

# Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers

Russell W. Bent, Pascal Van Hentenryck

Department of Computer Science, Brown University, Box 1910, Providence, Rhode Island 02912  
{rbent@cs.brown.edu, pvh@cs.brown.edu}

The multiple vehicle routing problem with time windows (VRPTW) is a hard and extensively studied combinatorial optimization problem. This paper considers a dynamic VRPTW with stochastic customers, where the goal is to maximize the number of serviced customers. It presents a multiple scenario approach (MSA) that continuously generates routing plans for scenarios including known and future requests. Decisions during execution use a distinguished plan chosen, at each decision, by a consensus function. The approach was evaluated on vehicle routing problems adapted from the Solomon benchmarks with a degree of dynamism varying between 30% and 80%. They indicate that MSA exhibits dramatic improvements over approaches not exploiting stochastic information, that the use of consensus function improves the quality of the solutions significantly, and that the benefits of MSA increase with the (effective) degree of dynamism.

*Subject classifications:* vehicle routing; stochastic model applications; sampling.

*Area of review:* Transportation.

*History:* Received July 2002; revisions received September 2002, July 2003; accepted July 2003.

## 1. Introduction

The multiple vehicle routing problem with time windows (VRPTW) is a hard combinatorial optimization problem with many important applications in distribution and transportation. It has received considerable attention in the last decades, and sophisticated local search procedures are now quite effective in finding near-optimal solutions in reasonable time. In recent years, increased attention has been devoted to dynamic and/or stochastic versions of the problem (e.g., Burckert et al. 2000; Gendreau et al. 1999, 1995; 1996a, b; Ichoua et al. 2000; Novaes and Graciolli 1999; Psaraftis 1995; Savelsbergh and Sol 1998; Secomandi 2000; Yang et al. 2000). The stochastic and dynamic versions are motivated by the inherent uncertainties arising in many industrial problems and technological developments such as onboard computers and communication systems, which give transportation systems the opportunity to update plans even after the vehicle has been deployed.

Almost all existing work, which we review later in the paper, focuses either on the stochastic version or on the dynamic version exclusively. In stochastic optimization, the goal is to find, given a VRPTW where some data are stochastic, an a priori routing plan that minimizes the expected objective function. These approaches often have a recourse function to correct the plan when constraints are violated. In dynamic optimization, customer requests are not known in advance and become available incrementally over time. Typically, optimization is performed on the known quantities until an event (such as a customer request) occurs. The plan is then adapted to accommodate the new request (if possible). Of course, in general,

stochastic information is also available, or can be made available, for the dynamic version of the VRPTW, either through historical data (as suggested, for instance, in Gendreau et al. 1998, 1999 and Kilby et al. 1999) or through some available probabilistic models (as advocated, for instance, in Gendreau et al. 1996b and Yang et al. 2000). (See also Psaraftis 1995 for a review of several interesting dynamic vehicle routing applications.) How to exploit the stochastic information to meet as many customer requests as possible is the main issue addressed in this paper, and it is often mentioned as a fundamental open research problem in this area (e.g., Bianchi 2000, Gendreau et al. 1999, Gendreau and Potvin 1997, Kilby et al. 1999).

This paper considers the partially dynamic vehicle routing problem with time windows, where some customers are known at planning time while others are dynamic. The goal is to service as many customers as possible given a fixed number of vehicles. Moreover, stochastic information is assumed to be available on the dynamic (or stochastic) customers. Note that partially dynamic vehicle routing problems may vary greatly in complexity according to the degree of dynamism, i.e., the ratio *dynamic customers/total customers* (Larsen et al. 2002).

To tackle this dynamic stochastic vehicle routing problem, the paper proposes a multiple scenario approach (MSA). The key idea behind MSA is to continuously generate and solve scenarios which include both static and dynamic requests. Decisions during plan execution are based on a distinguished plan which also evolves over time. The distinguished plan is selected by a consensus function that selects the plan most similar to the current pool

of routings. The consensus function implements a least-commitment approach, a well-known heuristic in the artificial intelligence community (Stefik 1981).

This paper evaluates the effectiveness of MSA on a collection of tightly constrained dynamic vehicle routing problems with time windows. These problems are based on a subset of the Solomon benchmarks and can be viewed as a generalization of the experimental setting of Gendreau et al. (1999) that introduces stochastic information. Practical applications similar to this model include courier service applications, which were the motivation for Gendreau et al. (1999), dynamic maintenance/repair services, and intermodal services (e.g., Psaraftis 1995).

Experimental results on these dynamic stochastic routing problems with a degree of dynamism between 30% and 80% indicate that MSA produces dramatic improvements over approaches not using stochastic information. They also indicate a strong synergy between MSA and the use of a consensus function, especially for problems with a high degree of dynamism or a large number of late dynamic customers (i.e., a high effective degree of dynamism in the terminology of Larsen et al. 2002). It is also important to stress that the approach is essentially domain-independent, and we expect that many other dynamic applications may benefit from these techniques, although our experimental results consider only our original VRPTW application.

The main contributions of this paper can be summarized as follows.

- This paper shows that MSA produces dramatic benefits in terms of solution quality by exploiting stochastic information on dynamic customers. This answers an open research issue mentioned in a variety of prior work (e.g., Bianchi 2000, Gendreau et al. 1999, Gendreau and Potvin 1997, Kilby et al. 1999), at least for problems with a degree of dynamism between 30% and 80% (the moderately dynamic problems of Larsen et al. 2002).

- This paper shows that it is beneficial to use the consensus function for choosing the distinguished plan instead of travel distance, especially for problems with high degrees of dynamism. In fact, it is the synergy between stochastic information and the consensus function that seems to be responsible for the dramatic improvements over other approaches.

- This paper also shows that maintaining multiple routing plans is fundamental in obtaining high-quality solutions in dynamic routing problems, even when no stochastic information is available. This confirms and abstracts the results of Gendreau et al. (1999) by making them independent of a specific local search.

The rest of this paper is organized as follows. Section 2 describes the problem formulation. Section 3 presents the traditional greedy approach. The main contributions of the paper are presented in the next three sections. Section 4 introduces the multiple plan approach (MPA) that continuously generates and maintains a pool of solutions but does not use stochastic information. Section 5 presents the MSA

approach that generalizes MPA with stochastic information. Section 6 gives the experimental results. The last two sections present the related work and conclude the paper.

## 2. Problem Formulation

**Customers.** Each problem contains  $N$  customer regions (regions for short) and  $N'$  customer service requests from the regions (customers for short). The customers are numbered  $1, \dots, N'$ , in the (chronological) order of their requests. The regions are numbered  $1, \dots, N$ . A depot is represented by the number 0. The set  $\{0, 1, \dots, N'\}$  represents all the sites in the problem. The travel cost between sites  $i$  and  $j$  is represented by  $c_{ij}$ . These travel costs satisfy the triangle inequality, i.e.,  $\forall i, j, k (c_{ik} \leq c_{ij} + c_{jk})$ . Every request has a demand  $q_i \geq 0$  and a service time  $s_i \geq 0$ .

**Vehicles.** Each problem has  $m$  identical vehicles available for use, each with capacity  $Q$ .

**Routes.** A vehicle route, or route for short, starts at the depot, serves some number of customers at most once, and returns to the depot. Formally, a route is a sequence  $[0, v_1, \dots, v_n, 0]$ , where  $1 \leq v_i \leq N'$  and all  $v_i$  are distinct. The demand of a route is denoted by  $q(r) = \sum_{i=1}^n q_i$ . The travel cost of a route  $r$  is denoted by  $c(r)$  and is the cost of visiting all of its customers, i.e.,  $c(r) = c_{0v_1} + c_{v_1v_2} + \dots + c_{v_{n-1}v_n} + c_{v_n0}$ .

**Routing Plan.** A routing plan, or plan for short, is a set of routes  $\{r_1, \dots, r_m\}$  servicing each customer exactly once. A routing plan assigns a unique successor and predecessor for each customer. For a plan,  $\sigma$ , the successor of customer  $i$  is denoted by  $\text{succ}(i, \sigma)$  and the predecessor is denoted by  $\text{pred}(i, \sigma)$ . The travel cost of a plan is denoted by  $c(\sigma)$ , i.e.,  $c(\sigma) = \sum_{r=1}^m c(r)$ . We also use  $\text{cust}(r)$  and  $\text{cust}(\sigma)$  to denote the customers of a route  $r$  and a plan  $\sigma$ .

**Time Windows.** Each site  $i$  has a time window specified by an interval  $[e_i, l_i]$ , which represents the earliest and latest possible arrival times, respectively. Vehicles must arrive by  $l_i$  but can arrive before  $e_i$ . However, they are required to wait until  $e_i$  to begin service. Observe that  $e_0$  represents the earliest time vehicles can leave the depot and  $l_0$  represents the time all the vehicles must return to the depot.

A routing plan implicitly specifies the earliest possible service time  $a_i$  of each customer  $i$ , as well as the earliest possible arrival time  $a(r)$  of each route  $r$ . These values are computed using simple recursive equations (Kindervater and Savelsbergh 1997). A plan also specifies the earliest and latest departure times from each customer and these quantities are used later in the paper.

A solution to the offline VRPTW is a routing plan  $\sigma = \{r_1, \dots, r_m\}$  that satisfies the capacity and time window constraints, i.e.,

$$\begin{cases} q(r_j) \leq Q & (1 \leq j \leq m), \\ a(r_j) \leq l_0 & (1 \leq j \leq m), \\ a_i \leq l_i & (\forall i \in \text{cust}(\sigma)). \end{cases}$$

The objective here is to find a solution maximizing the number of served customers  $|\text{cust}(\sigma)|$ , i.e., a routing plan satisfying all constraints and servicing as many customers as possible.

In the dynamic VRPTW, customer requests are not known in advance and become available during the course of the day. In general, a number of requests are available initially, while others become available during plan execution. We assume that the distribution of the requests—or an approximation thereof—is available either through historical data, as suggested in Gendreau et al. (1998, 1999) and Kilby et al. (1999), or through some available probabilistic models as advocated in Gendreau et al. (1996b) and Yang et al. (2000). For each incoming request, the dynamic algorithm must decide whether to accept or reject the customers. Once a request is accepted, it must be serviced. (We also experimented with the simpler problem, in which requests were not necessarily serviced. The results were, in fact, rather similar in our limited experiments.)

### 3. The Greedy Algorithm

The greedy algorithm (e.g., Bianchi 2000, Gendreau and Potvin 1997, Lund et al. 1996, Larsen et al. 2002) is the traditional approach to dynamic vehicle routing. (We are referring to instances based on the Solomon benchmarks, which are tightly constrained; see Krumke et al. 2002 for an application with fundamentally different properties.) It starts with a routing plan accommodating as many initial requests as possible and tries to insert new requests as they become available. The initial routing plan is, in general, obtained by a sophisticated local search, which is the only viable alternative for large-scale problems. When a request for customer  $c$  arrives, the greedy algorithm determines whether there is a feasible insertion point in the plan. If no such insertion point exists, the request is rejected. Otherwise, the new customer is inserted to minimize travel cost.

It is important to say a few words about plan execution. In general, a routing plan does not specify unique departure dates for customers, but rather it constrains the earliest and the latest feasible departure times. In the dynamic setting, it is important to delay decisions as much as possible (because new requests may materialize) while not inducing any delay in service. Hence, the traditional strategy is to depart from a customer to arrive at the next customer as early as possible but not before the start of its time window. In other words, a vehicle never waits at a customer site before being serviced but may wait before departing to the next customer.

It is also interesting to observe that, in general, simple extensions to the greedy approach do not work well for vehicle routing with time windows based on the Solomon benchmarks. For instance, in our experiments, reoptimizing the plan after an insertion (using the state-of-the-art vehicle routing algorithm presented in Bent and Van Hentenryck 2004) often led to solutions of poorer quality. The intuitive

reason is that optimized solutions tends to produce tight schedules for the vehicles in use, which reduce their ability to accommodate future (late) customers. New, or less appropriate, vehicles must then be used for servicing these customers. More creative solutions must be used such as the adaptive memory approach of Gendreau et al. (1999). Note, however, that Krumke et al. (2002) report a successful application of reoptimization for a large-scale problem of a different type of applications, where optimal solutions produce short routes.

### 4. The Multiple Plan Approach

MPA is a fundamental generalization of the greedy approach. Its key idea is to maintain a set of plans at every execution step. MPA was motivated by the seminal work of Gendreau et al. (1999), which proposed a parallel tabu-search algorithm organized around multiple solutions and an adaptive memory. MPA abstracts and generalizes their approach by making it independent of the search procedure used to generate solutions, because different search techniques may be appropriate for particular applications (e.g., Krumke et al. 2002). More precisely, MPA continuously generates routing plans that are consistent with the current decisions and deletes incompatible ones. Maintaining multiple plans is not sufficient in itself, because decisions must be taken regarding a specific plan to guarantee service of accepted requests. As a consequence, MPA maintains a distinguished plan at each execution step. The distinguished plan obviously evolves over time, and all the plans maintained by MPA must be consistent with it. How to select the distinguished plan is a critical aspect of MPA, which we discuss later in the section.

More precisely, MPA handles four types of events: (1) customer requests, (2) vehicle departures, (3) plan generations, and (4) timeouts. Customer requests update the set of plans to accommodate the new request. Vehicle departures may render some routing plans invalid. The generation of a new plan may change the distinguished plan. Finally, some plans may become invalid over time. This arises when a plan specifies a vehicle departure (the latest departure time from a customer has been reached), while the distinguished plan specifies that the vehicle should wait at its current location. Timeouts capture these events. We now specify how to handle each event in more detail and discuss how to rank the plans next.

#### 4.1. Event Handling

At each time  $t$ , MPA maintains a set of plans  $S_t$  and a distinguished plan  $\sigma_t^*$ . For each event, we specify how to compute  $S_{t+1}$  and  $\sigma_{t+1}^*$  from past decisions,  $S_t$  and  $\sigma_t^*$ . Each event type is specified in isolation, although several of them may occur simultaneously. It is easy to order them appropriately when this happens by selecting the events in the following order: timeouts, plan generations, customer requests, and vehicle departures. We make use of a set

of functions  $f_t$  to rank the plans. Given a time  $t$  and a plan  $\sigma$ ,  $f_t(\sigma)$  returns a real value. Finally, observe that the implementation is event driven. In other words, although we specify  $S_t$  and  $\sigma_t^*$  for all  $t$ , the implementation considers only the times where an actual event occurs.

**Timeout.** At time  $t$ , some plan  $\sigma$  may become infeasible. This happens when a vehicle  $v$  is waiting at a customer  $r$ , while plan  $\sigma$  specifies that  $t$  is the latest departure for  $r$ .

$$S_{t+1} := \{\sigma \in S_t \mid \text{FEASIBLE}(\sigma, t)\};$$

$$\sigma_{t+1}^* := \sigma_t^*;$$

where  $\text{FEASIBLE}(\sigma, t)$  holds if  $\sigma$  is feasible at time  $t$ :

$$\forall r \in \text{DEPART}(t): \text{LDT}(\sigma, \text{succ}(\sigma, r)) \leq t,$$

where  $\text{LDT}(\sigma, r)$  is the latest departure time from customer  $r$  in  $\sigma$  and  $\text{DEPART}(t)$  denotes the set of customers from which a vehicle departed before or at time  $t$ .

**Plan Generation.** When a new plan  $\sigma$  is generated at time  $t$ , it is added to  $S_t$  and the new distinguished plan is recomputed. Note that plans are guaranteed to be compatible with the current distinguished plan because they include all existing decisions, and plan generation is canceled whenever customer requests and vehicle departures occur.

$$S_{t+1} := S_t \cup \{\sigma\};$$

$$\sigma_{t+1}^* := \arg \max(\sigma \in S_{t+1}) f_t(\sigma).$$

**Customer Request.** For a customer request  $r$  at time  $t$ , MPA must determine which plan in  $S_t$  can accommodate  $r$ . If none of them can, the request is rejected. Otherwise, the request is accepted, and  $S_{t+1}$  is the set of plans where the request has been inserted at minimal travel cost.

$$F := \{\text{INSERT}(\sigma, r) \mid \sigma \in S_t \ \& \ \text{FEASIBLEINSERT}(\sigma, r)\};$$

if  $F \neq \emptyset$  then

$$S_{t+1} := F;$$

$$\sigma_{t+1}^* := \arg \max(\sigma \in S_{t+1}) f_t(\sigma);$$

else

$$S_{t+1} := S_t;$$

$$\sigma_{t+1}^* := \sigma_t^*;$$

where  $\text{FEASIBLEINSERT}(\sigma, r)$  returns true iff there is an insertion point in plan  $\sigma$  for customer  $r$  that satisfies the constraints, and where  $\text{INSERT}(\sigma, r)$  returns a plan  $\sigma'$  (if it exists) where  $r$  has been inserted in  $\sigma$  to minimize travel cost while satisfying the constraints.

**Vehicle Departure.** When plan  $\sigma_t^*$  specifies that vehicle  $v$  must depart from customer  $r$ , it is necessary to remove

all plans in  $S_t$  that are incompatible with this departure.

$$S_{t+1} := \{\sigma \in S_t \mid \text{COMPATIBLE}(\sigma, \sigma_t^*, t)\};$$

$$\sigma_{t+1}^* := \arg \max(\sigma \in S_{t+1}) f_t(\sigma);$$

where  $\text{COMPATIBLE}(\sigma, \sigma_t^*, t)$  is true if plan  $\sigma$  is compatible with plan  $\sigma_t^*$  up to time  $t$ . More formally,  $\text{COMPATIBLE}(\sigma, \sigma_t^*, t)$  holds iff

$$\forall r \in \text{DEPART}(t): \text{succ}(\sigma_t^*, r) = \text{succ}(\sigma, r).$$

## 4.2. The Consensus Function

MPA is parametrized by the ranking functions  $f_t$ , which select the distinguished plan at each time  $t$ . An obvious choice for  $f_t$  would be to select the plan with the smallest travel cost. However, it is possible to do substantially better on our collection of benchmarks, especially on instances in which there are many late customers and/or many dynamic requests. The key idea is to use a consensus function that selects a plan most similar to other plans in  $S_t$ . Because the resulting distinguished plan does not depart from other plans too dramatically, consensus functions may be viewed as a least commitment strategy, a well-known heuristic in the artificial intelligence community (Stefik 1981). More precisely, at each time  $t$ , the algorithm maintains a two-dimensional matrix  $M_t$ , and  $M_t[v, r]$  denotes the number of plans in  $S_t$  where vehicle  $v$  departs for customer  $r$  next. More formally, the matrix  $M_t$  is defined as

$$M_t[v, r] = \#\{\sigma \in S_t \mid \text{succ}(\sigma, \text{LDC}(v)) = r\},$$

where  $\text{LDC}(v)$  is the last customer from which vehicle  $v$  departed in the plan execution. The consensus function  $f_t$  is then defined as

$$f_t(\sigma) = \sum_{v=1}^m M_t[v, \text{succ}(\sigma, \text{LDC}(v))].$$

## 5. The Multiple Scenario Approach

MPA with the consensus function produces dramatic improvements in quality compared to the greedy algorithm. This section introduces MSA, which significantly outperforms MPA by exploiting stochastic information.

The key idea behind MSA consists of generating new routing plans for scenarios that include both existing requests as well as possible future requests. Future requests are simply obtained by sampling their probability distributions. Once a routing  $\sigma$  is obtained for such a scenario, MSA projects  $\sigma$  on the known requests by removing future customers from the plan. The resulting *projected plan*  $\sigma^-$  leaves room for accommodating future requests if they materialize and is added to the pool  $S_t$ . MSA closely resembles MPA, because its basic event-handling procedures are essentially the same. Only plan generation differs in that it uses scenarios involving future requests (instead of known requests only). The solution to these scenarios are then projected on known requests before being inserted in  $S_t$ .

## 6. Experimental Results

This section presents the experimental results. We first describe how we adapted the traditional Solomon benchmarks (Solomon 1987) to the dynamic setting. We then specify a few parameters for the algorithms. We then compare the various approaches and analyze the role of stochastic information and the consensus function in the results. We also give some interesting observations on the scenario survival rates.

### 6.1. The Experimental Setting

The dynamic problems are generated from the Solomon benchmarks (Solomon 1987). Each customer in the Solomon instances becomes a customer region in our stochastic problems. Our instances are generated so that the expected number of customer requests  $n$  is 100 and the expected number of requests from a particular customer region is 1. As a consequence, the instances preserve much of the structure of the Solomon instances, many of which are very challenging.

**Request Arrivals.** To specify the arrival distribution of customer requests, the time horizon  $H = l_0 - e_0$  is divided into four time periods. Period 0 corresponds to known requests (requests available before  $e_0$ ). Periods 1 to 3 can be thought of as representing morning, early afternoon, and late afternoon. Each customer region is labeled with a time region according to its time window and its distance to the depot. The distance to the depot is taken into account to avoid generating requests that clearly cannot be serviced. There are no requests labeled with period 3 (late afternoon). If region  $i$  is labeled with period 0, then this region requests service before  $e_0$ . If the region is labeled with period 1, there is a request for service before  $e_0$  with probability 0.5 and a request for service in period 1 with probability 0.5. If the region is labeled with period 2, then there are four possibilities, leading to four classes of instances. Problems in Class 1 have many initial requests and few late requests. Problems in Class 2 have many initial requests and many late requests. Problems in Class 3 have many initial requests and a mix of early and late requests. Finally, problems in Class 4 have few initial requests and a lot of late requests. More precisely, in Class 1, there is a request before  $e_0$  with probability 0.5, a request in period 1 with probability 0.4, and a request in period 2 with probability 0.1. In Class 2, there is a request before  $e_0$  with probability 0.5, a request in period 1 with probability 0.1, and a request in period 2 with probability 0.4. In Class 3, a request is generated according to Class 1 with probability 0.5 and according to Class 2 with probability 0.5. In Class 4, there is a request before  $e_0$  with probability 0.2, a request in period 1 with probability 0.2, and a request in period 2 with probability 0.6. Observe that these probabilities are independent as to allow multiple or no requests from a particular region.

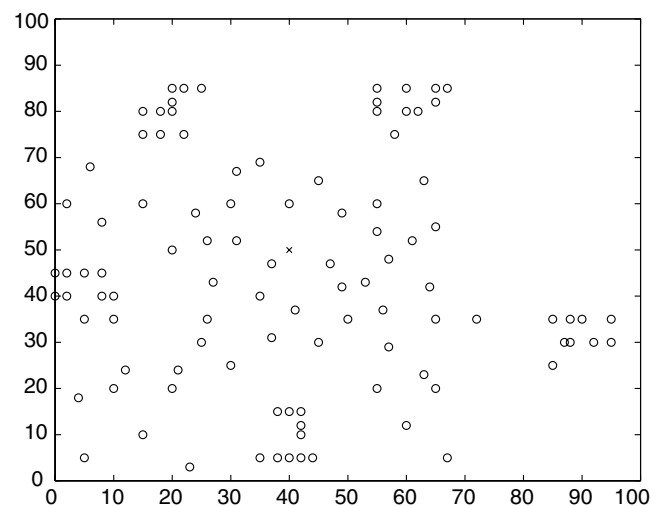
The instances were generated using these probabilities. If there is a request in time period  $k$ , then the request arrival time for customer  $c$  is drawn uniformly at random from the time interval  $[(k - 1) * H/3, \min(\lambda_c, k * (H/3) - 1)]$ , where  $\lambda_c$  is the latest time a vehicle can depart from the depot, service  $c$ , and return to the depot. The probabilities for each customer region are available to a dynamic algorithm.

**Instance Data.** The instances take their data (customer locations, time windows, demands, service times) from the class RC in the Solomon benchmarks. RC problems have a mix of clustered and randomly distributed customers, which makes them particularly interesting. Moreover, we selected problems with diverse time windows. In particular, RC101 problems have a good mix of short and long time windows and have a high percentage of 1-region customers. RC102 problems have many regions with short time windows, 45 1-region customers, and 45 2-region customers. RC104 problems have regions with extremely long time windows and 83 2-region customers. Figure 1 shows the location of the customer regions in Solomon's RC class. As a result, the instances cover a wide spectrum of possibilities and structures.

The number of vehicles available for the dynamic algorithms was determined by solving the offline problems (i.e., assuming that all requests are known in advance) and adding two vehicles. The offline and plan generation problems were solved using the state-of-the-art vehicle routing algorithm (local search) presented in Bent and Van Hentenryck (2004).

In the Solomon RC instances, the horizon is 240 time units, which is too short for the units to be seconds. To run a large number of experiments, the units cannot be minutes, either. As a consequence, to obtain a realistic experimental setting, we scaled all spatial positions, time windows, travel costs, and service times by the factor  $y/H$ . By choosing  $y = 30$  and minutes as units, we obtain meaningful instances that can be experimented within reasonable time.

**Figure 1.** Customer region locations.



**Table 1.** Class 1 problem results.

	Offline	Greedy		MPA <sup>d</sup>		MPA <sup>c</sup>		MSA <sup>d</sup>		MSA <sup>c</sup>	
	V	C	V	C	V	C	V	C	V	C	V
RC101-1	14	4.0	16.0	1.6	15.8	1.8	16.0	1.0	16.0	<b>0.6</b>	<b>16.0</b>
RC101-2	14	10.0	16.0	3.2	16.0	3.0	16.0	<b>1.8</b>	<b>16.0</b>	2.6	16.0
RC101-3	13	9.0	15.0	1.6	15.0	1.2	15.0	1.8	15.0	<b>1.0</b>	<b>15.0</b>
RC101-4	15	3.0	17.0	0.2	16.8	<b>0.0</b>	17.0	<b>0.0</b>	<b>17.0</b>	0.2	17.0
RC101-5	15	5.0	17.0	2.0	17.0	1.4	17.0	<b>0.4</b>	<b>17.0</b>	1.0	17.0
Avg	14.2	6.20	16.2	1.72	16.1	1.48	16.2	<b>1.00</b>	<b>16.2</b>	1.08	16.2
RC102-1	12	6.0	14.0	3.0	14.0	2.2	14.0	<b>2.0</b>	<b>14.0</b>	2.4	14.0
RC102-2	11	8.0	13.0	1.0	13.0	2.4	13.0	<b>0.4</b>	<b>13.0</b>	0.8	13.0
RC102-3	13	11.0	15.0	2.4	15.0	4.8	15.0	1.0	15.0	<b>0.8</b>	<b>15.0</b>
RC102-4	12	8.0	14.0	1.8	13.8	1.8	14.0	<b>1.2</b>	<b>14.0</b>	1.4	14.0
RC102-5	13	5.4	15.0	1.8	15.0	2.2	15.0	1.2	15.0	<b>0.6</b>	<b>15.0</b>
Avg	12.2	7.68	14.2	2.00	14.2	2.68	14.2	<b>1.16</b>	<b>14.2</b>	1.20	14.2
RC104-1	9	5.0	11.0	1.4	11.0	0.2	11.0	<b>0.0</b>	<b>11.0</b>	0.2	11.0
RC104-2	10	15.6	12.0	0.8	12.0	1.0	12.0	<b>0.0</b>	<b>11.8</b>	<b>0.0</b>	12.0
RC104-3	11	17.2	13.0	<b>0.0</b>	12.6	<b>0.0</b>	12.8	<b>0.0</b>	<b>12.4</b>	<b>0.0</b>	13.0
RC104-4	10	12.2	12.0	<b>0.0</b>	<b>11.4</b>	0.4	12.0	<b>0.0</b>	11.8	0.2	12.0
RC104-5	9	7.6	11.0	<b>0.0</b>	11.0	0.8	11.0	<b>0.0</b>	<b>10.8</b>	<b>0.0</b>	11.0
Avg	9.8	11.52	11.8	0.44	11.6	0.48	11.8	<b>0.00</b>	<b>11.6</b>	0.04	11.8
Avg	12.07	8.47	14.07	1.39	14.0	1.55	14.1	<b>0.72</b>	<b>14.0</b>	0.77	14.1

Note that all results are reported in the original units for readers who are familiar with the Solomon benchmarks.

**Algorithms.** All algorithms were run on a Sun Ultra 10 with Sun's c++ compiler. The MPA and MSA algorithms generate 50 initial plans using the local search algorithm mentioned earlier. Each plan is optimized for 30 seconds. During plan execution, plans are generated by using the local search algorithm for 10 seconds.

## 6.2. The Results

**Comparison of the Approaches.** Tables 1 through 4 compare the quality of the results produced by each approach. Time is not an issue here because vehicle travel is significantly larger than the computation time of our local search algorithm. The tables report the performance of the greedy, MPA, and MSA approaches on each class of instances. A line in the tables reports the average numbers

**Table 2.** Class 2 problem results.

	Offline	Greedy		MPA <sup>d</sup>		MPA <sup>c</sup>		MSA <sup>d</sup>		MSA <sup>c</sup>	
	V	C	V	C	V	C	V	C	V	C	V
RC101-1	11	2.0	13.0	0.6	13.0	2.0	13.0	<b>0.0</b>	<b>13.0</b>	0.2	13.0
RC101-2	13	3.0	14.0	2.0	14.0	2.0	14.0	<b>1.0</b>	<b>14.0</b>	1.4	14.0
RC101-3	15	7.0	17.0	0.4	17.0	0.8	17.0	<b>0.0</b>	<b>17.0</b>	<b>0.0</b>	17.0
RC101-4	15	6.0	17.0	1.8	17.0	2.2	17.0	<b>0.6</b>	<b>17.0</b>	0.8	17.0
RC101-5	14	6.0	16.0	1.8	16.0	<b>1.4</b>	<b>16.0</b>	1.6	16.0	<b>1.4</b>	16.0
Avg	13.6	4.80	15.4	1.32	15.4	1.68	15.4	<b>0.64</b>	<b>15.4</b>	0.76	15.4
RC102-1	13	8.2	15.0	<b>0.0</b>	15.0	0.4	15.0	<b>0.0</b>	<b>14.6</b>	0.4	15.0
RC102-2	12	8.6	14.0	1.4	14.0	1.2	14.0	<b>0.6</b>	<b>14.0</b>	1.2	14.0
RC102-3	12	10.0	14.0	3.0	14.0	3.2	14.0	<b>2.0</b>	<b>14.0</b>	<b>2.0</b>	14.0
RC102-4	13	6.0	15.0	0.6	15.0	0.8	15.0	<b>0.2</b>	<b>15.0</b>	0.4	15.0
RC102-5	12	8.0	14.0	3.0	14.0	3.4	14.0	<b>2.6</b>	<b>14.0</b>	2.8	14.0
Avg	12.4	8.16	14.4	1.60	14.4	1.80	14.4	<b>1.08</b>	<b>14.3</b>	1.36	14.4
RC104-1	10	15.2	12.0	11.2	12.0	8.6	12.0	6.2	12.0	<b>3.0</b>	<b>12.0</b>
RC104-2	10	17.8	12.0	9.4	12.0	10.0	12.0	5.4	12.0	<b>2.6</b>	<b>12.0</b>
RC104-3	10	16.4	12.0	7.8	12.0	7.2	12.0	2.0	12.0	<b>0.8</b>	<b>12.0</b>
RC104-4	11	11.0	13.0	2.4	13.0	2.0	13.0	0.8	13.0	<b>0.6</b>	<b>13.0</b>
RC104-5	10	14.6	12.0	5.2	12.0	9.0	12.0	4.2	12.0	<b>0.2</b>	<b>12.0</b>
Avg	10.2	15.00	12.2	7.20	12.2	7.36	12.2	3.72	12.2	<b>1.44</b>	<b>12.2</b>
Avg	12.07	9.32	14.0	3.37	14.0	3.61	14.0	1.81	14.0	<b>1.19</b>	<b>14.0</b>

**Table 3.** Class 3 problem results.

	Offline	Greedy		MPA <sup>d</sup>		MPA <sup>c</sup>		MSA <sup>d</sup>		MSA <sup>c</sup>	
	V	C	V	C	V	C	V	C	V	C	V
RC101-1	13	8.0	15.0	1.6	15.0	1.2	15.0	<b>0.6</b>	<b>15.0</b>	0.8	15.0
RC101-2	14	8.8	16.0	1.4	16.0	<b>1.2</b>	<b>16.0</b>	2.2	16.0	1.4	16.0
RC101-3	12	6.0	14.0	1.0	14.0	1.0	14.0	<b>0.6</b>	<b>13.8</b>	0.8	14.0
RC101-4	15	7.6	17.0	1.4	17.0	2.0	17.0	<b>0.6</b>	<b>17.0</b>	1.0	17.0
RC101-5	14	4.0	16.0	0.6	16.0	1.2	16.0	<b>0.0</b>	<b>15.4</b>	0.8	16.0
Avg	13.6	6.88	15.6	1.20	15.6	1.32	15.6	<b>0.80</b>	<b>15.6</b>	0.96	15.6
RC102-1	13	6.0	15.0	1.8	15.0	2.2	15.0	<b>1.6</b>	<b>15.0</b>	<b>1.6</b>	15.0
RC102-2	12	5.0	14.0	2.4	14.0	3.8	14.0	<b>0.8</b>	<b>14.0</b>	1.8	14.0
RC102-3	11	9.0	13.0	2.0	13.0	1.6	13.0	<b>0.8</b>	<b>13.0</b>	<b>0.8</b>	13.0
RC102-4	13	6.0	15.0	3.2	15.0	2.6	15.0	<b>0.8</b>	<b>15.0</b>	1.8	15.0
RC102-5	13	7.0	15.0	1.2	15.0	2.6	15.0	<b>1.4</b>	<b>15.0</b>	1.6	15.0
Avg	12.4	6.60	14.4	2.12	14.4	2.56	14.4	<b>1.08</b>	<b>14.4</b>	1.52	14.4
RC104-1	10	20.2	12.0	8.2	12.0	5.6	12.0	4.8	12.0	<b>2.4</b>	<b>12.0</b>
RC104-2	10	9.0	12.0	1.6	12.0	1.0	12.0	1.0	12.0	<b>0.2</b>	<b>12.0</b>
RC104-3	10	12.0	12.0	2.8	12.0	1.8	12.0	1.4	12.0	<b>0.4</b>	<b>12.0</b>
RC104-4	10	12.0	12.0	2.4	12.0	2.2	12.0	1.6	12.0	<b>0.2</b>	<b>12.0</b>
RC104-5	10	17.4	12.0	2.8	12.0	4.4	12.0	2.2	12.0	<b>0.6</b>	<b>12.0</b>
Avg	10.0	14.12	12.0	3.56	12.0	3.00	17.0	2.20	17.0	<b>0.76</b>	<b>12.0</b>
Avg	12.0	9.20	14.0	2.29	14.0	2.29	14.0	1.36	14.0	<b>1.08</b>	<b>14.0</b>

of unserved customers (C) and the average numbers of used vehicles (V) in the final plans for five runs of the algorithms on the instance. The number of vehicles for the offline solutions are also provided (recall that C = 0 for the offline solutions). We report the number of vehicles because some algorithms are able to reduce the number of vehicles and do not use all available vehicles. For MPA and MSA, we report results when travel distance (MPA<sup>d</sup> and MSA<sup>d</sup>)

and the consensus function (MPA<sup>c</sup> and MSA<sup>c</sup>) are used as ranking functions.

The results can be informally summarized as follows.

1. MPA produces significant improvements over the greedy algorithm. On average, MPA has about 1.5, 3.5, 2.3, and 6 unserved customers for Classes 1 to 4, while the greedy algorithm has about 8.5, 9, 9, and 16 unserved customers. This confirms the results of Gendreau et al.

**Table 4.** Class 4 problem results.

	Offline	Greedy		MPA <sup>d</sup>		MPA <sup>c</sup>		MSA <sup>d</sup>		MSA <sup>c</sup>	
	V	C	V	C	V	C	V	C	V	C	V
RC101-1	14	3.0	16.0	<b>0.0</b>	<b>15.8</b>	1.0	16.0	<b>0.0</b>	<b>16.0</b>	1.0	16.0
RC101-2	13	8.0	15.0	<b>2.2</b>	<b>15.0</b>	3.4	15.0	2.8	15.0	3.6	15.0
RC101-3	14	4.0	16.0	0.6	16.0	1.6	16.0	<b>0.0</b>	<b>16.0</b>	1.6	16.0
RC101-4	15	11.0	17.0	<b>1.0</b>	<b>17.0</b>	1.2	17.0	<b>1.0</b>	17.0	1.4	17.0
RC101-5	14	11.8	16.0	2.8	16.0	3.2	16.0	<b>2.0</b>	<b>16.0</b>	2.2	16.0
Avg	14.0	7.56	16.0	1.32	16.0	2.08	16.0	<b>1.16</b>	<b>16.0</b>	1.96	16.0
RC102-1	13	6.0	15.0	0.2	14.2	<b>0.0</b>	15.0	<b>0.0</b>	<b>14.8</b>	0.4	15.0
RC102-2	13	9.0	15.0	1.8	15.0	<b>1.4</b>	15.0	1.6	15.0	<b>1.4</b>	<b>15.0</b>
RC102-3	13	15.6	15.0	1.8	15.0	1.0	15.0	<b>0.2</b>	<b>15.0</b>	1.4	15.0
RC102-4	12	3.6	14.0	0.4	13.8	<b>0.0</b>	14.0	<b>0.0</b>	<b>14.0</b>	<b>0.0</b>	14.0
RC102-5	13	5.0	15.0	<b>0.2</b>	<b>15.0</b>	1.4	15.0	0.6	15.0	0.6	15.0
Avg	12.8	7.84	14.8	0.88	14.6	0.76	14.8	<b>0.48</b>	<b>14.8</b>	0.68	14.8
RC104-1	11	41.0	13.0	16.8	13.0	15.0	13.0	15.6	13.0	<b>3.2</b>	<b>13.0</b>
RC104-2	12	41.0	14.0	18.2	14.0	15.8	14.0	16.0	14.0	<b>3.4</b>	<b>14.0</b>
RC104-3	11	26.0	13.0	16.4	13.0	15.2	13.0	13.8	13.0	<b>5.6</b>	<b>13.0</b>
RC104-4	10	23.0	12.0	16.0	12.0	18.2	12.0	15.6	12.0	<b>2.4</b>	<b>12.0</b>
RC104-5	9	27.8	11.0	14.4	11.0	11.0	11.0	8.2	11.0	<b>2.0</b>	<b>11.0</b>
Avg	10.6	31.76	12.6	16.36	12.6	15.04	12.6	13.84	12.6	<b>3.32</b>	<b>12.6</b>
Avg	12.47	15.72	14.5	6.19	14.5	5.96	14.5	5.16	14.5	<b>1.99</b>	<b>14.5</b>

(1999) because  $MPA^d$  is a generalization and abstraction of their work.

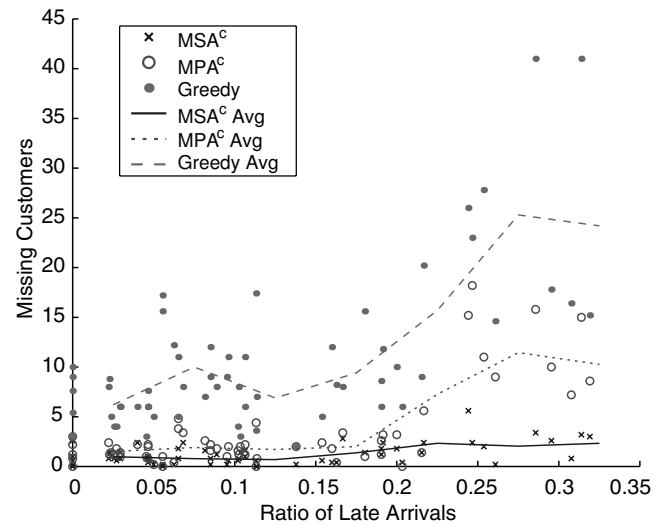
2. MSA produces significant improvements over MPA. On average, MSA has about 0.75, 1.2, 1, and 2 unserved customers for Classes 1 to 4. The benefits are particularly clear on the RC104 instances from Class 4, where the number of unserved customers drops from about 15 to about 3.5 on average. More generally, the improvements of MSA over MPA increase with the degree of (effective) dynamism. (Note that RC104 instances have many customers with late time windows and with large time windows, two features that seem to magnify the benefits of MSA.)

3. The consensus function brings substantial benefits over travel distance for high degree of (effective) dynamism and seems very robust in general. It is slightly outperformed by travel distance for low degree of dynamism in general. It produces significant improvements for high degree of dynamism. In fact, *it is the synergy between stochastic information and the consensus function that seems to be responsible for the dramatic improvements over other approaches.*

**The Benefits of Stochastic Information.** We now study the benefits of stochastic information in more detail. Figures 2 and 3 compare the quality of the solutions of greedy, MPA, and MSA approaches as a function of the degree of dynamism and as a function of the percentage of late stochastic customers, which is related to the effective degree of dynamism. The results indicate that the benefits of MSA are particularly dramatic for large percentages of stochastic customers and/or large percentages of late stochastic customers. Both the MPA and the MSA approaches use the consensus function here.

**The Benefits of Consensus.** We now try to understand the impact of the ranking function on the quality

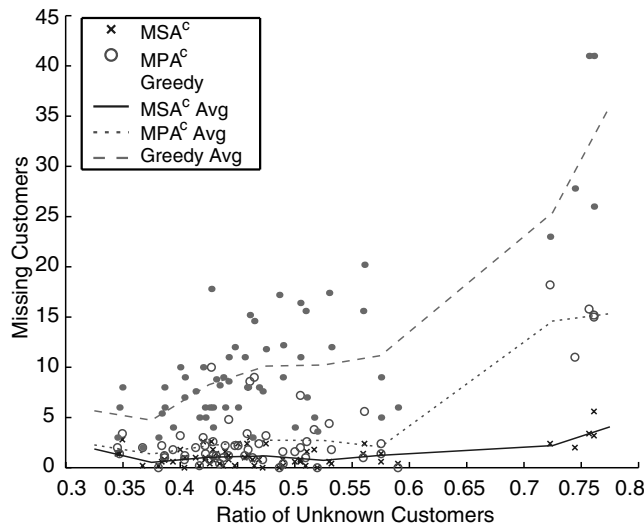
**Figure 3.** Quality with respect to percentage of late stochastic customers.



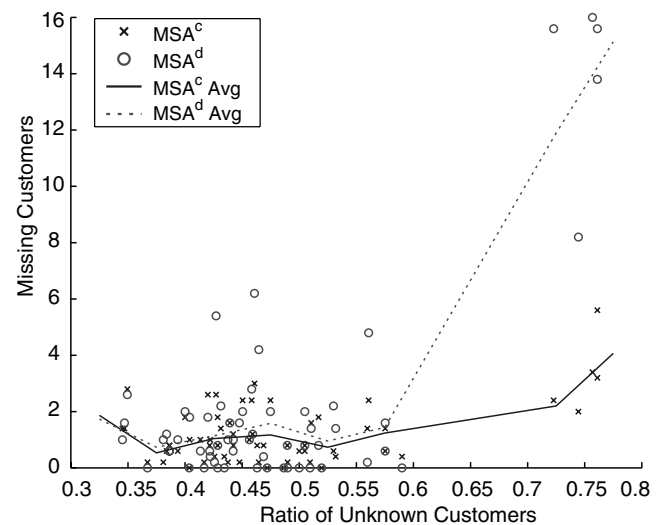
of the results. Figures 4 and 5 compare the quality of the MSA algorithm with two ranking functions: travel distance and the consensus function. The experimental results clearly show the value of the consensus function, especially when the degree of dynamism and/or the percentage of late stochastic customers increase. *As a consequence, the experimental results indicate that both stochastic information and the consensus function are critical in obtaining good quality solutions in these dynamic vehicle routing instances.* The use of a consensus function makes the MSA approach much more robust and allows it to accommodate many more (late) stochastic customers.

As a consequence, the synergy between MSA and the consensus function allows MSA to scale up to a higher

**Figure 2.** Quality with respect to the degree of dynamism.

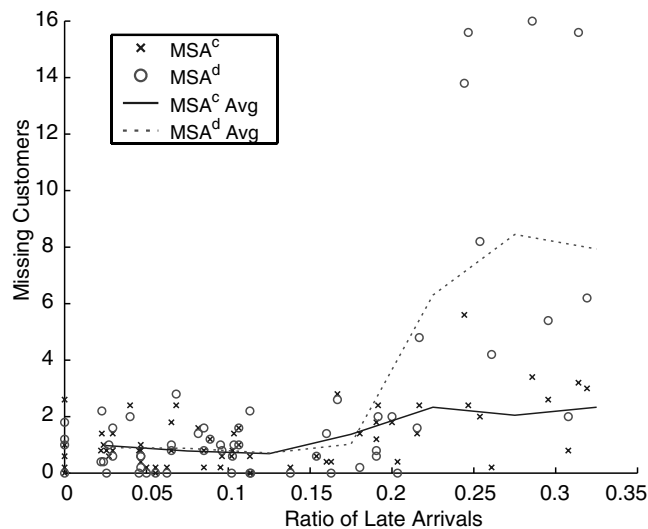


**Figure 4.** Quality with respect to the degree of dynamism.





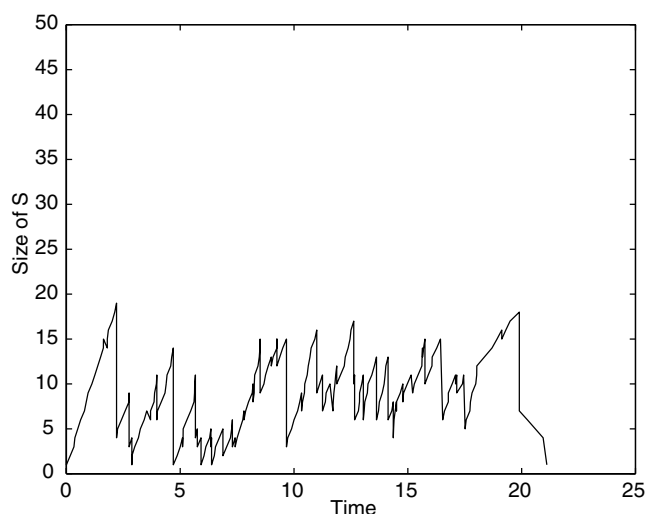
**Figure 5.** Quality with respect to percentage of late stochastic customers.



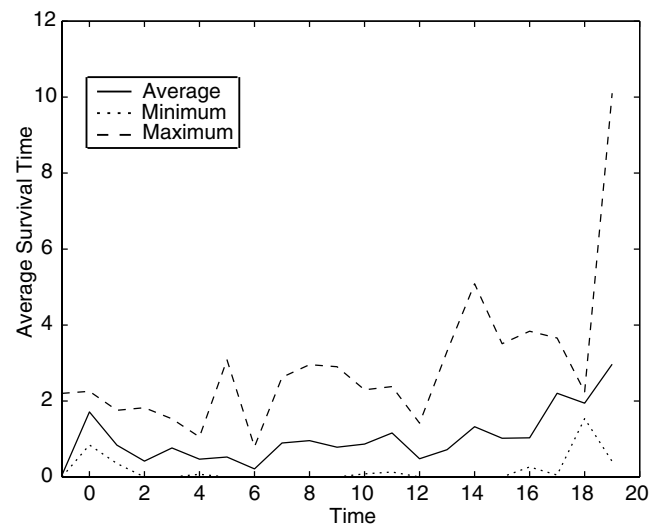
degree of dynamism, an open issue mentioned in Larsen et al. (2002).

**Survival Rates.** It is interesting to understand how many plans are kept by MSA and the survival time of these plans. Figure 6 shows the size of  $S$  over time for a Class 3 problem. During plan execution, 244 distinct plans were generated. Observe the large drops in the size of  $S$  at time 0, 2, 5, 10, and 20. These drops were preceded by vehicle departure events, indicating that vehicle departures are the most likely events to trigger large reductions in  $S$ . Figure 7 shows the average, minimum, and maximum survival times of scenarios generated during 1 minute intervals for the same Class 3 problem. The -1 block represents the scenarios generated before  $e_0$ . From both figures, it is clear that the one scenario to survive until the end (at time 21, when the size of  $S$  dropped to 1) was created at time 19,

**Figure 6.** Size of  $S$  for Class 3 Problem RC104-3.



**Figure 7.** Average survival time for Class 3 Problem RC104-3.



as indicated by the large spike in the plot of the maximum. The general trend of survival time of the scenarios is to increase over time. This indicates that there is less variability in the possible plans as time goes on, which also points to how important early decisions are. This is more evidence that stochastic information is of clear benefit for dynamic vehicle routing.

## 7. Related Work

There are very few papers in which stochastic information is used to solve dynamic vehicle routing problems. In general, papers focus either on static stochastic problems or on dynamic problems.

Most research on stochastic vehicle routing minimizes the expected travel distance. Generally, a simple recourse function (i.e., returning to the depot) is available during execution when feasibility constraints are violated. The recourse function adds some cost to the expectation. (See Bertsimas 1992 and Laporte et al. 1992 for some early work and Gendreau et al. 1996b for an overview of the various models and approaches.) More recently, Yang et al. (2000) studied the standard stochastic demand problem and adds the ability to preemptively return to the depot to unload capacity if the expectation makes it a better choice than a likely forced return later. Gendreau et al. (1995) approach the problem with stochastic demand and stochastic customers and solve the expectation optimally using linear programming techniques. Gendreau et al. (1996a) use the same model but use tabu search to solve larger instances of the problem. The recourse function in these problems is to return to the depot for unloading whenever capacity is violated. Novaes and Gracioli (1999) treat the problem as a space partitioning problem. Each vehicle is assigned regions of space, and the goal is to create the regions such that the probability of failing to service all customers in the regions is below some

threshold. Secomandi (2000) formulates the problem as a stochastic dynamic program and uses a reinforcement learning to approximate good solutions.

Most research in dynamic vehicle routing ignores stochastic information. (See Psaraftis 1995 for a survey of dynamic vehicle routing.) The most relevant work is probably the parallel tabu-search algorithm of Gendreau et al. (1999). This algorithm is based on an adaptive memory that stores potential solutions. As new customer requests arrive, the algorithm uses those solutions that allow service of the customer. The algorithm may violate time windows but there is a penalty incorporated into the objective function for late arrivals. As a consequence, feasibility is greatly simplified, but the objective function is more complex. The algorithm does not exploit stochastic information and uses travel time to choose between different solutions. Thanks to the use of the adaptive memory and the maintenance of multiple solutions, the algorithm significantly outperforms the greedy approach on moderately dynamic problems. It motivated the design and implementation of our MPA approach, which can be seen as a generalization and abstraction of this algorithm by isolating the component for searching solutions and the ranking function. See also Gendreau et al. (1998), who apply similar ideas to a dynamic pick and delivery problem and suggest adding historical/stochastic information as an avenue of future research. The other most relevant work is the excellent paper by Larsen et al. (2002), which defines the notion of “degree of dynamism,” i.e., percentage of dynamic customers, and the notion of “effective degree of dynamism,” which captures the lateness of the dynamic customers, as well as its generalization to time windows (see also Larsen 2000). Our experimental setting is precisely based on these critical concepts. The paper also studies a dynamic problem where service time is an independent random variable. It compares several heuristics: first come/first serve, stochastic queue median, nearest neighbor, and partitioning. The nearest neighbor heuristic, which is essentially our greedy heuristic, performs the best. Observe that the paper also indicates that little work has been done on incorporating stochastic information in a dynamic setting. The work by Krumke et al. (2002) is another interesting paper, which addresses a large (1,600 vehicles) real-world problem. The application is concerned with a large automobile club that provides service for club customers whose vehicles break down. The goal is to minimize a cost function which is a linear combination of operational costs (service, driving, overtime, contractors) and lateness costs with respect to soft time windows. The authors study how to optimize the objective functions given the known information (the approach can thus be viewed as an MPA approach with a single plan). They propose a mixed integer program (MIP) approach based on column generation to find optimal or near optimal solutions. Note that the authors mention that the instances are sufficiently well behaved for a MIP approach to be successful, which is not the case for the Solomon benchmarks.

Bertsimas and van Ryzin (1991, 1993) report some of the earliest work in mathematically studying some models of dynamic vehicle routing. They look at a model with Poisson arrival rates for customers that are uniformly distributed in the plane. The goal is to find a policy that minimizes the total system time of an infinite horizon (where system time is, for each customer, the total time spent servicing a customer and the waiting time of a customer). The policies developed here (and a finite time model version of this) are assessed in Larsen et al. (2002). Swihart and Papastavrou (1999) use the same model as Bertsimas and van Ryzin (1991, 1993) and apply similar techniques and analysis to the pick-up and delivery problem. Gendreau et al. (2001) present an algorithm for ambulance relocation, which consists of two subproblems. The first subproblem is to send an ambulance when a call comes in, and the second problem is to relocate ambulances such that areas of the region are covered by at least two ambulances. The objective function is to minimize the region that is not double-covered and the cost of relocation to double-cover more of the region. Ichoua et al. (2000) explore the benefits of allowing vehicles to change their destinations (diversion) when traveling. Savelsbergh and Sol (1998) use integer programming techniques to optimize the current state of information, while Burckert et al. (2000) decompose every entity of the problem (i.e., vehicles, customers) into interactive agents working together to satisfy the current state of information. Zhu and Ong (2000) present another agent-driven approach to react to traffic jams. There has also been some work in proving competitive-ratio bounds for various versions of the vehicle routing problem. These include work by Ascheuer et al. (2000) for finding an optimal 2-competitive algorithm for the online single vehicle dial-a-ride problem that minimizes the completion time. Other work on this problem has been performed by Hauptmeier et al. (2000).

As mentioned, very little work uses stochastic information for dynamic vehicle routing. A notable exception is Secomandi (2001), who considers stochastic demands and only one vehicle. All customers are known, and there are no time windows. They apply a rollout algorithm to approximate good solutions every time new information becomes available.

Finally, observe that in the related field of resource allocation and assignment problems, there has been some recent work to incorporate stochastic and forecasted information into dynamic models (e.g., Powell 1996, Powell et al. 2001). Their work and our research both confirm the usefulness and applicability of utilizing stochastic information.

## 8. Conclusion

This paper proposed a multiple scenario approach (MSA) to dynamic stochastic vehicle routing with time windows. MSA continuously solves scenarios that include both existing and future requests. Decisions use a distinguished plan selected by a consensus function. Experimental results

indicate that MSA produces dramatic improvements over approaches not using stochastic information. They also indicate a strong synergy between MSA and the consensus function, especially for problems with many stochastic customers.

There are many research avenues to explore in this area. An intriguing question, which arises when demand exceeds capacity, is to develop strategies to reject customers if such a decision is likely to improve the quality of the plan. It is also important to evaluate the effectiveness of MSA on unconstrained or loosely constrained dynamic vehicle routing problems (e.g., Larsen 2000) because they have a fundamentally different structure. Similarly, generalizing the algorithm for additional stochastic data (e.g., demands, service times) is important in practice. Finally, as mentioned, the solutions presented in this paper are essentially domain-independent, and we expect that many other dynamic applications may benefit from them.

## Acknowledgments

The authors thank Eli Upfal for his help in designing a meaningful experimental setting and the reviewers and editors for insightful comments that improved the presentation of this paper. Russell Bent is supported by NDSEG Fellowship from the American Society of Engineering Education. Pascal Van Hentenryck is partially supported by NSF ITR Award DMI-0121495.

## References

- Ascheuer, N., S. Krumke, J. Rambau. 2000. Online dial-a-ride problems: Minimizing the completion time. *STACS: 17th Annual Sympos. Theoret. Aspects Comput. Sci.* 639–650.
- Bent, R., P. Van Hentenryck. 2004. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Sci.* Forthcoming.
- Bertsimas, D. 1992. A vehicle routing problem with stochastic demand. *Oper. Res.* **40** 574–585.
- Bertsimas, D., G. van Ryzin. 1991. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Oper. Res.* **39** 601–615.
- Bertsimas, D., G. van Ryzin. 1993. Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. *Oper. Res.* **41** 60–76.
- Bianchi, L. 2000. Notes of dynamic vehicle routing—The state of the art. Technical report IDSIA-05-01, Manno-Lugano, Switzerland.
- Burckert, H., K. Fischer, G. Vierke. 2000. Holonic transport scheduling with Teletruck. *Appl. Artificial Intelligence* **14** 697–725.
- Gendreau, M., J. Potvin. 1997. Dynamic vehicle routing and dispatching. Technical report CRT 97-38, Centre de Recherche sur les Transport, Université de Montreal, Montreal, Quebec, Canada.
- Gendreau, M., G. Laporte, R. Seguin. 1995. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Sci.* **29** 143–155.
- Gendreau, M., G. Laporte, R. Seguin. 1996a. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Oper. Res.* **44** 469–477.
- Gendreau, M., G. Laporte, R. Seguin. 1996b. Stochastic vehicle routing. *Eur. J. Oper. Res.* **88** 3–12.
- Gendreau, M., G. Laporte, F. Semet. 2001. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Comput.* **27** 1641–1653.
- Gendreau, M., F. Guertin, J. Potvin, R. Seguin. 1998. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Technical report CRT-98-10, Centre de Recherche sur les Transport, Université de Montreal, Montreal, Quebec, Canada.
- Gendreau, M., F. Guertin, J. Y. Potvin, E. Taillard. 1999. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Sci.* **33**(4) 381–390.
- Hauptmeier, D., S. Krumke, J. Rambau. 2000. The online dial-a-ride problem under reasonable load. *Lecture Notes Comput. Sci.* **1767** 125–136.
- Ichoua, S., M. Gendreau, J. Y. Potvin. 2000. Diversion issues in real-time vehicle dispatching. *Transportation Sci.* **34** 426–438.
- Kilby, P., P. Prosser, P. Shaw. 1999. Dynamic VRPs: A study of scenarios. Technical report APES-06-1998, University of St. Andrews, St. Andrews, Scotland.
- Kindervater, G., M. Savelsbergh. 1997. Vehicle routing: Handling edge exchanges. E. Aarts, J. K. Lenstra, eds. *Local Search in Combinatorial Optimization*. John Wiley and Sons, New York, 337–360.
- Krumke, S., J. Rambau, L. Torres. 2002. Real-time dispatching of guided and unguided automobile service units with soft time windows. *Proc. 10th Annual Eur. Sympos. on Algorithms*, Rome Italy, 637–648.
- Laporte, G., F. Louveaux, H. Mercure. 1992. The vehicle routing problem with stochastic travel times. *Transportation Sci.* **26** 161–170.
- Larsen, A. 2000. The dynamic vehicle routing problem. Ph.D. thesis, Technical University of Denmark, Kongens Lyngby, Denmark.
- Larsen, A., O. Madsen, M. Solomon. 2002. Partially dynamic vehicle routing—Models and algorithms. *J. Oper. Res. Soc.* **53** 638–646.
- Lund, K., O. Madsen, J. Rygaard. 1996. Vehicle routing problems with varying degrees of dynamism. Technical report, IMM. Department of Mathematical Modeling, Technical University of Denmark, Kogens, Lyngby, Denmark.
- Novaes, A., O. Gracioli. 1999. Designing multiple-vehicle delivery tours in a grid-cell format. *Eur. J. Oper. Res.* **119** 613–634.
- Powell, W. 1996. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Sci.* **30** 195–219.
- Powell, W., J. Shapiro, H. Simao. 2001. A representational paradigm for dynamic resource transformation problems. *Ann. Oper. Res.* **104** 231–279.
- Psaraftis, H. 1995. Dynamic vehicle routing: Status and prospects. *Ann. Oper. Res.* **61** 143–164.
- Savelsbergh, M., M. Sol. 1998. DRIVE: Dynamic routing of independent vehicles. *Oper. Res.* **46** 474–490.
- Secomandi, N. 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Comput. Oper. Res.* **27** 1201–1225.
- Secomandi, N. 2001. A rollout policy for the vehicle routing problem with stochastic demands. *Oper. Res.* **49** 796–802.
- Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35**(2) 254–265.
- Stefik, M. 1981. Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence* **16** 111–139.
- Swihart, M., J. Papastavrou. 1999. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *Eur. J. Oper. Res.* **114** 447–464.
- Yang, W., K. Mathur, R. Ballou. 2000. Stochastic vehicle routing problem with restocking. *Transportation Sci.* **34** 99–112.
- Zhu, K., K. Ong. 2000. A reactive method for real time dynamic vehicle routing problem. *12th ICTAI*, Vancouver, British Columbia, Canada.

Copyright 2004, by INFORMS, all rights reserved. Copyright of Operations Research is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.