PB2002-104348

IDEA

Innovations Deserving
Exploratory Analysis Programs

Intelligent Transportation Systems Program

Implementable Dynamic Dispatching Algorithms for Carrier Fleet Operations: Demand Responsive Service for Standard Ground and Intermodal Freight Movements

Final Report for ITS-IDEA Project 72

Amelia C. Regan, University of California, Irvine CA

December 2001

TRANSPORTATION RESEARCH BOARD • NATIONAL RESEARCH COUNCIL

INNOVATIONS DESERVING EXPLORATORY ANALYSIS (IDEA) PROGRAMS MANAGED BY THE TRANSPORTATION RESEARCH BOARD (TRB)

This investigation by University of California, Irvine, CA was completed as part of the Intelligent Transportation Systems (ITS) IDEA program which fosters innovations in development and deployment of intelligent transportation systems. The ITS-IDEA program is one of the five IDEA programs managed by the Transportation Research Board (TRB). The other four IDEA program areas are: Transit-IDEA, which focuses on transit practice in support of the Transit Cooperative Research Program (TCRP), NCHRP-IDEA which focuses on highway systems in support of National Cooperative Highway Research Program, High Speed Rail-IDEA (HSR), which focuses on high speed rail practice, in support of the Federal Railroad Administration, and Transportation Safety Technology (TST), which focuses on motor carrier safety practice, in support of the Federal Motor Carrier Safety Administration and Federal Railroad Administration. The five IDEA program areas are integrated to promote the development and testing of nontraditional and innovative concepts, methods, and technologies for surface transportation systems.

For information on the IDEA Program contact IDEA Program, (phone: 202/334-3310, fax: 202/334-3471, http://www4.nationalacademies.org/trb/dive-idea.nsf)

The publication of this report does not necessarily indicate approval or endorsement of the findings, technical opinions, conclusions, or recommendations, either inferred or specifically expressed therein, by the National Academy of Sciences or the sponsors of the IDEA program from the United States Government or from the American Association of State Highway and Transportation Officials or its member states.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December	3. REPORT TYPE AND DATES COVERED Final Report April 1999 to August 1999

4. TITLE AND SUBTITLE Implementable Dynamic Dispatching Algorithms for Carrier Fleet Operations: Demand Responsive Service for Standard Ground and Intermodal Freight Movements	5. FUNDING NUMBERS
6. AUTHOR(S) Amelia C. Regan	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California 115 Administration Building Irvine, CA 92697	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Transportation Research Board/IDEA Program National Research Council 2101 Constitution Avenue NW Washington, DC 20418	10. SPONSORING/MONITORING AGENCY REPORT NUMBER TRB IDEA-ITS-72

11. SUPPLEMENTARY NOTES

The ITS-IDEA program is jointly sponsored by the Federal Highway Administration, National Highway Traffic Safety Administration, and the Federal Railway Administration of the US Department of Transportation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available to the public through the National Technical Information Service, Springfield, Virginia, 22161	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This project developed and tested algorithms for dynamic freight and fleet management associated with local trucking operations such as those supporting rail or maritime intermodal environments. Several assignment methods that mix global and local optimization techniques and require varying degrees of real-time updates were developed and tested. A GIS-based simulation environment was also developed that mimicked local operations of the largest rail-ground intermodal service provider in the United States. Tests were conducted to compare the performance of globally and locally optimal algorithms and heuristics and to examine the algorithm performance in problems of realistic size. Results indicate that globally optimal methods perform much better with respect to minimizing distance traveled and meeting customer deadlines than do the local optimal insertion, addition and reassignment heuristics. Work continues on improving the local assignment heuristics.

14. SUBJECT TERMS freight mobility, dispatching algorithm, optimal, heuristics, global, local, intermodal	15. NUMBER OF PAGES 34
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

Contents of Report

EXECUTIVE SUMMARY 3
INTRODUCTION 6
CONTEXT 6
RELATED RESEARCH 7
THE MYOPIC GLOBAL ASSIGNMENT PROBLEM 9
Notation 11
Formulation 11
Over-constrained and Under-Constrained Methods 12
Time Window Reduction and Partitioning 14
A Discretization Scheme with Monotonically non-Increasing Costs 15
Problem Testing 16
Tests 17
The Optimality of the Solutions 18
Tightness of Alternative Lower Bounds 20
Comparison of Solution Times for Three Lower Bound Alternatives 21
Tests of Larger Problems 22
CONCLUSIONS RE: MYOPIC ASSIGNMENT 24
MIXED LOCAL AND GLOBAL ASSIGNMENT TECHNIQUES 24
Mid-day Assignment and Re-Assignment 24
The Simulation Model 26
The GIS Component 26
Test Problems 27
CONCLUSIONS RE: MIXED ASSIGNMENT TECHNIQUES 32
REFERENCES 33

EXECUTIVE SUMMARY

This project is concerned with the development of operational strategies and dynamic dispatching algorithms time sensitive and rapidly-changing commercial vehicle operations. The algorithms are designed to use real-time information on vehicle locations, characteristics of service requests and traffic network conditions to improve the efficiency of freight mobility. These are specifically tailored to the needs of carriers providing service to intermodal freight operations though they are applicable to general truckload operations as well. The research involved the development of a new approximation algorithm for time-constrained vehicle routing and scheduling problems and examined the trade-offs between implementing global and locally optimal assignment methods.

Project Highlights

The project involved active collaboration with JB Hunt Transportation Inc. This trucking company was a natural partner in this research as they have the largest rail/ground intermodal operations in the US. The result of the project is a continued effort, sponsored by JB Hunt to develop and test a prototype dispatching system in several of their intermodal terminals early in the year 2000. In addition to supporting the research with data and regular access to company engineers and analysts, JB Hunt made two separate supporting awards totaling $65,000 to support this and related research on intermodal operations planning.

The project resulted in the development of a new and promising approximation algorithm for the time-constrained vehicle routing and scheduling problems. The approximation algorithm developed can be applied to any multiple Traveling Salesperson Problem with Time Windows (m-TSPTW) but is especially suited to problems in which the ratio of assigned tasks to servers in an period is relatively small. It was specially developed for the local truckload assignment problem and has very good average performance. Relative to other solutions developed for related problems over the years, the algorithm is extremely easy to implement. Anyone with an understanding of optimization and limited software engineering skills could perform the implementation which uses a commercial GIS package to obtain network travel times and Cplex or another similar commercial optimization tool.

A simulation framework was developed in order to allow for the testing of assignment algorithms and heuristics. The simulation framework combines a C++ model with the Transcad Geographic Information System. Transcad is used for both visual display and for off-line calculation of shortest network travel paths.

Extensive testing has been performed to compare the performance of globally and locally optimal algorithms and heuristics and to examine the performance of algorithms with problems of realistic size. Test reveal that globally optimal methods perform much better with respect to minimizing distance traveled and meeting customer deadlines than do the locally optimal insertion, addition and re-assignment heuristics examined to date.

However, lengthy discussion with trucking industry logistics managers and dispatchers have revealed that dispatchers prefer assignment methods that rely on making relatively small changes to schedules rather than rescheduling all unserved tasks several times during the day. For this reason work towards improving the local assignment heuristics continues.

A system for generating test problems was developed. Once a geographic region has been selected customer locations are identified, problems are generated by selecting customers randomly and assigning time windows from a distribution. Problems maybe static but drawn from a distribution or may be dynamic in that the timing of service requests are generated according to a Poisson process.

Both a library of test problems and the code to generate problems will be made available to other researchers via the web site at the Institute of Transportation Studies at the University of California, Irvine in March of 2000. The test library has been complete the summer of 1999 but it is being updated to allow for a higher degree of flexibility in problem generation.

Technology Transfer

The research described in this report has been presented at the meetings of the Transportation Research Board (TRB) at the Institute for Operations Research and the Management Sciences (INFORMS) and of the Optimization Days Conferences. Several other presentations were made at universities and to trucking company executives. These presentations are listed below:
"Real-Time Routing and Assignment for Trucking Operations", invited presentation, University of Texas at Austin, February, 1999.

"Routing and Assignment for Local Trucking Operations Supporting Intermodal Operations", Presentation to Executives and Managers of JB Hunt Transport, Inc., Irvine, February, 1999.

"Dynamic Routing and Assignment for Trucking Operations", invited presentation, University of Southern California, March, 1999.

"Real-Time Routing and Assignment for Trucking Operations", Optimization Days, Montreal, May, 1999.

"Local Truckload Vehicle Routing with Strict Time-Window Constraints", INFORMS, Philadelphia, November, 1999.

"Mixed Global and Locally Optimal Heuristics for Local Truckload Trucking Operations with Strict Time-Windows", 79th meeting of the Transportation Research Board, Washington DC, January 2000.

"An Algorithm for the Local Truckload Vehicle Routing Problem with Strict Time Window Constraints", 79th meeting of the Transportation Research Board, Washington DC, January 2000.

Two papers discussing this work have been submitted to academic journals. These are:

Regan, A.C., S. Jagannathan, and X. Wang (2000), "Mixed global and local assignment algorithms for quasi dynamic local truckload trucking operations with strict time-windows", *Transportation Research Record*, forthcoming.

Wang, X and A.C. Regan (2000), Local truckload vehicle routing with hard time window constraints, *Transportation Research, Part B,* under review.

The research problem, methods and results found are detailed in the rest of this report.

INTRODUCTION

This report examines the trade-offs associated with local and global, but myopic, assignment heuristics for local truckload trucking operations such as those associated with drayage operations near intermodal facilities. These operations involve a combination of loads that are known at the beginning of the day and those that arrive dynamically throughout the day. Some of the dynamically arriving loads are revenue generating moves while others are trailer, chassis or container repositioning moves. Since a significant fraction of the day's loads are known a priori, dispatchers would like to be able to construct schedules for the day and then to make minor changes to these schedules as the day progresses. We examine the efficiency of an operation in which new loads are added to or appended to schedules constructed at the start of the day versus one in which the whole system is re-optimized several times during the day. The re-optimization method does not seek to preserve current schedules while the local optimization techniques do. The examination of solutions is performed using a geographic information system (GIS) based simulation model developed for this purpose.

CONTEXT

The context of this research is local truckload trucking operations in which a driver moves a single load at a time from its origin to its destination. Each load has a specified time-window within which it has to be picked up at the origin location and delivered at the destination location. Local truckload operations contain a fairly high degree of stochasticity. Carriers typically know only a portion of loads to be served at the start of the day. Further, unexpected delays at intermodal terminals or customer locations can require the reassignment of previously assigned loads. In addition the need to reposition containers, trailers or chassis can arise at any time, effectively adding new moves to the system. Decisions to accept (or reject) newly requested loads and assign a vehicle to serve an accepted load take place very quickly. In local operations, the acceptance decision determines whether a load is moved by a company's drivers or contracted to another company. In the context of primary interest in this work, rail and maritime intermodal operations, there are many dray operators available. These operators specialize in taking last minute requests for relatively short moves. Decisions made in the present affect the future state of the system. Since the decisions have to be made in real-time, the speed of decision making is extremely critical.

This research has two equally important goals. The first is to develop assignment strategies suitable for real-time implementation. The second is to examine the costs associated with two customer service-driven operational strategies. One of these strategies is to try to maintain schedules developed early in the day and to limit changes to fairly simple ones. The trade-offs between implementing global optimization techniques that minimize the overall cost to provide service but may make significant changes to previous schedules and those which make only "local" changes to schedules (insertions, additions and removal of at most one load at a time) are examined. Dispatchers favor solutions with few major mid-day changes so that repositioning moves and extra driver tasks not included in the assignment problem may be scheduled and so that drivers may predict at the beginning of the day to within 30 minutes or an hour when their work day will end. The other strategy is one in which sub-fleets of drivers are fairly small and fairly stable, meaning drivers work in the same area every day and pickup and deliver to the same set of customers. The advantage here is that drivers become very familiar with the street network and traffic pattern in the relatively small geographic region to which they are assigned. This familiarity saves them time and, perhaps more importantly, can reduce the likelihood of accidents. In addition they develop good relationships with major customers in their primary zones. These relationships may facilitate reductions in both the length and variability of dock times.

The first of these operational considerations is examined in this paper. The second is the subject of on-going research motivated in part by research that has shown that significant network economies of scale (and density) exist in truckload drayage services (Walker, 1992). Whether large companies take advantage of these economies despite sub-fleet and sub-area partitioning is of significant interest. A geographic information system (GIS) based simulation model that is integrated with a CPLEX based optimization model has been developed for this purpose.

RELATED RESEARCH

The long haul truckload trucking problem has attracted considerable attention in the past few years. The dynamic vehicle allocation problem has been attacked from many angles incorporating present and future (forecast) demands, deterministic and stochastic variables by optimization-based and heuristic algorithms. Powell (1996) presents various formulations and solution methods for this problem which is typically includes long distance moves and a longer planning horizon. This research differs in that we restrict ourselves to local truckload moves such as those in and around intermodal facilities and to a set of work that must be completed during a single 24-hour period. Demands arise within a compact geographic region near one or more intermodal facilities (rail or maritime facilities or both). These problems are simpler in some respects than the traditional dynamic vehicle allocation problems in that we do not consider repositioning moves made in anticipation of future demands. Vehicles are busy, waiting at an intermodal facility or waiting at the depot for an assignment. In certain respects however, these problems are more complicated than the traditional vehicle allocation problems. Loads have strict time window constraints, and dock times (loading, unloading and waiting times) may be unpredictable as can service times at intermodal facilities. These stochastic elements, combined with travel times subject to recurring and non-recurring congestion have a significant affect on the ability of dispatchers to assign a driver a full day's work, even if one hundred percent of the days' demands are known at the start of the day.

This problem lends itself to formulation as a vehicle routing problem with time windows but differs from typical VRPTW problems in two ways. First, the moves are full truckload moves and do not allow in-vehicle consolidation. Second, in this problem the vehicles are assigned only a small number of moves during the planning horizon which is typically a single work day.

These problems fall into a class of those in which it is difficult to accurately describe the fluctuation of demands and service times and in which it is not cost effective to make the effort to characterize and explicitly include stochastic elements in the solution. In real time applications, trade-offs between computational complexity and solution quality exist. The complexity of accurately modeling uncertainty and the complexity of algorithms which explicitly consider stochastic elements justifies the use of a deterministic vehicle routing model as an important part of the strategies used to make online (real-time) assignments. Regan, Mahmassani and Jaillet (1996, 1998) presents a set of heuristics for real-time assignment and routing for dynamic carrier fleet operations. That work assumes that no demands are know a priori and that loads must be assigned (in order to ensure time-window feasibility) immediately after they are requested. Furthermore, the assignment rules rely on purely local optimization techniques, which miss out on system-wide assignment opportunities. The approach described in this research assumes that a significant fraction of demands are known at the beginning of the assignment period and seeks to take full advantage of this information while at the same time retaining flexibility to react to changes if need be. Yang, Jaillet and Mahmassani (1999) extended earlier analysis considerably, and developed a global optimization-based formulation of the real-time truckload pickup and delivery problem which they call myopic because it involves only information known at the time of solution in a highly dynamic environment. Their work suggests that even a myopic system-wide optimization technique performs better than purely local assignment techniques. The problem they solve corresponds to our problem. However they examine systems which are even more dynamic. Because this method is intended for eventual implementation in operations, we assume, as is typically the case in real operations, that a large fraction of demands are know at the start of day. In addition, the research described in Yang *et al.* (1999) has as its focus smaller problems as it was intended to develop new insight into dynamic problems rather than lead to an operational system.

The real time *local* truckload pickup and delivery problem has attracted relatively little attention from the research community. Until recently, few carriers had intermodal operations of the size inviting the development of automated routing and scheduling systems and the large local pickup and delivery problems faced by private fleets typically involved primarily fixed routes. Examples of work on this topic are found in Ball, Golden, Assad and Bodin (1981), Powell and Gittoes (1996) and Powell, Snow and Cheung (200X). The work by Ball *et al* develops route construction and improvement heuristics for truckload vehicle moves while Powell *et al* develop and test, fairly extensively, near optimal assignment heuristics for truckload vehicle operations, including short haul operations. Finally, Powell, Towns and Marar (200X), examine the performance of global verses local assignment techniques for truckload trucking operations when dispatchers reject some fraction of the assignments provided by the dispatching system.

The motivation for this research is the intermodal operations of one of the largest truckload carriers in the US. The company uses optimization software to assist with the development of schedules for their long haul (over-the-road) drivers but not for their local operations. Current assignment methods rely on dispatchers (load managers) to solve, without the assistance of a scheduling system, what is essentially a bipartite assignment problem at the beginning of the day followed by nearest-load assignments for the rest of the day. The variability in handling and travel times in congested urban networks coupled with some uncertainties about equipment availability have made this assignment method the norm in most local operations.

In addition, local operations are driven by many somewhat intangible factors including customer service and safety constraints that favor sub-fleets of relatively few drivers working in the same areas and with the same customer set from day to day. These operations have been historically fairly well managed by dispatchers. However, a sharp increase in recent years in the use of rail intermodal transportation has led local operations to become much more complex and increasingly large, inviting the development of computer aided dispatching systems. In addition to involving more than a hundred drivers and hundreds of loads everyday, these problems increasingly include more than one rail terminal and a fairly wide geographic region.

THE MYOPIC GLOBAL ASSIGNMENT PROBLEM

The problem solved is a myopic version of the truckload vehicle assignment problem with time window constraints, which does not anticipate the future and simply assigns the vehicles to serve as many known loads as possible. Because travel takes place in a compact region, it is not necessary to consider the future locations of vehicles as is typical in the long haul version of this problem (see for example Powell, 1988). Of course, local operations are in fact somewhat dynamic in nature. A fraction of loads to be moved in a given day become known only a short time before service must take place, trailer repositioning moves are added to the system as the day progresses and loads must sometimes be reassigned due to traffic, customer dock and intermodal facility delays. The assumption we make is that the assignment problem will be resolved several times as the day progresses and more information becomes known.

If we treat each loaded trip as a node, the problem may be viewed as an asymmetric multiple traveling salesman problem with time window constraints (m-TSPTW). A major difficulty to solve the m-TSPTW problem arises from the time window constraints.

There are generally three classes of approximation methods used to deal with the time windows. One explicitly considers the time window constraints in the construction of routes. This class of methods includes Dantzig-Wolf decomposition, which is used to decompose the coverage constraints, Lagrangian relaxation, in which the coverage constraints are relaxed, and state space relaxation in which the feasible space of a dynamic programming algorithm is reduced (see for example Kolen Rinnooy Kan and Trienekens, 1987). Both Dantzig-Wolf decomposition and Lagrangian relaxation lead to shortest path sub-problems with time window constraints, which has been shown to be NP-hard (Dror, 1994).

The second class of methods is the relaxation of the time window constraints. Network relaxation methods solve a network problem after relaxing the time window constraints and then partition the windows according to the last infeasible solution. According to Desrosiers *et al* (1986), this method is inferior to the Dantzig-Wolf decomposition method. Lagrangian relaxation of time window constraints is reported to generate results worse than those keeping the time window in the sub-problems (Desrosiers, Sauve and Soumis, 1983).

A third class of methods to deal with the time window constraints is to discretize them. The idea is to replace continuos time constraints and individual assignment variables with a bundle of assignment variables, each corresponding to a different point in time. An early example of this approach is seen in the work by Appelgren (1969, 1971) in which a ship scheduling problem is solved. A paper by Levin (1971) uses the same strategy to generate flight assignments in which each move has a set of alternate service times. The method described in this paper is inspired by the work of Appelgren and Levin but differs in several ways. In the work by Appelgren the service time windows are not actually continuous variables. Shipments must begin on exactly one day and travel times are naturally expressed as integer multiples of days. The paper by Levin introduces the notion of "bundles" of flow variables. However, no attempt is made to address the issue of how many flow variables the bundles should contain or to examine the trade-offs between coarser and finer discretization.

A more recent application of time window discretization can be seen in Swersey and Ballard (1984), where a school bus scheduling problem is solved using a time window discretization method to minimize fleet size. Graham and Nuttle (1986) compare the performance of a time window discretization method against two heuristics for solving the school bus scheduling problem and found that it had good results. The main complaint about the method was that due to computational issues of the time an LP relaxation of the problem was solved and that when this did not have an integer solution that manually adjusting the non-integer variables could be difficult.

Discretization methods have been rarely used in recent years. The likely reason for this is that the method results in an exponentially expanded network. However, recent advances in computing have made this method more attractive than in the past. In addition, we show later that the relatively small number loads assigned to each vehicle at any given time makes this problem well suited to this method.

In this work we develop an iterative method for solving m-TSPTW problems using time window discretization. At each iteration we generate and solve an over constrained version of the problem and an under constrained version. The over constrained problem provides us with a feasible solution and an upper bound on the cost of the optimal solution. The under constrained problem provides us with a lower bound on the cost of the optimal solution. As far as we know, no other researchers have provided such a bound.

We develop and implement a scheme in which the solution is guaranteed to be non-increasing in subsequent iterations.

The organization of this section of this report is the following: first we introduce the formulation, then we then present the over-constrained and under-constrained problems formulations. Next, we introduce the time window partitioning method in which non-increasing costs are guaranteed and present some empirical results.

Notation

Let:

N	=	the set of nodes for loads,
K	=	the set of vehicles,
o_i	=	the starting node for vehicle i,
a_i, b_i	=	the beginning and end of the time window for load i.
T_i	=	the service time for load i,

t_{ij} = the time needed to service load i and then travel to the pickup location of load j (the handling time at load i, the loaded travel time for load i and the empty travel time between the destination of load i and the origin of load j),

c_{ij} = the cost of travel from the destination point of load i to the origin of load j,

M = an infinitely large constant.

The flow variables are present in the problem if a feasible assignment of a vehicle from its starting location and each load and between loads is possible. The flow variable X_{ij} is equal to one when there is an assignment in which load j is served by the vehicle departing node i; it is equal to zero, otherwise.

Formulation

$$\text{obj min} \sum_{i \in N+\{o_k|k \in K\}} \sum_{j \in N \setminus \{i\}} (-M + c_{ij}) x_{ij} \qquad (1.0)$$

$$\sum_{j \in N} x_{o_i j} \leq 1 \qquad \forall i \in K \qquad (1.1)$$

$$\sum_{i \in N+\{o_k|k \in K\}} x_{ij} \leq 1 \qquad \forall j \in N \qquad (1.2)$$

$$\sum_{i \in N+\{o_k|k \in K\}} x_{ij} - \sum_{m \in N} x_{jm} \geq 0 \qquad \forall j \in N \qquad (1.3)$$

$$x_{ij}(T_i + t_{ij} - T_j) \leq 0 \qquad (1.4)$$

$$a_i \leq T_i \leq b_i \qquad \forall i \in N \qquad (1.5)$$

$$x_{ij} \text{ is binary;} \quad \text{for all } i \in N + \{o_k|k \in K\}, j \in N, i \neq j \qquad (1.6)$$

Constraints (1.1) require the vehicles to leave each load at most once. Constraints (1.2) indicate that each load be served at most once. Constraints (1.3) say that a vehicle departs from a load only if it serves the load first. Constraints (1.4) enforce the temporal relationship of consecutive loads. Constraints (1.5) specify the time window constraints. Constraints (1.6) are the binary constraints.

The objective function is a multi-objective one. The infinitely large value M is a sufficiently large constant that ensures that the assignment covers as many feasible loads as may be possibly served.

The problem in this paper is slightly different from the typical m-TSPTW problem. The vehicles are not required to return to depot after each service during the day. There is no a priori guarantee that each load could be served. The objective is to serve as many loads as possible. These differences are in fact trivial. It is easy to show that by adding dummy vehicles, dummy source (sink) node and links of zero cost that this problem may be transformed to the typical m-TSPTW.

Over-constrained and Under-Constrained Methods

The traditional way to deal with the non-linear time window constraints using integer programming is to linearize them. However, the linearized constraints are very loose because they are not the facets of the polytope of the convex hull of the feasible solutions (Langevin, Soumis and Desrosiers, 1990). In the method we present here the time constraints are taken into account in a pre-processing step in which two versions of the problem are constructed. The first is over constrained and the second is under constrained.

The flow variables X_{ij} correspond to links from vehicles to loads, and those between loads. Possible links are determined by the time window constraints associated with each load. If the time window is a single time point, then the problem is reduced to a fixed schedule problem (Derosiers *et al,* 1995). This kind of problem has a clear and exact network representation and can be solved very efficiently.

Suppose we consider only the end points of the time window of both the first and the second load when we set up links between two loads for the network; then we obtain a network that ignores some possible links. We refer to this method as the over constrained method. Suppose we only consider the starting point of the time window of the first load and the end point of the time window of the second load when we set up the link between two loads; then we end up with a network that includes some infeasible links. We refer to this method as the under constrained method. The over constrained method leads to a network from which we obtain a feasible solution while the under constrained method provides a lower bound for the solution.

Figures 1 and 2 show the feasible links excluded and infeasible links included in the two methods.

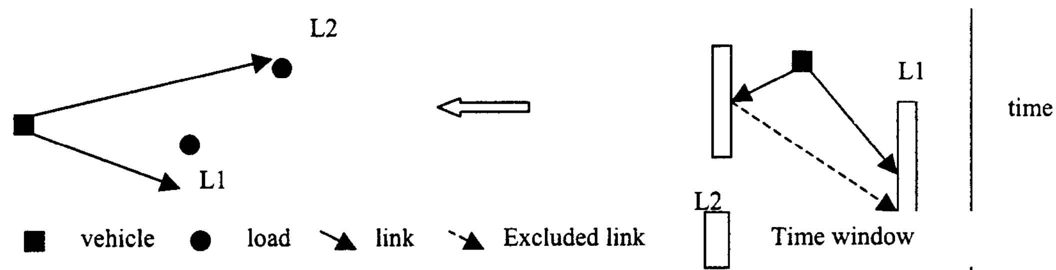


Figure 1) Over-constrained network

It is not possible to reach load 1 after leaving at the latest time in the time window for load 2 or to reach load 2 after leaving at the latest point in the time window for load 1. However, as is shown above, it is possible to serve load 1 after serving load 2 in the early part of the time window for load 2.

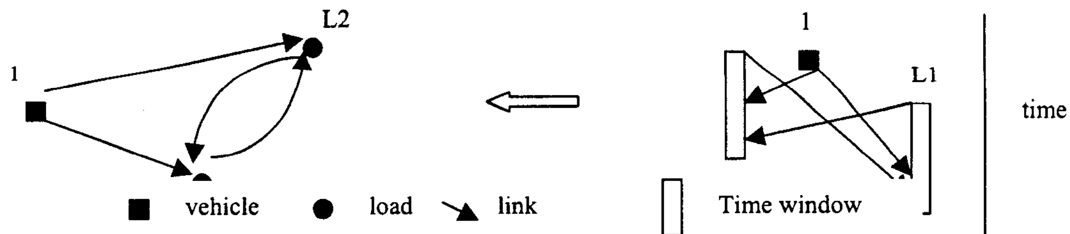


Figure 2) Under-constrained network

It may be observed that the under constrained network might permit infeasible links. If the vehicle leaves load 1 at the end of its time window it will have no way to reach load 2 within its time window. In fact, this network may contain cycles.

In the over constrained method we replace constraints (1.4) and (1.5) with the following:

$$X_{ij}(b_i+t_{ij}-b_j) \le 0 \qquad (1.7)$$

In the under constrained method we replace constraints (1.4) and (1.5) with the following:

$$X_{ij}(a_i+t_{ij}-b_j) \le 0 \qquad (1.8)$$

Constraints (1.7) define the links for the network of over constrained method. Constraints (1.8) define links of network from under constrained method. If the coverage constraints (1.1) are relaxed, the formulation from the over-constrained method is a network flow problem on an a-cyclic network. There are very efficient algorithms to solve such problems. However the network generated using the under constrained method is likely to contain cycles as can be seen in figure 2.

If we use the formulation from the under-constrained method, there is some infeasible space included in the solution space. While for over-constrained problem, there is some feasible space excluded. As a result, the optimal solution to the linear relaxation of the formulation by the under constrained method Z_l^{LP}, the integer solution to the under constrained method Z_l^{IP}, over constrained method Z_m^{IP} and the global optimal integer solution Z_o^{IP} can be placed in the following order:

$$Z_l^{LP} \leq Z_l^{IP} \leq Z_o^{IP} \leq Z_m^{IP}$$

Time Window Reduction and Partitioning

In general, the bigger the time windows, the bigger the gap between Z_l^{IP} and Z_o^{IP}, as well as between Z_o^{IP} and Z_m^{IP}. Sometimes the gap between the two methods is so large that we cannot determine if we have reached an acceptable solution with respect to the optimal value.

The partitioning method is based on the observation that if the time windows are smaller, the gap between Z_m^{IP} and Z_l^{IP} is reduced. To do this, the original window is partitioned into several parts. Each part is considered as a sub-load. At most one of the sub-loads of any load may be served. The vehicle that leaves a sub-load must have entered to the same sub-load. In this way, the number of feasible links excluded by the over constrained method is reduced; similarly, the number of infeasible links included in the under constrained method is also reduced.

The problem is how to select the width to partition the time windows because smaller widths lead to much larger problems. The iterative solution method in which an upper and lower bound is obtained at each time iteration allows us to begin by solving problems of reasonable size. If the ratio between lower bound and upper bound is unacceptable then the width selected is too large. In that case we select a smaller partition and solve the problem again. In the tests problems presented, a series of widths is arbitrarily selected to start with two hours and to end with 0.1 hours. The way to partition is as follows. Suppose the pre-selected width for partitioning is d, and that the load has a window (a_i, b_i). First, determine the number of sub-loads for this iteration by taking the smallest integer that is greater than (b_i-a_i) divided by d. That is ceil[(b_i-a_i)/d] where ceil(x) is the ceiling function that finds the smallest integer greater than x. Then partition the window into this many parts evenly.

This method is further improved by employing time window reduction methods commonly used for preprocessing VRPTW problems. There may be some part of the time window, which is of no use to the assignment since no vehicle is able to reach the load within that time. That part of the window is eliminated in order to reduce the size of the problem. This method is described in Desrochers, Desrosiers and Solomon (1992). From here on, we refer to reduced time windows when we mention them. Then the formulation after window partitioning can be modified as follows:

$$\text{obj Min} \sum_{\substack{i \in N+\{o_k|k\in K\} \\ \delta(i)\neq\delta(j)}} \sum_{j\in\omega\setminus\{i\}} (-M+c_{ij})X_{ij} \tag{2.0}$$

$$\sum_{\substack{j\in\omega \\ \delta(j)\neq\delta(i)}} X_{ij} \leq 1 \qquad \forall i \in \{O_k \mid k \in K\} + \omega \tag{2.1}$$

$$\sum_{\substack{i\in\omega+\{O_k|k\in K\} \\ \delta(i)\neq\delta(j)}} X_{ij} \leq 1 \qquad \forall j \in \omega \tag{2.2}$$

$$\sum_{\{j|\delta(j)=\xi\}} \sum_{\substack{i\in\omega+\{O_k|k\in K\} \\ \delta(i)\neq\delta(j)}} X_{ij} \leq 1 \qquad \forall \xi \in N \tag{2.2'}$$

$$\sum_{\substack{i\in\omega+\{O_k|k\in K\} \\ \delta(i)\neq\delta(j)}} X_{ij} - \sum_{\substack{m\in\omega \\ \delta(m)\neq\delta(j)}} X_{jm} \geq 0 \qquad \forall j \in \omega \tag{2.3}$$

$$X_{ij}(T_i + t_{ij} - T_j) \leq 0 \qquad \forall (i,j) \in A, \delta(i) \neq \delta(j) \tag{2.4}$$

$$a_i \leq T_i \leq b_i \qquad \forall i \in \omega \tag{2.5}$$

$$X_{ij} \text{ is binary} \quad \forall i \in \omega + \{O_k|k \in K\}, j \in N, i \neq j \tag{2.6}$$

$\delta(i)$ denotes the load associated with sub-load i. ω is the set of all sub-loads. O_k is the node for vehicle k. K is the set of vehicles. Constraints (2.2') stipulate that at most one sub-load be served for each load. a_i and b_i represent the beginning and end of the time window of the sub-load i. In the same way, the over constrained and under constrained methods define the links between sub-loads. After constraints (2.4) and (2.5) are replaced with definite links, the formulation possesses a structure that has a network flow sub-problem after decomposing or relaxing the constraints (2.1), (2.2) and (2.3). In this paper, a branch and bound, method is used to solve the problems by using Cplex®.

A Discretization Scheme with Monotonically non-Increasing Costs

Another observation is that a smaller length to partition the window usually but not necessarily leads to a better feasible solution. An example here shows this point.

Figure 3) Selection of the discretization point

Figure 3 shows that if the time window at the left is partitioned into two parts, there could be the assignment in which the vehicle goes to two loads in a row as shown by arrows in the figure. But if a smaller width is adopted and the longer window is partitioned into three parts the over constrained method would not allow this assignment. The vehicle can only serve one load, which is worse than the solution from a larger partitioning width.

To guarantee that the over constrained method lead to no worse solutions in subsequent iterations, a special scheme is used. First partition the time window into two parts at the optimal service time from last iteration. Then use the pre-selected width to partition both parts of the window. In addition, we include a special sub-load whose time window is limited to the first time point of the load's time window.

Here we summarize the procedure used:

1. Select a series of widths used for partitioning.
2. Partition the window into two parts at the time points where the service is delivered at last iteration from over constrained method, then select the first unused width in the pre-selected series to partition the two parts; at the first iteration, partition the whole window directly. Add the first time point of the original window as a sub-load.
3. Generate over and under constrained formulations and solve the two formulations.
4. If the ratio between lower bound and upper bound is acceptable, no smaller width can be used, or machine time is run out, stop; otherwise, select the next width to be used and return to step 2.

Problem Testing

Using the GIS package TransCAD, we generated a set of representative problems based on real data. The problem generation package is part of a larger GIS based fleet management simulation model described in Jagannathan (1999) and Regan, Jagannathan and Wang (2000). The loads are generated by selecting randomly from known customers in the service area. Time windows are randomly assigned based on the distribution shown in table 1, which roughly corresponds to the time windows associated with loads known at the start of day.

Table 1. Probabilities associated with time windows of varying length

Time window	7:00-7:30 AM (0.5 hours)	8:00-9:30AM (1.5 hours)	8:00AM-12:00PM (4 hours)	12:00-5:00PM (5 hours)
Probability	0.1	0.15	0.35	0.4

For the problems in the test set, we begin with the vehicles at the depot (which in this problem is very near the rail yard) and make all vehicles available for the duration of the day. Vehicles are not required to return to depot after each service. Travel distances correspond to the shortest network travel distance. Travel time is assumed to be 35 miles per hour, reflecting congestion levels in the test region. The average loaded distance is quite short, less than twenty miles long. A handling (typically dock time) of forty minutes is assumed for each load though in an actual operation the handling time would be customer specific. We present results from 30 problems of 20 vehicles and 75 loads. Twenty was selected because it is the typical maximum size of a local sub-fleet handled by a single load manager (dispatcher). In the test, the cost is the empty travel distance. We set the parameter M to 10000 in the test since it is much larger than even the sum of the total empty travel distance. We then present results from 30 similarly generated problems of 40 vehicles and 150 loads. For these larger problems two observations are made. The first is that these larger problems reach optimality in earlier iterations, on average. The second is that for larger problems only limited iterations are possible because the size of the problems expands too quickly.

The widths to partition the time window at iterations are as follows:

Table 2. Partitioning width

Iteration	1	2	3	4	5	6	7	8	9	10	11	12
Width (hours)	2	1.75	1.5	1.25	1.00	0.75	0.6	0.5	0.4	0.3	0.2	0.1

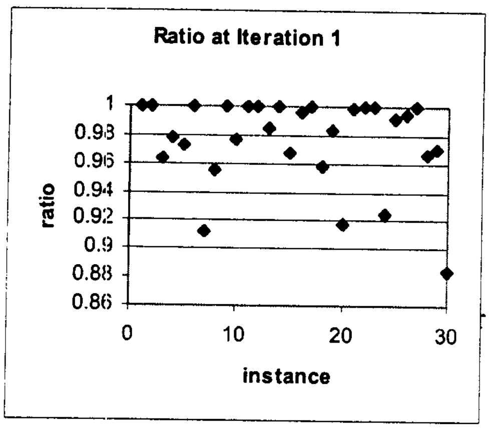
We solve the problems using CPLEX® version 5.0 with no special modifications. The later versions of this software (6.0 and 6.5) are known to have faster solution times, particularly for Mixed Integer Programming (MIP) problems. Therefore, exact times presented here should be considered relative solution times. In addition, the solution times could be further improved by the addition of a more sophisticated decomposition or relaxation method. One of the goals of this research was to develop a method simple enough to be implemented by engineers with only limited training in algorithm development. This method can be implemented by anyone with limited programming skills and an understanding of the workings of commercial optimization software.

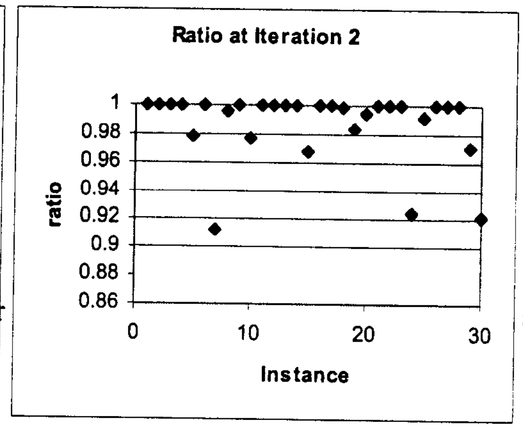
Tests

All tests are run on a desktop computer, a 400 MHZ Pentium II PC with 256 MB RAM. Three aspects of the method are tested. The optimality of the upper bound solution (feasible solution) in terms of the ratio of the cost of the lower bound to upper bound; the tightness of the lower bound as opposed to two other alternatives and their corresponding machine time.

The Optimality of the Solutions

The ratio of the cost of the solutions associated with the under and over constrained formulations after the first two iterations are provided in table 3 and figure 4. It may be observed that even when a width as large as two hours is used to partition the windows the solutions are quite good. Here the cost does not include the term M.





and 2

Table 3. The ratio of the cost of lower and upper bound in iterations 1 and 2

Problem Instance	Iteration 1	Iteration 2
1	1.0	NA
2	1.0	NA
3	0.96401	1.0
4	0.97838	1.0
5	0.97305	0.9782
6	1.0	NA
7	0.91203	0.91203
8	0.95538	0.99594
9	1.0	NA
10	0.97717	0.97717
11	1.0	NA
12	1.0	NA
13	0.98536	1.0
14	1.0	NA
15	0.96806	0.96806
16	0.99618	1.0
17	1.0	NA
18	0.95863	0.9993
19	0.98326	0.98358
20	0.91734	0.99524
21	0.99821	1.0
22	1.0	NA
23	1.0	NA
24	0.92423	0.92423
25	0.99227	0.99227
26	0.99518	1.0
27	1.0	NA
28	0.96672	0.96996
29	1.0	0.96996
30	0.88405	0.92211

Tightness of Alternative Lower Bounds

Because the time required to solve for the lower bound at each iteration may be long, two alternative methods are considered for the generation of a lower bound. The first is to solve the LP relaxation of the original formulation (1.0)-(1.6). We call this solution L1. The other alternative is to solve the LP relaxation of the under constrained formulation (2.0)-(2.6). We call L2. In this formulation the time window constraints are linearized as in Desrosiers, *et al*, (1986). Equations (2.4) are replaced with equations (2.4').

$$X_{ij}(T_i + t_{ij} - T_j) \leq 0 \qquad \forall\ (i,j) \in A, \delta(i) \neq \delta(j), \qquad (2.4)$$

$$T_i + t_{ij} - T_j \leq (1 - X_{ij})M \qquad \forall\ (i,j) \in A, \delta(i) \neq \delta(j), \qquad (2.4')$$

Figure 5 compares the tightness of each of these bounds for iterations 1-6, presenting only problems that remain unsolved in each iteration. We refer to the under constrained method in this figure as UC. It may be observed that the improvement obtained by using the lower bound from the under constrained, and more computationally expensive method is limited, on average. However this does not exclude the possibility that in some instances the under constrained method provides a much tighter lower bound than its LP relaxation. In one of the thirty problems examined the gap between the two lower bounds after two iterations was more than 5%. This suggests that if this method is applied in off-line situation where the solution times are allowed to be fairly long, that implementing the tighter lower bound has considerable benefit. However, if the method is used in an on-line situation then the bound associated with the LP relaxation of the under constrained formulation should be used because the limited improvement in the lower bound is not worth the corresponding increase in solution time. Three of the thirty problem instances are not included in the table because we were unable to obtain the optimal solution for these problems despite obtaining tight bounds for these solutions.

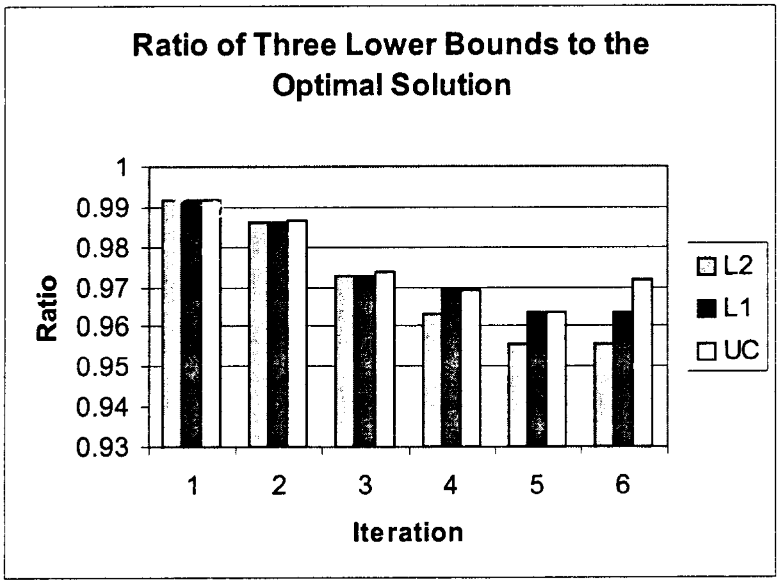


Figure 5) Comparison of Three Lower Bounding Method

Table 4) Number of Unsolved Problems in Iterations 1-6

Iteration	1	2	3	4	5	6
Width	2	1.75	1.5	1.25	1	0.75
Number Unsolved	30	19	10	7	5	5

Comparison of Solution Times for Three Lower Bound Alternatives

In the operation of interest here, solutions must be obtained rapidly. Figure 6 presents the cumulative solution times for iterations 1-6 for the three methods, L1, L2 and UC (under constrained IP). The cumulative time includes the solution time for the under constrained problems and over constrained problems.

It is worth mentioning here that it is very difficult to solve the original integer programming formulation of this problem directly. For the 30 problems examined, none could be solved using a standard branch and bound method directly. We contrast this with the fact that all of them can be solved in under constrained formulation by branch and bound method within a couple of minutes.

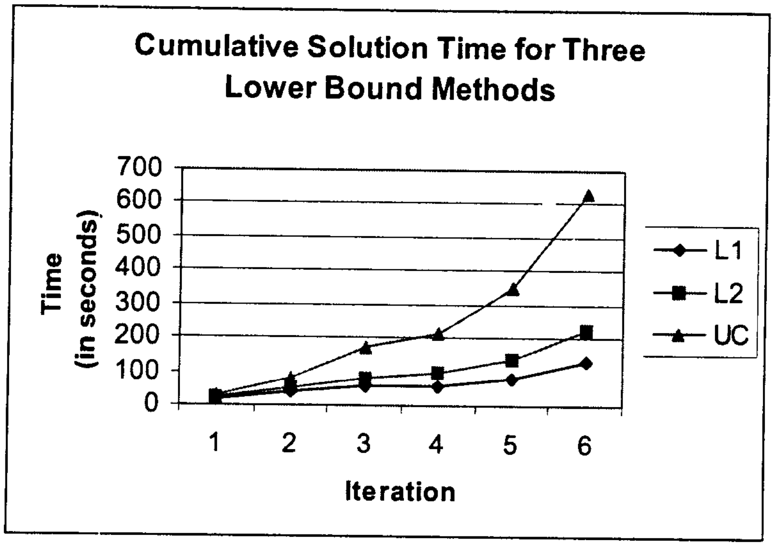


Figure 6) Cumulative solution times for iterations 1-6

Tests of Larger Problems

A natural question is how this method performs on larger problems. According to the same distribution as in the 30 problems just tested, we generated thirty more instances of 40 vehicles and 150 loads. The solution times for these problems were relatively long. In the system this method is intended to support solution times must be no more than 30 minutes. Less than five minutes is preferable because dispatchers will sometimes disagree with a solution found (due to information available to them but not available to the decision support system) and will need to remove some loads for assignment outside the optimization framework. For that reason we examine only the solution that may be obtained for the larger problems within this time constraint. We present the results obtained after the first iteration.

Figure 7 shows the ratio between the lower bound and upper bounds for the larger problems. It may be observed that the discretization method appears to work better on larger problems than on small ones. Even with a width of two hours, many of the problems are solved to optimality and most are solved to within a gap of 2%.

The average time required to obtain the first feasible solution (the solution of the over constrained formulation) for the thirty problems was 438.8 seconds. The average time required to generate the lower bound using the under constrained method was 255.8 seconds. The under constrained method takes less time because the solution obtained from the over constrained formulation is used as a cut off point for the branch and bound algorithm. The time required to solve the LP relaxation L1, in order to find a corresponding lower bound is less than one minute. For three of the thirty problems we were unable to solve for the lower bound from the under constrained formulation. However, for these problems the cost of the LP relaxation L1 was exactly equal to the cost of the feasible solution (the optimal solution was found). For the 27 problems for which we were able to obtain the bound associated with the first iteration of the under constrained problem, the value of these lower bounds were exactly the same. Figures 7 and 8 show the gap between the upper and lower bounds and also the solution times required to find the first feasible solution for each of the 30 larger problems. Figure 8 shows the solution times for over constrained formulation (feasible solution) only.

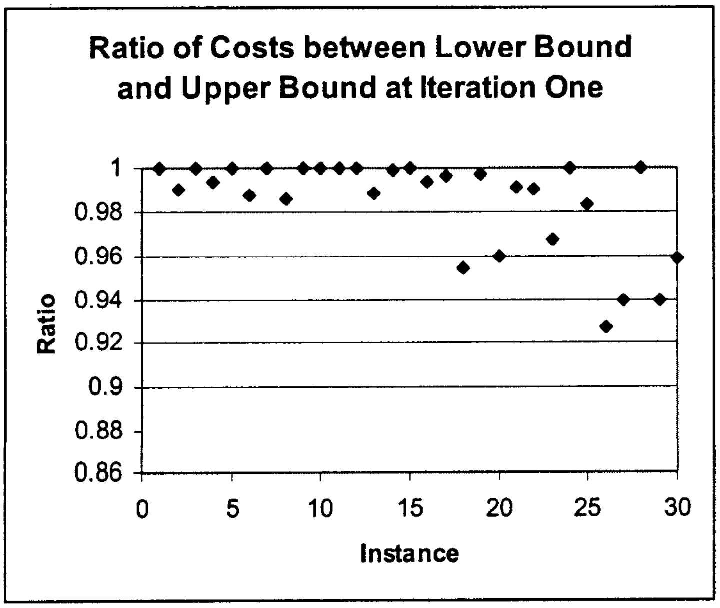


Figure 7) Ratio of Lower to Upper Bound At Iteration One

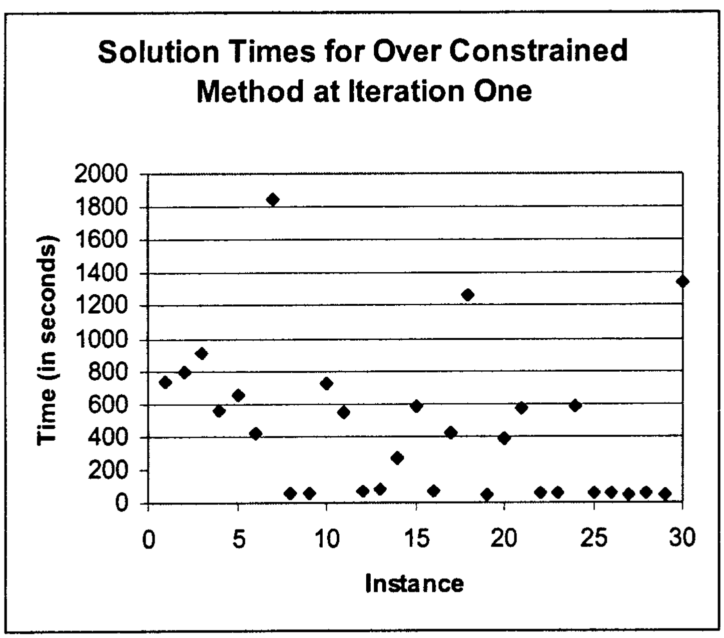


Figure 8) Solution Times for the upper bound at first Iteration

CONCLUSIONS RE: MYOPIC ASSIGNMENT

A method to develop vehicle assignments for local truckload operations is examined and a set of problems based on real data has been examined. Test results presented in this paper focus on relatively small, but operationally realistic problems. Under constrained and over constrained method are presented. The under constrained method is used to evaluate the optimality of the feasible solution obtained using the over constrained method. A cost non-increasing partitioning scheme is developed for implementation in the iterative solution process.

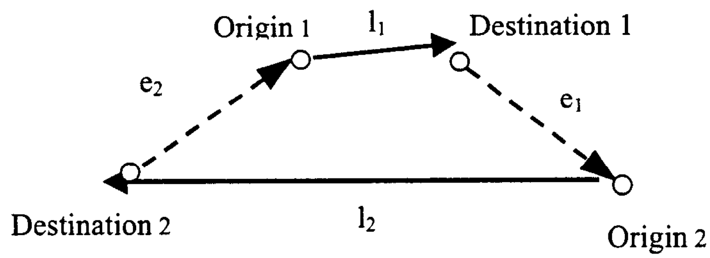
Test results show that smaller partitioning width and larger problems lead to higher level of optimality, but longer machine time. Over constrained and under constrained method provides a mean to compromise between the optimality and machine time. Test result suggest that using what is now a relatively slow commercial optimization package that operational problems of reasonable size can be solved quickly. In addition, the method examined results in sub-problems that are simply network flow problems. Therefore, decomposition techniques could be applied along with time window partitioning method discussed in order to solve larger problems more quickly. Empirical analysis suggests that for these problems, partitioning longer time windows into two-hour windows from which a solution will be selected results in good solutions. For the problems of this size, the loss of optimality introduced by discretization is limited as is indicated by the ratio between upper bound and lower bound. For real-time operations, solving the LP relaxation of the under constrained or original formulation seems the most efficient method for obtaining a lower bound. On the other hand, if machine time is not a big concern, under constrained method instead of its LP relaxation is preferable.

MIXED LOCAL AND GLOBAL ASSIGNMENT TECHNIQUES

Our research is concerned with the following: if dispatchers prefer purely local rather than global changes to assignments, what is the cost in terms of operational efficiency? Can we develop local assignment and reassignment techniques, which satisfy the desire of dispatchers and at the same time, provide good solutions relative to the optimal one? We compare a system in which we re-optimize (by solving the start of day problem) during the day to one in which we rely on local techniques for mid-day reassignments.

Mid-day Assignment and Re-Assignment

For mid-day assignments we rely on solving a set of asymmetric traveling salesman problems with time windows. Whenever a new load arrives in the system or a load currently assigned requires reassignment we solve a traveling salesman problem with time windows problem for each vehicle. Loads become candidates for reassignment if they become time-window infeasible because of unexpected delays or because their assignments have become relatively inefficient due to a series of locally optimal assignments. The traveling salesman problem (TSP) is one of the fundamental routing problems and has been a subject of extensive research. It requires the determination of a minimal cost cycle that passes through each demand node in a network exactly once. The costs considered could be the total distance traveled, the empty distance traveled, or the travel time. In this research context where each demand node is an origin-destination pair, we minimize the empty distances traveled. The costs therefore are asymmetric (Figure 9).



l_1, l_2: loaded distances
e_1, e_2: empty distances

$e_1 \neq e_2$

Figure 9) Asymmetric costs

The TSP with time windows (TSPTW) is a TSP with time window constraints introduced at each demand node. Without loss of generality, as in the start of day VRPTW, we consider time windows at only the pickup locations. Unless dual time windows are inconsistent, they may be easily transformed to one in which only pickup time windows are required. Such inconsistencies may be identified if need be in a preprocessing step.

The TSPTW can be formulated as a mathematical program to obtain an optimal solution. See for example, Desrosiers *et al* (1995). However, in this research we assume that we can solve the TSPTW problems encountered by complete enumeration. The characteristics of the problem limit the number of tasks assigned to any vehicle at a time to around five, though three is a more typical number.

The local assignment rules (dynamic dispatching heuristics) allow for the en-route diversion of vehicles moving empty in the system (figure 10). That is, in addition to being a candidate for assignment to any vehicle in the system, new loads may be assigned to be the current load of any vehicles not moving loaded. The rule used to assign the load to a vehicle simply looks for the vehicle for which the sum of the empty distances is least. That is, we solve a TSPTW for each vehicle that includes the candidate load and select the vehicle for which the cost of the TSPTW is least.

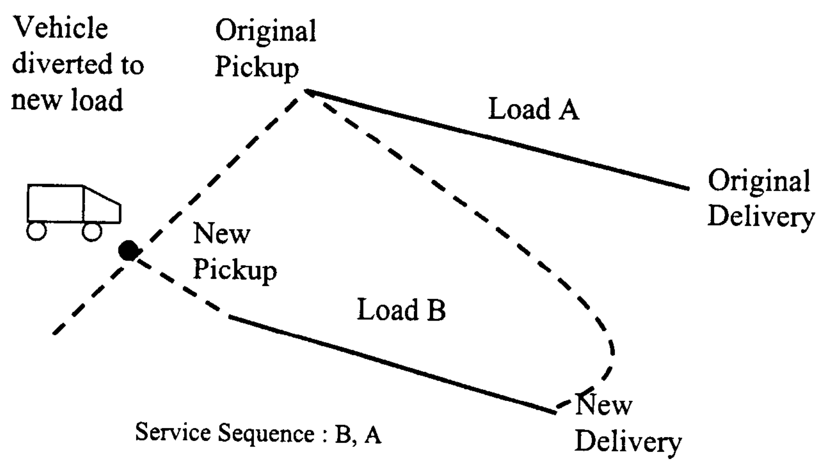


Figure 10) En-Route Diversion

The Simulation Model

A simulation model was developed to analyze the efficiency of various assignment techniques. It works in the following way. Assignments are provided to the simulation model, which then moves loads over a street network. Customer locations are drawn from a set of actual customer locations. As new loads arrive to the system they are immediately assigned to a vehicle. At preset reoptimization points the simulation is paused and a new globally optimal assignment is generated. This assignment is provided to the simulation model, which resumes moving vehicles and serving loads.

The GIS Component

Described in detail in Jagannathan, (1999) the simulation model developed as a custom application using the TransCAD Caliper Script programming language and GISDK. Many of the functions listed in GISDK are used to manipulate the data and maps. For example, to find the shortest path between any two points in the network, the function *ShortestPath( )* can be used. TransCAD database files provide the data (network information) for the analysis. The visualization of the routes is accomplished using the GIS tool. There are many in-built functions that can be used to achieve this. For example: the function *AddRoutes( )* can be used to add and display routes. Another function *AddAnnotation( )* displays routes, but as a freehand item.

An important requirement of the local assignment rules (dynamic dispatching heuristics) is the determination of the state of the system at any instant in time and specifically when new loads arrive to be serviced. (This is to enable diversion of a vehicle from one task to another to serve a load with close time-window constraints). By state of the system, we mean, the location of the new load, each vehicle's position and status and the sequence of loads assigned to each load. All but the vehicle's location are easily determined.

The vehicle's location is the point on its path (on which it is traveling) at the current time. Although the distance traveled from its origin can be calculated using the speed, current time and time it began its move, it is not possible to find the coordinates (latitude, longitude) of the location using TransCAD or the functions defined. To obviate this problem, a simple scheme was devised wherein the location of the vehicles and loads is identified using the node numbers.

Therefore, at any instant in time, a node number will identify the vehicle's location (each network is made of a set of nodes and corresponding links). This is made possible by the fact that the function that is used to calculate the shortest path between any two points (*ShortestPath()*) returns an array of links lying on the path. Using this information and the link lengths, a milepost between the origin and destination is created. This can be done for any given origin and destination. The milepost not only has the distance from origin information but also the node number and the link number. A schematic diagram of the milepost is shown in Figure 11.

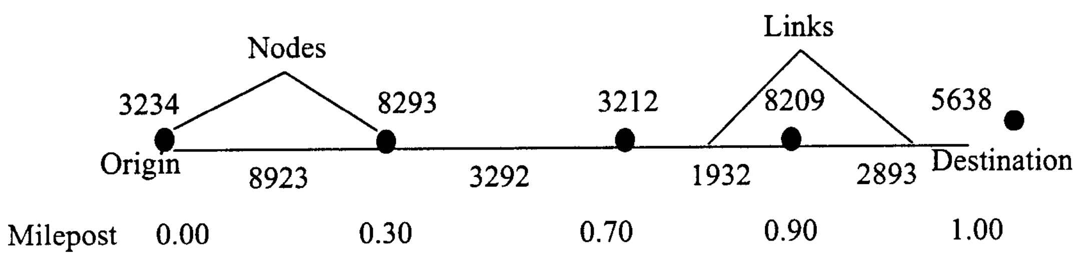


Figure 11) Schematic diagram of the milepost created

The positions of the vehicles are updated at the occurrence of every event. The update is done as follows: the distance traveled by the vehicle in the elapsed time (time between events) from its previous location can be calculated (speed is given). Using the milepost distance and node information, the node on the route closest to the distance traveled is identified as the current location of the vehicle. If the distance traveled is greater (or lesser) than the milepost of the closest node, that distance is stored as the extra distance traveled (or to be traveled) by the vehicle. For example: In Figure 11, if the distance traveled in the elapsed time is 0.55 distance units, and its previous location was the origin, then the current location will be Node 3212, and the extra distance to be traveled will be 0.15 distance units.

Test Problems

We present results based on thirty test problems that are similar to problems solved in the field. Demands are generated by selecting randomly from among known customers in the service area of the Los Angeles Basin region of Southern California (figure 12). All moves either originate or terminate at the rail intermodal facility. Time windows for loads known at the start of the day are randomly assigned based on the distribution shown in table 4, which roughly corresponds to the time windows associated with loads known at the start of day. Time windows for loads arriving during the day are randomly assigned as well and are equally likely to be two, four or eight hours. The objective function simply seeks to minimize empty distance traveled while serving as many loads as possible.

Table 4a. Probabilities associated with time windows for the start of day problem

Time window	7:00-7:30 AM (0.5 hours)	8:00-9:30AM (1.5 hours)	8:00AM-12:00PM (4 hours)	12:00-5:00PM (5 hours)
Probability	0.1	0.15	0.35	0.4

Table 4b. Probabilities associated with time windows for the mid day problem

Time window	2 hours from request	4 hours from request	8 hours from request
Probability	0.333	0.333	0.333

For the problems in the test set, we begin with the vehicles at the depot (which in this problem is very near the rail yard) and make all vehicles available for the duration of the day. Vehicles are not required to return to depot after each service. Travel distances correspond to the shortest network travel distance. Travel time is assumed to be 35 miles per hour, reflecting congestion levels in the test region. The average loaded distance is quite short, less than twenty miles long. A handling time (typically dock time) of forty minutes is assumed for each load though in an actual operation the handling time could be customer specific. We present here results related to solving 30 problems involving 20 vehicles in which 75 loads are known at the beginning of the day and an addition 75 loads are requested according to a Poisson arrival process between the hours of 10:00 AM and 2:00 PM. Twenty was selected because it is the typical maximum size of a local sub-fleet handled by a single dispatcher. Four solution methods are compared. In the base case the start of day solution is augmented only by local assignments. En-route diversion is allowed. In the second case the start of day solution is augmented by local assignments in which a re-assignment rule is applied whenever a new load is added to the system. This re-assignment rule seeks to identify sub-optimal assignments and improve these using exactly the same assignment technique as with new loads. Several simple reassignment rules have been examined (Jagannathan, 1999). The one used in this test is the following.

Reassignment rule

After the newly arriving load is assigned to a vehicle, ratio of the empty distance to loaded distance associated with each load is calculated. These costs are arranged in descending order and the top five percent of the loads are candidates for immediate reassignment. We consider reassignment for only those loads assigned to vehicles with more than two loads currently assigned. The loads to be reassigned are treated as new loads and are considered sequentially (beginning with the "worst") for assignment to any of the vehicles.

The third case is one in which reassignment is not considered, but at noon, after approximately half of the dynamically arriving loads are known, the system is reoptimized. The fourth case is one in which the system is reoptimized at noon and then again at 2:00 after all of the new loads are known.

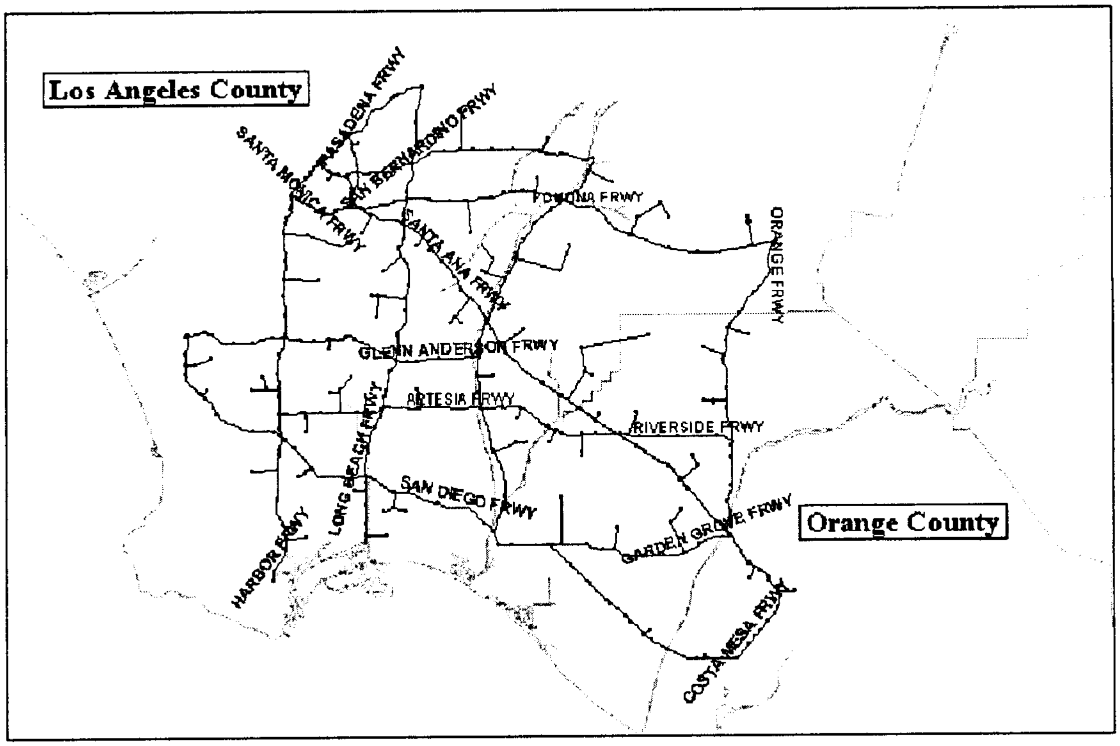


Figure 13) Map of study region

The test results are fairly conclusive. They indicate that significant opportunities for improvement lie in the re-optimization of the system. Test results are quite similar to those found by Yang *et al*(1999), in simulation experiments in an idealized network in which loads are generated randomly in a unit square and vehicles move according to Euclidean distance. In some cases they are more dramatic, this is likely travel in a realistic street network favors the global optimization solutions more than Euclidean travel.

Figures 14-16 present the average total distance traveled, the average system time (total length of day from start to finish) and the average idle time per vehicle under the four scenarios. The average values for the performance measures are presented as well as the upper and lower bounds on the confidence intervals for each performance measure. In all cases the system in which reoptimization is done twice (after which point no changes occur) performs best, followed by the system in which reoptimization is done after about half of the new loads have arrived, followed by the system in which in addition to local assignment heuristics the local reassignment technique is used.

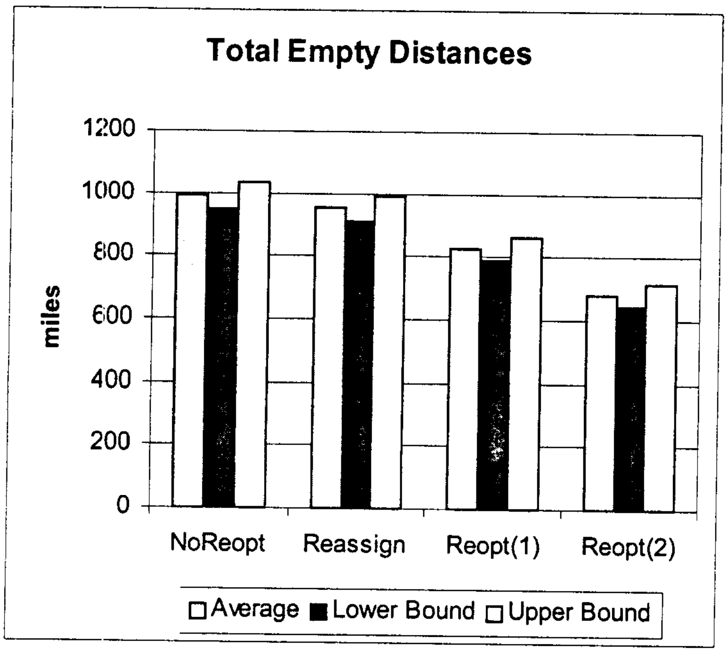


Figure 14) Average total empty distance traveled under four scenarios

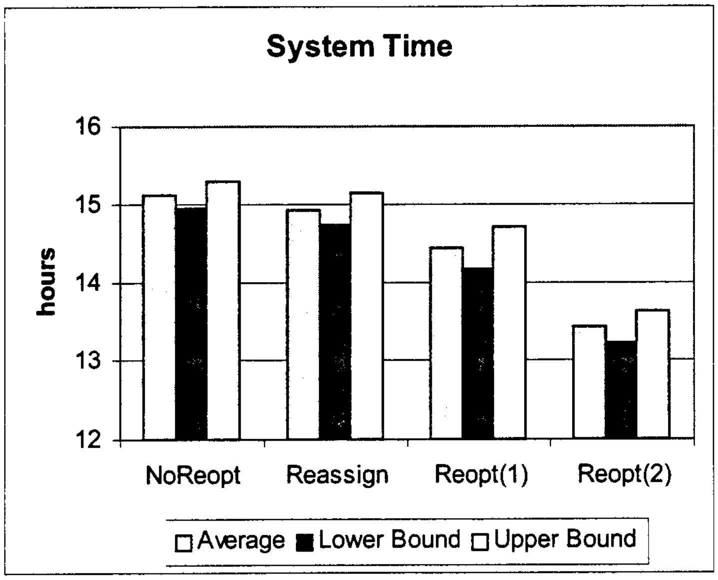


Figure 15) Average system time under four scenarios

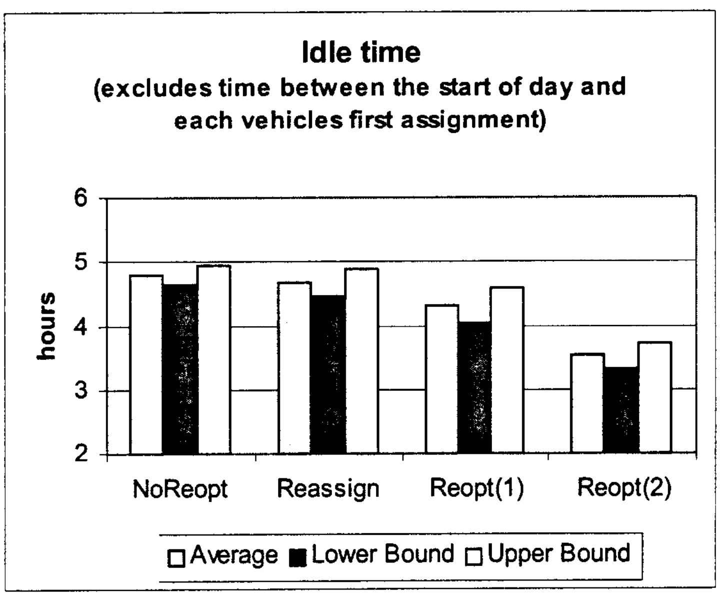


Figure 16) Average idle time per vehicle under four scenarios

Some caution should be taken in the interpretation of these results. Although they make a strong case for the implementation of a system wide (or sub-fleet wide) optimization system in which at any time schedules are subject to change, the effect of stochastic dock and travel times has not been examined in these tests. While these *may* make the system wide optimization system method perform better with respect to the base case involving only local changes, in order to maintain time window feasibility the system would need to be reoptimized more often during the day producing less stable assignments. On-going research is examining this question. In addition it may be possible to improve the base case. Its likely that afternoon loads assigned at the beginning of the day "anchor" schedules in ways that become inefficient as the day progresses. Excluding some of these loads in the beginning of the day may improve the overall efficiency of the assignment. Finally, from the point of view of service requests, the system examined has a higher degree of dynamism than most actual intermodal operations. Since loads that must be picked up at the rail yard are pre scheduled (albeit subject to delay) only around half of the loads can be requested dynamically (in fact, the Los Angeles region favors consumption over production so this value is less than half). In operations that are only twenty or thirty percent dynamic a start of day optimization system followed by local mid day changes may have advantages over the alternative.

CONCLUSIONS RE: MIXED ASSIGNMENT TECHNIQUES

Test results suggest that global optimization methods hold significant benefits over local assignment techniques for the development of cost-effective schedules. The purely local assignment and reassignment techniques under-perform global reoptimization with respect to all measures examined.

The GIS based simulation model developed provides a robust environment for studying the performance of assignment strategies. Travel times, now based on stable congestion levels could be modified to reflect typical traffic network conditions. Handling times could be easily drawn from a distribution rather than assumed static. These distributions can be developed easily from historical data.

Continuing research has the following goals: 1) to develop assignment techniques which combine the benefits of local assignment techniques and global optimization; 2) to examine the operational impacts of implementing the global reoptimization techniques; 3) to examine the level of information technology necessary to successfully implement a computer aided dispatching system for local rail intermodal operations; 4) to examine the trade-offs associated with larger or smaller sub-fleets and sub-regions.

REFERENCES

Appelgren, L.H. (1969), A column generation approach for a ship scheduling problem, Transportation Science, Vol 3, No 1, pp. 53-68.

Ball, M., B. Golden, A. Assad and L.Bodin (1981), Planning for truck fleet size in the presence of a common carrier option, Decision Sciences, Vol 14, No. 1, pp. 103-120.

Desrochers, M., J. Desrosiers and M.M. Solomon (1992), A new optimization algorithm for the vehicle routing problem with time windows, Operations Research 40, pp342-254

Desrosiers, J., F. Soumis and M. Sauve (1983), Lagrangian relaxation methods vehicle routing and scheduling with time windows, Les Cahiers du GERAD, 0709-986X, Paper 83-26.

Desrosiers, J., F. Soumis and M. Sauve (1983), Lagrangian relaxation methods vehicle routing and scheduling with time windows, Les Cahiers du GERAD, 0709-986X, 83-26.

Desrosiers, J., F. Soumis and M. Sauve (1988), Lagrangian relaxation methods for solving minimum fleet size multiple traveling salesman problems with time windows, Management Science, 34, pp 1005-1022.

Desrosiers, J., F. Soumis, M. Desrochers and M. Sauve (1986), Methods for routing with time windows, European Journal of Operations Research, 23, pp. 236-245.

Desrosiers, J., Y. Dumas, M.M. Solomon and F. Soumis (1995), Time constrained Routing and Scheduling, in Ball, M.O., T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), Handbooks in Operations Research and Management Science, Vol. 8, Network Routing, Elsevier, Amsterdam, pp. 35-140.

Dror, M. (1994), Note on the complexity of the shortest path models for column generations in the VRPTW, Operations Research, Vol 42, pp. 977-978.

Golledge, R.G. (1996), Object-oriented dynamic geographic information systems for transportation planning, working paper, University of California Transportation Center, no.337, p198.

Jagannathan, S. (1999), Analysis of dynamic dispatching heuristics for carrier fleet operations in a GIS framework, MS Thesis, University of California, Irvine.

Kolen, A.W.J., A.H.G. Rinnooy Kan and H.W.J.M. Trienekens (1987), Vehicle routing with time windows, Operations Research, 35, pp. 266-273.

Langevin, A., F. Soumis, and J. Desrosiers (1990), Classification of traveling salesman problem formulations, Operations Research Letters, 9, p. 127-132.

Levin, A. (1971), Scheduling and fleet routing models for transportation systems, Transportation Science, Vol 5 No 3, pp. 232-255.

Powell, W.B. (1988), A comparative review of alternative algorithms for the dynamic vehicle allocation problem, in Golden, B.L. and A.A. Assad (eds), Vehicle Routing: Methods and Studies, Elsevier (North-Holland), Amsterdam, pp. 249-291.

Powell, W.B. (1996), A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers, Transportation Science, Vol. 30, No. 3, pp. 195-219.

Powell, W.B., M.T. Towns and A. Marar (200X), On the Value of Globally Optimal Solutions for Dynamic Routing and Scheduling Problems, Transportation Science, to appear.

Powell, W.B., W. Snow and R.K. Cheung (200X), Adaptive Labeling Algorithms for the Dynamic Assignment Problem, Transportation Science, to appear.

Regan, A.C., H.S. Mahmassani and P. Jaillet (1996), Dynamic decision making for commercial fleet operations using real-time information, Transportation Research Record, No. 1537, pp. 91-97.

Regan, A.C., H.S. Mahmassani and P. Jaillet (1998), Evaluation of dynamic fleet management systems: a simulation framework, Transportation Research Record, No. 1645, pp. 176-184.

Regan, A.C., S. Jagannathan, and X. Wang (2000), Mixed global and locally optimal heuristics for local truckload trucking operations with strict time-windows, Transportation Research Record, accepted for publication.

Swersey, A. and W.Ballard (1984), Scheduling school buses, Management Science, Vol 30 pp. 844-853.

Walker, W.T.(1992), Network economies of scale in short haul truckload operations, Journal of Transport Economics and Policy, Vol 26, pp.3-17.

Wang, X and A.C. Regan (1999), Local Truckload Vehicle Routing and Scheduling with Strict Time Window Constraints, to be presented at the 79th annual meeting of the Transportation Research Board, January, 2000.

Yang, J., P. Jaillet and H.S. Mahmassani (1999), On-Line Algorithms for truck fleet assignment and scheduling under real-time information, Transportation Research Record, in press.