FINAL REPORT

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

Adel W. Sadek
Graduate Research Assistant

Brian L. Smith, Ph.D.
Faculty Research Scientist
Research Assistant Professor of Civil Engineering

Catherine C. McGhee
Senior Research Scientist

Michael J. Demetsky, Ph.D.
Faculty Research Scientist
Professor of Civil Engineering

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

In Cooperation with the U.S. Department of Transportation
Federal Highway Administration

Charlottesville, Virginia

October 1998
VTRC 99-R10

ABSTRACT

Real-time traffic flow routing is a promising approach to alleviating congestion. Existing approaches to developing real-time routing strategies, however, have limitations. This study explored the potential for using case-based reasoning (CBR), an emerging artificial intelligence paradigm, to overcome such limitations. CBR solves new problems by reusing solutions of similar past problems.

To illustrate the feasibility of the approach, the research team developed and evaluated a prototype CBR routing system for the interstate network in Hampton Roads, Virginia. They generated cases for building the system's case-base using a heuristic dynamic traffic assignment (DTA) model designed for the region. Using a second set of cases, the research team evaluated the performance of the prototype system by comparing its solutions with those of the DTA model.

The research team found that CBR has the potential to overcome many of the limitations to existing approaches to real-time routing and a CBR routing system is capable of producing high-quality solutions with reasonable a case-base size. In addition, the research team found that real-time traffic flow routing will likely lead to significant user cost savings.

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

Adel W. Sadek
Graduate Research Assistant

Brian L. Smith, Ph.D.
Faculty Research Scientist
Research Assistant Professor of Civil Engineering

Catherine C. McGhee
Senior Research Scientist

Michael J. Demetsky, Ph.D.
Faculty Research Scientist
Professor of Civil Engineering

INTRODUCTION

Managing traffic flow through real-time route guidance has emerged as one of the promising approaches to alleviating congestion. This approach uses sensors such as inductive loops and closed circuit television (CCTV) cameras to monitor traffic flow continuously on the different segments of the highway network. This information is then used to develop real-time route guidance strategies that suggest routes to drivers to try to use the network capacity fully.

Although considerable investment has been made in the hardware needed for implementing real-time route guidance, relatively little attention has been paid to developing effective decision support systems (DSS) for the development of sound real-time routing strategies. Phase I of this project, described in the Volume I report,[1] resulted in the development of heuristic search/dynamic traffic assignment (DTA) models that can be used to determine near-optimal routing strategies. Two models were developed: the first employed simulated annealing (SA) as the search algorithm, referred to as the SA-DTA model, and the other used genetic algorithms (GAs), referred to as the GA-DTA model. Given the current and predicted travel conditions, these models can be used to find the routing strategy that will optimize a particular network performance criterion, such as total travel time. This developed routing strategy describes how traffic should be distributed, over time, at each node of the network and is defined by the set of time-varying traffic splits for the prediction horizon considered.

However, for a routing DSS to be effective, it must be able to function in real time. As soon as traffic conditions change, such as when an incident occurs, routing strategies must be revised to mitigate the effects. The SA-DTA and GA-DTA models developed during the first phase of this study could not meet this requirement.

CBR is an emerging artificial intelligence paradigm that has the potential to provide for such real-time functionality. CBR solves new problems by reusing solutions of similar past problems. It is based on the observation that when people solve a new problem, they often base the solution on one that worked for a similar problem in the past. The motivation for adopting a CBR approach for real-time traffic routing is that a CBR system, by reusing successful routing strategies for similar conditions from its case-base, will avoid the need to solve the problem from the beginning using a complex mathematical model each time. This should enable the system to function in real time.

However, the concern with using CBR for a problem such as real-time routing is that the combinatorics of the problem might force the need for prohibitively large case-bases to achieve satisfactory performance. For example, the "status" of an urban freeway system can take on a nearly infinite number of states. For a CBR approach to be feasible, the system should be capable of producing solutions of satisfactory quality using case-bases of reasonable size.

PURPOSE AND SCOPE

The purpose of this second phase of the study was to investigate the feasibility of using CBR to provide for the real-time functionality required of a routing DSS. To illustrate the feasibility of the CBR approach for real-time traffic flow management, this phase of the study developed and evaluated a prototype CBR routing DSS for the same network considered in the first phase. This network, as shown in Figure 1, is composed of the interstate system in the Hampton Roads area of Virginia, namely, I-64, I-664, I-264, and I-464. In addition to being one of the most heavily congested areas in Virginia, the Hampton Roads area network is highly dependent on only two major water crossings: the Hampton Roads Bridge Tunnel (HRBT) and the Monitor-Merrimac Bridge Tunnel (MMBT).

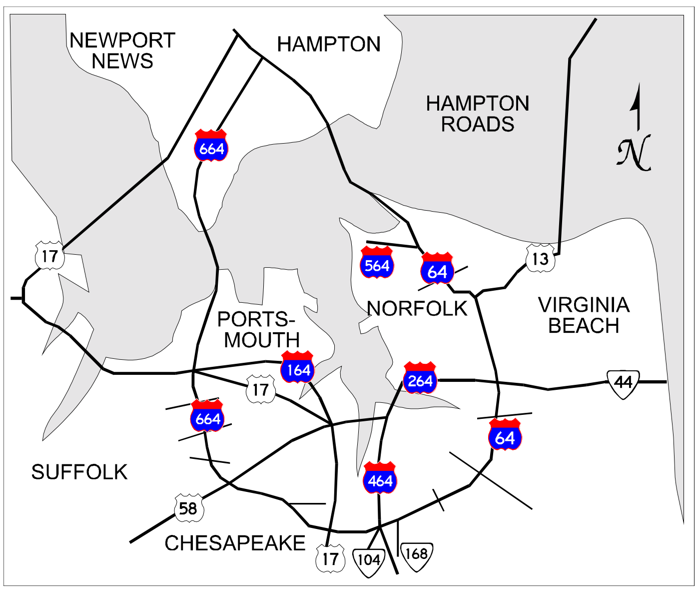

Figure 1. The Hampton Roads Network

METHODOLOGY

This phase of the project consisted of three major stages:

1. developing the system architecture
2. designing the prototype system
3. developing and evaluating the prototype system.

Developing the System Architecture

Case-Based Reasoning

Although CBR has been receiving increased attention over the last few years,[2] CBR applications in transportation engineering are still in a very early stage. A literature review identified only three examples of such applications. The first applied CBR to the air traffic control problem.[10] The second used CBR as a planning tool to select ITS projects,[11] and the third used CBR as a diagnostic tool for addressing transit maintenance problems.[12]

At a basic level, CBR solves a new problem by remembering a previous similar situation and reusing information and knowledge from that solution. CBR can be represented as a cycle consisting of four processes (Figure 2):

1. RETRIEVE the most similar case or cases.
2. REUSE the information and knowledge in that case to solve the problem.
3. REVISE the proposed solution.
4. RETAIN the parts of this experience to be used for future applications.

At the core of the CBR process is a case-base that stores previous instances of problems and their derived solutions. When faced with a new problem, the system first accesses the case-base and retrieves the case(s) most similar to the new case. During the reuse process, the solution of the retrieved case is adapted to address the current problem more appropriately. The solution is then tested for success during the revise process, by either directly implementing it in the real world or by evaluating it by a teacher (e.g., a human expert). If the case is a success, it is retained in the case-base for future reuse.

CBR uses specific knowledge of previous situations or cases and allows for incremental, sustained learning, since a new experience can be saved each time a problem has been solved. This new case becomes available for future problems. As pointed out by Aamodt and Plaza,[2] no universal CBR methods exist that can be applied for every domain of application. The challenge for any CBR research effort is thus to come up with methods suited to the application

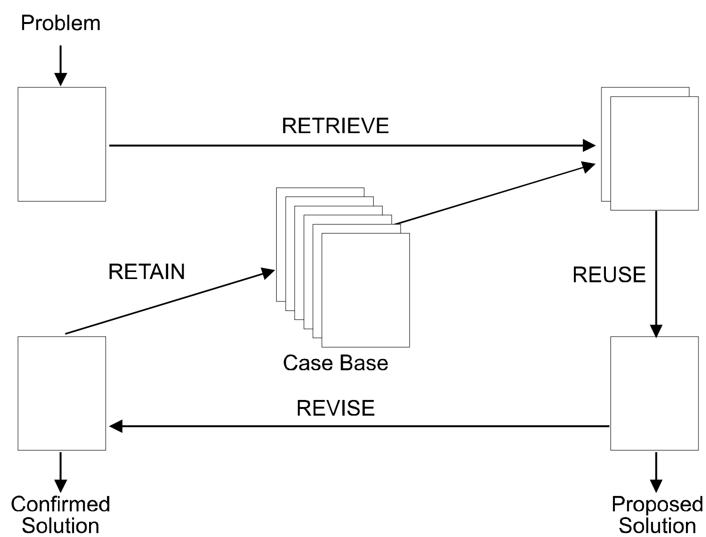

Figure 2. The CBR Cycle

environment under consideration. Core problems that need to be tackled by a CBR research effort can be divided into five areas:

1. case representation

2. case-base construction

3. indexing and retrieval methods

4. adaptation methods

5. revise and retainment methods.

A set of solutions to these five problems constitutes a CBR method.

Case Representation

According to Kolodner,[3] a case is a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to the goals of the reasoner. Kolodner thus views a case as consisting of two major parts: the lesson it teaches and the context in which it can teach its lesson. Watson,[4] on the other hand, gives a more specific definition of a case. He describes a case as consisting of three basic components:

1. the *problem,* which describes the state of the world when the case occurred

2. the *solution,* which gives the derived solution to that problem

3. the *outcome,* which depicts the state of the world after solution implementation.

Case-base Construction

Kolodner[3] gave the following three guidelines for selecting cases for building the case-base:

1. Cases should cover the range of reasoning tasks for which the system is responsible.

2. For the range of reasoning tasks, cases should cover the range of well-known solutions and common mistakes.

3. Collecting cases should be an incremental process. Whenever a certain deficiency is discovered in a system's case-base, cases should be added to fill in the revealed gap.

In selecting cases for the case-base, there is always a tradeoff between the size of the case-base (i.e., the number of cases stored) and the search speed. Storing more cases is likely to improve the quality of the CBR system's solutions since it provides for more problem coverage. However, as the size of the case-base increases, more search effort is needed to locate the most similar case.

Indexing and Retrieval Methods

Case indexing entails assigning indices to cases to facilitate their retrieval. A number of guidelines have been proposed for index selection[3,4]:

- Indices should be predictive. Predictive features are those combinations of descriptors of a case that led to solving it the way it was solved and those that influenced the outcome.

- Indices should be more abstract than the details of a particular case to allow for applying the case to as broad a collection of situations as appropriate.

- Indices should be concrete enough to be easily recognized in the future.

- Indices should be selected so as to make their predictions useful in reasoning.

In the retrieval process, the system uses the features of cases to retrieve the most similar case(s) to the current problem or situation. There are several methods for case retrieval. The most commonly used is the nearest neighbor (NN) approach. This approach assesses the similarity between the new and the stored cases based on matching a weighted sum of features. Given a query *q* and a case library *L*, the NN algorithm retrieves the most similar (i.e., least distant) case, *x*, in *L*. The distance is defined as:

$$distance\,(x,q) \quad = \quad \sqrt{\sum\nolimits_{f=1}^{n} w_f \;\; x\; difference(x_f}\,,q_f\,)^2 \qquad [1]$$

where

w_f is the parameterized weight value assigned to feature f

the difference (x_f, q_f) is equal to

$\lvert x_f - q_f \rvert$	if feature f is numeric
0	if feature f is symbolic and $x_f = q_f$
1	otherwise

Numeric features are typically normalized (by subtracting their mean and dividing by their standard deviation) to ensure they have the same range, and hence the expected impact.[36]

Adaptation Methods

The most trivial type of reuse is when the solution of the retrieved case is directly applied to the new situation. This rarely happens. In most applications, there will be a need to transform or adapt the old solution so as to fit the new problem. Adaptation algorithms can be broadly classified into structural adaptation and derivational adaptation.[4,6]

Structural adaptation directly adapts the solution of the retrieved case. The most common technique is to replace a component of the previous solution by a new value that may be provided by an auxiliary knowledge source.

In derivational adaptation, the methods, rules, or algorithms that generated the original solution are replayed to produce a new solution to the current problem. This will typically necessitate storing the planning sequence that constructed a solution along with the solution. The PRODIGY/ANALOGY program developed by Veloso and Carbonell at Carnegie Mellon is an example of this approach.[7]

Adaptation methods are usually domain specific. However, they generally employ either one or a combination of the following three techniques:

1. the use of a set of domain-dependent adaptation rules as is in CHEF, a CBR system for cooking recipe planning[8,9]

2. the use of a domain model as in CASEY[3]

3. the use of pieces of existing cases.

Revision and Retainment Methods

When a proposed solution fails to achieve the desired result, an opportunity exists for learning from failure. This phase is called the revision phase and consists of two subphases: (1) the evaluation phase, where the proposed solution is evaluated; and (2) the repair phase, where the solution is repaired using domain-specific knowledge.

The repair phase involves detecting errors in the current solution and retrieving explanations for them. Using the failure explanations generated, this phase then attempts to modify the solution so that the detected failures will not occur again. The CHEF system[8] is one of the best examples for this phase. CHEF uses an explanation-based technique to learn the situations that will cause failure. Steps are then added to the failed plan to prevent the causes of errors from occurring. In many instances, however, the repair phase will require human intervention.

Retainment describes the process of incorporating the new problem-solving episodes into the existing knowledge. The process is triggered by the outcome of the evaluation and possible repair phases and involves the following subtasks:

1. selecting which information from the case to retain

2. deciding how the case should be indexed for later use

3. integrating the new case into the memory structure.

Proposed System Architecture

Figure 3 shows the proposed architecture, which consists of five modules:

1. a match/retrieve module that retrieves cases from the case-base

2. a case-base that stores instances of previous congestion problems in the region along with the routing strategy recommended to solve each problem

3. an adaptation module that adapts retrieved cases to the needs of the current problem

4. an evaluation module

5. a learning module.

The system will receive current and predicted traffic information from the surveillance and prediction subsystems of an Advanced Traffic Management System (ATMS). Using this information, the match/retrieve module will access the case-base and attempt to retrieve the most similar previous case(s). The retrieved case(s) is then passed to the adaptation module. There are two possibilities here. First, if a sufficiently close case (this can be judged from the

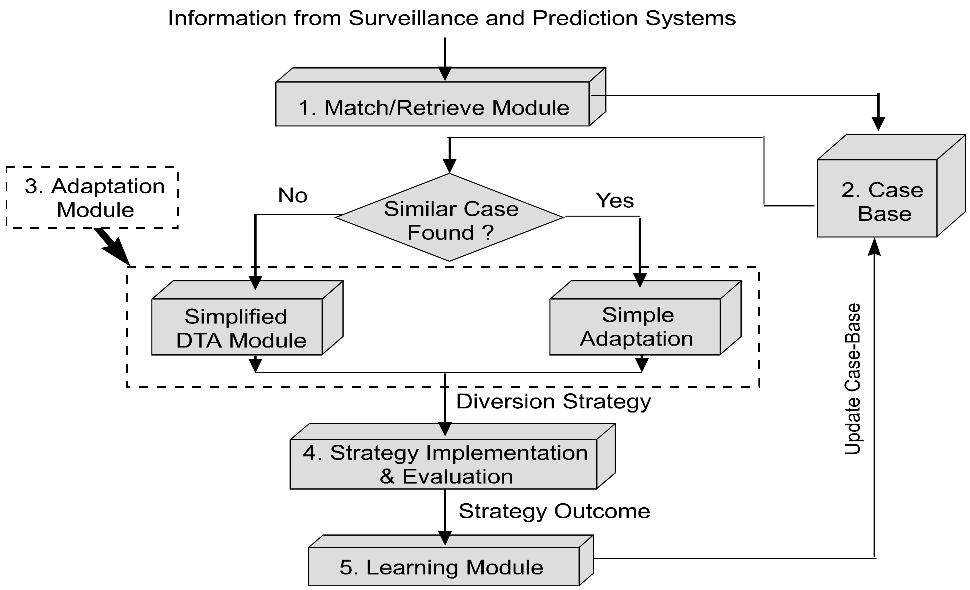

Figure 3. Proposed System Architecture

similarity score value) is retrieved, the solution of such case can be directly reused after the use of straightforward adaptation techniques. If this is not the case, control is passed to the DTA module.

The DTA component will determine the recommended traffic splits and their duration using the SA-DTA model developed during the first phase of this project. The case retrieved by the match/retrieve module, though not sufficiently close to allow for simple adaptation, should still provide for a good starting point for the search algorithm. In other words, the algorithm would not be starting its search procedure from scratch, but the retrieved case will help direct the search effort toward the most promising areas in the search space. Without these cases, the algorithm will have to start its search procedure at a random point, which could be far from the optimal solution.

The recommended routing strategy is then implemented through information dissemination devices such as variable message signs (VMS) and highway advisory radio. Using the feedback provided by the ATMS's surveillance system, the learning module will determine whether the strategy is a success or a failure (i.e., to determine the desirability index of the case). If the new case is a success, it is retained by the system to speed up the solution procedure when faced with a similar situation. If it is a failure, it is assigned a low desirability index. A low desirability index acts as a penalty term that prevents the case from being retrieved in the future. This prevents the system from repeating its mistakes.

Ideally, the case-base of the proposed system should consist of routing scenarios that have been applied in the real world and have had their outcome assessed. This will allow the system to take into account the uncertainty associated with the problem, since it ensures that the system's recommendations are based on strategies that have worked in the real world. Unfortunately, these routing cases are typically not available prior to the implementation of the system in the real world. Either the system is new or historical data with routing strategies employed have not been archived. There is, therefore, a need to "seed" the system with an initial case-base.

The current study built the initial case-base using the SA-DTA model developed in the first phase. The model uses the following inputs: the initial state or the destined traffic density matrix at the time of initiation of the routing strategy, the traveler O-D matrix for the upcoming 15 minutes, and the details of the incident scenario considered. The model then outputs the "optimal" routing strategy that will result in the minimum total vehicles travel time. A description of the SA-DTA model can be found in the Volume I report.[1]

Consequently, to build the case-base, the range of problems and traffic conditions expected to occur in the study area was first identified. For each problem, the SA-DTA model was used to arrive at the near-optimal routing strategy. These "prototypical" problems and their solutions, as obtained from the SA-DTA model, were then stored in the initial case-base. After implementation, the initial case-base will be augmented by the new real-world cases acquired through the learning module.

The focus of this study was on nonrecurrent congestion problems where real-time routing is most needed. For nonrecurrent congestion, the operation of the system is as follows. The occurrence and verification of an incident will trigger the system. The system will operate on a rolling horizon basis (as discussed in the first phase of the project, a rolling horizon approach helps in handling the uncertainty associated with predicting travel demand and driver behavior),[13] where the problem is solved in a number of stages (Figure 4). For each stage, the system considers a prediction horizon that is shorter than the total planning horizon of the problem. The system will remain in operation until the incident is cleared. At that time, it will recommend a strategy that will help flow return to normal.

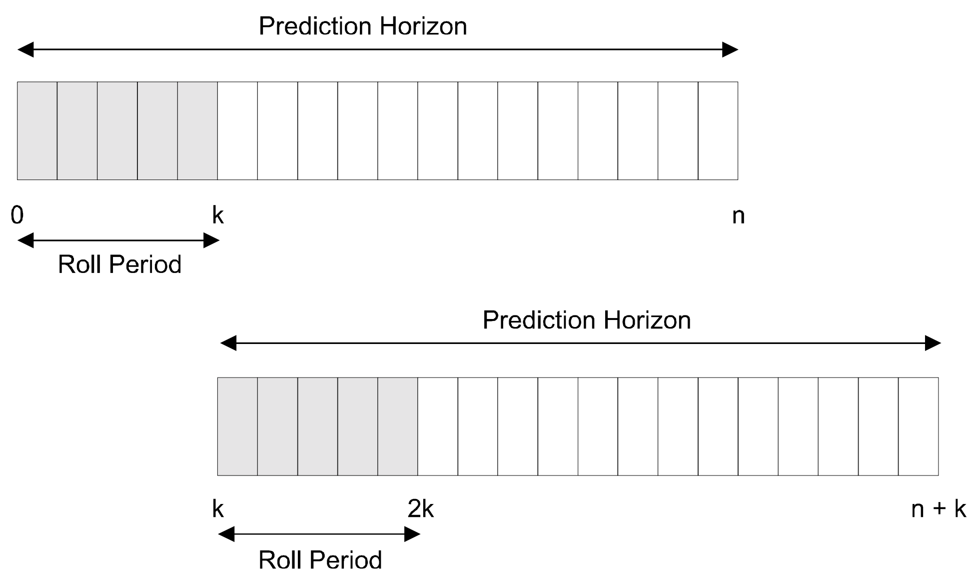

Figure 4. Rolling Horizon Approach

A rolling horizon approach has implications with respect to the real-time execution required of a routing DSS. For the first stage, real-time execution is crucial; within minutes of an incident being detected and verified, a routing strategy has to be implemented to mitigate the effects of the incident. For subsequent stages, however, the system is required to solve the problem in a time period that is short enough to allow the solution for the upcoming stage to reflect actual demand and network configuration (typically around 5 minutes).

To summarize, the proposed architecture offers a number of advantages over existing approaches to real-time traffic flow management. These include:

- By reusing similar routing scenarios from its case-base, the proposed system will avoid the need to solve a complex mathematical model each time. This should enable the system to function in real time.

- By acquiring new cases, the system will learn to refine its performance over time. In addition, the system will grow in a manner that reflects site-specific experience.

- CBR offers a means for handling the uncertainty associated with predicting travel demand and driver behavior. This is because recommended strategies stored in the case-base can be based on solutions that actually worked on similar cases in the real world. Moreover, using real-world cases could be advantageous over using a model-based approach, since any model, regardless of its level of sophistication, is still an abstraction of reality.

- The knowledge acquisition process is simplified since it merely involves acquiring past cases; this removes the difficulty associated with formulating rigid rules for route selection. Moreover, the system can be initially implemented using a partial case-base, since it will continuously grow through the feedback and learning processes. This is important since very little is currently known about the implications of route diversion.

- Real-time traffic flow management is a domain characterized by continuous, gradual change. For example, drivers' response to information is likely to vary with time, as they gain or lose confidence in the system's recommendations. For such domains, traditional rule-based expert systems would exhibit a continuous and mysterious degradation in performance (unless the rule base is revised on a regular basis). CBR, on the other hand, is well suited for continuously changing domain. Previous cases are applied to new situations, and new cases are created (through the learning process) to address situations for which the existing cases are inadequate.

Designing the Prototype System

With the system architecture developed, the current stage focused on designing the prototype routing DSS and its case-base. Figure 5 shows the network's link-node diagram used in developing the SA-DTA and the GA-DTA models of the project's first phase, as well as in

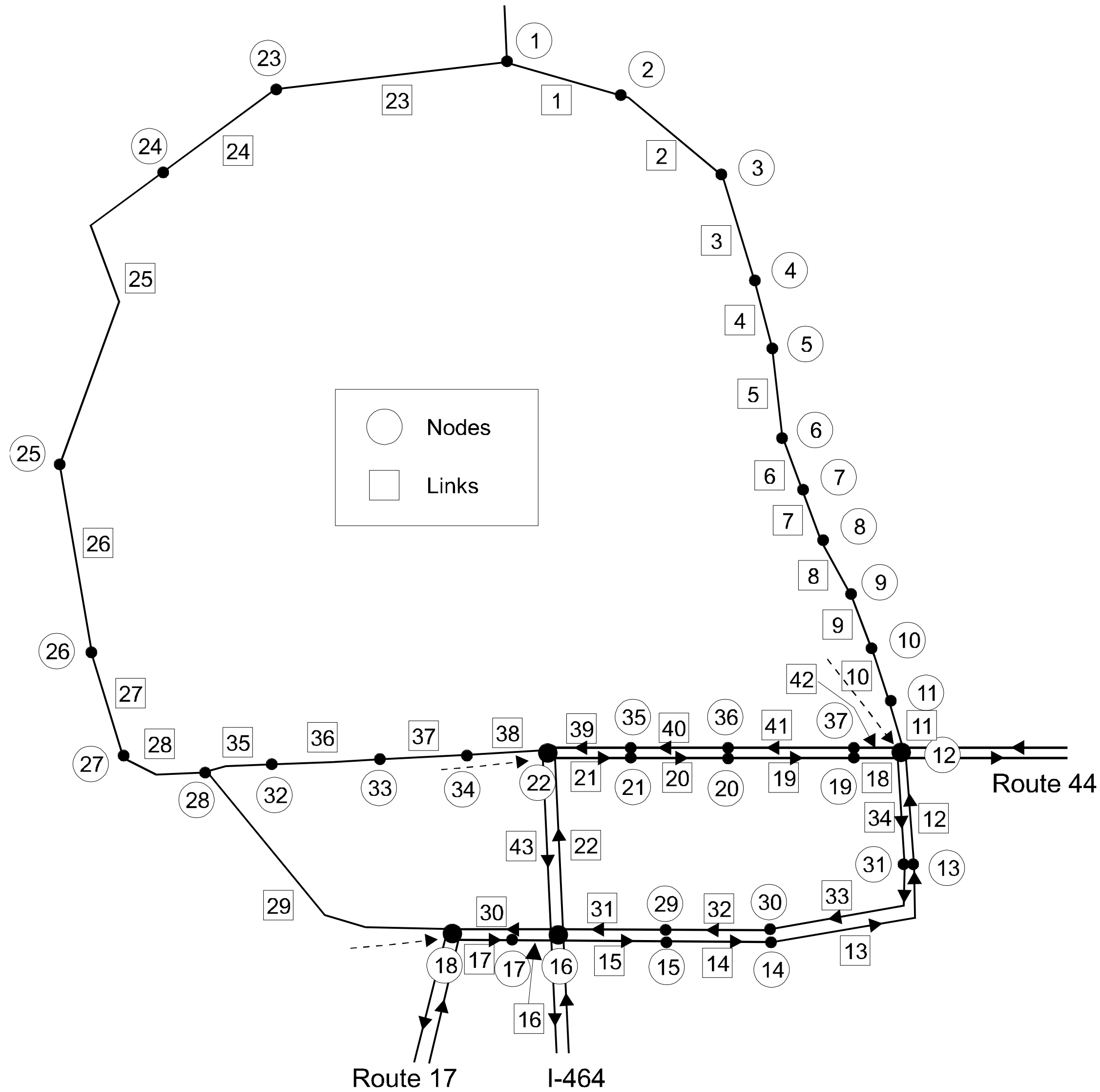

Figure 5. Link-Node Diagram of DTA Model Network

developing the prototype CBR system. As can be seen, the network consists of 37 zones and 43 one-way links. The network considered, given the location of the VMS, allows for routing traffic originating from the following routes: Route 44, I-464, and Route 17, and destined to I-64W.

Case Representation

A case typically consists of three components: the problem component, the solution component, and the outcome component.

The Problem Component

For a routing DSS, the features required to describe the problem component of a case are divided into two broad categories. The first includes those features needed to describe the current and the predicted traffic demand on the different links of the network. The second consists of the features describing the incident scenario under consideration.

Ideally, the current traffic demand for the Hampton Roads test network should be represented by the initial state destined density matrix. This matrix, which takes the form of a 43 x 37 matrix, describes the portion of the traffic density on each of the 43 links of the network that is "destined" for each of the 37 destinations (see Figure 5). The predicted demand is given by the O-D matrix (a 37 x 37 matrix) for the upcoming 15 minutes. From a practical standpoint, however, these data are not readily available to the operator of a traffic management center. The available data are the traffic volumes on the different links as obtained from the system's sensors. In a typical DTA approach to real-time traffic flow management, the current volumes are first used to predict future volumes using some sort of a traffic prediction model.[14,15] A dynamic O-D estimation procedure is then employed to deduce the O-D matrix from the current and predicted volumes.[16] Finally, the deduced matrices are fed into the DTA model to determine the "optimal" routing strategy.

For a CBR approach, it would therefore be quite advantageous if cases could be defined directly in terms of link volumes. Describing cases in terms of link volumes will allow one to "bypass" the need to perform the O-D matrix estimation on-line**.** This should result in additional savings in execution time. This is the approach the current study adopted for developing the prototype.

The initial state of a case is described by the current traffic volumes on the 43 links of the Hampton Roads network. The predicted demand, on the other hand, is represented by the volumes on the 43 links that would have resulted from assigning the O-D matrix for the upcoming 15-minute interval, had no routing strategy been in effect, and had no incident have taken place. To reflect the typical behavior of drivers in the region, the splitting rates for this case of no routing and no incident were obtained from a traditional static traffic assignment procedure.

Finally, the problem component is completed by defining the location of the incident; the incident severity; and the estimated duration of the incident. Representing the location and severity of an incident is quite straightforward. The incident location is described by the link number, on which the incident took place, whereas the incident severity is represented by the value of the capacity of the freeway section, remaining after the incident has occurred.

For describing the expected incident duration, however, it is important to remember that the prototype system is designed to operate on a rolling horizon basis, where the system solves the DTA problem in stages. For each stage, the system considers a duration that is equal to the length of the prediction horizon considered, which might be less than the expected overall duration of the incident. Since the system's routing cases represent routing strategies developed for a certain prediction horizon, the incident duration feature of a case should represent the time, within the prediction horizon considered, during which the incident is still in existence.

To illustrate, consider the example of an incident duration of 40 minutes, with a prediction horizon and a roll period of 15 minutes. Figure 6 shows the value of the incident duration that needs to be stored in the case structure for each of this problem's four stages.

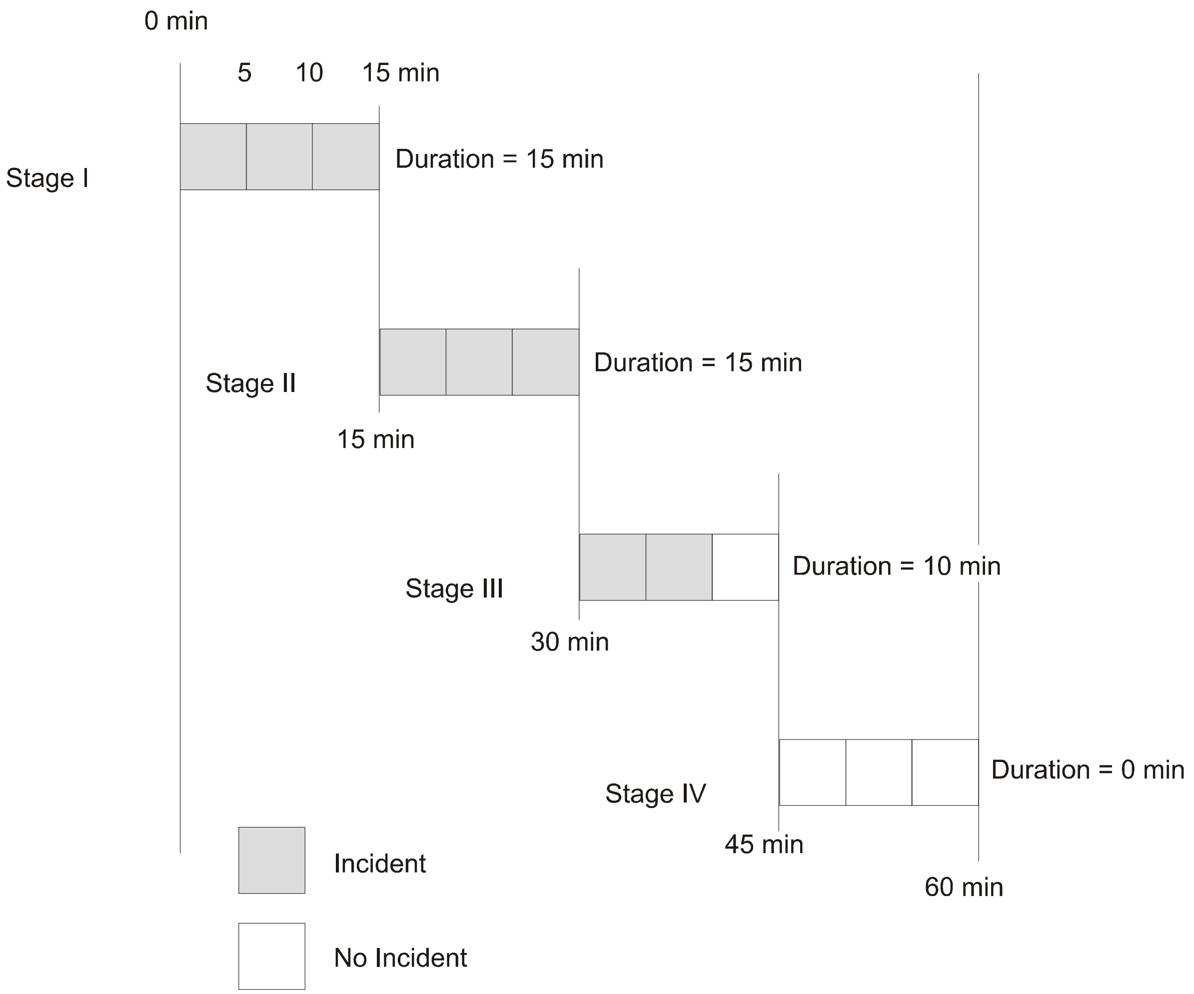

Figure 6. Representing Incident Duration

The Solution Component

For a routing DSS, the solution component describes the "optimal" routing strategy (i.e., the set of the time-varying traffic splits obtained from the DTA model) developed to address the combination of traffic demand and incident characteristics described in the problem component of the case. For the SA-DTA model considered in the current study, such a routing strategy is defined by 51 independent splitting rates.

The Outcome Component

For a routing DSS, this component captures the routing strategy impact on the network performance (as measured by the vehicles' total travel time, for example). The current study uses the value of the SA-DTA model's objective function as a measure of the quality of the implemented solution. This objective function describes the number of vehicles hours spent on the network over the entire planning horizon. This provides a surrogate measure of system delay.

The adopted case structure for the Hampton Roads prototype routing system can thus be summarized as follows (Figure 7). The problem component is represented by 89 features: 43 features for the initial volumes on the 43 links; 43 features for the predicted volumes on the links for the upcoming 15-minute interval; and 3 features for the incident location, severity, and duration. The solution component is represented by 51 features corresponding to the 51 independent splitting rates describing a routing strategy. Finally, the outcome component is described by one feature encoding the total cost of the implemented strategy.

Problem Component
42 features for initial volumes on the 43 links
43 features for predicted volumes on the links
3 features for incident location, severity, and duration

Solution Component
51 features for the 51 independent splitting rates

Outcome Component
1 feature for value of the objective function of the SA-DTA model

Figure 7. Cast Structure for Hampton Roads System

Case-base Construction

Ideally, the routing system's case-base should consist of routing scenarios that have been implemented in the real world and have had their outcome assessed. Unfortunately, these routing cases are typically not available prior to the implementation of the system in the real world. There is, therefore, a need to "seed" the system with an initial case-base. In this study, the SA-DTA model developed in the first phase of the project is used in building this seed case-base. This is done by solving the model for the range of problems and traffic conditions that are expected to occur in the study area.

As previously mentioned, running the DTA model requires the following data: the initial state or the destined traffic density matrix at the time of initiation of the routing strategy, the O-D matrix for the upcoming 15 minutes, and the incident scenario.

Ideally, the initial state destined traffic density and the O-D matrices should be available or estimated from real-time traffic data obtained from the traffic management center. However, since the Suffolk ATMS was not yet on-line at the time the study was conducted, alternate sources of data had to be sought.

The current study had access to two relevant traffic data resources: a calibrated MINUTP travel demand model and 5-minute traffic counts for the two tunnel facilities in the region (the HRBT and the MMBT) for 1995.

The MINUTP travel demand model provided three valuable pieces of information:

1. an estimate of the average daily volume on the different links of the network (from the results of the traffic assignment procedure)

2. an estimate of the average daily O-D trip matrix

3. the destinations of traffic on each link (i.e., the destined traffic density matrix) from the results of the select link analysis procedure.

However, these estimates are averages for an entire day. For the real-time traffic management problem, data are needed for much shorter periods to reflect the dynamic nature of traffic demand and flow.

To address this, the short-term traffic counts from the tunnels are used to scale the MINUTP model estimates. This is done by comparing the model's assigned volume on the tunnels' links to the short-term counts available to establish scaling factors. Two scaling factors are established. The first factor is derived using the HRBT volumes, and applied to the right side of the network (i.e., nodes 1 → 16, 19 → 22, 29 → 31, 35 → 37; along with their associated links). The second scaling factor, on the other hand, is derived using the MMBT volumes and applied to the left side of the network (i.e., nodes 17 → 18, 23 → 28, 32 → 34; along with their associated links).

The problem with such a procedure, however, is that it introduces undesirable correlation among the cases and hence may bias the results of evaluating the performance of the CBR system. In fact, such a procedure implies that one needs to know only the volumes on two links (i.e., the two tunnels) to deduce any 15-minute O-D matrix from the daily matrix; this is quite unrealistic. To overcome this problem, a noise term is added to each cell of the scaled travel demand matrix. The noise term assumes a random value that is within ± 20 percent of the cell's number of trips. This breaks the correlation and more closely resembles real-world traffic patterns and their inherent randomness.

Case Selection Framework

The large number of features used to describe a routing case (Figure 7), coupled with the wide range of values that many of these features can assume, makes it practically impossible to store every possible case in the system's case-base. It thus becomes particularly important to select cases in a fashion that provides an adequate coverage of the range of problems the system is expected to face and, at the same time, keeps the size of the case-base manageable. As previously mentioned, the features needed to describe the problem component of the cases can be divided into two groups: features for describing the current and predicted traffic demand (the 86 features for the initial and predicted volumes on the 43 links) and features for describing the incident scenario (incident location, severity, and duration). A case selection framework should thus strive to cover the expected range of values for these two groups of features while keeping the case-base size manageable.

Case Selection for Covering the Expected Range of Links Volumes

An important characteristic of traffic that justifies the application of a CBR approach is the fact that traffic patterns tend to recur. One should not expect a big difference between the traffic volume at, say, 8:15 A.M. on a certain Monday and the volume at 8:15 A.M. on the following Monday. Figure 8 illustrates this fact. The figure plots the traffic pattern in the Hampton Roads Bridge Tunnel on the four Mondays of the month of March 1995. As can easily be seen, the four traffic patterns plotted are quite similar.

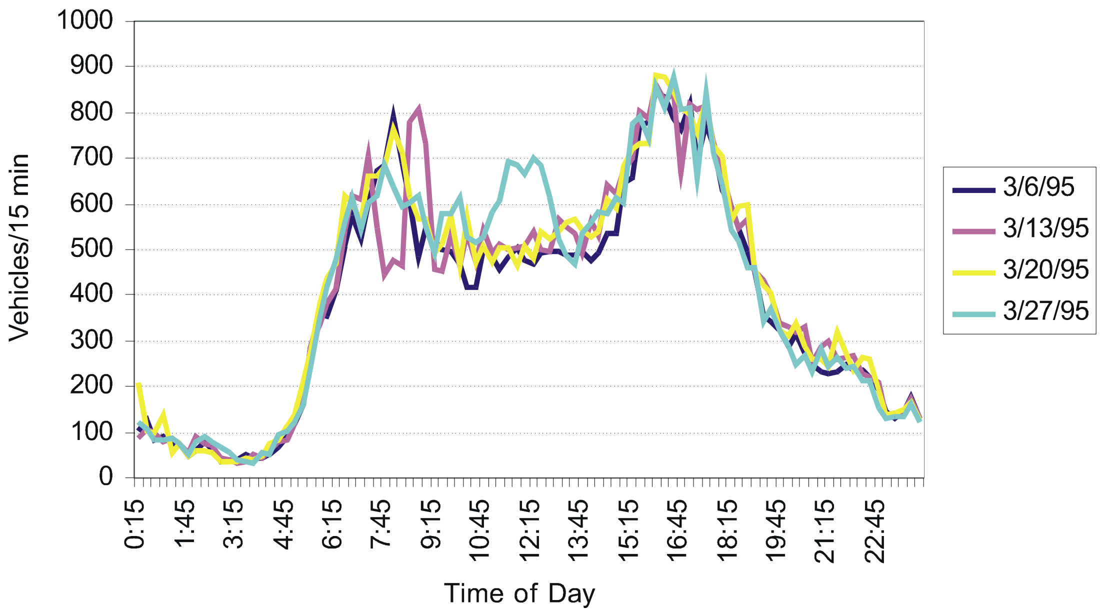

Figure 8. Volume Variation with Time of Day on Mondays for HRBT

Similarly, Monday traffic patterns should be expected to be comparable to Tuesday's patterns, but differences should be expected between traffic patterns on weekdays and weekends. Figures 9 and 10 illustrate this fact. The curves on these two figures represent average counts for March 1995 in 15-minute increments. For example, the Monday curve represents the average of all the Mondays in that month. As can be seen, for Monday through Thursday, traffic patterns are similar; on Fridays, however, they seem to exhibit a slightly different pattern. Moreover, the traffic pattern on weekends is quite different from that on weekdays.

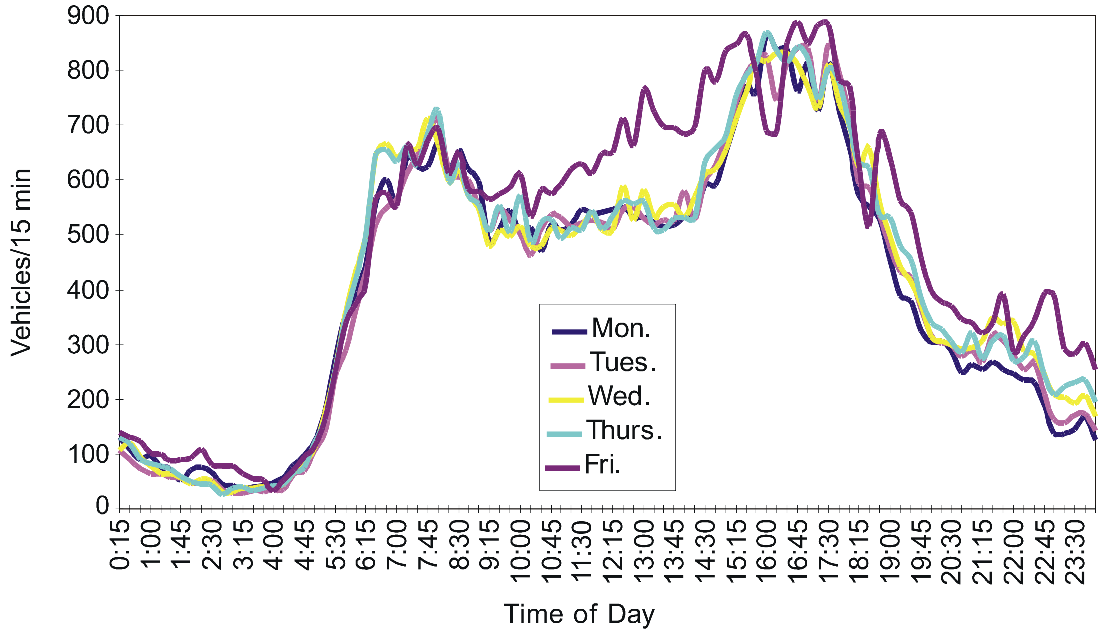

Figure 9. Volume Variation with Time of Day on Weekdays for HRBT

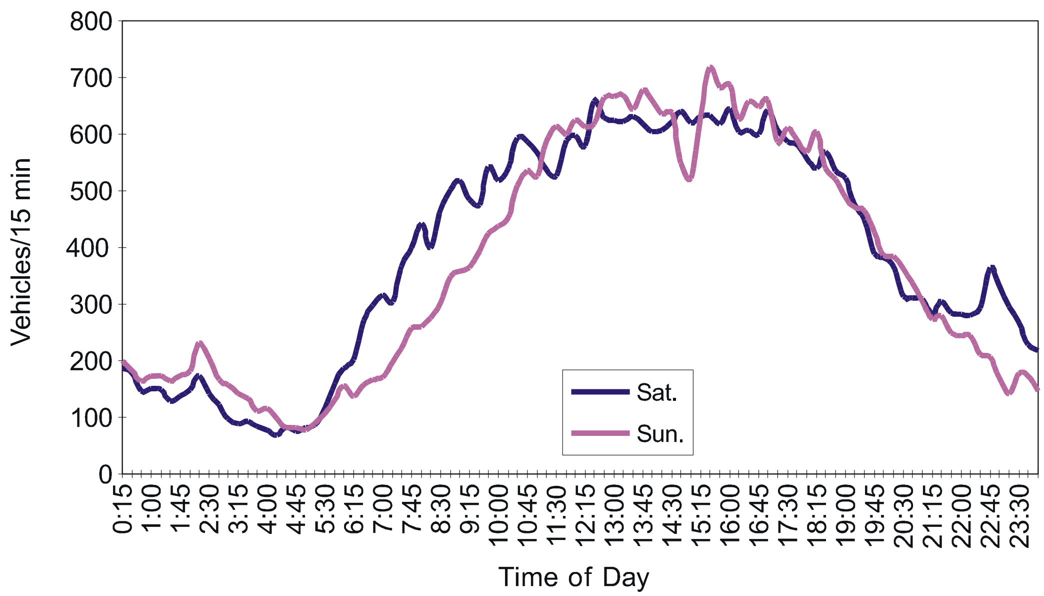

Figure 10. Volume Variation with Time of Day on Weekends for HRBT

Exploiting the recurrent nature of traffic patterns allows for a significant reduction in the size of the case-base. Based on analysis of Figures 9 and 10, the research team decided to consider three groups for defining prototypical traffic patterns for constructing the case-base:

1. group 1 for the weekdays Monday through Thursday

2. group 2 for Fridays

3. group 3 for weekends.

For each group, the average traffic counts for each 15-minute interval of the day were computed. The period over which averaging is performed could be as long as 1 year or limited to a particular season if seasonal variations in traffic patterns are significant. Seasonal variations are the norm for recreational areas.

Each group thus has 96 different traffic patterns (since there are 96 15-minute intervals within a 24-hour period). By considering these three groups, only 288 cases (96 cases/group x 3 groups) need to be stored for a particular incident scenario. The premise here is that these average cases would provide for adequate coverage for the range of traffic volumes that are expected to occur during the period considered. If more cases are required, for example to account for seasonal variation, they can be added.

Case Selection for Covering the Expected Range of Incident Locations

Within the framework of a DTA model, an incident location refers to the particular link on which an incident has taken place. For the Hampton Roads network (Figure 5), this means that one has 43 possible incident locations to consider in building the case-base, since the network has 43 one-way links. However, it might be possible to group links (i.e., incident locations) into clusters that have similar routing strategies; this would help reduce the number of incident locations that need to be considered. For example, it is very likely that the routing strategy determined for the case where an incident has occurred on link 6 (Figure 5) would be quite similar to that for link 7 incident, provided that the traffic volumes and the incident severity levels are similar.

To determine which incident locations could be grouped together, an experiment was set up where the SA-DTA model was used to find optimal strategies for a number of cases, where for each case, an incident was assumed to have taken place on one of the 43 links of the network. In this experiment, the incident severity level was kept constant and two levels (high and low) of traffic demand were considered, resulting in two sets of 43 cases each. For each traffic demand level, the resulting routing strategies for the 43 incident locations were then compared to see which locations resulted in similar solutions. Taking into account other considerations such as the links' number of lanes and their geographic adjacency, 17 link clusters were defined. Table 1 shows these clusters along with the number of lanes of the links of each cluster and the link chosen to be the cluster's representative link.

Case Selection for Covering the Expected Range of Incident Severity Levels

The incident severity level refers to the resulting reduction in the capacity of a roadway segment. Previous research within the incident management research community has established typical values for incidents' capacity reductions based on the number of lanes blocked and the original number of lanes of a segment (Table 2). In the current study, these values are considered for developing the case-base. By using these values, the system stores cases corresponding to three incident severity levels for three-lane segments and two for two-lane segments.

Table 1. Link Clusters

Cluster No.	Links	No. of Lanes	Representative Link
1	1-2	3	2
2	3-5	2	4
3	6-8	3	7
4	9-11	3	10
5	12-14	3	13
6	15-17	2	16
7	18-20	3	19
8	21	3	21
9	22	2	22
10	23-24	3	24
11	25-30	2	27
12	31,33	2	33
13	32,34	3	32
14	35	2	35
15	36-38	3	37
16	39-42	3	40
17	43	2	43

Table 2. Capacity Reductions Attributable to Incidents (%)

Lanes Blocked	Original No. of Lanes	
	3	2
1	50	60
2	80	100
3	100	-

Case Selection for Covering the Expected Range of Incident Duration

Since the prototype system is designed to operate on a rolling horizon basis, the incident duration feature of a case represents the time, within the prediction horizon considered, during which the incident is still in existence. Since it is usually impossible to estimate incident duration precisely, incident durations were rounded up to the nearest 5 minutes in the current study. Now, as previously mentioned, when adopting a rolling horizon approach, real-time execution is crucial for just the first stage. For subsequent stages, the system is only required to solve the problem in a time period that is short enough to allow the solution for the upcoming stage to reflect actual demand and network configuration (typically around 5 minutes). This quasi real-time execution requirement means that one has more time for adaptation when solving the subsequent stages of a rolling-horizon DTA problem, which in turn means that the quality of the case retrieved by the CBR component need not be as good as that retrieved for the first stage.

Given this fact, and since a prediction horizon of 15 minutes is being considered, the research team proposed to develop five case-bases to cover the expected range of incident duration (Figure 11). The first case-base, case-base I, is developed for the first stage, where real-time execution is crucial. This case-base considers an incident duration of 15 minutes, or the full

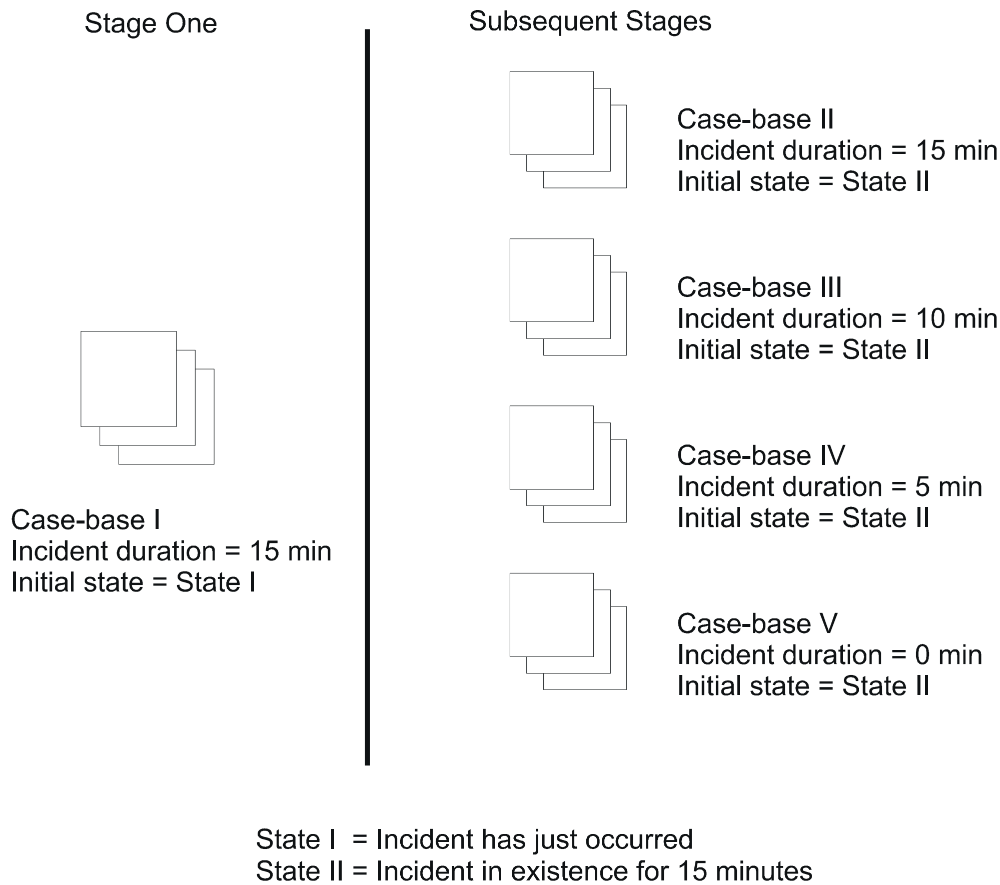

Figure 11. Case-bases to Cover Expected Range of Incident Duration

length of the prediction horizon (for the first stage, only a duration of 15 minutes is considered, since incidents whose duration is less than 15 minutes will typically not warrant region-wide routing). For subsequent stages, case-bases II, III, IV, and V are developed for incidents of duration 15, 10, 5, and 0 minutes, respectively. For case-bases II, III, IV, and V, the initial state is one where an incident has already been in existence for 15 minutes. The initial state for case-bases II, III, IV, and V is thus different from the initial state for the case-base I, where an incident has just occurred. This is the reason for having two case-bases, case-bases I and II, covering an incident duration of 15 minutes.

Now, with these five case-bases, the system is equipped with cases specifically developed for incidents of the following duration: 15, 20, 25, and 30 minutes. For incidents with a duration greater than 30 minutes, the system will combine cases from these five case-bases to cover the required duration. So, for example, to address the case of a 35-minute incident, the system will use the following case-bases (Figure 12):

1. For stage 1, where the incident persists for the entire length of the prediction horizon, the system will use case-base I to find a routing strategy for the first 15 minutes.

2. For stage 2, where the incident persists for the entire length of the prediction horizon, the system will use case-base II to find a routing strategy for the next 15 minutes.

3. For stage 3, where the incident exists for only the first 5 minutes of the prediction horizon, the system will use case-base IV (which considers an incident duration of 5 minutes) to find a routing strategy for the third 15-minute interval.

4. For stage 4, where no incident exists, the system will use case-base V (which considers an incident duration of 0 minutes). The purpose of this strategy is to help flow return to normal.

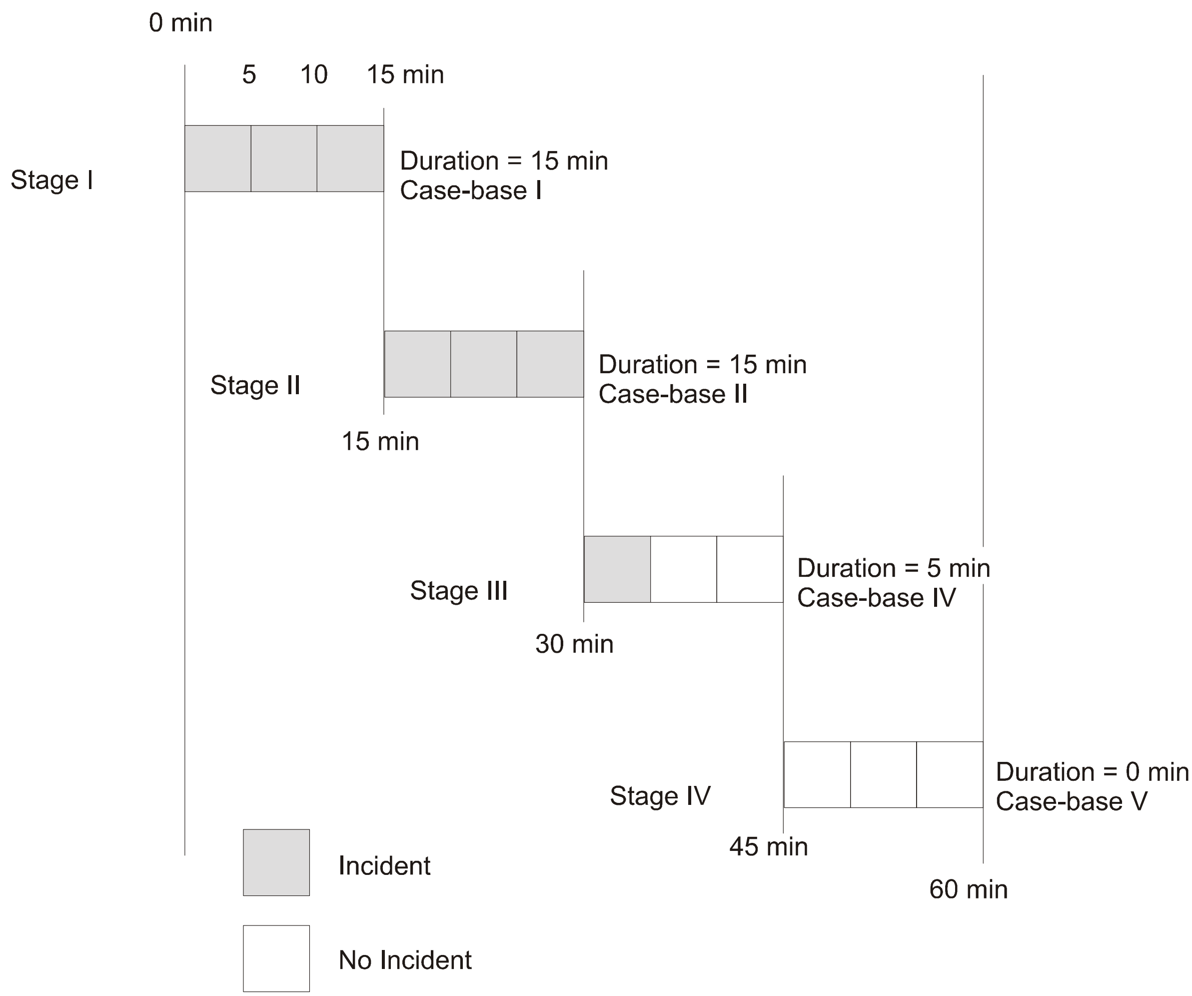

Figure 12. Case-bases Used for a 35-Minute Incident

For stages 1 and 2, the initial state for the cases stored in the case-bases is similar to the initial state for the problem being considered. This is, however, not the case for either stage 3 or 4. This means that the distribution of traffic on particular links might be different. However, since one has more time to adapt for subsequent stages, the research team conjectured that the cases retrieved in this fashion would still provide for a good starting point for the adaptation module. The evaluation of this proposed scheme is addressed later.

Case Storage Requirements

The number of cases needed for this design is 63,360 cases. Storing this number of cases requires only about 17.5 MB of storage space, which is reasonable.

Indexing and Retrieval

As discussed previously, the prototype system has five case-bases, each developed to cover a particular incident duration. To facilitate retrieval and speed up the search process within each case-base, an indexing scheme was developed. According to this scheme, cases are indexed based on the values for the incident location feature and the incident severity level (Figure 13). As can be seen, under each combination of incident location and severity level, the 288 cases, corresponding to the different traffic patterns, are stored.

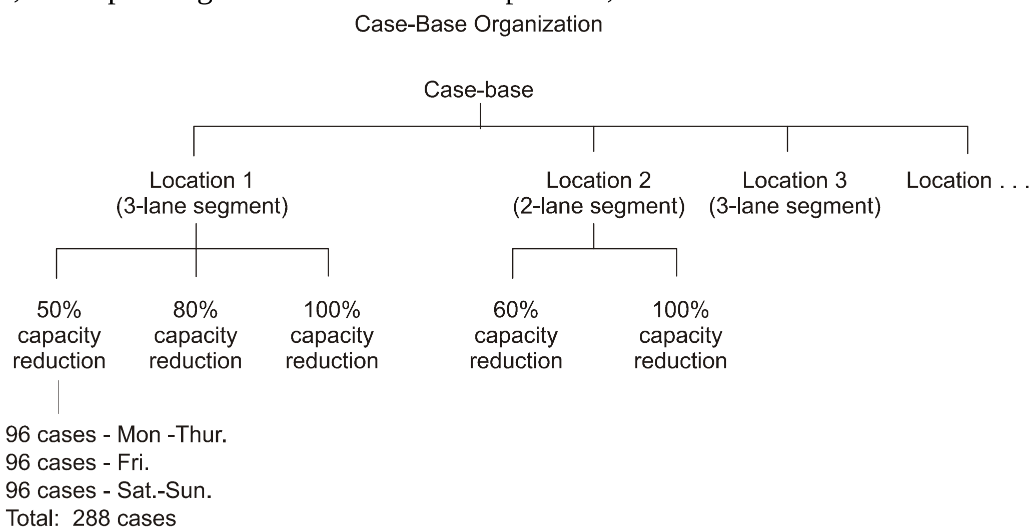

Figure 13. Case-base Organization

After the branch of the tree is determined, a nearest neighbor (NN) algorithm is used to retrieve the most similar case (i.e., least distant) of the cases stored under that branch. The NN algorithm uses the 86 features for the current and predicted traffic volumes on the 43 links of the network. The distance between a query *q* and a case , in case-base *L* was defined as:

$$distance\ (x,q) = \sqrt{\sum_{f=1}^{n} w_f\ x\ difference(x_f, q_f)^2} \qquad [2]$$

where

w_f , the parameterized weight value assigned to feature *f*, was assumed to be equal to 1.0

the difference (x_f, q_f) is equal to $|x_f - q_f|$.

The traffic volumes on the links were normalized by subtracting their mean and dividing by their standard deviation to ensure they have the same range and hence the expected impact.

Adaptation

As previously mentioned, there are two modes of operation for the adaptation module: simple and elaborate. For the prototype system, simple adaptation is used when the capacity reduction value of the new problem differs from the capacity reduction values of the cases stored within the case-base. To address this, the system uses linear interpolation to arrive at the recommended routing strategy. For example, for the case of a 70 percent reduction in capacity, the routing strategy is derived as a weighted average of the strategy for an 80 percent reduction and that for a 50 percent reduction.

Elaborate adaptation, on the other hand, involves running the SA-DTA model, developed in the first phase of the project, using the case retrieved by the NN algorithm as the start point. Since the start-point should be close to the "optimum," the system searches for any better solutions only in the vicinity of the retrieved solution (local search); no time is wasted exploring other portions of the solution space.

By operating using a rolling horizon approach, the DSS will solve the problem in a number of stages. For the first stage of this rolling horizon approach, real-time execution is crucial, whereas for subsequent stages, quasi real-time execution should suffice. Now, for the first stage, the initial state for the cases stored in the system's case-base will always be identical with the initial state of the current problem faced (both refer to the state where an incident has just occurred). Given this, and the fact that real-time execution is required, only simple adaptation will be used for this stage. For the subsequent stages of the rolling horizon approach, the initial state for the cases might be different from the initial state of the problem under consideration. However, for these subsequent stages, one has more time for adaptation, since only quasi real-time performance is required. Consequently, elaborate adaptation will be used for these stages.

Revise and Retainment

The proposed routing DSS will learn mainly through the acquisition of new cases from the real-world environment, along with their corresponding performance measures. As a recommended routing strategy is implemented, the ATMS surveillance system will continue to monitor the status of traffic flow, and will record the value of one or more performance measure(s) (the total vehicles travel time or the average speed, for example) resulting from the recommended strategy. The value of the same performance measure(s) in case routing has not been implemented (i.e., in case drivers were left to follow the routes they typically use) is estimated. This can be done using the evaluation component of the SA-DTA or a more detailed simulation model, if desired. By comparing the values of the performance measure with the routing strategy implemented to its value for the case of no routing, a measure of the desirability of the specific routing strategy can be derived. This desirability index will then be stored, along with the case itself, in the system's case-base, and will act as a decision factor to determine whether the strategy is a success or a failure.

If the new case is a success, it is retained by the system to speedup the solution procedure when faced with a similar situation in the future. On the other hand, if the case is a failure (i.e., showing a low desirability index), attempts should be made to understand the reasons that led to its failure, and to develop a more successful scenario to that particular problem. This repair process will typically require human intervention, and may even involve a modification of the evaluation component of the DTA module to more closely mimic real-world conditions. The new solution should then be stored in the case-base. In this manner, the DSS would be able to find a successful solution when faced with a similar problem in the future.

Developing and Evaluating the Prototype System

In this stage, the performance of the prototype routing DSS is evaluated. Since there is a difference between the performance required of the system for solving the first stage of the rolling horizon DTA problem and that required for solving subsequent stages, the system evaluation on these two problems is addressed separately.

Evaluating Performance on Stage 1 of the Rolling Horizon DTA Problem

Prototype System Development

Case-base I, whose initial state describes a state where an incident has just occurred, considers an incident duration of 15 minutes. To reduce the time required for developing the prototype system case-base, a partial case-base for only the right side of the network (Figure 5) was constructed. Table 3 shows the range of values for the cases' features considered while developing the case-base.

Table 3. Range of Values for Case-base I

Feature (or group of features)	Range of Values
Traffic patterns	Group 1 (Monday through Thursday)
Incident location	21 incident locations (Links 1 through 21)
Incident severity	3 severity levels for 3-lane segments 2 severity levels for 2-lane segments
Incident duration	15 min

The number of cases required for building this partial case-base was about 2,400 cases. The cases were generated as follows:

1. Short-term traffic counts from the two tunnels covering the months of March, April, and May 1995 were obtained. The average traffic count for each 15-minute interval of the day was computed (by averaging the corresponding 15-minute traffic counts for all the Mondays, Tuesdays, Wednesdays, and Thursdays of the 3-month period).

2. For each 15-minute interval, the corresponding 15-minute traffic counts from the two tunnels were used to scale the initial state matrix. The counts from the next 15-

minute interval were then used to scale the predicted travel demand matrix. As explained previously, a noise term was added to each cell in the travel demand matrix to replicate real-world conditions and their randomness more closely.

3. An incident scenario was generated by randomly selecting a link on which an incident was assumed to have taken place and randomly selecting the number of lanes blocked.

4. The SA-DTA model was then used to find the “optimal” routing strategy for each case.

5. To represent the cases according to the structure defined, the initial volume on each link of the network was derived from the initial state matrix by summing the cell values in the row of the destined traffic density matrix corresponding to the link considered. The predicted volumes for the upcoming 15 minutes were obtained from an assignment of the O-D matrix using the typical traffic splits.

Compiling this case-base of 2,400 cases required around 748 hours of CPU on a Pentium 200 MHz PC. The 96 cases corresponding to each incident scenario were each stored in a separate text file.

Prototype System Evaluation

To evaluate the performance of the system, a test-set consisting of 100 new cases was randomly generated in the following fashion:

1. A time within the 3-month period was randomly selected. The current and upcoming 15-minute traffic counts from the two tunnel facilities for the randomly selected time were then used to scale the initial state and the O-D matrices with a random noise term added to the matrix entries to more closely resemble real-world conditions.

2. For an incident location, a random number was drawn from the set {1,2,3, . . . ,21} to represent an incident on one of the 21 links considered in building the case-base.

3. For the incident severity, the capacity remaining values were selected as follows: A number was randomly drawn from the set {1,2,3} for three-lane segments and from the set {1,2} for two-lane segments to represent the number of lanes blocked. The value of the capacity remaining was then randomly selected from the ranges shown in Table 4.

4. Since this experiment was intended to evaluate the system’s performance on the first stage of the DTA problem, an incident duration of 15 minutes was assumed for all 100 cases.

5. To represent the cases according to the previously defined structure, the current and predicted volumes on the network links were deduced from the initial state and O-D matrices as previously explained.

Table 4. Capacity Remaining Under Different Incident Scenarios

Lanes Blocked	3-lane Segments	2-lane Segments
1	[40,41, . . . ,50]	[30,31, . . . ,40]
2	[10,11, . . . ,30]	[0,1, . . . ,10]
3	[0,1, . . . ,10]	

For each case of these 100 newly generated cases, the prototype CBR system was used to determine a recommended routing strategy. Since as previously discussed, real-time execution was required for the first stage of the DTA, the study initially focused on running the CBR system using only the simple adaptation mode. Under this simple mode of operation, solving a problem merely involved retrieving the most similar case(s), and applying linear interpolation whenever needed. The same case was then solved from scratch, using the SA-DTA model, to determine the "optimal" strategy. The quality of the routing strategy determined using the prototype CBR system was compared to the quality of the SA-DTA model solution. This was done by comparing the total vehicles travel time resulting from implementing the CBR system's recommended strategy to that resulting from the DTA model recommended strategies.

Evaluating Performance on Subsequent Stages of DTA Problem

Prototype System Development

As previously discussed, the current study proposes developing four case-bases, covering incidents of duration 15, 10, 5, and 0 minutes, for use when solving the subsequent stages of the rolling horizon DTA problem. The initial state for all four case-bases is one where an incident has been in existence for 15 minutes. To cut down the time required for developing and evaluating the prototype's performance, four partial case-bases were developed for a subset of the range of values that the different features of a case can assume. Table 5 shows the range of values considered in developing the four case-bases. Cases were generated according to the five-step procedure described in the previous section.

Table 5. Range of Values for Case-bases II, III, IV, and V

Feature (or group of features)	Range of Values
Traffic patterns	Group 1 (Monday through Thursday)
Incident location	Links 10
Incident severity	Three severity levels
Incident duration	15 min for case-base II, 10 min for case-base III, 5 min for case-base IV, 0 min for case-base V

Prototype System Evaluation

Twenty-five new cases were generated. However, in this case, the incident location was fixed (i.e., link 10) and the incident duration was allowed to vary between 15 and 60 minutes in units of 5 minutes (e.g., 25, 40, . . ., etc.). The number of stages needed to solve each problem thus varies based upon the duration of the incident. Consequently, to generate the predicted O-D matrices for the stages of a particular problem, successive 15-minute counts from the tunnels were used for scaling the daily values. For example, the initial and predicted traffic demands for the stages of a case with an incident that starts at 8:15 A.M. on a particular day and lasts for 40 minutes were generated as follows.

1. For stage 1, the initial state was generated by scaling the destined traffic volume matrix using the 8:15 A.M. traffic counts (traffic volume values were then converted into the corresponding density values as previously explained). Predicted demand for the upcoming 15 minutes was generated by scaling the O-D matrix using the 8:30 A.M. counts.

2. For stage 2, the initial state was the state resulting after implementing the recommended routing strategy for stage 1. This state was obtained from the output of the SA-DTA model. The predicted demand was then generated using the 8:45 A.M. counts.

3. Step 2 was repeated for stages 3 and 4.

Table 6 shows the incident duration for the 25 cases constituting the test problem set, along with the number of stages needed to solve each problem. The table also shows the case-bases utilized in solving the different stages of each problem.

The prototype system was then used to determine a routing strategy for the different stages of the 25 cases comprising the test set. Since, the system has more time to adapt when solving the DTA problem for stages other than the first stage, the adaptation module was run for a maximum of 5 minutes. The same cases were then solved using the SA-DTA model. The quality of the CBR system's routing strategies were then compared to the quality of the SA-DTA model's strategies.

Assessing Anticipated Time Savings from System Implementation

The partial case-base developed for building the prototype system was analyzed so as to obtain a measure of the expected time savings resulting from implementing the system. This was done by comparing the total vehicles travel time under the system's recommended routing strategy, to the total time that would have resulted if drivers were left to take the routes they typically use (i.e., no routing strategy was to be implemented). The total travel time for this latter case of no routing (base case) was determined using the typical traffic splits obtained from static assignment. It is important to note that this methodology provides a best-case estimate of time savings when the routing strategy is fully "followed" by the traveling public.

Table 6. Incident Duration for Test Problem Set

Case No.	Duration	No. of Stages	Case-bases Used
1	45	4	I, II, II, V
2	45	4	I, II, II, V
3	55	5	I, II, II, III, V
4	20	3	I, IV, V
5	25	3	I, III, V
6	50	5	I, II, II, IV, V
7	15	2	I, V
8	60	5	I, II, II, II, V
9	60	5	I, II, II, II, V
10	40	4	I, II, III, V
11	60	5	I, II, II, II, V
12	25	3	I, III, V
13	45	4	I, II, II, V
14	30	3	I, II, V
15	60	5	I, II, II, II, V
16	45	4	I, II, II, V
17	50	5	I, II, II, IV, V
18	60	5	I, II, II, II, V
19	15	2	I, V
20	60	5	I, II, II, II, V
21	30	3	I, II, V
22	30	3	I, II, V
23	55	5	I, II, II, III, V
24	15	2	I, V
25	50	5	I, II, II, IV, V

RESULTS AND DISCUSSION

System Evaluation for Stage 1 of DTA Problem

Performance with the Elaborate Adaptation Module Deactivated

Each of the 100 newly generated cases was solved using the prototype CBR system (employing only the simple adaptation module and the SA-DTA model. From an execution time standpoint, the CBR approach resulted in significant time savings. The average solution time for the SA-DTA model was 18.8 minutes. The CBR system needed less than 0.02 minutes and is therefore quite capable of functioning in real time.

To compare the quality of the CBR approach's solutions to the SA-DTA model's solutions, the percentage difference in the cost (i.e., total vehicles travel time) between the two solutions was computed. Figure 14 depicts a histogram for the distribution of the percentage difference in cost for the 100 cases.

As can be seen, the performance of the system seems to be quite adequate, especially given the fact that only simple adaptation was used. For 98 of 100 cases, the percentage

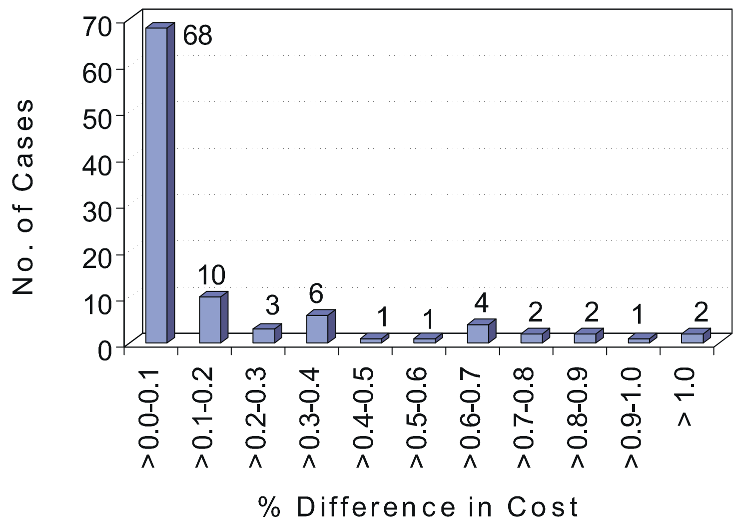

Figure 14. Histogram for Percentage Difference in Cost

difference in cost is less than 1 percent. Of these 98 cases, 68 cases actually had a percentage difference of less than 0.1 percent, and 10 cases had a difference between 0.1 and 0.2 percent. Only two cases had a percentage difference greater than 1 percent: case 42 (6.25%) and case 52 (3.36%).

Performance with the Elaborate Adaptation Module Activated

Although real time execution is required for the first stage of the DTA problem, it might still be possible to run elaborate adaptation module for a short time period that still satisfies the real-time execution requirement. Since the study was using only a Pentium 200 MHz PC, the study considered this short time period to be equal to 60 seconds of run time on that PC. Consequently, for each of the 100 cases, the elaborate adaptation module was run, with the case returned by the simple adaptation as a start point, for 60 seconds, and the quality of the solution recorded every 10 seconds of execution. Figures 15 and 16 show the distribution of the percentage difference in cost between the CBR solution and the SA-DTA after 30 seconds and 60 seconds of running the elaborate adaptation module.

As can be seen, running the adaptation module for even very short time intervals results in an improvement in the quality of the solutions returned by the system. After 30 seconds of adaptation, 77 cases had a percentage difference of less than 0.1 percent and 7 cases had a difference between 0.1 and 0.2 percent. On the other hand, after 60 seconds of adaptation, 82 cases had a percentage difference of less than 0.1 percent and 6 cases had a difference between 0.1 and 0.2 percent (when merely using simple adaptation, only 68 cases had a percentage difference of less than 0.1 percent and 10 cases had a difference between 0.1 and 0.2 percent). There was also an improvement in the percentage difference in cost for the greater than 1.0 percent group (cases 42 and 52). For case 42, the difference dropped from 6.25 percent down to 2.48 percent after 30 seconds, and down to 0.13 percent after 60 seconds. For case 52, the

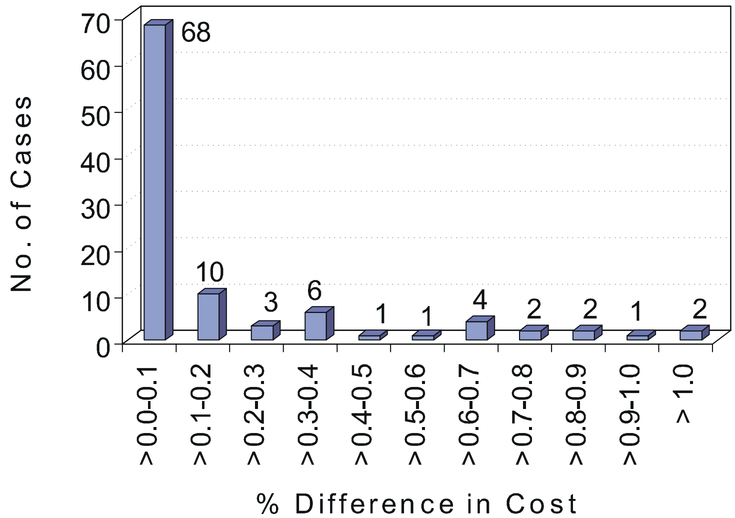

Figure 15. Percentage Difference Histogram After 30-Second Adaptation

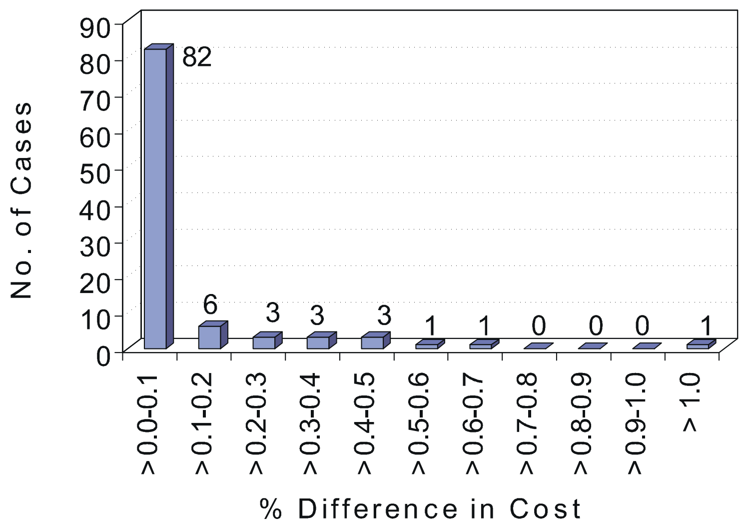

Figure 16. Percentage Difference Histogram After 60-Second Adaptation

difference dropped slightly from 3.36 percent to 3.35 percent after 30 seconds, and to 2.45 percent after 60 seconds.

Significance of Difference in Cost

To assess the significance of the difference between the cost of the CBR and the SA-DTA model solutions better, the range of values for all feasible solutions was determined for each of the 100 cases. This was done by modifying the search algorithm so as to search for the solution giving the maximum value for the objective function (i.e., the worst routing strategy). The difference in cost between the CBR solution (obtained after 60 seconds of adaptation) and SA-DTA solution was then expressed as a percentage of the feasible solution range.

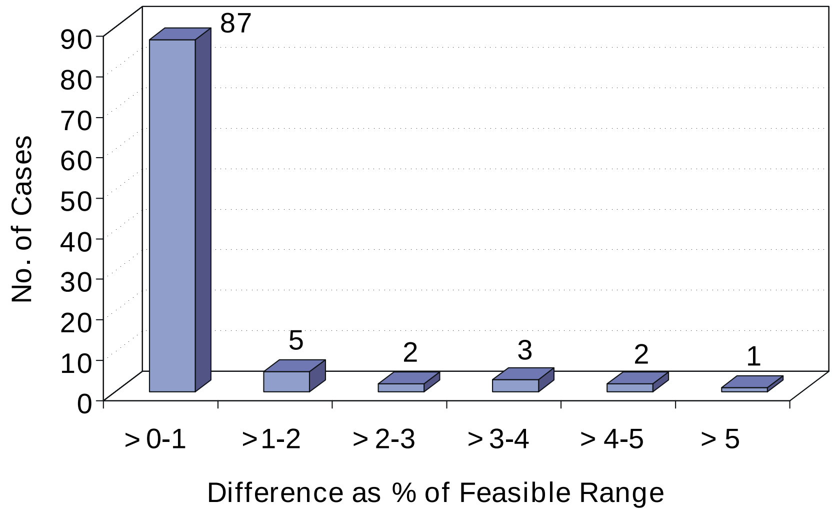

Figure 17. Histogram for Difference as Percentage of Feasible Range

Figure 17 is a histogram showing the distribution of the difference in cost expressed as a percentage of the feasible range. As can be seen, the difference was less than 1 percent of the feasible range for 87 cases. Only one case (case 52) had a percentage difference greater than 5 percent (namely, 7.12%). This shows that the quality of a CBR solution is comparable to that of the SA-DTA model. The performance of the proposed CBR routing DSS thus seems to be quite satisfactory.

The study then subjected the system to a more stringent evaluation process to assess the significance of differences between the CBR and the SA-DTA solution. In this process, the study compared the time savings resulting from implementing the CBR system's recommended routing strategy to savings resulting from the SA-DTA model's strategy. To do this, it was first necessary to calculate the total vehicle travel time for the instance where no routing strategy had been implemented (i.e., drivers were left to take the routes they typically use). The total vehicle travel time for this instance, referred to as the base condition travel time, was computed using the drivers' typical traffic splits. For each case, a score, referred to as the system's success score, was computed. The success score is given by:

$$Success\ score = \frac{(Base\ condition\ trav\_time\ -\ CBR\ strategy's\ trav\_time)}{(Base\ condition\ trav\_time\ -\ SA-TA\ \ strategy's\ trav\_time)} \times 100 \qquad [3]$$

Figure 18 is a histogram for the distribution of the success score. As can be seen, for 86 cases, the success score was between 90 and 100 percent. Only one case had a success score of less than 50 percent (namely, 47%). This shows that the CBR approach is capable of achieving time savings that are comparable to the SA-DTA's model approach. Moreover, the CBR approach does that in a fraction of the time needed by the SA-DTA model, and hence is capable of satisfying the execution time constraints imposed by the problem.

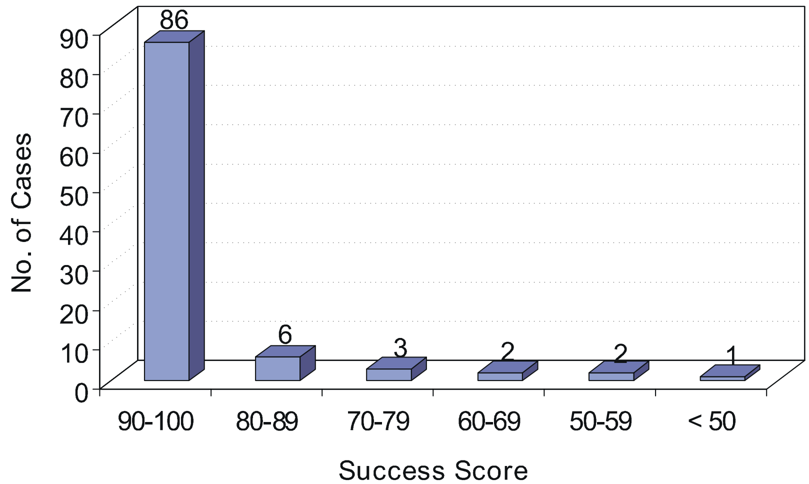

Figure 18. System's Success Score Distribution

System Evaluation for Subsequent Stages of the DTA Problem

Figure 19 shows the distribution of the percentage difference in cost between the CBR system and the SA-DTA model solutions for stage II. As can be seen, the difference in cost was less than 0.1 percent for 24 of the 25 cases, indicating that the quality of the solutions produced by the CBR system and the SA-DTA model is comparable. The results for stages III, IV, and V were comparable to the results for stage II. For stages III and IV, all cases had a percentage difference of less than 0.1 percent, and for stage V, 9 of the 11 cases had a percentage difference of less than 0.1 percent.

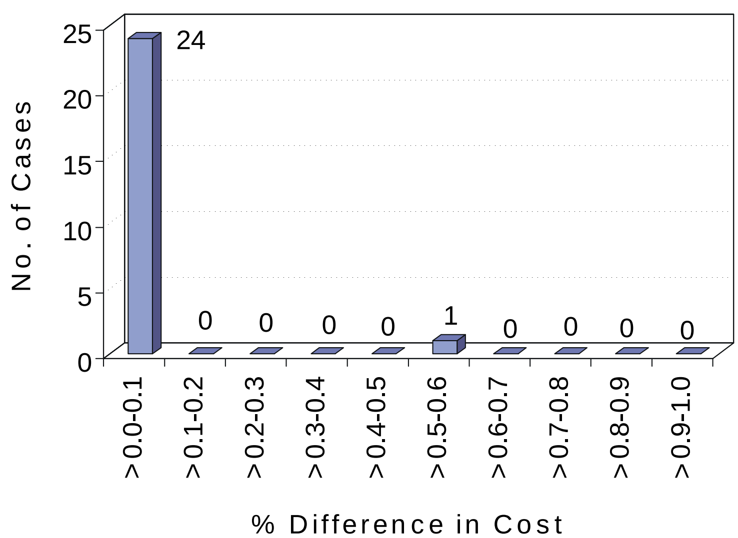

Figure 19. Percentage Difference Histogram for Stage 2

Although the CBR approach was capable of producing solutions of comparable quality to the SA-DTA model approach, it ran much faster (see Table 7).

Table 7. Average Execution Time for SA-DTA and CBR Models (min)

Stage	SA-DTA	CBR
II	20.68	3.62
III	21.91	3.41
IV	23.05	4.15
V	23.26	4.97

Anticipated Time Savings from System Implementation

Table 8 compares the time savings using different routing strategies. The times are for the cases where incidents are assumed to have occurred during the evening peak 15-minute period. As can be seen, the time savings range from 944.7 to 9515.0 vehicle.minutes/15 minutes, with an average of 4358.3 vehicle.minutes/15 minutes.

Table 8. Time Savings of Routing Strategies

Incident Location	Capacity Reduction (%)	Base Case Trav_time (no routing)	Trav_time (with routing)	Time Savings (veh.period/15 min)	Time Savings (veh.min/15 min)	% Time Saved
Link 2	50	99706	98417.5	1288.5	1073.8	1.3
	80	126803	118474	8329	6940.8	6.6
	100	129317	121004	8313	6927.5	6.4
Link 4	60	126401	117313	9088	7573.3	7.2
	100	167871	158938	8933	7444.2	5.3
Link 7	50	102319	99402.4	2916.6	2430.5	2.9
	80	137159	127663	9496	7913.3	6.9
	100	158580	151030	7550	6291.7	4.8
Link 10	50	118475	107057	11418	9515.0	9.6
	80	151517	141924	9593	7994.2	6.3
	100	163919	155324	8595	7162.5	5.2
Link 11	50	132116	125486	6630	5525.0	5.0
	80	155846	148599	7247	6039.2	4.7
	100	168777	163520	5257	4380.8	3.1
Link 13	50	95655.7	94407.5	1248.2	1040.2	1.3
	80	97009.8	95475.1	1534.7	1278.9	1.6
	100	100757	99251.1	1505.9	1254.9	1.5
Link 15	60	95655.7	94522.1	1133.6	944.7	1.2
	100	106639	105205	1434	1195.0	1.3
Link 19	50	100513	97934.3	2578.7	2148.9	2.6
	80	119805	114992	4813	4010.8	4.0
	100	129513	124714	4799	3999.2	3.7
Link 21	50	96360.5	94426.7	1933.8	1611.5	2.0
	80	106337	103578	2759	2299.2	2.6
	100	112862	110507	2355	1962.5	2.1

To give a crude estimate of the potential annual dollar savings resulting from the system, the research team assumed an incident frequency of one incident/day and an average duration of incident-related congestion of 1 hour per incident. According to Chui et al.,[17] a reasonable estimate for the value of time (in 1985) is $9.75/hour.persons. Assuming a vehicle occupancy of 1.3 and inflating the 1985 estimate to 1997 dollars (assuming a 3% inflation rate), the research team obtained a time value of $17/vehicle hours. Based on these estimates, VDOT could expect to provide $1.8 million per year in user cost savings.

CONCLUSIONS

- A CBR DSS for real-time traffic routing has the potential to overcome many of the limitations of existing approaches to the problem.
- Owing to the recurrent nature of traffic, a CBR system can produce high-quality solutions with reasonable-size case-bases.
- From an execution-time standpoint, a CBR routing system is quite capable of functioning in real time.
- By using the link volumes to describe the cases, a CBR system eliminates the need to perform the O-D matrix estimation step on-line, resulting in additional savings in execution time.
- Investments in TMS infrastructure are warranted, since these systems have the potential to result in substantial annual motorist time/cost savings.

RECOMMENDATIONS

1. *Provide TMSs with the capability for the easy archival and retrieval of historical traffic data.* This can be achieved by including the following items in TMS functional requirements: SQL-compliant database management system, capability to archive traffic flow data to a non-volatile medium, and hardware with a high storage capacity.
2. *Continue to add traveler information devices and services in major urban areas.* These devices are particularly needed in advance of major diversion interchanges to support the provision of routing information.

SUGGESTIONS FOR FURTHER RESEARCH

- *Since the retrieval algorithm used by the current study assigns equal weights to all features, attempt to determine appropriate values for the weights in a fashion that assigns higher* *weights to the more relevant features.* Methods exist that can automatically perform this task using little or no domain-specific knowledge.[18] One approach that seems most promising is to use genetic algorithms to update feature weights using randomly selected training cases. The developed match/retrieve module uses a single nearest neighbor algorithm that retrieves only the most similar case. There are advantages to exploring the use of *k* nearest neighbor algorithms to retrieve the *k* most similar cases, and combine the solutions of these *k* cases to arrive at the recommended strategy.

- *Once the Suffolk ATMS is completed, determine how devices such as VMS could be used to affect the route selection of motorists so as to achieve the required traffic splits.* The results from such a study should then be used in formulating a set of "information" strategies that can be used to achieve the desired diversion levels. These strategies could then be stored within the system's case-bases.

- *Once the Suffolk ATMS is completed, implement the learning module.* Then, a study could be undertaken to evaluate the performance of the system under real traffic conditions.

REFERENCES

1. A.W. Sadek, B.L. Smith, C.C. McGhee, and M.J. Demetsky. (1998). *Development of Decision Support Systems for Real-Time Freeway Traffic Routing, Volume I.* Virginia Transportation Research Council, Charlottesville.

2. A. Aamodt and E. Plaza. (1994). *Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches*. AI Communications, 7(I), pp. 39-59.

3. J. L. Kolodner. (1993). *Case-Based Reasoning.* Morgan Kaufmann.

4. I. Watson. (1995). The Case for Case-Based Reasoning. In *Proceedings of the First United Kingdom Workshop*. Springer-Verlag, Salford, U.K., pp. 1-16.

5. J. R. Quinlam. (1986). Induction of Decision Trees. *Machine Learning,* 1(1): 81-106.

6. I. Juriscia. (1993). *Representation and Management Issues for Case-Based Reasoning Systems.* Department of Computer Science, University of Toronto, Ontario, Canada.

7. M.M. Veloso and J.G. Carbonell. (1993). Derivational analogy in PRODIGY: Automating Case Acquisition, Storage and Utilization. *Machine Learning,* 10(3): 249-278.

8. R. Schank. (1989). *Inside Case-Based Reasoning.*

9. K.J. Hammond. (1989). *Case-Based Planning: Viewing Planning as a Memory Task.* Academic Press, Boston.

10. S.J. Bayles and B.K. Das. (1994). Using Artificial Intelligence to Support Traffic Flow Management Problem Resolution. *Case-Based Reasoning: Papers from the 1994 AAAI Workshop*, Seattle, Washington, pp. 133-137.

11. A. Khattak and A.K. Kanafani. Case-Based Reasoning: A Planning Tool for Intelligent Transportation Systems. *Transportation Research 4C*, Vol. 5, pp. 267-288.

12. I.P. Mulholland and R.A. Oren. Artificial Intelligence for Transit Railcar Diagnosis. *Transit Cooperative Research Program Report 1*.

13. S. Peeta and H. Mahmassani. (1995). Multiple User Classes Real-Time Traffic Assignment for Online Operations: A Rolling Horizon Solution Framework. *Transportation Research 3C(2)*, pp. 83-98.

14. B.L. Smith and M.J. Demetsky. (1994). Short-Term Traffic Flow Prediction: Neural Network Approach. *Transportation Research Record 1453*, Transportation Research Board, Washington, D.C., pp. 98-104.

15. B.L. Smith and M.J. Demetsky. (1996). Multiple-interval freeway traffic flow forecasting. *Transportation Research Record 1554*, Transportation Research Board, Washington, D.C., pp. 136-141.

16. S. M. Madanat, S. Hu, and J.V. Krogmeier. (1996). Dynamic Estimation and Prediction of Freeway Origin-Destination Matrices with Route Switching Consideration and Time Dependent Model Parameters. *Transportation Research Record 1537*, Transportation Research Board, Washington, D.C., pp. 98-105.

17. M.K. Chui, and W.F. McFarland. (1986). *The Value of Travel Time: New Estimates Developed Using a Speed-Choice Model.* Research Report 396-2F. Texas Transportation Institute, Texas A & M University, College Station.

18. D. Wettschereck and D.W. Aha. Weighting Features. In Proceedings of the *First International Conference on Case-Based Reasoning.* Springer-Verlag, Sesimbra, Portugal, pp. 359-370.