

# Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching

Soumia Ichoua

Département d'opérations et systèmes de décision, and Centre de recherche sur les technologies de l'organisation réseau,  
Université Laval, Québec, Québec, Canada G1K 7P4, soumia.ichoua@fsa.ulaval.ca

Michel Gendreau, Jean-Yves Potvin

Département d'informatique et de recherche opérationnelle, and Centre de recherche sur les transports,  
Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Québec, Canada H3C 3J7  
{michelg@crt.umontreal.ca, potvin@iro.umontreal.ca}

An important, but seldom investigated, issue in the field of dynamic vehicle routing and dispatching is how to exploit information about future events to improve decision making. In this paper, we address this issue in a real-time setting with a strategy based on probabilistic knowledge about future request arrivals to better manage the fleet of vehicles. More precisely, the new strategy introduces dummy customers (representing forecasted requests) in vehicle routes to provide a good coverage of the territory. This strategy is assessed through computational experiments performed in a simulated environment.

*Key words:* vehicle dispatching; real time; probabilistic knowledge; future events; parallel tabu search

*History:* Received: August 2001; revisions received: October 2003, July 2004, October 2004; accepted: January 2005.

## 1. Introduction

The rapid growth in communication and information technologies, as well as the increased interest in just-in-time distribution systems, have recently led researchers to focus on dynamic vehicle routing and scheduling problems. In the static version of these problems, the goal is to design a set of minimum-cost routes to satisfy the demand while satisfying various constraints. In a real-time setting, service requests arise continuously and randomly over time and must be assigned to vehicles. In this context, any decision is based on incomplete information, given that some requests have not arisen yet. Furthermore, the actual demand or time to start service at customer locations may be prone to random variations over time. In this context, uncertainty is an inherent characteristic of the problem.

Fortunately, future events often arise in a somewhat predictable way (e.g., probability distributions can be derived from historical data). An important issue that has seldom been investigated in the field of dynamic vehicle routing and dispatching is how to exploit probabilistic knowledge about future events to improve decision making. Actually, exploiting information about future events is likely to result in better management of the resources at the current time, which implies that future needs will be met more efficiently.

The work reported in this paper is motivated from the local operations of long-distance express mail

services (like Federal Express). Here, customers call a central office when their mail is ready to be delivered abroad. These requests are collected by vehicles and brought back to a central office for further processing and shipping. With regard to this practical problem, our contribution is the development, integration within a previously myopic approach, and empirical evaluation of a threshold-based heuristic that accounts for future customer requests, in a realistic context where multiple vehicles are involved.

The paper is organized as follows. First, §2 provides a brief literature review dedicated to stochastic and dynamic vehicle routing problems, with a particular emphasis on the exploitation of knowledge about future customer requests. In §3, a dynamic programming framework is used to formally introduce our problem. This is followed in §4 by a description of a problem-solving strategy that accounts for future requests. Section 5 briefly describes the original tabu search heuristic of Gendreau et al. (1999) and explains how it was modified to integrate the new strategy. Computational results are reported in §6. Finally, §7 summarizes our major findings.

## 2. Literature Review

In the field of vehicle routing and scheduling, the issue of exploiting knowledge about future demands to improve decision making has sparked a lot of interest in the last decade. This issue has been addressed

in the literature under two different problem classes (which are associated with two different ways of tackling the uncertainty, one static, the other dynamic). The existing work for each class is reported in the two following subsections.

### 2.1. Static Stochastic Vehicle Routing Problems

In this problem class, the customers and/or their demand are discrete or continuous random variables. Probabilistic information about future events is used to construct an *a priori* solution that optimizes the expected value of a given objective function. The probabilistic traveling salesman problem (PTSP), the vehicle routing problem with stochastic demands (VRPSD), the vehicle routing problem with stochastic customers (VRPSC), and the vehicle routing problem with stochastic customers and demands (VRPSCD) all belong to this class. A survey of these problems can be found in Gendreau, Laporte, and Séguin (1996a, b). In the following, these problems are briefly described and an overview of some related work is provided.

The PTSP is derived from the classical TSP by assigning to each potential customer a probability that this customer will require service. The goal is to find a single tour of minimum expected length. This problem was first introduced by Jaillet (1985, 1988) and investigated by many researchers (e.g., Jezequel 1985; Bertsimas and Howell 1993; Laporte, Louveaux, and Mercure 1994). In the VRPSD, vehicles of fixed capacity must service customers with stochastic demands. Some recent papers have been devoted to this problem (Bertsimas 1988; Dror, Laporte, and Trudeau 1989; Séguin 1994; Gendreau, Laporte, and Séguin 1995; Yang, Mathur, and Ballou 2000; Secomandi 1998, 2000). In the VRPSC, each customer has some probability of requiring service, but the demand is deterministic (Bertsimas 1988; Waters 1989). The VRPSCD combines the characteristics of the VRPSC and VRPSD, as both customers and demands are stochastic. This problem has been investigated in Bertsimas, Jaillet, and Odoni (1990), Bertsimas (1992), Gendreau, Laporte, and Séguin (1995, 1996), and Séguin (1994).

Apart from the work of Secomandi (1998, 2000) on the VRPSD, it is generally assumed that the probability distributions are known and are exploited to construct a solution in two stages. In the first stage, a planned (or “*a priori*”) route is constructed that minimizes an objective function related to some expectancy measure, like the expected length. When the routes are actually executed in the second stage, recourse actions are applied to the first-stage solution when needed to address the current realization (e.g., “jumping over” a customer who does not show). The interested reader will find more details about this kind of solution approach in Dror, Laporte, and

Trudeau (1989) and Gendreau, Laporte, and Séguin (1996).

Secomandi (1998, 2000) addressed a variant of the VRPSD for a single vehicle. Here, customer demands are assumed to follow a discrete probability distribution. It is also assumed that the actual demand is revealed on arrival at each customer location. In the case of a route failure (i.e., the vehicle capacity is exceeded at a customer location), a recourse action is taken that consists of sending the vehicle back to the depot before it resumes its tour. The author presents a stochastic shortest path formulation of the problem based on a Markov decision process. Different dynamic programming formulations are developed and analyzed. State-space decompositions, which can optimally solve small instances, are also derived. For large instances, suboptimal policies are developed using different heuristics based on a neurodynamic programming (NDP) approach—which is aimed at approximately solving large or complex dynamic programming problems; see Bertsekas and Tsitsiklis (1996), and Sutton and Barto (1998) for more details on NDP. The heuristics developed by the author assume that a set of states and “cost-to-go” values belonging to an initial policy can be generated through simulation. The current policy is then iteratively improved until a given policy is reached or a prespecified number of iterations is reached. At each iteration, the current policy is applied to derive a least-square approximation of the cost-to-go values. A simulation is then performed to generate a new set of states and the least-square approximation is used to evaluate the cost-to-go values of the next policy. Secomandi (2000) compared two variants of these heuristics: the rollout policy (RP) and the optimistic approximate policy iteration (OAPI). The former is a simple, yet more robust, variant of NDP. In OAPI, the cardinality of the generated sets is smaller than what is typically found in NDP, thus allowing more iterations to be performed. Computational experiments on small instances with 5, 10, and 15 customers show, however, that RP outperforms OAPI.

### 2.2. Dynamic Stochastic Vehicle Routing Problems

In this section, we consider problems where new events occur over time. Thus, no solution is constructed “*a priori*” in this case. Assuming that some probabilistic knowledge about future events is available, two different ways of exploiting this information are reported in the literature. This is the subject of the following subsections.

**2.2.1. Analytical Studies.** Here, the only source of uncertainty is the occurrence of new requests and/or the service (or dwell) times at customers. The probability distribution of these random variables is used

to perform analytical studies aimed at examining the robustness and asymptotic behavior of different deterministic and myopic dispatching strategies. Bertsimas and Simchi-Levi (1996) pointed out that this kind of work also provides new insights into the solution structure, thereby offering valuable guidelines for the design of efficient deterministic algorithms. An example is found in Bertsimas and van Ryzin (1991), where the authors consider the dynamic traveling repairman problem (DTRP). In this problem, a vehicle is used to service customer demands that occur according to a Poisson process. The service time at each customer is also a random variable. The goal is to find a routing policy that minimizes the expected system time, which is the time spent in the system by each customer. Using results from geometrical probability, combinatorial optimization and queuing theory, the authors provide an analytical study of the problem in the case of light traffic and heavy traffic. First, the authors derive lower bounds for the average system time. Then they analyze several simple policies (e.g., first-come-first-served) and compare their performance with these lower bounds. In 1993, the same authors (Bertsimas and van Ryzin 1993) extend their findings to the case of a homogeneous fleet of vehicles where each vehicle cannot visit more than a given number of customers before returning to the depot. The uncapacitated version of this problem is also investigated. The authors partition the given area into subregions and use the results of the first paper to propose and analyze new policies in case of light and heavy traffic, respectively. The work in Larsen, Madsen, and Solomon (2002) also focuses on routing policies for the (partially dynamic) DTRP, where only a subset of requests occurs in real time. Their study indicates that the nearest neighbor (NN) policy, where the repairman always goes to the nearest demand, is superior to the other tested policies, including an approach where the service area is first partitioned into subregions and a first-come-first-served policy is applied within each subregion.

Swihart and Papastavrou (1999) introduce a new variant of the DTRP where each service request has a pickup and a delivery location. The objective is to minimize the expected system time. The authors consider the unit-capacity case where the vehicle can carry no more than one item, as well as the case where the vehicle can carry an arbitrarily large number of items. In each of these two cases, lower bounds are derived for the average system time in light and heavy traffic conditions. Several policies are also proposed and compared.

Thomas and White (2004) addressed a variant of the vehicle routing problem with pickup and delivery, where a single vehicle travels from a known origin to a known destination. A penalty is incurred if

the destination is reached after a fixed deadline. It is assumed that potential customers may request service while the vehicle is in transit. The probability that a customer will request a pickup before time  $t$  is assumed to be known. The problem is to determine a policy for selecting the next destination that minimizes the expected total cost (travel time plus penalty for lateness). The problem is modeled as a finite-horizon Markov decision process. Some fundamental structural features of the optimal policy are established. These results are then used to characterize the behavior of policies that optimally select routes in anticipation of future service requests (for one to three dynamic requests). Numerical experiments compare these policies with a reactive strategy that ignores potential customer requests. Experiments were also performed in the case of a single dynamic request. However, the results did not clearly show the efficiency of the proposed anticipatory policies due to significant data dependencies.

In his dissertation, Rubio Maqueo (1995) studied the vehicle routing and inventory problem (VRIP). In this problem, a single capacitated vehicle operates out of a single warehouse and services a finite set of retailers. The arrival of customer requests at a given retailer is modeled as an independent renewal process while the demand magnitude is stochastic. Furthermore, the travel times between retailers are random variables. The objective is to minimize the expected inventory and transportation costs. Assuming that the system operates in heavy traffic, the author analyzes two models. In the first one, the vehicle follows a pre-defined routing scheme: deliver full loads to either a single retailer (direct shipping (DS)), or along a pre-specified TSP tour. In the second model, which is based on the current inventory levels, the controller dynamically chooses whether the vehicle should follow a TSP or DS routing scheme. Using simulation and queuing theory, the author fully characterizes a dynamic control policy that is asymptotically optimal for the fixed routing scheme. In addition, the author investigates the importance of various operational decisions (e.g., TSP versus DS, fixed versus dynamic routing), thereby giving insights into the nature of the optimal solution to the VRIP.

Kleywegt (1996), Kleywegt and Papastavrou (1998, 2001), and Papastavrou, Rajagopalan, and Kleywegt (1996) considered a problem called the dynamic and stochastic knapsack problem (DSKP). In this problem, demands for a given resource occur according to some stochastic process. Each demand requires a random amount of resource and has a random profit. Demands can be either accepted or rejected. If a demand is accepted, an associated reward is gained. Otherwise, a penalty is incurred. There is also a deadline, possibly infinite, after which demands can

no longer be accepted (although the process can be stopped at any time before this deadline). The objective is to determine an optimal policy for accepting the demands and for stopping the process while maximizing the expected reward value. The authors consider both the infinite and finite-horizon cases. They establish a number of theoretical characteristics for the optimal accepting rule and for the optimal value function, and use these characteristics to derive an optimal policy for the problem. In the second part of his thesis, Kleywegt (1996) formulates and analyzes the dynamic and stochastic distribution problem (DSDP) with a network of terminals and a fleet of vehicles. In this problem, customers call at random times for the transportation of a set of loads from given origins to given destinations. Each accepted request generates a reward and each rejected request incurs a penalty. There is also a holding cost per unit of time that depends on the number of vehicles and number of loads waiting at each terminal. The objective is to maximize the expected total value (rewards minus costs). The problem is formulated as a Markov decision process. Equations are derived for the optimal expected value and some fundamental structural characteristics of the solution are established. A two-stage algorithm is also proposed. The first stage is a threshold acceptance rule, which is shown to be optimal, and the second stage is an optimal dispatching rule.

**2.2.2. Stochastic Algorithms.** This line of research has sought to develop explicit stochastic and dynamic algorithmic approaches that incorporate the current information and probabilities of future events to produce more efficient solutions than those obtained through myopic deterministic strategies. The only work that we are aware of in this category is that of Powell and his team (Powell 1988, 1996; Godfrey and Powell 2002; Powell and Topaloglu 2003; Spivey and Powell 2004) and a research done in Larsen (2000) and Larsen, Madsen, and Solomon (2002). In the following, we briefly discuss these studies.

The problem considered by Powell is motivated from long-haul truckload trucking applications. In this context, the problem consists of dynamically assigning drivers to loads that arise randomly over time. Each load is characterized by its origin, its destination, and its time window. At any time, a vehicle can be either empty or carrying a single load. Furthermore, a demand may be rejected if it cannot be serviced within reasonable time. Powell (1988) first addressed the issue of relocating vehicles in anticipation of future demands. He proposed a dynamic network model based on a time-space graph where the nodes correspond to known and forecasted demands in different zones at different time periods. In addition to arcs that represent moves between regions, a

dummy arc from each origin to a super-sink captures the impact of the current time period on the future. The cost on this arc is the expected reward value of having an additional vehicle in the associated region at the beginning of the next time period. This model has some limitations because a forecasted load may be serviced even if it never materializes. To overcome this weakness, the author then proposed a stochastic, dynamic model where the demand between two nodes at a given time period is a random variable. Hence, the flows of loaded and empty vehicles moving between the two nodes are also random variables. This model is written as a two-stage stochastic program where the objective function combines a deterministic cost that corresponds to the first time period (the present) and the value of the recourse function that represents the expected cost from the next time period to the end of the time horizon (the future).

In Powell (1996), the author investigates in the same context a hybrid model for solving a combined assignment and fleet management problem. The model is based on a dynamic-stochastic network with two components. The first one is the assignment network aimed at assigning drivers to loads during the current time period. The second is the forecast network, which represents the remainder of the time horizon and includes all forecasted demands, as well as known loads to be picked up in the future. Arcs in this part of the network represent loaded and empty moves or waiting times from one period to the next. A set of links connecting the two parts of the network represent empty relocation moves in anticipation of future loads. The expected recourse function is represented by a cluster of "recourse links" that capture the expected marginal contribution of vehicles located in a given region during a given time period. An approximation of this function value is added to the arc costs, and the resulting transportation problem is then solved with a network simplex algorithm on a rolling horizon basis.

In Godfrey and Powell (2002) and Powell and Topaloglu (2003), a new approach is proposed for solving large-scale stochastic and dynamic problems, such as integer multicommodity flow problems. In these problems, new decisions are made continuously over time in response to new information. The approach is first presented within a general framework. Then, the authors report its application in different contexts, like vehicle dispatching. The proposed approach combines techniques from dynamic programming and multistage stochastic programming. More precisely, it uses the forward dynamic programming techniques of Bertsekas and Tsitsiklis (1996), which step forward in time to avoid an evaluation of all possible states. In Spivey and Powell (2004), the authors provide a mathematical model for a general class of dynamic

assignment problems that integrate an explicit representation of the exogenous information process. They propose adaptive dynamic programming algorithms for solving these problems, where sequences of assignment problems are iteratively solved, using linear approximations to evaluate future states.

The issue of relocating an empty vehicle in anticipation of future demands through exploitation of probabilistic knowledge is addressed in Larsen (2000) and Larsen, Madsen, and Solomon (2004). The problem studied is a dynamic TSP where the goal is to find a minimum-cost tour through a set of dynamic requests. The requests, each with a soft time window, unfold over time in a number of subregions according to a Poisson process with different arrival rates. Each subregion may have one or more resting locations (idle points) for the vehicles. Moreover, all requests must be serviced, even if some vehicles arrive late at the depot. The objective function is a weighted sum of total travel time and total lateness over all requests. Whenever a new service request occurs, the current tour is reoptimized using a simple heuristic based on an exchange procedure. Once the vehicle has finished serving its current customer, the algorithm determines the best resting location for the vehicle if the time window at its next destination does not allow immediate service (and assuming that the probability of having at least one request in the chosen location is high enough). This is achieved using the following online policies: (i) the vehicle goes to the nearest resting location; (ii) the vehicle goes to the busiest resting location (i.e., in the subregion with the largest arrival rate); (iii) the vehicle goes to the resting location that belongs to the subregion with the highest expected number of customers. Simulations on problems with about 40 customers were used to compare the different relocation policies, including a “no-rest” policy where the vehicle systematically waits at its current position.

### 3. Problem Definition

A static and deterministic version of our problem can be stated as follows. Let  $G = (\mathcal{V}, \mathcal{A})$  be a complete directed graph where  $\mathcal{V} = \mathcal{L} \cup \{0\}$  is the set of nodes, with  $\mathcal{L} = \{1, \dots, L\}$  the set of customers, 0 the depot, and  $\mathcal{A}$  the set of arcs  $(i, j)$ ,  $i, j \in \mathcal{V}$ ,  $i \neq j$ , with associated travel times  $t_{ij}$ . Each customer  $i \in \mathcal{L}$  has a time window  $\omega_i = [\underline{w}_i, \bar{w}_i]$  where  $\underline{w}_i$  is the earliest time and  $\bar{w}_i$  the latest time to start service. These time windows are “soft,” as a vehicle can arrive before  $\underline{w}_i$  or after  $\bar{w}_i$ . If the vehicle is too early, it must wait to start its service; if the vehicle is too late, a penalty for lateness is included into the objective. The time window at the depot is  $\omega_0 = [\underline{w}_0, \bar{w}_0] = [0, T]$ , where  $T$  is a hard deadline for the return of all vehicles to the depot.

Let  $\mathcal{K} = \{1, \dots, K\}$  be the set of identical, uncapacitated vehicles to be routed and scheduled to service the customers. The goal is to design a set of routes of minimum cost such that:

- each vehicle serves one route;
- each vehicle route originates from and terminates at the depot;
- each customer is served at most once by one vehicle;
- the feasibility of the time schedule is satisfied.

The solution cost typically relates to the number of unserved customers, total travel time, and lateness at customer locations. Mathematical programming formulations for this type of problem can be found, for example, in Toth and Vigo (2002).

In the dynamic version of the problem, which is the problem of interest here, new customer requests continuously unfold over time and must be dispatched in real time to vehicles. A dynamic programming formulation for this problem is proposed in the following (mathematical models for related dynamic and stochastic problems, like vehicle assignment and inventory routing, can be found in Adelman 2003, 2004; Kleywegt, Nori, and Savelsbergh 2004; Spivey and Powell 2004). Although intractable, the formulation is useful as it provides a formal setting for our study. The underlying system is clearly continuous, but to simplify the presentation we assume a discrete-time model where the time intervals are sufficiently small. We also assume the following:

- the distribution area is partitioned into a number of geographic zones;
- the time windows for the customers are grouped into a number of (possibly overlapping) time periods (e.g., all customers with a time window from 8:00 to 10:00 AM are in one group, all customers with a time window from 9:00 to 9:30 AM are in another group, etc.);
- a geographic zone plus a time period defines a customer type;
- request arrivals for each customer type are independent and follow a known distribution;
- a fixed number of vehicles is available and, at any given instant, they may be free or busy (traveling to or serving a customer);
- a vehicle is committed to a given customer when it departs from its current location to serve that customer. Accordingly,

—at any time before departure, the vehicle can be dispatched otherwise,

—after departure, the vehicle cannot be redirected elsewhere;

- a vehicle is not allowed to arrive at a given customer before the earliest time to start service. Instead, the vehicle must wait at its current location. As a vehicle is not committed to a customer until it departs to

serve it, this allows the vehicle to be dispatched otherwise, if the system changes in the mean time (due to new customer request arrivals).

The discrete model is now formally introduced in the following subsections.

### 3.1. Notation

- $\mathcal{K}$  set of vehicles indexed from 1 to  $K$ ;
- $\mathcal{K}_f^t$  set of free vehicles at instant  $t$ ;
- $\mathcal{K}_b^t$  set of busy vehicles at instant  $t$ ;
- $\mathcal{E}$  set of geographic locations or zones;
- $\mathcal{E}^+$  set of geographic locations, including the depot location;
- $\mathcal{W}$  set of periods for time windows, with  $\omega = (\underline{w}, \bar{w}) \in \mathcal{W}$ ;
- $\mathcal{L}$  set of customer types indexed from 1 to  $L$ , where type  $l \in \mathcal{L}$  corresponds to  $(\varepsilon_l, \omega_l) \in \mathcal{E} \times \mathcal{W}$ ;
- $\mathcal{L}^+$  set of customer types, including the depot;
- $c_{kl}^t$  cost of assigning a customer of type  $l \in \mathcal{L}^+$  to vehicle  $k \in \mathcal{K}$  at instant  $t$ ;
- $p_l$  penalty for unassigned customer of type  $l \in \mathcal{L}$ ;
- $T$  deadline at the depot; all vehicles should be back at the depot by that time.

In our case, the cost  $c_{kl}^t$ ,  $k \in \mathcal{K}$ ,  $l \in \mathcal{L}$  is the travel time from the current location of vehicle  $k$ , denoted  $\varepsilon_k^t$ , to a customer of type  $l$  plus any lateness at the customer. That is,

$$c_{kl}^t = t_{\varepsilon_k^t \varepsilon_l} + \max\{0, (t + t_{\varepsilon_k^t \varepsilon_l}) - \bar{w}_l\}, \quad l \in \mathcal{L}. \quad (1)$$

When the depot is assigned to a vehicle, the latter is simply sent back to the depot to end its workday. In this case, we have

$$c_{k0}^t = t_{\varepsilon_k^t \varepsilon_0}. \quad (2)$$

### 3.2. States

A state  $S^t$  at instant  $t$  is described through the following state vectors:

$$S^t = \{\varepsilon^t, \underline{s}^t, \underline{b}^t, q^t\}$$

where

$\varepsilon^t = (\varepsilon_k^t)_{k \in \mathcal{K}} \in \mathcal{E}^{+K}$  contains the current destination of each vehicle (or current position if free).

$\underline{s}^t = (s_k^t)_{k \in \mathcal{K}} \in [t, T]^K$  contains the time of end-of-service of each vehicle at the current destination. The dummy value  $-1$  is used when the vehicle is free.

$\underline{b}^t = (b_k^t)_{k \in \mathcal{K}} \in \{0, 1\}^K$  is a Boolean vector that indicates the status, committed (0) or free (1), of each vehicle.

$q^t = (q_l^t)_{l \in \mathcal{L}} \in N^L$  is the number of unassigned requests for each customer type.

### 3.3. Decisions

At every instant  $t$ , free vehicles are considered for commitment to unassigned requests, including the depot. When the depot is involved, the corresponding

vehicle is simply sent back to the depot to end its workday. Given a decision policy  $\pi$ , the decision vector at time  $t$  is noted  $X^{\pi t}$ , although it should really be  $X^{\pi t}(S^t)$  due to its dependency on state  $S^t$ ; this is implicitly assumed in the following. We thus have  $X^{\pi t} = (x_{kl}^{\pi t})_{k \in \mathcal{K}, l \in \mathcal{L}^+}$  with

$$\begin{aligned} x_{kl}^{\pi t} &= 1, && \text{if vehicle } k \text{ is committed to a} \\ &&& \text{request of type } l, \\ & & & \\ & & & = 0, \quad \text{otherwise.} \end{aligned}$$

For a decision vector  $X^{\pi t}$  to be feasible, the following constraints must hold:

$$\begin{aligned} x_{kl}^{\pi t} &= 0, && k \in \mathcal{K}_b^t, l \in \mathcal{L}^+, \\ x_{kl}^{\pi t} &= 0, && k \in \mathcal{K}_f^t, l \in \overline{\mathcal{L}^+}_{ok}^t, \\ \sum_{l \in \mathcal{L}^+} x_{kl}^{\pi t} &\leq 1, && k \in \mathcal{K}_f^t, \\ \sum_{k \in \mathcal{K}} x_{kl}^{\pi t} &\leq q_l^t, && l \in \mathcal{L}, \end{aligned}$$

where  $\mathcal{L}^+_{ok} (\overline{\mathcal{L}^+}_{ok}^t)$  is the set of customer types, including the depot, that are open (not open) for vehicle  $k$  at instant  $t$ . A customer type is open for a vehicle if there is at least one unassigned customer of that type and if the vehicle, by traveling immediately and directly from its current location to a customer of that type, reaches it at or after the earliest time to start service. A customer type is said to be closed otherwise. In the case of the depot, it opens for a vehicle if the latter, by traveling immediately and directly to the depot, reaches it at instant  $T$ . The first two constraints set decision variables associated with forbidden commitments to 0. The third constraint specifies that a free vehicle can be committed to at most one customer, including the depot. The last constraint specifies that the number of newly committed vehicles to customers of a particular type cannot exceed the number of unassigned requests of that type.

### 3.4. Transitions

A transition occurs when decision vector  $X^{\pi t}$  is applied to state  $S^t$  to obtain the postdecision state

$$S^{t+} = \{\varepsilon^{t+}, \underline{s}^{t+}, \underline{b}^{t+}, q^{t+}\},$$

where

$$\begin{aligned} \varepsilon_k^{t+} &= \varepsilon_l, && \text{if } k \in \mathcal{K}_f^t \text{ and } x_{kl}^{\pi t} = 1, l \in \mathcal{L}^+, \\ &&& \varepsilon_k^t \text{ otherwise,} \end{aligned}$$

$$\begin{aligned} s_k^{t+} &= \max\{\underline{w}_l, t + t_{\varepsilon_k^t \varepsilon_l}\}, && \text{if } k \in \mathcal{K}_f^t \text{ and } x_{kl}^{\pi t} = 1, l \in \mathcal{L}^+, \\ &&& s_k^t \text{ otherwise,} \end{aligned}$$

$$b_k^{t+} = 1 - \sum_{l \in \mathcal{L}^+} x_{kl}^{\pi t}, \quad \text{if } k \in \mathcal{K}_f^t; \quad 0 \quad \text{otherwise,}$$

$$q_l^{t+} = q_l^t - \sum_{k \in \mathcal{K}_f^t} x_{kl}^{\pi t}, \quad l \in \mathcal{L}.$$

Now, let  $o_l^{t+1}$ ,  $l \in \mathcal{L}$ , be the number of new request arrivals for each customer type in  $(t, t+1]$  and  $\mathcal{K}_{bf}^{t+1}$  be the subset of busy vehicles that become free at instant  $t+1$ . Then state  $S^{t+1}$  is updated at instant  $t+1$  to become state

$$S^{t+1} = \{\underline{\varepsilon}^{t+1}, \underline{s}^{t+1}, \underline{b}^{t+1}, \underline{q}^{t+1}\},$$

where

$$\begin{aligned} \varepsilon_k^{t+1} &= \varepsilon_k^t, \quad k \in \mathcal{K}, \\ s_k^{t+1} &= -1, \quad \text{if } k \in \mathcal{K}_{bf}^{t+1}; \quad s_k^t \quad \text{otherwise}, \\ b_k^{t+1} &= 1, \quad \text{if } k \in \mathcal{K}_{bf}^{t+1}; \quad b_k^t \quad \text{otherwise}, \\ q_l^{t+1} &= q_l^t + o_l^{t+1}, \quad l \in \mathcal{L}. \end{aligned}$$

Note that when  $\mathcal{K}_b^{t+1} = \mathcal{K}$ , it is possible to move directly from instant  $t$  to instant  $\min_{k \in \mathcal{K}} s_k^{t+1}$ , given that no commitments can be made when all vehicles are busy.

### 3.5. Objective

In the initial state  $S^0$ , all vehicles are located at the depot, and there might already be a number of unassigned customer requests. A feasible decision policy  $\pi$  leads from  $S^0$  to an admissible final state  $S^T$ , where all vehicles are back at the depot. The policy is then evaluated through the following expression:

$$\sum_{t=0}^{T-1} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}^+} c_{kl} x_{kl}^{\pi t} + \sum_{l \in \mathcal{L}} p_l q_l^T. \quad (3)$$

Given a family of decision policies  $\Pi$ , we are interested in

$$\min_{\pi \in \Pi} E \left[ \sum_{t=0}^{T-1} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}^+} c_{kl} x_{kl}^{\pi t} + \sum_{l \in \mathcal{L}} p_l q_l^T \right], \quad (4)$$

where  $E[X]$  denotes the expected value of random variable  $X$ . In practice, a policy is evaluated according to (3), on the basis of a particular realization (or an average over a number of realizations). This is what we have done with the policy introduced in §4.

In the following, we assume that the penalty  $p_l$  is set to the same large value for every customer type  $l \in \mathcal{L}$ . The goal is thus to maximize the number of served customers and, for the same number of served customers, to minimize travel time plus lateness.

### 3.6. Optimality Equations

Although intractable in practice, the optimality equations associated with our dynamic programming framework provide a way to compute an optimal policy. This is illustrated for a particular distribution of request arrivals, where each customer type  $l$  is associated with a distinct nonstationary Poisson process,

based on a continuous arrival rate function  $\lambda_l(t)$  (continuity is assumed here to facilitate the mathematical developments). Let  $\mathcal{O}$  be the set of all possible customer arrivals in  $(t, t+1]$ ; that is,

$$\mathcal{O} = \{\underline{o} = (o_1, \dots, o_L) \mid o_i = 0, 1, 2, \dots; i = 1, \dots, L\}.$$

For every customer type  $l$ , the probability of observing exactly  $o_l$  arrivals is given by

$$P(Y_l = o_l) = \frac{e^{-\int_{t+}^{t+1} \lambda_l(t') dt'} \left( \int_{t+}^{t+1} \lambda_l(t') dt' \right)^{o_l}}{o_l!}, \quad l \in \mathcal{L}.$$

Arrivals for all customer types being independent, the optimality equation associated with state  $S^t$ ,  $0 \leq t \leq T-1$ , is

$$\begin{aligned} f_t^*(S^t) = \min_{X^t \in \mathcal{X}^t(S^t)} & \left\{ \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}^+} c_{kl} x_{kl}^t + \sum_{\underline{o} \in \mathcal{O}} \left( \prod_{l=1}^L P(Y_l = o_l) \right) \right. \\ & \cdot f_{t+1}^*(S^{t+1}(S^t, X^t, \underline{o})) \Big\}, \end{aligned}$$

where  $\mathcal{X}^t(S^t)$  is the set of feasible decision vectors in state  $S^t$ ,  $f_t^*(S^t)$  is the optimal value from state  $S^t$ , and  $f_{t+1}^*(S^{t+1}(S^t, X^t, \underline{o}))$  is the optimal value from state  $S^{t+1}$ . Note that the latter depends on the decision  $X^t$  taken in state  $S^t$  and the particular occurrence  $\underline{o}$  of request arrivals in  $(t, t+1]$ .

For  $t = T$ , we have

$$\begin{aligned} f_T^*(S^T) &= \sum_{l \in \mathcal{L}} p_l q_l^T, \quad S^T \in \Psi^T, \\ f_T^*(S^T) &= \infty, \quad S^T \notin \Psi^T, \end{aligned}$$

where  $\Psi^T$  is the set of admissible final states at instant  $T$ , where all vehicles are back at the depot. Clearly, we are interested in  $f_0^*(S^0)$ , the optimal value from state  $S^0$ .

### 4. A Vehicle-Waiting Heuristic

The problem-solving approach described in the following solves the dynamic version of our problem approximately. In particular, unassigned customer requests are included into planned routes that are generated with a tabu search heuristic. These planned routes can be considered as an “algorithmic feature” used to (heuristically) decide about the next destination of each vehicle. The original contribution here is the inclusion of a threshold-based vehicle-waiting heuristic that takes into account future customer requests. The proposed approach exploits probabilistic knowledge about those future requests to better manage the fleet of vehicles and provide good coverage of the territory.

Here, a restricted case of the general formulation of §3 is considered. A customer type is defined by its

geographic location only, and each customer type is associated with a Poisson process, where the arrival rate is a step function of a number of time periods defined over the horizon. These choices come from practical considerations related to the development of the simulator, which dynamically generates new requests over time (see §6).

#### 4.1. Basic Assumptions

We assume that the distribution area is partitioned into  $J$  geographic zones  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_J$  and that the time horizon is divided into  $M$  time periods  $\omega_1, \omega_2, \dots, \omega_M$ . Furthermore, the probabilities that (at least) one request will occur in a zone  $\varepsilon_j$ ,  $j = 1, \dots, J$  during time period  $\omega_m$ ,  $m = 1, \dots, M$  can be calculated by assuming independent Poisson processes with arrival rates  $\lambda_{jm}$ ,  $j = 1, \dots, J$ ,  $m = 1, \dots, M$ . With this information, it might be better to require a vehicle to wait in its current zone instead of directing it towards its next planned destination (in another zone) if it appears that new customer requests are likely to unfold in the near future. In the following, the various elements of our vehicle-waiting strategy are discussed.

#### 4.2. Waiting Time

Actually, it can be useless for the vehicle to wait at its current location if the probability of occurrence of a new request in its neighborhood in the near future is not high enough. Moreover, while the vehicle is forced to wait, some “real” requests might suffer important service delays. To overcome this problem, a vehicle  $k$  is kept at its current location only for some amount of time  $\delta t_k$ . The value of  $\delta_k$  is important. If it is too small, there may not be enough time for forecasted requests to materialize. If it is too large, the vehicle may wait for a long time for nothing, while allowing service requests to build up in other areas.

In our implementation,  $\delta t_k = \min(\Delta, \bar{t}_k - t, (\bar{\bar{t}}_k - t) + \eta)$ . The first component  $\Delta$  is a fixed parameter that is simply aimed at avoiding too-large values. The second component is the difference between the latest feasible departure time  $\bar{t}_k$  (which allows vehicle  $k$  to return before the upper bound of the time window at the depot) and the current time  $t$ . The third component is the difference between the latest departure time that does not induce any additional lateness on the vehicle route  $\bar{\bar{t}}_k$  and the current time  $t$ , plus a given parameter  $\eta$ . This last parameter represents some tolerance to additional lateness.

#### 4.3. Probability of Occurrence of a New Request

In our strategy, a given vehicle  $k$  may be forced to wait at its current position for some amount of time  $\delta t_k$ . However, due to the probabilistic nature of the available information, the requests may never

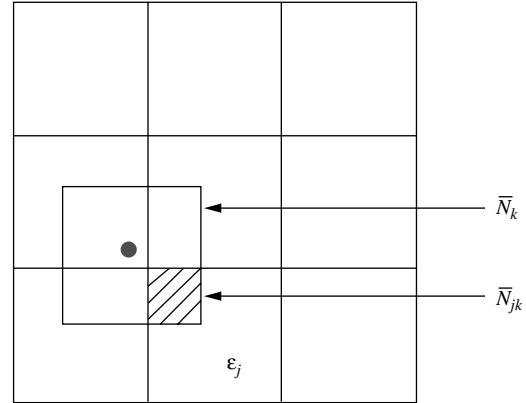


Figure 1 Vehicle Neighborhood

materialize. In this case, the vehicle will wait for nothing. Thus, waiting is allowed only if the probability for a request to occur in the vehicle neighborhood during time interval  $[t, t + \delta t_k]$  is greater than or equal to a given threshold. In the following, the probability calculation is explained.

Let us assume that vehicle  $k$  has just finished serving its current customer in some geographic zone at time  $t$ . Then, the neighborhood  $\bar{N}_k$  of vehicle  $k$  is defined as a square centered at the current vehicle location. In our experiments,  $\bar{N}_k$  has the same dimension as a geographic zone, although it does not need to be in general. This is illustrated in Figure 1 for a distribution area divided into nine zones. In this figure, the current vehicle location is represented by the large dot.

Let us also assume that vehicle  $k$  is ready to leave its current location at instant  $t$ . Then, the probability  $p^{\delta t_k}$  that at least one request will occur in time interval  $[t, t + \delta t_k]$  in the neighborhood  $\bar{N}_k$  is  $p^{\delta t_k} = 1 - p_0^{\delta t_k}$ , where  $p_0^{\delta t_k}$  is the probability that no new request occurs in time interval  $[t, t + \delta t_k]$  in  $\bar{N}_k$ . If we suppose that  $t$  belongs to time interval  $\omega_{m_1}$  and  $(t + \delta t_k)$  to time interval  $\omega_{m_2}$ ,  $1 \leq m_1 \leq m_2 \leq M$ , then the interval  $[t, t + \delta t_k]$  can be partitioned into  $(m_2 - m_1 + 1)$  intervals, each belonging to time period  $\omega_r$  with length  $\delta t_k^r$ ,  $r = m_1, \dots, m_2$ . This is illustrated in Figure 2, where the time horizon is partitioned into three time periods. In this figure,  $t$  belongs to time period  $\omega_1$  and  $(t + \delta t_k)$  belongs to time period  $\omega_3$ .

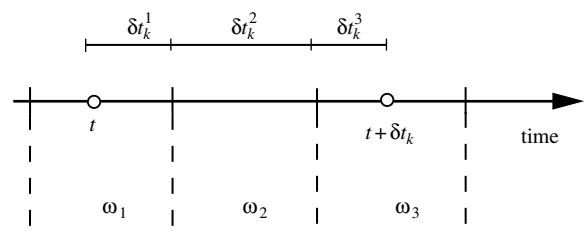


Figure 2 Partitioning  $\delta t_k$

1. if  $(d_k \leq \alpha\bar{d})$ , go to the next planned customer location (and exit).
2. if ((prob[at least one request in  $\bar{N}_k$  during  $\delta t_k$ ] > s) and ( $v_j \leq V$ )), wait for  $\delta t_k$  time units,  
 else go to the next planned customer location.

**Figure 3 Pseudocode of the Proposed Waiting Strategy**

Now we have  $p_0^{\delta t_k} = \prod_{r=m_1}^{m_2} p_0^{\delta t_k^r}$ , where  $p_0^{\delta t_k^r} = \exp(-\lambda_{\bar{N}_k^r} \delta t_k^r)$  is the probability that no new request occurs during  $\delta t_k^r$  time units in  $\bar{N}_k$ , and  $\lambda_{\bar{N}_k^r}$  is the arrival rate in  $\bar{N}_k$  during time period  $\omega_r$ . Because the  $J$  Poisson processes associated with the geographic zones are assumed to be independent,  $\lambda_{\bar{N}_k^r} = \sum_{j=1}^J \lambda_{jr}(S_{\bar{N}_{jk}}/S)$ , where  $\bar{N}_{jk} = \varepsilon_j \cap \bar{N}_k$ ,  $S_{\bar{N}_{jk}}$  is the area of  $\bar{N}_{jk}$ , and  $S$  is the area of a zone.

#### 4.4. Coverage of the Territory

While applying the waiting heuristic, vehicles will tend to cluster in a few zones with large request arrival rates. Thereby, new requests that appear in other zones might incur excessive service delays (or no service at all). To prevent too many vehicles from clustering in the same zone, an upper bound on the number of vehicles is associated with each zone.

#### 4.5. Summary

Figure 3 provides a pseudocode description of the proposed approach for some vehicle  $k$  that has just completed its service in some zone  $\varepsilon_j$  and is now ready to leave for its next destination.

In the pseudocode,  $\bar{d}$  is the average distance between two consecutive customer locations and  $d_k$  is the distance between the current location of vehicle  $k$  and its next destination. Also,  $v_j$  is the number of vehicles in the current zone  $j$  and  $V$  is an upper bound on the number of vehicles in each zone. Finally,  $\alpha$  and  $s$  are user-defined parameters. Thus, a vehicle waits at its current location if its next destination is far enough, the probability of a request arrival in the vehicle's neighborhood in the next  $\delta t_k$  time units is high enough, and there are not too many vehicles in the current zone.

The effectiveness of this strategy is assessed using a parallel tabu search heuristic, previously reported in Gendreau et al. (1999). The next section will briefly describe the original algorithm and explain how it was modified to integrate the proposed approach.

### 5. A Parallel Tabu Search Algorithm

#### 5.1. The Original Algorithm

In a dynamic context, a solution at any instant  $t$  is a set of planned routes, each beginning with the current destination of the associated vehicle. This current destination is fixed and cannot be modified (i.e., the vehicle must reach it). The algorithm proposed in

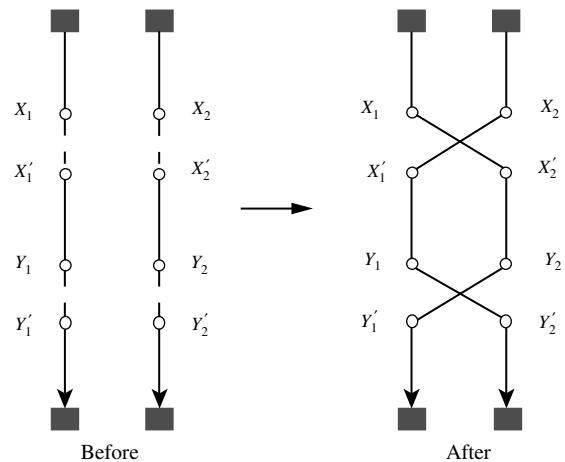
Gendreau et al. (1999) is a parallel tabu search heuristic with an adaptive memory (Rochat and Taillard 1995). The tabu search is an iterative local search technique where a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution, even if a degradation is observed. By allowing such degradation, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (1997). The algorithm used in Gendreau et al. (1999) can be summarized as follows.

- Construct  $H$  different initial solutions with a stochastic insertion heuristic (where the choice of the next customer to be inserted is randomized).
- Apply tabu search to each solution and store the resulting routes in the adaptive memory.
- While a stopping criterion is not met, do:
  - Combine routes stored in the adaptive memory to create an initial solution.
  - Decompose the problem into subproblems through a geographic decomposition procedure.
  - Apply tabu search to each subproblem.
  - Add the routes of the resulting solution in the adaptive memory.

The procedure for generating the neighborhood of the current solution is based on CROSS exchanges (Taillard et al. 1997). Basically, two segments of variable lengths are taken from two different routes and are swapped, as illustrated in Figure 4.

To speed up the algorithm, a parallel implementation on a network of workstations was developed. The parallelization of the procedure was achieved at two levels:

- Different tabu search threads run in parallel, each starting from a different initial solution.
- Within each search thread, many tabu searches run independently on the subproblems obtained through the decomposition procedure.



**Figure 4 A CROSS Exchange**

Whenever a new event occurs, the tabu search threads are interrupted. Then, after an appropriate update of the adaptive memory to reflect the current state of the system, the search is restarted with new solutions obtained from the updated memory. Thus, the tabu search improves the current planned routes between the occurrence of new events. An event may be either the occurrence of a new request, in which case the request must first be inserted in the current solution, or the completion of service at a given customer (as a new destination must be indicated to the driver).

During schedule generation, a vehicle never departs from its current location if, by doing so, a waiting time at its next destination is incurred. Instead, the vehicle waits at its current location to exploit potential opportunities offered by new request arrivals. In a sense, this implementation is not totally myopic, given that this waiting strategy can be viewed as an implicit way of accounting for future arrivals. Other ways to implicitly account for future arrivals through the definition of different objectives or waiting strategies can be found in Mitrović-Minić and Laporte (2004) and Mitrović-Minić, Krishnamurti, and Laporte (2004).

With regard to the objective, the number of served customers is maximized as the algorithm always inserts a customer when it is feasible to do so. For a given number of customers, the sum of total travel time and total lateness over all customers is minimized.

## 5.2. Implementing the Vehicle-Waiting Heuristic

The waiting approach described in §4.5 basically modifies the way new events are handled. This is explained in the following.

(a) When vehicle  $k$  has finished serving a customer, the new strategy is used to decide whether or not to wait. In the latter case, the vehicle is directed toward its next destination, as in the original algorithm. In the former case, a dummy customer  $D_{\delta t_k}$  is created at the current vehicle location. The intent is to force the vehicle to wait at its current location during time interval  $[t, t + \delta t_k]$ , where  $t$  is the current time and  $\delta t_k$  is calculated as in §4.2, based on the best solution in adaptive memory. Then, the dummy customer  $D_{\delta t_k}$  is inserted at the beginning of the planned route of vehicle  $k$  in every solution in adaptive memory, and solutions that become infeasible are discarded (due to the hard time window at the depot, it may not be possible to wait for  $\delta t_k$  time units at the current vehicle location). Finally, the search threads are restarted with new solutions constructed from the updated memory and the optimization is performed with the assumption that vehicle  $k$  restarts at time  $t + \delta t_k$ .

(b) When a new request occurs, we first see if it is included in the neighborhood of some waiting vehicles. Among all such vehicles, only those that can start service as soon as they reach the new request are kept. Then, among the remaining vehicles, we choose the one for which the inclusion of the new request leads to the smallest additional cost. The new request then replaces the dummy customer associated with the chosen vehicle route. If no vehicle qualifies for the new request, the latter is handled as in the original algorithm. Note that the new strategy offers additional insertion points for a new incoming request. These are located between the current vehicle location and its next destination for each waiting vehicle.

If vehicle  $k$  was not used to service any incoming request when time  $(t + \delta t_k)$  is reached (either because no new request occurred in its neighborhood or because it was never chosen), the dummy customer  $D_{\delta t_k}$  is removed from its route and the vehicle is sent to its next planned destination, as indicated by the best solution in adaptive memory.

## 6. Computational Results

In this section, we empirically demonstrate the value of the vehicle-waiting heuristic, when integrated within our previous tabu search implementation. To this end, we first briefly describe the simulator used to generate problem instances, before reporting the results obtained under different operating scenarios.

### 6.1. Simulation Framework

A simulator was developed to produce different operating scenarios that reflect as closely as possible what is observed in the real world. In the following, we describe the components of this simulator.

**6.1.1. Time Horizon.** The simulator first sets up a time horizon that represents the entire labor day. For example, a labor day running from 10h00 to 15h00 is represented by a time horizon of 300 minutes. Within this horizon, the simulator produces discrete time events that correspond to either a request arrival or the end of service at a given customer location. The new request arrivals take place only in the first 240 minutes of the time horizon, to allow vehicles to service their planned routes before returning to the depot.

**6.1.2. Generation of New Service Requests.** Typically, the arrival rate of new requests varies over time (e.g., peak hours) and space (e.g., densely populated areas). Hence, time-space Poisson processes are used. We recall that the distribution area is partitioned into  $J$  zones  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_J$  and that the time horizon is divided into  $M$  time intervals  $\omega_1, \omega_2, \dots, \omega_M$ . Also,  $\lambda_{jm}$  denotes the request arrival rate in zone  $\varepsilon_j$  during

time period  $\omega_m$ ,  $j = 1, \dots, J$ ;  $m = 1, \dots, M$ . The characteristics of a new service request are generated as follows. Given the current time interval  $\omega_m$ , the arrival rate  $\lambda_m = \sum_{j=1}^J \lambda_{jm}$  in the whole distribution area over time interval  $\omega_m$  is used to determine the arrival time of the next request. Then the zone  $\varepsilon_j$ , where the new request will appear, is determined using the following cumulative probabilities  $(P_j^m)_{0 \leq j \leq J}$ :

$$P_j^m = \begin{cases} 0, & \text{if } j = 0, \\ \sum_{l=0}^j p_{lm}, & \text{otherwise,} \end{cases}$$

where  $p_{lm} = \lambda_{lm} / \sum_{j=1}^J \lambda_{jm}$ .

The selected zone  $\varepsilon_j$  is such that  $P_{j-1}^m \leq u < P_j^m$ , where  $u$  is drawn from a uniform distribution  $U[0, 1]$ . To generate the time window associated with a new request  $i$ , the earliest time to start service  $\underline{w}_i$  is first generated in the interval  $[t, t + \Delta T_1]$ , where  $t$  is the current time and  $\Delta T_1$  is a constant. Then  $\underline{w}_i$  is possibly adjusted to avoid exceeding  $\bar{t}_i = T - t_{i0}$ , where  $T$  is the deadline at the depot (i.e., the end of the day) and  $t_{i0}$  is the travel time from  $i$  to the depot. Clearly, it is not possible to return to the depot on time if the service at request  $i$  starts after  $\bar{t}_i$ . Once the earliest time  $\underline{w}_i$  is determined, the latest time  $\bar{w}_i$  is set to  $\underline{w}_i + \Delta T_2$ , where  $\Delta T_2$  is a constant. The value  $\bar{w}_i$  is adjusted like  $\underline{w}_i$  if necessary.

## 6.2. Testing Scenarios

Different sets of problems were generated to perform numerical tests. In these problems, the fleet size is set to four or six vehicles with all vehicles moving at a constant average speed of 30 kms/hour. Parameters  $\Delta T_1$  and  $\Delta T_2$  are both set to 30 minutes. The time horizon is set to 300 minutes (i.e., five hours) and is divided into three time periods: morning ( $\omega_1$ ), lunch time ( $\omega_2$ ), and afternoon ( $\omega_3$ ). The lunch-time period is one hour long and the two others are two hours long. The distribution area is a 5 kms  $\times$  5 kms square area and the depot is located at (3.3 kms, 4.5 kms). Nine “active” zones are considered in the distribution area and are divided into two categories: Category 1 contains the intense central zone where the request arrival rate is high, while Category 2 contains the remaining peripheral zones where the arrival rate is low. Moreover, in each problem the set of requests is divided into two subsets. The first subset contains requests that are known at the beginning of the day (i.e., 25% of the total number of requests). The second subset contains requests that unfold in real time. In this case, the arrival time of each request is calculated using a  $3 \times 2$  time-dependent arrival rate matrix  $(\lambda_{c\omega})_{1 \leq c \leq 2, 1 \leq \omega \leq 3}$ , where  $\lambda_{c\omega}$  is the arrival rate in a zone of category  $c$  during time period  $\omega$ . The entries

**Table 1** Arrival Rate Matrix in Scenario 1

	$\omega_1$	$\omega_2$	$\omega_3$
Intense zone	16.50	0.000	33.00
Peripheral zones	2.04	2.04	2.04

are adjusted to create two different types of scenarios: Scenario 1 where the total number of requests is about 180 (27 requests per hour, on average), and Scenario 2 where the total number of requests is about 240 (36 requests per hour, on average).

The arrival rate matrices for the two scenarios, in number of requests per hour, are reported in Tables 1 and 2, respectively. It is worth noting that in real life, new requests rarely unfold in the intense zone (associated with downtown) during lunch time. Therefore, the corresponding rate is set to 0 for the two scenarios during that time period.

## 6.3. Experiments

The experiments reported in this section were performed on a network of three SUN UltraSparc-III workstations (300 MHz). Each process is programmed in C++, and communication between the processes is handled by the Parallel Virtual Machine Software. The parameter values in the original tabu search algorithm are kept as in Gendreau et al. (1999). Parameters related to our vehicle-waiting heuristic are examined in the following.

**6.3.1. Preliminary Tests.** To determine the best values for parameters  $\alpha$ ,  $s$ ,  $\eta$ ,  $V$ , and  $\Delta$  (see Figure 3), tests were performed with four vehicles over a sample of four problems. Several values had to be tested for every problem in the sample. Therefore, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a correct calibration. Because Scenarios 1 and 2 are different, separate tests were conducted for both of them. Tables 3 and 4 present a sample of results obtained with different combinations of values using Scenario 1 and Scenario 2, respectively. We show only the best combinations of values in these tables (i.e., fewer than 5 unserved customers on average for Scenario 1 and fewer than 23 unserved customers on average for Scenario 2). In these tests, parameter  $V$  was set to 1 for the peripheral zones, while different values were tested for the intense central zone. The four numbers

**Table 2** Arrival Rate Matrix in Scenario 2

	$\omega_1$	$\omega_2$	$\omega_3$
Intense zone	19.98	0.000	39.96
Peripheral zones	2.52	4.98	2.52

**Table 3 Best Parameter Values with Four Vehicles for Scenario 1**

$\alpha_1 = 0.20$	$\alpha_1 = 0.20$	$\alpha_1 = 0.20$	$\alpha_1 = 0.20$
$s_1 = 0.60$	$s_1 = 0.80$	$s_1 = 0.80$	$s_1 = 0.80$
$\eta_1 = 200$	$\eta_1 = 100$	$\eta_1 = 200$	$\eta_1 = 300$
$V_1 = 1$	$V_1 = 1$	$V_1 = 1$	$V_1 = 1$
$\Delta_1 = 300$	$\Delta_1 = 180$	$\Delta_1 = 300$	$\Delta_1 = 600$
4.00 <sup>a</sup>	4.66	<b>4.00</b>	4.66
227.813 <sup>b</sup>	224.70	<b>229.550</b>	223.936
234.413 <sup>c</sup>	244.67	<b>179.980</b>	207.950
462.226 <sup>d</sup>	469.46	<b>409.530</b>	431.890
$\alpha_1 = 0.20$	$\alpha_1 = 0.20$	$\alpha_1 = 0.30$	$\alpha_1 = 0.40$
$s_1 = 0.90$	$s_1 = 0.90$	$s_1 = 0.90$	$s_1 = 0.80$
$\eta_1 = 200$	$\eta_1 = 300$	$\eta_1 = 200$	$\eta_1 = 100$
$V_1 = 1$	$V_1 = 2$	$V_1 = 2$	$V_1 = 1$
$\Delta_1 = 300$	$\Delta_1 = 600$	$\Delta_1 = 300$	$\Delta_1 = 180$
4.00	4.66	4.33	4.66
225.810	225.353	229.623	235.08
223.780	271.556	244.913	248.92
449.590	496.909	474.536	484.01
$\alpha_1 = 0.50$	$\alpha_1 = 0.50$	$\alpha_1 = 1.00$	$\alpha_1 = 1.00$
$s_1 = 0.60$	$s_1 = 0.90$	$s_1 = 0.80$	$s_1 = 0.80$
$\eta_1 = 200$	$\eta_1 = 200$	$\eta_1 = 200$	$\eta_1 = 200$
$V_1 = 1$	$V_1 = 1$	$V_1 = 1$	$V_1 = 2$
$\Delta_1 = 300$	$\Delta_1 = 300$	$\Delta_1 = 300$	$\Delta_1 = 300$
4.33	4.66	4.66	4.33
229.588	226.335	230.051	223.710
204.350	209.080	173.428	223.000
433.940	435.415	403.480	446.710
			440.310

<sup>a</sup>Nb. of unserved customers.

<sup>b</sup>Travel time.

<sup>c</sup>Lateness.

<sup>d</sup>Sum of travel time and lateness.

**Table 4 Best Parameter Values with Four Vehicles for Scenario 2**

$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$	$\alpha_2 = 0.20$
$s_2 = 0.65$	$s_2 = 0.65$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$
$\eta_2 = 60$	$\eta_2 = 200$	$\eta_2 = 100$	$\eta_2 = 180$	$\eta_2 = 200$
$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 2$
$\Delta_2 = 100$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$
21.33 <sup>a</sup>	22.66	22.66	22	22.66
210.30 <sup>b</sup>	211.48	217.03	214.01	213.08
2423.00 <sup>c</sup>	2385.84	2357.01	2344.13	2420.80
2633.20 <sup>d</sup>	2597.34	2574.10	2558.02	2633.89
$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$
$s_2 = 0.60$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$
$\eta_2 = 60$	$\eta_2 = 60$	$\eta_2 = 60$	$\eta_2 = 100$	$\eta_2 = 100$
$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$	$V_2 = 1$
$\Delta_2 = 100$	$\Delta_2 = 100$	$\Delta_2 = 180$	$\Delta_2 = 180$	$\Delta_2 = 300$
21.66	21.66	21.66	<b>21</b>	22
217.00	212.20	216.72	<b>217.10</b>	214.64
2345.50	2346.00	2489.40	<b>2345.90</b>	2371.58
2562.40	2558.14	2706.01	<b>2563.13</b>	2586.22
$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$	$\alpha_2 = 0.40$
$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.80$	$s_2 = 0.85$	$s_2 = 0.90$
$\eta_2 = 180$	$\eta_2 = 200$	$\eta_2 = 200$	$\eta_2 = 60$	$\eta_2 = 100$
$V_2 = 1$	$V_2 = 1$	$V_2 = 2$	$V_2 = 1$	$V_2 = 1$
$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 300$	$\Delta_2 = 100$	$\Delta_2 = 180$
22.66	22.66	22.66	21.33	22
217.30	215.42	215.24	216.50	213.30
2429.01	2424.14	2397.37	2357.50	2573.00
2646.10	2639.57	2612.62	2574.05	2786.30

<sup>a</sup>Nb. of unserved customers.

<sup>b</sup>Travel time.

<sup>c</sup>Lateness.

<sup>d</sup>Sum of travel time and lateness.

in each entry in Tables 3 and 4 are the average number of unserved customers, travel time, lateness, and sum of travel time and lateness, respectively. Based on the results obtained, the values  $\alpha_1 = 0.20$ ,  $s_1 = 0.80$ ,  $\eta_1 = 200$  sec.,  $\Delta_1 = 300$  sec., and  $V_1 = 1$  for Scenario 1, and  $\alpha_2 = 0.40$ ,  $s_2 = 0.80$ ,  $\eta_2 = 100$  sec.,  $\Delta_2 = 180$  sec., and  $V_2 = 1$  for Scenario 2 were finally kept for further investigations (they are shown in boldface in the tables). In these experiments, we also found that  $\delta t_k$  is about 4.50 minutes for Scenario 1 and 2.90 minutes for Scenario 2.

**6.3.2. Numerical Results.** We performed 10 different simulation runs with each scenario, using a fleet of four vehicles. Then we repeated the experiments, but with a fleet of six vehicles. The results are reported below.

**Scenario 1.** Tables 5 and 7 report the results of each simulation run with the original algorithm (Gendreau et al. 1999), using four and six vehicles, respectively; Tables 6 and 8 report the same results, but with the new algorithm. These tables show that the new algorithm performs well against the original one. More precisely:

*Experiments with four vehicles.* The new algorithm leads to an improvement in 8 out of 10 runs. In the two other runs, the results obtained with the new algorithm are very close to those obtained with the original algorithm. Note that among the eight runs where an improvement is observed, the original algorithm succeeds in serving all customers in five cases.

**Table 5 Original Algorithm Under Scenario 1: Simulation of Five Hours with Four Vehicles and 27 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	152	12	243.24	167.67	410.91
2	140	1	244.58	229.06	473.65
3	127	0	200.84	143.21	344.05
4	144	0	240.81	281.25	522.06
5	132	0	230.15	119.24	349.39
6	120	0	207.83	221.58	429.41
7	126	0	233.27	57.97	291.24
8	155	3	243.96	801.50	1045.47
9	153	3	243.99	436.17	680.16
10	132	0	232.95	89.88	322.84
Average	138.1	1.9	232.16	254.75	486.92

**Table 6 New Algorithm Under Scenario 1: Simulation of Five Hours with Four Vehicles and 27 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	152	11	233.59	196.43	430.09
2	140	1	243.74	230.13	473.87
3	127	0	202.67	137.67	340.35
4	144	0	227.30	247.09	474.40
5	132	0	219.55	120.24	339.69
6	120	0	201.78	231.42	433.20
7	126	0	224.81	52.09	276.91
8	155	2	239.43	753.19	992.62
9	153	1	240.78	459.54	700.20
10	132	0	238.67	44.49	283.17
Average	138.1	1.5	227.23	247.93	474.48

In these cases, the new algorithm reduces the sum of total travel time and lateness by 1.1% to 12.2%. In the three other cases, the number of unserved customers is reduced and, in two cases, this reduction is achieved at the cost of a marginal increase in the sum of total travel time and lateness. Overall, the latter is improved by 4.1% on average.

*Experiments with six vehicles.* Because these problems are easier to solve, the original algorithm succeeds in serving all customers in 9 runs out of 10; the new one succeeds in all runs. The results obtained with the original algorithm are slightly better than those obtained with the new algorithm in two cases only. Overall, the new algorithm improves the sum of total travel time and lateness in seven cases and the number of served customers in one case. The sum of total travel time and lateness is reduced by 2.3% on average over the 10 runs.

**Scenario 2.** These instances are harder than those of Scenario 1 due to higher arrival rates. Tables 9 and 11 report results obtained with the original algorithm, with four and six vehicles, respectively, while

**Table 8 New Algorithm Under Scenario 1: Simulation of Five Hours with Six Vehicles and 27 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	152	0	240.24	54.57	294.82
2	140	0	236.13	83.99	320.12
3	127	0	219.57	67.48	287.06
4	144	0	224.68	91.90	316.58
5	132	0	208.29	40.21	248.51
6	120	0	198.49	103.53	302.02
7	126	0	199.25	28.88	228.14
8	155	0	243.26	42.44	285.71
9	153	0	223.01	48.26	271.30
10	132	0	220.61	24.84	245.45
Average	138.1	0	221.35	58.61	279.97

Tables 10 and 12 report the results obtained with the new algorithm. As expected, the new approach clearly improves the results obtained with the original algorithm.

*Experiments with four vehicles.* In three cases, the two algorithms lead to the same number of unserved customers, but the new algorithm improves the sum of total travel time and lateness. Conversely, in two cases, an increase in the sum of total travel time and lateness is observed, but this is rather marginal when compared with the improvement observed in the number of unserved customers (cf. 0.2% versus 3.0% and 0.4% versus 6.3%, respectively). In the five remaining cases, the new algorithm improves the original algorithm on both accounts. The reduction ranges from 3.4% to 29.0% for the number of unserved customers, and from 2.0% to 16.0% for the sum of total travel time and lateness. Over the 10 runs, the average improvement is 9.0% and 4.2%, respectively.

*Experiments with six vehicles.* Here also, the new algorithm leads to larger improvements when compared to Scenario 1. While the two algorithms both succeed in serving all customers, the new algorithm

**Table 7 Original Algorithm Under Scenario 1: Simulation of Five Hours with Six Vehicles and 27 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	152	3	248.11	25.84	273.98
2	140	0	232.64	85.79	318.43
3	127	0	218.84	90.70	309.50
4	144	0	233.54	93.37	327.91
5	132	0	226.65	29.47	256.13
6	120	0	198.02	107.32	305.34
7	126	0	217.74	29.93	247.67
8	155	0	255.32	52.53	307.85
9	153	0	235.88	40.82	276.71
10	132	0	221.48	21.69	243.18
Average	138.1	0.3	228.83	57.85	286.60

**Table 9 Original Algorithm Under Scenario 2: Simulation of Five Hours with Four Vehicles and 36 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	168	7	236.80	1258.40	1495.13
2	196	25	231.80	3077.60	3309.30
3	197	36	192.25	3249.91	3442.20
4	177	17	231.50	2158.10	2389.53
5	182	21	207.00	2732.53	2939.50
6	198	32	214.10	3395.54	3609.62
7	195	38	180.34	3284.86	3465.20
8	164	7	223.92	1391.63	1615.55
9	187	29	188.70	3275.00	3464.00
10	193	29	209.65	2374.30	2584.00
Average	187.00	24.10	211.60	2620.00	2831.40

**Table 10 New Algorithm Under Scenario 2: Simulation of Five Hours with Four Vehicles and 36 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	168	7	241.40	1201.76	1443.24
2	196	21	218.72	2578.63	2797.35
3	197	35	191.40	3257.40	3448.82
4	177	14	206.70	2119.40	2326.11
5	182	21	203.01	2518.18	2721.20
6	198	30	204.55	3419.40	3623.96
7	195	38	181.90	3178.28	3360.19
8	164	6	232.00	1277.83	1509.85
9	187	27	192.85	3206.80	3399.66
10	193	28	205.63	2286.77	2492.41
Average	187.00	22.07	207.82	2504.44	2712.28

reduces the sum of total travel time and lateness in 9 runs out of 10. The reduction ranges from 2.0% to 11.0%, with an average of 3.0%.

In conclusion, the experiments show that the proposed vehicle-waiting heuristic is effective, even if it is incorporated into a powerful algorithm, and even if it is enforced only a few times over the scheduling horizon, namely, one to four times per run for Scenario 1, and one to six times per run for Scenario 2. This strategy is also more effective when it is applied on harder problems (i.e., smaller fleet size or higher request arrival rates).

## 7. Conclusion

In this paper, a new strategy that exploits probabilistic knowledge about future request arrivals to better manage a fleet of vehicles is proposed. Some issues related to this strategy were addressed and investigated. Then, experiments were performed to assess

**Table 11 Original Algorithm Under Scenario 2: Simulation of Five Hours with Six Vehicles and 36 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	168	0	251.00	283.55	534.55
2	196	0	301.10	577.42	878.50
3	197	0	274.93	1194.33	1469.30
4	177	0	262.10	567.40	829.43
5	182	0	270.54	440.02	710.60
6	198	0	265.32	560.43	825.80
7	195	0	295.90	679.10	974.96
8	164	0	256.40	364.43	621.00
9	187	0	273.80	665.00	940.00
10	193	0	246.30	557.01	803.30
Average	187.00	0	270.00	589.01	859.04

**Table 12 New Algorithm Under Scenario 2: Simulation of Five Hours with Six Vehicles and 36 Requests per Hour on Average**

Instance number	Number of customers	Number of unserved customers	Travel time	Lateness	Sum
1	168	0	265.00	259.21	524.02
2	196	0	280.78	570.55	851.33
3	197	0	269.19	1377.41	1646.61
4	177	0	253.80	519.20	773.01
5	182	0	282.27	371.80	654.07
6	198	0	274.25	521.44	795.70
7	195	0	294.88	650.37	945.25
8	164	0	239.67	345.90	585.57
9	187	0	265.76	601.46	867.23
10	193	0	245.12	470.71	715.84
Average	187.00	0	267.07	568.80	835.86

its effectiveness within a previously reported tabu search heuristic. The results show that the proposed approach provides significant improvements over the original algorithm, especially in some “critical” situations involving a small fleet size and high request arrival rates.

## Acknowledgments

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) and by the Québec Fonds pour la Formation de Chercheurs et l’Aide à la Recherche. This support is gratefully acknowledged.

## References

- Adelman, D. 2003. Price-directed replenishment of subsets: Methodology and its application to inventory routing. *Manufacturing Service Oper. Management* 5 348–371.
- Adelman, D. 2004. A price-directed approach to stochastic inventory/routing. *Oper. Res.* 52 499–514.
- Bertsekas, D., J. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.
- Bertsimas, D. J. 1988. Probabilistic combinatorial optimization problems. Ph.D. dissertation, Report No. 193, Operations Research Center, Massachusetts Institute of Technology, Boston, MA.
- Bertsimas, D. J. 1992. A vehicle routing problem with stochastic demand. *Oper. Res.* 40 574–585.
- Bertsimas, D. J., L. H. Howell. 1993. Further results on the probabilistic traveling salesman problem. *Eur. J. Oper. Res.* 65 68–95.
- Bertsimas, D. J., D. Simchi-Levi. 1996. A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. *Oper. Res.* 44 286–304.
- Bertsimas, D., G. van Ryzin. 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Oper. Res.* 39 601–615.
- Bertsimas, D. J., G. van Ryzin. 1993. Stochastic and dynamic vehicle routing in the Euclidian plane with multiple capacitated vehicles. *Oper. Res.* 41 60–76.
- Bertsimas, D. J., P. Jaillet, A. R. Odoni. 1990. A priori optimization. *Oper. Res.* 38 1019–1033.
- Dror, M., G. Laporte, P. Trudeau. 1989. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Sci.* 23 166–176.

- Gendreau, M., G. Laporte, R. Séguin. 1995. An exact algorithm for the vehicle routing problem with stochastic customers and demands. *Transportation Sci.* **29** 143–155.
- Gendreau, M., G. Laporte, R. Séguin. 1996a. Stochastic vehicle routing. *Eur. J. Oper. Res.* **88** 3–12.
- Gendreau, M., G. Laporte, R. Séguin. 1996b. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Oper. Res.* **44** 469–477.
- Gendreau, M., F. Guertin, J.-Y. Potvin, É. D. Taillard. 1999. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Sci.* **33** 381–390.
- Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer, Boston, MA.
- Godfrey, G., W. B. Powell. 2002. An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times. *Transportation Sci.* **36** 21–39.
- Jaillet, P. 1985. Probabilistic traveling salesman problems. Ph.D. dissertation, Massachusetts Institute of Technology, Boston, MA.
- Jaillet, P. 1988. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Oper. Res.* **36** 929–936.
- Jezequel, A. 1985. Probabilistic vehicle routing problems. Master's thesis, Massachusetts Institute of Technology, Boston, MA.
- Kleywegt, A. J. 1996. Dynamic and stochastic models with freight distribution applications. Ph.D. dissertation, Massachusetts Institute of Technology, Boston, MA.
- Kleywegt, A. J., J. D. Papastavrou. 1998. The dynamic and stochastic knapsack problem. *Oper. Res.* **46** 17–35.
- Kleywegt, A. J., J. D. Papastavrou. 2001. The dynamic and stochastic knapsack problem with random sized items. *Oper. Res.* **49** 26–41.
- Kleywegt, A. J., V. S. Nori, M. W. P. Savelsbergh. 2004. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Sci.* **38** 42–70.
- Laporte, G., F. V. Louveaux, H. Mercure. 1994. A priori optimization of the probabilistic traveling salesman problem. *Oper. Res.* **42** 543–549.
- Larsen, A. 2000. The dynamic vehicle routing problem. Ph.D. dissertation, Technical University of Denmark, Lyngby, Denmark.
- Larsen, A., O. Madsen, M. M. Solomon. 2002. Partially dynamic vehicle routing—Models and algorithms. *J. Oper. Res. Soc.* **53** 637–646.
- Larsen, A., O. Madsen, M. M. Solomon. 2004. The a priori dynamic traveling salesman problem with time windows. *Transportation Sci.* **38** 459–472.
- Mitrović-Minić, S., G. Laporte. 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Res. Part B* **38** 635–655.
- Mitrović-Minić, S., R. Krishnamurti, G. Laporte. 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Res. Part B* **38** 669–685.
- Papastavrou, J. D., S. Rajagopalan, A. J. Kleywegt. 1996. The dynamic and stochastic knapsack problem with deadlines. *Management Sci.* **42** 1706–1718.
- Powell, W. B. 1988. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. B. L. Golden, A. A. Assad, eds. *Vehicle Routing: Methods and Studies*. North Holland, Amsterdam, The Netherlands, 249–291.
- Powell, W. B. 1996. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Sci.* **30** 195–219.
- Powell, W. B., H. Topaloglu. 2003. Stochastic programming in transportation and logistics. *Stochastic Programming, Handbook in Operations and Management Sciences*, Vol. 10. Elsevier, Amsterdam, The Netherlands, 555–635.
- Powell, W. B., P. Jaillet, A. Odoni. 1995. Stochastic and dynamic networks and routing. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. *Network Routing, Handbooks in Operations Research and Management Science*, Vol. 8. North-Holland, Amsterdam, The Netherlands, 141–295.
- Rochat, Y., É. D. Taillard. 1995. Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics* **1** 147–167.
- Rubio Maqueo, R. 1995. Dynamic-stochastic vehicle routing and inventory problem. Ph.D. dissertation, Massachusetts Institute of Technology, Boston, MA.
- Secomandi, N. 1998. Exact and heuristic dynamic programming algorithms for the vehicle routing problem with stochastic demands. Ph.D. dissertation, College of Business Administration, University of Houston, Houston, TX.
- Secomandi, N. 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Comput. Oper. Res.* **27** 1201–1225.
- Séguin, R. 1994. Problèmes stochastiques de tournées de véhicules. Ph.D. dissertation, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, Québec.
- Spivey, M., W. B. Powell. 2004. The dynamic assignment problem. *Transportation Sci.* **38** 399–419.
- Sutton, R. S., A. G. Barto. 1998. *Reinforcement Learning*. MIT Press, Cambridge, MA.
- Swihart, M. R., J. D. Papastavrou. 1999. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *Eur. J. Oper. Res.* **114** 447–464.
- Taillard, É. D., P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin. 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Sci.* **31** 170–186.
- Thomas, B. W., C. C. White. 2004. Anticipatory route selection. *Transportation Sci.* **38** 473–487.
- Toth, P., D. Vigo, eds. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA.
- Waters, C. D. J. 1989. Vehicle routing problems with uncertainty and omitted customers. *J. Oper. Res. Soc.* **40** 1099–1108.
- Yang, W. H., K. Mathur, R. H. Ballou. 2000. Stochastic vehicle routing problem with restocking. *Transportation Sci.* **34**(1)