FINAL REPORT

DEVELOPMENT OF PROTOTYPE DECISION SUPPORT SYSTEMS FOR REAL-TIME FREEWAY TRAFFIC ROUTING: VOLUME I

BRIAN L. SMITH, Ph.D.
Faculty Research Scientist

CATHERINE C. McGHEE
Senior Research Scientist

MICHAEL J. DEMETSKY, Ph.D.
Faculty Research Scientist

ADEL W. SADEK
Graduate Research Assistant

V·I·R·G·I·N·I·A
VTRC
TRANSPORTATION RESEARCH COUNCIL

VIRGINIA TRANSPORTATION RESEARCH COUNCIL

Standard Title Page - Report on Federally Funded Project

1. Report No. FHWA/VTRC 98-R17	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Development of Prototype Decision Support Systems for Real-Time Freeway Traffic Routing: Volume I		5. Report Date June 1998
		6. Performing Organization Code
7. Author(s) B.L. Smith, C. C. McGhee, M.J. Demetsky, and A.W. Sadek		8. Performing Organization Report No. VTRC 98-R17
9. Performing Organization and Address Virginia Transportation Research Council 530 Edgemont Road Charlottesville, VA 22903		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. 1465-030
12. Sponsoring Agencies' Name and Address Virginia Department of Transportation 1401 E. Broad Street Richmond, VA 23219	FHWA 1504 Santa Rosa Road Richmond, VA 23239	13. Type of Report and Period Covered Final October 1995 – December 1997
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

For a traffic management system (TMS) to improve traffic flow, TMS operators must develop effective routing strategies based on the data collected by the system. The purpose of this research was to build prototype decision support systems (DSS) for the real-time development of such strategies. We used the freeway system controlled by the Suffolk (Virginia) TMS as a test case.

A routing DSS has (1) a search mechanism that allows the space of possible routing strategies to be explored thoroughly but efficiently, and (2) an evaluation routine that estimates the effectiveness of a particular strategy. We combined the search and evaluation routines to develop two DSS prototypes: a simple shock-wave DSS that required very little input data and a heuristic search/dynamic traffic assignment (DTA) DSS that demanded more input and computations but captured traffic dynamics better.

We evaluated the prototypes based on the agreement of their recommended strategies with prior expectations and their potential for real-time applications. The results are promising. For the shock-wave DSS, the diversion percentages recommended agree with prior expectations. For the heuristic search/DTA model, the results are consistent regardless of the start point for the search algorithm.

17 Key Words	18. Distribution Statement
Freeway traffic management, traffic routing, intelligent transportation systems, decision support systems, stochastic search algorithms	No restrictions. This document is available to the public through NTIS, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	38	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

FINAL REPORT

DEVELOPMENT OF PROTOTYPE DECISION SUPPORT SYSTEMS FOR REAL-TIME FREEWAY TRAFFIC ROUTING: VOLUME I

Brian L. Smith, Ph.D.
Faculty Research Scientist

Catherine C. McGhee
Senior Research Scientist

Michael J. Demetsky, Ph.D.
Faculty Research Scientist

Adel W. Sadek
Graduate Research Assistant

Virginia Transportation Research Council
(A Cooperative Organization Sponsored Jointly by the Virginia Department of Transportation and the University of Virginia)

Charlottesville, Virginia

June 1998
VTRC 98-R17

ABSTRACT

For a traffic management system (TMS) to improve traffic flow, TMS operators must develop effective routing strategies based on the data collected by the system. The purpose of this research was to build prototype decision support systems (DSS) for the real-time development of such strategies. We used the freeway system controlled by the Suffolk (Virginia) TMS as a test case.

A routing DSS has (1) a search mechanism that allows the space of possible routing strategies to be explored thoroughly but efficiently, and (2) an evaluation routine that estimates the effectiveness of a particular strategy. We combined the search and evaluation routines to develop two DSS prototypes: a simple shock-wave DSS that required very little input data and a heuristic search/dynamic traffic assignment (DTA) DSS that demanded more input and computations but captured traffic dynamics better.

We evaluated the prototypes based on the agreement of their recommended strategies with prior expectations and their potential for real-time applications. The results are promising. For the shock-wave DSS, the diversion percentages recommended agree with prior expectations. For the heuristic search/DTA model, the results are consistent regardless of the start point for the search algorithm.

DEVELOPMENT OF PROTOTYPE DECISION SUPPORT SYSTEMS FOR REAL-TIME FREEWAY TRAFFIC ROUTING: VOLUME I

Brian L. Smith, Ph.D.
Faculty Research Scientist

Catherine C. McGhee
Senior Research Scientist

Michael J. Demetsky, Ph.D.
Faculty Research Scientist

Adel W. Sadek
Graduate Research Assistant

INTRODUCTION

To serve as the foundation for its Smart Travel program, the Virginia Department of Transportation (VDOT) made a significant investment in smart traffic centers. These centers, commonly known as traffic management systems (TMSs), monitor freeway traffic flow with sensors and closed-circuit television (CCTV) and relay travel information to motorists via devices such as variable message signs (VMS). The primary purpose of a TMS is to enable urban freeways to operate as safely and efficiently as possible.

To fulfill this purpose, TMS operators must make sound decisions based on the data collected by the system. TMSs use advanced software to process raw data to provide operators with information to support their decision making. Although significant research has been dedicated to developing improved TMS hardware, relatively little effort has gone into developing improved decision support software to assist TMS operators.

One of the most important decisions operators must make is how to "control" traffic flow. Operators attempt to do this by influencing the route choice of motorists through providing traveler information. By influencing route choice, TMS operators can attempt to distribute traffic flow evenly over the entire network, resulting in better overall network performance. However, developing sound system routing strategies is a complex task that must effectively address two fundamental tasks:

1. *Searching.* A search mechanism is required that allows the "space" of all possible routing strategies to be explored efficiently but thoroughly (the need for an efficient search algorithm is especially true for a complex urban freeway network, where the number of possible routing strategies is extremely large).

2. *Evaluating.* An evaluation routine is required that estimates the anticipated effectiveness of a particular routing strategy.

Approaches to developing routing strategies tend to be at one of two extremes. At one extreme is the intuitive approach where operators depend solely on their experience in formulating routing strategies. However, as the size and scope of TMSs grow, the solution space of the routing problem becomes prohibitively large, making the problem intractable for human operators. At the other extreme are the sophisticated analytical dynamic traffic assignment (DTA) and the combined simulation/assignment approaches.[1-10] Traditionally, these approaches are used as off-line tools since their complexity renders them inappropriate for real-time applications. In this study, we attempted to strike a balance between these two extremes.

PURPOSE AND SCOPE

The purpose of this research was to develop prototype DSSs for the real-time development of freeway routing strategies. As a test case, we used the freeway system to be controlled by the Suffolk TMS. The network's high dependence on the limited number of water crossings illustrates the need for sound routing strategies. The key regional freeway traffic control decision that needs to be made is how to allocate travelers between the Hampton Roads Bridge Tunnel (HRBT) (I-64 route) and the Monitor-Merrimac Memorial Bridge Tunnel (MMBT) (I-664 route).

At the time of this study, the TMS had not been completed. Once the TMS is on-line, we will conduct the field testing phase of this effort to define the "information" strategies (e.g., the messages on the VMS, the number of signs activated) needed to implement routing strategies recommended by the routing DSS.

METHODOLOGY

This study involved six tasks:

1. Review the literature on existing approaches to developing routing strategies. The task focused on how the search and evaluation components have been addressed.

2. Define the roadway system to be modeled in the test case. We chose the freeway system to be monitored by the Suffolk TMS (the I-64/I-664 loop) (see Figure 1). We chose only the interstate routes because (1) they were the routes where real-time traffic information was expected to be available after the completion of the TMS and (2) drivers are typically unwilling to divert to unfamiliar routes such as minor arterials.

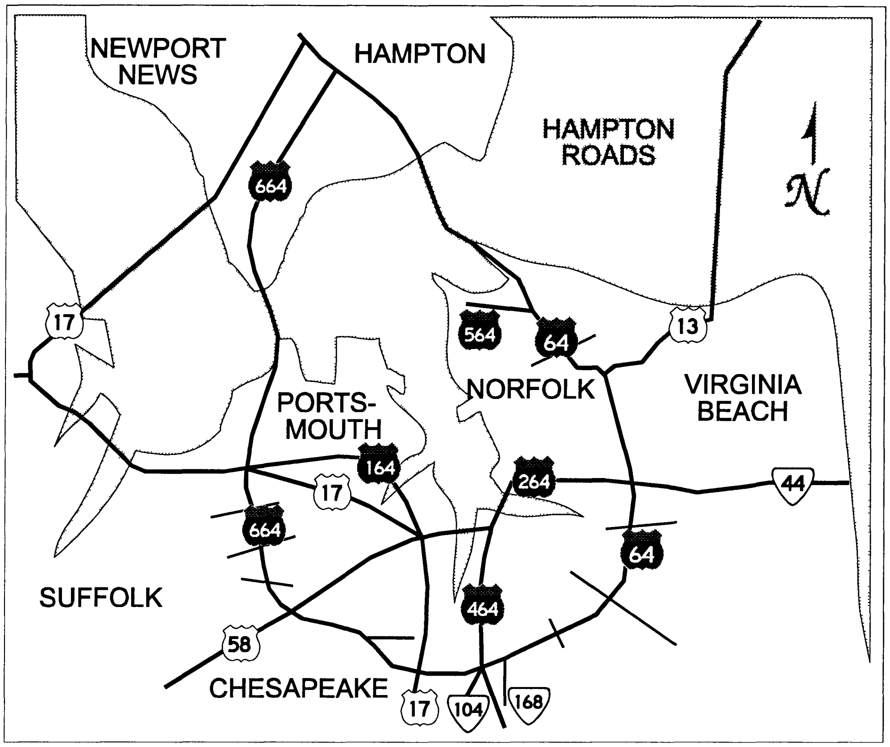

Figure 1. Suffolk TMS

The I-64/I-664 loop includes many interchanges. Including all of them in a model for evaluating routing strategies was not feasible because of data availability and complexity restraints. We, therefore, decided to limit the network to those interchanges that significantly affected the overall system. To identify significant interchanges, we used the MINUTP model developed by COMSIS for the Hampton Roads Crossing Study (HRCS)[11] since actual traffic counts were not yet available. Based on the distribution of volumes determined from the MINUTP model traffic assignment, we selected interchanges where significant changes in traffic volume or distribution occurred.

3. Identify data needs, and collect the data. Required data items included geometric data, traffic volumes, and origin-destination (O-D) estimates. We obtained some geometric data from construction plans maintained by VDOT and collected other data in the field and through discussions with VDOT district personnel. We obtained average 24-hour volumes on the network links from task 3. For short-term traffic counts, we obtained 15-min volumes from the two tunnel facilities. Finally, we obtained O-D estimates for the network modeled by manipulating the MINUTP model. We then scaled the estimates based on the short-term traffic counts obtained from the tunnels.

4. Develop search routines for determining effective routing strategies. The purpose of this task was to develop search routines to guide the exploration of the space of possible routing strategies in search of effective strategies. We chose exhaustive enumeration and two artificial intelligence stochastic search algorithms for application:

(1) genetic algorithms (GAs), which are based on the principle of survival of the fittest,[12] and (2) simulated annealing (SA), which is analogous to the process of atoms rearranging themselves in a cooling metal.[13]

5. Develop evaluation tools for testing the effectiveness of alternate routing strategies generated by the search routine. We examined two approaches that vary in accuracy, input requirements, and computational demands: (1) a simplified shock-wave model, and (2) a detailed dynamic macroscopic model of the region.

6. Develop and evaluate prototype routing DSSs. In this task, we combined the search and evaluation routines developed under tasks 4 and 5 to develop two routing decision support prototypes. We paired exhaustive enumeration with the simplified shock-wave model, resulting in a simple DSS that required very little input data. On the other hand, we linked GAs and SA to the detailed dynamic macroscopic model to yield a DSS that was more demanding in terms of input and computational requirements but promised to be more accurate than the shock-wave model. We evaluated the tools based on the agreement of their recommended strategies with prior expectations and their potential for real-time applications.

RESULTS

Literature Review

A number of avenues are being investigated by researchers in the United States, Europe, and Japan for developing effective decision support tools for real-time traffic routing. Approaches to developing routing strategy can be broadly classified into two categories: (1) the analytical dynamic traffic assignment (DTA) approach,[1-6] and (2) the combined simulation/assignment approach.[7-10] Traditionally, these approaches have been used as off-line analysis tools and not in real-time applications.

Analytical DTA Approach

In this approach, the routing problem is formulated as a mathematical programming model, commonly referred to as a DTA model. In a mathematical programming approach, a mathematical model is constructed for the system under consideration. This model takes the form of a system of equations and related mathematical expressions (commonly referred to as constraints) that describe how the system functions. Within this system of equations, the quantifiable decisions that need to be made and that affect the performance of the system are represented as decision variables whose values need to be determined. Solving a mathematical model involves determining the values for the decision variables that will optimize the system's performance. The system's measure of performance to be optimized is typically expressed as a mathematical function called the objective function.[14]

For a DTA model, the decision variables are typically the time-varying traffic volume assigned to each link (roadway segment) of the network. The objective function expresses the measure of highway network performance to be optimized (e.g., the total travel time for all vehicles), whereas the set of constraints attempts to model traffic flow in the region through the use of macroscopic traffic flow theory concepts. The formulated model is then solved, typically using a non-linear programming (NLP) technique, to obtain the routing strategy that will optimize the objective function.

There are two types of DTA models: (1) user-optimal or user-equilibrium models, and (2) system-optimal models. In user-equilibrium formulations, each user attempts to minimize his or her travel time. These models are typically based on a dynamic generalization of Wardrop's principle, which states that the individual costs along utilized routes connecting an origin to a destination are equal and minimal. In other words, people use paths of minimum cost. The goal behind user-equilibrium formulations is to replicate the patterns of traffic flows resulting from users' independent path choice decisions, and hence they are mainly used to predict traffic.

The system-optimal formulation, on the other hand, attempts to determine how traffic should be distributed in the network so as to optimize a systemwide criterion. Although the travel times of alternate routes might be unequal, the total travel time for all the vehicles in the system is minimal. It is this second type of dynamic assignment model that is more relevant to the real-time traffic routing problem for a centrally managed network similar to the one considered in this study. The Merchant and Nemhauser (M-N) model,[1] which is considered by many as the seminal work on system-optimal DTA models, illustrates the analytical DTA approach best.

Merchant and Nemhauser Model

The M-N model addresses the case of single-destination networks in which travelers from multiple origins are traveling to one destination. The purpose is to optimally assign traffic over a period of time known as the planning horizon. In the M-N formulation, the planning horizon is divided into equal time intervals of suitably small length $\{i \mid i = 0,1, \ldots I\}$. Each link of the network is assigned a cost function (h_{ij}) and an exit function g_j. When x_{ij} is the number of vehicles on link j at the beginning of time period i, it is assumed that a cost $h_{ij}(x_{ij})$ is incurred and an amount of traffic $g_j(x_{ij})$ exits from the link.

To model traffic dynamics, the function $g_j(x)$ will typically have to be concave over some part of its domain (Figure 2). This leads to non-convex model formulations that cannot be easily solved using traditional analytical approaches. The function $h_{ij}(x)$, on the other hand, should be nonnegative, nondecreasing, continuous, and convex to

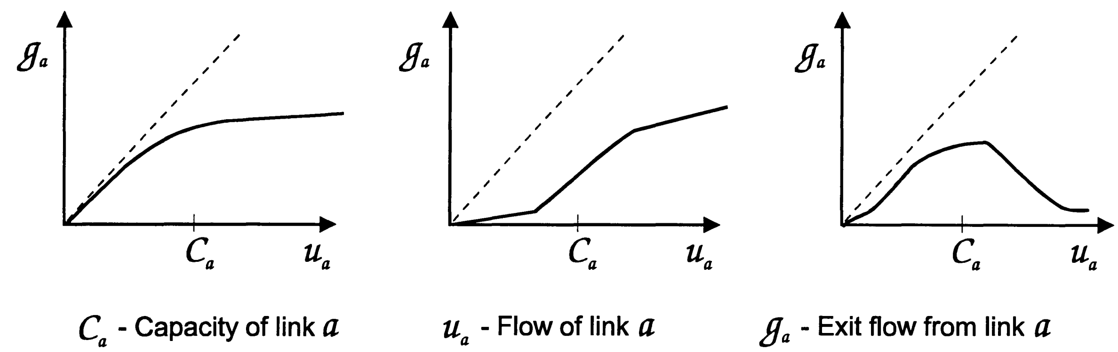

Figure 2. Examples of Exit Functions

represent the increase in travel cost with congestion. An example of a cost function is $h_{ij}(x) = x$ for all i and j, which gives the total number of vehicle-periods spent on the network.

The objective function of the M-N model is expressed as:

$$\textit{minimize} \qquad \sum_{i=1}^{I} \sum_{j=1}^{a} h_{ij}(x_{ij})$$

This function attempts to minimize the total cost as defined by the function $h_{ij}(x_{ij})$ over all links over all time periods. Optimizing this function is subject to four groups of constraints.

1. The state equations. Denoting the number of vehicles admitted onto link j during the ith period by d_{ij} (the decision variables), and assuming that the external inputs are known for each time period $F_i(q)$ and that the volume admitted onto a link cannot leave that link in the same time interval, the fundamental state equations can be written as:

$$x_{i+1,j} = x_{ij} - g_j(x_{ij}) + d_{ij} \qquad i = 0,1,\ldots, I-1 \tag{1}$$

This equation states that the number of vehicles on link j during time interval $(i + 1)$ is equal to the number of vehicles that were on that link during the previous time interval (x_{ij}) minus the volume that exited from the link, $g_j(x_{ij})$, plus the volume admitted on that link, d_{ij}.

2. The flow conservation constraints. The flow conservation equations at each node are given as:

$$\sum_{j \in A(q)} d_{ij} = F_i(q) + \sum_{j \in B(q)} g_j(x_{ij}) \qquad i = 0, \ldots, I-1 \tag{2}$$

where $A(q)$ is the set of links pointing out of node q, $B(q)$ is the set of links pointing into node q, and $F_i(q)$ is the external travel demand (number of vehicles) at node q. These equations reflect the fact that nodes cannot "store" vehicles. Therefore, the number of vehicles leaving a node during an interval must equal the number of vehicles entering the node.

3. The initial conditions. These are given as:

$$x_{0j} = R_j \geq 0 \tag{3}$$

which define the number of vehicles that were initially on each link, j, of the network.

4. The non-negativity constraints. These are expressed as:

$$d_{ij} \geq 0 \tag{4}$$

$$x_{ij} \geq 0 \tag{5}$$

These constraints require that the decision variables, d_{ij}, and the state variables, x_{ij}, are nonnegative. A negative value is physically impossible for the number of vehicles admitted, or present, on a link.

The M-N model formulation represents a discrete time, nonlinear, and nonconvex mathematical programming problem and, therefore, cannot be directly solved using NLP techniques. To allow for the application of such techniques, Carey[3] modified the model by introducing traffic flow control or congestion control constraint that can be used to keep the actual outflow from a link below the natural or the unrestricted capacity level given by the function $g_j(x_{ij})$. In the real world, this corresponds to having a traffic signal or ramp metering system regulating traffic entry into the different segments. Although such a modification allows for solving the model using traditional NLP techniques, there could be many instances in practice where one does not wish to consider traffic controls (e.g., an unmetered freeway). Carey also extended the formulation to handle multiple destinations.[16] Such an extension, however, renders the problem nonconvex once again and hence precludes its solution using traditional NLP methods.

Limitations

The analytical DTA approach places less weight on the evaluation than on the search. Traffic flow is represented by a set of mathematical equations and inequalities. In addition, various assumptions are typically made in formulating these equations to facilitate model solution using traditional NLP techniques. Given the "constraints" placed on the evaluation component, the analytical approach cannot fully capture traffic dynamics. Among the problems that have been reported by researchers in this regard are:

- *The violation of the first in, first out (FIFO) property, which means that the solution may involve holding traffic on one path in favor of traffic on other* *paths for a significant time.* This is unrealistic from an operations point of view.

- *The inability of some models to capture traffic spillback and lane blockage effects.*

- *The inability to capture dynamic traffic flow phenomena such as queue formation and discharge and congestion buildup and dissipation.*

On the other hand, the approach emphasizes the search aspect by attempting to locate the optimal solution. For a complex urban freeway network, finding the optimal solution is computationally intensive. In most cases, such complexity renders DTA models inappropriate for real-time applications.

Combined Simulation/Assignment Approach

Overview

This approach puts more weight on the evaluation than on the search aspect of the problem. Two of the best known examples of this approach are the INTEGRATION model developed by Van Aerde[7,8] and the DYNASMART model developed by Mahmassani et al.[9,10] These models are typically mesoscopic in nature, which means that vehicles are treated as separate entities carrying a set of attributes for assignment purposes but they travel from the entrance to the exit of a section based on speed-density-capacity relationships and not according to car-following logic as in microscopic simulation.

The combined simulation/assignment approach represents an attempt to overcome the limitations of the purely analytical approach in modeling traffic dynamics. In this approach, a simulation model (the evaluation component) is used to model traffic flow (simulation allows for more accurate modeling of traffic flow phenomena such as queue formation and dissipation). An assignment procedure (the search procedure) then assigns vehicles to the shortest path based on the travel time obtained from the simulation model. To allow for capturing the dynamic nature of the problem, the shortest paths are recalculated frequently (every 5 seconds in case of the INTEGRATION model).

Limitations

Simulation-assignment models are rather slow and, therefore, are not suited for real-time applications. Moreover, it is not envisioned that computer advancements in the foreseeable future will completely address this problem.

These models are mainly intended for solving user-equilibrium model formulations and, therefore, cannot be easily adapted to address system optimal assignments.

Despite the fact that they represent an improvement over analytical models in capturing traffic dynamics, they still have problems fully capturing true traffic dynamics because of their mesoscopic nature.

The Roadway System

The first step in developing the routing decision support tools was to define the highway network to be considered for modeling. This network had to include the major facilities in the area that were to be managed by the Suffolk TMS. The selected network is shown in Figure 3.

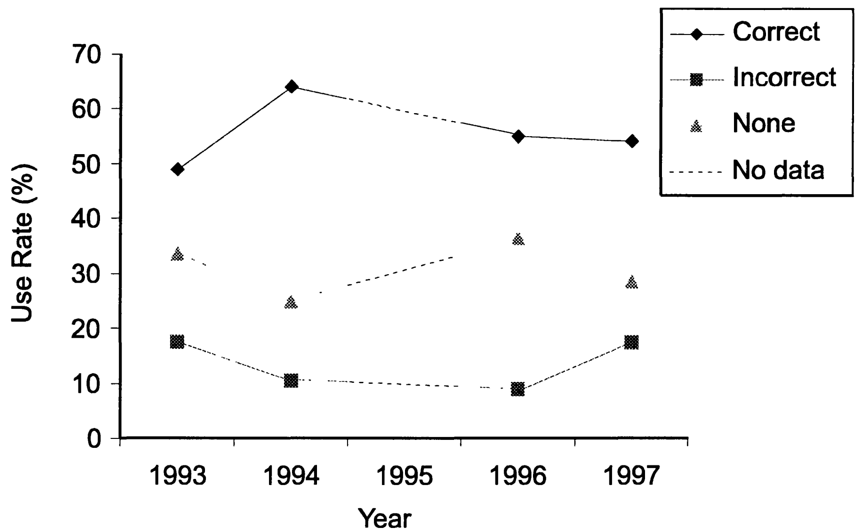

Figure 3. Network Selected for Modeling

As can be seen, the scope of the network selected is composed of the loop formed by I-64 and I-664, along with I-264 and I-464. The network selected, given the location of the VMSs of the Suffolk TMS, will allow for routing traffic originating from Route 44, I-464, and Route 17 as it enters the loop.

Given the scope of the network, the next step was to identify the location of those access/exit points where traffic volume changes significantly. This required a careful study of the variation in traffic volumes along the different road segments of the network. To do this, we used the travel demand model developed by COMSIS Corporation for the HRCS. This model was developed using the MINUTP travel demand forecasting software (version 96A). Validation checks were made to ensure that the model was replicating observed travel patterns. For most of the screen- and cut-lines considered, the deviation of assigned volumes from the observed counts was less than 15 percent.

From the results of the MINUTP traffic assignment step, we estimated the daily volume on each link of the system. We then used these volumes to identify the significant access/exit points. Tables 1 through 7 give the significant access/exit points selected along with the 24-hour traffic volume entering and leaving the system at each point.

Figure 4 shows the location of these points along the network selected for modeling.

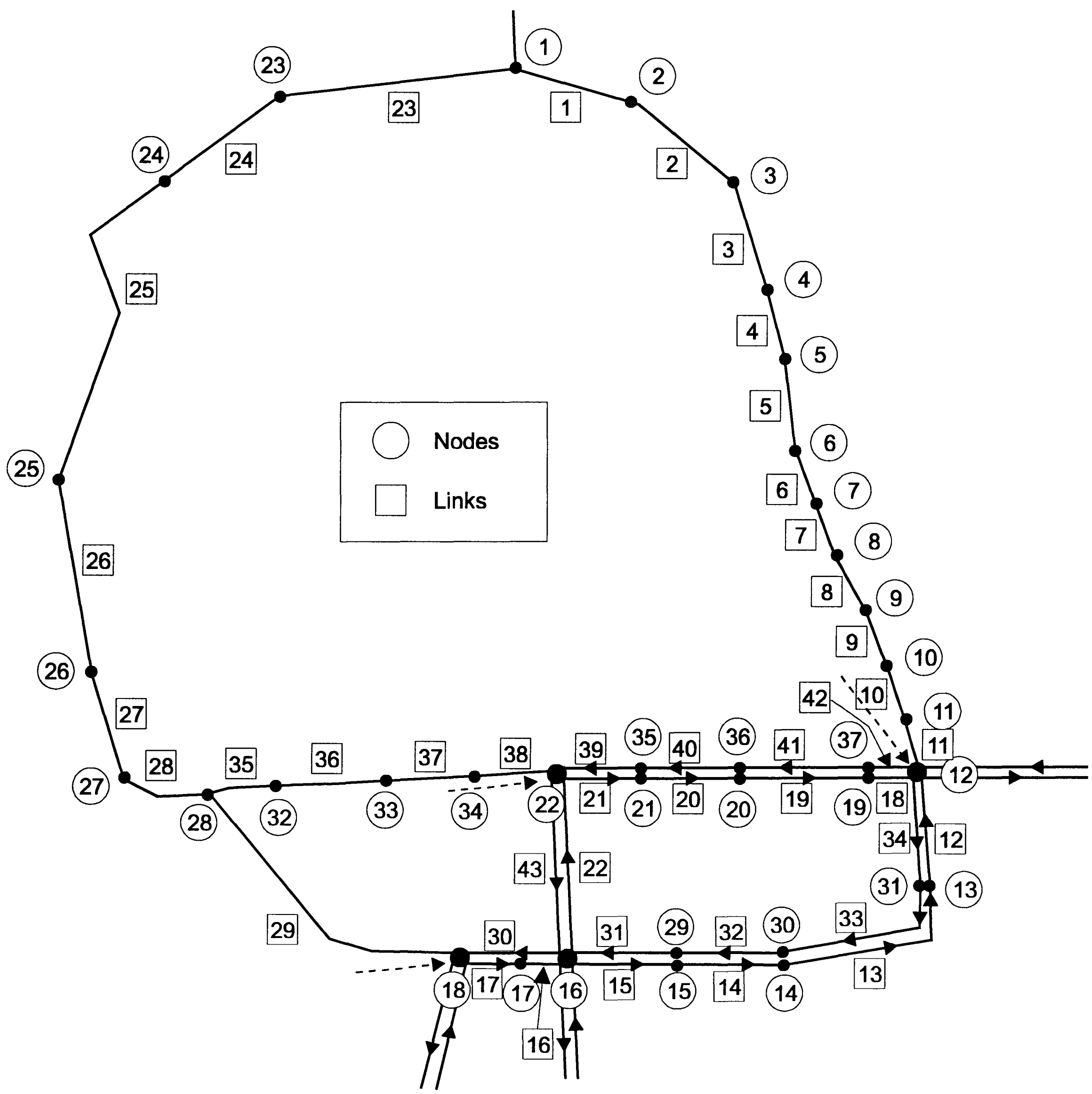

Figure 4. Location of Access/Exit Points with Significant Volume Changes

Table 1. Significant Access/Exit Points Along Segment of I-64 from I-64/Rt. 44 Junction to I-64/I-664 Junction

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Exit 284 - I-64/Rt. 44 Junction	12	103080	47800
Exit 282 - NorthHampton (Rt. 13)	11	24047	8567
Exit 281 - Military Hwy. (Rt. 165)	10	7587	15473
Exit 279 - Norview Ave.	9	13153	8513
Exit 278 - Chesapeake Blvd. (Rt. 194)	8	3093	9540
Exit 277 - Tidewater Dr. (Rt. 168)	7	15833	2787
Exit 276 - I564	6	20020	37073
Exit 274 - Bay Ave.	5	3260	13333
Exit 273 - 4th View	4	3187	7420
Exit 268 - Mallory St. (Rt. 169)	3	12673	860
Exit 265 - Lasalle Ave.	2	14793	5100
Exit 264 - I-64/I-664 Junction	1	0	93007

Table 2. Significant Access/Exit Points Along Segment of I-64 from I-64/Rt. 44 Junction to I-64/I-664 Junction (WB)

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Exit 286 - Indian River	31	9567	12547
Exit 289 - Greenbrier	30	7000	3233
Exit 290 - Battlefield Blvd.	29	6580	4920
Exit 291 - I-464	16	22740	15567
Exit 296 - Rt. 17	18	5660	5913

Table 3. Significant Access/Exit Points Along Segment of I-64 from I-64/Rt. 44 Junction to I-64/I-664 Junction (EB)

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Exit 286 - Indian River	13	25620	1100
Exit 289 - Greenbrier	14	5260	3613
Exit 290 - Battlefield Blvd.	15	12140	9127
Exit 291 - I-464	16	22740	15567
Exit 296 - Rt. 17	18	5660	5913

Table 4. Significant Access/Exit Points Along I-664

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Exit 13 - Military Hwy. (Rt. 13, 58, 460)	27	5453	18867
Exit 12	26	500	2247
Exit 10 - Taylor St/Exit 9 - I-164/Exit 8 - Rt. 135	25	5407	6147
Exit 5	24	10027	387
Intersection w/ Roanoke Ave.	23	7313	467

Table 5. Significant Access/Exit Points Along I-264 (WB)

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Military Hwy. (Rt. 13)	37	8047	20600
Balientine Blvd.	36	10473	3807
Brambleton Ave.	35	0	18127
I-264/I-464 Junction	22	35793	25307
Intersection w/ Rt. 17	34	1767	22293
Portsmouth (Rt. 337)	33	893	11853
Victory Blvd.	32	567	6440

Table 6. Significant Access/Exit Points Along I-264 (EB)

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Military Hwy. (Rt. 13)	19	26400	3027
Balientine Blvd.	20	4300	3107
Brambleton Ave.	21	26973	913
I-264/I-464 Junction	22	35793	25307

Table 7. Significant Access/Exit Points Along I-464

Node Description	VTRC Model Node Number (WB)	Entering Volume (veh./day)	Exiting Volume (veh./day)
I-264/I-464 Junction	22	35793	25307
Exit 291 - I-464	16	22740	15567

Data Collection

The data required for developing the routing strategy decision support tools can be divided into three main categories: (1) freeway geometrics, (2) traffic volumes, and (3) O-D estimates.

Freeway Geometrics

We extracted geometric data from construction plans for the selected freeway system. Data items collected were:

- length of freeway segments (links)
- number of lanes for each segment
- width of lanes
- length of acceleration and deceleration lanes
- location of lane add/drop.

Traffic Volumes

Estimated average 24-hour volumes on the different links of the network were available from the results of the MINUTP model traffic assignment step. For short-term traffic volumes, data were available only for the two tunnels in the region, since the TMS was not yet operational. We obtained traffic volumes, by lane, in 15-minute increments for 1 year from the two tunnel facilities.

For the winter, Figure 5 shows the variation of traffic volume with the time of the day for weekdays during a typical winter month for the westbound direction of the

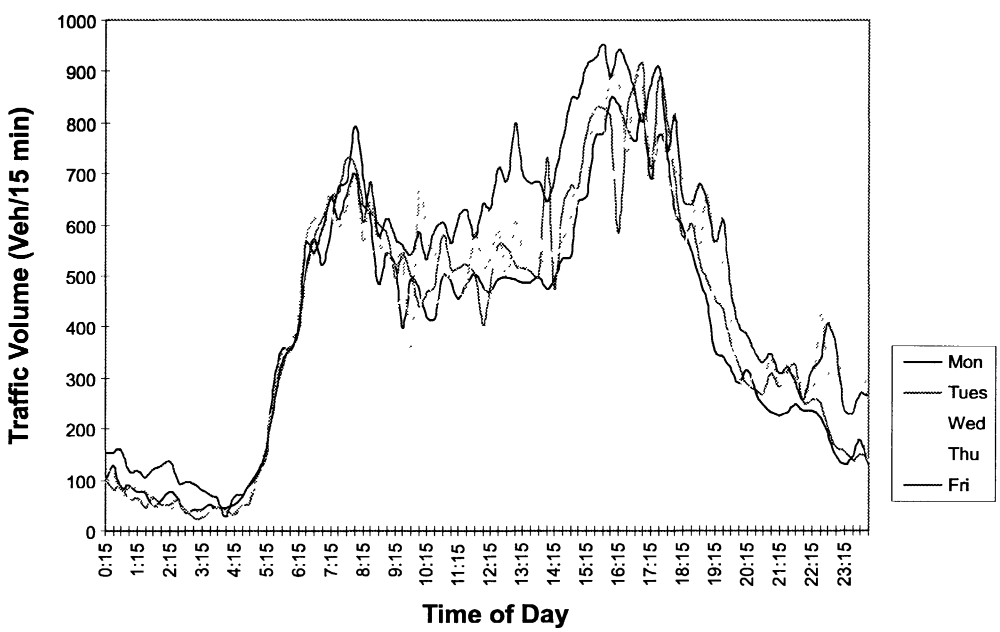

Figure 5. Volume Variation with Time of Day (Winter Weekdays)

HRBT. The evening peak period extends from around 3:30 P.M. to around 6:00 P.M., whereas the morning peak occurs between 6:30 A.M. and 9:00 A.M. In general, the evening peak traffic was heavier than the morning peak. Table 8 gives the peak hour traffic volume for the evening peak period, along with the ratio of the peak hour volume to the daily volume, ***k***. Values for ***k*** were 0.08 to 0.09. For weekends, there appears to be only one extended peak period (Figure 6). Values for ***k***, however, were still around 0.08.

Table 8. Peak Hour Traffic Volume on Weekdays During Winter

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Peak hour	**15:45-16:45**	**16:45-17:45**	**16:15-17:15**	**15:30-16:30**	**15:30-16:30**
Volume (veh./hr)	3251	3398	3236	3407	3709
Daily volume	37238	39431	37951	41811	46293
The ratio, *k*	0.087	0.086	0.085	0.081	0.080

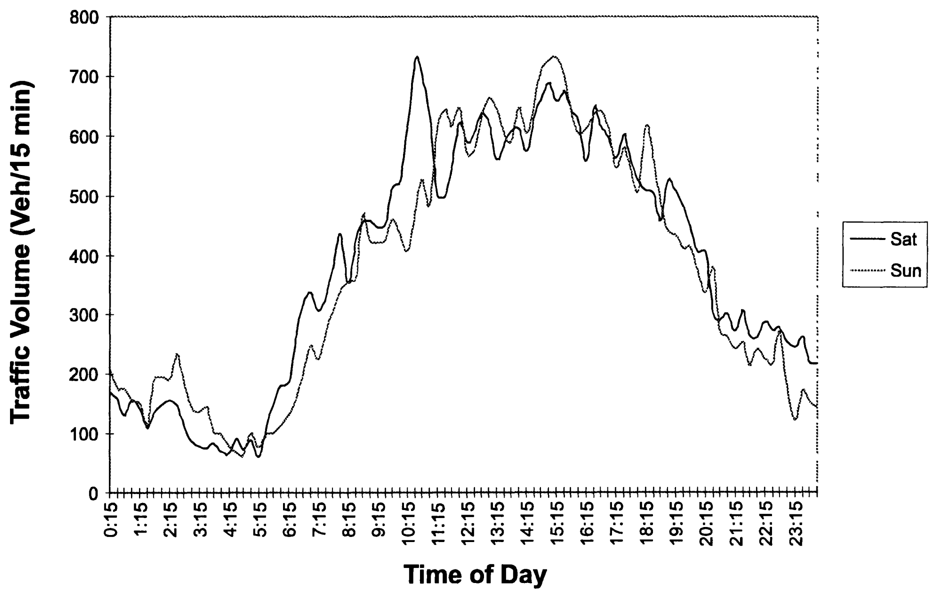

Figure 6. Volume Variation with Time of Day (Winter Weekends)

For the summer, Figure 7 depicts volume variation on weekdays during a typical summer month. The morning and evening peak periods are quite discernible for all weekdays except Friday. For Friday, the peak period is longer. Similar to the winter months, the evening peak traffic is heavier than the morning peak. For weekends, the single extended peak period can once again be discerned (Figure 8). As seen in Table 9, although the daily summer volumes are larger than the winter volumes, peak hour volumes are almost the same. This is obviously a direct result of the increase in recreational traffic during the summer months. The increase in recreational traffic also explains the lower ***k*** ratio values.

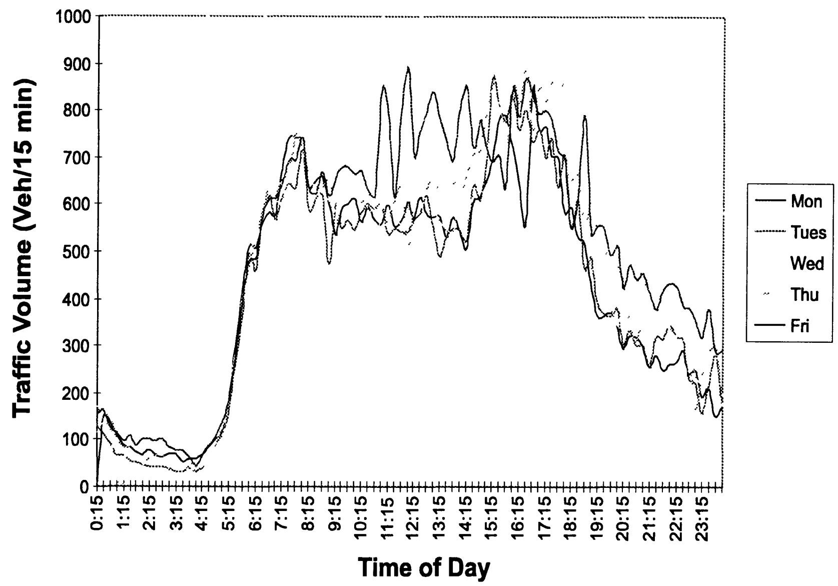

Figure 7. Volume Variation with Time of Day (Summer Weekdays)

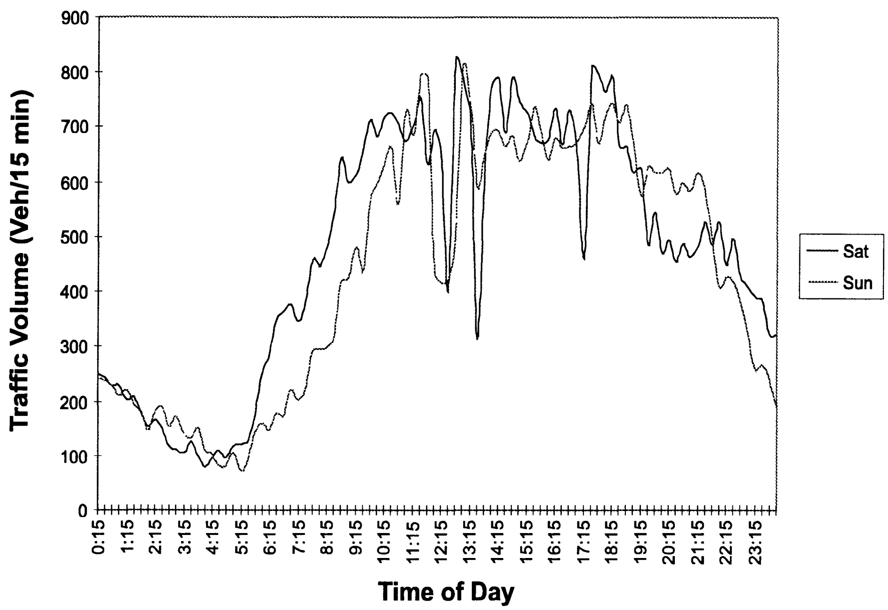

Figure 8. Volume Variation with Time of Day (Summer Weekends)

Table 9. Peak Hour Traffic Volume on Weekdays During Summer

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Peak hour	**16:45-17:45**	**15:15-16:15**	**16:00-17:00**	**15:15-16:15**	**16:00-17:00**
Volume (veh./hr)	3073	3301	3406	3483	3344
Daily volume	40639	40471	43071	45772	48680
The ratio, k	0.076	0.082	0.079	0.076	0.069

Origin-Destination Estimates

The development of effective routing strategies for a network requires a good estimate of the origins and destinations of the trips using the network. The HRCS MINUTP model was a valuable resource in this regard. However, the model, with nearly 1,450 zones and more than 12,000 links, covered a larger area than the network selected for modeling. There was thus a need to extract the O-D matrix for the network of interest (such a matrix is typically referred to as a freeway interchange matrix since it gives the distribution of trips between the on- and off-ramps of the freeway network). Table 10 lists the network's major generators and attraction zones.

Table 10. Network Major Generators and Attraction Zones

Node Description	VTRC Model Node Number	Entering Volume (veh./day)	Exiting Volume (veh./day)
Exit 264 - I-64/I-664 Junction	1	0	93007
Exit 276 - I-64/I564 Junction (Naval Base)	6	20020	37073
Exit 284 - I-64/Rt. 44 Junction	12	103080	47800
Exit 291 - I-64/I-464 Junction	16	22740	15567
I-264/I-464 Junction	22	35793	25307
I-264/Rt. 13 Intersection	19/37	34447	23627
I-264/Brambleton Ave. Intersection	21/35	26973	19040

However, although the compiled matrix gave the daily trips between each O-D pair, real-time routing required O-D matrices for shorter time intervals (typically 15-minute intervals). To address this, the matrix was appropriately scaled based on the short-term traffic counts obtained from the tunnels. Scaling an O-D matrix based on just two points is admittedly not desirable. Nevertheless, the availability of traffic data from the Suffolk TMS, once it is on-line, will provide for more precise estimates in the future.

Search Routines for Determining Effective Routing Strategies

The purpose of this task was to investigate different approaches to developing the search component of the routing strategy development methodology. The search routine is responsible for guiding the exploration of possible routing strategies in search of an effective strategy.

Overview of Search Techniques

Search techniques can be broadly classified into three groups: (1) exhaustive enumeration, (2) mathematical programming techniques, and (3) heuristic approaches. Exhaustive enumeration searches through all possible combinations of values for the decision variables and hence can be practically used only when the search space of the problem is very small.

The mathematical programming group contains a number of different algorithms such as gradient algorithms, separable programming techniques, and sequential-approximation algorithms. These techniques generally require the form of the mathematical program (i.e., the objective function and the constraints) to satisfy certain requirements. They are most appropriate when the program is convex, which means that any local optimum is also a global optimum. However, they face a real challenge when the problem has a number of optima that are different from the global one (a nonconvex problem). Figure 9 illustrates this fact. It shows a very simple function of just one variable that has a number of local optima. Traditional mathematical programming techniques can easily get trapped in a local optimum solution.

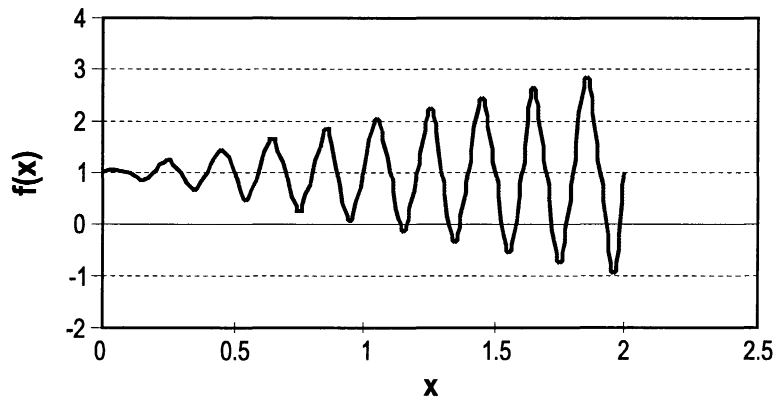

Figure 9. Function with a Number of Local Optima

Heuristic search algorithms have the advantage of not imposing any special requirements on the form of the objective function or the constraints, since they are capable of escaping out of local optima. These algorithms explore only the promising parts of a problem's search space and, hence, are much more efficient than exhaustive enumeration. Heuristic search algorithms are not guaranteed to find *the* optimal solution in every case. However, experience has shown that they yield near optimal results in most cases.

Nature of DTA Problem

As discussed previously, accurate representation of traffic flow invariably leads to nonconvex DTA models. In addition, the real-time nature of the problem precludes the use of exhaustive enumeration except for very simple networks. Consequently, the study focused on the use of heuristic search algorithms. As mentioned previously, we investigated two recently developed search algorithms: GAs, which are based on the principle of survival of the fittest, and SA, which is analogous to the process of atoms rearranging themselves in a cooling metal. GAs and SA can deal with any functional form of the objective function and the constraints. Moreover, they can strike a balance between the desire to explore the whole solution space of a particular problem and the need to focus on the most promising parts of this space. Therefore, they are well suited to solve combinatorial optimization problems such as the one in this study.

Genetic Algorithms

GAs are stochastic algorithms whose search methods are based on the principle of survival of the fittest. They use a vocabulary borrowed from natural genetics. In genetics, one speaks about individuals (sometimes called strings, or chromosomes) in a population. Chromosomes are made of genes arranged in linear succession. The basic procedure of the GA is straightforward (Figure 10). During each iteration, *t,* the procedure maintains a population of individuals, *P(t).* Each individual or chromosome represents a potential solution to the problem under consideration. The procedure starts with a randomly generated initial population of chromosomes (a set of potential solutions). Each solution, x_i^t , is evaluated to give some measure of its "fitness" (the evaluate step). Then, a new population (iteration $t + 1$) is formed by selecting the more fit individuals (the select step). Some members of this new population undergo alterations by means of genetic operations (typically referred to as crossover and mutation operations) to form new solutions (the alter step) while keeping the size of the population constant. After some number of generations (iterations of the select, alter, and evaluate steps), it is expected that the algorithm will "converge" to a near-optimum solution.[12]

Simulated Annealing

SA is a technique for finding a minimum or a near-minimum in a function of many variables proposed by Kirkpatrick et al. in 1984.[13] In recent years, the technique has been successfully applied to a large number of problems arising in computer design and other fields. SA is based on an analogy to the process of atoms rearranging themselves in a cooling metal. For a metal to be frozen into a near perfect crystal lattice (lowest energy state), it must be annealed by first melting and then cooling very slowly. If cooling is done in the correct fashion, atoms will eventually form a neat perfect crystal lattice even though they may have to pass through locally disordered states to do so. In function optimization problems, we face the same problem. We are searching for the global optimum (the lowest energy state), but we may get caught in local optima (locally disordered states).

In SA, a trial solution is chosen and the effects of taking a small random step (move) from this position are tested. If the test reveals that the step has changed the value of the objective function in the direction of the desired long-term trend, the move is immediately accepted. However, if the step changes the value of the objective function in the opposite direction, the move is accepted or rejected based on a probability related to an "annealing temperature." That is to say, moves that change the value of the objective function in the direction opposite to that of the desired long-term trend still have a chance of being accepted. In a minimization problem, for example, a move that increases the

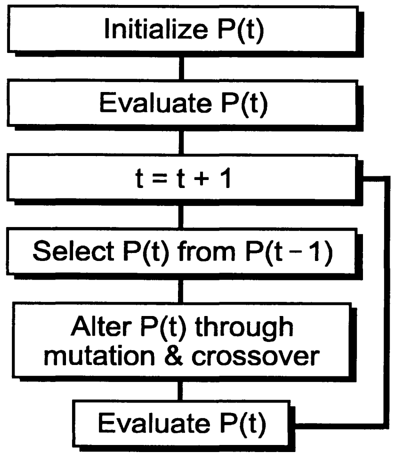

Figure 10. GA Procedure

objective function value (an uphill move) may be accepted as part of the full series of the downhill moves for which the general trend is to decrease the value of the objective function. It is argued that such controlled uphill steps allow one to break away from configurations leading to locally optimal solutions, and hence increases the likelihood of eventually obtaining a higher quality solution.

The probability, *P,* of accepting these uphill moves is given by the following equation, borrowed from the annealing process:

$$P = exp\,(-\Delta E/kT) \qquad (6)$$

where k is Boltzmann's constant, T is the temperature, and ΔE is the change in energy. For optimization problems, the energy, E, of equation (6) corresponds to the value of the objective function, and since the temperature is a numerical value that controls the probability of accepting uphill moves, the Boltzmann constant is not needed in the computation. The SA algorithm starts by initially setting the temperature at a high value, and then it periodically decrements such a value. While T is high, the optimization routine is free to accept many varied solutions, but as it drops, this freedom diminishes until the search is over. The success of the SA technique is, therefore, heavily dependent on the selection of a proper annealing schedule. An *annealing schedule* is the sequence of temperatures and the amount of time or number of iterations at each temperature needed to reach equilibrium at that temperature.

Evaluation Tools for Testing Effectiveness of Alternate Routing Strategies

The objective of this task was to explore different approaches for modeling traffic flow in the region that can be used in evaluating the effectiveness of the alternate routing strategies generated by the search algorithm. We developed two models that vary in complexity and input requirements: (1) a shock-wave model developed for a very simple version of the highway network; and (2) a detailed dynamic, macroscopic, deterministic model of the region.

Shock-Wave Model

The purpose here was to develop an evaluation tool that (1) was simple to use, (2) required input data that were readily available, and (3) was capable of real-time execution. The tool was developed for a very simple network consisting of just two routes and for one specific routing scenario that frequently faces traffic operators at Suffolk TMS (Figure 11). This scenario involves routing westbound traffic originating from Route 44 with destinations in Newport News when an incident occurs on the Hampton-Roads Tunnel segment. The tool would help the user determine the percentage of traffic that needs to be diverted to the MMBT and would give an estimate of the expected time required for flow to return to normal.

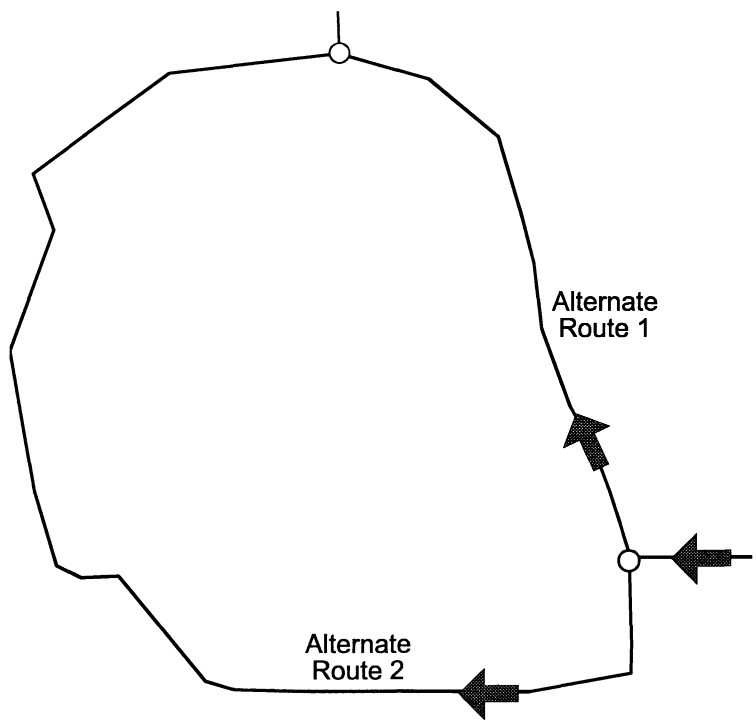

Figure 11. Network Considered for Shock-Wave Model

The tool is based on shock wave analysis and macroscopic traffic flow theory principles. To allow for developing a simple tool with the minimum input requirements, a number of simplifying assumptions had to be made (these assumptions should be expected to affect the accuracy of the results).

Assumptions

We made the following assumptions while developing the tool:

- Entering and exiting volumes between the origin (Route 44/ I-64 interchange) and the destination (I-64/I-664 junction) are ignored.
- Traffic volumes are assumed constant over the planning horizon considered.
- The recommended diversion percentage remains the same until flow returns to normal (i.e., remains constant throughout the planning horizon).
- The expected duration of an incident can be estimated.
- Traffic volumes using I-264 and I-464 are not treated as a part of the system modeled.
- A Greenshield's model[17] for capturing traffic flow dynamics is assumed with the following parameters: a jam density of 83.9 vehicles/km/lane (135 vehicles/mi/lane) and a free-flow speed of 112.6 km/hr (70 mph).

Shock-Wave Theory

The use of shock-wave analysis to model traffic congestion was first introduced by Lighthill and Whitham.[18] Shock waves are defined as boundary conditions in the time-space domain that mark a discontinuity in flow-density conditions. One may consider the example of a pretimed signal-controlled intersection. At some distance upstream of the signal and immediately downstream of the signal, free-flow conditions exist. However, just upstream of the signal during the red phase, vehicles will be stopped and densities will be high. As a result, there will be a discontinuity as vehicles join the rear of the queue (backward forming shock wave) and as vehicles are discharged from the front of the standing queue (backward recovery shock wave) when the signal turns green (see Figure 12). These two shock waves are backward moving because over time the discontinuity is propagating in the opposite direction of the moving traffic. The first shock wave is a forming wave because it is causing an increase in the congested part, and the second is a recovery wave because it is causing a decrease in the congested area.[17]

The shock wave speed between two traffic states is equal to the change in flow divided by the change in density. Shock waves can be analyzed if a flow density relationship is known and traffic flow states are specified.

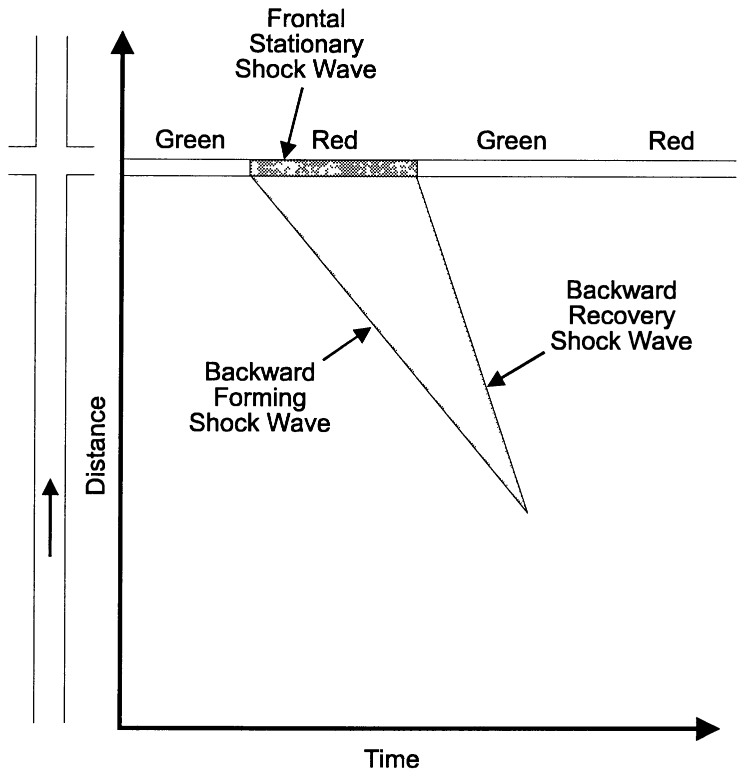

Figure 12. Shock Waves at Signalized Intersection

Hampton Roads Shock-Wave Evaluation Tool

The goal of the shock-wave evaluation tool is to find the travel time on the HRBT segment (alternate route 1) and the MMBT segment (alternate route 2) under different diversion percentages. The travel time is averaged over the time period spanning from the moment an incident is verified to the time needed for traffic flow to return to normal conditions.

For calculating the travel time on the HRBT segment where an incident is assumed to have occurred, a shock-wave analysis is conducted. The shock-wave diagram for this case is very similar to the one occurring at a signalized intersection (Figure 12). To find the average travel time over the period from the moment the incident is verified to the time flow returns to normal, the model traces the trajectories of representative vehicles that are 5 minutes apart, determines the travel time of each vehicle, and then averages the results.

For the MMBT segment, calculating the travel does not require shock-wave analysis since no incident conditions are involved. The procedure simply entails using the flow-density-speed relationships to determine the speed corresponding to the traffic volume on the segment, and hence the travel time.

Detailed Dynamic, Macroscopic Model

In this task, the purpose was to develop a more detailed dynamic, macroscopic mathematical model for the Hampton Roads region that could more accurately capture traffic flow dynamics. This model would then serve as the evaluation component for a routing strategy development methodology. The study team felt that for the model to provide more accurate results, it should have the following features:

- The model should account for the dynamic nature of traffic demand/supply.
- The model should take into account traffic entering and exiting at the various access/exit points to the network.
- The model should be able to capture spillback and lane blockage effects, as well as the effects of lane add/drop.
- The model should allow for considering multiple O-D pairs.
- The model should allow for considering more than one routing scenario.

The model has its roots in the modeling framework proposed by Papageorgiou.[6] We introduced a number of refinements to allow for more accurate representation of traffic dynamics.

Papageorgiou's Modeling Framework

Papageorgiou's model addresses the general case of a multi-origin, multi-destination network. The model is based on the concept of independent splitting rates at each ***node,*** $\beta_{nj}^{m}(k)$, which give the rate of traffic volume leaving node n and destined to node j that uses link m during interval k ($k = 0,1,2,\ldots$ is the discrete time index (i.e., $\beta(k) = \beta(k \,.\, T)$), T being the sample time interval or time step for the dynamic model).

The $\beta_{nj}^{m}(k)$s thus define how traffic is distributed among the alternate routes at a node and are given by:

$$\beta_{nj}^{m}(k) = q_{nj}^{m}(k)/q_{nj}(k) \qquad (7)$$

where $q_{nj}(k)$ is the traffic volume exiting node n and destined to node j, and $q_{nj}^{m}(k)$ is the volume exiting node n through link m and destined to node j during interval k. It follows therefore that the $\beta_{nj}^{m}(k)$ s assume values between 0 and 1.0 and that

$$\sum_{m} \beta_{nj}^{m}(k) = 1.0 \qquad (8)$$

Papageorgiou's model uses exit functions, first introduced by Merchant and Nemhauser,[1] which give the number of vehicles leaving a link as a function of the number of vehicles on that link. The state of the system is described in terms of the traffic density along the different links during each time step.

The formulation of the state equations of a DTA model (equation 1) dictates that the sample time interval or time step, *T*, be chosen so that the maximum distance a vehicle travels in one time period is less than the link length. This ensures that all vehicles entering a link during a given interval remain on that link during that time interval and hence are justifiably included in estimating the density on that link.

Given (1) an initial state, (2) a demand matrix (O-D matrix), and (3) a specified set of the splitting rates, $\beta(k)$, the model can be used to describe the dynamic evolution of the system. Therefore, the model can be used to evaluate the effectiveness of a particular routing strategy, as defined by the $\beta(k)$, given the initial state and the demand matrix.

Refinements to Papageorgiou's Framework

With Papageorgiou's framework as a starting point, we developed a dynamic model for the Hampton Roads network with the following refinements:

1. *The model was designed to check for the capacities downstream and to admit only such a volume that would not result in exceeding the downstream capacity.* Any excess volume is not allowed to exit and thus remains on the link till downstream capacity becomes available. Such a modification allowed the model to capture spillback and congestion buildup effects more closely.

2. *The model was modified to allow for different splitting rates on the different approaches to a node.* For example, for the node shown in Figure 13, one may have three different sets of the β_{nj}^{m} for that node, one set for each of the approaches A, B and C, instead of just one set for the node as a whole.

3. *The model was equipped with the capability to capture the effect of lane add/drop.* For example, in the case where a three-lane section leads into a two-lane section, the model was designed to ensure that the volume exiting from the three-lane segment does not exceed the minimum headway requirements for the two-lane segment capacity. This allows the model to approximate the effect of the shock wave occurring at such sections. This refinement could be viewed as a special case of refinement 1.

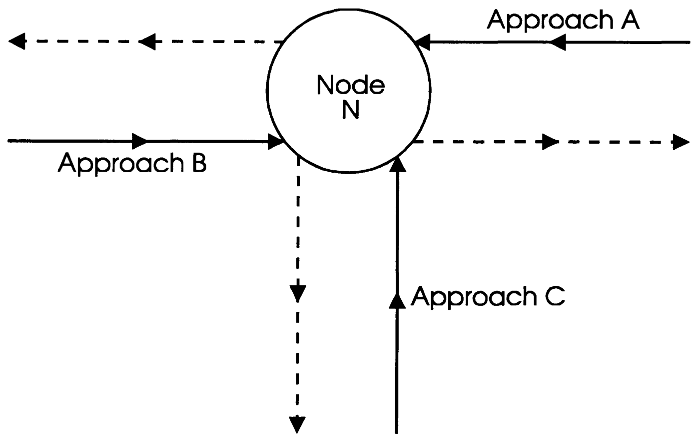

Figure 13. Splitting Rates at a Node

Hampton Roads Model

For the Hampton Roads model, the exit function was based on the modified Greenshields formula proposed by Chang et al.[19] To satisfy the requirement that the maximum distance a vehicle travels during a time interval should be less than the link's length, T had to be less than 50 seconds. The evaluation criterion selected to measure the system's operational efficiency under a particular routing strategy was the sum of the vehicles left on the network during the final time interval and those vehicles that were not able to depart from an origin node because all the links leaving that node were saturated. As pointed out by Merchant and Nemhauser,[1] attempting to ***minimize*** this sum serves the purpose of moving vehicles to their destinations as fast as possible and hence has the effect of reducing the total travel time.

The model was coded in C++. For a specific routing strategy, the program required less than 0.50 second to simulate a clock-time period of 20 minutes on a Pentium 166 MHz computer. However, before the model can be implemented in the real world, its exit functions must be calibrated using real-time traffic data. Calibration of an exit function essentially entails determining the function's parameter values that will allow the output of the function to resemble real-world conditions as closely as possible. The calibration process will be performed once the Suffolk TMS traffic data become available.

Prototype Routing Decision Support Systems

In this task, the search and evaluation routines developed under tasks 4 and 5 were combined to develop two routing decision support prototypes: (1) a simple shock-wave DSS prototype, and (2) a heuristic search/DTA DSS prototype. A preliminary evaluation of the two prototypes was then conducted to check the plausibility of their recommended strategies and their potential for real-time applications.

Shock-Wave DSS Prototype

The purpose of the shock-wave DSS is to find the diversion percentage that will equate or minimize the difference between the average travel time on the HRBT segment and the MMBT segment. As previously mentioned, our problem formulation had only one decision or control variable (i.e., the split rate at the Rt. 44/I-64 interchange). Moreover, this diversion percentage remained the same from the time an incident was verified to the time flow returned to normal (i.e., constant splitting rate throughout the planning horizon). This allowed for the use of exhaustive enumeration, since the size of the search space was quite small. The study team used a search routine that simply tried diversion percentages ranging from 1% to 100% in increments of 1%. For each diversion percentage, we ran the shock wave model to determine the travel time on the two routes of the network. We then selected the diversion percentage yielding the smallest difference between the travel time on the two routes.

The shock-wave DSS requires less than 0.50 second of CPU time on a Pentium 166-MHz PC and hence is quite capable of real-time execution. Table 11 provides the reader a flavor of the results. The table lists the diversion percentage recommended by the DSS for the case of (1) an incident with a duration of 20 minutes, (2) an initial queue length of 3.2 km (2.0 mi) at the time the incident was verified, (3) a traffic volume of 3,400 vehicles per hour on the HRBT segment, and (4) three traffic volume levels on the MMBT segment. The table also shows the travel time on the HRBT and the MMBT for both the case of diversion and the case of no/diversion. Moreover, the table gives the time savings for the number of vehicles entering the system during a 15-minute period that would result if the recommended diversion strategy was implemented.

Table 11. Shock-Wave DSS Results

			Diversion		No Diversion		
Case	**MMBT Vol. (veh/hr)**	**Diversion %**	**HRBT travel time (min)**	**MMBT travel time (min)**	**HRBT travel time (min)**	**MMBT travel time (min)**	**Time savings (veh min/15 min)**
1	2000	8	43.4	43.6	44.3	42.2	165
2	1600	17	42.8	42.9	44.3	40.2	180.6
3	1200	26	42.4	42.4	44.3	38.6	475

The DSS equated the travel time on the HRBT and MMBT for the three cases. The time savings increased with the increase in the difference in volumes between the two segments, which is quite reasonable. The rather small values for the time savings resulting from implementing the diversion strategies are to be expected, since the shock-wave DSS considers only a single O-D pair (e.g., traffic from Route 44/ I-64 interchange to I-64/I-664 Junction).

Heuristic Search/DTA Model DSS Prototype

As opposed to the shock-wave DSS search space, the search space for the dynamic macroscopic model is extremely large. For example, for a network with 40 independent diversion splits per time interval, and for a planning horizon of 25 minutes divided into five intervals of 5 minutes each (i.e., a total of 40 x 5 = 200 independent diversion rates), one would have a total of $\boldsymbol{100^{200}}$ combinations that need to be evaluated (assuming we are considering increments/decrements of 1% for each split rate). Running the macroscopic model $\boldsymbol{100^{200}}$ times would require 1.389×10^{396} hours on the Pentium 166-MHz PC. This is clearly infeasible from a practical standpoint. There is, therefore, a need for adopting a heuristic search approach. In addition, there was a need for attempting to reduce the complexity of the problem before applying the search routine.

Simplifying the DTA Problem

From a theoretical standpoint, the size of the DTA problem solution space is so big as to challenge solution using any search algorithm. Fortunately, however, several practical considerations allow for significantly simplifying the problem. We exploited four in this study:

1. Calculation of splitting rates based on clustered zones. For a routing system that uses VMSs, such as the one considered in the current study, it is not practical to have a distinct splitting rate for each traffic subflow at a node flowing to each destination zone. Given the limited information capacity of a VMS, attempting to do this would require the installation, at each node, of a number of VMSs that is equal to the number of destination zones with an independent splitting rate that is reachable from that node. One may consider, for example, node 12 on Figure 4. The number of destination zones with an independent splitting rate that is reachable from this node is 12 (nodes 1, 16, 18, 22, 23, 24, 25, 26, 27, 32, 33, and 34). Therefore, using a distinct splitting rate for each subflow would require us to install 12 VMSs for node 12 alone. Clearly, this is practically unfeasible.

One idea to overcome this problem is to group zones into clusters for the purpose of calculating the splitting rates. At each node, subflows with destinations belonging to the same cluster are assigned the same splitting rate. This clustering reduces the size of the search space and hence the complexity of the problem.

For the Hampton Roads network, we defined nine clusters based on the size of traffic demand at the zones and their locations with respect to the routing opportunities in the network (Figure 14). Clustering allowed for cutting down the number of the decision variables to 25 independent splitting rates for each time interval.

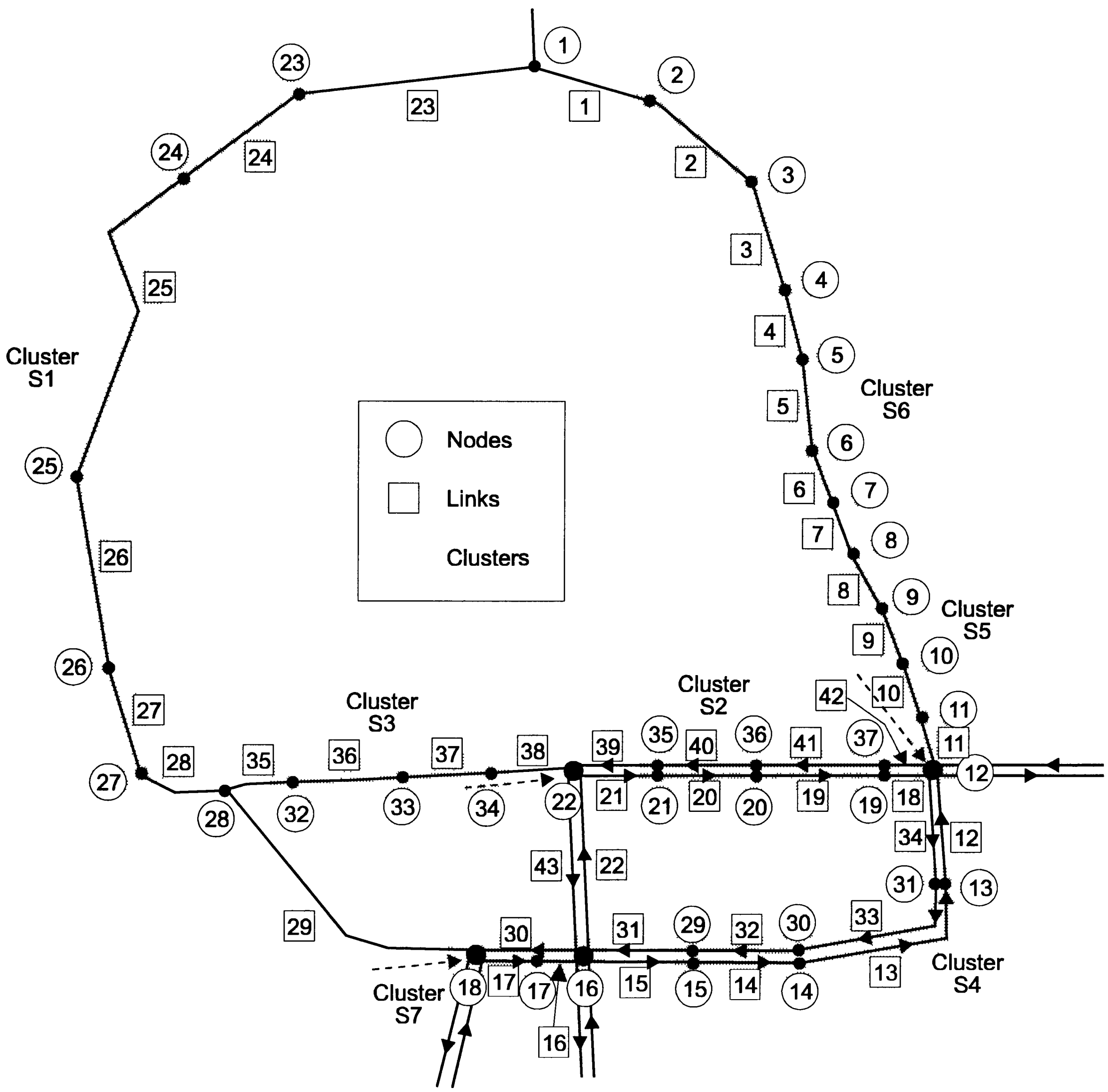

Figure 14. Zone Clusters

2. *Time interval size for updating splitting rates.* As was previously mentioned, the sample time interval, T, for the dynamic traffic model had to be shorter than 50 seconds to satisfy the requirement that traffic cannot enter and leave a link within the same time interval. This, however, does not mean that the splitting rate needs to be changed every 50 seconds. Instead, one could use a longer time interval for updating the traffic split. Longer intervals help reduce the complexity of the problem and are even more appropriate from a practical standpoint since drivers are not in favor of frequent changes. We used an update time interval of 5 minutes. During each of these 5-minute intervals, the splitting rates are kept constant.

3. Rolling horizon approach. As previously discussed, the complexity of the DTA problem increases dramatically with the increase in the number of time intervals considered. To address this difficulty, a rolling horizon approach may be employed where routing strategies are generated for a reduced prediction horizon with a small number of steps. The strategy determined would then be implemented, the projection horizon rolled, and the cycle repeated (Figure 15).

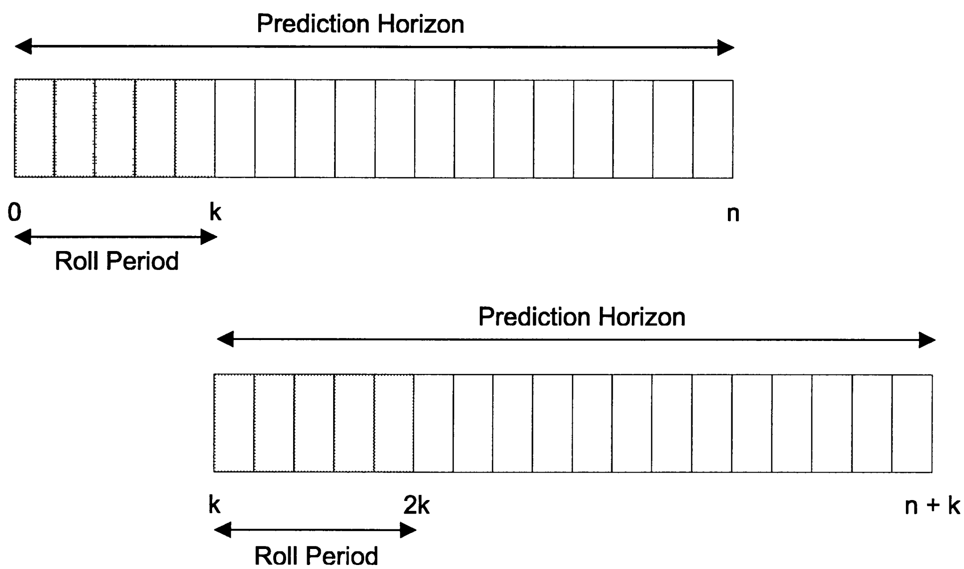

Figure 15. Rolling Horizon Approach

In addition to simplifying the problem, a rolling horizon approach is very well suited for the real-time DTA problem. This approach helps reduce the size of the prediction horizon and hence improves the reliability of the traffic forecast.[20] For the current study, we used a prediction horizon of 20 minutes, corresponding to four splitting rate update intervals of 5 minutes each.

4. Precision in estimating splitting rates. From a practical standpoint, estimating the splitting rate to a great level of precision is a poor use of computation resources. First, small changes in the diversion percentage are unlikely to have a significant effect on the performance of the network. Second, it is quite unreasonable to assume that one will be able to influence drivers' behavior so as to achieve precisely the recommended diversion percentage.

For the DTA problem, exploiting this fact can be advantageous since it can drastically reduce the size of the search space. For a problem with 100 splitting rates, for example, the solution space size would be in the order of 100^{100} if changes of 1% in the diversion percentages were considered, whereas it would be in the order of only 10^{100} if

10% increments/decrements are assumed. We searched the solution space using 10% increments/decrements for each split percentage.

Implementation Details

GA Program. Developing the program involved the following subtasks:

1. Developing a GA's representation scheme. Solving the DTA or the routing strategy development problem essentially involves determining the diversion percentage or the independent traffic split at each diversion point. As previously mentioned, the network selected for modeling had 25 independent splitting rates (i.e., diversion possibilities) for each time interval. Therefore, for a planning horizon of 20 minutes divided into four intervals of 5 minutes, the problem would involve determining the values for 100 splitting rates (25/interval x 4 intervals). In this case, a potential solution to the problem would be represented as a 100-element vector as follows:

$$\boldsymbol{u} = (u_1, u_2, u_3, u_4, u_5, u_6, \ldots u_{100})$$

where each element, u_i, is a real-valued number corresponding to an independent traffic split or diversion percentage (i.e., a value between 0 and 100).

2. Designing an approach for constraints handling. The basic idea in creating the initial population was first to determine the upper and lower bounds for each control variable and then to select a random number between these bounds for this variable. The lower bound for our split rates is 0, and the upper bound can be determined from the fact that the sum of splitting rates for a particular O-D pair at a node is equal to 1.0.

3. Designing a selection scheme. Evaluating a GA chromosome (i.e., a potential solution for the problem) involved running the dynamic model for the values of the traffic splits encoded in the chromosome and determining the corresponding value of the objective function. The selection scheme used to select the more fit individuals from a population was the roulette wheel procedure commonly used in GA applications.

4. Designing appropriate genetic operators. The mutation operator was designed to proceed in the following fashion. A gene (a variable from the solution vector) is randomly selected and replaced by a random number selected between that gene's bounds. Since this may change the bounds for the genes that follow, a check is made to ensure that such genes are within their new ranges. If any gene is outside its range, a new random number that is within the new bounds replaces it.

The crossover operator was designed to combine the features of two parent chromosomes to form two offspring by swapping corresponding segments of the parents. For example, if the two parents (a_1, b_1, c_1, d_1, e_1, . . .) and (a_2, b_2, c_2, d_2, e_2, . . .) are crossed after the second gene, the offspring (a_1, b_1, c_2, d_2, e_2, . . .) and (a_2, b_2, c_1, d_1,

$e_1, \ldots$) are produced. Similar to the mutation operator, a check is made to ensure that all genes are within admissible bounds.

SA Program. The design of the SA program involved the following subtasks:

1. Developing a representation for the problem's potential solutions. As in the GA program representation, a potential solution to the problem was represented as a 100-element vector as follows:

$\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, \ldots u_{100})$.

2. Designing a method for moving to neighboring points in the solution space (commonly referred to as the neighborhood structure). For moving to neighboring points in the solution space, the following procedure is executed. One element (splitting rate) from the solution vector is selected at random. A random number in the range [0,1] is then generated. If that random number is less than or equal to 0.50, the selected splitting rate is increased by 10%; otherwise, the chosen splitting rate is reduced by 10%. A check is then made to ensure that all the variables are within admissible bounds. If any variable is outside the specified range, its value is reset to that of the boundary.

3. Defining an annealing schedule. We modeled the annealing schedule after that proposed by Golden and Skiscim,[21] with minor modifications. The approach is based on the concept of an epoch that is made up of a prespecified number of accepted moves *(k)*. After an epoch is executed, the resulting solution is saved and testing for equilibrium is performed. This test compares the most recent objective function value with the values from all previous epochs at the same temperature. If the objective value of the most recent solution is close (a threshold value $[0 < \varepsilon < 1]$ is usually defined for this purpose) to any previously observed value from epochs at the same temperature, the system is declared to be at equilibrium and the next temperature is selected. The temperature is reduced by 20% at each step for a predefined number of steps *(x)*.

Preliminary Evaluation

We coded the SA and GA algorithms in C++ and linked them to the detailed dynamic macroscopic model developed for the Hampton Roads network. For the SA algorithm, we selected a value of 10 for the initial temperature control parameter after preliminary experimentation. We set the other two control parameters at the following values:

number of temperature steps *(x)* = 25

number of moves per epoch *(k)* = 25

since these were the values used by Golden and Skiscim.[21] For the GA program, we adopted the following control parameter values:

population size = 30

probability of crossover = 0.40

probability of mutation = 0.15

number of generations = 500.

These values are among the ones most commonly used in many GA implementations.

We considered two routing problems. In the first, an incident was assumed to have taken place on link 11, resulting in a 60% reduction in that link's capacity. In the second, a 75% capacity reduction was assumed. An important consideration in evaluating the performance of stochastic search algorithms such as the ones considered in this study is that they should yield consistent results regardless of their start point. To test this, we tried five runs using a different random number seed for each of the two cases.

SA Results. The SA results are given in Table 12, along with the number of evaluations performed by the program and its running time on a Pentium 166-MHz PC. As can be seen, the solutions from the five runs were very close. For case 1, the range of values was within less than 0.15% of the best value (8,547), whereas for case 2, the range was within less than 0.22%. This shows that the algorithm was yielding consistent results.

Table 12. SA Results

	Problem 1–60% Capacity Reduction			Problem 2–75% Capacity Reduction		
Run	Objective function (veh)	No. evaluations	Running time (min)	Objective function (veh)	No. evaluations	Running time (min)
1	8553	4075	18.8	46065	5512	25.5
2	8549	4775	22.1	46038	5515	25.5
3	8559	4101	18.9	46125	4904	22.7
4	8547	4300	19.9	46139	5491	25.4
5	8554	4726	21.8	46117	5766	26.6

To get a better appreciation of the execution-time characteristics of the algorithm, we plotted the objective function value against the number of evaluations performed by the program for each of the five runs (Figures 16 and 17). The algorithm seems to get close to the final value obtained quite early in the search. For case 1, a value within 5% of the best solution was attained in less than 1,450 evaluations for the five runs performed. For case 2, such a value was attained after 2,250 evaluations. This is quite significant for real-time applications since it means that a quick solution may be obtained

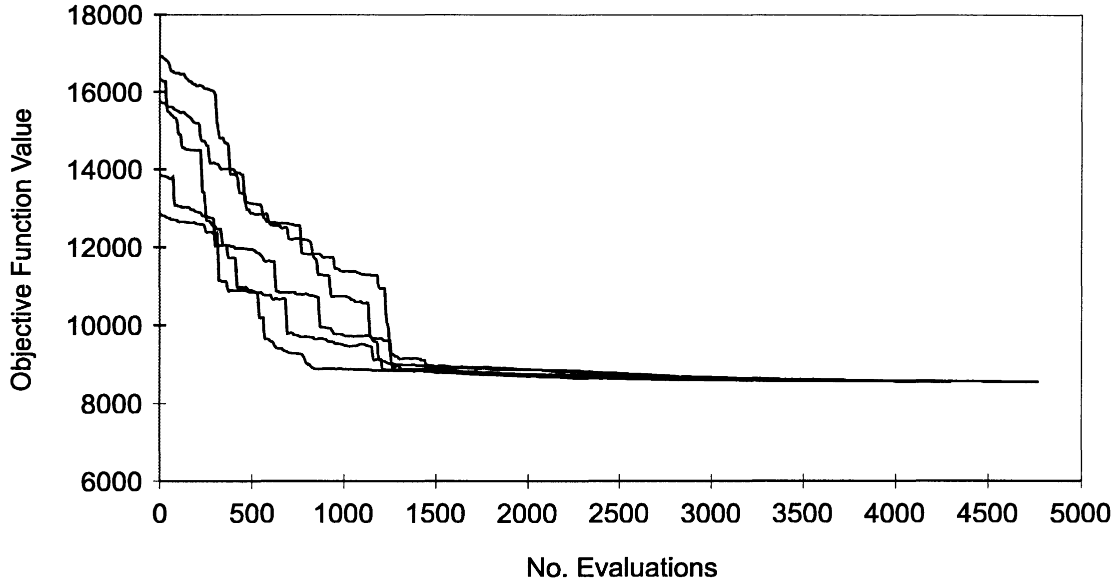

Figure 16. SA Results for Case 1

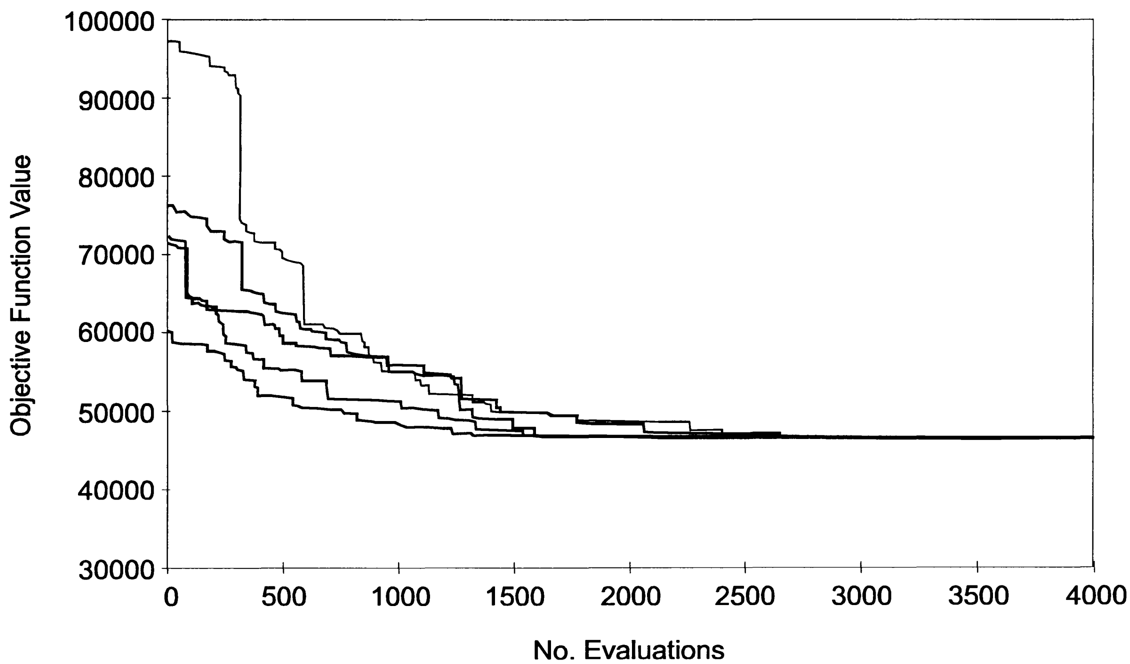

Figure 17. SA Results for Case 2

with a minor sacrifice in accuracy. Running the program for 1,450 and 2,250 evaluations requires less than 6.7 and 10.4 minutes, respectively, on the Pentium 166MHz PC. It is also clear from the figures that the better the quality of the start point, the sooner the SA approaches the final value.

Table 13 gives the total vehicles travel time under the routing and no/routing scenarios. For the no/routing scenario, drivers were assumed to use the routes they typically take to their destinations in the absence of any routing strategy. The travel time given in this table is for the vehicles that were initially on the system at the time the incident was verified, as well as those vehicles entering the system during the following 15-minute interval. As can be seen, time savings was much larger than that given by the shock-wave DSS since the heuristic search/DTA DSS considers travel demand between all the different O-D pairs in the network.

Table 13. Travel Time Under Routing and No/Routing Scenarios

	Travel Time No/Routing (veh. min)	Travel Time Routing Implemented (veh min)	Time Savings (veh min/15 min)
Problem 1 (60% reduction)	415711	369328	46383
Problem 2 (75% reduction)	424818	384840	39978

GA Results. Table 14 gives the GA results. Once again, the results obtained from the five runs were very close. They were also quite close to the SA program results. However, the SA results were slightly better. For case 1, the SA best solution was 1.60% less than the GA best solution, and for case 2, 1.00% less than that of the GA.

Table 14. GA Results

	Problem 1–60% Capacity Reduction			Problem 2–75% Capacity Reduction		
Run	Objective function (veh)	No. evaluations	Running time (min)	Objective function (veh)	No. evaluations	Running time (min)
1	8690	15000	61.5	46633	15000	61.5
2	8703	15000	61.5	46670	15000	61.5
3	8690	15000	61.5	46522	15000	61.5
4	8697	15000	61.5	46900	15000	61.5
5	8687	15000	61.5	46492	15000	61.5

Figure 18 shows the solution obtained as a function of the number of generations. The rate of improvement is very sharp at the beginning and then slows down dramatically. For case 1, a value within 5% of the best solution was reached after only 16 generations for the five runs conducted. For case 2, such a value was reached after 65 generations. Running the program for 16 and 65 generations requires less than 2 and 8 minutes, respectively.

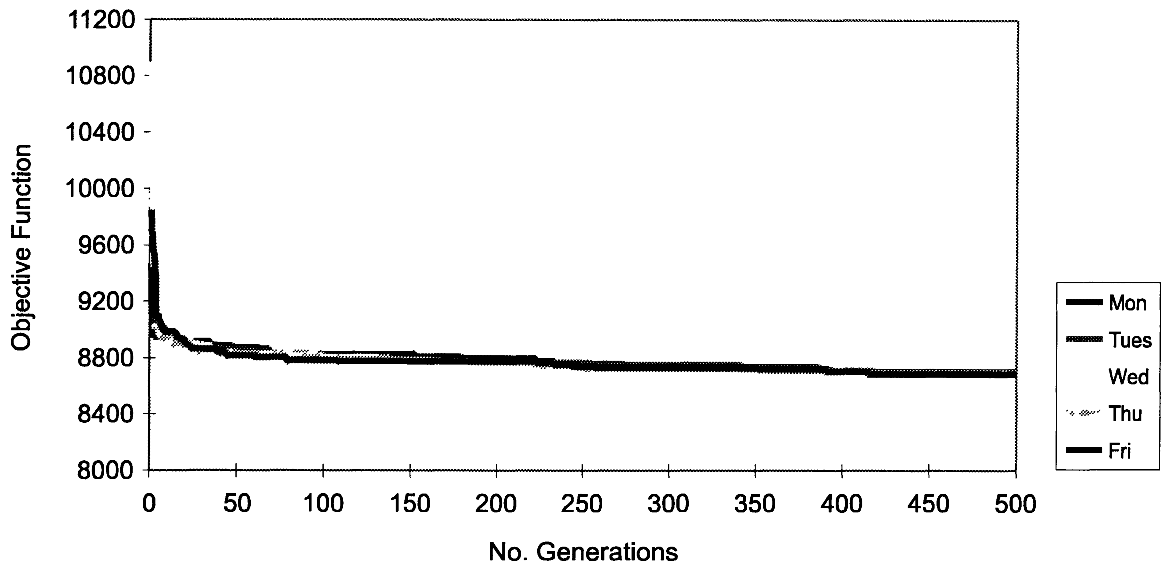

Figure 18. GA Results for Case 1

DISCUSSION

Preliminary experimentation with the developed prototypes was encouraging. The diversion percentages recommended by the shock-wave DSS agreed with prior expectations. For the heuristic search/DTA model, the results were consistent regardless of the start point, and the execution time was quite reasonable. More accurate assessment of the two models and a comparison of their performance are awaiting the availability of traffic data from the Suffolk TMS.

CONCLUSIONS

The two prototypes vary in their accuracy, complexity, input, and computational requirements. The shock-wave model (1) is simple to use, (2) requires the minimum amount of input data, and (3) can be executed in real time. However, it has a number of simplifying assumptions that are likely to affect the accuracy of the results adversely, is limited to one specific routing scenario, and merely attempts to influence traffic coming from a single approach (Route 44).

The dynamic traffic assignment/heuristic search model allows for a significant improvement in the capturing true traffic dynamics over the shock-wave model. This model (1) accounts for the dynamic nature of traffic demand/supply, (2) takes into consideration the traffic volumes entering and exiting at the various access/exit locations of the network, (3) is capable of capturing spillback and lane blockage effects, and (4) allows for considering multiple O-D pairs and multiple routing scenarios. The input and computational requirements of the model, however, are more demanding than the simple computational requirements of the model, however, are more demanding than the simple shock-wave model. Nevertheless, it is still suited for quasi real-time applications based on rolling horizon approaches, where the model will be rerun every 5 or 10 minutes.

RECOMMENDATIONS

1. *Since time savings resulting from the implementation of routing strategies increase with the increase in the number of alternate routes available, make decisions regarding the locations for any new VMS in the region after carefully considering the additional opportunities for routing the new VMS provides.*

2. *Since traffic data are crucial for developing, calibrating, and evaluating routing DSSs, provide TMSs with the functionality that allows for the easy archival and retrieval of historical traffic data.*

3. *Since the development of real-time routing strategies is demanding in terms of computational requirements, develop TMSs in a fashion that allows for incorporating higher performance computing resources as they become available.*

4. *Evaluate and test further the tools developed in this study.* To do this, we suggest that a detailed CORSIM simulation model of the network be developed and calibrated once traffic data become available from the Suffolk TMS. The CORSIM model can then be used for testing the two developed prototypes and comparing the effectiveness of the routing strategies recommended by each prototype.

5. *Since the development of effective routing strategies requires tools for the on-line estimation of O-D matrices, focus future research studies on this area as a means of improving upon the existing O-D estimation procedures.*

6. *Refine the search algorithms.* This could involve testing different sets of their control parameters, trying different annealing schedules to speed up the SA algorithm, or even designing a hybrid GA/SA approach where the GA is used as a preprocessor to perform the initial search before turning the search process over to the SA algorithm. An interesting observation that comes out of the SA results (Figures 16 and 17) is that if the initial solution is close to the optimum, the speed of convergence is greatly enhanced. This suggests that the procedure would be greatly aided by some means of generating good initial solutions. The feasibility of adding such functionality to the SAs should be investigated.

7. *Once the TMS is on-line, investigate the question of how to use motorist information for system control by studying how devices such as VMSs can be used to influence drivers' route selection.* The purpose of the study would be to identify the effects of VMSs on link flows and the extent to which traffic volume shifts because of traveler

information. The results would then be used in formulating a set of "information" strategies that could be used to achieve the desired diversion levels.

REFERENCES

1. Merchant, D. K., and G. L. Nemhauser. 1978. A Model and an Algorithm for the Dynamic Traffic Assignment Problem. *Transportation Science*, Vol. 12, No. 3, pp. 183-199.

2. Ho, J. K. 1980. A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem. *Transportation Science*, Vol. 14, No. 4, pp. 295-305.

3. Carey, M. 1987. Optimal Time-Varying Flows on Congested Networks. *Operations Research*, Vol. 35, No. 1, pp. 58-69.

4. Friesz, T. L., J. Luque, R Tobin, and B. W. Wie. 1989. Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem. *Operations Research*, Vol. 37, No. 6, pp. 893-901.

5. Ran, B., D. E. Boyce, and L. J. LeBlanc. 1993. A New Class of Instantaneous Dynamic User-optimal Traffic Assignment Models. *Operations Research*, Vol. 41, No. 1, pp. 192-202.

6. Papageorgiou, M. 1990. Dynamic Modeling, Assignment and Route Guidance in Traffic Networks. *Transportation Research*, Vol. 24B, No. 6, pp. 471-495.

7. Van Aerde, M., and S. Yagar. 1988. Dynamic Integrated Freeway/Traffic Signal Networks: A Routing-based Modeling Approach. *Transportation Research*, Vol. 22A, No. 6, pp. 435-443.

8. Yagar, S. 1993. INTEGRATION: A Critique. In *Proceedings of the Advanced Traffic Management Conference*, St. Petersburg, Florida, pp. 71-89.

9. Mahmassani, H. S., S. Peeta, T. Hu, and A. Ziliaskopoulos. 1993. Dynamic Traffic Assignment with Multiple-User Classes for Real-Time ATIS/ATMS Applications. *Proceedings of the Advanced Traffic Management Conference*, St. Petersburg, Florida, pp. 91-113.

10. Jayakrishnan, R., H. S. Mahmassani, and T. Hu. 1994. An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks. *Transportation Research*, Vol. 2C, pp. 129-147.

11. COMSIS Corporation. 1996. *Hampton Roads Crossing Study: Compendium of Technical Traffic Information.* Prepared for the Virginia Department of Transportation.

12. Michalewicz, Z. 1994. *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer-Verlag, Berlin, Heidelberg, New York.

13. Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science*, Vol. 220, pp. 671-680.

14. Hillier, F. S., and G. J. Lieberman. 1986. *Introduction to Operations Research.* Holden-Day, Inc., Oakland, Calif.

15. Drissi-Kaitouni, O., and A. Hameda-Benchekroun. 1992. A Dynamic Traffic Assignment Model and Solution Algorithm. *Transportation Science,* 26, pp. 119-128.

16. Carey, M. 1992. Nonconvexity of the Dynamic Traffic Assignment Problem. *Transportation Research*, Vol. 26B, No. 2, pp. 127-133.

17. May, A. D. 1990. *Traffic Flow Fundamentals.* Prentice-Hall, Englewood Cliffs, N.J.

18. Lighthill, M. H., and G. B. Whitham. 1957. On Kinematic Waves: A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society, Series A,* Vol. 229, pp. 317-345.

19. Chang, G. L., H. S. Mahmassani, and R. Herman. 1985. A Macroparticle Traffic Simulation Model to Investigate Peak-period Commuter Decision Dynamics. *Transportation Research Record 1005,* pp. 107-120. Transportation Research Board, Washington, D.C.

20. Peeta, S., and H. S. Mahmassani. 1995. Multiple User Classes Real-time Traffic Assignment for Online Operations: A Rolling Horizon Solution Framework. *Transportation Research,* Vol. 3C, No. 2, pp. 83-98.

21. Golden, B. L., and C. C. Skiscim. 1986. Using Simulated Annealing to Solve Routing and Location Problems. *Naval Research Logistics Quarterly*, Vol. 33, pp. 261-279.