Dynamic Vehicle Routing with Stochastic Requests

Russell Bent and Pascal Van Hentenryck
Brown University, Box 1910
Providence, RI 02912

Abstract

This paper considers vehicle routing problems (VRP) where customer locations and service times are random variables which are realized dynamically during plan execution. It studies a multiple scenario approach (MSA) which continuously generates plans consistent with past decisions and anticipating future requests, and compares it with the best available heuristics on dynamic VRP problems that model long-distance courier mail services [13]. In addition, it proposes a least-commitment refinement of MSA (MSA-LC), which also uses stochastic information to delay vehicle departures opportunistically. Experimental results shows that MSA, and MSA-LC in particular, may significantly decrease travel times (while not degrading service) and is robust with respect to reasonably noisy distributions.

1 Introduction

The vehicle routing problem (VRP) is a difficult combinatorial optimization problem with many important applications in distribution and transportation systems. It has received considerable attention for many years and sophisticated local search methods are quite effective at finding good quality solutions in reasonable amounts of time. In more recent years, technology has advanced so that it is now possible and practical to address dynamic and/or stochastic versions of the problem. These new versions are motivated by the inherent uncertainties that arise in many everyday VRPs and advances in onboard computers and communications systems that allow modification of routing plans even after vehicles are deployed.

Most of the existing work has focused solely on stochastic or dynamic versions of the problem exclusively. In stochastic optimization, the expected cost of a solution is optimized with respect to a recourse function which restores feasibility during plan execution. In dynamic optimization, various data items, such as customer requests, are unknown and are only revealed after some decisions are taken. Techniques here tend to focus on optimizing with respect to the current state of information (e.g. [9]). How to combine the two approaches is a research topic that is often mentioned (e.g., [4, 7, 9, 12]).

This paper considers the multiple scenario approach (MSA) recently proposed in [2]. Its key idea is to continuously generate plans that are consistent with past decisions but anticipate future requests by sampling the distributions of the stochastic variables. At every execution step, the algorithm makes decisions according to a distinguished plan which is selected from the current pool by a ranking function. A preliminary version of MSA [2] was shown to be effective on highly-constrained dynamic vehicle routing with time windows, where each customer had a certain probability of making a request at its location within its time window. On these problems (which closely model practical applications [6]), MSA was shown to improve customer service by missing significantly fewer customer requests. It was also shown that a consensus function was most appropriate to rank the scenarios.

The goal of this paper is to study the behavior of MSA on loosely constrained problems derived from the models in [13] to capture long-distance courier mail services. The problems have a fundamentally different structure. On the one hand, they are more stochastic, since customer, customer locations, and service times are now random variables. On the other hand, the main focus is on the objective function, i.e., minimizing travel distance, since the problems are unconstrained or loosely constrained. In contrast, the time windows in [2] makes the problems tightly constrained and customer service, not travel distance, becomes the main focus. In addition, the paper proposes a least-commitment refinement of MSA (MSA-LC) which uses stochastic information to delay vehicle departures opportunistically in order to accommodate new requests "predicted" by the stochastic model.

The experimental results, which compare several approaches, are particularly interesting and contain a few surprises on competing approaches. They indicate that MSA approaches typically service the same number of customers as earlier approaches but may reduce travel distance significantly. In addtition, the travel distance reduction may be particularly drastic when MSA-LC is used. The experimental results also indicate that MSA is robust with respect to reasonably noisy distribution (i.e., when the actual data does not follow the assumed distribution) and that it is beneficial to be opportunistic. The contributions of this paper are threefold.

1. It shows that MSA is an effective approach to exploit stochastic information in loosely-constrained dynamic vehicle routing applications. Since it also provided significant benefits for highly constrained problems [2], the new results indicate the generality and versatility of MSA. Interestingly, MSA contributions are radically different in the two settings. While MSA improves service on highly constrained problems, it improves travel distances (without degrading service) on loosely constrained problems. The results also indicate the benefits of sophisticated optimization techniques.

2. It presents a least-commitment refinement of MSA (MSA-LC) which produces significant advantages over the standard MSA for loosely constrained problems by reducing travel distance dramatically.

3. It indicates that MSA is robust in noisy environments and that it is beneficial to be optimistic in estimating stochastic information.

2 Problem Formulation

We briefly describe the general setting and notations for the dynamic vehicle routing problems considered in the paper. The specifics of the actual problems are presented in the section on experimental results.

Each problem contains N customers, numbered $1 \ldots, N$. The depot is represented by the number 0. The travel cost between sites i and j is represented by c_{ij}. The travel costs satisfy the triangle inequality, i.e. $\forall_{i,j,k}(c_{ik} \leq c_{ij} + c_{jk})$. Requests have a demand $q_i \geq 0$ and a service time $s_i \geq 0$. Each problem has up to $m \geq 1$ identical vehicles available for use, with capacity $Q \geq 0$.

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 $[0, v_1, \ldots, v_n, 0]$, where $1 \leq v_i \leq N$ and all v_i

are distinct. For a customer, c, the route of that customer is denoted by $Route(c)$. The demand of a route is denoted by $q(r) = \sum_{i=1}^{n} q_i$. The travel cost of a route r_i is denoted by $c(r)$ and is the cost of visiting all of its customers, i.e. $c(r) = c_{0v_1} + c_{v_1 v_2} + \ldots + c_{v_{n-1} v_n} + c_{v_n 0}$.

A routing plan, or plan, is a set of routes $\{r_1, \ldots, r_m\}$ servicing each customer exactly once. A routing plan assigns a unique successor and predecessor for each customer. For a plan, σ, the successor of customer i is denoted by $succ(i, \sigma)$. The travel cost of a plan is denoted by $c(\sigma)$, i.e. $c(\sigma) = \sum_{r=1}^{m} c(r)$. The depot has a deadline, l_0, which represents the latest time a vehicle can return. A routing plan σ implicitly specifies, for each customer c, an earliest departure time, denoted by $\text{EDT}(\sigma, c)$, and a latest departure time, denoted by $\text{LDT}(\sigma, c)$, to meet the deadline. Once specific times are chosen, the routing plan also specifies a return time $a(r)$ for each route r.

A solution to the problem is a routing plan $\sigma = \{r_0, \ldots, r_m\}$ that satisfies the capacity and deadline constraints., i.e., $q(r_i) \leq Q$ & $a(r_i) \leq l_0$ for all $1 \leq i \leq m$. The objective function lexigraphically maximizes the number of serviced customers and then minimizes $c(\sigma)$.

In the dynamic VRP, a number of requests are available initially, while others become available during the plan execution. In the applications considered in this paper, a request consists of the location of a customer and a service time, both of which are random variables. We assume that the distribution of the requests, or some approximation thereof, is available, which is typically the case in practical applications. For each incoming request, a dynamic algorithm must decide whether to accept or reject it. Once a request is accepted, it must be serviced. Problems are generally characterized by their degree of dynamism (DOD), i.e., the ratio of *unknown customers*/*total customers*, which measures how dynamic they are [13].

3 The Multiple Plan Approach

The Multiple Plan Approach (MPA) is a fundamental generalization of many modern approaches. Its key idea is to maintain a set of plans at every execution step. MPA was motivated by the work of [9] which proposes a parallel tabu search algorithm organized around multiple solutions in an adaptive memory. MPA here generalizes their approach by making it independent of the search procedure. In short, MPA continuously generates plans that are compatible with the current state of information and removes those that are not. In addition, since decisions must be made with respect to a specific plan to guarantee service, a *distinguished* plan is maintained via a ranking function. We will discuss possible ranking functions later in the section.

More precisely, MPA handles four types of events (1) customer requests, (2) vehicle departures, (3) plan generations, (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 can occur if the *distinguished* plan indicates a return to depot, causing the vehicle to wait, and another plan that sends the vehicle to a customer becomes infeasible during the wait. Timeouts capture these events.

At each time t, MPA maintains a set of plans, S_t, and a distinguished plan, σ_t^*. For each event, we specify how to compute S_{t+1} and σ_{t+1}^* from past decisions, S_t and σ_t^*. Each event 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 generation, 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 σ, $f_t(\sigma)$ returns a real value. Finally, observe that the implementation is event-driven. In other words, although we specify S_t and σ_t^* for all t, the implementation only considers the times where an actual event occurs.

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

$$
\begin{array}{l}
F := \{\text{INSERT}(\sigma, c) \mid \sigma \in S_t \ \& \ \text{FEASIBLEINSERT}(\sigma, c)\}; \\
\text{if } F \neq \emptyset \text{ then} \\
\quad S_{t+1} := F; \\
\quad \sigma_{t+1}^* := argmax(\sigma \in S_{t+1}) f_{t+1}(\sigma); \\
\text{else} \\
\quad S_{t+1} := S_t; \\
\quad \sigma_{t+1}^* := \sigma_t^*;
\end{array}
$$

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

Vehicle Departure When plan σ^* specifies that vehicle v must depart from customer c, it is necessary to remove all plans in S_t that are incompatible with this departure.

$$
\begin{array}{l}
S_{t+1} := \{\sigma \in S_t \mid \text{COMPATIBLE}(\sigma, \sigma_t^*, t)\}; \\
\sigma_{t+1}^* := argmax(\sigma \in S_{t+1}) f_{t+1}(\sigma);
\end{array}
$$

$\text{COMPATIBLE}(\sigma, \sigma_t^*, t)$ is true if σ is compatible with σ_t^* up to time t. More formally, $\text{COMPATIBLE}(\sigma, \sigma_t^*, t)$ holds iff $\forall c \in \text{DEPART}(t) : succ(\sigma_t^*, c) = succ(\sigma, c)$, where $\text{DEPART}(t)$ denotes the set of customers from which a vehicle departed before or at time t,

It is important to specify how the departure times are computed in MPA. The key idea is to delay the return of the vehicle as long as possible in order to accommodate the new requests. This leads to the following definition of departure times

$$
\text{DT}(\sigma, s) = \begin{array}{ll} l_0 - c_{s0} & \text{IF } succ(\sigma, s) = 0; \\ \text{EDT}(\sigma, s) & \text{OTHERWISE}. \end{array}
$$

which are also recomputed continously using the distinguished plan.

Plan Generation When a new plan, σ, is generated a 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 *distinguished* plan, since they include all existing decisions and plan generation is canceled whenever customer requests and vehicle departures occur.

$$
\begin{array}{l}
S_{t+1} := S_t \cup \{\sigma\}; \\
\sigma_{t+1}^* := argmax(\sigma \in S_{t+1}) f_t(\sigma);
\end{array}
$$

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

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

where $\text{FEASIBLE}(\sigma, t)$ holds if σ is feasible at time t: $\forall c \in \text{DEPART}(t) : \text{LDT}(\sigma, succ(\sigma, c)) \leq t$.

4 The Multiple Scenario Approach

The Multiple Scenario Approach (MSA) generalizes MPA by considering both existing and potential future requests during plan generation. Future requests are obtained by sampling their distributions. Once a routing plan σ is discovered, MSA stores the routing plan σ^- obtained by removing future requests from σ. As a result, plan σ^- leaves room to accommodate future requests, should they actually materialize. This ability to anticipate the future is the strength of MSA. Note that event handling in MSA is similar to MPA.

5 Ranking Functions

Both MPA and MSA are parametrized by a ranking function f_t, which selects the *distinguished* plan at each time t. We will evaluate two ranking functions for nominating σ^* in this paper. The obvious first choice for f_t would be to select the plan with the smallest travel cost (algorithms MPA^d and MSA^d). In [2], it was shown that it is possible to do substantially better in practice on highly-constrained problems by using a consensus function which selects the plan in S_t that most resembles all the plans in S_t. Since the resulting plans do not depart from other plans too dramatically, the consensus function may be also viewed as a least commitment strategy [19]. More precisely, at each time t, the algorithm maintains a two-dimensional matrix M_t where $M_t[v, c]$ denotes the number of plans in S_t where vehicle v departs for customer c next. More formally, the matrix M_t is defined as $M_t[v, c] = \#\{\sigma \in S_t \mid succ(\sigma, \text{LDC}(v)) = c\}$, where $\text{LDC}(v)$ is the last customer from which a vehicle departed in the plan execution. The consensus function f_t is then defined as $f_t(\sigma) = \sum_{v=1}^{m} M_t[v, succ(\sigma, \text{LDC}(v))]$ (algorithms MPA^c and MSA^c).

6 The Multiple Scenario Approach With Least Commitment

The MSA approach uses stochastic information to find a routing plan of the known requests which is more likely to accommodate future requests easily. However, the stochastic information is not used to deduce vehicle departure times, which are computed only from known requests. A close analysis of the behavior of MSA on loosely constrained problems indicated that the plans had a tendency to commit too quickly to the next customer and that it could be beneficial to delay vehicle departures in order to reduce travel distance.

MSA-LC is a refinement of MSA whose main motivation is to use stochastic information to suggest more "flexible" departure times. MSA-LC keeps pairs of plans $\langle \sigma^-, \sigma \rangle$, where σ, also called the sampled plan, contains both known and sampled customers and σ^- is the projection of σ on known customers. The event handlers are essentially unchanged, since they all work in terms of projected plans. The main novelty is in using the complete plans to choose the departure times. Informally speaking, the key idea is to delay a vehicle departure at customer i as long as there are sampled customers between i and its *known* successor. By delaying the departure, the plan is more likely to accommodate such requests, should they materialize. The sampled customers will either be replaced by actual requests or they will not materialize and hence will be removed from the plan. We now specify the departure times and the adaptions of the event handlers more formally. Although the event-handlers work in terms of projected plans, they also need to update the complete plans to remove and/or replace sampled customers.

Departure Times The departure time for a customer i and a pair of plans $\langle \sigma^-, \sigma \rangle$ are specified as follows:

$$\textsc{dt}(\sigma^-, \sigma, i) = \begin{array}{ll} l_0 - c_{i0} & \textsc{if } succ(\sigma^-, i) = 0; \\ \min(\textsc{ldt}(\sigma^-, i), \textsc{edt}(\sigma, pred(\sigma, succ(\sigma^-, i)))) & \textsc{otherwise}. \end{array}$$

The second term $\textsc{edt}(\sigma, pred(\sigma, succ(\sigma^-, i)))$ captures the least-commitment approach: it specifies that the vehicle should not depart from customer i until all sampled customers between i and its known successor have disappeared, since this term represents the departure time of the last sampled customer. The first term $\textsc{ldt}(\sigma^-, i)$ is necessary to ensure the feasibility of the plan.

Customer Request As mentioned earlier, the event-handlers are similar in MSA-LC as in MPA and MSA. However, it is necessary to specify how the complete plans are updated as well, since they are used to compute the departure times. For the feasible insertion of a customer c, the intuition is to update the sampled plan by substituting c in place of its nearest sampled customer. More formally, given a pair of plans $\langle \sigma^-, \sigma \rangle$ and a new request c such that $\textsc{feasibleInsert}(\sigma, c)$ holds, MSA-LC computes a new pair $\langle \sigma_n^-, \sigma_n \rangle$, where

$$\begin{array}{rcl} \sigma_n^- & = & \textsc{insert}(\sigma^-, c) \\ \sigma_n & = & \textsc{replaceNearest}(\sigma, pred(c, \sigma_n^-), succ(c, \sigma_n^-), c) \end{array}$$

where $\textsc{replaceNearest}(\sigma, p, s, c)$ returns a plan σ' where c replaces its nearest sampled customer scheduled between customers p and s or insert c between customers p and s if no such customer exists. It is important to note that the resulting plan σ_n is not necessarily feasible, although the operation is only applied to projected plans σ^- that can accommodate c. Restoring feasibility may require removing some sampled customers as in the timeout handler described next.

Timeout At time t, some plans may become infeasible as before. Moreover, some sampled customers may not be able to be serviced any longer. This can occur if the departure time of a sampled customer passes. The set of pairs that are kept at time t are given by the instruction

$$\{\langle \sigma^-, \textsc{filter}(\sigma, t) \rangle \in S_t \mid \textsc{feasible}(\sigma, t)\}$$

where $\textsc{filter}(\sigma,t)$ returns the plan where all sampled customers i such that $\textsc{edt}(\sigma, i) \leq t$ have been removed.

Ranking Function The consensus function for MSA-LC is slightly different. M_t contains an additional column for the samples. The additional column, $M_t[v, N+1]$ counts the number of plans in S_t where vehicle v's is waiting, i.e., its next known customer is preceded by sampled customer.

7 Optimization

It remains to specify how MPA, MSA, and MSA-LC optimize plans for our applications. Both of them use large neighborhood search (LNS), which has been shown to be very effective for vehicle routing [1, 18]. LNS combines the advantages of branch and bound, constraint propagation, and local search. Its key idea is to remove c customers from the best-known solution and to reinsert them to produce a better solution. The customers are reinserted according to a heuristic and, in general, limited discrepancy search [10] is used to bound the number of times the heuristic can be violated. The c customers are removed randomly using a relatedness function, which is parametrized by a determinism factor. When LNS is not able to improve the best solution by removing c customers for some iterations, it increases the number of customers to remove to $c + 1$. In addition to using LNS, we also use a nearest neighbor heuristics as in [13]. This makes it possible to see the effect of using more sampling on a less sophisticated optimization procedure.

8 Experimental Results

We now report some experimental results on a variety of models and discuss the robustness of MSA when the stochastic data is noisy.

8.1 The Models

The starting point of this research was the experimental model in [13], where customers are uniformly distributed in a 10km×10km region and must be served by a single vehicle with uniform speed of 40 km/h. Service times for the customers are generated according to a log-normal distribution with parameters (.8777, .6647). With this distribution, the mean service time is 3 min. and the variance is 5 min. The service times were chosen to mimic the service times of long-distance courier mail services [13]. We use n to denote the expected number of customers and T to denote the time horizon which is 8 hours. Problems are generated with a degree of dynamism (DOD) (i.e, the ratio of known customers over stochastic customers) in the set $\{0\%, 5\%, \ldots, 100\%\}$. For a DOD x, there are $n(1 - x)$ known customers. The remaining customers are generated using an exponential distribution with parameter $\lambda = \frac{nx}{T}$ for their inter-arrival times. It follows from the corresponding Poisson distribution (with parameter λT) that the expected number of unknown customers is nx, the expected number of customers is n, and the expected DOD is x. We generated 15 instances for each configuration, which gives about 315 problems for each model to be described. All results are the average of 5 runs on each instance.

Several models are used to evaluate the various approaches. The first two models, M1 and M2, use a single vehicle. Model M1 is the basic model with 40 customers (all other parameters have been given already). Model M2 is similar to M1, except that the region is now 40km×40km. The objective function consists of minimizing the travel distance, since these are unconstrained models. MSA-LC is not applied to these models, as the optimal strategy simply waits until all customers have requested service before deploying the vehicle. Thus, there is an implicit constraint that forces the vehicle to service any unserviced customer, so MSA-LC cannot be applied here.

The next two models are multiple-vehicle models. Model M3 is the basic model with 4 vehicles, 160 customers, and a 20km×20km. Each vehicle can serve at most 50 customers and the vehicle must return to

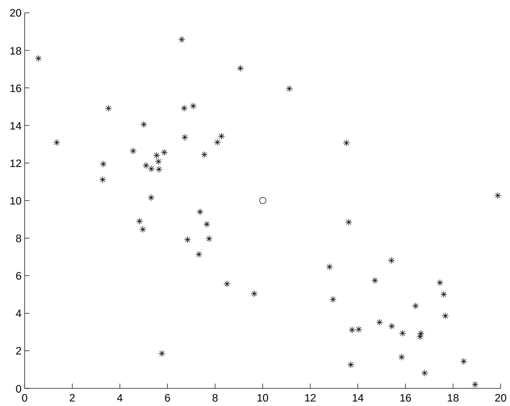

Figure 1: Example of M4 Customers

the depot by the time horizon. Model M4 is similar to M3, except that the customers are generated using 2-D Gaussians centered at two points in the region. For the multiple-vehicle models, the objective function consists in minimizing the number of missed customers and minimizing the travel distance. Such an example may more accurately model what could be seen in real life. Figure 1 shows an instance of this model. It is possible that some customers be left unserviced, since models M3 and M4 have capacity constraints, as well as a hard deadline.

8.2 Setting of the Algorithms

We now describe the configuration of our algorithms used to obtain the experimental results. Initially, 25 different scenarios are created and optimized by our large-scale neighborhood search (LNS) for 1 minute. These initial solutions are used to determine the first customer for each vehicle. An additional 25 scenarios are created and optimized for 1 minute with the first customers fixed. It was verified experimentally that this second step improves the quality of the final solutions.

Subsequent scenarios are optimized for about 10 seconds using LNS. The parameters for LNS are as follows: 30 for the maximum number of customers to remove at one time, 100 attempts at removing c customers without improvement before removing $c + 1$ customers, 15 for the determinism factor of the relatedness function, and 4 discrepancies.

8.3 Single Vehicle Results

Reference [13] tested various heuristics on Model M1 and reported extensive experimental results. Their best heuristic is nearest neighbor (NN), where a pool of unserviced customers is maintained and the vehicle is sent to the nearest customer in the pool once it served its current request. Interestingly, in Model M1, the vehicle is able to service customers faster than they arrive. As a consequence, all "reasonable" heuristics converge towards a first come, first serve (FCFS) strategy as the DOD converges to 100%. This same behavior is also exhibited by our approaches. LNS optimization has some benefits for low DODs. Overall, exploiting stochastic information provides only marginal benefits in Model M1. In Model M2, which has

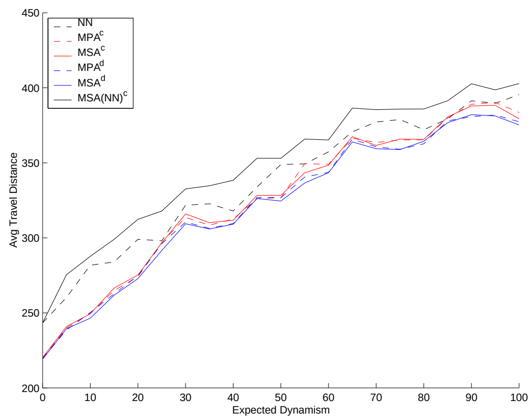

Figure 2: Experimental Results on M2

a larger region, the vehicle is not able to service customers as quickly and the heuristics do not converge to FCFS as quickly. Figure 2 depicts the experimental results. They indicate that the MPA and MSA approaches may bring significant benefits, especially for low DODs. In general, MSA approaches are slightly superior to MPA approaches. Only MSA(NN)^c is not effective, indicating the value of optimization on these problems.

8.4 Multiple Vehicle Results

We now turn to the multiple vehicle models M3 and M4. Recall that these models have capacity and deadline constraints and that the objective function consists of first minimizing the number of missed customers and then the travel distance. The NN heuristics was generalized to providing guarantees on servicing customers. Whenever a request arrives, the NN algorithm is simulated to determine if it can accommodate the new request. If it cannot, the request is rejected.

Figure 3 describes the experimental results concerning the number of serviced customers for various degrees of dynamism. The results clearly indicate that the MSA approaches are superior to MPA, as MPA is unable to service as many customers. A detailed look at the trace of the decisions performed by MPA indicate that MPA waits too long to deploy some of the vehicles. This is because optimal solutions use as few vehicles as possible to minimize travel distance and MPA believes it can use fewer vehicles than necessary until late in the simulation. The remaining approaches service a comparable number of customers when compared to the best available heuristic. With higher degrees of dynamism, the benefits of using a consensus function for ranking are clear, as it reduces the number of missed customers significantly compared to using travel distance. MSA approaches do not bring significant benefits in terms of serviced customers. MSA(NN) is generally superior to NN, while MSA^c is roughly similar to NN (except for very high degrees of dynamism). Note that MSA-LC does not perform as well as MSA for these very high degrees of dynamism.

Figure 4 depicts the results for the travel distance. No results are given for the MPA approaches, since they are far from being competitive for customer service. The results indicate that MSA significantly reduces travel distance compared to NN. The results are particularly impressive for MSA-LC, whose travel distance is essentially not affected by the degree of dynamism (results above 80% are not shown because its quality of

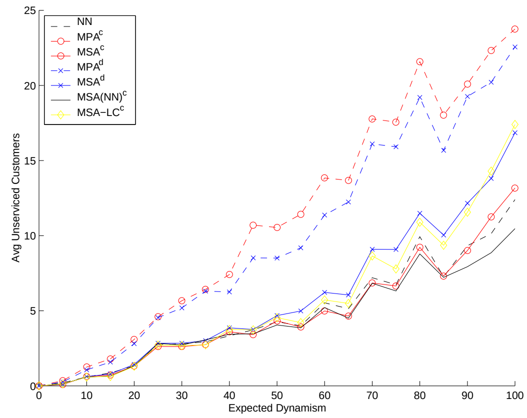

Figure 3: Serviced Customer Results for M3

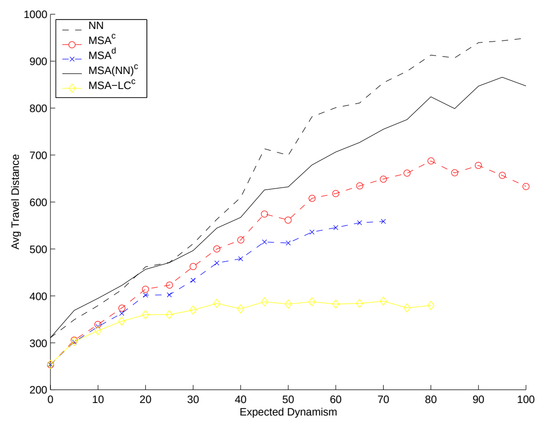

Figure 4: Travel Distance Results for M3

service is not comparable to other MSA approaches). Observe also that the comparison between MSA(NN) and the other MSA approaches tend to indicate that it seems more beneficial for these problems to use a more sophisticated optimization algorithms on fewer samples than a weaker method on more samples.

Figures 5 and 6 show the same results under the M4 model. These figures confirm much of what was seen in model M3. An interesting aspect is that the gap in unserviced customers between MSA-LC^c and the other approaches has narrowed, while the large gap in travel distance was preserved. This seems to indicate that the value of using this approach increases under more varied (and perhaps more realistic) customer distributions.

8.5 Robustness

It is natural to question how MSA behaves when the stochastic information is not entirely accurate. This situation could arise from faulty historical data, predictions, and/or approximations in machine learning

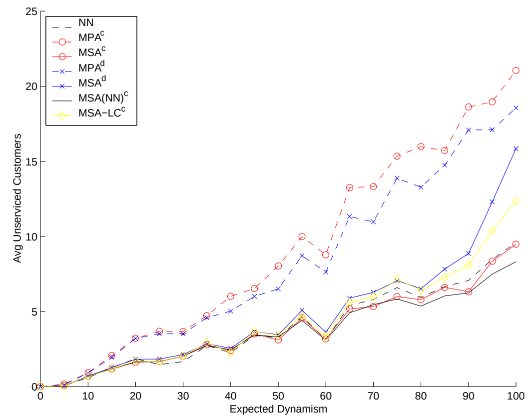

Figure 5: Serviced Customer Results for M4

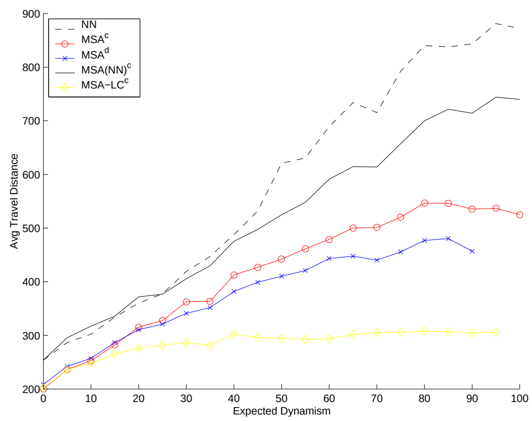

Figure 6: Travel Distance Results for M4

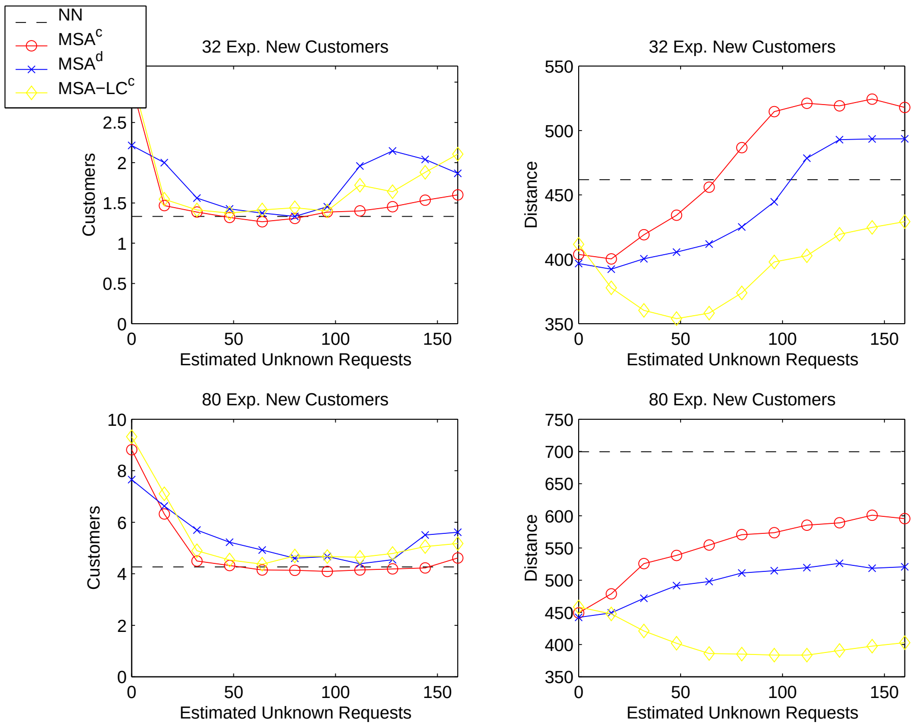

Figure 7: Robustness Results

algorithms. Figure 7 shows some results when run on the 20% and 50% dynamism instances of M3 (32 and 80 expected new customers respectively). It is interesting to see that, in both cases, it is better to be optimistic when estimating the number of dynamic customers. For example, on 20% dynamism, MSA-LC^c is able to service roughly the same number of customers when it expects between 20 and 100 dynamic customers. However, it performs the best in terms of travel distance when it expects 50 dynamic customers, slightly more than the 32 of actual problem sets themselves. In addition, these results show that, even in the presence of significant amounts of noise, MSA approaches are able to still achieve good results.

9 Related Work

There are a limited number of papers that incorporate stochastic information into dynamic VRPs. In general, most papers focus on one or the other exclusively. The research on stochastic VRPs focuses on minimizing the expected travel distance with a simple recourse function (i.e. returning to the depot) when feasibility is violated. See [8] for a general overview of the models and approaches considered. More recently [20] has looked at preemptively using the recourse function before constraints are violated. There is also some work in approximating solutions using reinforcement learning techniques in [16]. The research in dynamic VRP tends to mostly ignore any stochastic or historical data that may be available. The most relevant work occurs in [9] which pioneers the approach of keeping many potential solutions available for incorporating new requests. The models used in this paper are inspired by [13] which evaluates many simple heuristics against different objectives, like minimizing travel distance or customer wait time. A survey on other work in the field can be found in [14]. A sample of more recent work includes [5], [11], and [15].

As mentioned previously, very little work has addressed incorporating stochastic information into dynamic vehicle routing, though it has received more attention in recent years. [17] considers a problem with stochastic demands and a single vehicle. The customers are all known ahead of time and they apply a rollout algorithm to approximate expected good solutions every time new information becomes available. An interesting paper by [6] considers using stochastic information to determine whether it is profitable to accept a customer request in light of what may occur in the future. It also uses a technique resembling the MPA approach described in this paper. A preliminary version of MSA exhibited very good results, especially using the consensus function, on problems with time windows and high DOD [2]. The model in [2] is very similar to that of [6] and it would be interesting to combine the two approaches. Preliminary work for MSA on the model addressed in this paper was described in [3].

10 Conclusion

This paper considered vehicle routing problems (VRP) where customer locations and service times are random variables that are realized dynamically during plan execution. It reconsidered a multiple scenario approach (MSA) that continuously generates plans consistent with past decisions and anticipating future requests. It also proposed a new least commitment refinement of MSA (MSA-LC) that uses stochastic information to choose the vehicle departure times and to avoid committing too soon to a known request. MSA and MSA-LC were were compared to the best available heuristics on dynamic VRP problems that model long-distance courier mail services [13]. It was shown that MSA may substantially decrease travel times (while not degrading service). MSA was also shown robust when the distribution is reasonably noisy, in which case it seems beneficial to be optimistic in predicting customer requests. The results also show the value of optimization inside a MSA framework.

There are several interesting directions for future work. On the one hand, it seems important to incorporate stochastic information to decide whether to accept a customer request. Our current algorithms are greedy in that respect, while [6] has shown the benefits of using stochastic information on that aspect of the problem as well. On the other hand, it is worth stressing that MSA is domain-independent. As a consequence, applying these ideas to other domains is an important avenue for future research.

References

[1] R. Bent and P. Van Hentenryck. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. To appear in *Transportation Science*.

[2] R. Bent and P. Van Hentenryck. Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. To appear in *Operations Research* (pending minor revisions).

[3] R. Bent and P. Van Hentenryck. Dynamic Vehicle Routing with Stochastic Requests. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

[4] L. Bianchi. Notes of Dynamic Vehicle Routing - The State of the Art. *Technical Report IDSIA-05-01*, 2000.

[5] H. Burckert, K. Fischer, and G. Vierke. Holonic Transport Scehduling with Teletruck. *Applied Artificial Intelligence*, 14:697–725, 2000.

[6] A. Campbell and M. Savelsbergh. Decision Support for Consumer Direct Initiatives. *Report TLI-02-09, Georgia Institute of Technology*, 2002.

[7] M. Gendreau and J. Potvin. Dynamic Vehicle Routing and Dispatching. *Technical Report CRT 97-38, Centre de Recherche sur les Transport, Universite de Montreal*, 1997.

[8] M. Gendreau, G. Laporte, and R. Seguin. Stochastic Vehicle Routing. *European Journal of Operational Research*, 88:3–12, 1996.

[9] M. Gendreau, F. Guertin, J. Y. Potvin, and E. Taillard. Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33 (4):381–390, 1999.

[10] W.D. Harvey and M.L. Ginsberg. Limited Discrepancy Search. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.

[11] S. Ichoua, M. Gendreau, and J. Y. Potvin. Diversion Issues in Real-Time Vehicle Dispatching. *Transporation Science*, 34:426–438, 2000.

[12] P. Kilby, P. Prosser, and P. Shaw. Dynamic VRPs: A Study of Scenarios. *Technical Report APES-06-1998*, 1999.

[13] A. Larsen, O. Madsen, and M. Solomon. Partially Dynamic Vehicle Routing-Models and Algorithms. *Journal of Operational Research Society*, 53:637–646, 2002.

[14] H. Psaraftis. Dynamic Vehicle Routing: Status and Prospects. *Annals of Operations Research*, 61:143–164, 1995.

[15] M. Savelsbergh and M. Sol. DRIVE: Dynamic Routing of Independent Vehicles. *Operations Research*, 46:474–490, 1998.

[16] N. Secomandi. Comparing Neuro-Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. *Computers and Operations Research*, 27:1201–1225, 2000.

[17] N. Secomandi. A Rollout Policy for the Vehicle Routing Problem with Stochastic Demands. *Operations Research*, 49:796–802, 2001.

[18] P. Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *Principles and Practice of Constraint Programming*, pages 417–431, 1998.

[19] M. Stefik. Planning with Constraints MOLGEN: Part 1. *Artificial Intelligence*, 16:111–139, 1981.

[20] W. Yang, K. Mathur, and R. Ballou. Stochastic Vehicle Routing Problem with Restocking. *Transportation Science*, 34:99–112, 2000.