CHAPTER 6
SUMMARY AND CONCLUSIONS

6.1 Summary of Objectives

The principal objectives of this research study are summarized below.

1. Motivate the need for specialized algorithms for traffic routing in dynamic and static ITS networks, using the computer network analogy.
2. Analyze the theoretical and implementation aspects of the time-dependent shortest path problem.
3. Classify the various time-dependent shortest path and *k*-shortest path algorithms available in the literature.
4. Prescribe a routing algorithm for ATIS applications that uses time-dependent *k*-shortest paths (TD-*k*SP) to obtain multiple optimal paths for dynamic networks.
5. Suggest suitable implementation schemes for the proposed algorithm.
6. Conduct an extensive statistical study of the computational performance of the algorithm using randomly generated dynamic link-delays and networks of various sizes.
7. Design a suitable interactive software environment where such routing algorithms can be used in tandem with other network optimization methods for ATMS/ATIS applications.
8. Integrate the algorithm into the Geographical Information System (GIS) based wide-area incident management software WAIMSS developed at the Center for Transportation Research, Virginia Tech.

6.2 Summary of the Research Work Completed in this Study

Traffic congestion has been cited as the most conspicuous and expensive problem in managing road traffic. To mitigate this problem, road users, including emergency response vehicles, need to be assigned multiple alternate routes to reach their destinations. At times of high traffic congestion, the state of the network (traffic flow and density) varies with time. Hence, the accuracy of predicting future link delays and the stability of the algorithm used is critical is determining the efficacy of these routing algorithms and dynamic traffic assignment procedures. Further, the throughput and average delay of a network are opposing goals and this further complicates any route assignment. In many cases, traffic may need to be routed via multiple paths.

Time-dependent shortest path (TDSP) algorithms, when combined effectively with a delay predicting mechanism can be effectively used to generate optimal dynamic routes in real-time. Although such TDSP problems are generally NP-Hard, they can be solved efficiently using pseudo-polynomial dynamic programming algorithms.

In this study, an efficient TDSP algorithm was combined with a *k*-shortest path algorithm to generate multiple dynamic shortest paths. While this algorithm does not always generate optimal paths, it can be used to obtain optimal simple paths for practical ATIS applications. Another feature of this algorithm is that it can be used to determine optimal schedules (departure times) for emergency response vehicles taking into account the variation in travel delays in the vicinity of the destination. An effective implementation using the double-ended queue has been accomplished. The computational performance of the algorithm was monitored by measuring the computational time as a function of various network parameters (nodes, density) and shortest path parameters (the value of *k*, and number of starting times). A logarithmic regression model was used to derive an empirical expression for the computational time-complexity of the algorithm. The results of the regression model and other statistical results are also presented.

Finally, an object-oriented method using the Booch technique of object-oriented design was used to design an interactive network optimization environment where such algorithms could be applied. The structure of the basic prototype was completed and class scenario studies were conducted. As a beginning, the algorithm was integrated into the

GIS-based wide-area incident management software system (WAIMSS) developed at the Center for Transportation Research, Virginia Tech.

6.3 Conclusions

Routing algorithms are most effective when they incorporate both quantity (throughput) and quality (delay) of service measures in their computations. In future ITS networks, the road users will be subjected to a greater degree of control and flow regulation using ramp-metering, on-board road diversion plans, variable message signs, etc., to generate user-optimal or system-optimal traffic assignments. A study of today's computer networks can be used to obtain information on the architecture of future ITS networks and ATIS implementation scenarios. In such networks, shortest-path based routing may be an efficient and cost-effective tool in directing traffic. Partially or fully disjoint paths or *k*-shortest paths can be generated to provide multiple optimal paths to users. TDSP algorithms can be combined with an effective delay-feedback mechanism to produce an extremely stable and useful routing tool. In addition, they can be combined with a *k*-shortest path algorithm (TD-*k*SP) to efficiently generate multiple optimal paths in real-time. The double-ended queue is an efficient data structure that can be used as a path-processing data structure in label-correcting (LC) algorithms such as TD-*k*SP. Computational results show that the double-ended queue used is an extremely efficient data structure for sparse networks. However, it can be expected that as the networks become more dense, label-setting (LS) algorithms will be relatively more effective. Transportation networks are naturally (extremely) sparse and lend themselves to LC implementation.

The results of the computational performance tests suggest that Algorithm TD-*k*SP is ideal for generating dynamic diversion routes and schedules in real-time, since it is capable of high-speed computations (in the order of seconds) even for large-sized networks (up to 3000 nodes and 5000 arcs). Using more efficient data structures for sorting and storing calculations can further speed up the algorithm. Also, automated graphical input and output codes can be written so that algorithm can be made part of a larger GIS and Java - based network optimization module by using software integration methods.

6.4 Recommendations for Future Research

Algorithm TD-*k*SP is one of the many algorithms that can be utilized for traffic routing. Sometimes the *k*-SP paths obtained are practically useless as they do not differ significantly from each other. Hence, using selectively or completely disjoint paths to route traffic may be an useful idea in cases where *k*-SP algorithms do not generate paths that can avoid bottlenecks. A more detailed study of the computational complexity of TDSP algorithms can be conducted. Also, Algorithm-TD-*k*SP can be tested for various types of PPDS. The application of TDSP algorithms to other problems such as construction management (PERT networks), air-traffic management (taxiway networks), and emergency vehicle response scheduling can be investigated. The information transfer protocol under ATIS can be analyzed to determine the optimal means of disseminating real-time information to vehicles. Particular attention needs to be paid to the user-behavior factor. The algorithm developed in this study is deterministic in nature. Stochastic shortest path algorithms can also be developed for ITS applications. Another scope for future research is the development of an algorithm that could adaptively calculate routes based on current available information in a time-dynamic fashion.

The primary focus of this research study was on the development of shortest path algorithms. To further study the effectiveness of such algorithms as an effective routing tool for traffic assignment, the algorithm has to be incorporated within a complete dynamic traffic assignment procedure and a closed-loop simulation must be conducted, so that the feedback effects of traffic diversion on the *O-D* flows can be analyzed.

APPENDIX A

DEFINITIONS FOR REGRESSION RESULTS (TABLES 16-18)

Definitions for the Regression Coefficient Tables [NCCC 6.0 JR. User's Guide]

Regression Coefficient

The regression coefficients $\{B\}$ are the least squares estimates of the parameters. The value B_j indicates how much change in Y occurs for a one-unit change in X_j when the remaining X's are held constant.

Standard Error

The standard error of the regression coefficient (the standard deviation of the estimate).

T-Value (H_0: $B \equiv 0$)

This is the t-test value for testing the hypothesis that $B_j = 0$ versus the alternative that $B_j \neq 0$, after removing the influence of all other X's. This t-value has n-p-1 degrees of freedom, where

n = number of observations, and

p = number of dependent variables.

Prob. Level

This is the p-value for the significance test of the regression coefficient. The p-value is the probability that this t-statistic will take on a value at least as extreme as the actually observed value, assuming that the null hypothesis (H_0) is true (i.e., the regression estimate is equal to zero). If the p-value is less than α, say 0.05, the null hypothesis is rejected.

Decision (5%)

This is the conclusion reached about the null hypothesis. It will be either accept H_0 or reject H_0 at the 5% level of significance. Note that the level of significance was specified as the value of α.

Power (5%)

Power is the probability of rejecting the null hypothesis that $B_j = 0$ when B_j is actually non-zero. High power is desirable. High power means that there is a high probability of rejecting the null hypothesis when the null hypothesis is false. This is a critical measure of sensitivity in hypothesis testing.

R^2

R^2 is probably the most popular statistical measure of how well the regression model fits the data. R^2 may be defined either as a ratio or a percentage. Since we use the ratio form, its values range from zero to one. A value of R^2 near zero indicates no linear relationship between the Y and the X's, while a value near one indicates a perfect linear fit.

Lower - Upper 95% C.L.

These are the lower and upper values of a 100(1-α)% interval estimate for B_j based on a t-distribution with n-p-1 degrees of freedom.

Standardized Coefficient

Standardized regression coefficients are the coefficients that would be obtained if you standardized each independent and dependent variable. Here standardizing is defined as subtracting the mean and dividing by the standard deviation of a variable. A regression analysis on these standardized variables would yield these standardized coefficients.

T-Critical

This is the value of "t" used to construct the confidence limits.

Definitions for the Analysis of Variance (ANOVA) Table

Source

This represents the partitions of the variation in *Y*. There are four sources of variation listed: intercept, model, error, and total (adjusted for the mean).

DF

The degrees of freedom are the number of dimensions associated with this term. The degrees of freedom for the intercept, model, error, and adjusted total are 1, p, n-p-1, and n-1, respectively.

Sum of Squares

These are the sums of squares associated with the corresponding sources of variation. Note that these values are in terms of the dependent variable, *Y*.

Mean Square

The mean square is the sum of squares divided by the degrees of freedom. This mean square is an estimated variance.

F-Ratio

This is the F statistic for testing the null hypothesis that all $B_j \equiv 0$. This F-statistic has p degrees of freedom for the numerator variance and n-p-1 degrees of freedom for the denominator variance.

Prob. Level

This is the p-value for the above F test. The p-value is the probability that the test statistic will take on a value at least as extreme as the observed value, assuming that the null hypothesis is true. If the p-value is less than α (0.05), the null hypothesis is rejected. If the p-value is greater than α, then the null hypothesis is accepted.

Power (5%)

Power is the probability of rejecting the null hypothesis that all the regression coefficients are zero when at least one is not.

PRESS Value

PRESS is an acronym for prediction sum of squares. It is used to validate a regression model in predictability. To calculate PRESS, each observation is individually omitted. The remaining *n*-1 observations are used for regression and to estimate the value of the omitted observation. This is done *n* times, once for each observation. The difference between the actual *Y* value, Y_i, and the predicted *Y* with the i^{th} observation deleted, is called the prediction error. The sum of the squared prediction errors is the PRESS value. The smaller PRESS is, the better the predictability of the model. PRESS is especially popular in variable selection, where it is often used to compare various regression models.

Definitions for the Normality Test Table

Decision (5%)

This states whether the hypothesis of normality for the residuals is accepted or rejected at the given value of α (0.05).

APPENDIX B

PSEUDOCODE FORM OF ALGORITHM TD-*k*SP

Stage 1: Input

```
arc_count = 0
read origin and destination numbers, K, and M
while file od.dat is not empty          // count number of arcs in the network
do
skip (arc_class) size of bytes
arc_count = arc_count+1
end do

rewind od.dat
allocate memory for arc_count instances of the arc class

for i = 1 to arc_count
do
read fore_node, back_node, length and arc_no for arc i.
Read delay data from delay.dat          // also create delay and coefficient lists for each arc
end do
```

Stage 2: Preprocessing

```
NL = create_list()                      // a temporary list NL that stores node numbers
Set flags for nodes = FALSE             // FALSE: Not in  NL, TRUE: in NL
for i = 1 to arc_count                  // read CONNECTIVITY data
do
enter fore_node and back_node of arc i in list NL if not already there.
insert_node (NL,fore_node)                          // Try insert fore node in NL
insert_node (NL,back_node)                          // Try insert back node in NL
```

```
end do

n = get_length (NL)                   // number of nodes in the network
allocate memory for n node classes
set_node_no (origin,1)                        // assign node no = 1 for chosen origin
renumber (n)                          // renumber rest of the nodes in ascending order
destroy (NL)                          // deallocate memory for the list

for i = 1 to n          // obtain ADJACENCY data for each node
do
FS = create_forward_star_list(i)      // linked list of  FS arcs
for j = 1 to arc_count
if back_node (j) = i
insert_list(FS,j)
end do

initlialize labels ()          // set labels for nodes 2, ..n = 0
set_DQ_flags  ()               // initialize DQ flags = 0
```

Stage 3: Run the LC Algorithm using the double ended queue

```
Create_Q()                     // create the double ended queue Q
enque(Q,1)                     // enter origin into Q
while Q is not empty
do.
i = deque (Q)
        for each j belong to FS(i)
        ij = current FS arc
                for t = 1 to M
                        for k = 1 to K
```

```
                    flag = false
                    Lj = label(i, t, k) + delay(ij, t)
                    check = insert_label(Lj,j,K)          // store best K labels
                    // check also returns the position of insertion in the K vector
                    if check >0                    // if there was an update
                    pre (j, t, check) = ij         // store arc as a potential part of path
                    pre_K (j, t, check) = k        // also store k value of label(i, t, k)
                    if DQ_flag(j) < 2      // if j is not already in Q
                    enque(Q, j)            // enques j and sets DQ(j) = 2
                                           // position of insertion may be front or back
                    end  for k
                end for t
            end for j = FS(i)
end do.                                    // end of LC algorithm

module        delay (ij, t)                    //   pseudocode   for   time-dependent
delay
D = get_delay_list (ij)
while D is not empty and delay interval is not found
do
D_curr = current delay list element
get_bounds (D_curr)         / get time limits for current member
if t < Low or t > High      / / we are not in the right position
D_curr = D_curr->next       // continue to search next delay member
else                    // we are in the right position
C = get_coefficient_list(D_curr)      // get polynomial coefficients
degree = 0
delay = 0
        while C is not empty
```

```
        do
        ai = get_coefficient (C_curr)                    // get i^th coefficient
        delay = delay + ai * power (t, degree)
        degree = degree + 1
        C_curr = C_curr->next
        end do C
end do D
return  delay.
```

REFERENCES

1. Bazaraa, M. S, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows, Second Edition*. Wiley and Sons, New York (1990) .

2. Bellman, R. On a Routing Problem. *Quarterly of Applied Mathematics*. Vol. 16 (1958), 87-90.

3. Bertsekas, D., and R. Gallagher. *Data Networks*. Second Edition, Prentice Hall. Englewood Cliffs, N. J. (1992).

4. Booch, G. *Object-Oriented Analysis and Design with Applications, Second Edition*. Benjamin/Cummings Pub. Co., Redwood City, CA (1994).

5. Broadwater, R. *Object-Oriented Design for Engineers - EE 5984 Lecture Notes*. Virginia Polytechnic Institute and State University, Blacksburg, VA (1996).

6. Cook, K. L., and E. Halsey. The Shortest Route Through a Network With Time-Dependent Internodal Transit Times. *Journal of Mathematical Analysis and Application*. Vol. 14 (1966), 493-498.

7. Dial, R., F. Glover, D. Karney, and D. Klingman. A Computational Analysis of

Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees. *Networks*. Vol.9 (1979), 215-245.

8. Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Journal of Numerical Mathematics*, Vol. 1 (1959), 269-271.

9. Drew, D. R. *Traffic Flow Theory and Control*. McGraw-Hill, New York (1968).

10. Dreyfus, S. E. An Appraisal of Some Shortest Path Algorithms. *Operations Research*. Vol. 17 (1969), 395-412.

11. Friesz, T. L., D. Bernstein, T. E. Smith, R. L. Tobin, and Wie. A variational Inequality Formulation of the Dynamic Network User Equilibrium Problem. *Operations Research*, Vol. 41, No. 1 (1993), 179-191.

12. M. R. Garey, and D. S. Johnson. *Computers and Intractability*. W.H. Freeman. San Francisco, CA (1979).

13. Glover, F., D. Klingman, and N. Phillips. A New Polynomially Bounded Shortest Path Algorithm. *Operations Research.* Vol. 33, No. 2 (1985), 65-73.

14. Hall, R. W. The Fastest Path Through a Network With Random Time-Dependent Travel Times. *Transportation Science*. Vol. 20 (1986), 182-188.

15. Halpern, J. L. Shortest Route With Time-Dependent Length of Edges and Limited Delay Possibilities in Nodes. *Zeitschrift fur Operations Research*. Vol. 21 (1977), 117-124.

16. Kaufman, D. E., and R. L. Smith. Fastest Paths in Time-Dependent Networks for IVHS Application. *IVHS Journal*. Vol. 1 (1993), 1-11.

17. Koutsopoulos, H. N. and H. Xu, An information discounting routing strategy for ATIS. *Transportation Research Part C* (*Emerging Technologies*),Vol.1C, no.3 (1994), 249-264.

18. Lawler, E. L. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York (1976).

19. Mahmassani, H. S. Development and Testing of Dynamic Traffic Assignment and Simulation Procedures for ATIS/ATMS Applications. *Technical report prepared for U.S. DOT, FHWA*. University of Texas, Austin (1994).

20. Malandraki, C. *The Time-Dependent Vehicle Routing Problem*. Doctoral Dissertation, Northwestern University (1989).

21. Miller, E. D., H. S. Mahmassani, and A. Ziliaskopoulos. Path search techniques for Transportation Networks with Time-Dependent, Stochastic Arc Costs. *IEEE*

International Conference on Systems, Man, and Cybernetics. Humans, Information and Technology. Vol. 2 (1994), 1716-1721.

22. NCSS Statistical Software. *NCSS 6.0 JR. User's Guide*. Kaysville, U. T (1996).

23. Orda, A., and R. Rom. Shortest-Path and Minimum-Delay Algorithms in Networks With Time-Dependent Edge-Lengths. *Journal of the Association for Computing Machinery*. Vol. 37 (1990), 603-625.

24. Orda, A., and R. Rom. Traveling Without Waiting in Time-Dependent Networks is NP-Hard. *Manuscript, Dept. of Electrical Engineering, Technion - Israel Institute of Technology*. Haifa, Israel (1989).

25. Sherali, H. D., K. Ozbay, and S. Subramanian. The Time-Dependent Shortest Pair of Disjoint Paths Problem: Complexity, Models and Algorithms. *Working Paper, Virginia Polytechnic Institute and State University*, Blacksburg, V.A (1996).

26. Sherali, H. D., L. D. Brizendine, T. S. Glickman, and S. Subramanian. Low Probability-High Consequence Considerations in Routing Hazardous Material Shipments. *To appear in Transportation Science*, Virginia Polytechnic Institute and State University, Blacksburg, V.A (1996).

27. Shier, D. R. On algorithms for finding the *k* shortest paths in a network. *Networks*. Vol.9 (1979), 195-214.

28. Skiscim, C. C. and B. L. Golden. Solving k-Shortest and Constrained Shortest Path Problems Effectively. *Network Optimization and Applications*, B. Shetty (Ed.), *Annals of Operations Research.* Vol. 20 (1989), 249-261.

29. Tarjan, R. E.. Data Structures and Network Algorithms. *Society for Industrial Applications of Mathematics*. Philadelphia, PA (1983).

30. Vliet Van, D. Improved Shortest Path Algorithms for Transport Networks. *Transportation Research*, Vol.12, no.1 (1978), 7-20.

31. Ziliaskopoulos, A. and H. S. Mahmassani. Time-Dependent Shortest Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications. *Transportation Research Record*. Vol. 1408 (1993), 94-100.