LA-UR-09-00015

Approved for public release;
distribution is unlimited.

Title: The Price of Commitment in Online Stochastic Vehicle Routing

Author(s): Russell Bent
Pascal Van Hentenryck

Intended for:

The Price of Commitment in Online Stochastic Vehicle Routing

Russell Bent and **Pascal Van Hentenryck**
LANL and Brown University

Abstract

This paper considers online stochastic multiple vehicle routing with time windows in which requests arrive dynamically and the goal is to maximize the number of serviced customers. Early work has focused on very flexible routing settings where the decision to assign a vehicle to a customer is delayed until a vehicle is actually deployed to the customer. Motivated by real applications that require stability in the decision making, this paper considers a setting where the decision to assign a customer request to a vehicle must be taken when that request is accepted. Experimental results suggest that this constraint severely degrades the performance of existing algorithms. However, the paper shows how the use of stochastic information for vehicle assignment and request acceptance improves decision quality considerably. Moreover, the use of resource augmentation quantifies precisely the cost of commitment in online vehicle routing.

1 Introduction

Vehicle routing with time windows is a hard combinatorial optimization problem with many important applications in distribution and transportation scheduling. It has received considerable attention in the last decades and sophisticated algorithms are now available to find near-optimal solutions in reasonable time. In recent years, attention has shifted to online and/or stochastic versions of the problem. The stochastic and online 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.

In online stochastic problems, customers arrive dynamically as the algorithm proceeds. Each customer request has a time window (possibly the entire time horizon) during which it can be served and, obviously, a request cannot be served before it occurs. Upon arrival, the algorithm must decide whether to accept or reject the request. If the request is accepted, the online algorithm must serve it before the time horizon. The online algorithm typically has two black-boxes available to make decisions: an optimization algorithm for the deterministic version of the problem and a sampling procedure to generate future requests.

Online stochastic vehicle routing were first studied in [6] using a sampling-based approach. More precisely, the idea was to generate scenarios consisting of existing and sampled customers, to solve the scenarios using large neighborhood search [17], and to make online decisions based on the scenario solutions. These resulting algorithms were then generalized and abstracted in a generic online stochastic framework [4; 7] and applied to other problems such as packet scheduling [5; 10] and reservation systems [1; 20]. The prior work in online stochastic vehicle routing has focused on experimental settings where the decision to assign a customer to a vehicle is delayed until the last possible moment, i.e., when the decision to a deploy a vehicle to a customer is taken. However, in some applications, there is strong desire to promote more stability and robustness in the decision making process, as the logistics of delaying the decision until the last possible moment may be fundamentally undesirable or not available [15; 16]. Moreover, simple adaptions of existing algorithms exhibit poor decision qualiy when vehicle commitment is required at acceptance time, questioning whether online stochastic approaches would scale for these applications. This paper addresses this challenge and make three main contributions.

1. The paper proposes novel online algorithms for online stochastic vehicle routing. The algorithms provide a unified framework by which all the three main decisions, request acceptance, vehicle assignment, and vehicle dispatch, are based on the same principles and use stochastic information. The novel algorithms produce significant gains in solution quality, demonstrating the value of stochastic information for these applications as well.
2. Unlike prior work in less restrictive settings [9], the paper shows that relocation and waiting strategies bring very little benefit.
3. The paper provides empirical quantification of the price of commitment and shows how resource augmentation [12], the idea of increasing the resources in online algorithms, may reduce or even eliminate this cost.

The rest of the paper recalls the main concepts in vehicle routing and online vehicle routing. It then briefly surveys the existing results when no vehicle commitment is necessary, before presenting the new setting and its associated algorithms and methodologies.

2 The Offline Problem

The Input Data A vehicle routing problem is specified by a number of customers that must be visited by a pool of vehicles. Each customer makes a request that must be served within a time window and takes some capacity from the vehicle. Each vehicle starts at the depot, serves some customers, and must return to the depot by the deadline.

Each problem contains a set R of n customers and a depot o. The set S of sites is thus $R \cup \{o\}$. The travel time between sites i and j is denoted by $d(i, j)$. Each request is associated with a customer and, since each customer makes at most one request per instance, we use the names customer and request interchangeably. Every request c has a capacity $q(c) \geq 0$ and a service time $p(c) \geq 0$, which is the time to serve the request once the vehicle is on site.

Each instance has a pool of m identical vehicles with capacity Q. Each vehicle starts from the depot and the algorithm may choose to deploy all of them or to use a subset of them only. Each customer c has a time window specified by an interval $[e(c), l(c)]$ satisfying $e(c) \leq l(c)$. The time window represents the earliest and latest possible arrival times of a vehicle serving customer c. In other words, the service for customer c may start as early as $e(c)$ and as late as $l(c)$. A customer c may not be served before $e(c)$ but a vehicle arriving early to serve c may wait at the site until time $e(c)$ before beginning service. The depot has a time window $H = [e_o, l_o]$, which represents the earliest departure and latest possible return for the vehicles. Typically, e_o denotes the beginning of the day and l_o is the deadline by which all vehicles must return to their depot.

Routing Plans Optimization algorithms for vehicle routing typically return a routing plan that specifies the order in which each vehicle visits its customers. The routing plan does not prescribe departure times for the vehicles but constrains because of the time windows.

A vehicle route, or route for short, starts at the depot, serves some customers, and returns to the depot. A customer appears at most once on a route. Hence a route is a sequence $\langle o, c_1, \ldots, c_n, o\rangle$, where $c_i \in R$ and all c_i are distinct. The capacity of a route ρ is the sum of its customer capacities, i.e., $q(\rho) = \sum_{i=1}^{n} q(c_i)$.

A routing plan is a tuple of routes $(\rho_1, \ldots, \rho_m)$ one for each vehicle, in which each customer appears at most once. We also use $cust(\rho)$ and $cust(\gamma)$ to denote the customers of a route ρ and a plan γ. Because a customer makes exactly one request, a routing plan assigns a unique successor and predecessor for each served customer and depot. For a plan γ, the successor of site c is denoted by $succ(c, \gamma)$ and the predecessor is denoted by $pred(c, \gamma)$. Since, in general, the discussion or definitions assume an underlying routing plan γ, we abuse notations and use c^+ and c^- to denote the successor and predecessor of c in γ.

Departure Times Routing plans do not prescribe departure times for the vehicles. These departure times are typically not uniquely defined: a vehicle may depart at different times from specific customers and still visit all its assigned customers before the deadlines. In addition to the routing plan, a solution will also consist of an assignment $\delta : R \to H$ of starting times to all customers.

The Vehicle Routing Problem We are now in position to describe the vehicle routing problem. A solution to a vehicle routing problem with time windows (VRPTW) is a routing plan $\gamma = (\rho_1, \ldots, \rho_m)$ and a starting time assignment δ satisfying the capacity and time window constraints, i.e.,

$$C(\gamma) \equiv \begin{cases} q(\rho_j) \leq Q \\ \delta(c) - p(c) \leq l(c) \\ \delta(c) \geq \max(e(c), \delta(c^-) + d(c^-, c)) + p(c) \\ \delta(c) + d(c, c^+) \leq l(o) \end{cases}$$

for $1 \leq j \leq m$ and $c \in cust(\gamma)$. The objective is to find a solution maximizing the number of served customers, i.e., $w(\gamma) = |cust(\gamma)|$ Observe that this objective function differs from the optimization criterion used, for instance, in the Solomon benchmarks [18]. In the Solomon problems, the goal is to minimize the number of vehicles and, in case of ties, to minimize the total travel time, which corresponds more to strategic planning than the operational decision making of online optimization.

3 The Online Problem

In the online problem, requests, which are defined by a site and a time window, arrive dynamically and the algorithm must make three types of decisions:

1. Decide whether to accept or reject an incoming request;
2. Decide which vehicle will serve an accepted request;
3. Decide where to dispatch an idle vehicle.

Earlier work in online stochastic vehicle routing [6] ignored the second type of decisions and considered only frameworks in which the vehicle assignment is flexible and can be determined subsequently. The paper is motivated by applications in which this flexibility is not available and the vehicle assignment must be committed when the request is accepted. Our goal is to study the impact of this requirement and to adapt existing algorithms and methodologies to this new setting. We called these two settings *online flexible routing* and *online commitment routing* respectively.

States In online flexible routing, the algorithm maintains a current state which represents the state of each vehicle and the set of accepted requests which have not yet been served. The state of each vehicle is a pair (s, t) indicating that the vehicle will be at site s at time t or, alternatively, has arrived at site s at time t and is now idle. As a result, the states are of the form

$$\langle\{(s_1, t_1), \ldots, (s_n, t_n)\}, A\rangle$$

where A is the set of accepted requests at this state of the computation. Given a state σ, we use $\mathrm{REQUESTS}(\sigma)$ to denote the set of accepted requests in σ. In online commitment routing, the accepted requests must be assigned to vehicles and the states are of the form

$$\{(s_1, t_1, A_1), \ldots, (s_n, t_n, A_n)\}$$

where A_i denotes the set of accepted requests assigned to vehicle i. Hence, in online commitment routing, the state is a collection of single vehicle routing problems. We use a number of operations on states. In flexible routing, operation ADDREQUEST(r) adds request r to the state. In commitment routing, operation ADDREQUEST(r, v) which adds request r on vehicle v. We also have an operation that dispatches a vehicle to a particular site. More precisely, given a state σ, a vehicle v which is idle, and a site s, operation DISPATCH(σ,v,s) dispatches vehicle v to site s. Note that the site s does not necessarily correspond to a request: It can be the current site ("waiting") or a relocation [9].

Note that the problem could be modeled as a MDP over these states. However, the sheer size of the search space and the limited time available for decisions precludes the use of these techniques. Instead, we will use the exogenous nature of the uncertainty and the availability of an optimization algorithm and of a sampling procedure to design in a one-step anticipatory algorithm.

Sampling and Optimization The online algorithms have two black-boxes at their disposal: a conditional sampling procedure which generates scenarios of the future and an optimization algorithm to solve static routing problems. The sampling procedure returns a set of requests, while the optimization algorithm returns an optimal plan (or an approximation thereof). More precisely, the optimization algorithm $\mathcal{O}(\sigma, R)$ receives a state σ and a set of requests R and returns a routing plan γ. We are only interested in three pieces of information on a routing plan γ: the next location of a vehicle, the vehicle on which a request is served, and the number of customers served by the plan. These are denoted by NEXTLOCATION(γ, v), VEHICLE(γ, r), and $w(\gamma)$ respectively. It is important to note that the optimization algorithm is slightly different in online commitment routing, since it must also satisfy the vehicle allocation constraints.

4 Online Flexible Routing

We now review the earlier algorithms in online flexible routing proposed in [6; 9], since this research builds on them. For simplicity, we consider only the consensus algorithm from [6; 9], although it is easy to generalize the results to the regret algorithm. In this setting, the online algorithm must take two decisions: (1) to accept or reject an incoming request (functions PROCESSREQUEST and ACCEPTREQUEST in Figure 1) and (2) to choose where to dispatch an idle vehicle (function DISPATCH in Figure 1). The first decision is typically greedy: accept the incoming request if it can be served by routing plan serving all existing requests. (See line 1 in function ACCEPTREQUEST.) The second decision however is critical and uses stochastic information. It is based on the following principle: Generate a number of scenarios, use the optimization algorithm to solve them, and choose the resulting routing plan that is most appropriate for all scenarios. The vehicle

```
PROCESSREQUEST(State σ, Request r)
1  if ACCEPTREQUEST(σ, r)
2    then return ADDREQUEST(σ, r);
3    else  return σ;

ACCEPTREQUEST(State σ, Request r)
1  return w(O(σ, {r})) = |REQUESTS(σ)| + 1;

DISPATCH(State σ, Vehicle v)
 1  {ξ_1, ..., ξ_k} ← SAMPLE(k);
 2  for s ∈ Sites
 3  do f(s) ← 0;
 4  for i ∈ 1..k
 5  do γ_i ← O(σ, ξ_i);
 6     for j ∈ 1..m
 7     do s ← NEXTLOCATION(γ_i, j);
 8        f(s) ← f(s) + w(γ_i);
 9  γ* = arg-max_{i∈1..k} Σ_{j=1}^{m} f(NEXTLOCATION(γ_i, j));
10  return DISPATCH(σ, v, NEXTLOCATION(γ*, v));
```

Figure 1: Online Flexible Routing

Problem Name	MO	OSR	OSR-W	OSR-R	OSA-WR
20-20-60-rc101-1	2.08	2.24	4.16	3.30	3.64
20-20-60-rc101-2	6.78	5.42	5.94	3.62	4.26
20-20-60-rc101-3	3.06	2.06	3.06	2.28	3.08
20-20-60-rc101-4	2.90	3.16	4.30	5.54	3.06
20-20-60-rc101-5	7.70	4.02	5.48	5.12	3.44
20-20-60-rc102-1	1.74	1.78	1.22	0.54	0.36
20-20-60-rc102-2	4.28	1.94	3.44	2.76	3.32
20-20-60-rc102-3	8.70	3.24	5.06	3.32	3.96
20-20-60-rc102-4	2.18	0.92	1.48	1.84	1.92
20-20-60-rc102-5	3.76	2.46	2.90	2.02	1.88
20-20-60-rc104-1	21.10	19.70	14.40	13.82	13.94
20-20-60-rc104-2	25.56	28.58	13.92	11.70	14.38
20-20-60-rc104-3	20.90	16.64	10.40	8.84	9.78
20-20-60-rc104-4	19.60	19.28	14.08	6.36	13.08
20-20-60-rc104-5	15.86	18.96	14.00	9.94	10.96
Average	9.75	8.69	6.92	5.40	6.07

Table 1: Number of Rejected Customers in Online Flexible Routing

is then sent to its next location in the selected plan. To select the routing plans, the algorithm uses the desirability of the next decision for each vehicle, i.e., the number of times a request is scheduled next on a vehicle. More precisely, the algorithm generates k scenarios (line 1 in function DISPATCH) and solves them optimally (line 5). It then considers every vehicle (line 6) and determines which request is served first (line 7). Any such request is credited by the objective value of the plan (line 8). At this point, all the requests have been evaluated and the algorithm selects the plan γ^* that maximizes the selection of the desirable requests (line 9). The function returns a state in which vehicle v has been dispatched to its next location in plan γ^*. Note that the algorithm in [9] consider not only sending a vehicle to an existing request: it also allows the vehicle to wait at their existing location and to relocate to any customer location.

Table 1 reports our results on online flexible routing which provide a basis for comparison when we moved to online commitment routing. The experimental results are based on

```
PROCESSREQUEST(State σ, Request r)
1  if ACCEPTREQUEST(σ, r)
2    then v ← SELECTVEHICLE(σ, r);
3         return ADDREQUEST(σ, r, v);
4    else return σ;
```

Figure 2: Online Flexible Routing

Problem Name	MO	OSR	OSR-W	OSR-R	OSR-WR
20-20-60-rc101-1	8.0	9.0	10.0	10.0	10.0
20-20-60-rc101-2	12.0	6.3	7.1	6.7	6.9
20-20-60-rc101-3	2.0	8.0	8.8	8.8	8.7
20-20-60-rc101-4	8.0	7.0	9.0	9.2	9.0
20-20-60-rc101-5	5.0	6.0	6.0	6.0	6.1
20-20-60-rc102-1	3.0	2.2	2.2	2.4	2.6
20-20-60-rc102-2	10.0	8.5	9.3	9.4	9.3
20-20-60-rc102-3	6.0	5.1	9.1	9.5	9.6
20-20-60-rc102-4	5.0	7.0	4.3	4.2	4.6
20-20-60-rc102-5	10.0	7.0	7.3	7.8	7.6
20-20-60-rc104-1	29.0	32.5	36.1	36.7	36.5
20-20-60-rc104-2	39.0	32.2	34.5	33.6	33.7
20-20-60-rc104-3	20.0	24.6	26.6	26.8	27.2
20-20-60-rc104-4	28.0	25.1	25.5	25.7	24.2
20-20-60-rc104-5	21.0	23.3	23.6	23.9	24.4
Average	13.7	13.6	14.6	14.7	14.7

Table 2: Number of Rejected Customers in Online Commitment Routing with Greedy Acceptance and Vehicle Assignment.

some of the harder benchmarks proposed in [6] (class 4) which are stochastic versions of the hard Solomon problems which includes 100 customers. The results are the average of 50 runs for each type of instances. The stochastic algorithms use 10 scenarios for each decision, except for the first decision for which they use 100. The table reports the number of rejected customers by various algorithms: myopic optimization (MO) which uses the optimization algorithm but no stochastic information, the basic online stochastic routing algorithm (OSR), and then the variants with waiting (OSR-W), relocation (OSR-R), and both (OSR-WR). The results are consistent with those in [9] (they could be improved by using the regret algorithm). They indicate that stochastic information brings significant benefits in online flexible routing, particularly when waiting and/or relocation are used. The benefits are particularly significant on the harder instances (e.g., 20-20-60-rc104-2), where myopic optimization may reject about 25 customers in average, while the best stochastic algorithm would reject only about 12.

5 Online Commitment Routing

We now move to online commitment routing in which a request must be assigned a vehicle upon acceptance. This adds a third decision, selecting a vehicle, to the framework, captured in function PROCESSREQUEST (line 3) of Figure 2.

Since request acceptance is greedy in earlier work [6], it is tempting to proceed similarly for vehicle assignment and to assign the incoming request to minimize the total travel distance. Unfortunately, the solution quality of the algorithms deteriorates significantly with this choice, as indicated in Table 2. Obviously, the increased problem difficulty partly explains this quality loss, since the online myopic algorithm now rejects about 14 customers in average (instead of about 10) and may reject 39 customers (out of 100) in some instances. However, the performance of the stochastic algorithms is also extremely disappointing. The basic algorithm (OSR) is roughly comparable to online myopic optimization but the waiting and relocation variant perform even more poorly. This raises two fundamental questions:

1. Can the algorithms be enhanced to bridge most of the gap in solution quality?
2. Is there a value of stochatic information in online commitment routing and is there a real price of commitment?

The rest of this paper addresses both issues.

```
SELECTVEHICLE(State σ, Request r)
1  {ξ_1, ..., ξ_k} ← SAMPLE(k);
2  for v ∈ 1..m
3  do f(v) ← 0;
4  for i ∈ 1..k
5  do for v ∈ 1..m
6       do f(v) ← f(v) + w(O(ADDREQUEST(σ, r, v), ξ));
7  return arg-max_{v∈1..m} f(v);
```

Figure 3: Vehicle Selection for Commitment Routing

Problem Name	MO	OSR	OSR-W	OSR-R	OSR-WR
20-20-60-rc101-1	8.0	7.8	9.3	9.8	9.9
20-20-60-rc101-2	5.0	6.7	6.8	6.8	7.4
20-20-60-rc101-3	4.0	6.2	8.5	7.9	7.5
20-20-60-rc101-4	7.0	5.7	8.0	7.6	8.0
20-20-60-rc101-5	6.0	7.0	7.9	7.7	7.1
20-20-60-rc102-1	2.0	2.3	2.6	3.1	3.0
20-20-60-rc102-2	9.0	6.4	6.7	6.7	7.8
20-20-60-rc102-3	4.0	4.3	6.5	6.3	6.0
20-20-60-rc102-4	6.0	4.6	6.6	5.7	5.9
20-20-60-rc102-5	11.0	5.2	7.8	7.7	6.1
20-20-60-rc104-1	26.0	18.1	20.1	18.2	19.5
20-20-60-rc104-2	37.0	22.4	24.5	26.8	23.0
20-20-60-rc104-3	26.0	12.2	16.1	17.0	17.6
20-20-60-rc104-4	25.0	16.7	18.0	13.6	17.5
20-20-60-rc104-5	25.0	10.8	12.8	13.3	12.9
Average	13.7	9.1	10.8	10.5	10.6

Table 3: Number of Rejected Customers in Online Commitment Routing with Greedy Acceptance and Stochastic Vehicle Assignment.

6 Stochastic Vehicle Assignment

To improve solution quality, we first use stochastic information to select the vehicle for an incoming request. The algorithm for selecting the vehicle of a request is depicted in Figure 3. The key idea is to generate scenarios and to evaluate the consequences of the various allocation decisions on the number of serviced customers. The algorithm generates scenarios (line 1) and initializes the scores of the vehicles (lines 2–3). Then, for every scenario, it calls the optimization algorithms for each of the allocation decision (line 4–6). The algorithm then returns the allocation with the best score. The algorithm requires km optimizations per decision. The number can be reduced by using the regret algorithm [5].

Table 3 depicts the results which show significant improvements for stochastic algorithms and, in particular, on the

ACCEPTREQUEST($State\ \sigma, Request\ r$)

1 $\{\xi_1, \ldots, \xi_k\} \leftarrow$ SAMPLE(k);
2 **for** $v \in 1..m$
3 **do** $f(v) \leftarrow 0$;
4 $f^r \leftarrow 0$;
5 **for** $i \in 1..k$
6 **do for** $v \in 1..m$
7   **do** $f(v) \leftarrow f(v) + w(\mathcal{O}(\text{ADDREQUEST}(\sigma, r, v), \xi)))$;
8     $f^r \leftarrow f^r + w(\mathcal{O}(\sigma, \xi))$;
9 **return** $\max_{v \in 1..m} f(v) \leq f^r$;

Figure 4: Request Acceptance for Commitment Routing

Problem Name	MO	OSR	OSR-W	OSR-R	OSR-WR
20-20-60-rc101-1	8.0	9.2	11.6	11.9	11.7
20-20-60-rc101-2	5.0	5.9	5.7	6.1	5.9
20-20-60-rc101-3	4.0	5.9	7.6	7.6	7.3
20-20-60-rc101-4	7.0	8.0	7.6	7.4	7.7
20-20-60-rc101-5	6.0	6.7	8.0	7.7	8.2
20-20-60-rc102-1	2.0	2.5	3.5	3.2	3.1
20-20-60-rc102-2	9.0	6.4	6.6	7.4	7.5
20-20-60-rc102-3	4.0	4.3	5.7	6.4	6.7
20-20-60-rc102-4	6.0	5.8	6.6	6.9	5.9
20-20-60-rc102-5	11.0	4.6	5.7	5.5	5.3
20-20-60-rc104-1	26.0	17.4	19.2	19.9	18.5
20-20-60-rc104-2	37.0	20.7	22.3	21.7	22.0
20-20-60-rc104-3	26.0	13.3	12.7	14.3	15.5
20-20-60-rc104-4	25.0	13.8	13.1	11.4	14.6
20-20-60-rc104-5	25.0	11.1	12.6	11.3	12.1
Average	13.7	9.0	9.9	9.9	10.1

Table 4: Number of Rejected Customers in Online Commitment Routing with Stochastic Acceptance and Stochastic Vehicle Assignment.

harder instances. Algorithm OSR now rejects about 9 customers in average and it now accepts about 10 additional customers on the harder instances. The waiting and relocation algorithms show similar improvements but are still dominated by OSR. These results clearly indicated the value of stochastic information for vehicle assignment.

7 Stochastic Acceptance

We now move to the stochastic acceptance of requests, using stochastic information to decide whether to accept or reject a request. The algorithm is depicted in Figure 4 and follows the same pattern as the stochastic vehicle selection. The only additions are line 4 and 8 which are used to evaluate the decision to reject the request and, of course, line 9 which returns the decision. Intuitively, the idea underlying the algorithm is to evaluate the scenarios for each possible vehicle allocation decision and for request rejection. If the best vehicle allocation is superior to rejection, the request is accepted; It is rejected otherwise. Once again, the algorithm require km optimization which can be reduced by using the regret algorithm. The resulting algorithm exploits the same stochastic techniques for all three decisions, request acceptance, vehicle assignment, and vehicle dispatching, and provides a unified framework for online routing. Note also that our implementation combines request acceptance and vehicle allocation since the optimizations are similar.

Table 4 depicts its experimental results, which exhibit slight improvements for each of the stochastic algorithms, albeit marginal for OSR. Overall, the results indicate the value of stochastic information for online commitment routing. The solution quality of OSR in this seeting is close to its performance in online flexible routing. However, the waiting and relocation variants are not, suggesting that waiting and relocation strategies have little benefit in online commitment routing. This is rather unexpected, since the vehicle assignment does not prescribe any specific dispatching decision but just the vehicle allocation. As a consequence, the benefits of waiting and relocation seem to be the main loss of moving from a flexible to a commitment framework.

Problem Name	MO	OSR	OSR-W	OSR-R	OSR-WR
20-20-60-rc101-1	5.0	6.4	7.9	8.3	8.2
20-20-60-rc101-2	9.0	4.7	4.4	5.0	5.1
20-20-60-rc101-3	0.0	4.5	6.0	5.7	5.5
20-20-60-rc101-4	5.0	6.2	8.4	7.9	7.7
20-20-60-rc101-5	3.0	5.0	6.3	5.9	6.2
20-20-60-rc102-1	2.0	1.4	0.7	0.6	1.3
20-20-60-rc102-2	8.0	3.6	5.1	4.6	4.7
20-20-60-rc102-3	2.0	3.3	5.1	4.4	5.2
20-20-60-rc102-4	2.0	2.6	3.8	3.5	3.8
20-20-60-rc102-5	6.0	3.5	3.6	3.1	4.2
20-20-60-rc104-1	23.0	11.7	15.3	16.1	14.7
20-20-60-rc104-2	29.0	13.9	13.9	13.1	13.8
20-20-60-rc104-3	18.0	11.0	13.1	14.3	13.8
20-20-60-rc104-4	25.0	9.1	12.3	10.3	10.7
20-20-60-rc104-5	18.0	6.7	9.3	7.5	9.0
Average	10.3	6.2	7.7	7.4	7.6

Table 5: Number of Rejected Customers for Online Commitment Routing with Resource Augmentation: One Additional Vehicle

Problem Name	MO	OSR	OSR-W	OSR-R	OSR-WR
20-20-60-rc101-1	3.0	5.0	6.0	6.1	5.7
20-20-60-rc101-2	6.0	3.1	3.2	3.1	2.9
20-20-60-rc101-3	0.0	2.4	2.8	3.4	3.5
20-20-60-rc101-4	4.0	4.0	5.9	6.0	6.0
20-20-60-rc101-5	2.0	4.2	5.4	5.6	5.7
20-20-60-rc102-1	0.0	0.4	0.3	0.6	0.1
20-20-60-rc102-2	5.0	1.8	2.6	2.2	3.1
20-20-60-rc102-3	0.0	1.6	2.4	2.9	2.9
20-20-60-rc102-4	1.0	2.0	2.1	3.5	2.5
20-20-60-rc102-5	4.0	1.9	1.3	1.8	1.7
20-20-60-rc104-1	19.0	9.1	10.1	12.3	11.8
20-20-60-rc104-2	25.0	10.8	11.5	11.4	13.6
20-20-60-rc104-3	16.0	9.2	10.6	10.0	11.5
20-20-60-rc104-4	21.0	7.9	6.9	10.9	8.4
20-20-60-rc104-5	16.0	5.5	7.1	5.2	7.7
Average	9.5	4.6	5.2	5.8	5.8

Table 6: Number of Rejected Customers for Online Commitment Routing with Resource Augmentation: Two Additional Vehicles

8 Resource Augmentation

A traditional methodology in online algorithms consists in adding resources to account for the additional cost of taking decisions dynamically. In this section, we consider resource augmentation for online commitment routing by adding one or more vehicles.

Table 5 reports the results with one addition vehicle. The results are illuminating. OSR now improves on its results for online flexible routing, moving from 8.69 to 6.2 rejections in average and significantly reducing the number of rejected customers on the harder instances. This should be contrasted with online myopic optimization whose average number of rejections for online commitment routing with one additional vehicle (10.3) is still higher than its flexible counterpart. Resource augmentation also improves the waiting and relocation variants but they are still dominated by OSR. In fact, the distance between OSR on the one hand and OSR-W, OSR-R, and OSR-RW on the other hand increases both in percentage and absolute terms with resource augmentation.

Table 6 depicts the results with 2 additional vehicles. Once again, the stochastic algorithms benefit the most from the additional resource, but the waiting and relocation variants still do not improve the basic stochastic algorithm.

9 Conclusion

In this paper, we studied online stochastic vehicle routing in a setting which requires to commit specific vehicles to accepted requests (online commitment routing). This online commitment routing setting is motivated by actual applications in which the flexibility of moving a request from a vehicle to another is fundamentally undesirable or not available. Experimental results indicated that simple generalizations of existing algorithms for online flexible routing do not scale to online commitment routing. We then proposed a unified framework in which all three decisions, i.e., request acceptance, vehicle selection, and vehicle dispatching, are based on similar principles and use a combination of sampling and optimization. The experimental results show significant improvements in quality, demonstrating the value of stochastic information in this setting and showing the criticality of vehicle selection. However, the solution quality for online commitment routing was still inferior to its flexible counterpart. For this reason, we consider resource augmentation and showed that the proposed algorithm with one additional vehicle outperforms its flexible counterpart. However, there is an additional price of commitment: the waiting and relocation strategies so effective in online flexible routing degrade solution quality when vehicle commitments are required. Understanding why this is the case and characterizing when waiting and relocation are valuable are interesting open issues.

References

[1] T. Benoist, E. Bourreau, Y. Caseau, and B. Rottembourg. Towards stochastic constraint programming: A study of online multi-choice knapsack with deadlines. In *CP'01*, pages 61–76, London, UK, 2001. Springer-Verlag.

[2] R. Bent, I. Katriel, and P. Van Hentenryck. Sub-Optimality Approximation. In *CP'05*, Stiges, Spain, 2005.

[3] R. Bent and P. Van Hentenryck. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 8(4):515–530, 2004.

[4] R. Bent and P. Van Hentenryck. Online Stochastic and Robust Optimization. In *ASIAN'04*, Chiang Mai University, Thailand, December 2004.

[5] R. Bent and P. Van Hentenryck. Regrets Only. Online Stochastic Optimization under Time Constraints. In *AAAI'04*, San Jose, CA, July 2004.

[6] R. Bent and P. Van Hentenryck. Scenario Based Planning for Partially Dynamic Vehicle Routing Problems with Stochastic Customers. *OR*, 52(6), 2004.

[7] R. Bent and P. Van Hentenryck. The Value of Consensus in Online Stochastic Scheduling. In *ICAPS 2004*, Whistler, British Columbia, Canada, 2004.

[8] P. Van Hentenryck and R. Bent. Online Stochastic Combinatorial Optimization. MIT Press, 2006.

[9] R. Bent and P. Van Hentenryck. Waiting and Relocation Strategies in Online Stochastic Vehicle Routing. In *IJCAI 2007*, Hyderbad, India, 2007.

[10] H. Chang, R. Givan, and E. Chong. On-line Scheduling Via Sampling. *Artificial Intelligence Planning and Scheduling (AIPS'00)*, pages 62–71, 2000.

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

[12] P. Jaillet and M. Wagner. Generalized Online Routing: New Competitive Ratios, Resource Augmentation and Asymptotic Analyses. Operations Research, 2008.

[13] A. Larsen, O. Madsen, and M. Solomon. The a-priori dynamic traveling salesman problem with time windows. *Transportation Science*, 38:459–472, 2004.

[14] M. Mitrovic-Minic, R. Krishnamurti, and G. Laporte. Double-Horizon Based Heuristics for the Dynamic Pickup and Delivery Problems with Time Windows. *Transportation Research Record Part B*, 38:669–685, 2004.

[15] M. Mitrovic-Minic and G. Laporte. Waiting Strategies for the Dynamic Pickup and Delivery Problem with Time Windows. *Transportation Research Record Part B*, 38:635–655, 2004.

[16] David Montana, Jose Herrero, Gordon Vidaver, and Garrett Bidwell. A Multi-Agent Society for Military Transportation Scheduling. *Journal of Scheduling*, pages 225–246, 2000.

[17] P. Shaw. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In *CP'98*, pages 417–431, Pisa, October 1998.

[18] M.M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35 (2):254–265, 1987.

[19] J. van Hemert and J. La Poutré. Dynamic Routing with Fruitful Regions: Models and Evolutionary Computation. In *Parallel Problem Solving from Nature VIII*, pages 690–699, 2004.

[20] P. Van Hentenryck, R. Bent, and Y. Vergados. Online Reservation Systems. In *CP-AI-OR'06*, Cork, Ireland, 2005.