J Math Model Algor (2007) 6:87–107
DOI 10.1007/s10852-006-9051-4

A Solution Method for a Two-dispatch Delivery Problem with Stochastic Customers

Raymond K. Cheung · Dongsheng Xu · Yongpei Guan

Received: 27 October 2004 / Accepted: 30 November 2005 /
Published online: 28 September 2006

Abstract We study a vehicle routing problem in which vehicles are dispatched multiple times a day for product delivery. In this problem, some customer orders are known in advance while others are uncertain but are progressively realized during the day. The key decisions include determining which known orders should be delivered in the first dispatch and which should be delivered in a later dispatch, and finding the routes and schedules for customer orders. This problem is formulated as a two-stage stochastic programming problem with the objective of minimizing the expected total cost. A worst-case analysis is performed to evaluate the potential benefit of the stochastic approach against a deterministic approach. Furthermore, a sample-based heuristic is proposed. Computational experiments are conducted to assess the effectiveness of the model and the heuristic.

Key words vehicle routing · stochastic customers · stochastic programming.

Mathematics Subject Classifications (2000) 90B06 · 90C15 · 90C59 · 90C10.

R. K. Cheung (✉) · D. Xu
Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
e-mail: rcheung@ust.hk

D. Xu
e-mail: xuds@ust.hk

Y. Guan
School of Industrial Engineering, The University of Oklahoma, Norman, OK 73019, USA
e-mail: yguan@ou.edu

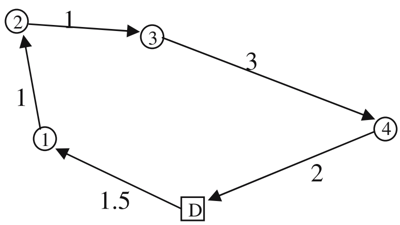
1 Introduction

Vehicle routing problems (VRP) have long been an active research area because of their practical value. In this paper, we consider a version of VRP where vehicles are dispatched multiple times a day for product delivery. Some customer orders are known in advance, while others are uncertain but are progressively realized through the day. At a dispatch decision epoch, we are allowed to delay the delivery of some known orders to a later dispatch. We determine the routing and scheduling of the orders that are to be delivered immediately. The delayed orders are grouped together with those that are realized before the next vehicle dispatch. We then determine the routing and scheduling for the combined set of orders. This process is applied on a rolling horizon basis to solve the multiple-dispatch VRP. For simplicity and for practical reasons, we assume that there are two dispatches (which is true in the motivating applications described below), for instance, the morning dispatch and the afternoon dispatch. More specifically, in order to accommodate late realized orders, we define an earliest start time for the second dispatch. This routing problem is formulated as a two-stage stochastic programming problem with the objective of minimizing the expected total cost. We propose a heuristic solution strategy to solve this stochastic programming problem. Computational experiments are conducted to evaluate the benefit of this approach over existing practices.

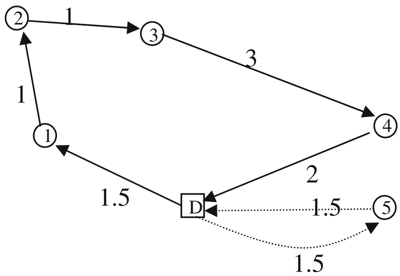
One motivating application is the delivery problem of a chemical product supplier that produces and delivers products to a fixed set of customers, which we refer to as the supplier case. The supplier dispatches vehicles two times a day. Normally, for orders that are placed before a set cut-off time, currently 4 P.M., the supplier will make the products and deliver them the next day. Since there is a production lead time, only the orders that are received before a given time, say 2 P.M., are scheduled to be delivered in the first dispatch. The orders received between this time and the cut-off time are considered as *late* orders. There are also a number of *urgent* orders that only come in the next morning after the first dispatch decisions are made, but require same day delivery (in the afternoon). The current practice of the supplier is to dispatch as many early orders as possible in the first dispatch while reserving the second dispatch for the late and urgent orders. We call this practice the First-Come-First-Served policy. Note that there can be cost-savings if some early orders are scheduled in the second dispatch. It has been observed that the return times of the vehicles to the depot vary significantly and that the workloads of the vehicles during different dispatches are highly uneven. Without a decision support system for their planning, the supplier's transport department spends more than two hours per day on determining a schedule manually with low truck utilization. Moreover, there are orders that cannot be delivered using the company's own trucks and a third-party trucking company is used for delivering these unserved orders (at a higher cost and with a lower service level).

Another motivating application comes from a distributor who needs to deliver consumer products to retail stores in a city. On each day, the distributor dispatches vehicles two times. The orders received the day before are delivered in the first dispatch and the orders received in the morning or *urgent* orders are delivered in the second dispatch. We refer to this application as the distributor case. Unlike the supplier case, the delivery area in this case is divided into zones and a particular vehicle is reserved for each particular zone. Due to road traffic, side road parking restrictions (no parking is allowed during certain periods of a day), and the need to visit retailers during their non-peak hours, the start times of the two dispatches need to be more regular than in the supplier case. When there are more known orders than can be delivered in the first dispatch (due to vehicle capacity constraints), orders that arrive earlier will be delivered first.

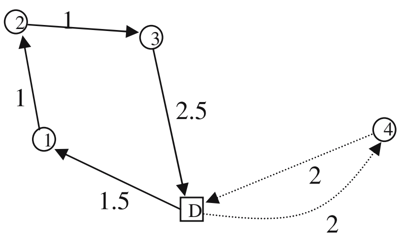
Delaying some known orders to the second dispatches may result in some benefit. Consider the example shown in Figure 1 that has four known orders (orders 1–4) and one uncertain order (order 5). Assume that the capacity of the truck is large enough to hold all five orders. Figure 1a shows the vehicle serving all four known orders in one dispatch. If order 5 appears later, a second dispatch is needed to deliver it (see Figure 1b). Let p be the probability that order 5 appears. Then, the expected total travel time for this approach is $8.5 + 3p$. If we delay order 4 to the second dispatch, the travel time for the first dispatch is 6 while the travel time for the second dispatch is 4 without order 5 (see Figure 1c) and 4.5 with order 5 (see Figure 1d). Then, the expected total travel time for this approach is $10 + 0.5p$. Therefore, when $p > 0.6$, delaying order 4 to the second dispatch will result in a shorter expected total travel time.



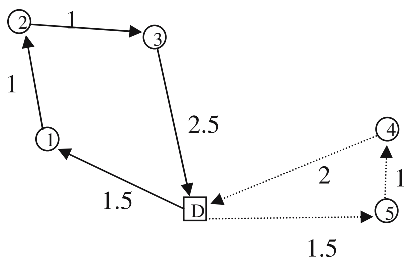
(a) Serving all known demands in one trip



(b) A separate trip is used to deliver demand 5, which is not known when making the initial routing decision.



(c) Delaying demand 4 to the second trip and no new demand is realized



(d) Delaying demand 4 to the second trip and demand 5 is realized

Figure 1 An illustration of delaying a known order to the second dispatch.

The problem considered here is related to the class of VRP with stochastic demand (VRPSD) in the literature [1, 5, 16]. Gendreau et al. [8, 9] consider the version of VRPSD for shipment collection from customers, whose presences are uncertain. In their model, vehicles follow a set of planned routes and skip the absent customers. Whenever a vehicle capacity is attained or exceeded, the vehicle returns to the depot and resumes its collections along the planned route. Another version of the VRPSD assumes that all customers are present but the sizes of their demands are random (see [22]). Laporte et al. [13] formulate this problem as a stochastic programming problem with chance constraints. This problem in turn is transformed into a deterministic VRP problem. An implementation of the integer L-shaped method for the exact solution of the problem is studied by Laporte et al. [14] as well. Gendreau et al. [9] study this problem and construct vehicle routes with minimum expected cost. Laporte and Louveaux [12], Waters [25], and Dror et al. [6] develop various properties and formulations for this type of VRPSD with recourse. Yang et al. [26] present an idea of preventive breaks or restocking. The idea is to consider returning vehicles to the depot for replenishment after serving each customer in anticipation of possible stock-outs. Secomandi [19, 20] studies a rollout policy for the vehicle routing problem with stochastic demands and analyzes this approach to sequencing problems with stochastic routing applications. Finally, Psaraftis [17] describes the status and prospects of the related dynamic vehicle routing problems. Larsen et al. [15] introduce the related partially dynamic travelling repairman problem and describe several dynamic policies to minimize routing costs. Haughton [10] and Savelsbergh and Goetschalckx [18] evaluate the benefit of route reoptimization.

Unlike in those discussed in the literature, in our problem, when a vehicle starts a dispatch, all orders have to be delivered before the vehicle returns to the depot. Furthermore, instead of the order size or travel time, the main source of uncertainty comes from the exact day on which an order is placed, which, as shown later, can also be used to model random order size. Since historical data help to estimate the probability of the occurrence of such an order, we would like to make use of such information to devise an effective schedule.

The contributions of this paper are as follows. First, we study a different version of the routing problem with uncertainty in which we look at whether or not a known order should be delayed to a later dispatch. Despite the problem not having been formally studied in the literature, it has a great practical value. We formulate this problem in the framework of a two-stage stochastic program. Second, we perform a worst-case analysis for the ratio of the total cost using the deterministic approach that is often used in practice to the total cost using our stochastic programming approach, showing that this ratio can be as high as two for a special case and infinity for general cases. Third, we derive a sampling-based heuristic to solve the problem. The performance of the method is then evaluated using randomly generated data sets and real data sets.

The remainder of the paper is organized as follows. Section 2 provides the two-stage stochastic programming formulation of the problem while the worst-case analysis is discussed in Section 3. Section 4 describes our solution approach and various implementation issues. Section 5 presents the computational results based on modified Solomon's benchmark problems and real application problems. Finally, concluding remarks are offered in Section 6.

2 A Two-stage Formulation

Based on the dispatching practices in the two motivating applications, we make a number of assumptions. First, the demand size of each order is no larger than the capacity of the largest vehicle and each order cannot be split. Second, not all orders must be served by the company's vehicles. The unserved orders will be subcontracted out to a third-party transportation provider for delivery with a higher cost. To reflect this in the model, we assume that there is a *super* vehicle that has infinite capacity and zero travel time between orders. Third, the probability of whether a customer will place an order on a particular day is given, which typically can be estimated through historical data. Finally, there is a relatively long delivery time window for each order. In our motivating applications, most orders are required to be delivered within a day rather than at a specific time.

In our formulation, we model the first dispatch of vehicles as the stage-one problem and the second dispatch as the stage-two problem. Let $(\Omega, \mathcal{A}, \mathcal{P})$ be the probability space and $\omega \in \Omega$ be an outcome reflecting the scenario that a particular set of stochastic customers actually place their orders.

The random customer presence can also partially model random order size. For example, we can use several virtual customers to represent an actual customer. The order size for each virtual customer is fixed, while whether or not the virtual customer will place an order is random. The joint demand of these virtual customers can be used to model the order size of the actual customer demand. For ease of presentation, we simply assume that the order size of each customer is fixed. Furthermore, for simplicity, we refer to the order placed by the stochastic customer as a *stochastic order*.

2.1 Customers

$\mathcal{N}$ = Set of all possible customer orders.
$\mathcal{N}_d$ = Set of known customer orders, $\mathcal{N}_d \subset \mathcal{N}$.
$\mathcal{N}_s$ = Set of stochastic orders which may or may not appear.
$\mathcal{N}_s(\omega)$ = Set of stochastic orders that actually appear when the outcome is ω.
d_i = Size of known order, $i \in \mathcal{N}_d$.
$\widetilde{d}_i$ = Size of stochastic order, $i \in \mathcal{N}_s$.
p_i = Probability of order i being present, $i \in \mathcal{N}_s$.

Note that $\widetilde{d}_i$ is known for a given $\mathcal{N}_s(\omega) \subset \mathcal{N}$ and $\mathcal{N}_s(\omega) \cap \mathcal{N}_d = \emptyset$.

2.2 Vehicles

K = Number of vehicles.
U_k = Capacity of vehicle k.

We assume that vehicle 0 is the super vehicle which reflects the subcontracting option and let $\mathcal{K}$ be the set of vehicles, $0, 1, 2, \ldots, K$.

2.3 Time and Cost Parameters

$c_{ij,k}$ = Travel cost from the location of order i to the location of order j by vehicle k.
t_{ij} = Travel time from the location of order i to the location of order j.
s_i = Service time for order i.
T = Duty time for the drivers.
Q = Latest time that stochastic orders can be scheduled for delivery.
= Earliest vehicle departing time of the second dispatch.

If a stochastic order arrives after Q, this order will not be delivered on this day. The cost, $c_{ij,k}$, depends on the travel distance between the locations of orders i and j, the service time of order j, and a fixed set-up cost at the customer's location. In this paper, we assume that the costs satisfy the "triangle inequality." That is, $c_{ij,k} + c_{j\ell,k} \geq c_{i\ell,k}$, which is reasonable since the travel distance satisfies the "triangle inequality" and the service time and the fixed set-up cost are positive. In dense city areas, different orders may be delivered to locations that are within a short walking distance from each other. For example, different retailers from the same shopping mall may place orders or the same retailer may place multiple orders in a day. Thus, there is a fixed set-up cost (e.g., the cost for parking) in the order-serving process. Note that this cost is shared by the orders occurring at the same customer location.

2.4 Decision Variables

$$\delta_i = \begin{cases} 1, & \text{if known order } i \text{ is served in dispatch 1,} \\ 0, & \text{otherwise.} \end{cases}$$

$$y^t_{ik} = \begin{cases} 1, & \text{if known order } i \text{ is delivered by vehicle } k \text{ in dispatch } t, t = 1, 2, \\ 0, & \text{otherwise.} \end{cases}$$

$$x^t_{ij,k} = \begin{cases} 1, & \text{if vehicle } k \text{ visits customer } i \text{ and then customer } j \text{ in dispatch } t, t = 1, 2, \\ 0, & \text{otherwise.} \end{cases}$$

In our presentation, a symbol with an index omitted is used to describe the set of the symbols over all possible values of this index. For example, y^1_k represents the set of y^1_{ik} over all i and y^1 represents the set of y^1_k over all k. We also define two node sets:

$\mathcal{N}^1_k$ = Set of orders served by vehicle k in dispatch 1.
$\mathcal{N}^2_k(\omega)$ = Set of orders served by vehicle k in dispatch 2 when the outcome is ω.

Our objective is to minimize the cost of the first dispatch and the expected cost of the second dispatch. The stochastic programming formulation can be written as

$$Z = \min_{\delta} g^1(\delta) + E_\omega g^2(\delta, \omega), \tag{1}$$

where $g^1(\delta)$ and $E_\omega g^2(\delta, \omega)$ represent the stage-one cost and the expected stage-two cost for a given δ, respectively. For simplicity, we do not explicitly write the integrality requirement of the decision variables as constraints.

2.5 The Stage-one Problem

For a given δ, the value of $g^1(\delta)$ is the minimum total cost of routing all vehicles in their first dispatches. That is, $g^1(\delta)$ is defined by

$$g^1(\delta) := \min \sum_{k\in\mathcal{K}} f_k^1(y_k^1) \tag{2}$$

subject to:

$$\sum_{k\in\mathcal{K}} y_{ik}^1 = \delta_i \qquad \forall i \in \mathcal{N}_d, \tag{3}$$

$$\sum_{i\in\mathcal{N}_d} d_i y_{ik}^1 \le U_k \qquad \forall k \in \mathcal{K}. \tag{4}$$

Constraint set (3) says that if known order i is served in dispatch 1, then it must be served by exactly one vehicle. Constraint set (4) ensures that the capacity constraints are satisfied. The route cost $f_k^1(y_k^1)$ for each vehicle k in the objective function is computed by solving a time-constrained travelling salesman problem. Notice that $j \in \mathcal{N}_k^1$ means $y_{jk}^1 = 1$. Thus, the problem is defined as

$$f_k^1(y_k^1) \ := \ \min \sum_{ij} c_{ij,k} x_{ij,k}^1 \tag{5}$$

subject to:

$$\sum_{i\in\mathcal{N}_k^1} x_{ij,k}^1 = 1 \qquad \forall j \in \mathcal{N}_k^1, \tag{6}$$

$$\sum_{j\in\mathcal{N}_k^1} x_{ij,k}^1 = 1 \qquad \forall i \in \mathcal{N}_k^1, \tag{7}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij,k}^1 \le |S| - 1 \qquad \forall S \subseteq \mathcal{N}_k^1, 2 \le |S| \le |\mathcal{N}_k^1|, \tag{8}$$

$$\sum_{ij} (t_{ij} + s_j) x_{ij,k}^1 \le T. \tag{9}$$

Constraint sets (6) and (7) enforce flow conservation; constraint set (8) avoids the formation of sub-tours; and constraint set (9) makes sure that the tour can be finished within the driver's duty period.

2.6 The Stage-two Problem

Let T_k^1 be the time when vehicle k finishes dispatch 1. That is, $T_k^1 = \sum_{ij}(t_{ij} + s_j)x_{ij,k}^1$. Note here T_k^1 depends on the first stage decision variables $x_{ij,k}$. For a given outcome, ω, the cost function of the stage-two problem, $g^2(\delta, \omega)$, is given as

$$g^2(\delta, \omega) := \min \sum_{k\in\mathcal{K}} f_k^2(y_k^2, T_k^1) \tag{10}$$

subject to:

$$\sum_{k\in\mathcal{K}} y_{ik}^2 = 1-\delta_i \qquad \forall i\in\mathcal{N}_d, \tag{11}$$

$$\sum_{k\in\mathcal{K}} y_{ik}^2 = 1 \qquad \forall i\in\mathcal{N}_s(\omega), \tag{12}$$

$$\sum_{i\in\mathcal{N}_s(\omega)} \widetilde{d}_i y_{ik}^2 + \sum_{i\in\mathcal{N}_d} d_i y_{ik}^2 \le U_k \qquad \forall k\in\mathcal{K}. \tag{13}$$

Constraint set (11) says that, among all known orders, only those that are not dispatched in dispatch 1 will be considered in dispatch 2. Furthermore, constraint sets (12) and (13) ensure that each order in dispatch 2 can be delivered and that the vehicle capacity constraint must be satisfied. Similar to the stage-one problem, for the set $\mathcal{N}_k^2(\omega)$, the route cost can be obtained by solving a problem analogous to Eqs. (5)–(9) with T in Eq. (9) replaced by $T-\max\{T_k^1, Q\}$, which implies that if vehicle k returns to the depot from the first dispatch before time Q, it must wait until Q to start the second dispatch; otherwise, it can start the second dispatch immediately.

3 Worst Case Analysis

In this section, we compare the costs of two approaches. The first is a deterministic approach in which we put all known orders in the first dispatch and then schedule the second dispatch only after the uncertain orders are realized. The second is the stochastic approach in which we delay some known orders according to the solution of the two-stage stochastic programming formulation. The following propositions show the relationship of the optimal total costs when solving the problem using these two approaches.

Proposition 1 If $T\to+\infty$, the worst-case ratio of the cost of using the optimal deterministic approach to the cost of using the optimal stochastic approach is 2. Furthermore, the ratio is asymptotically tight.

Proof First, we consider the deterministic approach. In this approach, we set $\delta_i=1$ for all $i\in\mathcal{N}_d$. That is, all known orders are put in the first dispatch and all realized uncertain orders are put in the second dispatch. Let $g^{\mathrm{D},1}$ and $g^{\mathrm{D},2}(\omega)$ be the optimal stage-one cost and the optimal stage-two cost for an outcome, ω, respectively. Then, the value of the deterministic approach is given by

$$Z^{\mathrm{D}} = g^{\mathrm{D},1} + E_\omega g^{\mathrm{D},2}(\omega). \tag{14}$$

Second, we consider a wait-and-see [2], or posterior, approach where both routing decisions are made with knowledge of the realized demand. For a given δ and an outcome ω, let $g^{\mathrm{P},1}(\delta)$ and $g^{\mathrm{P},2}(\delta,\omega)$ be the stage-one cost and the stage-two cost for this approach, respectively. Then, the optimal value corresponding to an outcome ω by the wait-and-see approach is

$$g^{\mathrm{P}}(\omega) = \min_\delta g^{\mathrm{P},1}(\delta) + g^{\mathrm{P},2}(\delta,\omega). \tag{15}$$

By taking expectation, the optimal expected value is

$$Z^{\mathrm{P}} = E_\omega\left\{g^{\mathrm{P}}(\omega)\right\} = E_\omega\left\{\min_\delta g^{\mathrm{P},1}(\delta) + g^{\mathrm{P},2}(\delta,\omega)\right\}. \tag{16}$$

Note that, in this approach, we make a decision with perfect information (that is, after uncertain orders are realized) while, in the stochastic approach, we make a decision with partial information. We also observe that the optimal solution, δ^*, for Eqs. (1) to (13) is also a feasible solution for the wait-and-see case. Thus,

$$Z^{\mathrm{P}} \leq Z. \tag{17}$$

We now compare Z^{D} and Z^{P}. Note that since $T \to +\infty$, there is no time constraint for the vehicle routing problems corresponding to $g^{\mathrm{D},1}$, $g^{\mathrm{D},2}(\omega)$ and $g^{\mathrm{P}}(\omega)$. The two vehicle routing problems corresponding to $g^{\mathrm{D},1}$ and $g^{\mathrm{D},2}(\omega)$ can be solved independently. Then, $g^{\mathrm{D},1}$ and $g^{\mathrm{D},2}(\omega)$ are optimal values corresponding to two mutually exclusive subsets of orders for outcome ω without time constraints, while $g^{\mathrm{P}}(\omega)$ is the optimal value of the whole set of orders for the same outcome. Due to the "triangle inequality" cost assumption, for a given ω, the optimal value of each of the subsets is less than the optimal value of the whole set. Thus, we have

$$\max\left\{g^{\mathrm{D},1}, g^{\mathrm{D},2}(\omega)\right\} \leq g^{\mathrm{P}}(\omega).$$

Since $g^{\mathrm{D},1} \leq \max\left\{g^{\mathrm{D},1}, g^{\mathrm{D},2}(\omega)\right\}$ and $g^{\mathrm{D},2}(\omega) \leq \max\left\{g^{\mathrm{D},1}, g^{\mathrm{D},2}(\omega)\right\}$, we have

$$g^{\mathrm{D},1} + g^{\mathrm{D},2}(\omega) \leq 2\cdot\max\left\{g^{\mathrm{D},1}, g^{\mathrm{D},2}(\omega)\right\} \leq 2g^{\mathrm{P}}(\omega).$$

After taking expectations on both sides, we have

$$Z^{\mathrm{D}} \leq 2Z^{\mathrm{P}}. \tag{18}$$

Combining Eqs. (17) and (18), we have

$$Z^{\mathrm{D}} \leq 2Z^{\mathrm{P}} \leq 2Z.$$

To show that this worst-case bound is asymptotically tight, we construct the following example. Suppose that there is only one truck serving two orders, say order 1 and order 2, located at the same location with the travel cost to the depot as $L/2$ and the capacity of the truck large enough to hold both orders. Assume that order 1 is known and order 2 occurs with probability $1-\eta$. In the deterministic approach, we need to make two dispatches and the expected travel distance is

$$Z^{\mathrm{D}}(\eta) = g^{\mathrm{D},1} + E_\omega g^{\mathrm{D},2}(\omega) = L + (1-\eta)L = 2L - \eta L.$$

In the stochastic approach, the optimal solution is to delay order 1 to the second dispatch and there is only one dispatch in the optimal solution with the value of

$$Z(\eta) = \eta L + (1-\eta)L = L.$$

Therefore,

$$\frac{Z^{\mathrm{D}}(\eta)}{Z(\eta)} = \frac{2L-\eta L}{L} = 2-\eta \to 2 \ \text{ as } \eta \to 0.$$

This example shows that we can construct an example such that the ratio is arbitrarily close to 2. □

Our analysis shows that if $T \rightarrow +\infty$ (that is, no time constraints), the total cost for the deterministic approach can be twice the total cost for the stochastic approach in the worst case. This corresponds to the case that allows over-time operation in practice. If T is a finite number, the worst-case ratio can be unbounded.

Proposition 2 If $T < \infty$, the worst-case ratio of the cost of using the optimal deterministic approach to the cost of using the optimal stochastic approach can be unbounded.

Proof We prove it by constructing an instance. Considering the case that there is one regular vehicle and two orders, order 1 is deterministic and order 2 is stochastic with probability $1 - \epsilon$ to appear before $Q = T/3$. These two orders have the same weights of less than half of regular vehicle capacity and are at the same location which is $L/2$ away from the depot. The round trip travel time between the depot and the order is $2T/3$. For the optimal deterministic approach, the regular vehicle can only finish one trip within T and order 2 has to be delivered by a super vehicle. For the optimal stochastic approach, we can deliver the regular vehicle at time $Q = T/3$ to serve orders 1 and 2 together and return to the depot at time T. When the super vehicle costs M times as much as the regular vehicle, the myopic solution has the expected cost of $L + (1 - \epsilon)ML$ and the stochastic solution has a cost L. Therefore, the ratio depends on the value of M and can be arbitrarily large when $M \rightarrow +\infty$. □

In the next section, we develop a heuristic with which we can significantly improve the solution quality produced by the deterministic approach.

4 The Solution Approach

The problem Eqs. (1)–(13) is a two-stage stochastic integer programming problem. Even the deterministic version of this is well known to be $\mathcal{NP}$-hard, making the use of general solution methods for stochastic programming difficult (see, for example, [2] for a review of these methods). Here, we develop a sampling-based heuristic. The use of a sampling scheme to solve stochastic programming problems has appeared in the literature. For example, the class of stochastic quasi-gradient methods uses sample sub-gradients iteratively to guide the optimization process (see, for example, [3, 7]). One issue is that their rate of convergence is well known to be small. Recently, Kleywegt et al. [11] proposed a sample average approximation (SAA) method that can control the number of samples used to obtain solutions that are within a given confidence interval of the optimal value. Applications of this method in solving the shortest path problems with random travel time and with random arc failures can be found in Verweij et al. [24]. Applying SAA method to our problem, however, is very computationally demanding. Our preliminary experiments indicated that a few iterations could take hours of CPU time. These details are reported in the appendix. Finally, Cheung and Hang [4] study the use of sampling to estimate dual prices when solving multistage networks with random arc durations in the context of dynamic assignments, which is a special case of VRP that can be formulated as minimum cost flow problems. Here, we also use sampling to estimate dual prices, but the underlying problem structure is much more complex.

4.1 Temporal Order Exchange Procedure

In the procedure, we use samples of the stochastic orders to approximate the benefit of moving a known order from the first dispatch to the second dispatch and vice versa in the optimization process. The decision to move orders between the two dispatches is made based on the approximated values. Since exchanging the orders in different dispatches is involved, we call the heuristic *Temporal ORder Exchange procedure* (TORE). To describe the procedure, let

$\pi_i^{1,-}$ = Estimated change of Z if known order i in dispatch 1 is delayed to dispatch 2,
$\pi_j^{2,-}$ = Estimated change of Z if known order j in dispatch 2 is moved to dispatch 1,
r_i^t = Estimated change of stage t cost if known order i is removed from dispatch t,
a_i^t = Estimated change of stage t cost if known order i is added to dispatch t.

Notice that $\pi_i^{1,-}$ and $\pi_j^{2,-}$ represent the estimates of the dual price with respect to constraint (3), that is, the change of Z with respect to δ_i when its value changes from 1 to 0 and when its value changes from 0 to 1. The values of $\pi_i^{1,-}$ and $\pi_j^{2,-}$ can be obtained as

$$\pi_i^{1,-} = r_i^1 + a_i^2, \tag{19}$$

$$\pi_j^{2,-} = r_j^2 + a_j^1. \tag{20}$$

Since some stochastic orders are not known in stage 1, N samples of the set of stochastic orders are used to estimate r_j^2 and a_i^2. For sample n, let

$r_j^{2,n}$ = Estimated change of stage 2 cost if order i is removed from dispatch 2,
$a_i^{2,n}$ = Estimated change of stage 2 cost if order i is added to dispatch 2,

and we naturally define

$$r_j^2 = \frac{1}{N}\sum_{n=1}^{N} r_j^{2,n}, \tag{21}$$

$$a_i^2 = \frac{1}{N}\sum_{n=1}^{N} a_i^{2,n}. \tag{22}$$

A known order, i, is called *movable* if $\pi_i^{1,-} < 0$ when $\delta_i = 1$ or $\pi_i^{2,-} < 0$ when $\delta_i = 0$. That is, rescheduling a movable order from dispatch 1 to 2 or from dispatch 2

Step 0: *Initialization*
Initialize the values of δ_i, $i \in \mathcal{N}_d$.
Create N samples of stochastic orders.
Set $\ell = 1$.

Step 1: *Solving the stage 1 problem*
Solve the deterministic VRP for the orders with $\delta_i = 1$.

Step 2: *Estimation of dual prices*
For each order i with $\delta_i = 1$, estimate r_i^1.
For each order i with $\delta_i = 0$, estimate a_i^1.
For scenario ω_n, estimate $a_i^{2,n}$ for order i with $\delta_i = 1$ and estimate $r_j^{2,n}$ for order j with $\delta_j = 0$
Update $\pi_i^{1,-}$ and $\pi_j^{2,-}$ using Eqs. (19), (20), (21) and (22).

Step 3: *Selection of movable orders*
Set $m = \min\{m^{\max},\ m^{1,-} + m^{2,-}\}$.
Obtain $\mathcal{Q}(m)$.

Step 4: *Order exchange*
For each $i \in \mathcal{Q}(m)$, set $\delta_i = 1 - \delta_i$.

Step 5: *Termination*
Set $\ell = \ell + 1$.
Solve the deterministic VRP for the orders with $\delta_i = 1$.
If $m^{1,-} > 0$, $m^{2,-} > 0$ and $\ell < \ell^{\max}$, go to **Step 1**.

Figure 2 Sketch of the temporal order exchange procedure (TORE).

to 1 can produce savings. Suppose that all movable orders are ranked by increasing value of the cost changes, $\pi_i^{1,-}$ and $\pi_i^{2,-}$ (that is, the largest cost reduction first). Let

$m^{1,-}$ = Number of movable orders with $\pi_i^{1,-} < 0$,
$m^{2,-}$ = Number of movable orders with $\pi_i^{2,-} < 0$,
$\mathcal{Q}(m)$ = Set of the first m ranked movable orders.

Furthermore, we define several control parameters:

ℓ = Iteration counter,
$\ell^{\max}$ = Maximum number of iterations allowed,
$m^{\max}$ = Maximum number of orders being exchanged per iteration.

The key steps of the TORE procedure are summarized in Figure 2 and we describe the details as follows.

Step 0: *Initialization*. In this step, we decide which deterministic orders to place on the first dispatch through initializing the values of δ_i for each $i \in \mathcal{N}_d$. One way is to use the First-Come-First-Served strategy. That is, according to the increasing sequence of order arrival times, we set $\delta_i = 1$ until the vehicle capacity constraint or trip time constraint are violated. Another way is to utilize a deterministic VRP solution approach (see below) to select the orders from the known order set, with the objective of minimizing the operational cost of stage 1. That is, we consider all known orders together for this approach and determine the orders scheduled in stage 1. We set $\delta_i = 1$ if order i is scheduled in stage 1 and set $\delta_i = 0$ otherwise. Further details can be found in the subsequent section.

Step 1: *Solving the stage-one problem.* In this step, we solve a deterministic VRP. Note that the main focus in this paper is to assess the value of considering stochastic information rather than how to solve deterministic VRP. Thus, we use a rather standard and efficient solution approach (see, for example, [23]). First, we construct the initial routes by the least-cost insertion method. Next, we use the 2-opt local search route improvement technique to improve the sequence of the orders being served. Finally, we employ an inter-route exchange of orders to reduce the cost further. In our cases, the vehicles have different capacities. For example, in the supplier case, there are three types of vehicles. We follow the industry practice that larger vehicles are used first with the consideration of taking advantage of the vehicle capacity to reduce the unit cost.

Step 2: *Estimation of dual prices.* In this step, we estimate r_i^1, a_j^1, $a_i^{2,n}$, and $r_j^{2,n}$. The estimations can be done by solving a sequence of VRPs for the N samples. However, the computational effort is very high. In our approach, we use an approximation strategy. Note that each of the estimators is affected by the change in travel distance, the change in the set-up cost at the customer locations and the change in the penalty cost due to the use of a super vehicle. First, we estimate the change in travel distance by adding order i to or removing order i from a dispatch based on the least cost insertion rule. For example, when adding order i to dispatch 2 for each sample n, we consider all adjacent order pairs in dispatch 2 to insert order i in between, which leads to a minimal travel distance increment. Second, to estimate the change in the set-up cost at the location, we check the location for each order i. If we add order i to a dispatch at some location where no other orders are placed, a fixed set-up cost will be incurred in this dispatch. Otherwise, if there are some other orders at this location, then no additional set-up cost will be incurred. Third, taking the cost of using a super vehicle into account, we notice that there are two contributing factors: one is the trip time and the other is the vehicle capacity. We measure the change of the trip time and the vehicle capacity when adding order i to or removing order i from a vehicle dispatch and then in turn estimate the total demand size that cannot be served by the regular vehicles, which therefore implies the cost of using the super vehicle.

Step 3: *Selection of movable orders.* We delay order i in dispatch 1 if $\pi_i^{1,-} < 0$ or move order i from dispatch 2 to dispatch 1 if $\pi_i^{2,-} < 0$. We select the orders with the largest savings to move. Moreover, because the estimations of r_i^1, a_j^1, $a_i^{2,n}$, and $r_j^{2,n}$ are based on adding or removing individual orders, the estimations can be poor if too many orders are added or removed simultaneously in an iteration. Thus, the maximum number of movable orders is limited by $m^{\max}$.

Step 4: *Order exchange.* In this step, after the order exchange, we update δ_i for each $i \in \mathcal{N}_d$. For each order, i, moved from dispatch 2 to dispatch 1, we update δ_i from 0 to 1. Similarly, for each order moved from dispatch 1 to dispatch 2, we update δ_i from 1 to 0.

Step 5: *Termination.* In this step, we provide two termination conditions. One criterion limits the total number of iterations. That is, if $\ell > \ell^{\max}$, then we terminate the procedure. The other criterion terminates the procedure if the number of movable orders becomes 0. That is, if $m^{1,-} = 0$ or $m^{2,-} = 0$, then we terminate the procedure.

5 Numerical Experiments

In this section, we report the results of computational experiments and in turn illustrate the benefits of considering stochastic order information. Section 5.1 describes the testing problems in the experiment. Section 5.2 introduces the methods being compared. Finally, experimental results and their interpretations are presented in Section 5.3.

5.1 Testing Problems

In our experiment, we have ten sets of problems: six sets are based on the modified Solomon's [21] benchmark problems and four sets are adapted from real application problems.

In Solomon's data sets, there are three groups of problems, differing in their geographical characteristics. They are the clustered group "c," the random group "r," and mixed group "rc." In each group, there are two subgroups differing in order size and service time features. They are denoted by three-digit numbers beginning with "1" and "2," respectively. The first problem from each subgroup in Solomon's data set is used, resulting in six testing problems. Each of them has 100 customers. We randomly select 70 of these 100 locations and denote them as known orders with the same order size as in the data file. No two known orders are at the same location. On the other hand, from the initial 100 customer locations, we randomly select 50 of them as the locations for the stochastic orders. The probability for a customer to place an order is uniformly distributed in [0.4, 0.8]. Note here that the locations of stochastic orders might be the same as the ones of known orders. In each problem, four vehicles are being used and each of them has capacity of 180. By assuming that the vehicle speed is one, the magnitudes of distance and travel time are identical. Time is measured in *minutes* and the order size is measured in *kg*. According to the characteristics of different problems, the vehicle duty time is set as 3, 000 in problems "c101" and "c201" and as 750 for the rest. Moreover, the earliest start time of the second dispatch Q is set as 40% of the total vehicle duty time. Let τ_j represent the set-up cost at location j. The costs are therefore calculated as

$$c_{ij,k} = \alpha_t t_{ij} + \alpha_s s_j + \tau_j \tag{23}$$

for a regular vehicle, $k > 0$, and

$$c_{ij,0} = \alpha_d d_j \tag{24}$$

for the super vehicle, 0. The cost, $c_{ij,0}$, represents the penalty for unserved orders and depends on order size. In aligning the set-up cost to real industrial practice, we set α_t = \$1/min, α_s=\$0.1/min, τ_j=\$25 and α_d=\$0.15/kg in all problem sets. Note that τ_j here denotes the shared fixed set-up cost at the corresponding customer location. Suppose that there are three orders, j_1, j_2, j_3, located in the same customer location, J, and the fixed set-up cost in this customer location is τ_J. Then, $\tau_{j_1} = \tau_{j_2} = \tau_{j_3} = \tau_J/3$.

For the supplier case and the distributor case, we create testing problems as follows. In the supplier case, the first dispatch of a vehicle is called the regular dispatch while the second one is called the urgent dispatch. Currently, the regular dispatches are used for known orders while the urgent dispatches are used for stochastic orders. The geographical characteristic in this problem is similar to one of the problems in Solomon's "rc" group. We generate the probabilities of the stochastic orders to appear on a day based on one month historical data. There are three large-size vehicles that normally deliver ten orders each, two middle-size vehicles that normally deliver seven orders each, and three small vehicles that normally deliver five orders each. On average, there are about 110 orders (including the realized stochastic orders) per day. In the distributor case, the whole region is divided into 20 non-overlapping zones. The operations in different zones are quite independent and there is usually one vehicle used for one zone. Similarly, we create the sets of stochastic orders based on one month's historical data. According to the historical data, approximately 35% of orders delivered in a day are unknown when the first dispatch takes off. We set the order size for a customer as the average order size of this customer over a month. Normally, a vehicle could deliver 30–40 orders per dispatch.

For both the supplier and the distributor cases, we are interested in evaluating the effectiveness of the methods under a high-demand situation, in which the size of each order is increased by a percentage that is uniformly distributed in the interval [10%, 20%]. In total, there are four problem data sets: a normal demand pattern and a high demand pattern for both the supplier case and the distributor case, respectively.

For all these four problem data sets, the duty time for each drive is set at ten hours per day and the earliest departing time of second dispatch is four hours after the first dispatch starts. The travel time and service time are measured in *minutes* and the order size in *kgs*. We set the values of α_t, α_s, α_d and τ_j to be the same as those used for the modified Solomon's data sets, except $\alpha_d = 0.015$ in the supplier case, where the delivery product is gasoline and the weight of the demand is on a different scale.

5.2 Methods for Comparison

We compare two deterministic methods and the TORE procedure. The deterministic methods do not use the stochastic order information. The first deterministic method is the First-Come-First-Served (FCFS) method described earlier. Due to its logical simplicity and customer oriented fairness, the FCFS method is widely used in practice. In this method, we schedule the known orders in dispatch 1 in the sequence of their arrival times until the vehicle capacity constraint and the trip time constraint are violated. The remaining known orders (together with the stochastic orders) are then scheduled in dispatch 2.

The second deterministic method utilizes all information about the known orders and it is called the Selection (SELECT) method. This method aims at minimizing the total cost of dispatch one by considering all known orders. In terms of the dispatch-one cost, it is clear that the SELECT method will be better than the FCFS method. For the TORE procedure, we use both the FCFS method and SELECT method to obtain the initial allocation of the known orders in the two dispatches (in *Step 0*). In

Table I Results on the modified Solomon's data

	FCFS (a)	TORE(F) (b)	$\frac{(a)-(b)}{(a)}$ (%)	SELECT (c)	TORE(S) (d)	$\frac{(c)-(d)}{(c)}$ (%)
c101	3,398.55	2,618.15	22.96	3,518.25	2,929.40	16.74
c201	3,487.90	2,685.05	23.02	3,627.70	2,888.55	20.38
r101	3,377.55	2,677.60	20.72	3,500.55	2,956.25	15.55
r201	3,400.30	2,709.10	20.33	3,484.90	2,978.90	14.52
rc101	3,112.25	2,463.75	20.84	3,228.35	2,651.75	17.86
rc201	3,115.30	2,510.20	19.42	3,246.10	2,763.50	14.87
Average	3,315.31	2,610.64	21.22	3,434.31	2,861.39	16.65

our experiment, we set $N = 30$ (that is, 30 samples are used in *Step 2*), $m^{max} = 2$, and $\ell^{\max} = 5$.

5.3 Experimental Results

For each problem set, we generate 20 instances. Each instance is referred to as a set of known orders and a set of stochastic orders according to the characteristics of the problem set. For an instance of a particular problem set, after we solve the two-stage problem and implement the first-stage solution, we create replications of the second stage orders and obtain the average second-stage cost. The sum of the first-stage cost and the average second-stage cost is an estimation of Z for that instance. For each problem set, we compute the average value of Z among 20 instances. For the modified solomon's data sets, 200 replications are used to evaluate the expected cost of the second-stage VRP. Thus, for each of these problems, more than 10,000 second-stage VRP are solved for evaluating the expected cost. For real problem sets, we use 20 replications.

The experimental results are presented in Tables I and II. Table I reports the results on the modified Solomon's data while Table II presents the results on the real problem data set. In each table, the first four columns indicate the problem names, the average total costs obtained by FCFS, the costs obtained by TORE initiated by FCFS, and the advantage of TORE over FCFS, respectively. Besides this, the fifth

Table II Results on real problem data

	FCFS (a)	TORE(F) (b)	$\frac{(a)-(b)}{(a)}$ (%)	SELECT (c)	TORE(S) (d)	$\frac{(c)-(d)}{(c)}$ (%)
S-N	3,014.20	2,712.30	10.02	2,943.30	2,537.15	13.80
S-H	3,062.05	2,775.95	9.34	2,937.30	2,557.10	12.94
D-N	2,547.35	2,424.20	4.83	2,271.35	2,056.80	9.45
D-H	2,540.75	2,379.40	6.35	2,298.20	2,061.60	10.30
Average	2,791.09	2,572.96	7.64	2,612.54	2,303.16	11.62

Table III Vehicle capacity utilization

	FCFS	TORE(F)	SELECT	TORE(S)
Random (stage 1)	0.88	0.76	0.99	0.84
Random (stage 2)	0.99	0.99	0.97	0.99
Real (stage 1)	0.90	0.90	0.99	0.71
Real (stage 2)	0.92	0.94	0.91	0.97

column gives the average total costs obtained by SELECT, the sixth column gives the costs obtained by TORE initiated by SELECT, and the last column reports extra savings of TORE over SELECT.

From Table I, we can see that for the modified Solomon's data set, TORE has significant benefits over the deterministic methods. More specifically, on average, TORE produces a 21.22% savings over FCFS and a 16.65% over SELECT.

One interesting observation is that SELECT does not outperform FCFS. This reflects the insight that a better deterministic method does not necessarily do better in a stochastic environment. In fact, it does worse in most testing problems. Without taking the future information into consideration, a better solution in the current stage might result in a worse situation in the next stage.

Table II shows the results when using real data sets. A problem is characterized by its application ("S" for the supplier case and "D" for the distributor case) and its demand pattern ("N" for normal and "H" for high). Similar to the random problem cases, TORE can produce a substantial savings although not as high as those for the random test problems. On average, it produces a 7.64% savings over FCFS and an 11.62% savings over SELECT. There are some interesting observations. First, SELECT is better than FCFS for the real data sets as opposed to FCFS is better for the random problem tests. There may be some problem characteristics (such as some dependency among the parameters or orders) that are not captured in the randomly generated data sets. Second, the solution quality of TORE is dependent on the initial solution. In the case of D–N, for example, the solution of SELECT is even better than the solution of TORE(F).

Table III reports the average capacity utilization of the vehicles. In stage 1, for the two deterministic initial solutions, SELECT achieves a higher utilization as it

Table IV CPU seconds for each method to solve one problem

	FCFS	TORE(F)	SELECT	TORE(S)
c101	0.34	170.52	1.10	162.31
c201	0.32	178.79	1.09	163.49
r101	0.66	292.02	1.67	230.29
r201	0.80	291.38	1.89	235.55
rc101	0.41	192.10	1.19	176.50
rc201	0.49	168.89	1.24	144.35
S-N	9.05	180.79	13.66	134.97
S-H	7.20	135.85	11.63	97.94
D-N	7.68	604.62	24.56	385.73
D-H	3.88	609.28	21.19	309.67

considers all known orders for dispatch 1. On the other hand, the heuristics usually lead to a lower capacity utilization as some orders are purposely delayed to dispatch 2 to save cost.

Table IV reports the average CPU time used to solve each problem set over 20 instances by each method. All the computational experiments were carried out on a PC with 2.54 GHz CPU and 1, 024 M memory. The figures are in units of *seconds*. The third column presents the CPU time needed for TORE with the initial solution generated by FCFS and the fifth column reports the CPU time for TORE with initial solution obtained by SELECT.

From Table IV, we can see that solving the "D" case needs much more computational time than the other case. This is because, in "D" case, a vehicle normally delivers 40–50 orders and, in other cases, a vehicle delivers 10–20 orders. When solving the VRP, constructing and improving longer routes takes much more time than working on shorter routes. Note that the VRP problems with 10–20 orders can be solved optimally in reasonable time. However, we use the same heuristic even for small problems to have a consistent comparison.

6 Conclusion

We considered VRP with multiple dispatches. In such a problem, before scheduling the first dispatch, there are uncertain orders that may need to be delivered within the same day. One class of decisions is to determine which known orders should be delayed in order to minimize the total expected cost over the planning horizon. We developed a two-stage stochastic programming formulation for the problem, performed a worst-case analysis for the formulation, developed a heuristic and conducted numerical experiments using both randomly generated problems (modified from the standard test problems in the literature) and real data sets.

The experimental results are encouraging. In applying our method, we observe that utilizing stochastic future information can bring significant cost savings. However, there are still several possible improvements. First, instead of computing the cost savings on moving individual orders from dispatch 1 to 2 or vice versa, we may compute the savings on moving the whole group of orders in a cluster and exchanging them accordingly. Second, to avoid moving a particular order back and forth in the optimization process, one possible change is to move an order only if the corresponding savings is larger than a threshold. Third, in the current methods, we compute r_i^1 and a_i^2 from scratch in each iteration. One possible variation is to use the weighted average of the values of these estimators over several iterations. Nevertheless, these improvements are primarily for increasing the speed of the heuristic. We need to evaluate the tradeoffs between speed and quality. In terms

Table V Results on applying sample average approximation method

Iteration	M	n	N	$\lvert g\rvert$	$\lvert\bar{v}_M\rvert$	$\lvert t_{\alpha/2,M+N-1}S\rvert$	CPU time (min)
1	20	20	20	337.30	5,314.97	102.67	24.03
2	40	40	40	348.24	5,145.00	73.87	93.07
3	80	80	80	328.02	4,980.36	53.70	341.22

of modelling, one possible extension is to consider a multiple dispatch model where orders can have different types of urgency. Some orders must be delivered in the coming dispatch, while some can be delayed for more than one dispatch. Tackling such a problem becomes more challenging and deserves more research effort.

Appendix

This appendix reports some preliminary results using the sample average approximation (SAA) method to solve our problem (Table V). The key steps of the algorithm are summarized in Figure 3 for easy reference. Besides the notation introduced in Section 2, let $g(y^1, \omega^i)$ be the objective value corresponding to scenario ω^i. In the

---

Step 1: For $m = 1, \dots, M$, repeat the following step:
(a) Generate i.i.d. random vectors $\omega^1, \dots, \omega^n$.
(b) Solve the SAA problem

$$\min \ g(y^1) = \sum_{k=1}^{K} f^1(y_k^1) + n^{-1} \sum_{j=1}^{n} \left(\sum_{k=1}^{K} f^2\left(y_k^2, \omega^j\right) \right),$$

and let $\widehat{y}^m$ be the solution vector of y_k^1, $\widehat{v}^m$ be the optimal objective value.
(c) Generate i.i.d. random variables $\omega^1, \dots, \omega^N$. Obtain an estimate of the objective function value of $\widehat{y}^m$

$$g_N(\widehat{y}^m) := f^1(\widehat{y}^m) + \frac{1}{N} \sum_{i=1}^{N} f(y_k^2, \omega^i),$$

and an estimate of the variance of the estimator

$$S_N^2(\widehat{y}^m) := \frac{1}{N-1} \sum_{i=1}^{N} \left[g(\widehat{y}^m, \omega^i) - g_N(\widehat{y}^m) \right]^2.$$

Step 2: Evaluate $\bar{v}_M = \frac{1}{M} \sum_{m=1}^{M} \widehat{v}^m$ and $S_M^2 = \frac{1}{M-1} \sum_{m=1}^{M} (\widehat{v}^m - \bar{v}_M)^2$.

Step 3: For each solution $\widehat{y}^m$, an estimate of the optimality gap is given by $(g_N(\widehat{y}^m) - \bar{v}_M)$ with a variance estimate of

$$S^2 = \frac{S_N^2(\widehat{y}^m)}{N} + \frac{S_M^2}{M}.$$

Step 4: Select the solution with the smallest optimality gap $g = (g_N(\widehat{y}^m) - \bar{v}_M)$ and calculate the corresponding 95% confidence interval $[\bar{v}_M - t_{\alpha/2, M+N-1} S,\ \ \bar{v}_M + t_{\alpha/2, M+N-1} S]$.

Step 5: If the optimality gap satisfies termination tolerances ($|g| < \epsilon_1$ and $t_{\alpha/2, M+N-1} S < \epsilon_2$), terminate with the current solution. Otherwise, if $|g| > \epsilon_1$, set $M = 2M$ to reduce gap; else if $t_{\alpha/2, M+N-1} S > \epsilon_2$, set the numbers $N = 2N$ or $n = 2n$ to reduce the variance, and go to Step 1.

Figure 3 Sketch of sample average approximation algorithm.

SAA method, M is the number of replications, n is the sample size for the SAA problem, and N is the sample size to estimate the objective value of a solution. Table V shows the numbers of replications and samples used for the first three iterations for a problem based on the "c101" class. The experiment was carried out in the same computational environment as were the experiments described in Section 5. The two critical values, $|g|$ and $|t_{\alpha/2,M+N-1}S|$, did show the tendency of convergence. However, it took around six hours for only three iterations. Therefore, some adjustment may be required for speeding up the SAA method when it is applied to our problem.

Acknowledgements We would like to thank the three referees for their valuable comments and suggestions. The research was supported by Grant HKUST6217/02E of the Research Grants Council of Hong Kong.

References

1. Bertsimas, D.J., Simchi-Levi, D.: A new generation of vehicle routing research: robust algorithms, addressing uncertainty. Oper. Res. **44**, 216–304 (1996)
2. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, Berlin Heidelberg New York (1997)
3. Cheung, R.K., Powell, W.: SHAPE: a stochastic hybrid approximation procedure for two-stage stochastic programs. Oper. Res. **48**, 73–79 (2000)
4. Cheung, R.K., Hang, D.D.: A time-window sliding procedure for driver-task assignment with random service times. IIE Trans. **35**, 433–444 (2003)
5. Dror, M.: Vehicle routing with stochastic demands: models & computational methods. In: Dror, M., L'Ecuyer, P., Szidarouszky, F. (eds.) Modeling Uncertainty: An Examination of Stochastic Theory, Methods and Applications. International Series in Operations Research and Management Science, vol. 46, Kluwer, Boston, Massachussetts (2002)
6. Dror, M., Laporte, G., Trudeau, P.: Vehicle routing with stochastic demands: properties and solution frameworks. Transp. Sci. **23**, 166–176 (1989)
7. Ermoliev, Y.: Stochastic quasigradient methods. In: Ermoliev, Y., Wets, R. (eds.) Numerical Methods in Stochastic Programming, pp. 143–185. Springer, Berlin Heidelberg New York (1988)
8. Gendreau, M., Laporte, G., Seguin, R.: An exact algorithm for the vehicle-routing problem with stochastic demands and customers. Transp. Sci. **29**, 143–155 (1995)
9. Gendreau, M., Laporte, G., Seguin, R.: A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. Oper. Res. **44**, 469–477 (1996)
10. Haughton, M.A.: Quantifying the benefits of route reoptimisation under stochastic customer demands. J. Oper. Res. Soc. **51**, 320–332 (2000)
11. Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**, 479–502 (2001)
12. Laporte, G., Louveaux, F.V.: Formulations and bounds for the stochastic capacitated vehicle routing with uncertain supplies. Economic Decision-Making: Games, Econometrics and Optimization, pp. 443–455. North Holland, Amsterdam, The Netherlands (1990)
13. Laporte, G., Louveaux, F.V., Mercure, H.: Models and exact-solutions for a class of stochastic location-routing problems. Eur. J. Oper. Res. **39**, 71–78 (1989)
14. Laporte, G., Louveaux, F.V., van Hamme, L.: An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. Oper. Res. **50**, 415–423 (2002)
15. Larsen, A., Madsen, O., Solomon, M.: Partially dynamic vehicle routing – models and algorithms. J. Oper. Res. Soc. **53**, 637–646 (2002)
16. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Network Routing, Handbooks in Operations Research and Management Science, vol. 8, Elsevier, Amsterdam, The Netherlands (1995)
17. Psaraftis, N.: Dynamic vehicle routing: status and prospects. Ann. Oper. Res. **61**, 143–164 (1995)
18. Savelsbergh, M.W.P., Goetschalckx, M.: A comparison of the efficiency of fixed versus variable vehicle routes. J. Bus. Logist. **16**, 163–187 (1995)

19. Secomandi, N.: A rollout policy for the vehicle routing problem with stochastic demands. Oper. Res. **49**, 796–802 (2001)
20. Secomandi, N.: Analysis of a rollout approach to sequencing problems with stochastic routing applications. J. Heuristics **9**, 321–352 (2003)
21. Solomon, M.: Algorithms for the vehicle routing and scheduling problems with time windows. Oper. Res. **35**, 254–265 (1987)
22. Stewart, W.R., Golden, B.L.: Stochastic vehicle-routing: a comprehensive approach. Eur. J. Oper. Res. **14**, 371–385 (1983)
23. Toth, P., Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania (2002)
24. Verweij, B., Ahmed, S., Kleywegt, A., Nemhauser, G.L., Shapiro, A.: The sample average approximation method applied to stochastic routing problems: a computational study. Comput. Optim. Appl. **24**, 289–333 (2003)
25. Waters, C.D.J.: Vehicle-scheduling problems with uncertainty and omitted customers. J. Oper. Res. Soc. **40**, 1099–1108 (1989)
26. Yang, W.H., Mathur, K., Ballou, R.H.: Stochastic vehicle routing problem with restocking. Transp. Sci. **34**, 99–112 (2000)