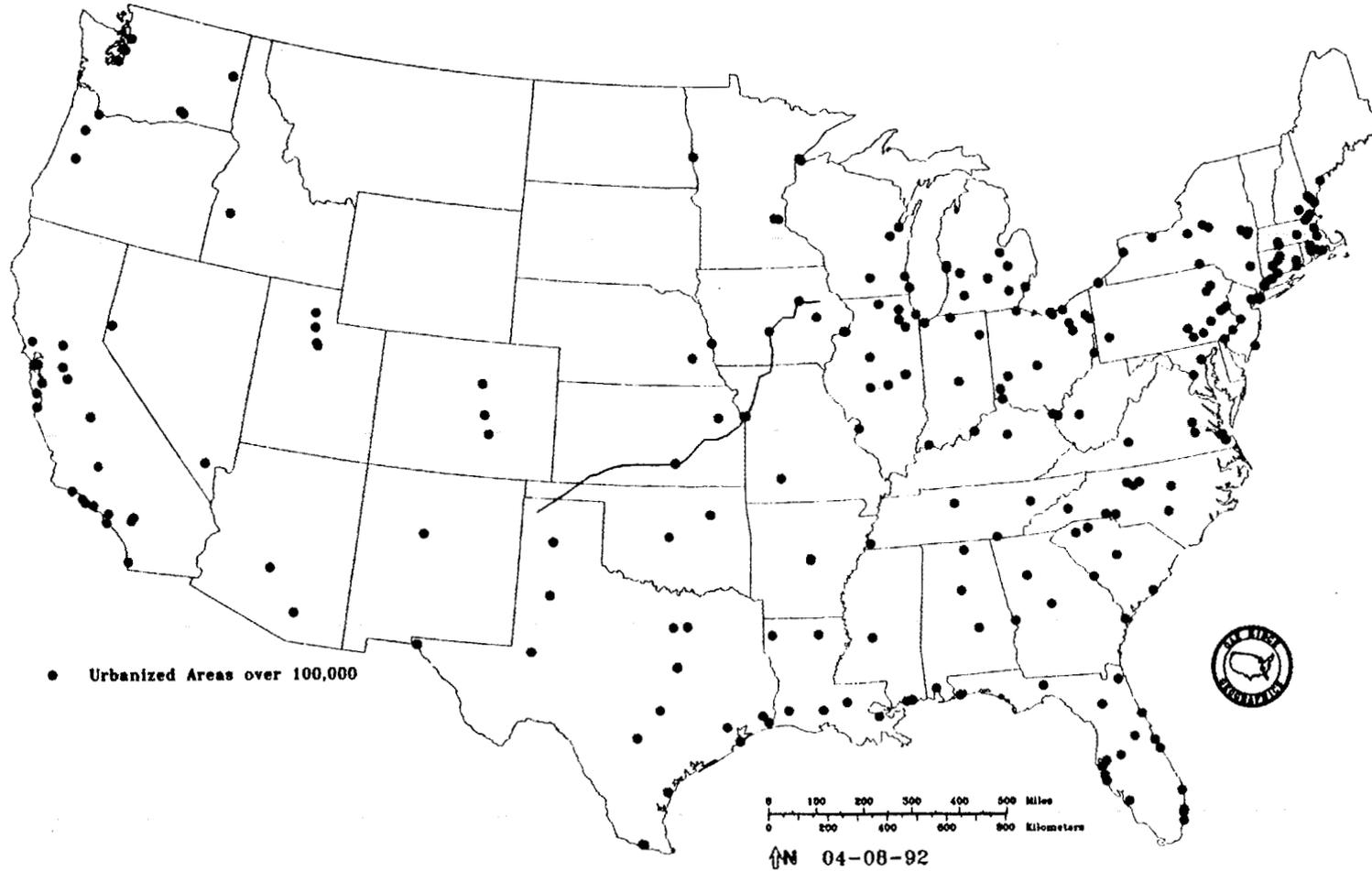


Fig. 7. Route between Manchester, Iowa, and Dalhart, Texas, that bypasses I-35 between Des Moines, Iowa, and the Missouri state line.

Table 2. Highway usage for various routes

Route	Constraint	Distance (miles)	Driving time (h)	Highway type (% of total distance)		
				Interstate	U.S.	State
1	None	869	19:20	47.1	50.6	2.3
2	Bypass MO	879	20:16	31.2	61.1	7.7
3	Bypass Wichita, KS	859	19:45	37.2	55.1	7.7
4	Bypass urban areas	911	21:45	16.2	72.8	11.0
5	Bypass I-35	885	20:06	34.8	62.9	2.3

constraints will reduce the ability to use the Interstate highways. The extreme case is the route that was required to bypass all major cities, where only 16.2% of the route was along Interstate highways.

1.4.5 Routes for Highway Route-Controlled Quantities of Radioactive Materials (HM-164 Routes)

One of the major features of the HIGHWAY model is its ability to estimate routes for the transport of highway route-controlled quantities of radioactive materials. The U.S. Department of Transportation (DOT) has established a set of routing regulations for the transport of these materials (49 CFR 177.825). Routes following these guidelines are frequently called HM-164 routes. Basically, the DOT regulations require that these shipments be transported over a preferred highway network, which includes:

1. Interstate highways,
2. an Interstate System bypass or beltway around a city, and
3. state designated preferred routes.

State routing agencies may designate preferred routes as an alternative to, or in addition to, one or more Interstate highways. In making this determination, the state must show that the alternative preferred route is as safe as the Interstate route that it is replacing and must register all such designated preferred routes with DOT.

Frequently, the origin and destination of highway route-controlled shipments of radioactive material are not located on Interstate highways. The DOT routing regulations require the carrier to select the shortest distance route between the pickup location to the nearest

preferred route entry location and the shortest distance route to the destination from the nearest preferred route exit location.

An example of an HM-164 route between Manchester, Iowa, and Dalhart, Texas, is shown in Fig. 8. Since there are no Interstate highways in Manchester, the HM-164 routing algorithm selected the most direct route along U.S. 20 to the junction of I-380 southeast of Raymond, Iowa. The HM-164 route then follows Interstate highways through Iowa, Missouri, Kansas, and Oklahoma to Vega, Texas. From that point, the model again selected the most direct route (U.S. 385) from the Interstate highway system to Dalhart, Texas. The HM-164 route is 1115 miles long, with an estimated driving time of approximately 23 h.

The reader will notice that the HM-164 route is approximately 250 miles longer than the commercial route discussed in the previous section. This unusually large difference is a result of the Interstate network in the central portion of the country and the particular origin and destination chosen for this example. In general, HM-164 routes tend to be somewhat longer than other routes; however, the increased safety associated with Interstate travel is the primary purpose of the DOT routing regulations.

1.4.6 Alternative Routing

The HIGHWAY model can also be used to generate a series of alternative routes between the same origin and destination. A number of alternative routes between Marshall, Michigan, and Phoenix, Arizona, are shown in Fig. 9. For simplicity, these routes were constrained to travel along Interstate highways. The base route traveled west, along I-94 and I-80, from Marshall to the junction of I-55 near Joliet, Illinois. The route then followed I-55 to the junction of I-44 southwest of St. Louis, Missouri, and continued along I-44 to Oklahoma City, Oklahoma, I-40 to Flagstaff, Arizona, and I-17 to Phoenix. This base route extends for 1961 miles and passes through eight states. The estimated driving time for the base route is 39 h and 40 min.

The first alternative route goes directly south from Marshall to Indianapolis, Indiana, along I-69, before turning west on I-70 to the St. Louis area. After the alternative route reaches I-44 southwest of St. Louis, Missouri, it follows the same path as the base route through Oklahoma City and Flagstaff. The first alternative route is 1982 miles long (21 miles longer than the base route). The estimated driving time is slightly over 40 h.



Fig. 8. Route between Manchester, Iowa, and Dalhart, Texas, that maximizes the use of Interstate highways.

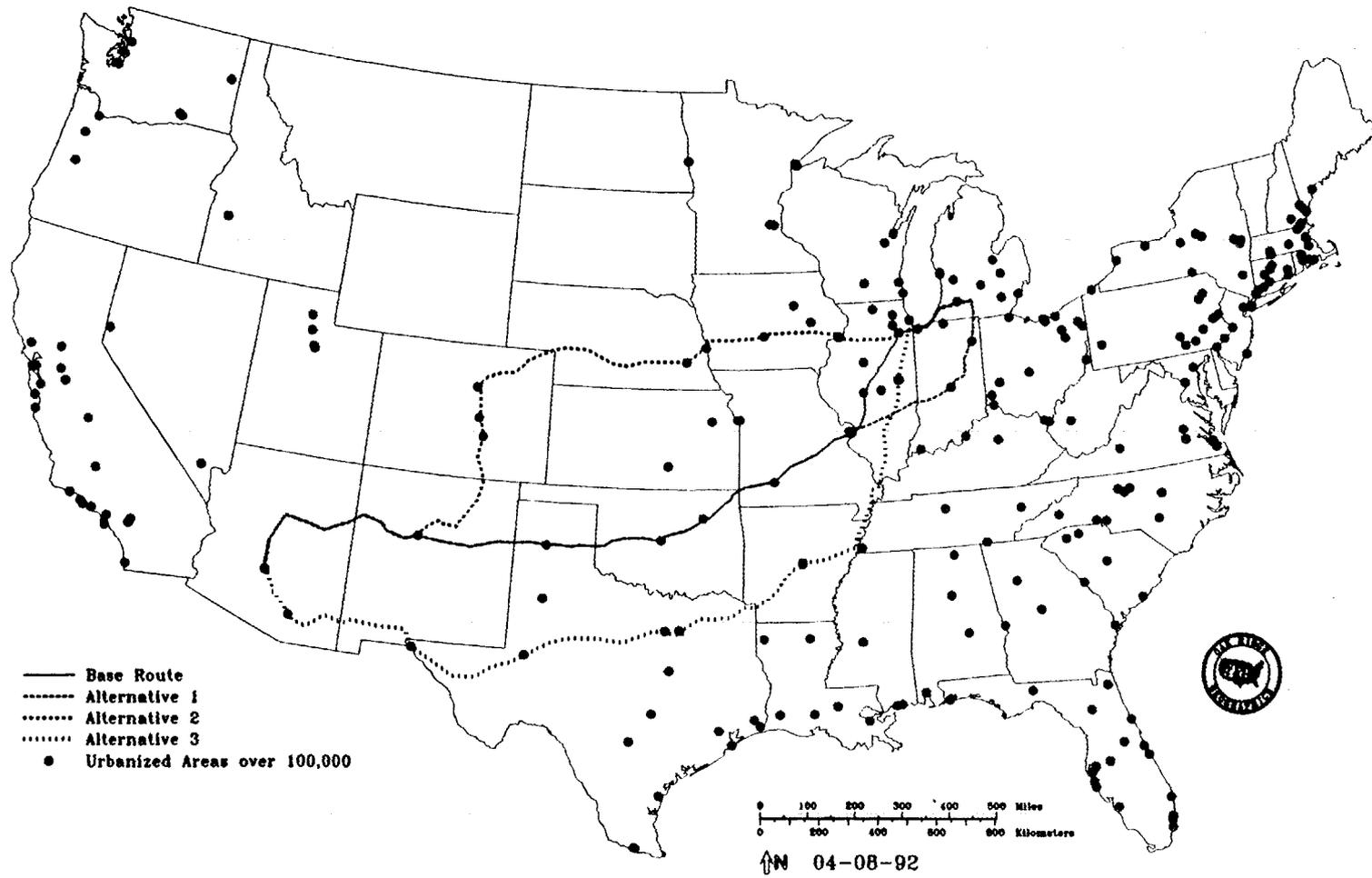


Fig. 9. Alternative routes between Marshall, Michigan, and Phoenix, Arizona.

As shown in Fig. 9, the second alternative route followed a significantly different path. This route traveled along I-94 to the junction of I-80 in northern Indiana and then followed I-80 westward to Big Springs, Nebraska, before turning southwestward along I-76 to the junction of I-25 near Denver, Colorado. From this point the route continued southward along I-25 to Albuquerque, westward along I-40 to Flagstaff, and finally south along I-17 to Phoenix. The second alternative route is 2081 miles long (120 miles longer than the base route), and the estimated driving time is 42 h and 22 min.

A third alternative route that followed a more southerly path than the other routes was generated. This route turned south along I-57 in northeastern Illinois and followed I-57 and then I-55 to the junction of I-40 in West Memphis, Arkansas. The route then follows I-40, I-30, I-20, and I-10 through El Paso, Texas, southern New Mexico, and southern Arizona. The third alternative route is 2186 miles long with an estimated driving time of 44.25 h.

The distances and estimated driving times for the alternative routes are summarized in Table 3.

Table 3. Comparison of alternative routes

Route	Distance (miles)	Driving time (h)
Base	1961	39.67
Alternative 1	1982	40.05
Alternative 2	2081	42.37
Alternative 3	2186	44.25

The alternative routing capability is a useful tool for estimating a number of different routes between the same origin and destination. The alternative routing algorithm 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 meaningful differences.

2. METHODOLOGY

This chapter presents a description of the shortest path techniques used to calculate routes in the HIGHWAY model. 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, which 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. highway 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 HIGHWAY model to handle decomposed networks. This technique is illustrated in Sect. 2.2.5 by showing a calculation for a route that spans four state subnetworks. This chapter also includes a more complicated example showing alternative routing.

2.1 ORGANIZATION OF THE NETWORK FOR ROUTING ANALYSIS

In its basic structure, the highway network model is essentially a data base containing a computerized representation of the U.S. highway system. For route analysis, this complete U.S. network is broken down, or decomposed, into a series of subnetworks representing highway systems in individual states, which are connected at border crossings. The use of state subnetworks reduces the amount of computer memory required to operate the program and allows some state-specific options to be included in the model. After decomposition, the network is then organized into a form convenient for the routing algorithms.

2.1.1 State Highway Network Representation

The networks for the individual states are developed from the information in the master network data base. This is done in a rather straightforward fashion. Individual highway segments in the master data base contain, among other attributes, a "from" node, a "to" node, "from" and "to" states, and two corresponding mileage fields. When a link lies exclusively

within one state, both state fields are the same and the first mileage field contains the total length of the link. If the state fields differ, the link crosses a state boundary, and the mileage fields denote the link mileage within each state. The "from" node lies in the first state, and the "to" node lies in the second.

During the decomposition step, links that cross state boundaries are split into two links by adding a new node at the state border. The first link lies within the "from" state of the original link, with a length equal to the corresponding mileage. Its "from" node is the "from" node from the original link. The second link lies in the "to" state of the original link and has as its "to" node the "to" node of the original link. The new border node serves as the "to" node of the first link and the "from" node of the second link.

After all border-crossing links are divided, the set of nodes belonging to each state's network is determined by scanning the nodes for each link within the state. The routing algorithm requires that the nodes in a state network be numbered in sequence, starting with 1. Then nodes in the master data base are named. After decomposition, the nodes in each state are automatically assigned sequential numbers, starting at 1.

Figure 10 illustrates this decomposition process. Figure 10(a) shows a portion of the combined network in the vicinity of a state border. For clarity, dashed lines indicate links crossing a state boundary. Letters are used to represent the original node names. Figure 10(b) shows the new nodes placed at border crossings during the link splitting step. Figure 10(c) then shows the network decomposed into the two separate states. Nodes in each state network have been numbered sequentially. Links crossing state borders have been split into separate links, each of which lies entirely within one state. The tables in Fig. 10(c) show the correspondence between the new sequential state node numbers and the master network node names.

Once the network has been decomposed, it is 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.

2.1.2 Border Crossings

During the decomposition step, the network was broken into separate state networks by splitting links that crossed state boundaries. Clearly, highways cross state boundaries, and therefore, routes must be able to cross as well. HIGHWAY, therefore, reconnects the highway

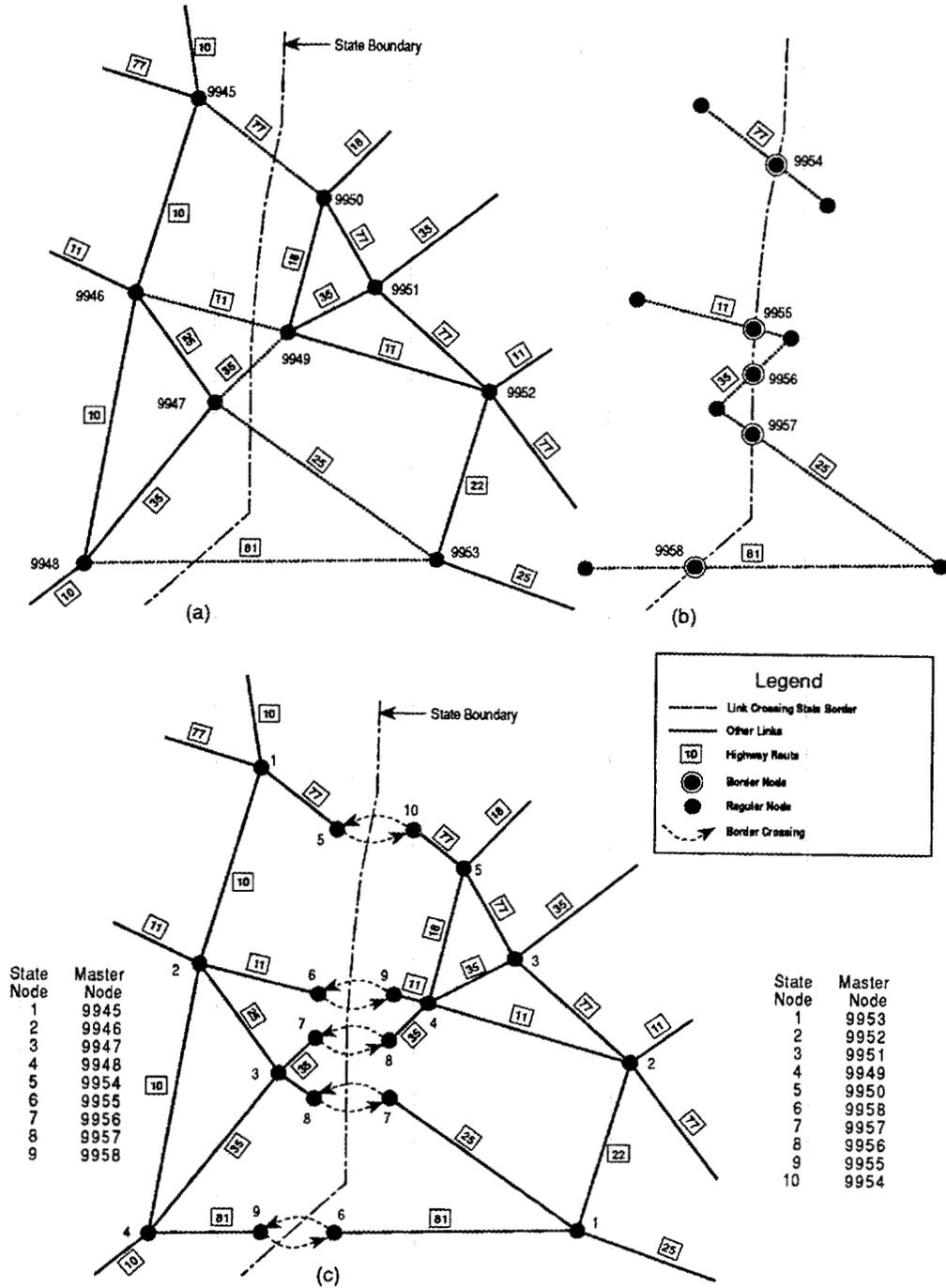


Fig. 10. Graphic illustration of (a) an undecomposed network, (b) a border node creation, and (c) a decomposed network.

segments crossing borders with a special type of link. These links connect the individual highway subnetworks and allow routes to cross subnetwork boundaries. They have no physical representation in the real world, however. The border crossing links are added strictly to implement the routing logic, which will be discussed in a later section. Figure 10(c) shows the crossing links as dashed lines with arrowheads.

2.2 THE HIGHWAY ROUTING ALGORITHM

This section presents the logic used in the HIGHWAY model to estimate routes. The general procedure is based upon a mathematical procedure called a shortest path algorithm. However, the algorithm in HIGHWAY has been modified to handle interconnected subnetworks. The following subsections describe the principles employed by the algorithm. The user is referred to refs. 3 and 4 for additional information on shortest path algorithms.

2.2.1 Routing Nomenclature

To understand the following discussion, the reader should be acquainted with some terms and concepts.

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 and link in a path appears only once. This means that the path does not cross itself.

Each link in a network has an associated nonnegative impedance that is incurred for using the link in a path. This may be any attribute of interest to the analyst. Typical values include distance, travel time, cost of making a shipment, or risk. In the HIGHWAY model, link impedance is defined as a function of distance and travel time.

The impedance for a route 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 shortest path is, then, the path with the minimum impedance.

The shortest path algorithm takes a network that has a number of interconnections (like a web) and creates a shortest path tree, which is the collection of the shortest 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 which the routing algorithm uses to evaluate the minimum impedance or shortest path. For example, HIGHWAY produces routes consisting of line segments according to the following objective:

$$L = \text{Min} \sum_i \tau_i (\alpha d_i + \beta t_i) , \quad (3)$$

where

- L = impedance for route;
- τ_i = toll road penalty for link i;
- α = distance bias;
- d_i = length of link i, miles;
- β = time bias;
- t_i = driving time on link i, minutes.

The impedance of a highway segment is a weighted combination of travel time and distance. Links representing toll facilities (bridge, road, or tunnel) have their impedance increased by the toll penalty, which is normally set to 15%. Transfer links at border crossings have zero impedance. Depending upon the options selected for a run of the model, the objective function may be modified. The discussion on routing options describes these modifications.

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.

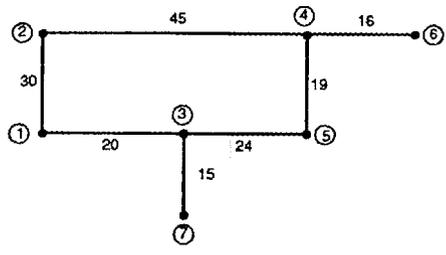
At the start of the algorithm, the program:

- sets the label of the origin node to 0,
- sets the status of the origin node to TRUE,
- places the source node into 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.

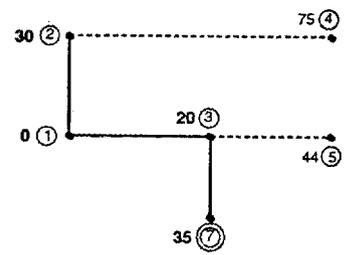
The main portion of the algorithm procedure is described below.

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 this 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. For every node having a status of FALSE at the end of a link emanating from the pivot node, do the following.
 - 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 first step.

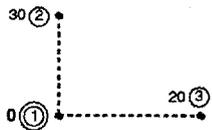
Figure 11 provides a graphic demonstration of the shortest path algorithm. Figure 11(a) shows the links and nodes of an example network. The numbers alongside the links represent the impedances. Figures 11(b)-11(h) show the successive stages of the algorithm for an origin at node 1. The node number within 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



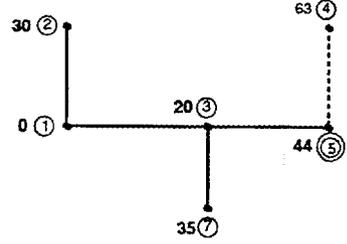
(a)



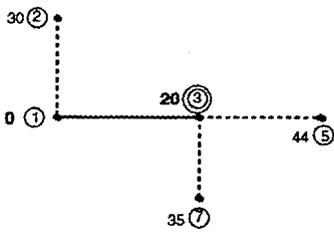
(e)



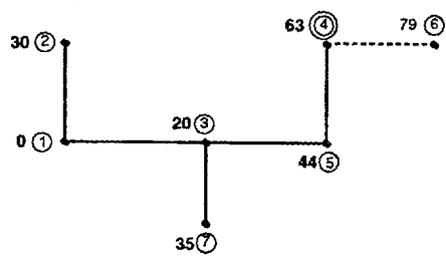
(b)



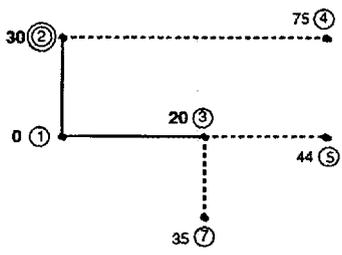
(f)



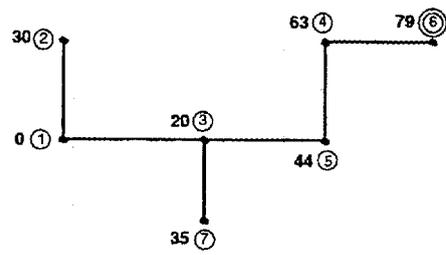
(c)



(g)



(d)



(h)

Fig. 11. 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.

TRUE); others 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 HIGHWAY Routing Algorithm

HIGHWAY uses a modification of the shortest path algorithm to find the best route through a series of state subnetworks. The algorithm must route within a state and decide how to move between states. To do this, a two-stage approach is used. The top-level procedure decides which state 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 state using a traditional shortest path approach. The behavior of the algorithm is described below.

Conceptually, the upper-level procedure treats states like nodes and border crossings like links. These entities are given additional attributes to reflect this, as explained below.

- Each state has a label that reflects the minimum impedance of all paths reaching the state from the origin node.
- Each border crossing has a label that is used in transferring label values for transfer nodes between state subnetworks.

For path traceback, individual nodes must have both a predecessor state identifier and a predecessor node number. Number alone will not uniquely identify a node, since all subnetworks have nodes sequentially numbered, 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 state subnetworks are set to infinity.
- The status of all nodes in all state subnetworks is set FALSE.
- The label of each state is set to infinity.
- The label of each border crossing is set to infinity.
- The label of the source node in the source state is set to 0.
- The status of the source node in the source state is set TRUE.
- The label of the source state 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 state subnetwork.

2.2.4.2 Main Loop

The body of the algorithm consists of the following steps.

1. Select the state having the lowest label, and treat this as the pivot. Initially, the origin state will be selected since its label is 0. The path tree will be extended within this pivot state.
2. Consider the following conditions:
 - a. If the label of the pivot state is greater than or equal to the label of the destination node, then the route is complete, so stop.
 - b. If no pivot state is selected, no direct path to the destination exists, so stop.
3. Examine all border crossings into the pivot state. If the label of a transfer link is less than the label of its end node in the pivot state, do the following.
 - a. Make the label of the border-crossing node in the pivot equal to the label of the crossing link. The path from the origin via the connecting state is shorter than the current path.
 - b. Set the predecessor state of the border-crossing node in the pivot equal to the identifier of the state from which the crossing originates.
 - c. Set the predecessor node of the border-crossing node in the pivot equal to the node in the state from which the crossing originates.
 - d. Set the status of the crossing 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 state using a shortest path algorithm. The procedure starts with the first node in the candidate pool of the state subnetwork.
5. Set the label of the pivot state to infinity so that it will not be reselected in the next iteration.
6. Examine all border crossings into the pivot state again. Set the label of each crossing link equal to the label of its end node in the pivot state.
7. Examine all border crossings from the pivot state to surrounding states. Compare the impedance of the end node in the pivot state with the crossing link's label.
 - a. If the impedance is less than the crossing link's label, then the label is set to the impedance of the end node in the pivot state.
 - b. If the end node's impedance is less than the crossing state's label, set this label to the end node's impedance.

8. Return to the first step.

The reader may wish to review this sequence of steps carefully because the logic may not be intuitively obvious. The example in Sect. 2.2.5 should help to clarify the procedure.

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 Sample Route

To demonstrate the algorithm, the sample network of Fig. 12 is presented. The overall network consists of four state subnetworks connected at border points. The circled numbers identify nodes within each state subnetwork. Predetermined impedance values, which already include appropriate factors, are provided beside each link. Transfer links are indicated by dashed lines with arrowheads.

The route of interest lies between node 1 in state 1 and node 2 in state 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 state 1 and node 1 in subnetwork 1 are set to 0. Node 1 is placed in the candidate pivot pool for state 1. All other transfer labels, state labels, and node labels are set to infinity. To begin the first iteration, state 1 is selected as the pivot because it has the smallest label.

Figure 13(a) shows the initial conditions at the beginning of the pivot operation for state 1. Figure 13(b) shows the label values and path tree at the conclusion of step 7 of the algorithm. Solid links indicate the path tree from the origin. Nodes 3 and 7 of state 1 connect to state 2, and the minimum impedance path is 15 units via node 3. This value (15 units) is assigned to the label for state 2. Likewise, nodes 4 and 5 of state 1 connect to state 3, and the minimum impedance to this network is 16 units for a path passing through node 4. The updated state labels indicate that state 2 will be selected for the next pivot.

Figure 14 shows the start and end conditions for the pivot involving state 2. Note that the calculations continue from both nodes 1 and 5 of this state and that the path extension generated **does not form a tree** within this state subnetwork. The paths shown in Fig. 14(b) are an extension of the tree rooted at the origin in state subnetwork 1. The reader should trace through the algorithm steps carefully to determine how the label values are derived. State 3 will be the next pivot.

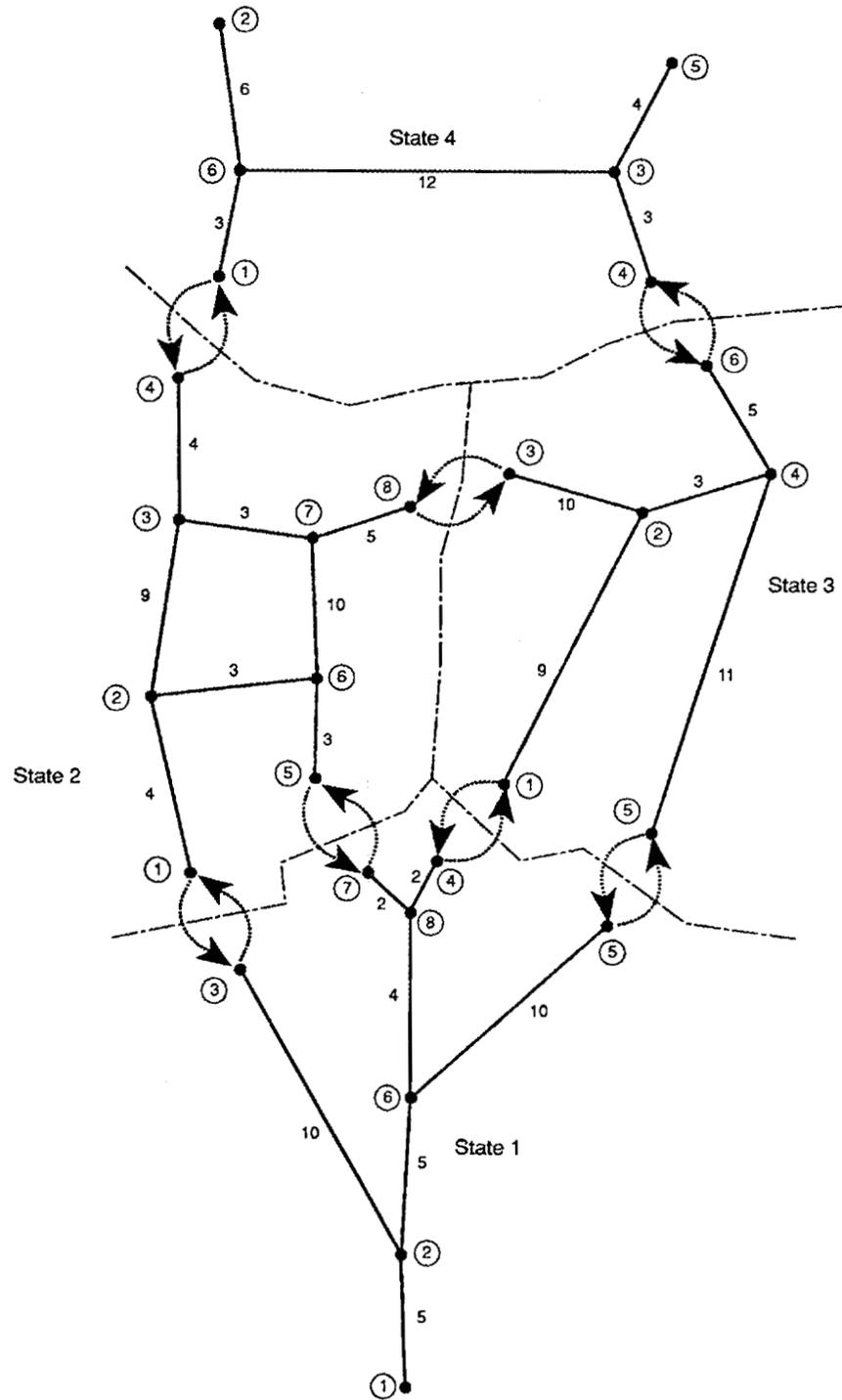


Fig. 12. Example of network structure.

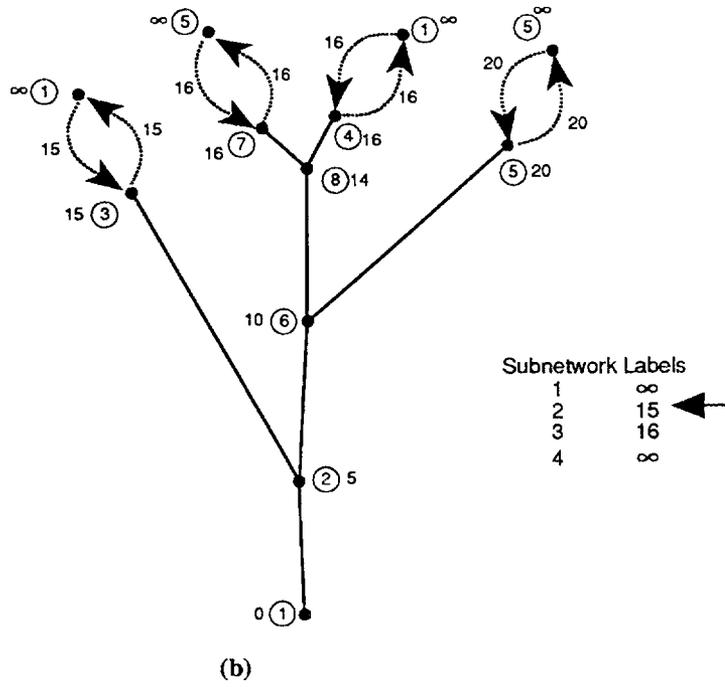
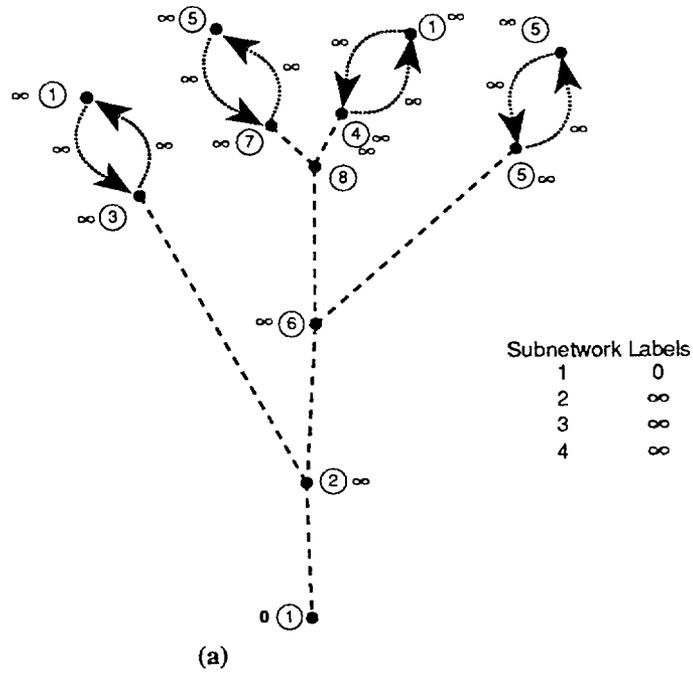


Fig. 13. Pivot through state 1 at (a) the start and (b) the end of iteration.

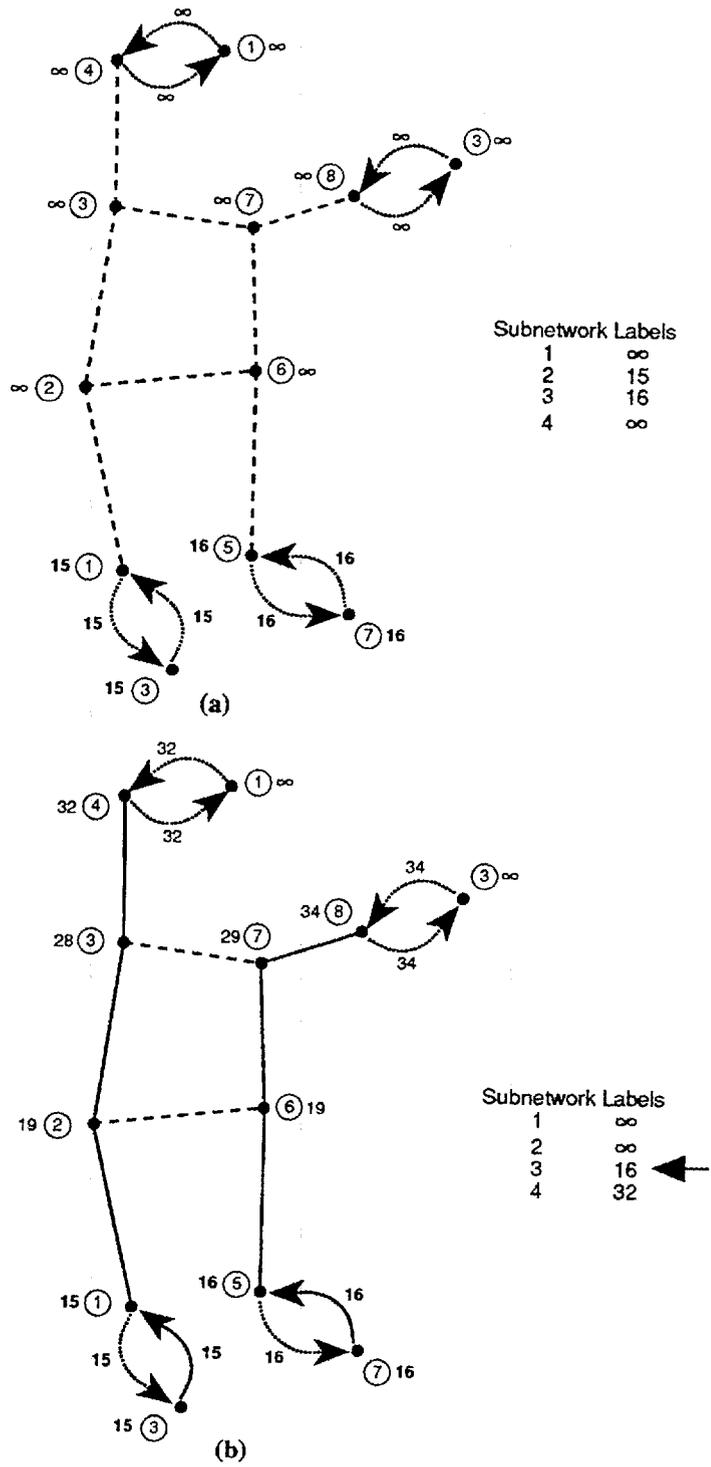


Fig. 14. Pivot through state 2 at (a) the start and (b) the end of iteration.

Figure 15 continues the process by expanding the path tree through state 3, and Fig. 16 completes the algorithm by extending the path through state 4. The final impedance of the path from the origin to node 2 is 41 units. No further iterations are required, since the label of node 2 is smaller than the label of any state.

Figure 17 shows the complete path tree through the four state subnetworks, along with the crossing 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.

2.3 PROGRAM OPTIONS

HIGHWAY has a number of options that can be used to influence the routing algorithm. The ability of the user to change the travel time and toll road weights has been previously mentioned. Program options allow the user to directly replace the default values of these parameters with desired values. The program also allows the user to make the route avoid selected highway segments, nodes, or states and to calculate alternative routes. This section describes how these options are incorporated into the HIGHWAY routing algorithms.

2.3.1 Link Avoidance

If the impedance of a link is increased by a vary 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 (see Sect. 3.1.6.3), HIGHWAY adds 70,000 units 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 segment may be included in the route, but the route will include the high-impedance segment only if there is **no alternative path with a lower impedance**. A message will be printed if penalized segments have been included in a route.

2.3.2 Node Avoidance

A route can be constrained from passing through a specified node (see Sect. 3.1.6.2) if a high-impedance penalty is applied to all links that emanate from the node. To avoid nodes, HIGHWAY adds a 70,000-unit penalty to the original impedance of each emanating link. The

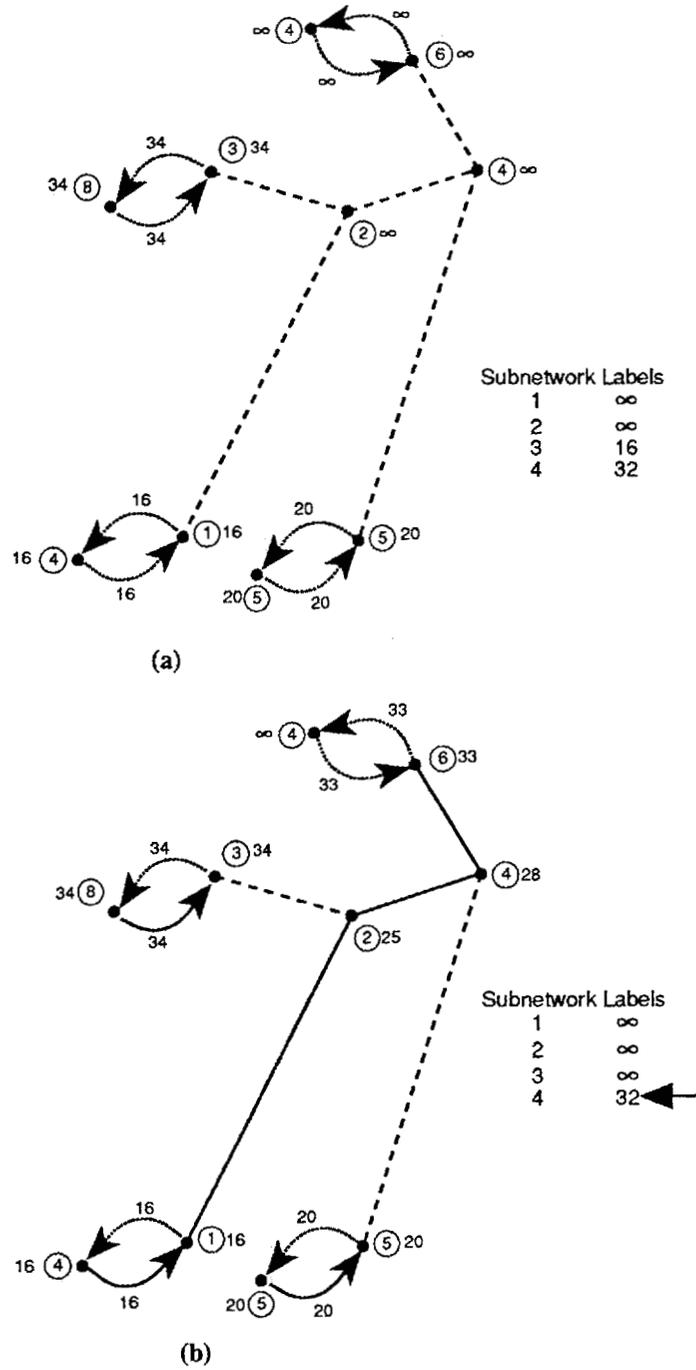


Fig. 15. Pivot through state 3 at (a) the start and (b) the end of iteration.

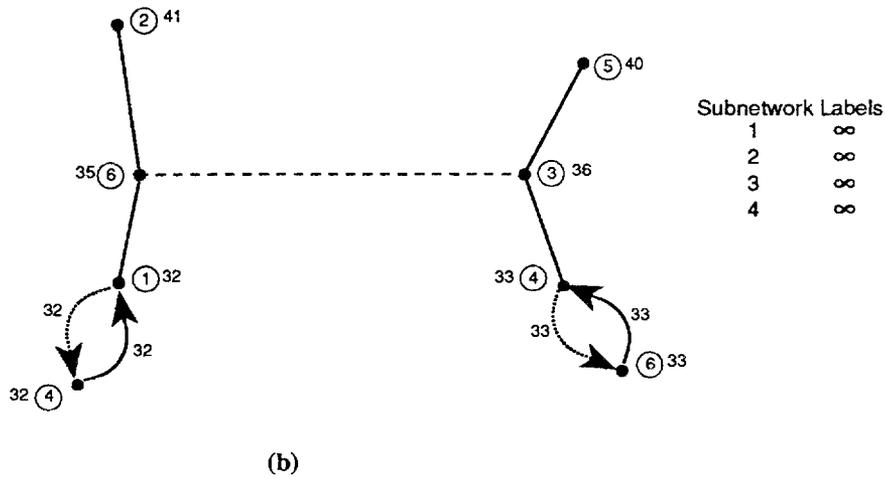
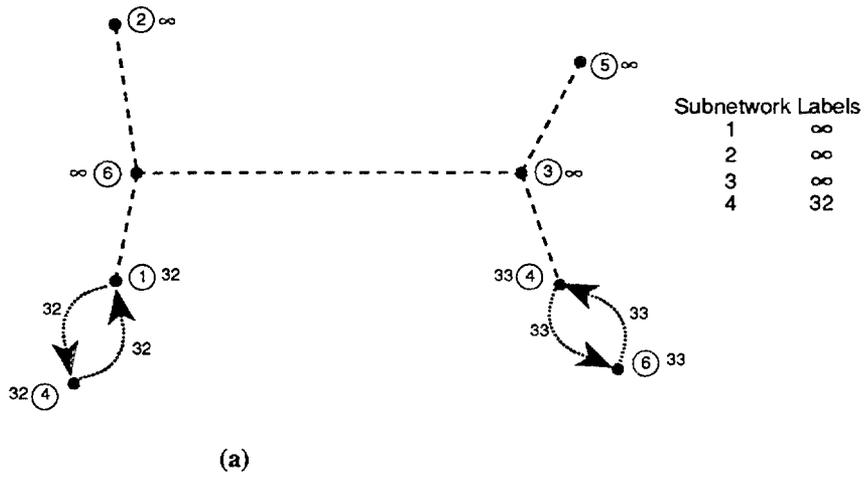


Fig. 16. Pivot through state 4 at (a) the start and (b) the end of iteration.

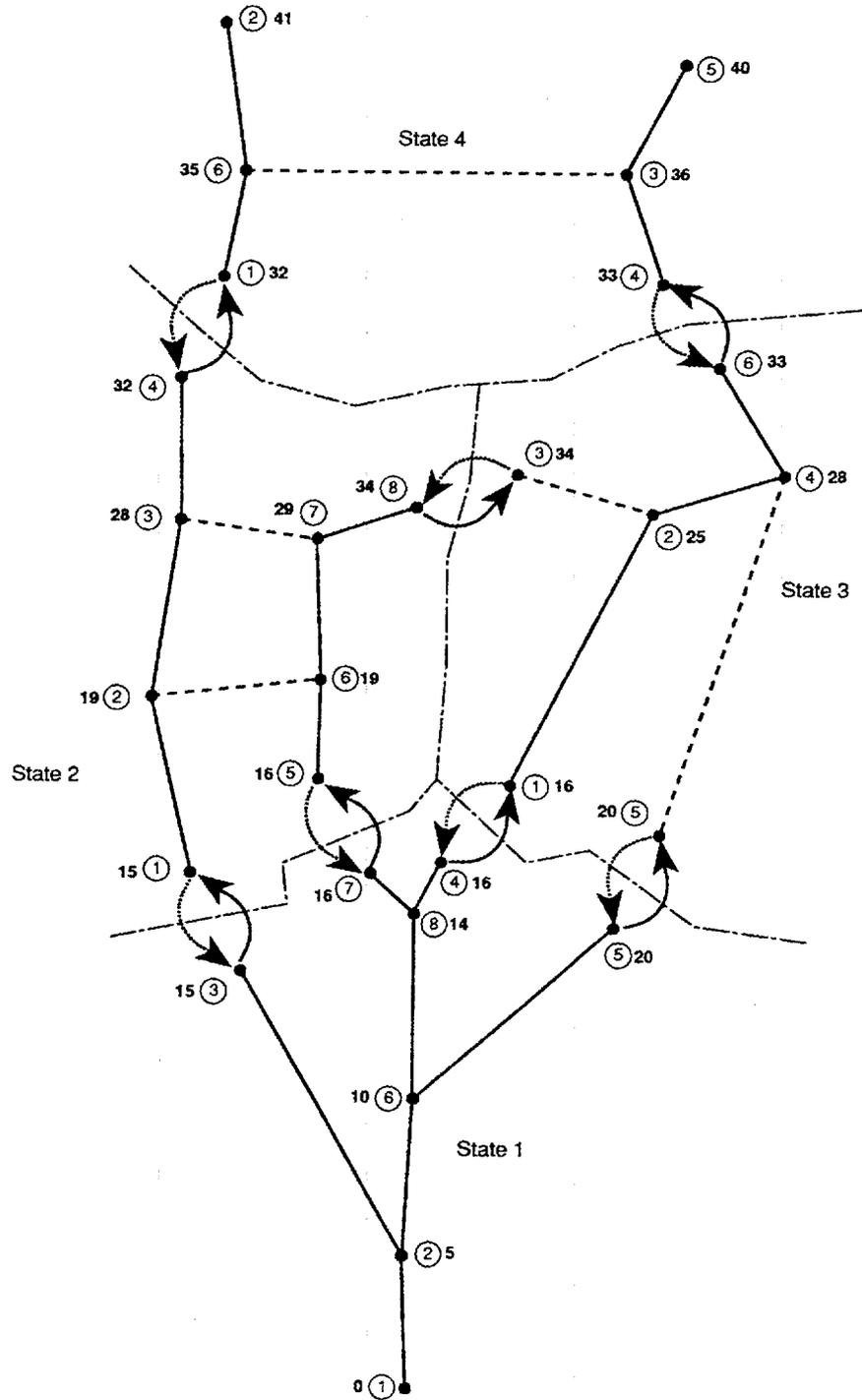


Fig. 17. Complete path tree.

high impedance associated with using the node will discourage its inclusion in the route provided that an alternative path exists around it. Node avoidance should not be applied to the origin and/or destination of a route, because that node must be part of the route. A message will be printed if the calculated route passes through a penalized node.

2.3.3 Route Weighting by Highway Sign Type

The highway sign type listed in Table 4 is the route type designation of the highway. All highway segments in the data base have an associated sign type. If this option is activated, as shown in Sect. A.2.7, HIGHWAY will multiply the impedance for each link by a sign type factor. This feature could be used, with the appropriate selection of factor values, to either encourage or discourage a path from including segments of a given sign type or types. Assigning a very low sign type weight to Interstate highways would, for example, encourage the model to maximize use of Interstate highway links in a path.

Table 4. Sign types and default weights

Sign type	Default weight
Interstate	1.0
U.S.	2.0
State	2.0
Turnpike	2.0
County	2.0
Local	2.0
Other	2.0

2.3.4 Route Weighting by Highway Lane Type

As shown in Table 5, highway segments in the data base also have an associated lane type, which is an indication of the design quality of the highway, its capacity, its level of service, etc. If this option is activated, as shown in Sect. A.2.7, HIGHWAY will multiply the impedance for each link by a lane type factor. As with the sign type, this feature can be used to encourage or discourage a path from including segments of a given lane type or types.

Table 5. Lane types and default weights

Lane type	Default weight
Limited access multilane	1.0
Limited access single lane	1.1
Multilane divided	1.2
Multilane undivided	1.3
Principal highway	1.5
Through highway	2.0
Other	2.5

2.3.5 State Prohibition

HIGHWAY allows the user to specify states 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 any part of the state subnetwork in the tree.

State subnetwork prohibition is implemented as follows. Each state has a status flag. If the value of this flag is FALSE, which is the normal condition, the state subnetwork is eligible to be a pivot candidate during step 1. If the status flag is TRUE, the state will never be considered as a pivot, and, therefore, will never be included in the path tree. The user has, as discussed in Sect. 3.1.6.1, the capability to set these state subnetwork status flags.

2.3.6 Weighting by Other Link Attributes

Links have a set of attribute flags that can be used in the routing process. These flags represent link classifications or attributes. HIGHWAY allows the user to specify whether links belonging to a class (i.e., with a specific flag set) should be included in a route. When such a link is evaluated during the route generation process, it receives a penalty. In some cases, this will be a slight increase so that use of the link is discouraged. In other cases, the link will receive a full 70,000-unit increase to its impedance, which will force the algorithm to avoid the link unless there is no feasible alternative.

Table 6 lists the flags available for each link, along with the action that HIGHWAY takes if the flag is considered in route selection. The default program status indicates whether HIGHWAY examines a flag by default. The reader is referred to Sects. 3.1.5.4 and 3.1.5.8 for a discussion of how to change the status of the attribute flags.

Table 6. Link flags

Link flag	Default program status	Action if considered
Prohibited for truck use	Consider	Avoid
Within urbanized area over 100,000 population	Ignore	Avoid
Within an Interstate beltway	Ignore	Avoid
HM-164 preferred route	Ignore	Penalize links not in class
Ferry crossing	Consider	Avoid
Contains a tunnel	Ignore	Avoid
Nonintersecting Interstate access	Ignore	Avoid

2.3.7 HM-164 Routing

HIGHWAY allows the user to generate routes using that portion of the highway system approved for the transportation of high-level radioactive materials under DOT docket HM-164. These routes consist of Interstate highways and designated state highways. When the HM-164 routing option is in effect, HIGHWAY examines the HM-164 flag of each link to see if it is an HM-164 approved segment. If not, a penalty is applied to the segment that discourages its use in the route. The HM-164 option also bases the link impedance strictly on travel time, rather than on a weighted combination of travel time and distance. HM-164 routing is a special case of sign type routing where all the individual weight factors are automatically defined for the user. For a further discussion of HM-164 routing, see Sect. 3.1.5.4.

2.3.8 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. This capability is discussed in Sect. 3.1.5.1.

To develop alternative routes, HIGHWAY first finds the optimal route using the user-specified criteria. The impedance of each link in the base route is then increased to encourage the algorithm to find an alternative route between the origin and destination. Normally, the program increases base route link impedances by 10%. The algorithm is then rerun. Due to the increased impedances of links included in the base route, the second route will tend to use a different set of links. To generate a third route, the impedances of the links used in the second route are also increased and the procedure is repeated. Links that appear in both routes, the base and first alternative routes, 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 18-20 demonstrate the alternative routing process. The link impedances of the path of the base route, shown in Fig. 17, between node 1 on state 1 and node 2 in state 4 are increased. For this example only, a 20% factor has been used so that the procedure will rapidly converge. Figure 18 shows the resulting link impedances. The resulting path tree, with associated impedances, is shown in Fig. 19. The alternative route reaches node 2 of state 4 with a impedance of 48.6 units. The reader is encouraged to perform the detailed calculations to verify the shortest path shown in Fig. 19. The alternative and base routes are compared in Fig. 20.

2.3.9 Population Density

The HIGHWAY 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. 5 for additional details.

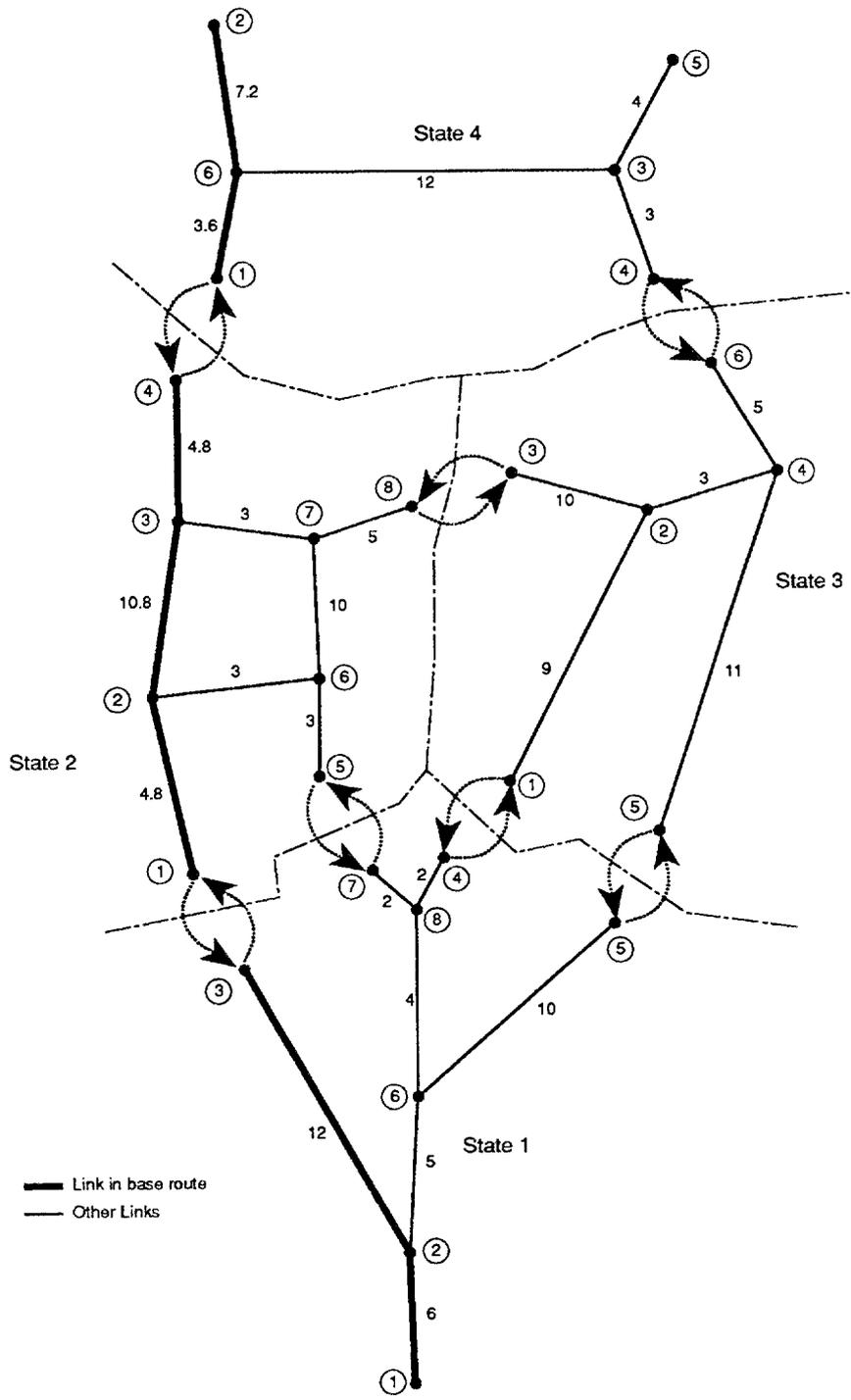


Fig. 18. Network impedance after increasing impedances on the base route.

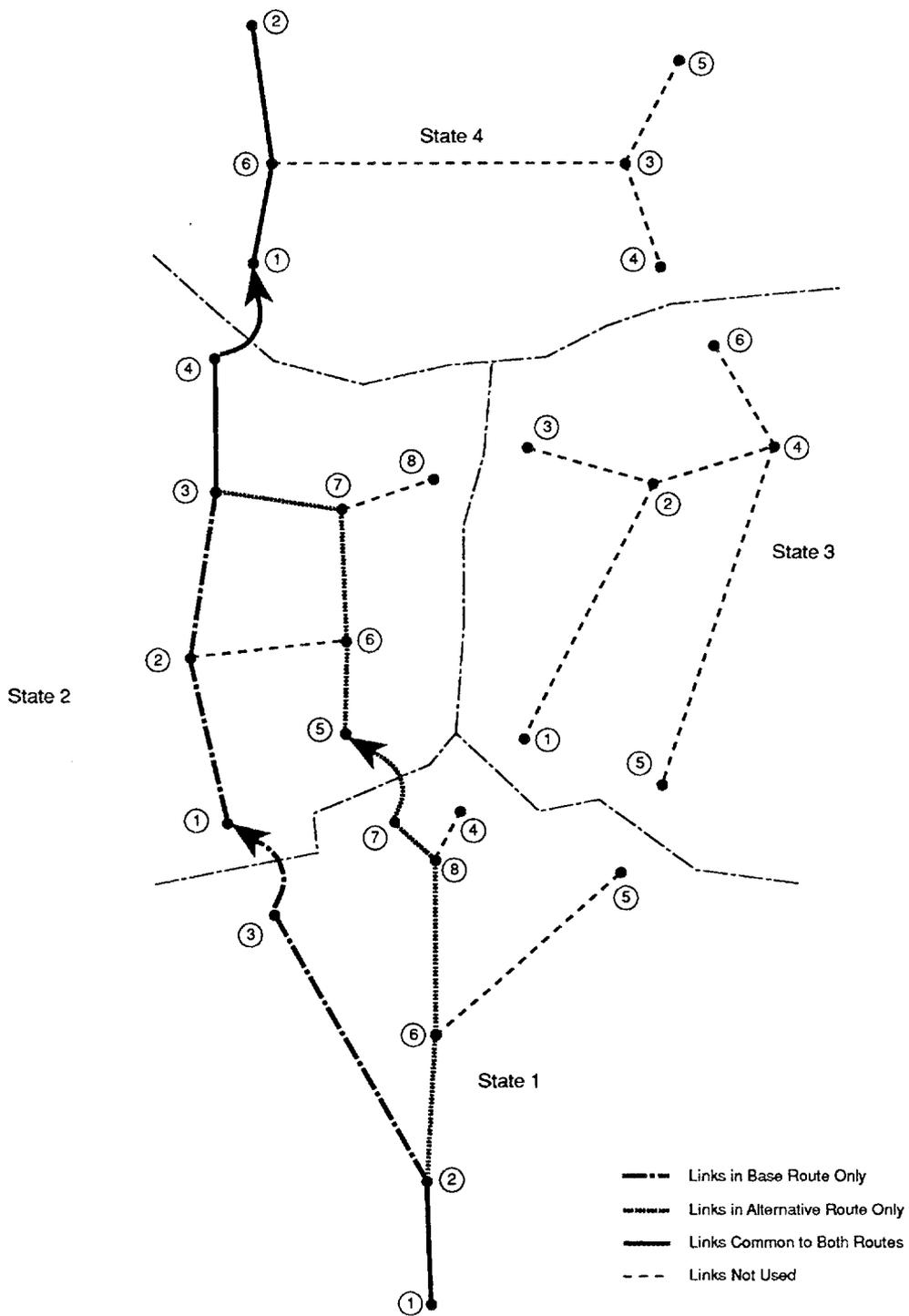


Fig. 20. Path comparison—normal vs alternative routing.

3. HIGHWAY PROGRAM USER'S MANUAL

The HIGHWAY User's Manual is written with the assumption that the user is familiar with the operation of a computer terminal or a personal computer and knows how to activate the HIGHWAY program on the appropriate computer system.

This section is designed to guide the user through the operation of the HIGHWAY routing model and to 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.

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 HIGHWAY program. Program prompts will be indicated by **BOLD UPPERCASE**. User responses will be printed in **bold lowercase**. The user may respond with uppercase, lowercase, or a combination of the two. However, for clarity, lowercase will be used in this manual. Commentary information pertaining to various command responses and options will be enclosed in brackets ([]). Additional information will be highlighted with *italics*. These conventions are used to distinguish user actions from program information. Program output is included in many of the figures. This output will be reproduced exactly as it appears on the screen.

Once the HIGHWAY routing model has been activated, a copy of the master data base is made and used for all subsequent routing calculations. This allows the user to temporarily change the data base while performing certain routing operations without permanently altering the master data base. When program operation is terminated, the temporary data file is deleted and any data base modification will be erased. If it is necessary to retain the temporary data base, the user should contact the appropriate computer system for further information and instructions.

The information shown in Fig. 21 will be displayed when the HIGHWAY program is activated.

As shown in Fig. 21, the first request for information is the prompt:

DO YOU WANT TO PLACE A PENALTY ON THE USE OF TOLL ROADS (Y/N) ?

[The default for this prompt is no.]

```
*      HIGHWAY 3.1 Highway Routing Program      *
*      Oak Ridge National Laboratory              *
```

HIGHWAY 3.1

Portions of this System are licensed for use by the United States Government and its contractors under copyrights claimed by Rand McNally-TDM, Inc. Use of this program for day-to-day operational purposes by commercial concerns for other than governmental purposes incidental to the transport of nuclear and other hazardous materials is prohibited.

Do you want to place a penalty on the use of toll roads (Y/N)?

Fig. 21. Sample of opening screen.

As discussed in Sect. 1.4.2, the use of toll roads or toll bridges can be controlled by the toll bias. This prompt gives the user the option of either activating or deactivating the use of the toll bias.

If the user activates the toll bias by answering the prompt with a **yes**, the information shown in Fig. 22 is displayed and the toll bias is set to include the standard 15% penalty. The user is referred to Sect. 3.1.2.1 or Sect. A.1.7 in the appendix if a different value of the toll bias is required.

If the user answers the toll bias prompt with a **no**, the toll bias is set to 1.0 (no penalty for use of toll roads or toll bridges) and the program displays:

INITIALIZATION COMPLETE - ENTER "HELP" FOR A COMMAND LIST.

The standard penalty of 15% will be used.
To set penalty to another value:

- 1) Do not use the standard routing features in the ROUTE command (changes toll bias for a single route), or
- 2) Use the ROUTE subcommand in the OPTION command section (see Appendix A of the User's Manual).

Initialization complete - Enter "Help" for a command list.

*** COMMAND *** Type HELP for a list of Commands

Fig. 22. Toll bias penalty prompt.

3.1 COMMANDS

After the initial question pertaining to the toll bias has been answered, the following statement is displayed:

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

At this point, the program is ready to accept specific instructions defining the actions to be taken.

Table 7 indicates the section of this report in which each of the valid commands is discussed.

Table 7. HIGHWAY commands

Section	Command
3.1.1	Help
3.1.2	Route
3.1.3	Reroute
3.1.4	Retype
3.1.5	Options
3.1.6	Advanced
3.1.7	Stop

The commands shown in Table 7 fit into three general categories—operate, help/define, and modify. Four commands—**route**, **reroute**, **retype**, and **option**—relate to program operation. The **option** command allows the user to choose different constraints that can alter the program operation. The **advanced** command permits the user to modify the data base. The **help** command is used to obtain additional information at specific locations in the program. The various actions of the **help** command will be discussed in the appropriate sections of this manual. The **stop** command terminates the program.

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**. However, in this manual all the commands will be completely spelled out.

3.1.1 Help

The **help** command will print a brief description of the basic **COMMAND** list. A copy of the output generated by the **help** command is shown in Fig. 23. The **help** command is a useful reminder of the appropriate responses for a user who is familiar with the **HIGHWAY** model.

```

HIGHWAY 3.1 Command List
This is a brief summary of the basic commands for the routing program.

"ROUTE"      Calculate a route between an origin and destination.
"REROUTE"    Calculate another route from last origin defined.
"RETYPE"     Display output for the latest route calculated.
"OPTIONS"    Defines routing constraints.
"ADVANCED"   Advanced commands for modifying the data base.
"STOP"       Exit from the program.

See the HIGHWAY User's Manual for further information.

*** COMMAND *** Type HELP for a list of Commands

```

Fig. 23. Command list.

3.1.2 Route

The **route** command, which generates a route from an origin to a destination, is the most frequently used command. A sample of the computer output generated during the **route** command is shown in Fig. 24 and will be referred to repeatedly during this discussion. In order to completely define the parameters to be used during the routing calculations, a series of prompting statements will be displayed on the monitor during the execution of the **route** command. The first prompting statement is as follows:

NOTE: A <CR> WILL RETURN YOU TO THE COMMAND STATEMENT
ENTER ORIGIN CITY,STATE (EXAMPLE ERIE,PA)

The user responds with the city and state of the origin or starting point of the route. City names may be spelled out completely or entered in an abbreviated form by deleting characters from the end of the city name. The city and state names are separated by a comma (,) and the standard two-letter post office abbreviations are used for the states. Referring to the example in Fig. 24, the desired origin is Manhattan, Kansas, which was entered as **man,ks**.

*** COMMAND *** Type HELP for a list of commands
route

NOTE: A <CR> will return you to the command statement

Enter origin City,State (Example ERIE,PA)

man,ks

1 MANHATTAN KS 2 MANHATTAN S 170 S177 KS
3 MANKATO E U36 S14 KS

Select origin city ? 1

Enter destination City,State (Example ELY,MN)

bart,ok

1 BARTLESVILLE OK 2 BARTLESVILLE E U60 U75 OK
3 BARTLESVILLE SE U60 U75 OK

Select destination city ? 1

Do you want to change the current routing options - which are:
Commercial route, 0% toll bias, and 2 drivers (Y/N) ? no

By default the program uses the current date and time.
Do you want to set the departure date and time (Y/N) ? no
<<Routing>>

From: MANHATTAN KS Leaving : 3/30/92 at 13:17 CST
to : BARTLESVILLE OK Arriving: 3/30/92 at 18:41 CST

Route type: C with 2 driver(s) Total road time: 5:24
Time bias: .70 Mile bias: .30 Toll bias: 1.00 Total miles: 232.0

The following constraints are in effect:

- 1 - Links prohibiting truck use
- 7 - Avoid ferry crossings

State mileage:

OK 22.0 KS 210.0

Mileage by highway sign type:

Interstate: 49.0 U.S.: 174.0 State: 9.0 Turnpike: .0
County: .0 Local: .0 Other: .0

Mileage by highway lane type:

Limited Access Multilane: 49.0 Limited Access Single Lane: .0
Multilane Divided: 14.0 Multilane Undivided: .0
Principal Highway: 160.0 Through Highway: 9.0 Other: .0

Continue with route listing (Y/N) ? yes

From: MANHATTAN KS Leaving : 3/30/92 at 13:17 CST
to : BARTLESVILLE OK Arriving: 3/30/92 at 18:41 CST

Routing through:

.0	MANHATTAN	KS	.0	0:00	3/30 @ 13:17
9.0 S177	MANHATTAN	S 170 S177 KS	9.0	0:15	3/30 @ 13:32
42.0 I70	TOPEKA	W 1470 170 KS	51.0	1:00	3/30 @ 14:17
7.0 I470	TOPEKA	S I335 1470 KS	58.0	1:08	3/30 @ 14:25
63.0 U75	BURLINGTON	S U75 S57 KS	121.0	2:29	3/30 @ 15:46
1.0 U75 S57	LE ROY	W U75 S57 KS	122.0	2:30	3/30 @ 15:47
65.0 U75	INDEPENDENCE	KS	187.0	3:55	3/30 @ 17:12
3.0 U160 U75	INDEPENDENCE	W U160 U75 KS	190.0	3:59	3/30 @ 17:16
16.0 U75	HAVANA	S U166 U75 KS	206.0	4:20	3/30 @ 17:37
3.0 U166 U75	CANEY	KS	209.0	4:54	3/30 @ 18:10
21.0 U75	BARTLESVILLE	E U60 U75 OK	230.0	5:21	3/30 @ 18:38
2.0 U60	BARTLESVILLE	OK	232.0	5:24	3/30 @ 18:41

*** COMMAND *** Type HELP for a list of commands

Fig. 24. Sample output generated by using the route command.

After the user enters the origin city and state, a list of all intersections in that state with names that begin with the letters entered will be displayed. In Fig. 24, three highway intersections are listed. Two intersections are identified as Manhattan, while the third intersection listed is east of Mankato, Kansas. (Note: Mankato starts with the specified letters, "man.") The user is then asked to select a highway intersection from the list with the prompt:

SELECT ORIGIN CITY ?

For this example, the intersection at Manhattan was selected (locations without highway designations always refer to the center of the city), which corresponds to the first entry in the origin city list. Hence, a 1 was entered next to the question mark. A number must be entered even if the origin city list contains only a single entry. If the origin city list does not contain the desired entry, the selection process can be restarted by entering a zero (0) after the question mark.

After an origin has been selected, the program will ask for the destination city and state with the prompt:

ENTER DESTINATION CITY,STATE (EXAMPLE ELY,MN)

Following the example in Fig. 24, the desired destination was Bartlesville, Oklahoma, which was entered as bart,ok. Three highway intersections associated with Bartlesville were reported, and the first entry was selected by entering a 1 after the question mark of the following statement:

SELECT DESTINATION CITY ?

As with the origin prompt, a number must be entered even if the city list contains only a single entry.

At this time, the end points of the route have been defined, and the user will be asked to select the type of route, the toll bias, and number of drivers. The following statement will be displayed on the monitor:

**DO YOU WANT TO CHANGE THE CURRENT ROUTING OPTIONS - WHICH ARE:
COMMERCIAL ROUTE, 0% TOLL BIAS, AND 2 DRIVERS (Y/N) ?**

[The default for this prompt is no.]

Here the user has some flexibility in the information that is to be entered. The current values for these parameters are shown in the prompt and will be used if this question is answered with a no or a <cr>. Instructions for changing these parameters will be given in the next section.

The final prompting statement is as follows:

**BY DEFAULT THE PROGRAM USES THE CURRENT DATE AND TIME.
DO YOU WANT TO SET THE DEPARTURE DATE AND TIME (Y/N) ?**

[The default for this prompt is no.]

The computer output includes a listing of the date and time a shipment is expected to reach various points along the route. Frequently, the departure date and time are not important, and the user can simply answer this question with a **no** or a **<cr>**, and the program will use the current time and date. The default time will be the current clock time for the computer being used. All times are expressed using a 24-h clock convention. Instructions for specifying a departure time and date are given in the next section.

In the example shown in Fig. 24, both prompts were answered with a **no**. At this point, the word **<<ROUTING>>** will be displayed, signifying that all questions have been answered satisfactorily and that the routing calculations are being performed. The length of the delay before results are displayed on the monitor is a function of the computer used to run the program and the particular origin and destination selected.

When the calculations have been completed, a brief summary will be printed showing the origin, destination, departure and arrival times, type of route, number of drivers, time bias, distance bias, other routing constraints, toll bias, estimated driving time, and total distance. The mileage traveled in each state is also listed, along with the mileage traveled on the various highway types. A sample output is included in Fig. 24.

After the summary, the program will display the prompt:

CONTINUE WITH ROUTE LISTING (Y/N) ?

[The default for this prompt is yes.]

If the summary data already listed are sufficient, the user needs to type a **no** to return to the command level. If a route description is needed, the user types **yes**.

The first line of the route listing following the words **ROUTING THROUGH:** lists the origin and the departure date and time. Additional lines of information are printed each time the highway designation changes. For example, the second line of the output indicates that the route follows state highway 177 (S177) for 9 miles to the intersection of I-70 and S177, which is located south of Manhattan, Kansas. The cumulative distance is 9 miles, and the expected time of arrival at this intersection is 13:32 (1:32 p.m.). The next line of the printout shows that the route then follows I-70 for 42 miles to the junction of I-470 just west of Topeka, Kansas. The

cumulative distance to this point is 51 miles, and the shipment would be expected to arrive at 14:17, or 2:17 p.m. The final line lists the destination as Bartlesville, Oklahoma. The total distance of the route is 232 miles, and the estimated driving time is 5:24.

When the route listing has been completed, the program signifies that it is ready for the next command with the prompt:

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

3.1.2.1 Changing the Default Routing Parameters

In some instances, the user needs to change the default routing parameters. There are two prompting questions in the route command that give the user this capability.

The first prompting statement is as follows:

**DO YOU WANT TO CHANGE THE CURRENT ROUTING OPTIONS - WHICH ARE:
COMMERCIAL ROUTE, 0% TOLL BIAS, AND 2 DRIVERS (Y/N) ?**

[The default for this prompt is no.]

The current values of the route type, toll bias, and number of drivers parameters are included in the prompt. For this example, the values are assumed to be commercial route, no toll bias, and two drivers respectively. If the values shown in this prompt are suitable, the user responds with a **no** or a **<cr>** to proceed to the next question. To change any of these parameters, a **yes** must be entered, and the computer will respond with a series of questions to determine which parameters are to be changed and to request new data. These questions are preceded by the following statement:

**NOTE: THE FOLLOWING CHANGES APPLY ONLY TO THIS ROUTE OR
SUBSEQUENT ROUTES GENERATED BY THE REROUTE COMMAND.**

The first question is as follows:

DO YOU WANT TO CHANGE THE ROUTE TYPE (Y/N) ?

[The default for this prompt is no.]

If the user responds with a **no**, the program uses the route type shown in the first prompt, and no further route type information will be requested. If the user answers **yes**, the program will respond with:

ENTER THE ROUTE TYPE (S=SHORTEST, Q=QUICKEST, C=COMMERCIAL) ?

The user must enter an **s**, **q**, or **c** to define the appropriate route type. The next question asked is this:

DO YOU WANT TO CHANGE THE TOLL BIAS (Y/N) ?

[The default for this prompt is no.]

If the user answers **no**, the program will retain the toll bias shown in the original prompt. The value of the toll bias may be changed by answering **yes**. At this point the program will ask:

ENTER THE TOLL BIAS (VALUE MUST BE BETWEEN 0 AND 99) ?

[The default for this prompt is 0.]

The appropriate toll bias is entered after the question mark. For example, a value of 75 would place a 75% penalty on toll roads (sets the toll bias to 1.75). The next question asked is:

DO YOU WANT TO HAVE ONE DRIVER (Y/N) ?

[The default for this prompt is no.]

The user answers **yes** to specify a single driver or **no** to specify a two-driver team. This prompt affects only the current route calculation.

The next series of computer-generated statements pertains to the departure date and time.

The program will issue the following statement:

**BY DEFAULT THE PROGRAM USES THE CURRENT DATE AND TIME
DO YOU WANT TO SET THE DEPARTURE DATE AND TIME (Y/N) ?**

[The default for this prompt is no.]

A negative answer to this question will define the departure date and time as the current date and time. If a specific date or time or both are needed, the prompt must be answered affirmatively and the following prompt will appear:

**ENTER DEPARTURE DATE (MM/DD/YY) EXAMPLE 10/26/91
ENTER A <CR> FOR THE CURRENT DATE**

Entering a **<cr>** will tell the program to use the current date and ask for the time. A different date may be specified by entering the month, day, and year, using two-digit numbers separated by slashes to identify the needed information. **Two digits must be used for each entry.** As seen in Fig. 25, the date March 1, 1992, was entered as **03/01/92**. If the date was entered as **3/1/92**, the program would not interpret the information correctly and would continue to repeat the prompt until the date is specified correctly.

The departure time is specified after the next prompt:

**ENTER DEPARTURE TIME (HHMM) IN 24 HOUR CLOCK TIME
ENTER A <CR> FOR THE CURRENT TIME**

The time is entered as a four-digit number with no colon (:) separating the hour and minutes. (Please remember that a 24-h clock convention is used.) For times earlier than 1000 (10 a.m.), the leading zero must be entered (e.g., 0830 is 8:30 a.m.).

Figure 25 illustrates how the parameters are changed. In this example, the following information was specified: shortest route, 75% toll bias, two drivers, a departure date of March 1, 1992 (03/01/92), and a departure time of 4:45 p.m. (1645).

```

Do you want to change the current routing options - which are:
Commercial route, 0% toll bias, and 2 drivers (Y/N) ? yes

NOTE: Any of the following changes apply only to this route or subsequent
routes generated by the REROUTE command

Do you want to change the route type (Y/N) ? yes
Enter the route type (S=Shortest, Q=Quickest, C=Commercial) ? s
Do you want to change the toll bias (Y/N) ? yes
Enter the toll bias (value must be between 0 and 99) ? 75
Do you want to have one driver (Y/N) ? no

By default the program uses the current date and time.
Do you want to set the departure date and time (Y/N) ? yes

Enter departure date (MM/DD/YY) Example 10/26/91
Enter a <CR> for the current date
03/01/92

Enter departure time (HHMM) in 24 hour clock time
Enter a <CR> for the current time
1645

```

Fig. 25. Example of changing routing parameters.

3.1.3 Reroute

The **reroute** command is extremely useful for generating a series of routes from the same origin. The first route must be calculated using the **route** command; however, subsequent routes can then be generated quickly using the **reroute** command since all the data calculated for the previous route are retained and reused in subsequent calculations. Response time using the **reroute** command is significantly shorter than for the original routing calculation.

It should be noted that the parameters such as origin city and state, route type, toll bias, and number of drivers cannot be changed while using the **reroute** command. The values of these parameters were established during the prior **route** command and apply to all subsequent routing calculations generated by the **reroute** command.

One other aspect that extends the usefulness of the **reroute** command is route symmetry. Since the path between any two points is not a function of the direction traveled, the origin and destination may be reversed with no loss of information. For example, if a number of routes are needed from several different origins to a common destination, the **reroute** command may be used if the actual destination is considered as the origin for routing purposes.

The prompting statements (see Fig. 26) for execution of the **reroute** command are very similar to those for the **route** command. The first prompt that appears after typing **reroute** is this:

ENTER DESTINATION CITY/STATE (EXAMPLE ELY,MN)

The user types the appropriate city and state abbreviation, as discussed for the **route** command. The same rules apply. The next statement asks the user to make a selection:

SELECT DESTINATION CITY ?

Again, a number associated with the appropriate city in the list must be entered after the question mark. The first prompting statement is as follows:

**BY DEFAULT THE PROGRAM USES THE CURRENT DATE AND TIME
DO YOU WANT TO SET THE DEPARTURE DATE AND TIME (Y/N) ?**

[The default for this prompt is no.]

The user simply enters a **no** or a **<cr>** if the current date and time are desired. An actual date and time can also be entered at this point if needed (see instructions in previous section).

Figure 26 is a continuation of the example given in Fig. 24 in which a route between Manhattan, Kansas, and Bartlesville, Oklahoma, was calculated. For Fig. 26, a route between Manhattan, Kansas, and Cheyenne, Wyoming, was generated using the **reroute** command, since the origin is the same as that used in the previous run. Note that the specification of the origin city, route type, time bias, and other parameters is not requested. By definition, the **reroute** option assumes the values specified in the last execution of the **route** command (i.e., values defined in Fig. 24).

*** COMMAND *** Type HELP for a list of commands
reroute

Enter destination City,State (Example ELY,MN)

chey,wy

```

1 CHEYENNE          WY  2 CHEYENNE      NW I25 U85 WY
3 CHEYENNE          S  I80 U85  WY  4 CHEYENNE      SW I25 I80 WY
    
```

Select destination city ? 1

By default the program uses the current date and time.
Do you want to set the departure date and time (Y/N) ? no

<<Routing>>

```

From: MANHATTAN      KS      Leaving : 3/30/92 at 13:17 CST
to : CHEYENNE       WY      Arriving: 3/31/92 at 0:16 MST
    
```

```

Route type: C with 2 driver(s)          Total road time: 11:59
Time bias: .70 Mile bias: .30 Toll bias: 1.00 Total miles: 550.0
    
```

The following constraints are in effect:

- 1 - Links prohibiting truck use
- 7 - Avoid ferry crossings

State mileage:

```

WY 43.0 NE 303.0 KS 204.0
    
```

Mileage by highway sign type:

```

Interstate: 298.0 U.S.: 252.0 State: .0 Turnpike: .0
County: .0 Local: .0 Other: .0
    
```

Mileage by highway lane type:

```

Limited Access Multilane: 298.0 Limited Access Single Lane: .0
Multilane Divided: 2.0 Multilane Undivided: .0
Principal Highway: 250.0 Through Highway: .0 Other: .0
    
```

Continue with route listing (Y/N) ? yes

```

From: MANHATTAN      KS      Leaving : 3/30/92 at 13:17 CST
to : CHEYENNE       WY      Arriving: 3/31/92 at 0:16 MST
    
```

Routing through:

```

.0      MANHATTAN      KS      .0  0:00  3/30 @ 13:17
16.0 U24      RILEY      E  U24  U77  KS  16.0  0:21  3/30 @ 13:38
4.0 U24      U77      RILEY      KS  20.0  0:26  3/30 @ 13:43
112.0 U24      OSBORNE     N  U24  U281  KS  132.0  2:52  3/30 @ 16:10
22.0 U281      SMITH CENTER  KS  154.0  3:21  3/30 @ 16:38
29.0 U36      PHILLIPSBURG E  U183 U36  KS  183.0  3:59  3/30 @ 17:16
1.0 U183      U36      PHILLIPSBURG KS  184.0  4:00  3/30 @ 17:18
26.0 U183      ALMA      NE  210.0  5:04  3/30 @ 18:22
2.0 U136      U183      ALMA      N  U136 U183  NE  212.0  5:07  3/30 @ 18:25
38.0 U183      ELM CREEK  S  I80  U183  NE  250.0  5:57  3/30 @ 19:14
298.0 I80      CHEYENNE  S  I80  U85  WY  548.0  11:52  3/31 @ 0:09
2.0 U85      CHEYENNE  WY  550.0  11:59  3/31 @ 0:16
    
```

*** COMMAND *** Type HELP for a list of commands

Fig. 26. Use of the reroute command.

3.1.4 Retype

Occasionally, extra copies of a route listing are desired. The **retype** command will automatically output another copy of the last route generated. The **retype** command eliminates the necessity of rerunning the route calculation.

The **retype** command is particularly useful if a detailed output, which shows all the nodes and links used on a route, is needed for the previously calculated route. Detailed output can be specified using the **option** command (see Sect. 3.1.5.2) and the data displayed by using the **retype** command.

To receive another copy of the last route generated, type **retype** after the COMMAND request is printed:

```
*** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS
```

3.1.5 Options

The **option** command is used to set up all routing parameters and constraints. Nine codes are defined to perform a number of operations. New values for the parameters changed with the **option** command remain in effect until further changes are made or until the calculation session has ended. When the user first activates the HIGHWAY program, all parameters are set to their default values. A list of the nine option codes is included in Table 8. The reader is referred to Sect. 3.1.5.5 for a discussion of the default values for these parameters. Additional specialized options are discussed in Sect. A.2.

Table 8. Option codes

Section	Option code	Description
3.1.5.1	AL	Allow calculation of alternative routes
3.1.5.2	DE	Detailed route listings
3.1.5.3	DR	Change the number of drivers
3.1.5.4	HM	Establish HM-164 routing constraints
3.1.5.5	OP	Show current parameter values
3.1.5.6	PD	Prompt for population density table
3.1.5.7	PW	Change HM-164/state preferred route weight
3.1.5.8	SH	Set detailed shipment parameters
3.1.5.9	RE	Return to the command statement

To use the **option** command, the user enters **option** after the **COMMAND** statement:

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

The computer will then respond with the following:

OPTION ROUTINE

**NOTE: PARAMETERS CHANGED VIA THE OPTION COMMAND REMAIN IN EFFECT
UNTIL A SUBSEQUENT CHANGE IS MADE**

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

If the user types **help** at this prompt, a list of options along with a brief explanation will be displayed. This list is very similar to the information shown in Table 8.

3.1.5.1 AL (Alternative Route) Code

The **al** code provides the user with the option of generating alternative routes between an origin and destination. To generate alternative routes, the user answers the prompt

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

with **al**. The program then asks the user:

DO YOU WANT TO GENERATE ALTERNATIVE ROUTES (Y/N) ?

[The default for this prompt is no.]

If the user answers **yes**, the program will then prompt:

**CURRENT PENALTY FOR PREVIOUSLY USED LINKS IS 10%
ENTER NEW VALUE (0 IF NO CHANGE WANTED)?**

It is recommended that the user not change the default 10% penalty for previously used links. This feature has been thoroughly tested, and satisfactory results are obtained for HM-164 routes by using the default 10% penalty. Therefore, the user should respond to this prompt with a **<cr>** or a zero (**0**). If the user wants to change the penalty, a number between 1 and 99 must be entered.

After the alternative routing capability has been activated, a base route must be calculated using the **route** command. The base route **cannot** be generated with a **reroute** command. Attempts to use **reroute** with the **al** option activated will return the user to the command statement.

The base route establishes the origin, destination, and any routing constraints. The base route for a series of alternative routes will be identified in the route summary as **BASE ROUTE**

FOR ALTERNATIVE ROUTING. At the end of the route listing, the user will be asked whether an alternative route is desired by the prompt:

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

[The default for this prompt is no.]

To calculate an alternative route, the user must answer **yes**. The program will then automatically calculate an alternative route and indicate which alternative route calculation has been processed (see Fig. 27).

Answering the alternative route prompt with a **no** or a **<cr>** will return the user to the **COMMAND** statement.

Since the alternative routing algorithms use information from previously calculated routes, the user is not permitted to change any of the routing constraints during a series of alternative route calculations. Note that in Fig. 27, the alternative route did not ask the user about changes for route types, toll bias, or number of drivers. Alternative routes are calculated using the same parameters and constraints that were used in the base route.

3.1.5.2 DE (Detailed Route Listing) Code

The **de** code will produce a detailed listing of the route. While such a listing is useful under certain circumstances, the **de** code will produce a lengthy output. An example of the standard and detailed listings for a route between Knoxville and Memphis, Tennessee, is shown in Fig. 28. In this example, the standard listing consists of only 7 lines, while the detailed listing contains more than 40 lines. One can see that a detailed list of a cross-country route could easily cover many pages.

To generate detailed listings, the user responds to the prompt

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

by typing **de**. The computer will respond with the following:

DETAILED ROUTE LISTING (ALL LINKS AND NODES) (Y/N) ?

[The default for this prompt is no.]

The user responds with a **yes** or a **no**. Once turned on, the **de** option will remain in effect until turned off by a subsequent use of the **de** code.

[Base route]

From: Dubuque IA Leaving : 3/10/92 at 20:09 CST
 To : Dalhart TX Arriving: 3/11/92 at 16:26 CST

Route type: C with 2 driver(s) Total road time: 20:17
 Time bias: .70 Mile bias: .30 Toll bias: 1.00 Total Miles: 913.0

Base route for alternative routing

The following constraints are in effect:

- 1 - Links prohibiting truck use
- 7 - Avoid ferry crossings

State mileage:

TX 51.0 OK 56.0 KS 418.0 MO 112.0 IA 276.0

Mileage by highway sign type:

Interstate: 409.0 U.S.: 484.0 State: 20.0 Turnpike: .0
 County: .0 Local: .0 Other: .0

Mileage by highway lane type:

Limited Access Multilane: 492.0 Limited Access Single Lane: .0
 Multilane Divided: 1.0 Multilane Undivided: .0
 Principal Highway: 420.0 Through Highway: .0 Other: .0

Continue route listing (Y/N) ? no

Do you want to calculate an alternative route (Y/N) ? yes

[Alternative Route]

From: Dubuque IA Leaving : 3/10/92 at 20:09 CST
 To : Dalhart TX Arriving: 3/11/92 at 16:55 CST

Route type: C with 2 driver(s) Total road time: 20:47
 Time bias: .70 Mile bias: .30 Toll bias: 1.00 Total Miles: 931.0

Alternative route number 1 generated with weighting factor of .100

The following constraints are in effect:

- 1 - Links prohibiting truck use
- 7 - Avoid ferry crossings

State mileage:

TX 51.0 OK 56.0 KS 425.0 MO 112.0 IA 287.0

Mileage by highway sign type:

Interstate: 525.0 U.S.: 353.0 State: 53.0 Turnpike: .0
 County: .0 Local: .0 Other: .0

Mileage by highway lane type:

Limited Access Multilane: 529.0 Limited Access Single Lane: .0
 Multilane Divided: 11.0 Multilane Undivided: .0
 Principal Highway: 391.0 Through Highway: .0 Other: .0

Continue route listing (Y/N) ? no

Do you want to calculate another alternative route (Y/N) ? no

*** COMMAND *** Type HELP for a list of commands

Fig. 27. Alternative routing example.

From: Knoxville TN Leaving: 3/07/92 at 0:10 EST
 To : Memphis TN Arriving: 3/07/92 at 6:49 CST

[Standard Route Listing]

.0		KNOXVILLE		TN	.0	0:00	3/07 @	0:10
3.0	140	KNOXVILLE	W	I40	1640	TN 3.0	0:04	3/07 @ 0:13
18.0	140	175 OAK RIDGE	S	I40	175	TN 21.0	0:23	3/07 @ 0:33
156.0	140	NASHVILLE	E	I24	140	TN 177.0	3:14	3/07 @ 2:23
2.0	I24	140 NASHVILLE	SE	I24	140	TN 179.0	3:16	3/07 @ 2:26
1.0	I40	165 NASHVILLE	S	I40	165	TN 180.0	3:18	3/07 @ 2:27
212.0	I40	MEMPHIS		TN	392.0	7:39	3/07 @	6:49

[Detailed Route Listing]

.0		KNOXVILLE		TN	.0	0:00	3/07 @	0:10
2.0	140	KNOXVILLE	SW	I40	U129	TN 2.0	0:03	3/07 @ 0:12
1.0	I40	KNOXVILLE	W	I40	1640	TN 3.0	0:04	3/07 @ 0:13
5.0	140	175 WEST HILLS		I40	X380	TN 8.0	0:09	3/07 @ 0:19
3.0	I40	175 KNOXVILLE	W	I40	S162	TN 11.0	0:13	3/07 @ 0:22
10.0	I40	175 OAK RIDGE	S	I40	175	TN 21.0	0:23	3/07 @ 0:33
4.0	I40	LENOIR CITY	NW	I40	S95	TN 25.0	0:28	3/07 @ 0:37
7.0	I40	KINGSTON	E	I40	S58	TN 32.0	0:35	3/07 @ 0:45
9.0	I40	HARRIMAN	SW	I40	U27	TN 41.0	0:45	3/07 @ 0:55
18.0	I40	CRAB ORCHARD				TN 59.0	1:05	3/07 @ 1:14
7.0	I40	CROSSVILLE	E	I40	S101	TN 66.0	1:13	3/07 @ 0:22
5.0	I40	WOODLAWN				TN 71.0	1:18	3/07 @ 0:27
30.0	I40	COOKEVILLE	S	I40	S111	TN 101.0	1:51	3/07 @ 1:00
30.0	I40	GORDONSVILLE		I40	S53	TN 131.0	2:23	3/07 @ 1:33
19.0	I40	LEBANON	SE	I40	U70	TN 150.0	2:44	3/07 @ 1:54
2.0	I40	LEBANON	S	I40	U231	TN 152.0	2:46	3/07 @ 1:56
6.0	I40	MARTHA	SE	I40	S109	TN 158.0	2:53	3/07 @ 2:02
17.0	I40	NASHVILLE	E	I40	S155	TN 175.0	3:11	3/07 @ 2:21
2.0	I40	NASHVILLE	E	I24	140	TN 177.0	3:14	3/07 @ 2:23
2.0	I24	140 NASHVILLE	SE	I24	140	TN 179.0	3:16	3/07 @ 2:26
1.0	I40	165 NASHVILLE	S	I40	165	TN 180.0	3:18	3/07 @ 2:27
2.0	I40	NASHVILLE	W	I265	140	TN 182.0	3:20	3/07 @ 2:30
1.0	I40	NASHVILLE	W	I40	1440	TN 183.0	3:21	3/07 @ 2:31
25.0	I40	FAIRVIEW	NW	I40	S96	TN 208.0	3:49	3/07 @ 2:58
10.0	I40	POMONA	SE	I40	S46	TN 218.0	4:00	3/07 @ 3:09
9.0	I40	PINEWOOD	N	I40	S48	TN 227.0	4:09	3/07 @ 3:19
20.0	I40	LOBELVILLE	N	I40	S13	TN 247.0	5:01	3/07 @ 4:11
17.0	I40	MC ILLWAIN	SW	I40	U641	TN 264.0	5:20	3/07 @ 4:29
18.0	I40	CLARKSBURG	S	I40	S22	TN 282.0	5:39	3/07 @ 4:49
21.0	I40	JACKSON	NE	I40	U70	TN 303.0	6:02	3/07 @ 5:12
5.0	I40	JACKSON	N	I40	U45	TN 308.0	6:08	3/07 @ 5:17
2.0	I40	JACKSON	N	I40	S58	TN 310.0	6:10	3/07 @ 5:19
1.0	I40	JACKSON	NW	I40	U412	TN 311.0	6:11	3/07 @ 5:20
13.0	I40	LEIGHTON	NW	I40	U70	TN 324.0	6:25	3/07 @ 5:34
10.0	I40	BROWNSVILLE	S	I40	S76	TN 334.0	6:36	3/07 @ 5:45
38.0	I40	BARTLETT	E	I40	U64	TN 372.0	7:18	3/07 @ 6:27
8.0	I40	MEMPHIS	E	I240	140	TN 380.0	7:26	3/07 @ 6:36
4.0	I40	MEMPHIS	NE	I40	S14	TN 384.0	7:31	3/07 @ 6:40
5.0	I40	MEMPHIS	N	I40	S300	TN 389.0	7:36	3/07 @ 6:45
2.0	I40	MEMPHIS	N	I40	S14	TN 391.0	7:38	3/07 @ 6:47
1.0	I40	MEMPHIS		TN	392.0	7:39	3/07 @	6:49

Fig. 28. Examples of standard and detailed route listings.

3.1.5.3 DR (Number of Drivers) Code

The number of drivers used to calculate travel time can be changed with the **dr** code. Although the default value is two drivers (essentially around-the-clock driving), a single driver can be specified with this option. Note that the number of drivers can be defined for a single route calculation, as described in Sect. 3.1.2.1; however, if a number of routes using a single driver are desired, the **dr** option is a useful way to set the number of drivers. To change the number of drivers, the user responds to the prompt

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

by typing **dr**. The program will respond with this prompt:

ENTER NUMBER OF DRIVERS (1 OR 2) ?

[The default for this prompt is 2.]

The user must respond with a **1** or a **2**, as shown in Fig. 29.

```
Enter option code or type HELP for assistance
dr
Enter number of drivers (1 or 2) ? 2
```

Fig. 29. Use of the dr code.

3.1.5.4 HM (HM-164 Routing Constraint) Code

The **hm** code allows the user to generate routes based on DOT routing regulations (49 CFR 177.825) for the shipment of route-controlled quantities of radioactive materials. These routes, which primarily follow Interstate highways, are frequently called HM-164 routes. States have the right to define other alternative preferred highways that either augment or replace portions of the Interstate network. If the origin or destination of a route or both origin and destination are not located adjacent to an Interstate highway, the most direct route between the facility and the preferred routing network must be used.

To generate HM-164 routes, the user responds to the prompt

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

by entering **hm**. The program will respond with:

DO YOU WANT TO GENERATE ROUTES BASED ON HM-164 (Y/N) ?

[The default for this prompt is no.]

The user must respond with either a **yes** or a **no**. An example of an HM-164 route was included in Sect. 1.4.5.

3.1.5.5 OP (Parameter Values) Code

The **op** code will output a list of the current values for the routing parameters. The default values that are assigned at the start of the program are shown in Fig. 30. The parameters can be changed with the various options discussed in this chapter.

Figure 30 contains much information. The time and distance biases for the various route types are reported. The commercial, quickest, and shortest route types are discussed in Sect. 1.4.1. The remaining route type, other (O), is provided for a user to establish different combinations of the time and distance biases. The reader is referred to Sect. A.2.6 for a discussion of setting these parameters. The current route type is shown to be equal to C, which means that the commercial route criterion will be used to calculate the next route. The number of drivers is set to 2, indicating a two-driver team.

```

Enter option code or type HELP for assistance
op
      Current default values for options

Route   Time   Dist
Type    Bias   Bias
C       .70    .30
Q       1.00   .00
S       .00    1.00
O       .50    .50
Current route type = C
Number of drivers = 2
HM-164 routing = F
Alternative routing = F

Detailed listings = F
Population density prompt = F
Toll bias for all routes = 1.00
Penalty on non-preferred roads = 10.0
Penalty on previously used links = 1.10

The following constraints are in effect:
1 - Links prohibiting truck use
7 - Avoid ferry crossings

```

Fig. 30. Output for op code—current default values of routing parameters.

A number of switches are also listed. These switches control whether HM-164 routes will be calculated, whether alternative routes will be calculated, whether detailed route listing will be reported, and/or whether population statistics will be calculated. In Fig. 30, all of these switches are set to FALSE (F), that is, turned off. If any of these capabilities were activated, the corresponding switch would be set to TRUE (T).

Finally, the values of three weighting factors used in route calculations are also reported. These factors control the toll bias, the penalty on nonpreferred roads, and the penalty on previously used links.

The toll bias is discussed in Sect. 3.1.2.1. Since the toll bias is set to a value of 1.00, no penalty will be applied to toll roads. The penalty on nonpreferred roads is used in the calculation of HM-164 routes (Sect. 3.1.5.4), and the value of this penalty can be adjusted with the **pw** option (Sect. 3.1.5.7). The penalty on previously used links is part of the alternative routing algorithm. Adjustment of this value was discussed in Sect. 3.1.5.1.

Two additional constraints are also reported in Fig. 30—"Links prohibiting truck use" and "Avoid ferry crossings." Constraints that appear in this list are active and will influence route selection. These specialized constraints are discussed in Sect. 3.1.5.8, the shipment option.

A second set of routing parameters is shown in Fig. 31. In this case, the **hm** option has been activated; that is, the next route will be HM-164 route. Activating this option results in a number of changes in the parameter settings. The HM-164 route switch is now set to TRUE (T), and the route type is set to Quickest (Q). Also, two additional constraints have been added to the constraint list—HM-164/state preferred routes and nonintersecting Interstate access. The first of these constraints restricts the route to the HM-164 preferred routing network. The nonintersecting Interstate access constraint will prevent routes from using a short section of non-Interstate highway between Interstate highways, if the respective Interstate highways do not have a common interchange. An example of nonintersecting Interstates involve I-475 and I-80/I-90 southwest of Toledo, Ohio. These Interstates do not intersect at an interchange. To change between Interstates at this point, a shipment would have to follow another route for a short distance.

All of these changes are automatically inserted when the HM-164 routing capability is activated. Deactivating the HM-164 capability will automatically remove these items.

3.1.5.6 PD (Population Density) Code

The **pd** code allows the user the option of requesting a population density table for the most recently calculated route. To activate the population density capability, the user enters **pd** after the option code prompt:

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

Enter option code or type HELP for assistance
op

Current default values for options

Route Type	Time Bias	Dist Bias	
C	.70	.30	
Q	1.00	.00	
S	.00	1.00	
O	.50	.50	

Current route type = Q
 Number of drivers = 2
 HM-164 routing = T
 Alternative routing = F

Detailed listings = F
 Population density prompt = F
 Toll bias for all routes = 1.00
 Penalty on non-preferred roads = 10.0
 Penalty on previously used links = 1.10

The following constraints are in effect:

- 1 - Links prohibiting truck use
- 6 - HM-164/State preferred routes
- 7 - Avoid ferry crossings
- 11 - Nonintersecting Interstate Access

Fig. 31. Output for op code—routing parameters for calculating HM-164 routes.

The program will then respond with the following:

DO YOU WANT TO SEE POPULATION DENSITY INFORMATION FOR ROUTES (Y/N)?

[The default for this prompt is no.]

The user may answer with a **yes** or a **no**.

Once activated, the following prompt will appear after the route calculations are completed:

DO YOU WANT A POPULATION DENSITY TABLE (Y/N) ?

[The default for this prompt is yes.]

If a population density table is desired, the user must respond with a **yes** or a **<cr>**. A negative response will suppress the generation of the population density table.

A population density table for a route between Milwaukee, Wisconsin, and Denver, Colorado, is shown in Fig. 32. 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. 32, the route between Milwaukee and Denver is 1043 miles long. Approximately 204 miles of this route travels through areas with a population density between 0 and 5 people per square mile and slightly over 340 miles of the route passes through areas with a population density between 5.0 and 22.7 people per square

From: Milwaukee WI Leaving: 3/18/92 at 19:25 CST
 To : Denver CO Arriving: 3/19/92 at 15:24 MST

Routing through:

.0	MILWAUKEE	WI	.0	0:00	3/18 @ 19:25
6.0 194	MILWAUKEE	W 1894 194	WI	6.0 0:08	3/18 @ 19:32
4.0 1894	MILWAUKEE	SW 143 1894	WI	10.0 0:12	3/18 @ 19:37
61.0 143	BELOIT	NE 143 190	WI	71.0 1:19	3/18 @ 20:43
3.0 190	SOUTH BELOIT	E 190 U51	IL	74.0 1:22	3/18 @ 20:46
17.0 190 \$	CHERRY VALLEY	NE 139 190	IL	91.0 1:41	3/18 @ 21:05
25.0 139	ROCHELLE	SE 139 188	IL	116.0 2:08	3/18 @ 21:32
34.0 188 \$ TEWT\$	ROCK FALLS	SE 188 U30	IL	150.0 2:45	3/18 @ 22:09
44.0 188	RAPIDS CITY	S 180 188	IL	194.0 3:33	3/18 @ 22:57
175.0 180	DES MOINES	N 1235 135	IA	369.0 7:14	3/19 @ 2:38
14.0 135 180	DES MOINES	W 1235 135	IA	383.0 7:29	3/19 @ 2:53
120.0 180	COUNCIL BLUFFS	SE 129 180	IA	503.0 10:10	3/19 @ 5:34
2.0 129 180	COUNCIL BLUFFS	SW 129 180	IA	505.0 10:12	3/19 @ 5:36
353.0 180	BIG SPRINGS	SW 176 180	NE	858.0 17:07	3/19 @ 11:31
179.0 176	COMMERCE CITY	W 125 176	CO	1037.0 20:53	3/19 @ 15:16
6.0 125	DENVER	CO	1043.0 21:00	3/19 @ 15:24	

Do you want a population density table (Y/N) ? **yes**

Population Density from: MILWAUKEE WI
 to : DENVER CO

		----- 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

CO	183.0	.0	110.0	32.7	15.5	8.1	5.7	4.3	1.9	1.4	3.3	.0	.0
NE	354.0	.0	55.2	178.2	59.9	23.4	13.6	8.6	2.6	7.7	4.9	.0	.0
IA	308.0	.0	36.1	103.0	80.2	37.7	21.3	17.2	11.2	1.5	.0	.0	.0
IL	124.0	.0	2.6	20.5	46.1	30.1	10.0	13.8	1.0	.0	.0	.0	.0
WI	74.0	.0	.0	6.0	13.0	11.2	14.3	13.9	5.7	.6	4.5	4.0	.9

Totals
 1043.0 .0 203.9 340.4 214.7 110.5 64.8 57.7 22.3 11.2 12.6 4.0 .9

Percentages
 .0 19.5 32.6 20.6 10.6 6.2 5.5 2.1 1.1 1.2 .4 .1

Basis: 1980 Census data

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

RADTRAN Input Data	Rural	Suburban	Urban	Total
Weighted Population				
People/sq. mi.	28.8	686.4	5618.8	
People/sq. km.	11.1	265.0	2169.4	
Distance				
Miles	869.5	156.0	17.6	1043.0
Kilometers	1399.2	251.0	28.3	1678.5
Basis (people/sq mi.)	<139	139-3326	>3326	1980 Census data

Fig. 32. Routing calculation including population density table.

mile. This route also travels 183 miles within the state of Colorado. Over 60% of the distance in Colorado (110 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 useable in the RADTRAN risk analysis code when the following question is displayed:

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

[The default for this prompt is no.]

The user must answer yes if RADTRAN data are desired or a no if these data are not needed. An example of RADTRAN input data for the same route is also included in Fig. 32. 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.

3.1.5.7 PW (Preferred Route Weight) Code

The **pw** code allows the user to change the nonpreferred route multiplier used to calculate HM-164/state preferred routes. The impedance equation for calculating these routes is shown in Eq. (4):

$$L = \text{Min} \sum_i (\alpha D_i + \beta T_i) \gamma_i, \quad (4)$$

where

- L = total impedance of a route;
- α = distance bias;
- D_i = distance of segment i , miles;
- β = time bias;
- T_i = time required to travel along segment i , minutes;
- γ_i = nonpreferred route multiplier, with
 - $\gamma_i = 1.0$ if preferred route,
 - $\gamma_i = 10.0$ if nonpreferred route.

The default value of the nonpreferred route multiplier (γ) has been determined empirically and ensures that the calculated route will conform to HM-164 requirements. Lower values of the multiplier (e.g., 2 or 3) may result in routes that do not conform with the DOT regulations. It is recommended that the value of the nonpreferred route multiplier not be changed.

To change the nonpreferred route multiplier, the user responds to the prompt

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

with **pw**. The program will then display the statement:

**CURRENT HM-164/STATE PREFERRED ROUTE WEIGHT IS 10.0
DO YOU WANT TO CHANGE THIS VALUE (Y/N) ?**

[The default for this prompt is no.]

If the user answers **yes**, the program will respond with the statement:

ENTER NEW WEIGHT ?

The user must respond with a new value for the weighting factor (e.g., 8.5, 2, or 20).

As with the other option codes, changes made to the nonpreferred route multiplier will remain in effect until another change is made or the program is terminated.

3.1.5.8 SH (Shipment Parameters) Code

The **sh** code allows the user to set various constraints that will control the route calculation procedure. Two constraints are automatically set when the program is started—"Links prohibiting truck use" and "Avoid ferry crossings." A number of highways in the United States do not permit through commercial truck traffic. Two examples are the Utah state highway 9 through Zion National Park and I-66 inside the I-495 beltway around Washington, D.C. The "no truck" constraint prevents the use of these links. The **HIGHWAY** data base contains a ferry crossing between Long Island, New York, and Connecticut. The "avoid ferry constraint" prevents routes from using this ferry.

There are a number of other constraints that can be activated with the shipment parameter. The user may constrain routes from passing through urban areas with populations exceeding 100,000; from traveling within beltways that circle many metropolitan areas (e.g., Indianapolis, Indiana, or Minneapolis/St. Paul, Minnesota); or from passing through the tunnels identified in the data base that prohibit hazardous materials.

If the user wishes to activate or deactivate any of the constraints controlled by this option, **sh** must be entered after the following prompt:

ENTER OPTION CODE OR TYPE HELP FOR ASSISTANCE

The computer will then respond with the following:

ENTER: CHECK, SET, ZERO, OR RETURN (C/S/Z/R) ?

If the user enters a **< cr >**, the program will list a brief description of each option available under this code (see Fig. 33).

```
Enter option code or type HELP for assistance
sh

Shipment Sub-option
    This option allows the user to set various flags to avoid links.

Enter: Check, Set, Zero, or Return (C/S/Z/R) ?
The functions of the subcommands are:
    Check - List the flags that are set
    Set   - Set or remove individual flags
    Zero  - Zero out (remove) all flags
    Return - Exit this routine and return to option statement

Enter: Check, Set, Zero, or Return (C/S/Z/R) ?
```

Fig. 33. Subcommands functions for sh code.

After this information is displayed, the program again prompts the user to make a valid choice by displaying this prompt:

ENTER: CHECK, SET, ZERO, OR RETURN (C/S/Z/R) ?

Check. The check command (c) will display a list of the constraints that are currently activated.

Set. The set command (s) allows the user to activate or deactivate any constraint. If the s command is used, the program will list all available constraints and ask the user to enter the number or numbers of the constraint to be activated or deactivated. As shown in Fig. 34, the number of the constraint to be modified must be followed with a slash (/) (e.g., 2/). Multiple constraints can be set at the same time by separating the constraint numbers with a comma (e.g., 3,7,8/). To deactivate a constraint, the constraint number is entered as a negative number (-1/).

Zero. The zero command will deactivate all active constraints. This includes constraints 1 and 7, which remove links prohibiting truck use and ferry crossings. Note that these constraints are automatically activated when the HIGHWAY program is started.

Return. This command will return the user to the option code prompt.

```
Enter: Check, Set, Zero, or Return (C/S/Z/R) ? c
The following constraints are in effect:
 1 - Links prohibiting truck use
 7 - Avoid ferry crossings
Enter: Check, Set, Zero, or Return (C/S/Z/R) ? s
The following flags are available:
 1-No trucks                2-Avoid urbanized areas over 100,000
 3-Block within beltways    7-Avoid ferry crossings
 7-Avoid ferry crossings    8-Block tunnels
Enter the flag number(s) to set (Enter as negative to clear) end with /
2/
Enter: Check, Set, Zero, or Return (C/S/Z/R) ? c
The following constraints are in effect:
 1 - Links prohibiting truck use
 2 - Links in urbanized areas over 100,000
 7 - Avoid ferry crossings
Enter: Check, Set, Zero, or Return (C/S/Z/R) ? r
Enter option code or type HELP for assistance
```

Fig. 34. Use of check and set subcommands.

As noted in Sect. 3.1.5.5, two additional constraints are listed when the HM-164 routing capability is activated, and these constraints will appear in the list generated by the **set** command. These constraints are entitled 6—HM-164/state preferred routes and 11—nonintersecting Interstate access. While these constraints can be activated and deactivated via the **set** command, it is recommended that the user **always** use the **hm** option to activate or deactivate the HM-164 routing capability.

3.1.5.9 RE (Return) Code

The **re** code will terminate the **option** command and return the user to the basic command statement.

3.1.6 Advanced Routing Commands

The basic routing commands discussed in the previous sections allow the user to perform most of the routing operations needed for day-to-day operations. Occasionally, certain applications require a more in-depth routing analysis. The advanced routing commands discussed in this section provide the additional capability for modifying the data base in order to analyze more complicated routing problems. The user should be familiar with the basic commands,

discussed in Sects. 3.1.2 through 3.1.5, before trying to use the commands discussed in this section. Table 9 shows the **advanced** commands that can be accessed.

Table 9. Advanced commands

Section	Command
3.1.6.1	Remove
3.1.6.2	D node
3.1.6.3	D link
3.1.6.4	Reset
3.1.6.5	Links

When the user types **adv** following the prompt

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

the program will list the **advanced** commands available. This listing is shown in Fig. 35. After the listing is completed, the program will return to the command statement, expecting the user to respond with one of the five commands shown in Fig. 35 or any of the other commands discussed in this manual.

```

*** COMMAND *** Type HELP for a list of commands
advanced

HIGHWAY 3.1 Advanced Command List

"REMOVE" Remove a state from or reactivate a deleted state
to the data base.
"D LINK" Remove a link from or reactivate a deleted link to the
data base.
"D NODE" Remove a node from or reactivate a deleted node to the
data base.
"RESET" Resets all the deleted links or nodes.
"LINKS" Trace link and node connections.

NOTE: Refer to the HIGHWAY User's Manual for further information
on these advanced commands. They are not as user friendly as
the other more frequently used commands.

```

Fig. 35. Example of advanced command list.

For simplicity, the **advanced** commands will be discussed in the following order: **remove, d node, d link, reset, and links.**

3.1.6.1 Remove

The **remove** command is one of three commands that can be used interactively to alter the user's personal copy of the HIGHWAY data base. The **remove** command may be used either to delete an entire state network from the data base or to reinsert a state that was previously deleted.

At the start of any HIGHWAY routing session, all states are in the data base; that is, the removal status of each state is set to false. If a state is removed, the removal status is changed to true. The removal status for that state will remain true unless the state is reinserted with the **remove** command or the calculations are terminated with the **stop** command. When the model is executed at a later time, the data base again includes all states.

If a number of states are to be removed from the data base, the user must make sure that a route exists between the origin and destination. If none exists, execution of the **route** command will result in an error message being displayed.

After the **remove** command has been entered, the computer will respond with the primary prompt:

THIS COMMAND WILL DELETE A STATE FROM OR ADD BACK A PREVIOUSLY DELETED STATE INTO THE DATA BASE.

ENTER: ADD, DELETE, LIST, RETURN (A/D/L/R) ?

To remove a state, the user enters **delete**. The computer will respond with this prompt:

ENTER ABBREVIATION OF THE STATE TO BE DELETED ?

The two-letter state abbreviation must be entered after the question mark. For example, if **mo** is entered, the program will respond with the following message to indicate that the state of Missouri has been deleted:

MO HAS BEEN DELETED FROM THE DATA BASE.

The program will then return to the basic prompt shown above and request another subcommand. To delete multiple states, the user will have to use the **delete** subcommand for each state to be deleted.

The addition of a previously deleted state to the data base is accomplished by entering **add** after the basic prompt. The computer will display this prompt:

ENTER ABBREVIATION OF THE STATE TO BE ADDED ?

Again, the two-letter state abbreviation must be entered after the question mark. The program will respond with a message indicating that the state has been added.

Occasionally, the user wishes to ask the computer to display the states currently deleted from the data base. This can be done by entering the subcommand **list** after the primary prompt. The computer will respond with a list showing any deleted states.

The **return** subcommand will return the user to the command statement.

An example of using the **remove** command is shown in Fig. 36, where two states (Missouri and Kansas) were deleted. The **list** subcommand was used to list the deleted states, and then the **add** subcommand was used to reinsert the Missouri network. The **return** subcommand was used to terminate the operation of the remove command and display the command statement.

```

*** COMMAND *** Type HELP for a list of commands
remove

This command will delete a state from or add back a previously deleted state
into the data base

Enter: Add, Delete, List, Return (A/D/L/R) ? delete
Enter abbreviation of the state to be deleted ? mo
mo has been deleted from the data base

Enter: Add, Delete, List, Return (A/D/L/R) ? delete
Enter abbreviation of the state to be deleted ? ks
ks has been deleted from the data base

Enter: Add, Delete, List, Return (A/D/L/R) ? list

The following states are deleted from the data base:
    MO
    KS

Enter: Add, Delete, List, Return (A/D/L/R) ? add
Enter abbreviation of the state to be added ? mo
mo has been added to the data base

Enter: Add, Delete, List, Return (A/D/L/R) ? list

The following states are deleted from the data base:
    KS

Enter: Add, Delete, List, Return (A/D/L/R) ? return

*** COMMAND *** Type HELP for a list of commands

```

Fig. 36. Example of remove command functions.

3.1.6.2 D Node

The **d node** command will effectively remove specific highway intersections or nodes from the data base by placing a large impedance on all of the links emanating from that node. After a node has been deleted, all routes will bypass that particular node unless no other route exists. The **d node** command effectively blocks any shipments through a particular town or city.

This command actually alters the user's personal copy of the HIGHWAY data base; however, deleted nodes can also be reinserted into the data base by this command. If a large number of nodes have been removed, the user may find it easier to use the **reset** command (described in Sect. 3.1.6.4) to restore the nodes.

If the user's personal copy of the HIGHWAY data base contains any deleted nodes or links when the HIGHWAY program is activated, a message will be displayed indicating the data base contains some deletions. As shown in Fig. 37, this message occurs before the user is able to perform any routing calculations. If the user needs a clean (i.e., one with no deletions), the **reset** command (Sect. 3.1.6.4) can be used to reset any deleted nodes or links.

```

*      HIGHWAY 3.1 Highway Routing Program      *
*      Oak Ridge National Laboratory            *
-----
HIGHWAY 3.1
-----

Portions of this System are licensed for use by the United States
Government and its contractors under copyrights claimed by Rand
McNally-TDM, Inc. Use of this program for day-to-day operational
purposes by commercial concerns for other than governmental purposes
incidental to the transport of nuclear and other hazardous
materials is prohibited.

**** Data base contains      5 deletions ****

Do you want to place a penalty on the use of toll roads (Y/N)?

```

Fig. 37. Sample of link deletion notice.

A series of examples will be given in this discussion to demonstrate the execution of the **d node** command. Following the prompt

```
*** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS
```

the user types **d node**. (Note the blank space between the **d** and the **n**.) The program responds with a brief set of instructions, and the user will be asked to identify a city and state:

NODE DELETE/REPLACE ROUTINE

ENTER CITY,ST FOR LIST. THEN CHOOSE INDIVIDUAL CITIES OR A RANGE. DELETIONS ARE PERFORMED UNLESS A "+" IS ENTERED IN COLUMN 7.

ENTER CITY,ST OR <CR> TO END

The user enters a city and a state name in the same way locations were identified as origins and destinations of routes (see **route** command, Sect. 3.1.2). In Fig. 38, the city of Pewaukee, Wisconsin, is specified by entering **pew,wi**. The program then prints a list of all cities in Wisconsin starting with the letters **pew**, and the user is asked to select the desired city or cities as follows:

SELECT ONE CITY, A RANGE, OR <CR> TO END

-- -- -

Since only a single Pewaukee was found, either a **1** or a **01** must be entered under the first two dashes. The computer continues to ask the user to select a city from the list until all the desired

[Removing a single node]

*** COMMAND *** Type HELP for a list of commands
d node

Node delete/replace routine
 Enter City, ST for list. Then choose individual cities or a range.
 Deletions are performed unless a "+" is entered in column 7.

Enter City,ST or <CR> to end

pew,wi

1 PEWAUKEE S16 S190 WI

Select one city, a range, or <CR> to end

-- -- -

01

Select one city, a range, or <CR> to end

-- -- -

Enter City,ST or <CR> to end

1 Nodes Deleted

0 Nodes Replaced

Fig. 38. Use of the **d node** command to remove a single node.

nodes have been deleted, and the user terminates this action by entering a <cr>. The user will then be asked to specify another location. If no other cities are needed, a <cr> will end the **d node** operation and return the user to the command statement after typing a message showing the number of nodes deleted or replaced.

The second example (Fig. 39) shows how to delete a number of nodes rapidly. In this case, all three nodes in the vicinity of Marshalltown, Iowa, are to be deleted. When the program prompts the user to select a city or a range, the user responds with a range of nodes (**01 03**). This signifies that all cities from no. 1 through no. 3 in the list are to be deleted.

[Removing a range of nodes]

```

*** COMMAND ***  Type HELP for a list of commands
d node

Node delete/replace routine
Enter City, ST for list.  Then choose individual cities or a range.
Deletions are performed unless a "+" is entered in column 7.

Enter City,ST or <CR> to end
mars,ia
  1  MARSHALLTOWN          IA      2  MARSHALLTOWN   S  U30  S14  IA
  3  MARSHALLTOWN   SW  U30  S330  IA

Select one city, a range, or <CR> to end
-- -- -
01 03

Select one city, a range, or <CR> to end
-- -- -

Enter City,ST or <CR> to end

      3 Nodes Deleted          0 Nodes Replaced

```

Fig. 39. Use of the **d node** command to remove a range of nodes.

The more complicated third example (Fig. 40) deletes a number of nodes (i.e., the first five and the eighth) in the vicinity of Minneapolis, Minnesota. The range feature was used to delete the first five nodes by typing **01 05** at the first request for a city, while the eighth node was deleted by typing **08** on the second request for a city.

 [Reactivating a range nodes]

```

*** COMMAND ***  Type HELP for a list of commands
d node

Node delete/replace routine
Enter City, ST for list.  Then choose individual cities or a range.
Deletions are performed unless a "+" is entered in column 7.

Enter City,ST or <CR> to end
minn,mn
 1 MINNEAPOLIS                MN    2 MINNEAPOLIS        N  I94  C81  MN
 3 MINNEAPOLIS        NW I94  S55  MN    4 MINNEAPOLIS        S  I35W I94  MN
 5 MINNEAPOLIS        S  S77  C62  MN    6 MINNEAPOLIS        SE I35W I94  MN
 7 MINNEAPOLIS        SE I494 S5    MN    8 MINNEAPOLIS        SE S55  C62  MN
 9 MINNEAPOLIS        W  I394 I94  MN   10 MINNETONKA         I494 S7    MN
11 MINNETONKA         N  I494 U12  MN

Select one city, a range, or <CR> to end
-- -- -
01 05 +

Select one city, a range, or <CR> to end
-- -- -

Enter City,ST or <CR> to end

      0 Nodes Deleted          5 Nodes Replaced

*** COMMAND ***  Type HELP for a list of commands

```

Fig. 41. Use of the d node command to reinsert nodes into the data base.

3.1.6.3 D Link

The **d link** command effectively removes a specific highway segment or link from the data base by increasing the impedance of that link to a large value. After a link has been deleted, routes will tend to bypass that segment of highway. Deleted links are removed from the data base until they are reactivated by using this command. If a large number of links have been removed, the user will find it easier to use the **reset** command (Sect. 3.1.6.4) to restore deleted links.

If the user's personal copy of the data base contains deleted links when the **HIGHWAY** program is activated, a warning message will be displayed. Please see the discussion in Sect. 3.1.6.2 for more details.

Examples of the **d link** command operation are shown in Figs. 42 and 43. Figure 42 shows how a particular link may be removed, and Fig. 43 illustrates reinsertion of a link using the **d link** command.

After the command statement, the user enters **d link**. (Note the blank space between the **d** and **l**.) The computer types a short set of instructions and then requests that the user specify a city and state:

LINK DELETION/INSERTION ROUTINE - CURRENT MODE IS 2-WAY

ENTER CITY,ST FOR NODE LIST. CHOOSE A NODE AND THE LINKS EMANATING FROM THAT NODE WILL BE DISPLAYED. THEN CHOOSE LINKS FROM LIST. DELETIONS ARE PERFORMED UNLESS A "+" IS ENTERED IN COLUMN 10.

ENTER CITY,ST OR <CR> TO END

The user enters a city and state name, using the rules previously described in the **route** command section. As illustrated in Fig. 42, a list of nodes is displayed, and the user is asked to select the desired intersection:

SELECT A CITY OR <CR> TO END

The number of the intersection is then entered.

All highway segments associated with the selected intersection will then be listed. In Fig. 42, two intersections are identified for Hays, Kansas. At this point, a good road atlas, should be consulted to visualize the physical layout of the highway network for that city. The first intersection, the junction of I-70 and U.S. 183 north of Hays, Kansas, was selected in this example. The first link listed is I-70 heading west to Wakeeney, Kansas; the second link is U.S. 183 heading south to the junction of U.S. 183 and a local road south of Hays; the third link is U.S. 183 heading north to Stockton, Kansas; and the last link is I-70 heading east to Russell, Kansas.

The user will be asked to specify the links to be altered:

ENTER LINK#, A RANGE, OR <CR> TO END

M D

--- -- - -

The specific link to be deleted is entered beneath the first set of dashes as a single- or two-digit number (e.g., **1** or **01**). A range of links can be deleted by entering the numbers of the first and last link selected under the first two sets of dashes (see Sect. 3.1.6.2). The **ENTER LINK #** prompt will be repeated until the user has completed all of the desired deletions. At this point,

[Removing links]

```

*** COMMAND *** Type HELP for a list of commands
d link

Link deletion/insertion routine - Current mode is 2-way

ENTER CITY, ST FOR NODE LIST. CHOOSE A NODE AND THE LINKS EMANATING
FROM THAT NODE WILL BE DISPLAYED. THEN CHOOSE LINKS FROM LIST.
DELETIONS ARE PERFORMED UNLESS A "+" IS ENTERED IN COLUMN 10.

Enter City,ST or <CR> to end
hays,ks
1 HAYS          N 170 U183 KS   2 HAYS          S U183 LOCL KS

Select a city or <CR> to end
1

Links from HAYS          N 170 U183 KS

1 Hwys:170          55 mph   33 miles to WAKEENEY          KS
2 Hwys:U183         28 mph    3 miles to HAYS          S U183 LOCL KS
3 Hwys:U183         46 mph   38 miles to STOCKTON          KS
4 Hwys:170          55 mph   25 miles to RUSSELL          S 170 U281 KS

Enter Link#, a range, or <CR> to end
M D
-- -- --
01

Enter Link#, a range, or <CR> to end
M D
-- -- --
04

Enter Link#, a range, or <CR> to end
M D
-- -- --

Enter City,ST or <CR> to end

4 Links Deleted          0 Links Inserted

*** COMMAND *** Type HELP for a list of commands

```

Fig. 42. Example of removing links.

the user simply enters a <cr> to end this operation. The computer will then respond with this prompt:

ENTER CITY,ST OR <CR> TO END

Another city can be selected, or a <cr> will return the user to the command statement if all desired deletions have been made. In the example shown in Fig. 42, links 1 and 4 were deleted in separate operations. Note that, although only two links were deleted, the **d link** report shows that four links were deleted. From a mathematical standpoint, it is convenient to represent

each segment by two unidirectional links. Traffic can flow along a highway in two directions; thus to block flow along a particular highway segment, both of the unidirectional links must be deleted. For this reason, if the current deletion mode is two-way, the number of deleted links reported will always be twice the number of highway segments the user has actually removed.

An example of reinserting a deleted link is shown in Fig. 43. The responses for adding a link are identical to those described for deletion, with one exception. When the computer responds with

ENTER LINK#, A RANGE, OR <CR> TO END

M D

-- -- - -

a plus sign (+) must be entered in column 10, under the dash labeled with an M.

[Restoring links]

*** COMMAND *** Type HELP for a list of commands

d link

Link deletion/insertion routine - Default mode is 2-way

Enter City,ST or <CR> to end

hays,ks

1 HAYS N 170 U183 KS 2 HAYS S U183 LOCL KS

Select a city or <CR> to end

1

Links from HAYS N 170 U183 KS

1	Hwys:170	55 mph	33 miles to WAKEENEY	KS
2	Hwys:U183	28 mph	3 miles to HAYS	S U183 LOCL KS
3	Hwys:U183	46 mph	38 miles to STOCKTON	KS
4	Hwys:170	55 mph	25 miles to RUSSELL	S 170 U281 KS

Enter Link#, a range, or <CR> to end

M D

-- -- - -
1 +

Enter Link#, a range, or <CR> to end

M D

-- -- - -

Enter City,ST or <CR> to end

0 Links Deleted 2 Links Inserted

*** COMMAND *** Type HELP for a list of commands

Fig. 43. Example of link insertion.

Since each link is actually composed of two unidirectional links, it is possible to prevent routes from using only one of the unidirectional links. See Appendix A for additional information about one-way link deletions.

3.1.6.4 Reset

The **reset** command resets all deleted links and nodes in the data base. At the command prompt

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

the user enters **reset**. The program will then respond with this prompt:

RESET BLOCKED LINKS OR NODES ON FILE (Y/N) ?

[The default for this prompt is no.]

If the user responds with a **yes**, the program will reset any blocked links and/or nodes and will then display the statement

LINK FILE RESET AS REQUESTED

before returning to the command prompt.

3.1.6.5 Links

The **links** command traces link and node connections. While this command may not be used frequently, there are occasions when the name used in the HIGHWAY data base for a particular location is not readily recognizable. This is particularly true for highway intersections located away from any towns or for locations in suburbs of major cities. The **links** command allows the user to trace the highway network from a location where the highway name is known to the point of interest.

The best way to illustrate the use of the **links** command is through an example such as that shown in Fig. 44. There is an intersection in southeastern Nevada whose name is not apparent from a highway map. The junction of U.S. 93, state highway 375, and state highway 318, approximately 100 miles north of Las Vegas, could be called either Ash Springs or Hiko. There are two courses available. The user can try both names in the **route** command and see if either is listed, or the highway network can be traced from an intersection where the highway name is obvious. From a map, it is obvious that there must be a highway intersection identified as Ely, Nevada, since Ely is a major town. In Fig. 44, the highway network is traced from Ely to the intersection of interest.

```

*** COMMAND *** Type HELP for a list of commands
links
Enter City,ST or <CR> to end
ely,nv
1 ELY NV 2 ELY AIRPORT NV
Select a city or enter a <CR> to end
1
Links from ELY NV
1 Hwys:U6 46 mph 118 mi. to WARM SPRINGS NV
2 Hwys:U50 46 mph 80 mi. to EUREKA NW U50 S278 NV
3 Hwys:U50 U6 28 mph 1 mi. to EAST ELY NV
Enter Link# or a <CR> to enter a new node name ? 1
Links from WARM SPRINGS NV
1 Hwys:S375 37 mph 98 mi. to HIKO S U93 S375 NV
2 Hwys:U6 46 mph 118 mi. to ELY NV
3 Hwys:U6 46 mph 37 mi. to TONOPAH E U6 LOCL NV
Enter Link# or a <CR> to enter a new node name ?
Enter City,ST or <CR> to end
*** COMMAND *** Type HELP for a list of commands

```

Fig. 44. Example of link and node tracking.

After the prompt:

***** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS**

the user enters **links** to begin tracing link connections. The program then prompts with

ENTER CITY,ST OR <CR> TO END

and the user responds with a city and a state name. The convention used here for entering a city name is identical to that previously described for the **route** command. In Fig. 44, the city name was entered as **ely,nv**, two locations in Nevada having Ely as the first three letters were found. The first is the town of Ely, Nevada, and the second is the Ely airport, which is located north of town. The town was selected by entering a **1**. The three highway segments (or links) leading away from Ely are listed next as shown in Fig. 44. The program will request the user select a link by prompting:

ENTER LINK# OR A <CR> TO ENTER A NEW NODE NAME ?

One of the computer-listed links goes southwest 118 miles to Warm Springs, Nevada. However, this is not the intersection of interest. After entering a **1** to request Warm Springs, the links leading from Warm Springs are displayed (Fig. 44). The first link (no. 1), which leads to Hiko,

is the intersection desired. In general, this process is repeated until the user has traced all the link connections desired.

To exit from the **links** command, a **<cr>** must be entered rather than a link number. The computer will respond with another request for a city and state. If no further searches are required, the user enters a **<cr>** to return to the command statement.

3.1.7 Stop

After all of the desired routing calculations have been completed, the execution of the HIGHWAY model is terminated by entering **stop** at the command prompt:

```
*** COMMAND *** TYPE HELP FOR A LIST OF COMMANDS
```

The program will respond with several lines of type as shown in Fig. 45.

```
*** COMMAND *** Type HELP for list of commands  
stop
```

```
-----  
HIGHWAY 3.1  
-----
```

```
Portions of this System are licensed for use by the United States  
Government and its contractors under copyrights claimed by Rand  
McNally-TDM, Inc. Use of this program for day-to-day operational  
purposes by commercial concerns for other than governmental purposes  
incidental to the transport of nuclear and other hazardous  
materials is prohibited.
```

```
Stop - Program terminated.
```

Fig. 45. The stop command.

REFERENCES

1. Joy, D. S.; Johnson, P. E. *HIGHWAY, A Transportation Routing Model: Program Description and Revised User's Manual*, ORNL/TM-8159, October 1983.
2. Newhauser, K. S.; Kanipe, F. L. *RADTRAN 4: Volume 3, User Guide*, SAND89-2370, January 1992.
3. Even, S. *Graph Algorithms*, Rockville, MD: Computer Sciences Press, 1979.
4. Reingold, E. M.; Nieverelt, J.; Deo, N. *Combinational Algorithms: Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1977.
5. Durfee, R. C.; Coleman, P. R. *Population Distribution Analyses for Nuclear Power Plant Siting*, NUREG/CR-3056, December 1983.

Appendix. SPECIALIZED FEATURES

Appendix. SPECIALIZED FEATURES

This appendix discusses several highly specialized features not included in the HIGHWAY User's Manual, Sect. 3, primarily because the normal use of the model does not require a knowledge of these aspects of the program.

A.1 SPECIALIZED OPTION COMMANDS

Several specialized options of HIGHWAY are not discussed in the user's manual portion of this document; these commands are listed in Table A.1. While in the option command, the user can enter **sp** to obtain a list of the specialized commands. As shown in Fig. 30 of Sect. 3 of this report, the default settings of the specialized options are not displayed with the **op** code within the **option** command. However, if any of the specialized options are activated, then the **op** code will list the current settings of all options including the specialized options discussed in this appendix. An example of the expanded **op** code reporting is shown in Fig. A.1, where the **im** option is activated to report the route impedance.

The use and operation of the specialized features are discussed in this section.

Table A.1. Specialized option codes

Section	Option code	Description
A.1.1	DI	Set link deletions 1 or 2 way
A.1.2	DL	Alert to deleted source or sink
A.1.3	IM	Print the impedance of the route
A.1.4	IN	Generate routes following Interstate System highways
A.1.5	MA	Prompt for map file
A.1.6	PR	Queue routes for printing
A.1.7	RO	Route criterion options
A.1.8	WE	Allow weights for highway type in routing

A.1.1 DI (Directional Link Deletion) Code

The default status of link deletion in the HIGHWAY model is for two-way traffic to be blocked when a link is removed via the **d link** command. The **di** code provides the specialized user the capability to block routes from using a link in one direction while allowing a route to

Current default values for options

Route Type	Time Bias	Dist Bias	
C	.70	.30	
Q	1.00	.00	
S	.00	1.00	
O	.50	.50	

Current route type = C
 Number of drivers = 2
 HM-164 routing = F
 Alternative routing = F
 Map prompt = F
 Impedance reporting = T
 Deletion warning = T
 Weight links by lane type = F

Detailed listings = F
 Population density prompt = F
 Toll bias for all routes = 1.00
 Penalty on non-preferred roads = 10.0
 Penalty on previously used links = .10
 2-way link deletions = T
 Print = F
 Weight links by sign type = F

The following constraints are in effect:
 1 - Links prohibiting truck use
 7 - Avoid ferry crossings

Enter option code or type HELP for assistance

Fig. A.1. Expanded op code report including specialized features.

incorporate the link for routes traveling in the other direction. For example, if a unidirectional block is placed in the link connecting node A with node B, routes traveling from A toward B will avoid the A-B link. However, a route from B to A could still use the link. When the **di** code is entered by the user, the program will respond with this prompt:

MAKE LINK DELETIONS 2-WAY (Y/N) ?

[The default for this prompt is yes.]

To change link deletions to one-way, the user must respond **no**. To return to the default status of two-way deletion for links, the user must respond with **yes** or a **<cr>**.

When the link deletion has been changed to one-way, removal of links proceeds as discussed in Sect. 3.1.6.3 of this report, with the exception that links selected from the chosen node are removed in only one direction, from the chosen node to the end point of the link. For example, in Fig. 42 (Sect. 3.1.6.3) the node north of Hays, Kansas, was selected, and link number 1 to Wakeeney and link number 4 to Russell were removed. With link deletions set to one-way, traffic from Hays to Wakeeney and from Hays to Russell would be blocked. Traffic in the opposite direction (Wakeeney to Hays and from Russell to Hays) would be allowed.

A.1.2 DL (Deleted Link) Code

The deleted link code informs the user if either the origin or destination of a route is the end point of a link that has been removed via the **d link** or **d node** command. This feature is useful if the user has been removing numerous links or nodes and wants to be assured that a subsequent origin or destination of a route has not been previously removed. An example of the deleted node message is shown in Fig. A.2. (See Figs. 38 and 39 for nodes that were deleted for this example.)

The deleted link message can be turned off by using the **dl** code. It is strongly recommended that this feature not be deactivated. When the **dl** code is entered by the user, the program will respond with this prompt:

ANNOUNCE DELETED LINKS EMANATING FROM ORIG OR DEST (Y/N) ?

[The default for this prompt is yes.]

To deactivate this feature, the user would respond with **no**.

```

*** COMMAND *** Type HELP for a list of commands
route

NOTE: A <CR> will return you to the command statement

Enter origin City,State (Example ERIE,PA)
pew,wi
1 PEWAUKEE          S16 S190 WI

Select origin city ? 1
%Warning 3 links out of 3 from this node appear deleted

Enter destination City,State (Example ELY,MN)
mars,ia
1 MARSHALLTOWN      IA    2 MARSHALLTOWN  S U30 S14 IA
3 MARSHALLTOWN     SW U30 S330 IA

Select destination city ? 2
%Warning 4 links out of 4 from this node appear deleted

```

Fig. A.2. Example of message alerting user of deleted nodes or links.

A.1.3 IM (Impedance) Code

The **im** code is used to report the impedance of the calculated route. As shown in Sect. 1.4.1, the impedance is defined as a function of distance and driving time along a particular highway segment. The routing algorithm determines the set of links between the origin and destination that minimizes the total impedance. If the user is interested in seeing the total impedance, the **im** code needs to be activated.

When the **im** code is entered as an option, the program will respond with this prompt:

DO YOU WANT TO SEE THE IMPEDANCE VALUE FOR ROUTES (Y/N) ?

[The default for this prompt is no.]

To report the route impedance, the user must respond to the prompt with **yes**. A sample of the route summary of the route listing showing the route impedance is included in Fig. A.3.

```

From: MANHATTAN          KS      Leaving : 3/30/92 at 13:17 CST
to : CHEYENNE           WY      Arriving: 3/31/92 at 0:16 MST

Route type: C with 2 driver(s)          Total road time: 11:59
Time bias: .70 Mile bias: .30 Toll bias: 1.00 Total miles: 550.0
Impedance for this route: 626.2                                     [Route impedance]

The following constraints are in effect:
1 - Links prohibiting truck use
7 - Avoid ferry crossings
State mileage:
WY 43.0 NE 303.0 KS 204.0
Mileage by highway sign type:
Interstate: 298.0 U.S.: 252.0 State: .0 Turnpike: .0
County: .0 Local: .0 Other: .0
Mileage by highway lane type:
Limited Access Multilane: 298.0 Limited Access Single Lane: .0
Multilane Divided: 2.0 Multilane Undivided: .0
Principal Highway: 250.0 Through Highway: .0 Other: .0

```

Fig. A.3. Route summary showing route impedance.

A.1.4 IN (Interstate System Highway) Code

The **in** code is used to calculate routes that preferentially follow Interstate System highways. The user should not confuse this seldom-used option with the **HM** option discussed in Sect. 3.1.5.4. The user should always use the **HM** command to generate routes following **HM-164** routing regulations. The **IN** option is used primarily when a user wants routes that preferentially follow Interstate System highways but which do not have to use Interstates

exclusively as in HM-164 routes. This code automatically sets the standard weighting function, which doubles the calculated impedance for all non-Interstate System roads. If other weighting functions are desired, the reader is referred to the we code discussed in this appendix. When the **in** code is entered by the user, the program will respond with this prompt:

GENERATE ROUTES FAVORING INTERSTATE HIGHWAYS (Y/N) ?

[The default for this prompt is no.]

To activate routing following Interstate System highways, the user must respond with **yes**.

Please note that activation of this code will result in routes that preferentially follow Interstate signed routes, not routes that are of Interstate System quality or routes that follow the HM-164 routing regulation for highway route-controlled quantities of radioactive materials.

A.1.5 MA (Map File) Code

When the map file code is activated, the user has the option of generating a sequential list of node and link numbers for a route in the data file **HIGHWAY.OUT**. The **HIGHWAY.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. The data recorded in the **HIGHWAY.OUT** file are used with an auxiliary ORNL program (not part of the **HIGHWAY** program) to generate maps for routes.

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

When the **ma** code entered by the user, the program will respond with this prompt:

PROMPT FOR MAP FILE (Y/N) ?

[The default for this prompt is no.]

When the **ma** code is activated, the program will output an additional question after a routing calculation. This question will ask the user whether the route data should be recorded in the map file.

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

A.1.6 PR (Print Route) Code

When the print route code is activated, a hard copy of all routes run on the HIGHWAY model will be created. Depending upon the computer facility on which HIGHWAY is installed, it may be difficult for the user to retrieve these hard copies; therefore, **it is recommended that the print route capability not be activated.**

When the *pr* code is entered by the user, the program will respond with this prompt:

QUEUE ROUTE LISTING TO LINE PRINTER (Y/N) ?

[The default for this prompt is no.]

To activate of the printouts of routes, the user must respond with **yes**.

A.1.7 RO (Route Options) Code

The route options code is used to modify several of the parameters that the HIGHWAY model uses to calculate routes. **It is strongly recommended that only experienced users attempt to make changes to these routing parameters.** When the *ro* code is entered, the program will respond with this prompt:

CHANGE ROUTE BIAS PARAMETERS (Y/N) ?

[The default for this prompt is no.]

If the user responds with **yes**, the program will issue the following prompts that allow the user to change the default values for the time and distance parameters of individual route types. The first prompt requests the user to identify the route type to be modified:

WHICH ROUTE TYPE (C, O, Q, OR S) ?

The user must enter one of the four route types listed; Commercial, Other, Quickest, or Shortest (see discussion in Sect. 3.1.5.5). (Route type *o* is available if the advanced user wishes to define a special route type without altering one of the standard route types.) If the user enters a letter other than one of the four listed in the prompt, the program will print **UNRECOGNIZED ROUTE TYPE** and reissue the previous prompt. After a route type has been selected (assumed to be type *o* in this example), the following prompt is issued:

ENTER NEW VALUE FOR TIME BIAS FOR ROUTE TYPE O (1 TO 99) ?

The user must enter a value from 1 to 99 representing the new time bias for the selected route type. The value of the distance bias is calculated by subtracting the time bias from 100 and dividing the result by 100. (Please refer to Eq. (2) in Sect. 2.2.2 for a discussion of how the time and distance biases are used in the routing objective function.)

The next prompt the program issues is as follows:

CURRENT DEFAULT ROUTE TYPE IS C DO YOU WISH TO CHANGE IT (Y/N) ?

[The default for this prompt is no.]

By responding yes, the user will be able to change the current default route type (commercial) to one of the other types (shortest, quickest, or other). The program will issue the following prompt:

WHICH ROUTE TYPE (C, O, Q, OR S) ?

The user must enter one of the four route types identified in the prompt. The route type selected will be used by the model until the route type is changed by the **ro** code or by the prompts in the **route** command (see Sect. 3.1.2.1) or until the program is terminated.

Questions related to the toll bias are the final set of inquires generated by the route option code. The next prompt issued by the program is as follows:

CHANGE TOLL BIAS (Y/N) ?

[The default for this prompt is no.]

If the user responds with yes, the program will prompt:

ENTER NEW VALUE (0 TO 100) ?

The user is required to enter a new toll bias at this prompt. The value of the new toll bias must be between 0 and 100. If 0 is entered, the program will not penalize the use of toll roads. Entering a value of 100 (the maximum value that this prompt will accept) will double the standard impedance for a toll road link.

A.1.8 WE (Weighted Highway) Code

The HIGHWAY model allows the user to preferentially weight links based upon the sign type of the road and/or by the lane type of the road. When the we code is entered by the user, the program will respond as follows:

ALLOW WEIGHTED ROUTING BY HIGHWAY SIGN TYPE (Y/N) ?

[The default for this prompt is no.]

If the user responds with yes, the program will weight each road segment based upon the sign type of the road. The sign types recognized by the HIGHWAY model include Interstate, U.S., state, turnpike, county, local, and other. The default weights of each sign type will then be listed followed by this prompt:

CHANGE WEIGHTS (Y/N) ?

[The default for this prompt is no.]

If the user responds **yes**, a prompt will be issued that allows the user to change any of the sign class weightings. If no value or a value of zero is entered below each category, the current weight will be retained for that particular sign type.

The next prompt the program issues is as follows:

ALLOW WEIGHTED ROUTING BY HIGHWAY LANE TYPE (Y/N) ?

[The default for this prompt is no.]

If the user responds with **yes**, the program will weight each road segment based upon the lane classification of the road. The lane types used by the HIGHWAY model include limited-access multilane, limited-access single lane, multilane divided, multilane undivided, principal highway, through highway, and other. The default weights of each lane types will then be listed followed by the prompt:

CHANGE WEIGHTS (Y/N) ?

[The default for this prompt is no.]

If the user responds **yes**, a prompt will be issued that allows the user to change any of the lane type weightings. If no value or a value of zero is entered below each category, the current weight will be retained for that particular lane type.

INTERNAL DISTRIBUTION

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