

**Anticipatory Route Guidance
Using Rolling Horizon Procedures**

Daniel J. Reaume

Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 96-9

July 1996

ANTICIPATORY ROUTE GUIDANCE USING ROLLING HORIZON PROCEDURES

Daniel J. Reaume

Abstract

The SAVaNT optimization package has been developed to determine user-optimal routings for vehicles in a traffic network. To facilitate practical implementation of SAVaNT, we have re-implemented it in a rolling horizon framework. We present the modeling difficulties involved in this effort and examine algorithm performance when applied to the Troy, Michigan traffic network.

* This work was supported by the University of Michigan Research Center of Excellence in Intelligent Transportation Systems.

1. INTRODUCTION

SAVANT

One of the goals of the ITS (Intelligent Transportation Systems) initiative is to reduce travel delays experienced due to congestion by choosing a more appropriate routing for vehicles. Traditionally, due to the difficulty of the problem, algorithms determining routing decisions have not been anticipatory, ignoring congestion effects caused by re-routing vehicles in order to avoid existing traffic congestion.

The SAVaNT (Simulation of Anticipatory Network Vehicle Traffic) route optimization system [1] is an anticipatory traffic optimization algorithm which has been mathematically proven to generate user-optimal vehicle routings. By user-optimal, we mean that no vehicle may reduce its experienced trip time by unilaterally changing its route. This is as opposed to a system-optimal routing, which minimizes the total travel time of all vehicles on the network. Since it is difficult to measure the level of user-optimality of a routing, we will use average trip time as our performance measure for SAVaNT throughout this paper, even though this measure relates to system optimality.

As mentioned above, not only does the choice of a non-anticipatory optimal routing depend on the current time-dependent link travel time profiles, but by re-routing vehicles to avoid congestion, this routing also modifies this link travel time profile. The routing, optimal for the original set of link travel times is therefore in general no longer optimal for the modified link travel times. SAVaNT is an iterative procedure which proceeds to find a fixed point in this feedback loop such that link travel times yielding a routing remain unchanged if this routing is implemented.

SAVaNT proceeds as illustrated in Figure 1. From an initial estimate of link travel times, a routing program computes a non-anticipatory optimal routing based on the assumption that these travel times will hold regardless of routing computed. A traffic simulator then applies this routing to determine the time-dependent link travel times it yields. The new set of link travel times are then input back into the routing program and the process iterates. If, at any time, two consecutive iterations yield identical link travel time profiles, then a user-optimal routing has been computed [1].

SAVaNT-CNV

Unfortunately, it is quite possible that the link travel time profiles from two non-consecutive iterations of SAVaNT may be identical. In such a situation, which we call a cycle, it is clearly impossible to ever generate a fixed point. Wunderlich [2],[3] has developed a method of avoiding such cycles by deliberately introducing error into link travel time profiles reported by the simulation program. The resulting algorithm is called SAVaNT-CNV since it ensures convergence to a fixed point.

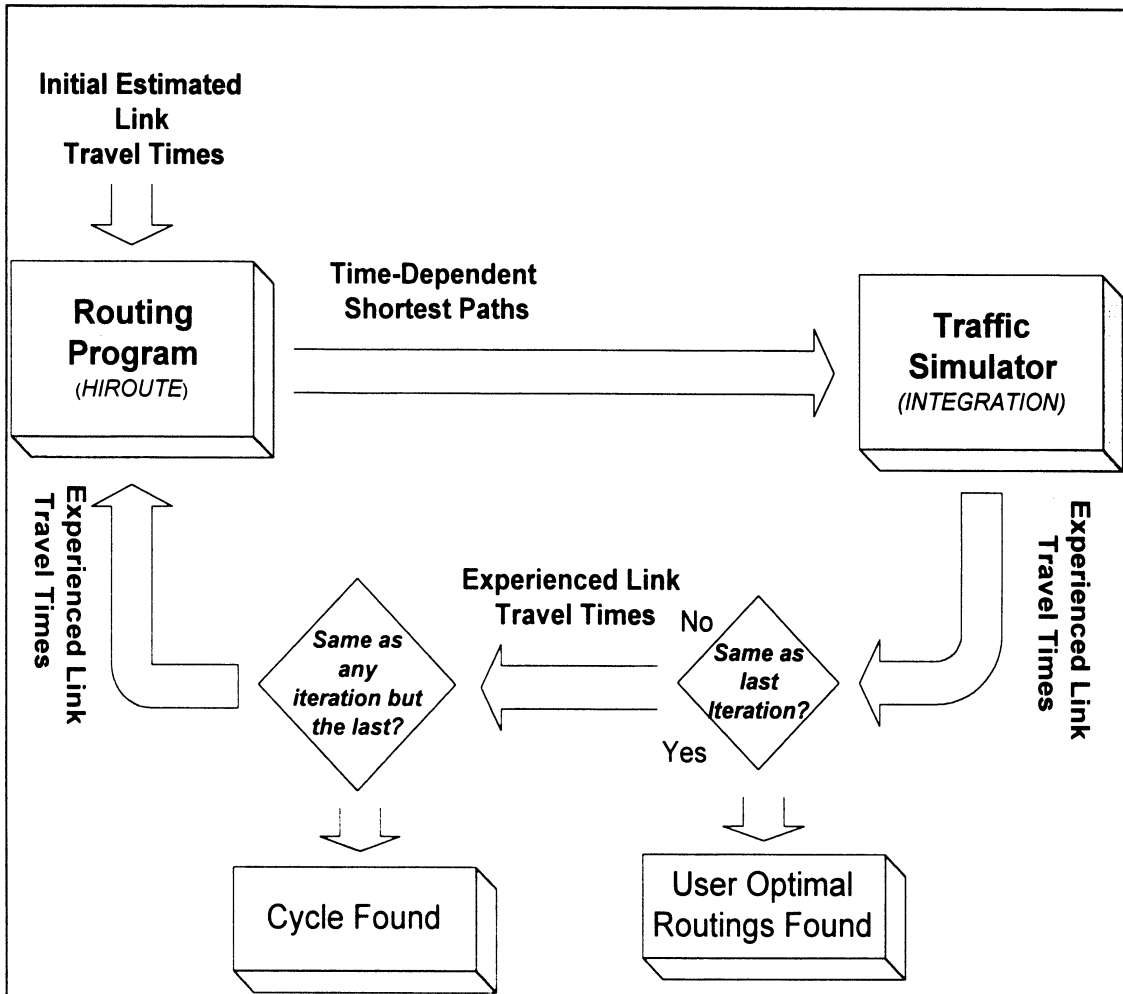


Figure 1: The SAVaNT optimization system

At the heart of SAVaNT-CNV is a parameter called δ which regulates the degree of inaccuracy of reported link travel times. If a vehicle enters link l at time t and requires time $e(l,t)$ to traverse the link, then under SAVaNT-CNV, it will be reported that vehicles entering l at time $t + (\delta)e(l,t)$ will traverse the link in time $e(l,t)$. As illustrated in Figure 2, a δ value of 0.0 corresponds to accurate information. The more δ is increased, the more inaccurate are reported link travel times. When δ is 1.0, we have a situation called probe accounting, where if a vehicle leaves a link at time t , then its link travel time will be reported as if it had entered the link at time t .

SAVaNT-CNV proceeds identically to SAVaNT until a cycle is detected. Then, as illustrated in Figure 3, delta is incremented and the traditional SAVaNT loop restarts, but with more inaccurate link travel time information. In the course of the optimization process, delta may be increased several times, but we are guaranteed that eventually SAVaNT-CNV will converge. This is since, for delta large enough, all vehicles will have their effective link entry times “shifted” past the end of the simulation, hence only free-flow link travel times are reported to the routing program.

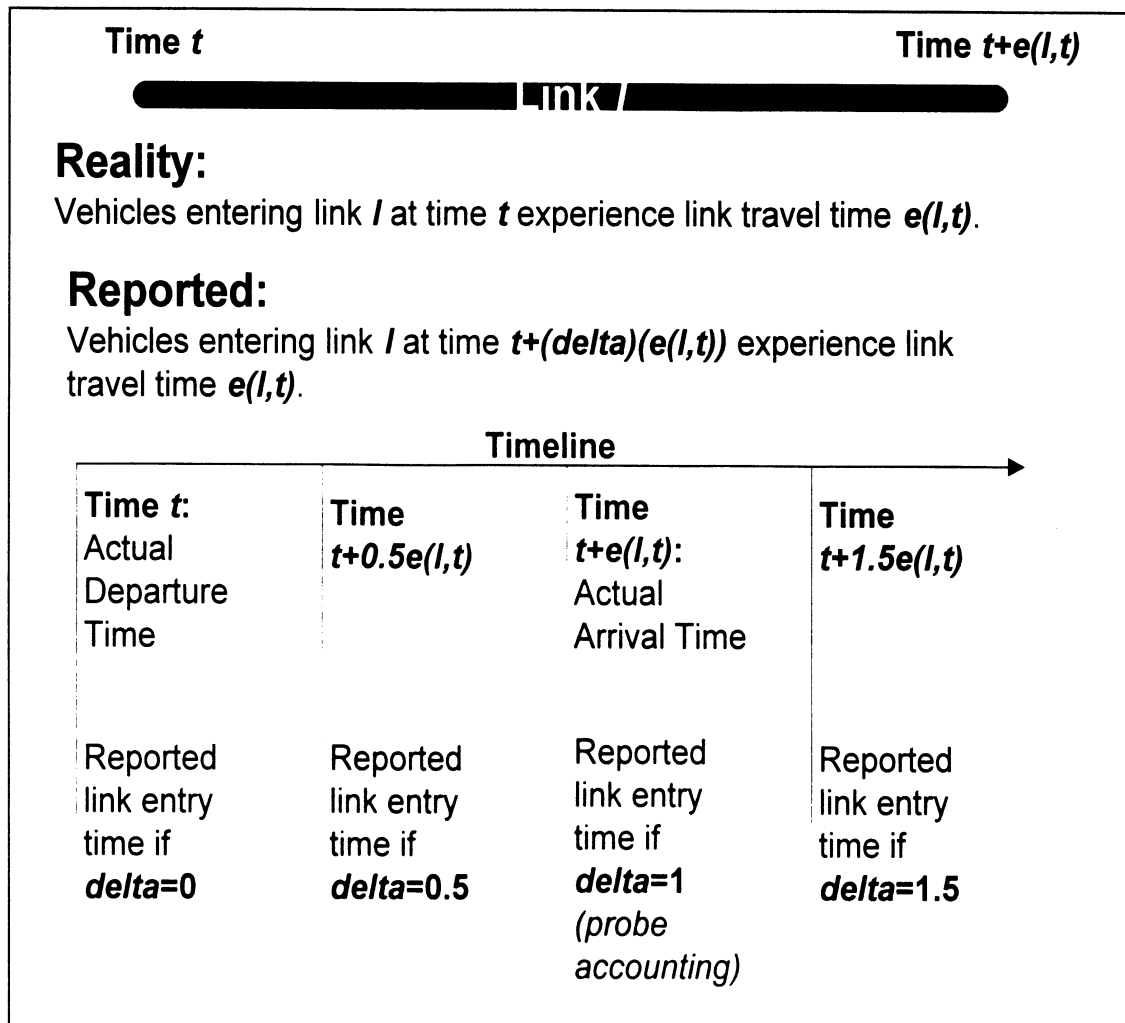


Figure 2: The effect of delta

ROLLING HORIZON PROCEDURES

At each iteration of the SAVaNT algorithm, an optimal routing must be determined for the entire problem horizon, and a simulation must then be run up to this horizon. If the problem horizon is long, this may lead to extremely slow execution. Further, in a practical situation, traffic demand patterns will not be known exactly until they occur, hence any traffic simulated in the SAVaNT loop will, at best, be an approximation. It therefore seems reasonable to attempt to

break the optimization process up such that a sequence of routings is determined for short time periods, each smaller routing problem being solved only when required and using the most recent data available. Such an approach is called a rolling horizon procedure and is illustrated in Figure 4. The length of each subproblem is called the rolling horizon (R), and the length of time during which a subproblem solution is implemented before another subproblem is solved is called the roll (r). Obviously, the roll may be no greater than the rolling horizon, but it may be shorter, in which case part of each subproblem solution is discarded. The case where roll and rolling horizon are identical is called a full roll.

In practice, rolling horizon procedures generally execute more quickly, since several the small subproblems are usually much easier to solve than the large original problem. More importantly from a practical viewpoint, only one subproblem need be solved at a time to yield implementable routing decisions. Unfortunately, since a rolling horizon procedure only considers an immediate part of the original optimization problem at a time, its solutions tend to be myopic and are generally not optimal. In the case of vehicle routing, this myopia may not be significant, since travel demand is not known a priori, hence routings are only optimal for an approximation of the actual vehicle routing problem. Since the rolling horizon procedure need only solve subproblems when the corresponding decisions need to be implemented, it therefore benefits from more recent and accurate traffic data.

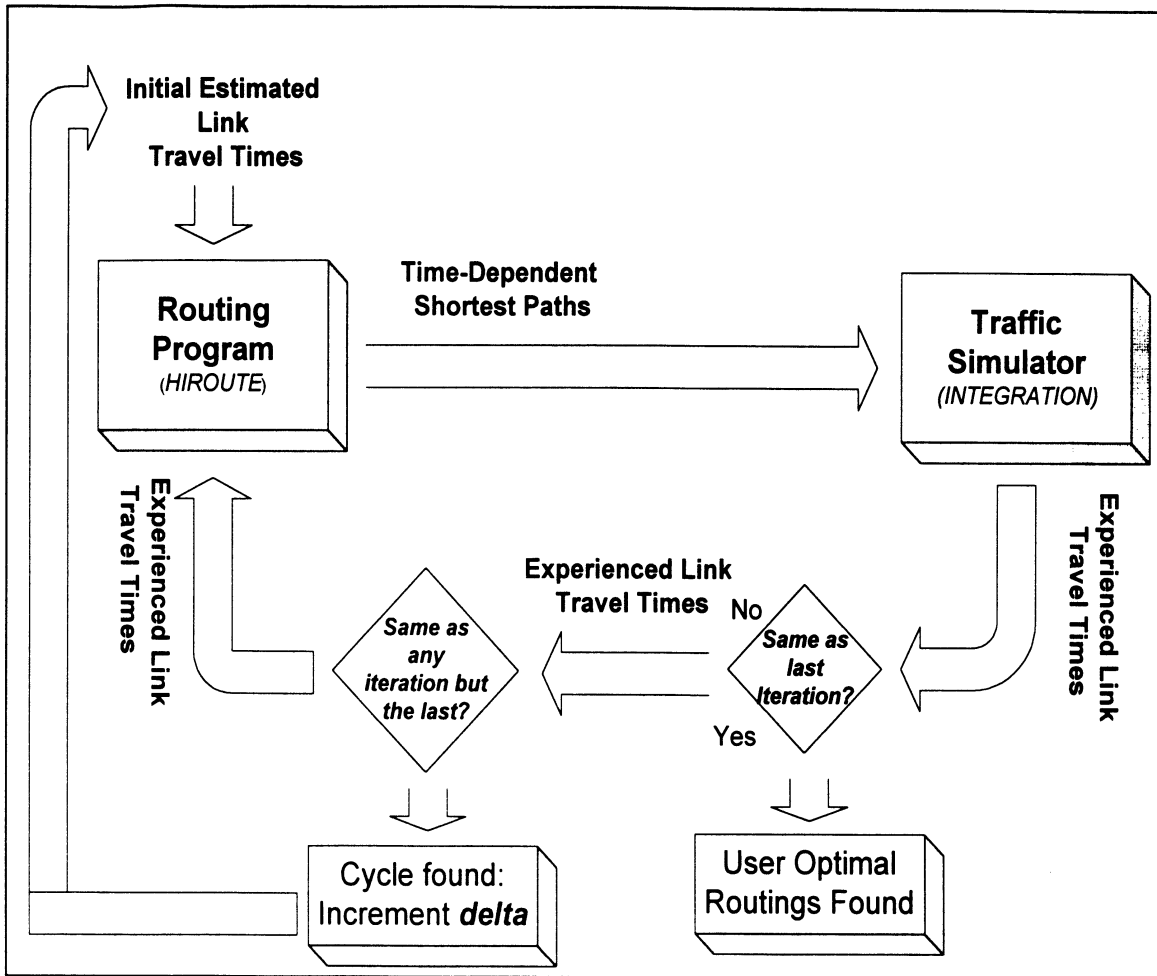


Figure 3: SAVaNT-CNV

In this paper we will examine rolling horizon implementations of the SAVaNT-CNV algorithm using the INTEGRATION traffic simulator and the Hiroute vehicle router. Section 2 will present such an algorithm as well as empirical results from its application to a simple traffic network. The algorithm will be refined to better deal with end of study effects in Section 3, and results of its application to optimization of the Troy, Michigan traffic network will be presented. Section 4 will examine the effect of varying traffic conditions on the performance of the rolling horizon algorithm by introducing a major incident into the Troy network. In Section 5, performance of the algorithm will be examined using a lower penetration level of guided vehicles and higher congestion in the Troy network in order to highlight algorithm performance characteristics. Finally, Section 6 will suggest avenues for further research and improvement of the algorithm.

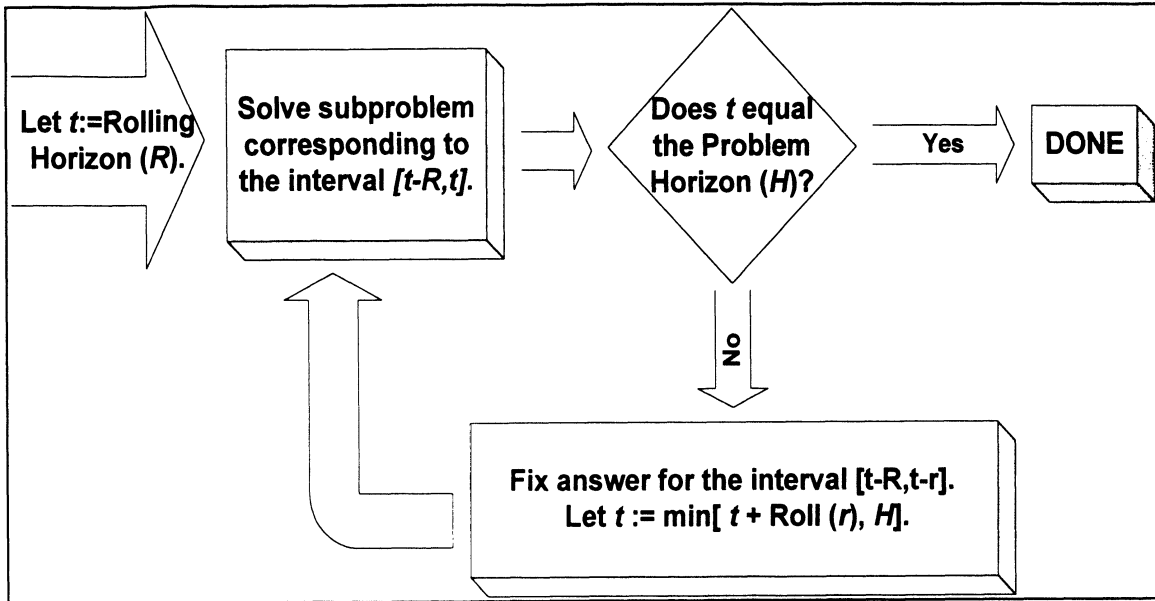


Figure 4: The rolling horizon procedure

2. AN INITIAL ROLLING HORIZON IMPLEMENTATION

PROBLEM DEFINITION

To construct a rolling horizon implementation of SAVaNT, it is essential to first have a well-defined concept of a subproblem over an interval $[t_1, t_2]$. The most obvious and simple way to define such a subproblem is to apply SAVaNT-CNV, starting with the network in its state at t_1 , and optimizing until t_2 . Such a definition has one very large drawback: If there is no way for a vehicle to reach its destination in time $(t_2 - t_1)$, what is considered to be an optimal route for such a vehicle? For short rolling horizons, the great majority of vehicles may fall in such a category. One possibility, the one used in our implementation, is to assume that link travel times will stay constant after time t_2 , thus giving a meaningful definition to an “optimal” route for each vehicle. Note that such a solution is subject to a very large end-of-study effect, since most vehicles are thus assumed to spend the majority of their trip under constant link travel time conditions. The rolling horizon SAVaNT-CNV algorithm corresponding to this subproblem model is as depicted in Figure 5. Note that the INTEGRATION traffic simulator may only begin execution with an empty network. Hence, to effectively begin a subproblem with a loaded network, it is necessary to execute INTEGRATION starting at time 0, using a fixed set of vehicle routing tables until the start of the subproblem. In practice, this arrangement loses much of the computational attractiveness of the rolling horizon procedure, but this is only an artifact of the choice of simulation software, not of the algorithm itself.

EMPIRICAL RESULTS

Our initial implementation was applied to a simple traffic network consisting of four nodes with bi-directional links forming the edges of a square, as depicted in Figure 6. This small network corresponds to a segment studied in [4] of the Sioux Falls network. Vehicles were routed onto the network from each node to each other node, the demand rate for each origin-destination pair varying according to a normal distribution centered on the mean presented in [4] and updated each minute.

The INTEGRATION traffic simulator requires a new routing table each time slice. These routing tables contain, for each (n, d) pair of node n and destination node d the identification number of the next link a vehicle currently at node n should take to get to node d . For the purposes of these empirical tests, we used a time slice of 2 minutes and a problem horizon of 30 minutes. In addition to using a single routing table for each time slice, INTEGRATION also employs the same link travel times throughout a time slice, updating them by exponential smoothing at the start of each slice. At the start of each subproblem, delta is set to 0.0, denoting accurate reporting of link travel times. When cycling occurs, delta is incremented in steps of 0.5.

In our simulations, twenty five percent of vehicles were designated to be of Class 3, or “intelligent”, routed according to the SAVaNT routes, another twenty-five percent were of Class 2 and were routed quasi-dynamically, and the remainder were routed as background traffic, Class 1. Quasi-dynamic vehicles are routed under the assumption that current link travel times will be maintained indefinitely and are assigned new routes every time slice. Background vehicles are routed according to free-flow link travel times, as if no congestion existed.

Figure 7 expresses the results of applying the rolling horizon SAVaNT-CNV algorithm to the routing problem using various rolls and rolling horizons. Note that in this figure, both rolls and rolling horizons are expressed in terms of 2 minute time slices, and that the case of a 15 time slice roll is actually the original, non-rolling horizon version of SAVaNT. The groups of bars in the graph represent different rolls corresponding to a same rolling horizon, though not all roll-rolling horizon combinations were evaluated. No clear relationship between roll or rolling horizon and performance is obvious from Figure 7. In fact, the non-myopic original version of SAVaNT produced the worst solution. This apparent contradiction is due to the use of varying delta values and to SAVaNT’s search for user-, rather than system-optimal solutions.

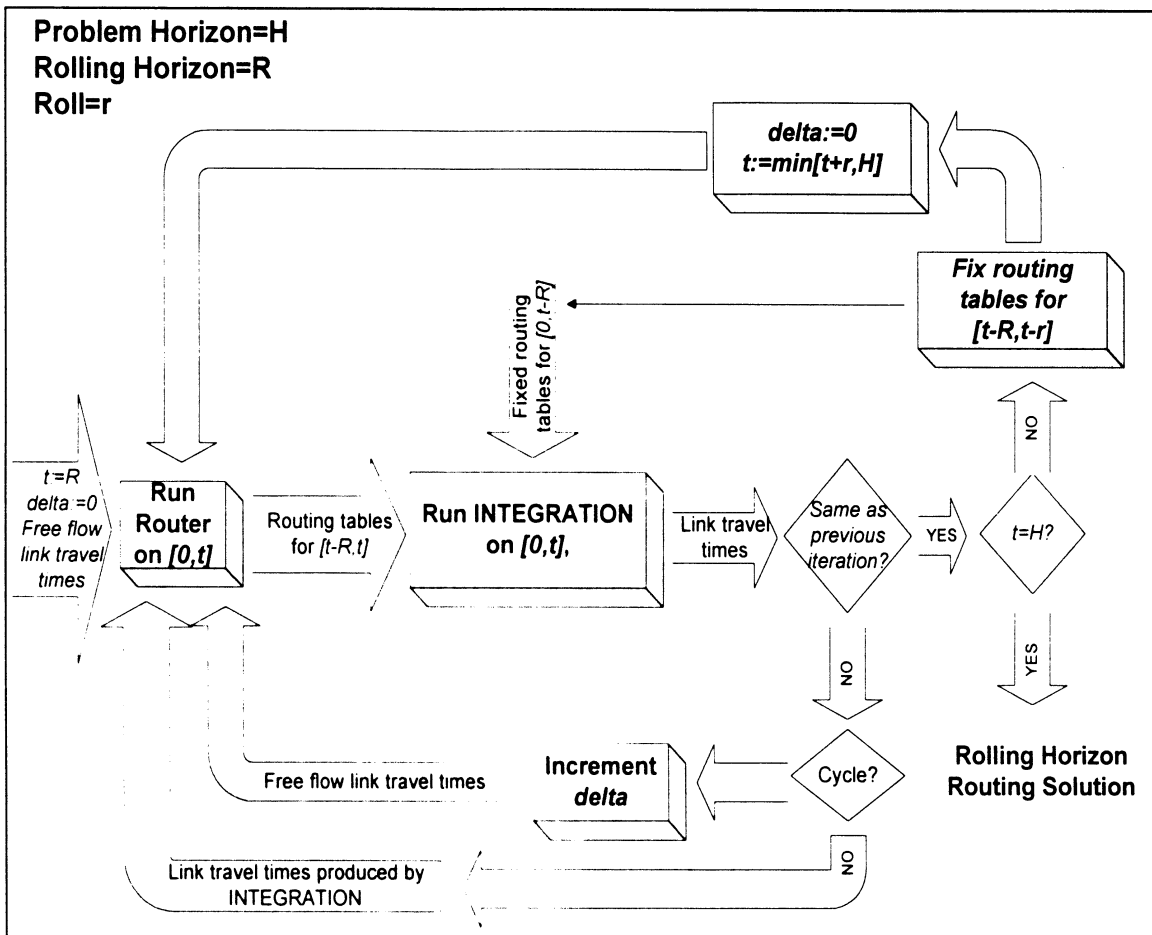


Figure 5: Rolling horizon SAVaNT-CNV

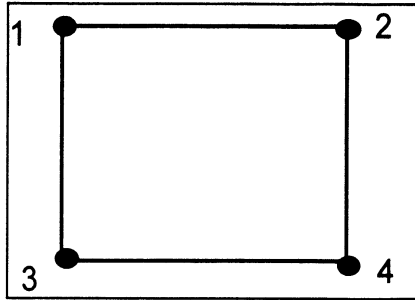


Figure 6: Fragment of the Sioux City network used in empirical tests

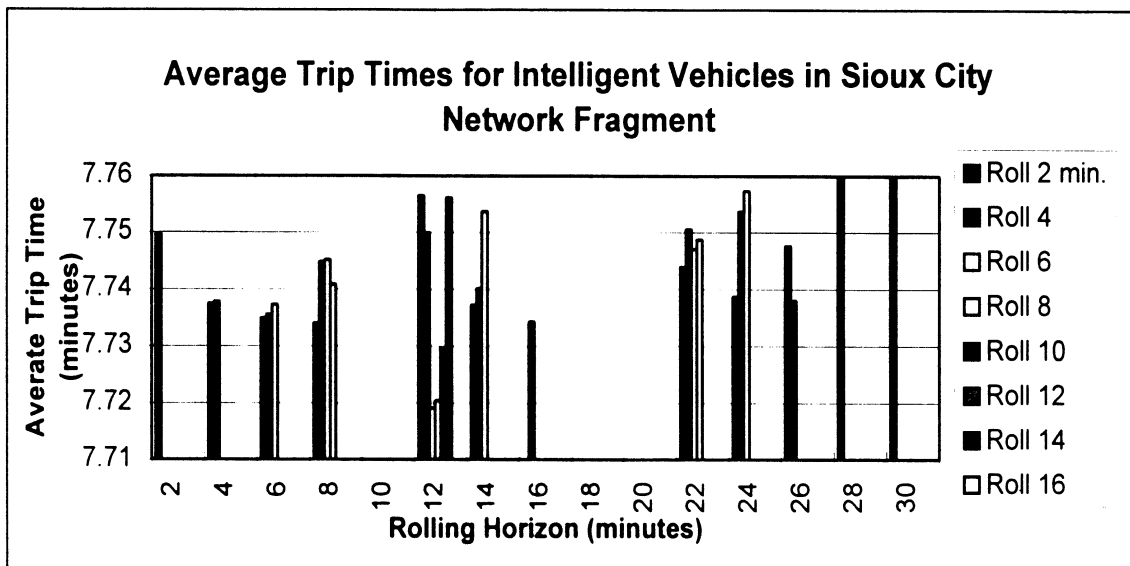


Figure 7: Empirical results of rolling horizon SAVaNT-CNV

3. A REVISED ROLLING HORIZON IMPLEMENTATION

PROBLEM DEFINITION

In the previous section, we noted that our definition of a subproblem led to very large end of study effects. These effects, we postulate, may contribute to the apparent randomness between average trip time and roll or rolling horizon. Hence, to minimize these effects, we have redefined the rolling horizon subproblems. Rather than terminating simulations and route computations at the end of the rolling horizon, we allow these to continue up to the problem horizon. We delineate our rolling horizon by curtailing all inflow of vehicles to the traffic network after the rolling horizon. Thus, since far more vehicles will be allowed to complete their trips in each subproblem, we reduce the magnitude of the end of study effects. The revised algorithm is illustrated in Figure 8. Obviously, this comes at a computational cost of running longer simulations and computing more routing table entries. However, since the routing table construction is not the computationally intensive module of the SAVaNT system and since no new vehicles enter the network after the rolling horizon, rolling horizon subproblems are still less computationally costly than the original problem. Additionally, subproblems generally require far fewer SAVaNT iterations than does the original problem.

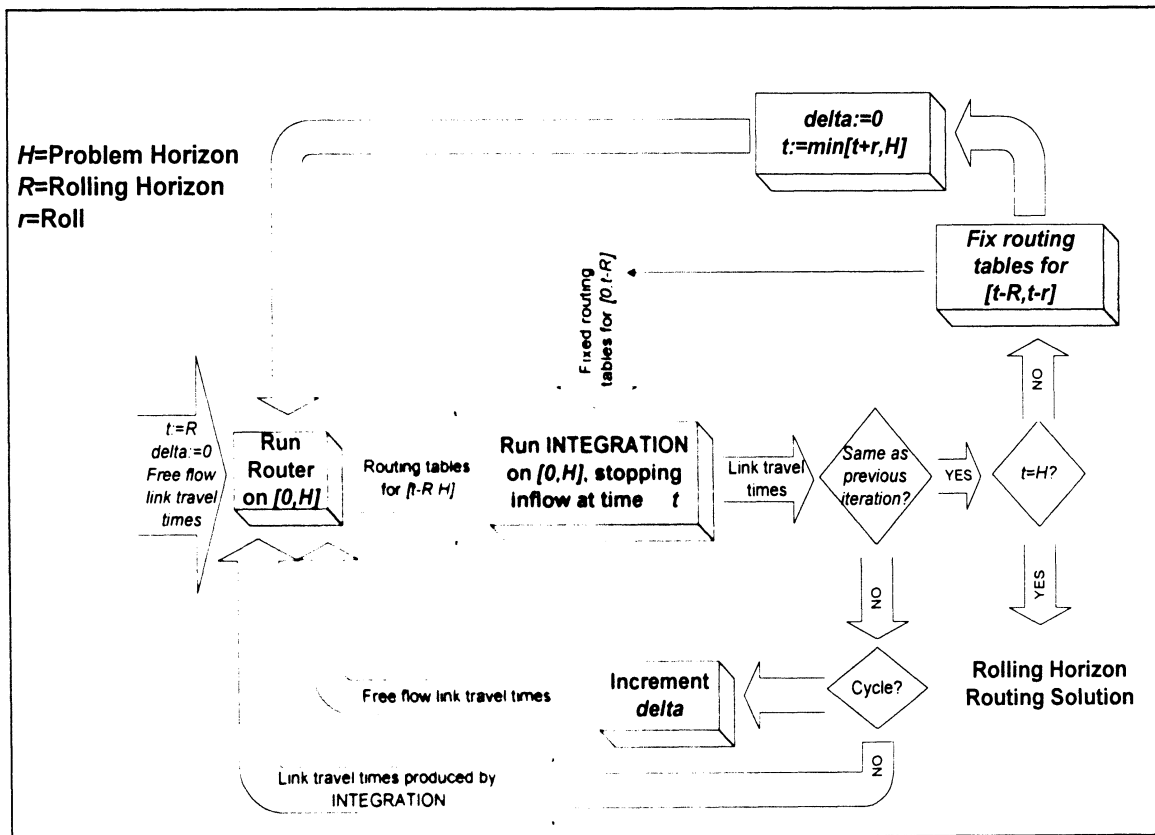


Figure 8: Revised rolling horizon SAVaNT-CNV

The current traffic simulator, INTEGRATION, is time-driven and scans the traffic network every 0.1 seconds to check for events such as vehicles arriving at nodes, vehicles entering the network, etc. An event-driven simulator, on the other hand, computes in advance the time of the next event and skips immediately to that point in time. Thus, if INTEGRATION were replaced by an event-driven simulator, the computational savings would be considerable, since, after the rolling horizon, the number of vehicles remaining in the network to trigger events diminishes rapidly.

EMPIRICAL RESULTS

In a further effort to distinguish the effects of roll and rolling horizon from possible artifacts of the problem definition, we have replaced the Sioux City subnetwork with a far more detailed model of the Troy, Michigan traffic network. This model, containing 529 links, 202 nodes, 72 origin/destination nodes, and over 100 signalized intersections, represents all major freeways and arterials in the city of Troy. Developed by Wunderlich [5], it is depicted in Figure 9. A constant inflow of approximately 50000 vehicles per hour, representing heavy congestion, is applied to the initially empty network for 30 minutes. The problem horizon is 30 minutes, divided into fifteen 120 second time slices. As in Section 2, 25% of the vehicles were routed by SAVaNT, 25% were routed quasi-dynamically, and the remainder were routed according to the shortest free-flow paths.

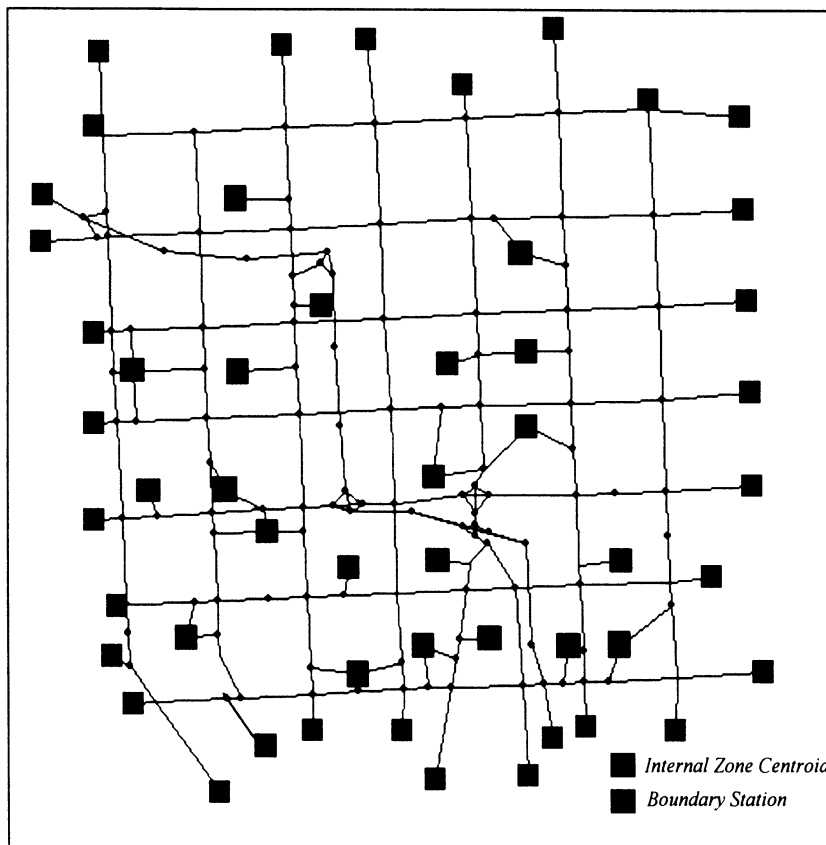


Figure 9: The Troy, Michigan traffic network

The average trip times of SAVaNT-routed vehicles as a function of different rolls and rolling horizons are illustrated in Figure 10. Results are grouped by rolling horizon, with the bars within each group representing different rolls. There appears to be a trend toward shorter trip times with longer rolls and shorter rolling horizons. This appears counterintuitive, since myopic solutions seem to be preferable. However, the results maybe be interpreted to be reasonable, since longer rolls allow subproblem solutions to be used to better effect and since longer rolling horizons tend to cause more cycling, and hence require larger values of delta with correspondingly less accuracy in link travel time estimates. Note, however, that the greatest and least average trip time experienced differ by only approximately 1%, hence any trend may not be significant.

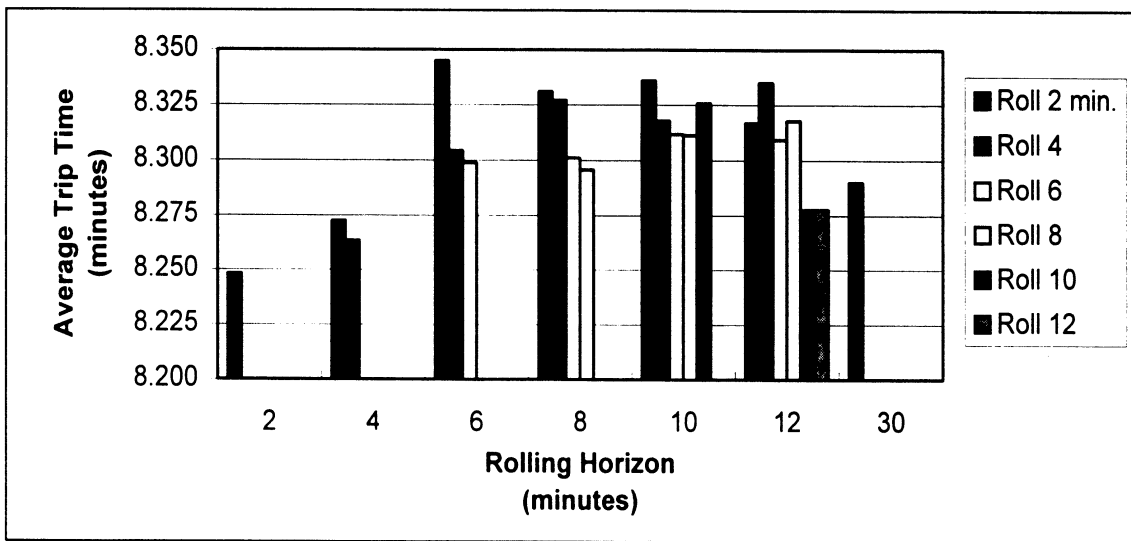


Figure 10: Average trip time of vehicles guided by rolling horizon SAVaNT-CNV

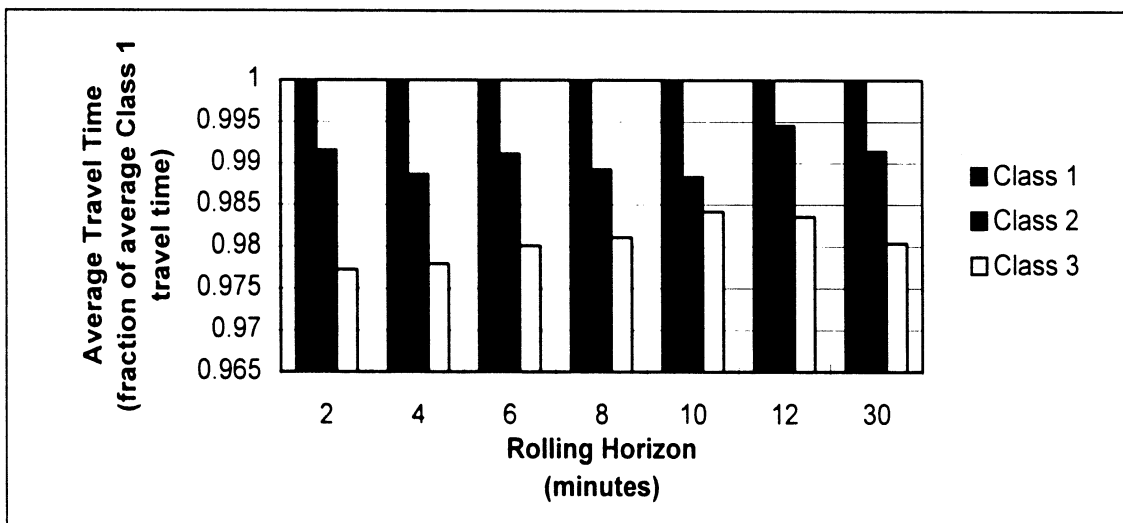


Figure 11: Performance of rolling horizon SAVaNT-CNV with respect to vehicle class

We also examined the improvement in trip times of SAVaNT-guided (Class 3) vehicles over background (Class 1) and quasi-dynamic (Class 2) vehicles. Normalizing with respect to the trip times of Class 1 vehicles and considering only full-roll scenarios, we obtain the results of Figure 11. Each group of three bars represents the average trip time of vehicles of the three classes using the corresponding rolling horizon and a full roll. What is significant in this figure is not the appearance of a trend, but the complete lack of such a trend. Figure 11 suggests that the choice of roll has insignificant effect on the performance of SAVaNT-CNV.

Though the revised version of SAVaNT-CNV reduced the magnitude of end-of-study effects, such effects still exist and may distort the meaning of performance parameters. The average trip time data, for instance, is taken from an INTEGRATION output file. Unfortunately, INTEGRATION only counts the trip times of vehicles that have completed their trips when computing this information. Thus, a particularly bad routing that excessively delays vehicles with long trips may in fact be reported to yield a lower average trip time, since shorter trip times will be over-represented! We have examined two methods for minimizing this distortion.

The first alternative assigns to all vehicles stranded on the network at the end of the problem horizon the average trip time experienced by vehicles completing the corresponding trip. Figures 12 and 13 correspond to Figures 10 and 11, but use this new accounting scheme for stranded vehicles to compute average trip times. With this new accounting, it appears from Figure 12 that longer rolling horizons yield longer average trip times. Note also from Figures 11 and 13 that the improvement in performance between SAVaNT routings over the other routings is significantly greater when stranded vehicles are accounted for. As before, the average trip times of the three classes of vehicles remained relatively constant as the rolling horizon was varied.

A second method for dealing with distortions in reported average trip times by INTEGRATION is to eliminate the possibility of stranded vehicles. Though few stranded vehicles are left on the network in the first rolling horizon subproblems since vehicles have nearly 30 minutes to reach their destinations, this number increases as the algorithm progresses. Rather than always terminating simulations and routings at the problem horizon, in our case 30 minutes, we instead terminated them at 60 minutes to allow the vast majority of vehicles to complete their trips. The results of this series of simulations are expressed in Figures 14 and 15. The average trip times in Figure 14 vary by almost an order of magnitude more than those in the corresponding Figures 10 and 12. Contrary to these figures, the trend in Figure 14 suggests that longer rolling horizons, rather than shorter ones, yield shorter average trip times. This is the result one's intuition would suggest. However, comparing Figure 15 to Figures 13 and 11, it is apparent that when the network is allowed to clear, quasi-dynamic routing tends to perform almost as well, and in some cases better, than SAVaNT-CNV routing. We postulate that this is since a significant portion of total trip time is incurred after the problem horizon of 30 minutes, in a long period of relatively light congestion where quasi-dynamic routing is nearly optimal.

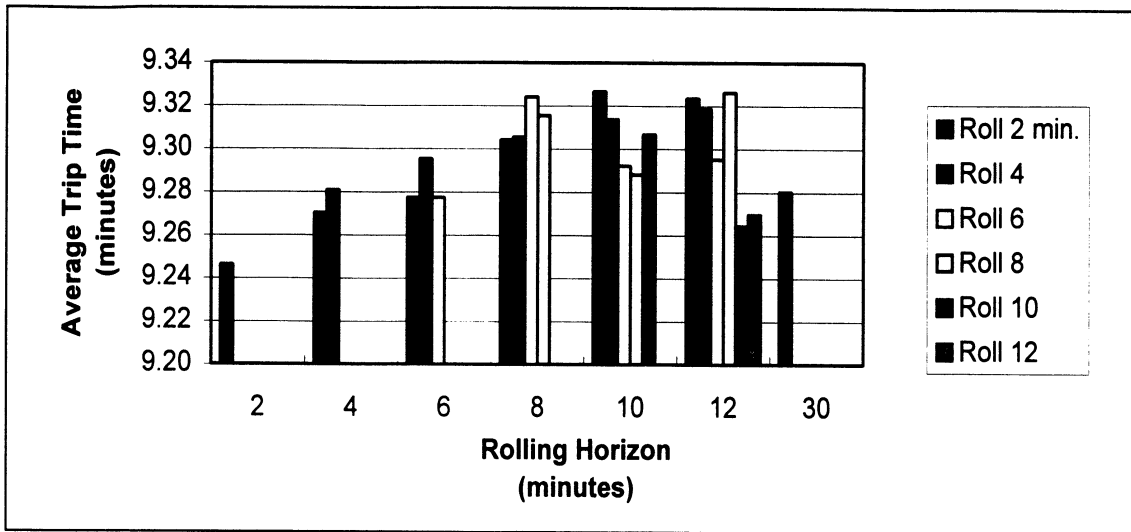


Figure 12: Average trip times of vehicles guided by rolling horizon SAVaNT-CNV when accounting for stranded vehicles

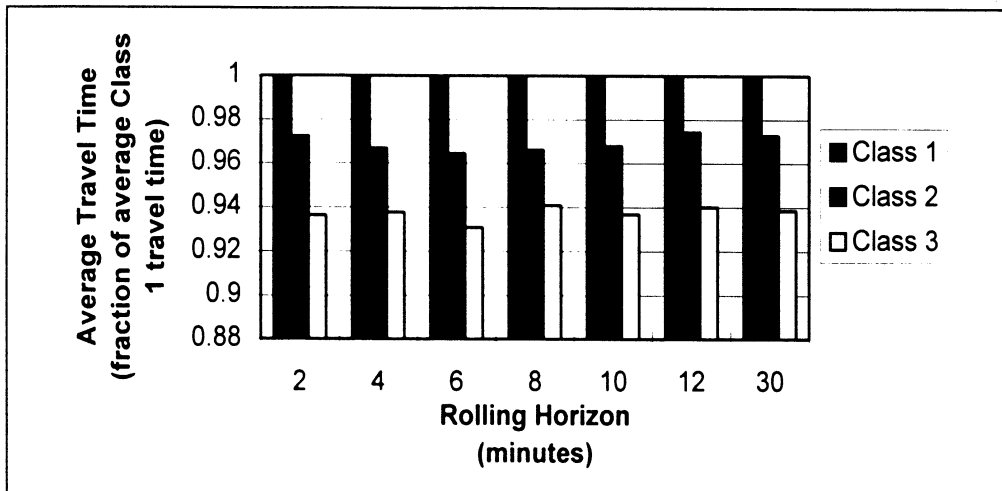


Figure 13: Performance of rolling horizon SAVaNT-CNV with respect to vehicle class when accounting for stranded vehicles

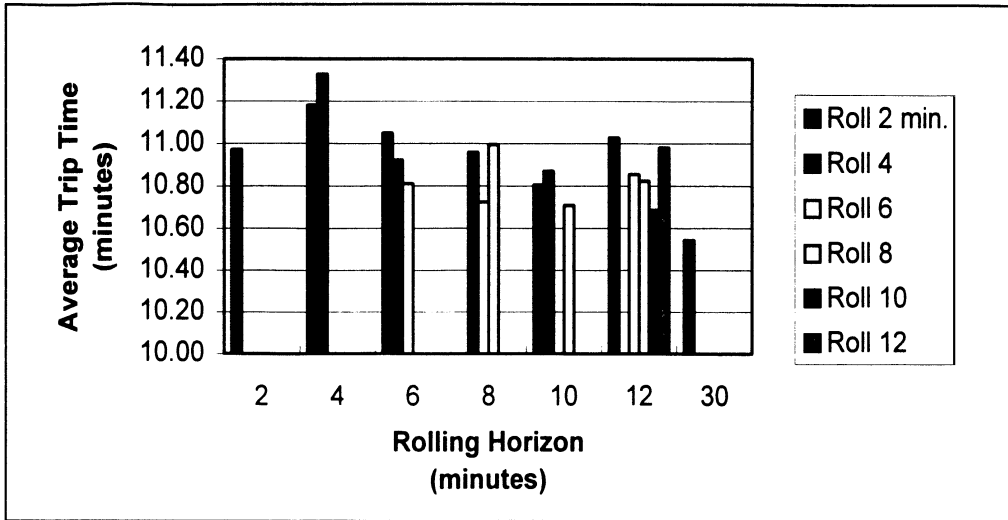


Figure 14: Average trip times of vehicles guided by rolling horizon SAVaNT-CNV when allowing network to empty

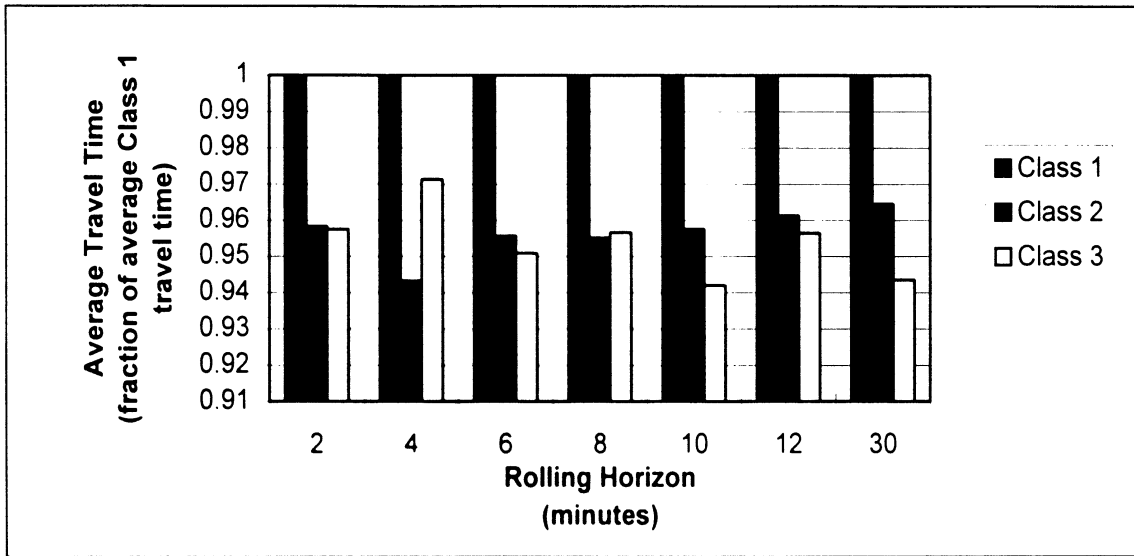


Figure 15: Performance of rolling horizon SAVaNT-CNV with respect to vehicle class when allowing network to empty

4. DEALING WITH VARYING TRAFFIC CONDITIONS

TRAFFIC SCENARIO

In the previous section, we always assumed that travel demands and road capacities stayed constant throughout the routing problems. Clearly, this is an unrealistic assumption. Because of their myopic nature, the performance of rolling horizon algorithms clearly may be degraded if either of these assumptions fails to hold. To measure the extent of this degradation, we again consider the traffic scenario from Section 3, but we now simulate an incident by completely closing a link representing part of the I-75 freeway, the largest road in Troy, from 15 minutes to 30 minutes into the scenario, as depicted by the bold arrow in Figure 16. Such a shutdown should reduce the congestion on the remainder of the I-75 and hence modify traffic patterns throughout the network.

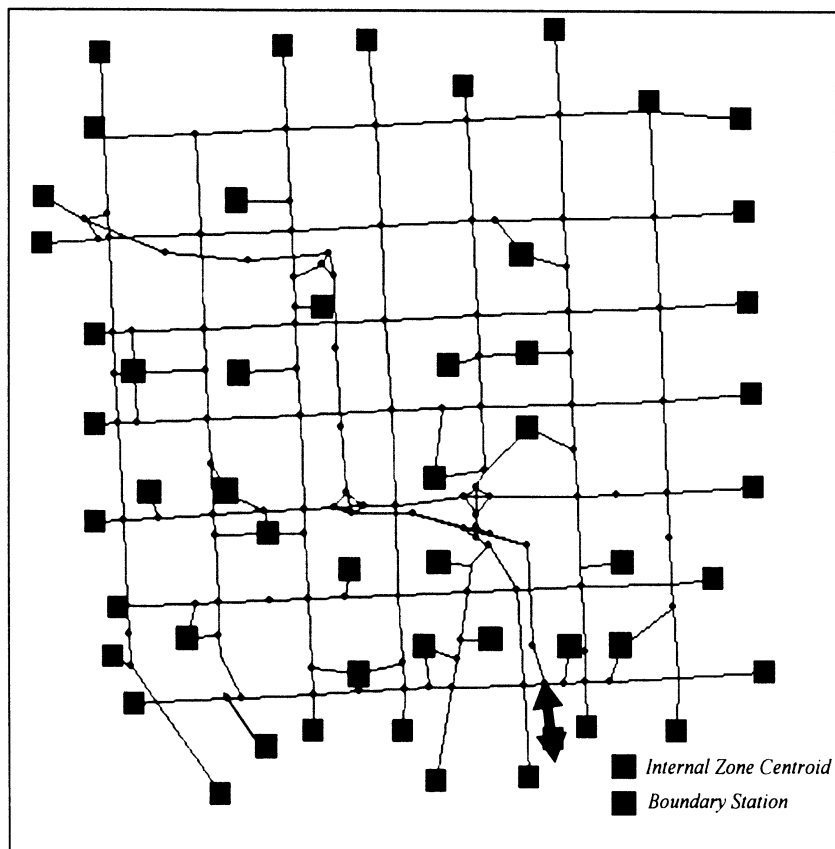


Figure 16: The Troy, Michigan traffic network with an incident

EMPIRICAL RESULTS

Since it produced the most significant trend in previous experiments, we chose to allow the network to empty to minimize end of study effects. The average trip times for vehicles of each of the three classes in the Troy network in the presence of an incident are graphed in Figure 17. As usual, each group of three bars represents the average trip time of vehicles of each class using the corresponding rolling horizon with a full roll. The performance of rolling horizon SAVaNT-CNV in this scenario seems to have little dependence on the length of the rolling horizon. Note that the performance of quasi-dynamic routing approached that of SAVaNT more closely than in previous experiments. This agrees with our intuition, since in the presence of varying traffic conditions, the myopic nature of rolling horizon SAVaNT-CNV should lead it to approximate quasi-dynamic routings.

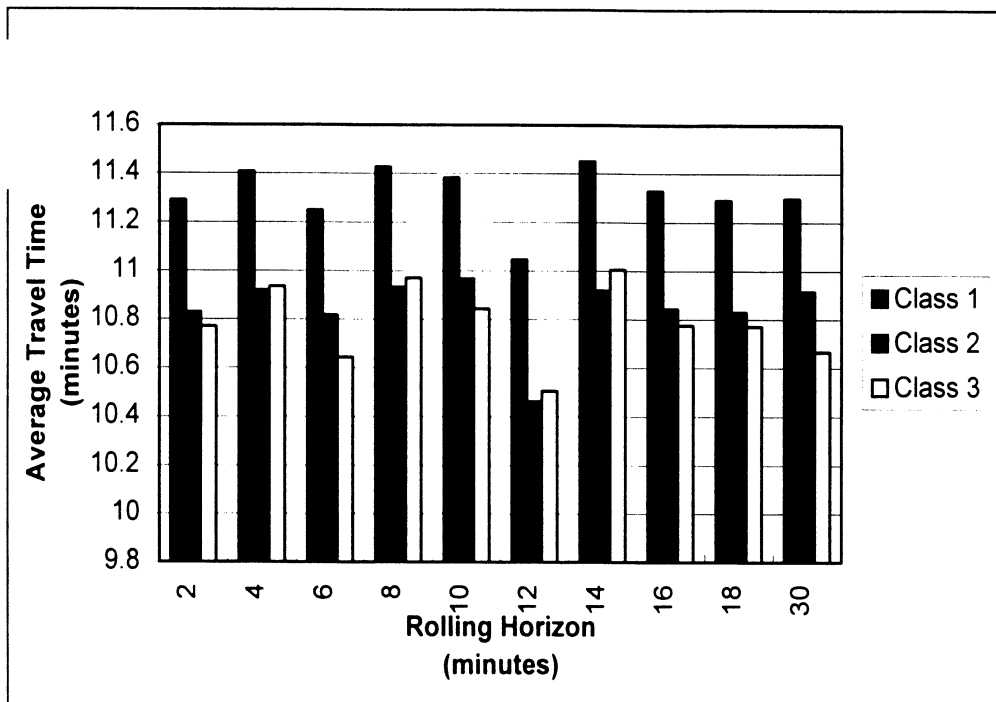


Figure 17: Average trip times of vehicles guided by rolling horizon SAVaNT-CNV in the presence of an incident when allowing network to empty

5. THE TROY NETWORK REVISITED

TRAFFIC SCENARIO

The greater the congestion, the more benefits stand to be reaped from using an anticipatory routing algorithm such as SAVaNT. Further, the lower the penetration of SAVaNT-guided vehicles into the traffic network, the greater will be the reduction in the trip times for such vehicles, as observed by Wunderlich [3]. Intuitively, this is since at high penetration levels, the large volume of guided vehicles shifting to more efficient routes has a more significant effect on raising the congestion, and hence the travel time, along such routes. To more clearly distinguish the performance characteristics of rolling horizon SAVaNT-CNV, we will therefore consider a scenario of high congestion and low penetration, first presented in Wunderlich [3]. Traffic inflow rates are increased by 33% over those in Sections 3 and 4, but inflow is terminated after a problem horizon of 24 minutes. Simulations are allowed to run for 2 hours to allow the network to clear completely in each rolling horizon subproblem. Time slice length is reduced to 90 seconds to improve the accuracy of link travel time estimates. The penetration levels of Class 2 (quasi dynamic) and Class 3 (SAVaNT-guided) vehicles are both dropped to 5%. No incidents are modeled.

DELTA-CURTAINS

Recall that the INTEGRATION traffic simulator must begin all simulations with an empty network, hence, to solve a rolling horizon subproblem, it is necessary to run INTEGRATION from time 0 using a fixed set of routing tables until a desired loaded network is attained. If a positive delta value is employed to deal with cycling, it is therefore possible for vehicles arriving at their destinations before the start of the rolling horizon subproblem to have their arrivals shifted into the subproblem time interval. Thus, supposedly “fixed” events, vehicle arrivals occurring before the start of the rolling horizon subproblem, may vary with delta and influence the subproblem solution. Further, in any practical implementation of rolling horizon SAVaNT-CNV, the traffic simulator should be capable of starting with a fixed loaded network when applied to subproblems. Hence, in such an implementation, the shifting of vehicle arrivals from before the start of a subproblem to the subproblem interval itself could not take place. Effectively, a “delta-curtain” is placed at the start of the subproblem time interval, preventing any vehicle arrivals occurring before the curtain to be shifted forward through it in time.

In order to determine the effect of the presence of a delta-curtain, we applied rolling horizon SAVaNT-CNV to the traffic scenario described above using full rolls with rolling horizons of 4, 8, 12, and 16 time slices, both with delta-curtain and without. Note that the last case, a rolling horizon of 16 time slices, is equivalent to non-rolling horizon SAVaNT-CNV.

EMPIRICAL RESULTS

The results of applying SAVaNT-CNV to the new traffic scenario are expressed in Figure 18. The presence of a delta-curtain always had a negative effect. In the two instances where the delta curtain had a significant detrimental effect, extreme cycling was noted. Obviously, since there is only one subproblem when a rolling horizon of 16 time slices is used, the delta-curtain had no effect in this instance. Figure 19 graphs the average time spent in the network per vehicle as a function of time for each rolling horizon/curtain combination. The discontinuity after 16 time slices results from the termination of new vehicle inflow, thus eliminating a source of vehicles with little or no time in the network. As time progresses, the network empties and average travel time converges to the values expressed in Figure 19. Note that the rolling horizon used and the presence or absence of a delta-curtain had virtually no effect before the problem horizon of 16 time slices.

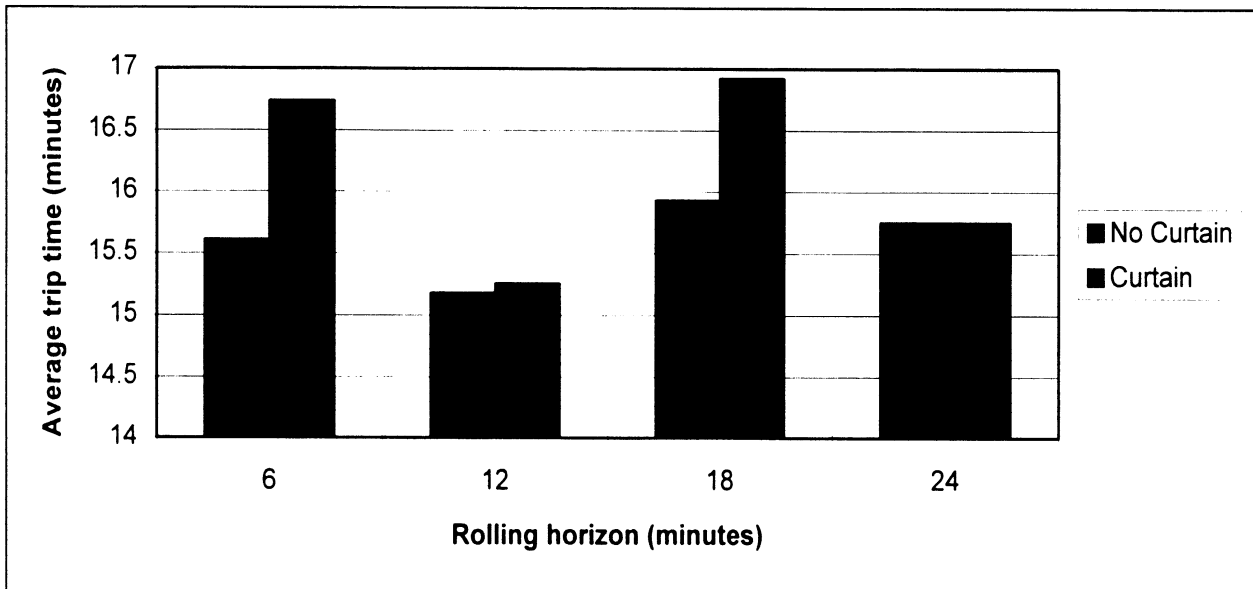


Figure 18: Average trip times of vehicles guided by rolling horizon SAVaNT-CNV as a function of rolling horizon and presence of a delta-curtain

Let us consider one instance where the presence of a delta curtain led to significant deterioration of routing quality and examine what went wrong. In this case a rolling horizon of 4 time slices was employed in concert with a delta-curtain. From the INTEGRATION output, we found that a maximum of 658 Class 3 (SAVaNT-guided) vehicles were present on the network at the end of any time slice. Hence, most Class 3 vehicles were usually the only member of their class on links. Cycling was thus encouraged, since these lone Class 3 vehicles, the sole source of Class 3 congestion on links, would tend to flipflop between two routes having similar travel

times. As delta was incremented to deal with the cycling, only vehicles arriving at their destinations after the start of the rolling horizon subproblem had their trips shifted forward in time by delta. Because of the delta-curtain “holding back” all vehicles arriving at their destinations before the start of the rolling horizon subproblem, it appeared to the routing program that conditions similar to free flow existed near the start of the subproblem. Only Class 3 vehicles arriving at their destinations after the start of the subproblem therefore affected the SAVaNT solution, and the sparsity of such vehicles was further exacerbated as they were spread out chronologically by the effect of delta. Cycling was thus further encouraged, leading to further increases in delta, which in turn encouraged more cycling. Eventually, delta tended to reach values so large that SAVaNT computed an optimal routing little different from a free flow routing.

Figures 20 and 21 illustrate the performance differences between the cases where a rolling horizon and roll of 4 time slices was employed respectively with and without a delta-curtain. Note the change in the graph of the average time spent in network by SAVaNT-guided vehicles in these two figures. Figures 22 and 23 are corresponding graphs illustrating the number of vehicles of each vehicle class in the network as a function of time using a delta curtain and without a delta-curtain, respectively. In these last two figures, the number of Class 2 and Class 3 vehicles is multiplied by 18 to normalize the quantities with respect to the number of Class 1 vehicles. Due to the low penetration level of Class 3 vehicles, considerable cycling was present in both cases, leading SAVaNT-guided vehicles to experience longer travel times than quasi-dynamically guided vehicles. Note in Figures 22 and 23 the jog in the graph indicating a sudden disparity in the number of Class 2 and Class 3 vehicles in the network resulting from particularly bad cycling in the subproblem routing Class 3 vehicles in the interval from 8 to 16 time slices.

Note further from Figures 20-23 that relatively little difference exists between the average times spent in the network by the three classes of vehicle during the problem interval itself, from 0 to 16 time slices, as compared to at the end of 2 hours. Partially, this is due to the small fraction of vehicles that managed to complete their trips during the problem interval, but it also suggests that the unrealistic “rampdown” phase with zero vehicle inflow may be biased toward quasi-dynamically routed vehicles. This observation is reinforced by Figures 24 and 25 which contain a comparison of average time in the network for each class of vehicle after 16 time slices (the problem horizon) and after 2 hours, respectively, for various rolling horizons. These figures represent the case when no delta-curtain was employed. Values in these Figures 24 and 25 are normalized such that the average time in the network for Class 1 vehicles is 1.0. From these figures, it appears that rolling horizon has little impact on the performance of Class 3 vehicles with respect to the other vehicle classes. The performance gap between Class 2 and Class 3 vehicles increases in all cases from approximately 6.5% of the Class 1 travel time after 16 time slices to 9% after 2 hours. In absolute terms, the performance gap triples from an average of 0.5 minutes to 1.5 minutes during the same interval.

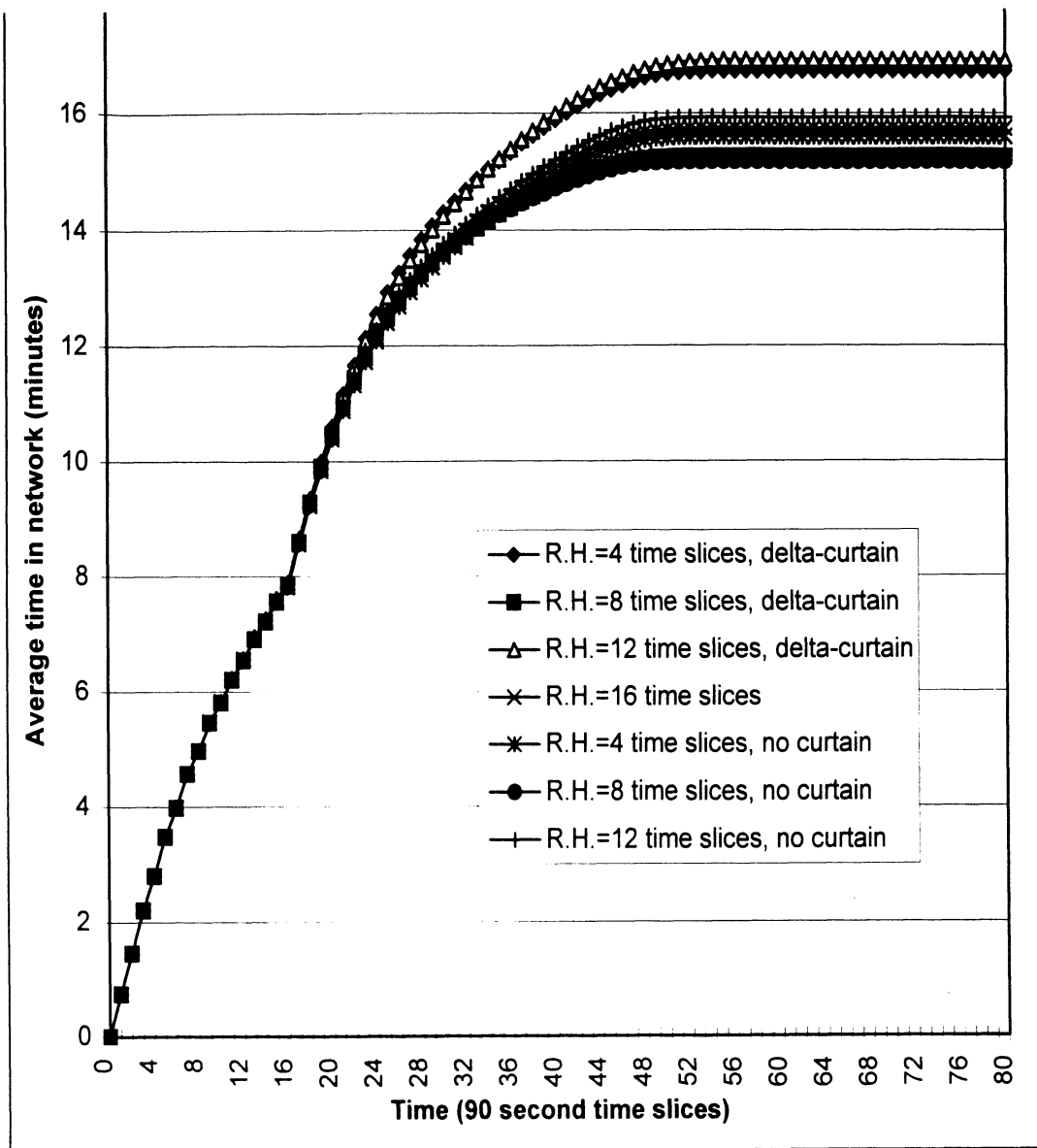


Figure 19: Average time spent in network by SAVaNT-CNV guided vehicles as a function of time, rolling horizon, and presence of a delta-curtain

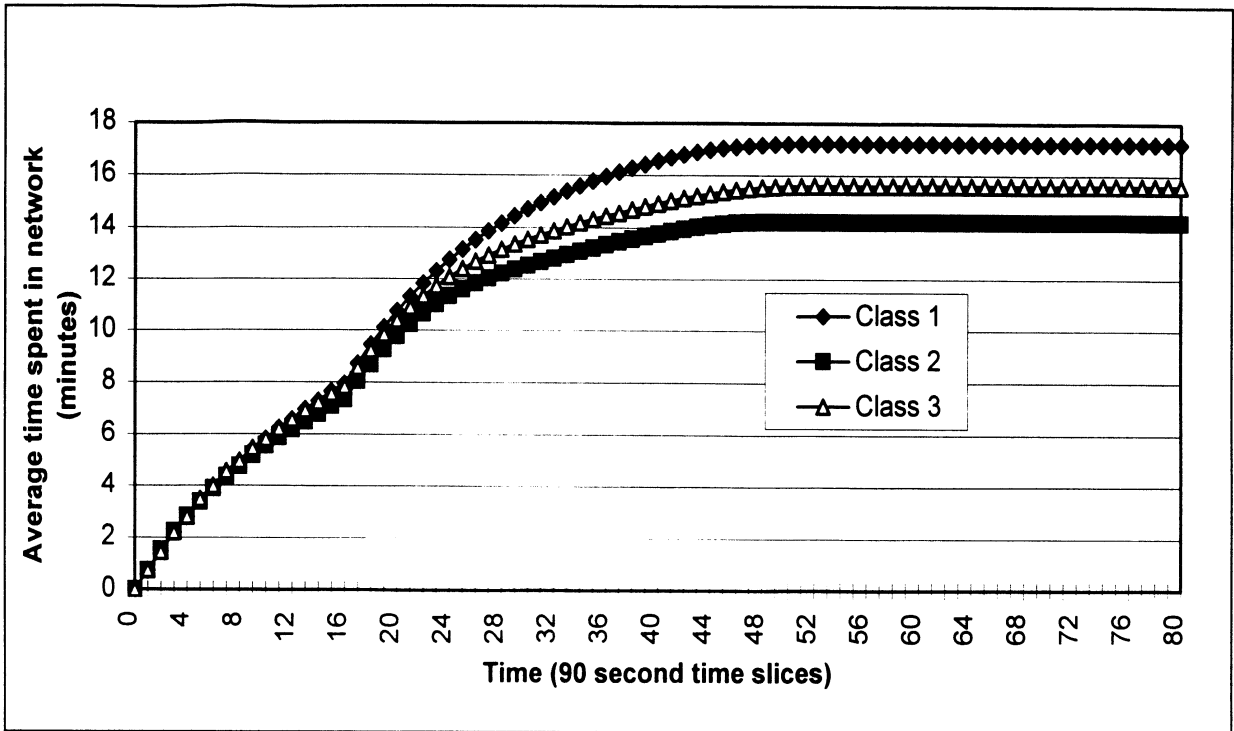


Figure 20: Average time spent in network by vehicles from each class as a function of time using a rolling horizon of 4 time slices and no delta-curtain

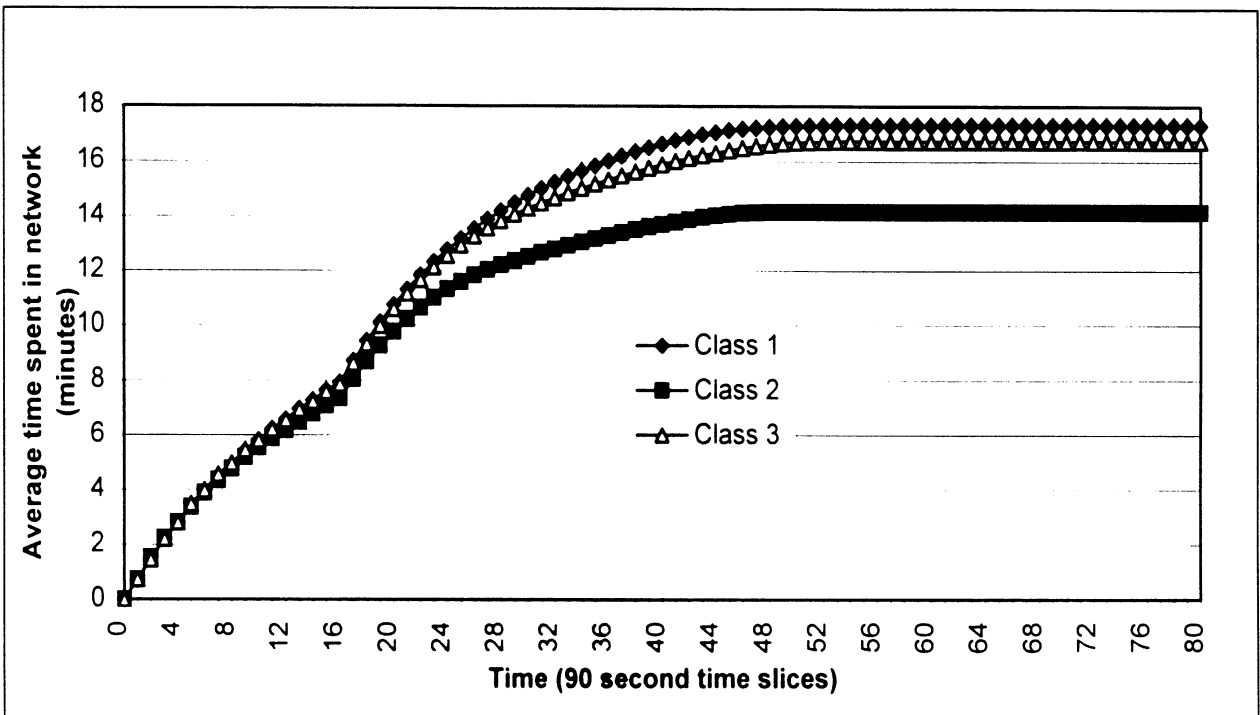


Figure 21: Average time spent in network by vehicles from each class as a function of time using a rolling horizon of 4 time slices and using a delta-curtain

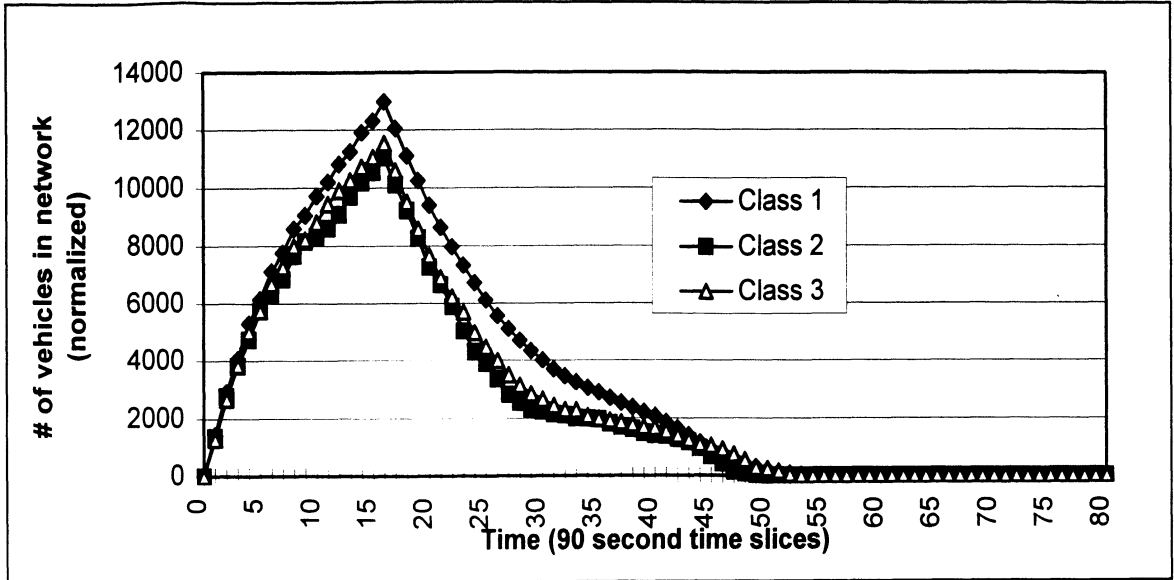


Figure 22: Number of vehicles in the network as a function of time using a rolling horizon of 4 time slices and no delta-curtain

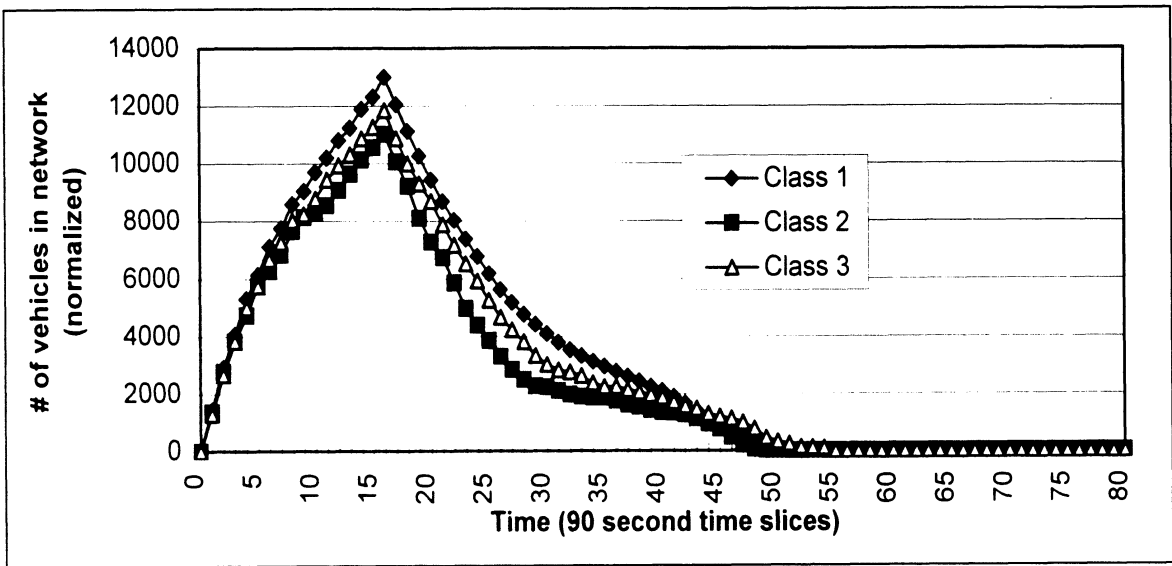


Figure 23: Number of vehicles in the network as a function of time using a rolling horizon of 4 time slices and using a delta-curtain

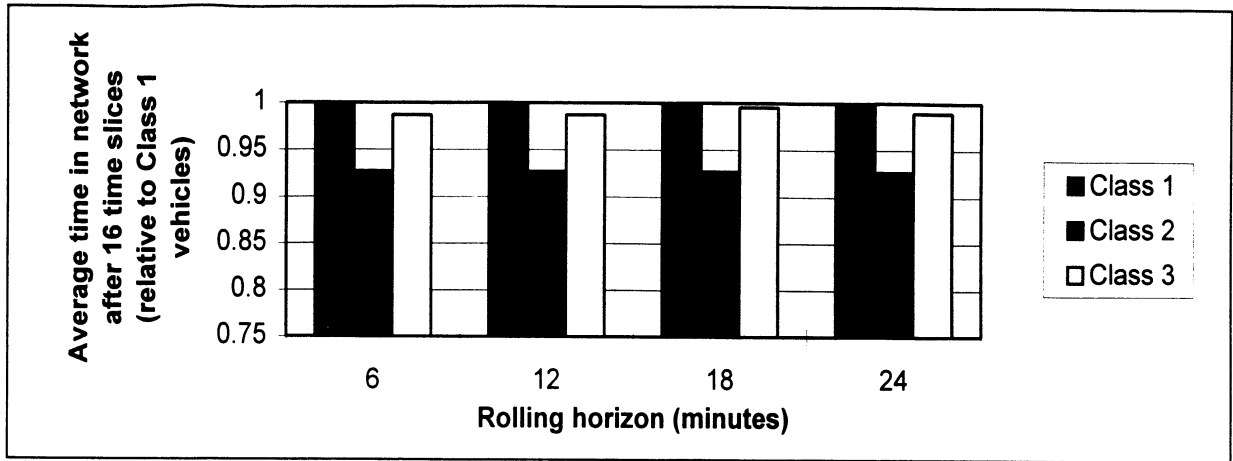


Figure 24: Average time spent in network at the end of the problem horizon relative to Class 1 (background) vehicles.

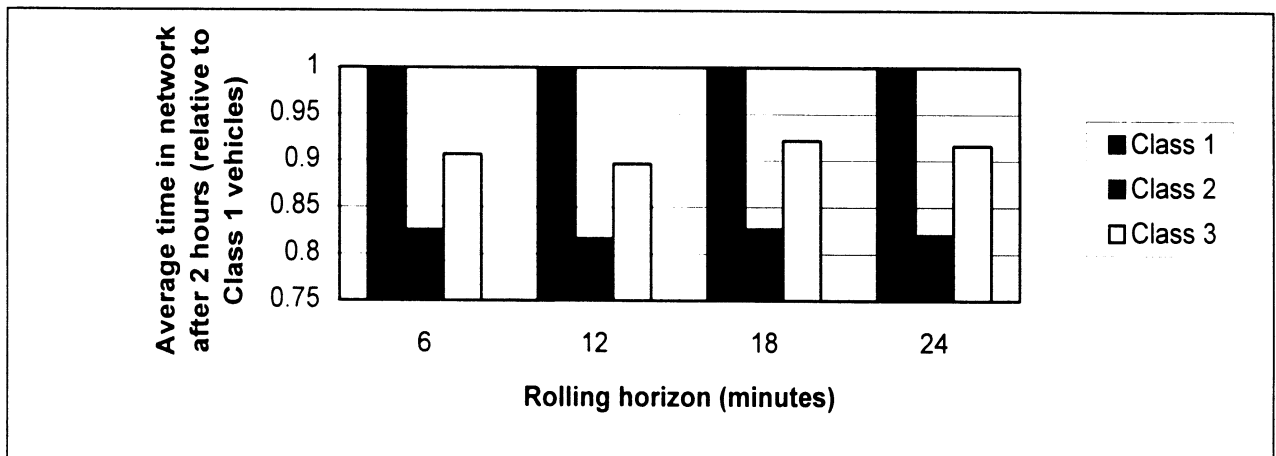


Figure 25: Average time spent in network at the end of 2 hours relative to Class 1 (background) vehicles

6. EXTENSIONS AND CONCLUSIONS

The SAVaNT optimization system has been shown to yield user-optimal routings for detailed, realistic traffic networks, such as the Troy network. For practical implementation, some form of rolling horizon version of the algorithm such as outlined in this paper is vital. We have experimented with several different models of the rolling horizon subproblem in an effort to improve the performance of this algorithm. Above all other factors, this modeling issue appears to be the key factor in algorithm performance, far outweighing the choice of rolling horizon and roll in importance. Coupled to the modeling question is the design of appropriate test networks to evaluate the performance of SAVaNT. To date, all experiments have been conducted on initially empty networks which, due to the limitations of computer time, have never been simulated long enough to reach approximately steady state loadings. We hope to examine more realistic test scenarios to more fairly judge the SAVaNT algorithm.

A second major issue to address is the use of proper modules in the SAVaNT system. Because of its time-driven nature and inability to start with a loaded network, INTEGRATION is unsuitable for real-world implementations of rolling-horizon SAVaNT-CNV. However, since SAVaNT has been developed using the INTEGRATION traffic simulator from the beginning, it is unknown how much of the behavior of the SAVaNT algorithm is an artifact on the INTEGRATION traffic model. Currently, INTEGRATION exponentially smoothes the flow on each link and then uses this flow in a BPR impedance function to compute link travel times [5] at the start of every time slice. These new link travel times are then exponentially smoothed with the previous ones to produce a set of link travel times to be used throughout the next time slice. In our numerical experiments, the BPR impedance function utilized was linear, yielding relatively minor slowdowns due to congestion. In addition to fixed link travel times, a single routing table is employed for each vehicle class throughout each time slice. Before replacing INTEGRATION in the SAVaNT system, it is of key interest to determine how the damping effect of exponential smoothing and a linear BPR function, together with the coarseness imposed by time slices, affects algorithm performance. The importance of this task is highlighted by our discovery of the very significant performance effects of a delta-curtain and on INTEGRATION's method of accounting for stranded vehicles.

Though subproblem modeling and traffic simulator are of primary importance in the design of any rolling horizon implementation of SAVaNT-CNV, any such design must place considerable emphasis on dealing with cycling. Throughout our research on rolling horizon implementations of SAVaNT-CNV, excessive cycling has almost always been the cause of poor algorithm performance. In practice, it appears that many routings yielding cycles produce lower average trip times than does the solution SAVaNT eventually produces by incrementing delta. One approach to dealing with cycling is therefore to reduce the algorithm's optimality criterion. Rather than insisting on identical link travel time tables from consecutive iterations of SAVaNT, some margin for error could be allowed. Only that portion of the link travel time tables corresponding to the current subproblem, for instance, could be compared to test for

convergence. Alternatively, some threshold level of differences between the two iterations could be tolerated.

In conclusion, now that the SAVaNT-CNV optimization algorithm has been successfully implemented in a rolling horizon framework, we realize that further experiments are necessary to obtain a fuller understanding of the underpinnings of its performance, primarily with respect to scenario modeling, algorithm modules, and cycling behavior.

REFERENCES

1. Kaufman, D.E., Smith, R.L., and Wunderlich, K.E. (1991) "An Iterative Routing/Assignment Method for Anticipatory Real-Time Route Guidance", SAE Vehicle Navigation and Information Systems Conference Proceedings, P-253,701-708.
2. Wunderlich, K. E. (1994) "Link Travel Time Prediction Methods and Convergence for Iterative Anticipatory Route Guidance Methods."
3. Wunderlich, K.E. (1995) Ph.D. Thesis, University of Michigan.
4. Nonis, J. (1992) "Application of the MILP Dynamic Routing Model to the Sioux Falls Network".
5. Wunderlich, K.E. (1994) "User's Guide to INTEGRATION-UM Version 1.1"