

Real-Time Multi-Vehicle Truckload Pick-Up and Delivery Problems*

Jian Yang Patrick Jaillet Hani Mahmassani

*Department of Industrial and Manufacturing Engineering
New Jersey Institute of Technology, Newark, NJ 07102
yang@adm.njit.edu*

*Department of Civil and Environmental Engineering
Massachusetts Institute of Technology, Cambridge MA 02139
jaillet@mit.edu*

*Department of Civil and Environmental Engineering
The University of Maryland, College Park, MD 20742
masmah@wam.umd.edu*

Abstract

In this paper we formally introduce a generic real-time multi-vehicle truckload pick-up and delivery problem. The problem includes the consideration of various costs associated with trucks' empty travel distances, jobs' delayed completion times, and job rejections. Although very simple, the problem captures most features of the operational problem of a real-world trucking fleet that dynamically moves truckloads between different sites according to customer requests that arrive continuously over time.

We propose a mixed integer programming formulation for the off-line version of the problem. We then consider and compare five rolling horizon strategies for the real-time version. Two of the policies are based on a repeated re-optimization of various instances of the off-line problem, while the others use simpler local (heuristic) rules. One of the re-optimization strategies is new while the other strategies have recently been tested for similar real-time fleet management problems.

The comparison of the policies is done under a general simulation framework. The analysis is systematic and consider varying traffic intensities, varying degrees of advance information, and varying degrees of flexibility for job rejection decisions. The new re-optimization policy is shown to systematically outperform the others under all these conditions.

*Accepted for publication in *Transportation Science*, December 2002

Introduction

Continuing developments in telecommunication and information technologies provide unprecedented opportunities for using real-time information to enhance the productivity, optimize the performance, and improve the energy efficiency of the logistics and transportation sectors. Interest in the development of dynamic models of fleet operations and of fleet management systems which are responsive to changes in demand, traffic network and other conditions is emerging in many industries and for a wide variety of applications. Managing and making use of the vast quantities of real-time information made available by navigation technologies, satellite positioning systems, automatic vehicle identification systems and spatial GIS databases require the development of new models and algorithms.

The area of vehicle routing and scheduling, including dynamic vehicle allocation and load assignment models, has evolved rapidly in the past few years, both in terms of underlying mathematical models and actual commercial software tools. While some of the approaches may well be adaptable to operations under real-time information availability, underlying existing formulations do not recognize the possible additional decisions that become available under real-time information.

In this paper we formally introduce a generic real-time multi-vehicle truckload pick-up and delivery problem called hereafter TPDP. The problem includes the consideration of various costs associated with trucks' empty travel distances, jobs' delayed completion times, and job rejections. The TPDP captures most features of the operational problem of a real-world trucking fleet that moves truckloads between different sites according to customer requests that arrive continuously in time. On the other hand, the problem is still a simplification of real-world problems in that the latter also needs to address issues such as working hour regulations, getting drivers home, and suitability of the driver and his equipment for a load. Nevertheless, good solutions for this artificial TPDP should provide good insights and building blocks for more realistic real-time pick-up and delivery problems.

We propose a mixed integer programming formulation for the off-line version of the problem. We then consider and compare five rolling horizon strategies for the real-time version. Two of the policies are based on a repeated re-optimization of various

instances of the off-line problem, while the others use simpler local (heuristic) rules. One of the re-optimization strategies is new while the other strategies have recently been tested for similar real-time fleet management problems. The comparison of the policies is done under a general simulation framework. The analysis is systematic and consider varying traffic intensities, varying degrees of advance information, and varying degrees of flexibility for job rejection decisions.

Before going into more details about the organization and content of this paper let us first provide an overview of the related existing literature.

Vehicle routing problems (VRPs) are usually concerned with efficiently assigning vehicles to jobs (such as picking up and/or delivering given loads) in an appropriate order so that these jobs are completed in time and vehicles' capacities are not exceeded. Deterministic and static versions, with all the characteristics of the jobs being known in advance and every parameter assumed certain, have been widely studied in the literature. Bodin et al. 1983, Christofides 1985, Fisher 1995, Golden and Assad 1988, and Solomon 1987 provide extensive surveys of the various VRPs and solution techniques. Bienstock et al. 1993, Bramel and Simchi-Levi 1996, 1997, Bramel et al. 1994, and Bramel et al. 1992 present probabilistic analyses of many heuristics for deterministic and static VRPs.

Stochastic and static versions of the vehicle routing problem (SVRP) have also been widely studied. Several authors have addressed the case in which loads are random. Golden and Stewart 1978 tackle problems with Poisson-distributed loads. Golden and Yee 1979 consider other load distributions and give theoretical explanations for the relations found empirically by Golden and Stewart. Stewart 1981 and Stewart and Golden 1983 formulate SVRP as a stochastic programming problem with recourse. Bastian and Rinnooy Kan 1992 show that with one vehicle and independent identically distributed loads, SVRP could be reduced to the time-dependent traveling salesman problem (TDTSP) (Garfinkel 1985). Work in this direction was also done in papers such as Tillman 1969, Dror and Trudeau 1986, Yee and Golden 1980, Bertsimas 1992, and Dror et al. 1989. Researchers have further considered the case in which travel times between jobs are random. Cook and Russell 1978 examine a large SVRP with random travel times and random loads. Berman and Simchi-Levi 1989 examine the problem of finding the optimal depot in a network with random travel

times.

Some authors have considered cases in which the number and possibly the locations of the jobs are not known in advance but are described instead by probability distributions. The goal is to find optimal a priori routes through all jobs and update these routes at the time the specific subset of jobs to be served is known. Such a problem, the probabilistic traveling salesman problem, was first introduced in Jaillet 1985, 1988, who develops an extensive analysis of the case where all potential jobs have the same probability to materialize. Jezequel 1985, Rossi and Gavioli 1987, Bertsimas 1988 and Bertsimas and Howell 1993 investigate additional theoretical properties and heuristics for the problem. Berman and Simchi-Levi 1988 discuss the problem of finding an optimal depot under a general job-appearing distribution. Laporte et al. 1994 formulate the problem as an integer program and solve it using a branch-and-cut approach.

When information on jobs is gradually known in the course of the system's operation, real-time techniques become increasingly important. In his review of dynamic vehicle routing problems, Psaraftis 1988 points out that very little had been published on real-time VRPs as opposed to classical VRPs. Powell et al. 1995 present a survey of dynamic network and routing models and identify general issues associated with modeling dynamic problems. For more recent surveys on dynamic vehicle routing problems and related routing problems, see Psaraftis 1995, Bertsimas and Simchi-Levi 1996, and Gendreau and Potvin 1998.

Bertsimas and Van Ryzin 1991, 1993a, 1993b analyze a dynamic routing problem in the Euclidean plane with random on-site service times. They use queueing models to compare the impact of various dispatching rules on the average time spent by the customer in the system. They derive the asymptotic behavior of the optimal system time under heavy traffic, and find several policies that result in asymptotic system times that are within constant factors of that of the optimal one in heavy traffic.

For the more general dynamic VRP with time windows, Gendreau et al. 1999 have proposed a general heuristic strategy (a continuously running tabu search attempting to improve on the current best solution, interrupted by a local search heuristic for inserting newly arrived demand). Their objective takes into account job rejections, operational cost due to vehicle travel distances, and cost due to customer waiting.

Ichoua et al. 2000 further consider methods that allow vehicle diversions for these vehicle routing problems with time windows. Empirical tests show a reduction in the number of unserved customers if diversion is allowed. Due to computational limitations and the notorious difficulty of the off-line VRP with time windows, it is unclear that a re-optimization based strategy similar to the one proposed in our paper could also be effective for this problem and improve on this tabu search procedure.

Closer to the class of problems considered here, Bookbinder and Sethi 1980, Powell et al. 1984, Powell 1986, 1987, Dejax and Crainic 1987, Powell 1988, Frantzekakis and Powell 1990, and Powell 1996 all address the dynamic vehicle allocation problem (DVA) for which a fleet of vehicles is assigned to a set of locations with dynamically occurring demands. In all these models, both locations and decision epochs are discrete. Due to the curse of dimensionality, the models have limited time horizons and cannot effectively address the issue of job delays. Most effective DVA models are of a multi-stage stochastic programming type. Frantzekakis and Powell 1990 use linear functions to approximate separable convex recourse objective functions and solve the problem at each decision epoch using backward recursion. Powell 1996 shows that it is advantageous to take forecasted demands into consideration when deciding on the vehicle-location assignment, compared to a model which reacts after new demands have arrived. This however presumed that one can accurately predict future demands.

More recently, Powell et al. 2000a consider a dynamic assignment of drivers to known tasks. Their formulation includes many practical issues and driver-related constraints and generalizes the off-line version of the problem we consider in this paper. Two primal-dual iterative methods are developed to solve the off-line problem. Powell et al. 2000b implement the previous primal-dual approaches into a dynamic driver assignment problem where there are three sources of uncertainty: customer demands, travel times, and user noncompliance, and compare these with simpler non-optimal local rules. They find that the increase in future uncertainty may reduce the benefit of fully re-optimizing the off-line problem each time a new request comes. This contrasts with some of our findings which indicate that fully re-optimizing each time leads to an overall better performance, under our various testing situation. We should however be very cautious in comparing these results because the problems and the comparison settings are quite different.

Regan et al. 1995, 1996a, 1996b, 1998 evaluate vehicle diversion as a real-time operational strategy for similar truckload pick up and delivery problems, and investigate various local rules for the dynamic assignment of vehicles to loads under real-time information. That approach features relatively simple, easy-to-implement and fast-to-execute local rules that might not always take full advantage of the existing past and present information. The empirical analysis of these local rules was conducted using a limited exploratory simulation framework, typically with small fleet sizes and under the objective of minimizing total empty distance. Re-optimization real-time policies for truckload pick-up and delivery problems are further introduced and tested under a more general objective function in Yang et al. 1998.

In this paper, we build upon this previous work and use computer simulation to experimentally identify and test good strategies under varying situations. The main contributions of the paper are the introduction of a new optimization-based policy (OPTUN) for the TPDP, and its comparison with the simple local rules of Regan et al. 1998 and other strategies introduced in Yang et al. 1998). The comparison is done under a general framework in which the objective function relaxes the hard constraints associated with the delivery of a job and introduces a penalty function for delay beyond the due time. The analysis is systematic and considers the performance of the policies under varying traffic intensities, varying degrees of advance information, and varying degrees of flexibility for job rejection decisions. The OPTUN policy turns out to be the best-performing policy under all these different conditions, and clearly outperforms the simple local rules, and other myopic strategies.

The paper is organized as follows. In Section 1, we present the detailed definition of the problem. In Section 2, we discuss formulations for the off-line problem corresponding to our real-time problem. In Section 3, we discuss the various policies and present the detailed mechanism of how a trucking company would operate under these policies. In Section 4, we present the details of our simulation studies. In Section 5, we present results and conclusions from the simulation studies. Section 6 concludes the paper.

1 Problem Statement

Overview

In this stylized problem, we consider a trucking company with a fleet of K trucks. The company faces a sequence of future and unknown requests for truckload moves, hereafter called jobs, within a predefined region. Each truck can carry only one job at a time, and cannot serve another job until the current job is delivered to its final destination. At the arrival time of a request, the company is given the pick-up location, the delivery location, the earliest pick-up time, and the latest delivery time of the job. The company can either accept or reject a job request within a small prescribed amount of time. The revenue generated from a given accepted job is proportional to the length of the job, defined as the distance between its pick-up and delivery locations. Completion beyond the latest delivery time is allowed but penalized, and the penalty is proportional to both the job's length and the amount of delay occurred. In case a job request is not accepted, the cost of rejection is the gross revenue the company would have otherwise obtained had it accepted the job. Over the course of serving the sequence of requests, the company incurs additional operating costs proportional to the empty distance traveled by trucks in order to serve the accepted jobs. Finally we assume that the trucks all move at the same constant unit speed.

The objective is to find a good strategy for handling this sequence of future unknown requests in order to maximize the overall net revenue. The strategy needs to address job acceptance/rejection decisions in “real-time” as well as job-truck assignment decisions for currently accepted jobs, not knowing the timing and characteristics of possible future requests.

Formal Notation and Model Statement

The time evolution of the system is indexed by a continuous variable $t \in [0, \infty)$. Initially, at time $t = 0$, all K trucks are empty and idle at a common depot. Job pick-up and delivery locations, as well as truck positions at any given time $t \geq 0$, are assumed to be points in a bounded region \mathcal{B} of a metric space. For simplicity, we assume that this space is the Euclidean plane and that the distance between any two points in that plane is the Euclidean distance hereafter denoted $D(., .)$.

The exogenous stimulus of the system is provided by a sequence of job requests. Formally it is represented by a sequence of increasing real numbers $(\tau_i^{ARV})_{i \geq 1}$, where τ_i^{ARV} denotes the arrival time of job i (we assume that jobs are labeled according to their order of arrival). At the arrival time of job i , its characteristics are then revealed through a 5-tuple $I_i \equiv (\mathbf{o}_i, \mathbf{d}_i, T_i^{ADV}, T_i^{SLK}, T_i^{RES})$ with the following definitions, (some being also illustrated in Figure 1):

- \mathbf{o}_i and \mathbf{d}_i are the pick-up and delivery locations, respectively. The corresponding distance between these two locations, i.e., the length of job i , is denoted W_i .
- T_i^{ADV} measures the time between the arrival epoch of job i and its earliest pick-up time. In other words, if τ_i^{AVL} is the earliest pick-up time, then $\tau_i^{AVL} = \tau_i^{ARV} + T_i^{ADV}$.
- T_i^{SLK} is the slack time available between earliest possible and latest allowed delivery, and captures the tightness of job i 's completion deadline. In other words, if τ_i^{DLN} is the time of latest delivery, then $\tau_i^{DLN} = \tau_i^{AVL} + W_i + T_i^{SLK}$.
- Finally, T_i^{RES} is the time within which the company needs to respond to a job request with a final acceptance or rejection decision. In other words, the latest time for the trucking company to decide whether to accept or reject job i is $\tau_i^{ARV} + T_i^{RES}$.

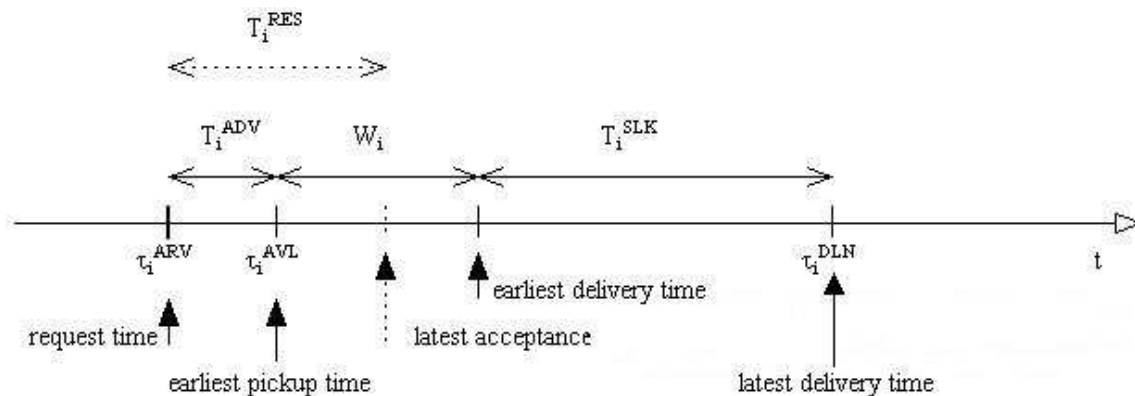


Figure 1: Illustration of the time elements in I_i

The joint sequence $(\tau_i^{ARV}, I_i)_{i \geq 1}$ completely characterizes the job requests. Let $(\mathcal{F}_t)_{t \geq 0}$ be the (information) filtration generated by the sequence $(\tau_i^{ARV}, I_i)_{i \geq 1}$. Informally, \mathcal{F}_t represents the known information up to time t , as contained in all the past requests j such that $\tau_j^{ARV} \leq t$.

Facing this sequence of job requests the company responds with a series of decisions including job acceptance/rejection decisions and job-truck assignment decisions on accepted jobs. We call such a series of decisions a policy or a strategy. We restrict the policies to be \mathcal{F}_t -adapted, i.e., any decisions at time t must only depend on the information up to time t (decisions are \mathcal{F}_t -measurable). Because exogeneous information is updated only at job arrival epochs, a policy can be described in a rolling-horizon fashion: At any time that is not a job arrival epoch, there is a previously agreed assignment plan being carried out. At every job arrival epoch, the previous plan is interrupted and a new plan is decided for the time to come. For this reason, we also call a job arrival epoch a decision epoch. At every decision epoch, a myopic policy is to optimize a new assignment plan without recognizing that it may not be fully carried out because of future unknown requests. We also require that, at any decision epoch τ , the new plan decided by a policy does not change the previous acceptance/rejection decisions associated with any job i such that $\tau_i^{ARV} + T_i^{RES} \leq \tau$. Indeed, the acceptance/rejection decision status of job i becomes permanent by time $\tau_i^{ARV} + T_i^{RES}$ and cannot then be changed.

Under such a policy π , each truck k , $1 \leq k \leq K$, is at any time t , either idle, moving empty, or moving loaded. We formally represent this status with an integer variable $s_k^\pi(t)$ with three possible values: $s_k^\pi(t) = 0$ if idle; -1 if moving empty; and $+1$ if moving loaded. Let $\mathbf{I}_k^\pi(t)$ be truck k 's current location at time t (a two-dimensional real-valued vector in case of cartesian coordinates). Let $Q_k^\pi(t)$ be the current (time t) ordered list of non-completed jobs assigned to truck k under the last updated assignment plan associated with the policy π . $Q_k^\pi(t) = \emptyset$ if and only if $s_k^\pi(t) = 0$. For $Q_k^\pi(t) \neq \emptyset$, let $q_k^\pi(t)$ be the first element of the ordered list and $LQ_k^\pi(t)$ the remaining other elements of the list (i.e., when non-empty, $Q_k^\pi(t) = \{q_k^\pi(t)\} \cup LQ_k^\pi(t)$). Finally let $L_{TEMP}^\pi(t)$ be the current (time t) set of jobs temporarily rejected under the last updated assignment plan associated with the policy π .

Together with the fact that vehicles move at a constant unit speed, it should be

clear that $(s_k^\pi(t), \mathbf{I}_k^\pi(t), Q_k^\pi(t))$, $1 \leq k \leq K$ and $L_{TEMP}^\pi(t)$ allow a full description of the dynamics of the system under policy π . The details are as follow. Assume a sequence of requests $(\tau_i^{ARV}, I_i)_{i \geq 1}$ and a given policy π to serve these requests. Consider that we are at time t^- in a state described by $(s_k^\pi(t^-), \mathbf{I}_k^\pi(t^-), Q_k^\pi(t^-))$, $1 \leq k \leq K$ and $L_{TEMP}^\pi(t^-)$.

1. Assume first that t is a decision epoch, i.e. $t = \tau_j^{ARV}$ for a given job j . A new assignment plan is then made according to the policy π . The parameters $(s_k^\pi(t), \mathbf{I}_k^\pi(t), Q_k^\pi(t))$, $1 \leq k \leq K$ and $L_{TEMP}^\pi(t)$ are then fully updated according to the specifics of the policy π , which only depends on the past up to time t .

2. Assume now that t is not a decision epoch. Let $\tau^{ARV} > t$ be the next job arrival. The dynamics are then as follows:

- Update on the set of temporarily rejected jobs: For all $j \in L_{TEMP}^\pi(t^-)$ such that $t \leq \tau_j^{ARV} + T_j^{RES} = t' \leq \tau^{ARV}$, $L_{TEMP}^\pi(t') = L_{TEMP}^\pi(t^-) \setminus \{j\}$.
- Update on the idle vehicles: For any k such that $s_k^\pi(t^-) = 0$, we have, for $t \leq t' \leq \tau^{ARV}$, $s_k^\pi(t') = 0$, $\mathbf{I}_k^\pi(t') = \mathbf{I}_k^\pi(t)$, and $Q_k^\pi(t') = \emptyset$.
- Update on the vehicles moving empty: For any k such that $s_k^\pi(t^-) = -1$, let i be $q_k^\pi(t^-)$. Moving at unit constant speed, vehicle k would reach the origin of job i at time $t_i = t + D(\mathbf{I}_k^\pi(t), \mathbf{o}_i)$. So for $t \leq t' \leq \min\{\tau^{ARV}, t_i\}$ we have $s_k^\pi(t') = -1$, $\mathbf{I}_k^\pi(t') = \mathbf{I}_k^\pi(t) + (\mathbf{o}_i - \mathbf{I}_k^\pi(t))(t' - t)/(t_i - t)$, and $Q_k^\pi(t') = Q_k^\pi(t^-)$. Then if $t_i \leq \tau^{ARV}$, $s_k^\pi(t_i) = +1$, $\mathbf{I}_k^\pi(t_i) = \mathbf{o}_i$, and $Q_k^\pi(t_i) = Q_k^\pi(t^-)$. Otherwise at $t' = \tau^{ARV}$ we are back in Case 1 above.
- Update on the vehicles moving loaded: For any k such that $s_k^\pi(t^-) = +1$, let i be $q_k^\pi(t^-)$. Moving at unit constant speed, vehicle k would reach the destination of job i at time $t_i = t + D(\mathbf{I}_k^\pi(t), \mathbf{d}_i)$. So for $t \leq t' \leq \min\{\tau^{ARV}, t_i\}$ we have $s_k^\pi(t') = +1$, $\mathbf{I}_k^\pi(t') = \mathbf{I}_k^\pi(t) + (\mathbf{d}_i - \mathbf{I}_k^\pi(t))(t' - t)/(t_i - t)$, and $Q_k^\pi(t') = Q_k^\pi(t^-)$. Then if $t_i \leq \tau^{ARV}$, $s_k^\pi(t_i) = 0$ if $LQ_k^\pi(t^-) = \emptyset$, $s_k^\pi(t_i) = -1$ otherwise, $\mathbf{I}_k^\pi(t_i) = \mathbf{d}_i$, and $Q_k^\pi(t_i) = Q_k^\pi(t^-) \setminus \{i\}$. Otherwise at $t' = \tau^{ARV}$ we are back in Case 1 above.

We have made no assumption on how to model the uncertainty associated with the sequence of requests, because we want to devise real-time strategies assuming little,

if any, knowledge (deterministic or probabilistic) of the future requests. Of course, present actions influence the company's performance in the future. The difficulty is to make decisions based only on the past and current requests. The approach usually taken is to assume some probabilistic model of the future and act on this basis. This is the starting point of the theory of Markov Decision Processes (see for example Heyman and Sobel 1984). Another approach is to devise and evaluate strategies under the worst possible scenario using the concept of "competitive analysis", now well known in the analysis of online problems and algorithms (see for example Borodin and El-Yaniv 1998). In this paper we are taking a middle ground. We assume that the strategies have to be developed with no knowledge of the future (with the exception of one of the proposed strategies, OPTUN, which uses some minimal probabilistic information on the location of job requests, as explained in Section 3). The analysis and comparison of the proposed strategies however are performed under some very specific probabilistic assumptions. Specifically, we consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ under which is defined a Poisson process $(N_t)_{t \geq 0}$ with intensity λ . The sequence of job arrival epochs $(\tau_i^{ARV})_{i \geq 1}$ corresponds to the Poisson process arrival times. We also define in this probability space a stochastic process $(I_i)_{i \geq 1}$ with values in \mathbb{R}^5 describing the sequence of job characteristics.

Under such probabilistic assumptions one can go further in properly defining an objective function for the evaluation of strategies. We first define time-dependent set of random variables that record the system's performance when a certain stationary policy π is adopted. For each vehicle k we let $\Upsilon_k^\pi(t) = \mathbf{1}[s_k^\pi(t) < 0]$ be a 0-1 random variable indicating whether or not truck k is moving empty at time t . Now let $N(t) = \{i : \tau_i^{ARV} \leq t\}$ be the set of jobs which have been requested by time t . Let $M(t) = \{i : \tau_i^{ARV} + T_i^{RES} \leq t\}$ be the subset of jobs in $N(t)$ for which a final acceptance or rejection decision is mandatory by time t . Finally let $R(t)$ be the subset of jobs in $N(t)$ which have been fully served by time t . Note that for all $s \leq s'$, $N(s) \subset N(s')$, $M(s) \subset M(s')$, and $R(s) \subset R(s')$. For each job $i \in M(t)$, we let $y_i^\pi(t)$ be a 0-1 random variable indicating whether job i has been permanently rejected. Note that the policy π imposes consistencies in the sense that for each job $i \in M(t)$ we have $y_i^\pi(t') = y_i^\pi(t)$ for all $t' \geq t$. For each job $i \in R(t)$, we let $\tau_i^{\pi, COM}(t) \leq t$ be the time of completion of job i .

We will assume that as a function of t , $(\Upsilon_k^\pi(t))_t$, $(y_i^\pi(t))_t$, and $(\tau_i^{\pi, COM}(t))_t$ are well-defined stochastic processes with right-continuous left-limit sample paths.

Let us now specify applicable cost parameters. Let α be the operational cost per unit distance of truck-empty-movement, and let β be the penalty cost per unit of time delay and per unit of job length (5 units of time delay for a job i of length $W_i = 10$ costs 50β , i.e., the longer the job the more costly proportionally it is to delay its final delivery).

Because of unit speed, the total empty distance covered by truck k up to time t is the random variable $\int_0^t \Upsilon_k^\pi(s) ds$. Under policy π and up to time t , the cumulative cost $C^\pi(t)$ is a random variable defined as

$$C^\pi(t) \equiv \alpha \sum_{k=1}^K \int_0^t \Upsilon_k^\pi(s) ds + \beta \sum_{i \in R(t)} W_i \left(\tau_i^{\pi, COM}(t) - \tau_i^{DLN} \right)^+ + \sum_{i \in M(t)} W_i y_i^\pi(t). \quad (1)$$

$C^\pi(t)$ captures the fleet's operational cost due to empty movement, the loss of customer good will due to delay, and the loss of revenue from job rejections. For the second cost term, accounting at time t is done for completed jobs; while for the third term, accounting at time t is done for those jobs whose acceptance/rejection decisions have been finalized.

We assume that all the policies π under consideration in this paper are stable ergodic policies, by which we mean that there exists a constant c^π such that

$$\lim_{t \rightarrow \infty} \frac{C^\pi(t)}{t} = c^\pi \text{ (a.s.)} \quad \text{and} \quad \lim_{t \rightarrow \infty} E \left[\frac{C^\pi(t)}{t} \right] = c^\pi. \quad (2)$$

Mathematically the overall optimization problem is to find a π^{opt} among the set of all stable ergodic policies Π such that

$$c^{\pi^{opt}} = \inf_{\pi \in \Pi} c^\pi.$$

This is in that precise sense that an *optimal* policy is defined in this paper.

One can define an equivalent optimization problem. For any integer n , let $\tau_n^{\pi, *} = \inf\{t : \text{for all } 1 \leq i \leq n, y_i^\pi(t) = 1 \text{ or } i \in R(t)\}$. ($\tau_n^{\pi, *}$ is the smallest time t by which all n first jobs have either been served or rejected.) For stable ergodic policies as defined above, we then have

$$\lim_{n \rightarrow \infty} \frac{C^\pi(\tau_n^{\pi, *})}{n} = c_a^\pi \text{ (a.s.)} \quad \text{and} \quad \lim_{n \rightarrow \infty} E \left[\frac{C^\pi(\tau_n^{\pi, *})}{n} \right] = c_a^\pi. \quad (3)$$

Note that for any $\pi \in \Pi$ the two constants c^π and c_a^π defined in (2) and (3) respectively are such that $c_a^\pi = c^\pi / \lambda$. The two problems $\inf_{\pi \in \Pi} c^\pi$ and $\inf_{\pi \in \Pi} c_a^\pi$ are thus equivalent. The constant c_a^π can be interpreted as the *long-run average cost per requested job*.

For each stable ergodic policy π introduced in this paper we are going to numerically estimate the constant c_a^π by considering a finite approximation. More precisely for a large enough n , we will assume that the following measure

$$E \left[\frac{C^\pi(\tau_n^{\pi,*})}{n} \right] \tag{4}$$

is a good approximation of the constant c_a^π to minimize. It is this approximate objective function (4) which we numerically estimate via our simulation experiments in Section 5, and which we call *AvgCost*. Section 4 precisely describes how *AvgCost* relates to (4).

Before we can describe the five proposed online policies for TDPP, it is important to first understand the following corresponding off-line problem: given a set of trucks and known jobs, find an optimal plan to serve these jobs assuming no future requests. Even though we introduce the off-line problem as a problem being repeatedly solved at decision epochs by a myopic policy for the real-time problem, it models a specific and interesting problem in its own right. This is the subject of the next section.

2 The Off-line Problem

In this off-line problem we consider a problem with K trucks. We assume that truck k is first available at time τ_k^0 and at location \mathbf{l}_k . We assume that there are N known jobs, each being characterized by an arrival epoch and a 5-tuple I as described above. For notational simplicity, we let D_{0i}^k be the distance from truck k 's location \mathbf{l}_k to job i 's pick-up location and D_{ij} be the distance from job i 's delivery location to job j 's pick-up location. Out of the N jobs, we assume that a subset A of these have to be served. The other jobs could be rejected, if it is economically optimal to do so. For an arbitrary choice of A the given off-line problem could be infeasible.

Note that when the off-line problem is the problem solved at a decision epoch in an on-line strategy as described in the previous section, τ_k^0 and \mathbf{l}_k are either the current time and location of truck k if it is idle or moving empty, or the time and

location at which truck k will finish its current job if it is moving loaded. Also, the N jobs would be those in the real-time problem which are already known at t and neither have been picked up nor have been permanently rejected yet. In this setting some jobs may have been permanently accepted, and form the elements of the set A . Since the off-line problem is always called at the arrival epoch of a new job and we can always reject the new job and keep the previous plan, introducing this set A does not make the off-line problem infeasible.

We have looked at two equivalent formulations for the problem. The first formulation is of a multi-commodity network-flow type and has been inspired by the work of Desrochers et al. 1988. All the nodes except for one dummy node, node 0, represent jobs. All the arcs except for those linking job nodes to the dummy node represent possible connections in real services. A truck's route is represented by a flow unit from the dummy node, through some job nodes, and then back to the dummy node. Empirically, this first formulation is not as competitive as the second one, so we omit going into its details here.

In the chosen formulation, we model the problem as an assignment problem with timing constraints. The assignment problem, in turn, consists of finding a least-cost set of cycles going through all the nodes of $(1, \dots, K, K + 1, \dots, K + N)$, where node k for $k = 1, \dots, K$ corresponds to truck k and node $K + i$ for $i = 1, \dots, N$ corresponds to job i . In the formulation, we use binary variable x_{uv} for $u, v = 1, \dots, K + N$ to indicate whether arc (u, v) is selected in one of the cycles. In the truck-job terminology, $x_{k, K+i}$ indicates whether truck k first serves job i , $x_{K+i, K+j}$ indicates whether there is a truck that serves jobs i and j consecutively, $x_{kk} = 1$ means that truck k serves no job, and $x_{K+i, K+i} = 1$ means that job i is rejected. We also use continuous variables τ_i^{PICK} and δ_i to represent the pick-up time and amount of delay of job i , respectively.

The timing constraints presented below prevent cycles to be formed with job nodes only. As a result there is a clear interpretation of a feasible cycle using our truck-job terminology. For instance, if a cycle goes as $1, K + 1, K + 2, 2, K + 3, K + 4, K + 5, 1$, the interpretation is that truck 1 serves jobs 1 and 2, truck 2 serves jobs 3, 4, and 5. The mixed integer programming formulation is presented below:

$$\min \alpha \left(\sum_{k=1}^K \sum_{i=1}^N D_{0i}^k x_{k, K+i} + \sum_{i=1}^N \sum_{j=1, j \neq i}^N D_{ij} x_{K+i, K+j} \right) + \beta \sum_{i=1}^N W_i \delta_i + \sum_{i=1}^N W_i x_{K+i, K+i}$$

subject to

$$\sum_{v=1}^{K+N} x_{uv} = 1 \quad \forall u = 1, \dots, K + N \quad (1)$$

$$\sum_{v=1}^{K+N} x_{vu} = 1 \quad \forall u = 1, \dots, K + N \quad (2)$$

$$x_{uv} = 0, 1 \quad \forall u, v = 1, \dots, K + N \quad (3)$$

$$-\sum_{k=1}^K (D_{0i}^k + \tau_k^0) x_{k,K+i} + \tau_i^{PICK} \geq 0 \quad \forall i = 1, \dots, N \quad (4)$$

$$-T x_{K+i,K+j} - \tau_i^{PICK} + \tau_j^{PICK} \geq -T + W_i + D_{ij} \quad \forall i, j = 1, \dots, N, i \neq j \quad (5)$$

$$\tau_i^{PICK} \geq \tau_i^{AVL} \quad \forall i = 1, \dots, N \quad (6)$$

$$\delta_i - \tau_i^{PICK} \geq W_i - \tau_i^{DLN} \quad \forall i = 1, \dots, N \quad (7)$$

$$\delta_i \geq 0 \quad \forall i = 1, \dots, N \quad (8)$$

$$x_{K+i,K+i} = 0 \quad \forall i \in A \quad (9)$$

Constraints (1), (2), and (3) are classical assignment constraints. Constraints (4), (5), and (6) are the timing constraints, with T a large number. Constraints (4) ensure that truck k arrives at the pick-up location of job i after $D_{0i}^k + \tau_k^0$ if i is the first job being served by k . Constraints (5) insures that the truck arrives at the pick-up location of job j at least $W_i + D_{ij}$ amount of time after reaching job i 's pick-up location if j is to be served after i . Because T is large enough, when $x_{K+i,K+j} = 0$, the constraints are nonrestrictive. We note that constraints (4) and (5) are those that prevent cycles without a truck. Constraints (6) simply enforce that a job's pick-up time is no earlier than its earliest pick-up time. Constraints (7) and (8) specify ranges of the amount of delay. Constraints (9) prevent rejection of jobs in the specified subset A .

3 Real-time Policies

In all the policies considered in this paper, a truck remains idle at the destination of its last job when not assigned to a new job. Under any given plan, a truck k is assigned a queue of jobs which have been (permanently or tentatively) accepted. If

truck k is currently idle, the queue is empty. If truck k is currently moving empty, the queue has at least one waiting job and truck k is moving toward the pick-up location of the first waiting job. Finally if truck k is moving loaded, the queue contains at least the job being currently served. For all policies, queues are non-preemptive: once a job is picked up, it is delivered without disruption.

3.1 A Simple Benchmark Policy

The first policy considered in this paper, **BENCH**, reflects what a company might do without the aid of sophisticated decision support systems. At a job arrival epoch, **BENCH** decides whether or not this new job is accepted, and, if accepted, assigns it to the queue of a specific vehicle k . These decisions are permanent and are based on a sequential evaluation. For each truck k , **BENCH** calculates the marginal cost of serving this new job if inserted at the end of its queue. In case all marginal costs are higher than the cost of rejection, the job is rejected. Otherwise it is assigned at the end of the queue of the truck k with the lowest marginal cost.

3.2 Advanced Policies

In these policies, initial acceptance/rejection decisions are not necessarily permanent, and a job being accepted or rejected at one decision epoch could be reconsidered before a permanent decision has to be made (based on the extra time T^{RES}). As introduced in Section 1, we use a list $L_{TEMP}^{\pi}(t)$ to represent at any time t the tentatively rejected jobs whose acceptance decision deadlines have not expired yet.

At a job arrival (decision) epoch τ , we first permanently remove jobs in $L_{TEMP}^{\pi}(\tau)$ whose response deadlines have expired (they become permanently rejected). For all remaining jobs in $L_{TEMP}^{\pi}(\tau)$ as well as the current new job (which triggered the current decision epoch), we need to decide whether to tentatively accept or reject them. For convenience, we refer to these jobs as the pending jobs. Also at the same time, some waiting jobs in the queue of some vehicles will have passed their acceptance decision deadlines and hence become permanently accepted. The other waiting jobs (tentatively accepted) could be potentially rejected as well at this new decision epoch. Out of the four advanced policies to be presented below, the last three will consider

this as an option.

The first two policies are local in the sense that, like **BENCH**, they evaluate the insertion of each pending job, one at a time, into each truck’s queue, also one at a time. Pending jobs are evaluated in the decreasing order of their arrival epochs. Each pending job is either inserted into a particular queue if the corresponding marginal cost is the smallest among all queues and is smaller than the cost of rejection, or is tentatively rejected otherwise. In the latter case, the pending job is added (back) to $L_{TEMP}^{\pi}(\tau)$.

The two local policies differ in how they consider insertion of a pending job in a truck’s queue. **NS** does not modify the relative ordering of the jobs already in the queue and only considers all possible insertions in between these jobs. It also doesn’t consider rejecting a tentatively accepted job in a queue while trying to insert the pending job. On the contrary, **SE** evaluates all possible orderings of the original waiting jobs together with the current pending job, and does so by solving a one-truck instance of the off-line problem. The optimal solution determines whether the current pending job and previously tentatively accepted jobs of the queue are accepted and, if accepted, in what order in the queue they should be served. If the pending job or a previously tentatively accepted job becomes tentatively rejected, it is added to $L_{TEMP}^{\pi}(\tau)$.

Strategies **BENCH**, **NS**, and **SE** are similar to strategies evaluated previously by Regan et al. 1998, albeit with very different implementations (in particular, we allow in the current paper rejection of a job based on cost considerations, and, acceptance/rejection decisions may not be immediately permanent). These strategies have also been considered in our previous paper Yang et 1998.

We also propose two re-optimization policies which consider, in one optimization run, all trucks, all acceptance/rejection and allocation decisions of pending and tentatively accepted waiting jobs, and all reallocation decisions of permanently accepted waiting jobs. **MYOPT** optimizes the acceptance and (re-)allocation decisions as if no future new job would ever be requested. It corresponds to solving a full instance of the off-line problem. Conceivably, this policy should perform better than any of the local policies. However this remains an empirical question and is investigated using a systematic simulation framework introduced in Section 4.

OPTUN operates in almost the same way as MYOPT. The only difference is that OPTUN introduces opportunity costs of serving jobs, somewhat accounting for future job requests. It assumes some knowledge about the probability law of future job pick-up (and delivery) locations. More precisely, let $\bar{D}(a)$ be the expected distance from a random point to point a and $\bar{\bar{D}}$ be the expected distance between two independent random points, where random points are distributed according to the probability law of job pick-up and delivery locations. Instead of using D_{0i}^k 's, D_{ij} 's, and W_i 's in the formulation of the off-line problem, OPTUN uses C_{0i}^k 's, C_{ij} 's, and γW_i 's, respectively. The new parameters are:

$$C_{0i}^k \equiv D_{0i}^k + K_1^O (\bar{D}(\mathbf{o}_i) - \bar{D}(\mathbf{l}_k)) + K_2^O (\bar{D}(\mathbf{d}_i) - \bar{\bar{D}}),$$

$$C_{ij} \equiv D_{ij} + K_1^O (\bar{D}(\mathbf{o}_j) - \bar{D}(\mathbf{d}_i)) + K_2^O (\bar{D}(\mathbf{d}_j) - \bar{\bar{D}}),$$

and

$$\gamma = (1 + K_3^O) \left(\sum_{k=1}^K \sum_{i=1}^N C_{0i}^k + \sum_{i=1}^N \sum_{j=1, j \neq i}^N C_{ij} \right) / \left(\sum_{k=1}^K \sum_{i=1}^N D_{0i}^k + \sum_{i=1}^N \sum_{j=1, j \neq i}^N D_{ij} \right),$$

where K_1^O , K_2^O , and K_3^O are exogenous parameters.

In the expressions of C_{0i}^k and C_{ij} , the term associated with K_1^O is to influence the vehicle-job assignment decision, and the term associated with K_2^O is to influence the job acceptance/rejection decision. The rationale behind these corrective terms is based on the following crude heuristic arguments, illustrated here for C_{0i}^k (the arguments being similar for C_{ij}):

The multiplicative term of K_1^O , i.e., $\bar{D}(\mathbf{o}_i) - \bar{D}(\mathbf{l}_k)$ represents a measure of the change in the opportunity cost for a truck moving empty from \mathbf{l}_k to \mathbf{o}_i . One can think of $\bar{D}(a)$ as the average distance between a and a potential future job's pickup location, and is thus a measure of how isolated the point a is. If the truck moves from \mathbf{l}_k to \mathbf{o}_i , and a new request comes when the truck is a fraction $0 \leq \theta \leq 1$ away from its departure, its new position has an "isolation" measure of $(1 - \theta)\bar{D}(\mathbf{l}_k) + \theta\bar{D}(\mathbf{o}_i)$. The difference between this and the isolation of the initial starting point is then $(1 - \theta)(\bar{D}(\mathbf{l}_k) - \bar{D}(\mathbf{o}_i))$. The parameter K_1^O is exogenously chosen and partially reflects what $(1 - \theta)$ would be on average.

The multiplicative term of K_2^O , i.e., $\bar{D}(\mathbf{d}_j) - \bar{\bar{D}}$ represents a measure of the resulting action of accepting job i and, after serving it up, of ending up in a location \mathbf{d}_i with an isolation measure, $\bar{D}(\mathbf{d}_j)$, significantly different from the one of a random point, $\bar{\bar{D}}$. This corrective term results in penalizing remote locations, and favoring central locations.

Finally the term γ and the associated parameter K_3^O are simply used to compensate for the inflation in the other parameters in the formulation.

4 Simulation Set-up

The goal of the simulation experiments is to compare the proposed policies under typical probabilistic settings and under various parameters. Because of the heavy computational requirements of individual simulation runs, a full factorial experimental design is neither practical nor necessary. Therefore, the policies are tested under several typical scenarios rather than the full range of possible occurrences.

Throughout the simulation study, we assume that (i) job arrival rate λ to be $1/T^{INT}$; (ii) pick-up and destination locations of the jobs are independent identically distributed uniform random variables in a unit square; and (iii) T_i^{ADV} 's, T_i^{SLK} 's, and T_i^{RES} 's are all drawn independently from uniform distributions with mean T^{ADV} , T^{SLK} , and T^{RES} , and ranges $[0, 2T^{ADV}]$, $[0, 2T^{SLK}]$, and $[0, 2T^{RES}]$, respectively.

In a unit square, the average distance between two points is approximately 0.522. So the maximum possible service rate per truck is $\mu \simeq 1/0.522 \simeq 1.916$ (this maximum service rate corresponds to a very high job arrival rate for which the empty distance from a job's destination to a next job's origin can be made arbitrarily small in expectation). We define the traffic intensity ρ to be $1/(KT^{INT}\mu)$. Without job rejection, ρ should be below 1 for the system to be stable. In order to be realistic and allow trucks to have some operational flexibility, we have chosen $\rho = 0.5$ as a default value. For given values of K and ρ , the inter-arrival time for demands is chosen as $T^{INT} = 1/(K\rho\mu)$.

Finally we assume that $\alpha = 1.0$ and $\beta = 0.2$. This choice of α implies that the cost per unit of empty distance has the same weight as the loss revenue per unit of loaded distance. Also $\beta = 0.2$ implies that 5 units of delay would offset the revenue

from any accepted job.

For every input parameter vector, and for every policy under investigation, we simulate $R = 10$ independent runs. Each policy experiences the same 10 independent runs. Each run starts with all trucks located at the central depot and simulates the arrivals of $n = 100 \times K$ jobs. Let $C_{n,r}$ denote the value of the function $C^\pi(\tau_n^{\pi,*})/n$ (see (3)) that we record for the r th run. The way $C_{n,r}$ is computed in our simulation is as follows. For each truck, we have a double-precision variable $ET[k]$ which records the truck's total empty travel distance at decision and job-completion epochs. For each job i , we have a double-precision variable $TCOM[i]$ variable which records the job's completion time and a binary variable $REJ[i]$ which indicates whether this job has been permanently rejected, at decision and job-completion epochs. When $\tau_n^{\pi,*}$ has been reached in this r th run, it is straightforward for us to use (1) and the three arrays of variables to calculate the corresponding $C_{n,r}$. The sample mean $AvgCost = \sum_{r=1}^R C_{n,r}/R$ serves as our approximation of the policy measure (4) defined in Section 1. From extensive initial tests, we find that this number of simulated arrivals is sufficient to guarantee steady-state behaviors and remove the effects of initial conditions. Also, due to the option of job rejection, the actual traffic intensity of the system is much smaller than 1 and so, for every policy and every batch of 10 independent runs, the sample variation of various results across runs stay well below 1% of their corresponding sample means.

The SE policy and the two re-optimization policies need to call CPLEX to solve instances of the off-line problem. To guarantee robustness and timeliness of the solutions, we limit the number of jobs involved in each optimization to a fixed upper-bound N_B (10 for the SE policy and 20 for the re-optimization policies). In order to do this, we both limit the size of $L_{TEMP}^\pi(t)$ to $N_B - 1$ (if the output of an off-line problem optimization leads to $TR > N_B - 1$ tentatively rejected jobs, then $TR - N_B + 1$ of them are picked at random and permanently rejected) and, if needed consider only few waiting jobs at the end of each queue (for the re-optimization strategies, this is done as evenly as possible across all trucks, by keeping on average only the last $(N_B - 1 - L_{TEMP}^\pi(t))/K$ jobs per queue). We also limit the total amount of time the SE and re-optimization policies spend solving each optimization problem to a fixed T_{LIM} (20 seconds). When the SE policy is used with N pending jobs, each

optimization is allocated a maximum time of $T_{LIM}/(K * N)$. When a re-optimization policy is used, each optimization is allocated a maximum time of T_{LIM} .

Table 1 lists various values of the parameters used in our main comparisons of the five proposed policies. The default parameter values in Table 1 are used as starting

Table 1: Values of Parameters in Main Simulation Results

	K	T^{SLK}	α	β	ρ	T^{ADV}	T^{RES}
Default	10	2.0	1.0	0.2	0.5	0.0	0.0
Table 2	10	2.0	1.0	0.2	0.5	0.0	0.0
Table 3	10	2.0	0.2	1.0	0.5	0.0	0.0
Table 4	10	0.5	1.0	0.2	0.5	0.0	0.0
Figure 2	10	2.0	1.0	0.2	$0.2 \sim 0.8$	0.0	0.0
Figure 3	10	2.0	1.0	0.2	0.5	$0.0 \sim 1.5$	0.0
Figure 4	10	2.0	1.0	0.2	0.5	0.25	$0.0 \sim 0.375$

points to find good values for the remaining parameters associated with the policies under investigation. We find that OPTUN works best when $K_1^O = 0.12$, $K_2^O = 0.10$, and $K_3^O = 0.06$. Finally, as mentioned before, for the SE and re-optimization policies, we always let T_{LIM} to be 20.0 seconds and N_B to be 10 for SE and 20 for the re-optimization policies.

Assuming that all these parameters are given, each simulation is now parameterized by a vector $(K, T^{SLK}, \alpha, \beta, \rho, T^{ADV}, T^{RES})$. In our implementation, the instances of the off-line problem are solved by the commercial CPLEX 6.5 solver. The simulation source code is written in C language. All the runs have been conducted on a Dell OptiPlex machine with a Pentium II processor.

List of all input parameters:

R : number of independent runs; Its default value: 10;

K : number of trucks; Its default value: 10;

n : number of jobs; Its default value: $100 \times K = 1000$;

ρ : traffic intensity; Its reasonable range: $0.2 \sim 0.8$; Its default value: 0.5;

α : relative weight of cost due to empty traveling vs. cost due to job rejection; Its

default value: 1.0;

β : relative weight of cost due to job waiting vs. cost due to job rejection; Its default value: 0.2;

T^{ADV} : the average T_i^{ADV} ; Its reasonable range: 0 ~ 1.5;

T^{SLK} : the average T_i^{SLK} ; Its default value: 2.0;

T^{RES} : the average T_i^{RES} ; Its default value: 0.0;

N_B : maximumly-allowed number of jobs to be involved in each optimization; Its default values: 10 for the SE policy and 20 for the re-optimization policies;

T_{LIM} : maximumly-allowed amount of optimization time to be spent during one decision epoch in the SE and re-optimization policies; Its default value: 20 seconds.

5 Simulation Results

The first set of simulation experiments are performed with the default parameter values. The results are shown in Table 2. In the table, $RjcRate$ is the average rate of

Table 2: Performance of Policies under Typical Parameters

Policy	$RjcRate$	$EmpDist$	$DelayWt$	$RjLDist$	$AvgCost$
BENCH	0.154	0.197	0.061	0.236	0.213
NS	0.097	0.177	0.091	0.221	0.198
SE	0.101	0.174	0.107	0.226	0.199
MYOPT	0.092	0.155	0.050	0.209	0.169
OPTUN	0.076	0.155	0.047	0.188	0.166

jobs being rejected, $EmpDist$ is the average empty distance traveled by the trucks per accepted job, $DelayWt$ is the average weighted delay per accepted job, $RjLDist$ is the average distance of the rejected jobs, and $AvgCost$ is the average cost incurred per requested job. Note that $AvgCost$ is the value of the objective minimized (expressed on a per requested job basis), and is therefore the ultimate figure of merit in this evaluation. Under the default parameters, the re-optimization policies appear to outperform the more limited policies by a significant margin. The results confirm the

value of seeking optimal solutions at each decision epoch, even when the formulation is limited to consideration of only those loads that have already materialized.

The policies are compared under a different combination of parameter values, in which delay time is given greater weight by increasing the value of β from 0.2 to 1.0, and the empty distance is correspondingly de-emphasized by reducing α from 1.0 to 0.2. The results, shown in Table 3, again indicate that the re-optimization policies outperform the local policies.

Table 3: Performance of Policies when Delay Penalty is Relatively More Important

Policy	<i>RjcRate</i>	<i>EmpDist</i>	<i>DelayWt</i>	<i>RjLDist</i>	<i>AvgCost</i>
BENCH	0.013	0.226	0.014	0.179	0.061
NS	0.006	0.211	0.012	0.163	0.054
SE	0.006	0.210	0.012	0.087	0.054
MYOPT	0.004	0.181	0.008	0.065	0.045
OPTUN	0.004	0.180	0.007	0.064	0.043

In the next set of experiments, all parameters are kept at their default levels, with the exception of the average time until latest pick-up T^{SLK} , which is reduced from 2.0 to 0.5, reflecting tighter pick-up windows and greater job urgency than the default scenario. The results are shown in Table 4, indicating that re-optimization policies again outperform local policies, though by a smaller margin (about 10% in terms of *AvgCost*) than in the less constrained cases. The simulation experiments shown here clearly indicate that optimization over available job requests at each decision epoch leads to better overall (over the entire sequence of load requests) job acceptance/rejection decisions, and shorter empty distance than the more local strategies considered here. Under all situations considered, applying re-optimization policies appears to produce significant savings in operating costs.

The next set of simulations examine how the policies fare under varying degrees of relative saturation in the system, captured by the index ρ . The results are shown in Figure 2. The most striking phenomenon here is the widening of the gap between the respective performance of the local and re-optimization policies up to $\rho = 0.7$.

Table 4: Performance of Policies when Jobs are very Urgent

Policy	$RjcRate$	$EmpDist$	$DelayWt$	$RjLDist$	$AvgCost$
BENCH	0.168	0.194	0.162	0.251	0.231
NS	0.132	0.182	0.191	0.247	0.224
SE	0.133	0.180	0.196	0.252	0.224
MYOPT	0.122	0.166	0.170	0.228	0.203
OPTUN	0.102	0.167	0.160	0.217	0.201

When ρ increases, the average number of jobs at each decision epoch increases and action on one job affects more jobs. Also re-optimization policies generate even better payoffs under higher traffic intensities. Finally, the increase of ρ makes the knowledge about future jobs more important. From the widening of the gap between OPTUN and MYOPT in the first half of the experiment, we see that OPTUN utilize the distributional information about jobs in a more efficient way. The jump in the average cost under both re-optimization policies at highest saturation rate (from $\rho = 0.7$ to $\rho = 0.8$) is due to the computational limitations imposed on the solution of individual problem instances at each decision epoch. In fact, in these experiments (with a 20-second limit on any problem instance), only about 14% of the optimizations reached duality gaps within one percent.

Next, we investigate the effect of advanced information. With all other parameters at their default levels, we vary the average time that a job is requested prior to its earliest pick-up time, T^{ADV} . The results are shown in Figure 3. BENCH is not very sensitive to the change of T^{ADV} . Its performance even degrades when T^{ADV} becomes too big. This degradation can be partially explained by the fact that this policy inserts jobs at the end of the queues, even though they could be available for pick-up earlier than for any other jobs currently in the queues.

The performance of SE and the two re-optimization policies improve as T^{ADV} increases. For SE, the range where T^{ADV} has a visible effect is from 0 to 0.7 and the maximal improvement is about 3%. For the two re-optimization policies, the range where T^{ADV} has visible effects is from 0 to 1.0 and the maximal improvements are

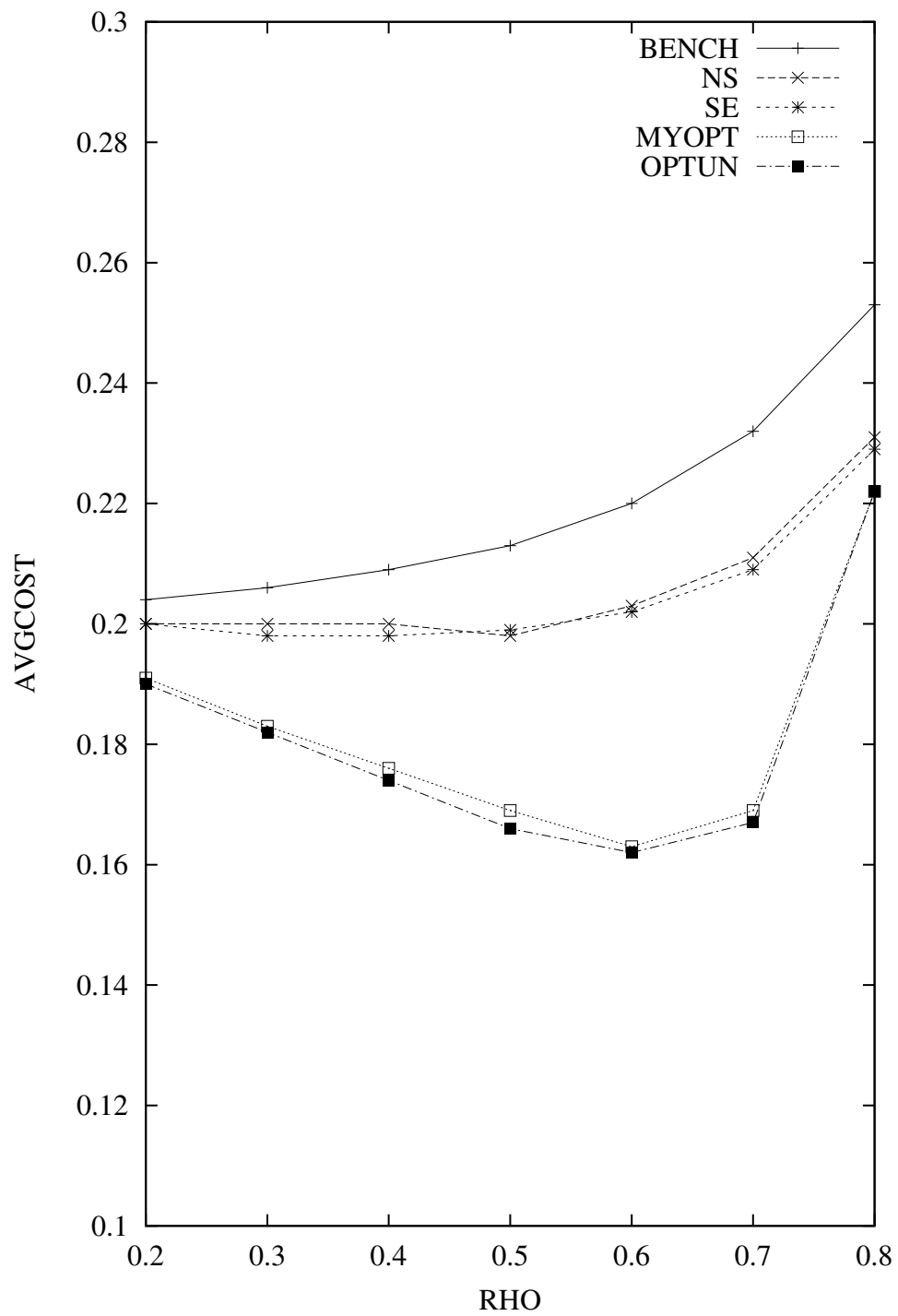


Figure 2: Performance of Policies when ρ is varying

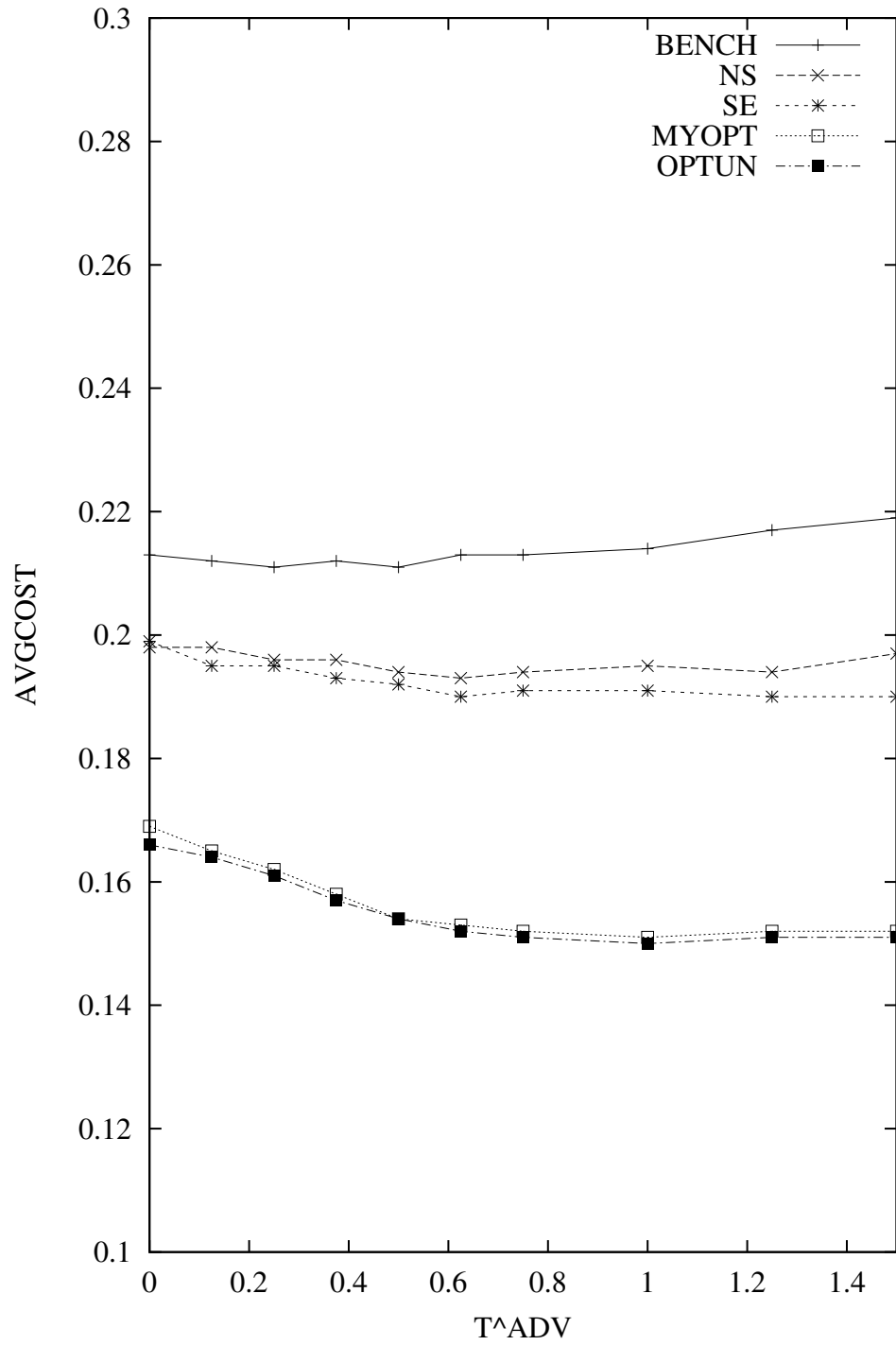


Figure 3: Value of Advance Information: Comparative Performance of Policies when T^{ADV} is varying

about 10%.

Finally, we conduct another simulation to study the effect of T^{RES} . In this simulation, all the parameters stay typical, except that we let $T^{ADV} = 0.25$ and T^{RES} vary. The results are shown in Figure 4.

By definition, BENCH is not affected by T^{RES} at all since its decisions are made permanently at job arrival epochs. For all other policies, changes brought by the varying T^{RES} are visible yet not remarkable.

6 Concluding Remarks

In this paper, we have introduced and studied a generic real-time truckload pick-up and delivery problem in a very general framework in which various costs due to job rejection, empty travel of trucks and delay time of job completions are taken into considerations. The framework also facilitates investigation of the value of advanced information.

We have evaluated several rolling-horizon policies based on various heuristics either previously introduced in the literature, or proposed here for the first time. We found that the policies based on fully optimizing the off-line model of the problem perform very competitively with other policies under typical cost structures. The best policy we found is the one that takes some future job distribution into consideration. We also found that advanced information is very useful for some of the policies.

We think future research should concentrate on the search for better policies that utilize some information about future jobs more efficiently. From the improvement of OPTUN over MYOPT, we believe that there is still much potential for progress left uncovered.

Acknowledgements

The authors thank two referees and Teo Crainic for comments that greatly helped improve the quality and clarity of the paper. The first author was supported by the National Center of Transportation and Industrial Productivity, Grant No. 992518.

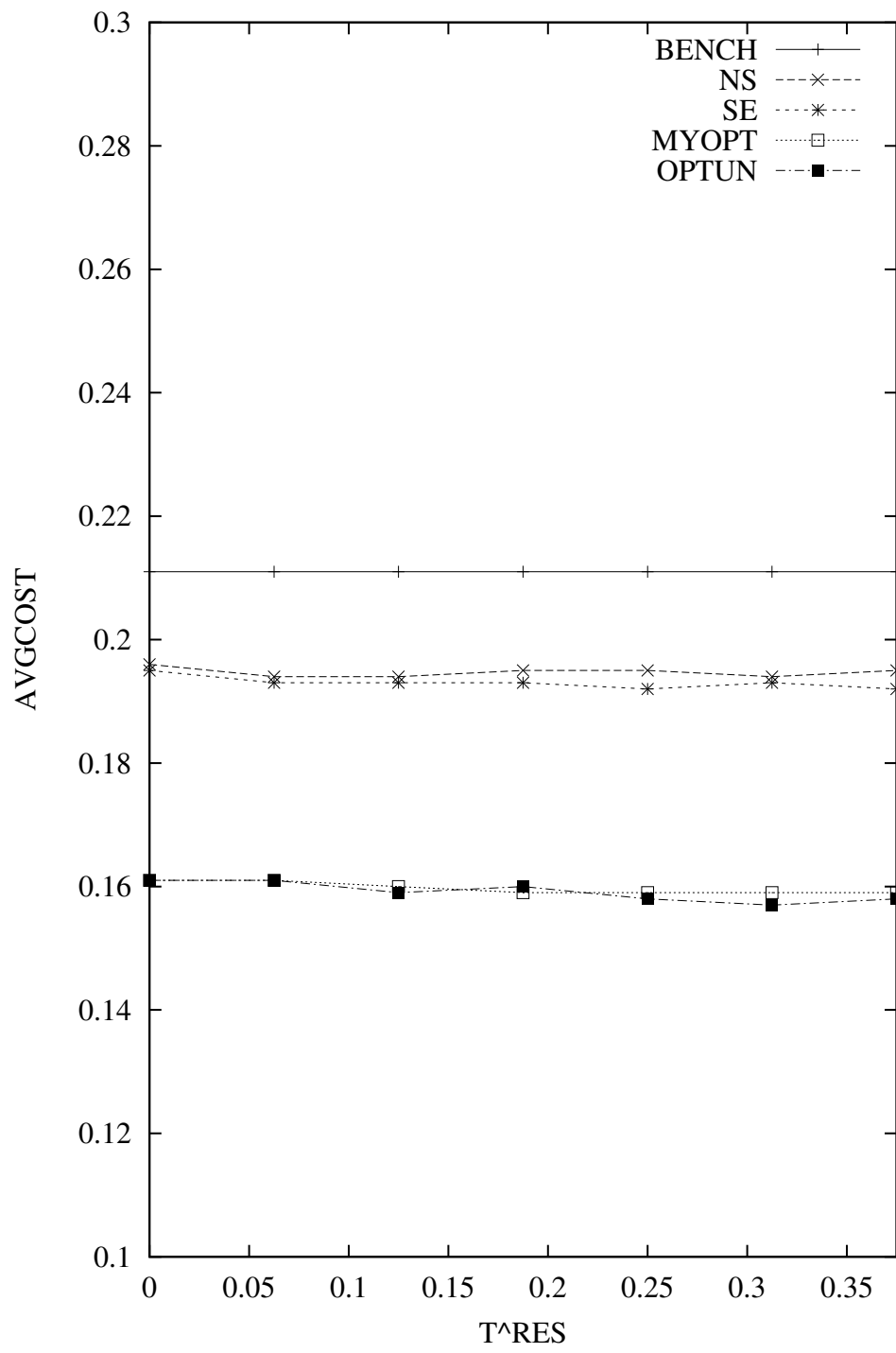


Figure 4: Performance of Policies when T^{RES} is varying

The second author was supported by NSF, Grant DMI-9713682.

References

Bastian, C., A.H.G. Rinnooy Kan. 1992. The Stochastic Vehicle Routing Problem Revisited. *European Journal of Operational Research* **56** 407-412.

Berman, O., D. Simchi-Levi. 1988. Finding the Optimal a Priori Tour and Location of a Traveling Salesman with Nonhomogeneous Customers. *Transportation Science* **22** 148-154.

Berman, O., D. Simchi-Levi. 1989. The Traveling Salesman Location Problem on Stochastic Networks. *Transportation Science* **23** 54-57.

Bertsimas, D.J. 1988. Probabilistic Combinatorial Optimization Problems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Bertsimas, D.J. 1992. A Vehicle Routing Problem with Stochastic Demand. *Operations Research* **40** 574-585.

Bertsimas, D.J., L.H. Howell. 1993. Further Results on the Probabilistic Traveling Salesman Problem. *European Journal of Operational Research* **65** 68-95.

Bertsimas, D.J., D. Simchi-Levi. 1996. A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty. *Operations Research* **44** 286-304.

Bertsimas, D.J., G. Van Ryzin. 1991. A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane. *Operations Research* **39** 601-615.

Bertsimas, D.J., G. Van Ryzin. 1993a. Stochastic and Dynamic Vehicle Routing in the Euclidean Plane with Multiple Capacitated Vehicles. *Operations Research* **41** 60-76.

Bertsimas, D.J., G. Van Ryzin. 1993b. Stochastic and Dynamic Vehicle Routing With General Demand and Interarrival Time Distributions. *Advances in Applied Probability* **25** 947-978.

Bienstock, D., J. Bramel, D. Simchi-Levi. 1993. A Probabilistic Analysis of Tour

Partitioning Heuristics for the Capacitated Vehicle Routing Problem with Unsplit Demands. *Mathematics of Operations Research* **18** 786-802.

Borodin, A., Ran El-Yaniv. 1998. *Online Computation and Competitive Analysis* Cambridge University Press.

Bodin, L.D., B.L. Golden, A.A. Assad, M. Ball. 1983. Routing and Scheduling of Vehicle and Crews, the State of the Art. *Computers and Operations Research* **10** 69-211.

Bookbinder, J.H., S.P. Sethi. 1980. The Dynamic Transportation Problem: A Survey. *Naval Research Logistics* **27** 447-452.

Bramel, J., D. Simchi-Levi. 1996. Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows. *Operations Research* **44** 505-509.

Bramel, J., D. Simchi-Levi. 1997. On the Effectiveness of the Set Covering Formulations for the Vehicle Routing Problem with Time Windows. *Operations Research* **45** 295-301.

Bramel, J., C.L. Li, D. Simchi-Levi. 1994. Probabilistic Analysis of the Vehicle Routing Problem with Time Windows. *American Journal of Mathematics and Management Science* **13** 267-322.

Bramel, J., E.G. Coffman Jr., P. Shor, D. Simchi-Levi. 1992. Probabilistic Analysis of Algorithms for the Capacitated Vehicle Routing Problem with Unsplit Demands. *Operations Research* **40** 1095-1106.

Christofides, N. 1985. Vehicle Routing. in Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, D. Shmoys (eds.), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* John Wiley and Sons, New York.

Cook, T.M., R.A. Russell. 1978. A Simulation and Statistical Analysis of Stochastic Vehicle Routing with Timing Constraints. *Decision Science* **9** 673-687.

Desrochers, M., J.K. Lenstra, M.W.P. Savelsbergh, F. Soumis. 1988. Vehicle Routing with Time Windows: Optimization and Approximation. in B.L. Golden and

- A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*. Elsevier (North-Holland), Amsterdam, 65-84.
- Dejax, P.J., T.G. Crainic. 1987. Models for Empty Freight Vehicle Transportation Logistics. *Transportation Science* **21** 227-247.
- Dror, M., P. Trudeau. 1986. Stochastic Vehicle Routing with Modified Savings Algorithm. *European Journal of Operational Research* **23** 228-235.
- Dror, M., G. Laporte, P. Trudeau. 1989. Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks. *Transportation Science* **23** 166-176.
- Fisher, M.L. 1995. Vehicle Routing. in Ball, M.O., T.L. Magnanti, C.L. Monma, G.L. Nemhauser (eds.), *Handbooks in Operations Research and Management Science, Vol 8, Network Routing* Elsevier, Amsterdam, 1-33.
- Frantzekakis, L.F., W.B. Powell. 1990. A Successive Linear Approximation Procedure for Stochastic, Dynamic Vehicle Allocation Problems. *Transportation Research* **24** 40-57.
- Garfinkel, R.S. 1985. Motivation and Modeling. in Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (eds.), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* Wiley, New York, 17-36.
- Gendreau, M., J.-Y. Potvin. 1998. Dynamic Vehicle Routing and Dispatching. in Crainic, T. Laporte, G. (eds), *Fleet Management and Logistics* Kluwer, 115-126.
- Gendreau, M., F. Guertin, J.-Y. Potvin, E. Taillard. 1999. Parallel Tabu Search for Real-time Vehicle Routing and Dispatching. *Transportation Science* **33** 381-390.
- Golden, B.L., A.A. Assad (eds.). 1988. *Vehicle Routing: Methods and Studies* Elsevier (North-Holland), Amsterdam.
- Golden, B.L., W.R. Stewart. 1978. Vehicle Routing with Probabilistic Demands. in Hogben, D., D. Fife (eds.), *Computer Science and Statistics: Tenth Annual Symposium on the Interface* NBS Special Publication **503** 252-259.
- Golden, B.L., J.R. Yee. 1979. A Framework for Probabilistic Routing. *AIEE Trans-*

actions **11** pp. 109-112.

Heyman, D., M. Sobel. 1984. *Stochastic Models in Operations Research*. Mc Graw-Hill.

Ichoua, S., M. Gendreau, J.-Y. Potvin. 2000. Diversion Issues in Real-time Vehicle Dispatching. *Transportation Science* **34** 426-438.

Jaillet, P. 1985. Probabilistic Traveling Salesman Problem. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Jaillet, P. 1988. A Priori Solution of a Traveling Salesman Problem in Which a Random Subset of the Customers are Visited. *Operations Research* **36** 929-936.

Jezequel, A. 1985. Probabilistic Vehicle Routing Problems. M.Sc. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Laporte, G., F.V. Louveaux, H. Mercure 1994. A Priori Optimization of the Probabilistic Traveling Salesman Problem. *Operations Research* **42** 543-549.

Powell, W.B. 1986. A Stochastic Model of the Dynamic Vehicle Allocation Problem. *Transportation Science* **20** 117-129.

Powell, W.B. 1987. An Operational Planning Model for the Dynamic Vehicle Allocation Problem with Uncertain Demands. *Transportation Research* **21B** 217-232.

Powell, W.B. 1988. A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem. in Golden, B.I., A.A. Assad (eds.), *Vehicle Routing: Methods and Studies* North-Hollana, Amsterdam, 249-291.

Powell, W.B. 1996. A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carrier. *Transportation Science* **30** 195-219.

Powell, W.B., P. Jaillet, A. Odoni. 1995. Stochastic and Dynamic Networks and Routing. in Ball, M.O., T.L. Magnanti, C.L. Monma, G.L. Nemhauser (eds.), *Handbooks in Operations Research and Management Science, Vol 8, Network Routing* EL-

sevier, Amsterdam, 141-296.

Powell, W.B., Y. Sheffi, S. Thiriez. 1984. The Dynamic Vehicle Allocation Problem with Uncertain Demands. *Ninth International Symposium on Transportation and Traffic Theory* VNU Science Press, Utrecht.

Powell, W.B., W. Snow, R.K. Cheung. 2000a. Adaptive Labeling Algorithms for the Dynamic Assignment Problem. *Transportation Science* **34** 50-66.

Powell, W.B., M.T. Towns, A. Marar. 2000b. On the Value of Globally Optimal Solutions for Dynamic Routing and Scheduling Problems. *Transportation Science* **34** 67-85.

Psaraftis, H.N. 1988. Dynamic Vehicle Routing Problems. in Golden, B.L., A.A. Assad (eds.), *Vehicle Routing: Methods and Studies* Elsevier (North-Holland), Amsterdam, 223-248.

Psaraftis, H.N. 1995. Dynamic Vehicle Routing: Status and Prospects. *Annals of Operations Research* **61** 143-164.

Regan, A.C., H.S. Mahmassani, P. Jaillet 1995. Improving the Efficiency of Commercial Vehicle Operations Using Real-Time Information: Potential Uses and Assignment Strategies. *Transportation Research Record* **1493** 188-198.

Regan, A.C., H.S. Mahmassani, P. Jaillet 1996a. Dynamic Decision Making for Commercial Fleet Operations Using Real-Time Information. *Transportation Research Record* **1537** 91-97.

Regan, A.C., H.S. Mahmassani, P. Jaillet 1996b. Dynamic Dispatching Strategies under Real-Time Information for Carrier Fleet Management. in Lesort, J.B. (ed), *Transportation and Traffic Theory* Pergamon, 737-756.

Regan, A.C., H.S. Mahmassani, P. Jaillet 1998. Evaluation of Dynamic Fleet Management Systems: Simulation Framework. *Transportation Research Record* **1645** 176-184.

Rossi, F.A., I. Gavioli. 1987. Aspects of Heuristic Methods in the Probabilistic

Traveling Salesman Problem. in Andreatta, G., F. Mason, and P. Serafini (eds.), *Advanced School on Stochastics in Combinatorial Optimization* World Scientific, Singapore, 214-227.

Solomon, M.M. 1987. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research* **35** 254-265.

Stewart, W.R. 1981. New Algorithms for Deterministic and Stochastic Vehicle Routing Problems. Working Paper No. 81-009, University of Maryland.

Stewart, W.R., B.L. Golden. 1983. Stochastic Vehicle Routing: A Comprehensive Approach. *European Journal of Operational Research* **14** 371-385.

Tillman, F.A. 1969. The Multiple Terminal Deliver Problem with Probabilistic Demands. *Transportation Science* **3** 192-204.

Yang, J., P. Jaillet, H.S. Mahmassani. 1998. On-line Algorithms for Truck Fleet Assignment and Scheduling under Real-time Information. *Transportation Research Record* **1667** 107-113.

Yee, J.R., B.L. Golden. 1980. A Note on Determining Operating Strategies for Probabilistic Vehicle Routing. *Naval Research Logistics Quarterly* **27** 159-163.

Received: December 2000; revisions received: November 2001, January 2002, June 2002, November 2002; Accepted December 2002.