Solving the Vehicle Routing Problem with Stochastic Demands using the Cross Entropy Method

Krishna Chepuri
Decision Analysis & Portfolio Management
J&J PRD
Titusvile, NJ 08560
KChepuri@PRDUS.JNJ.com

Tito Homem-de-Mello
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208
tito@northwestern.edu

October 9, 2003

Abstract

An alternate formulation of the classical vehicle routing problem with stochastic demands (**VRPSD**) is considered. We propose a new heuristic method to solve the problem. The algorithm is a modified version of the so-called *Cross-Entropy method*, which has been proposed in the literature as a heuristics for deterministic combinatorial optimization problems based upon concepts of rare-event simulation. In our version of the method, we incorporate Monte-Carlo sampling in order to estimate the objective function at each point in the domain. Many practical issues arise form this new feature, especially the decision as to *when* to draw new samples and *how many* samples to use. We address these issues and propose a formal algorithm, which is used to solve the underlying **VRPSD**. We also develop a framework for obtaining exact solutions and tight lower bounds for the problem under various conditions, which include specific families of demand distributions. This is used to assess the performance of the heuristics. Finally, numerical results are presented for various problem instances to illustrate the ideas.

1 Stochastic Vehicle Routing

The classical vehicle routing problem (**VRP**) is defined on a graph $G = (V, A)$, where $V = \{v_0, v_1, \dots, v_n\}$ is a set of vertices and $A = \{(v_i, v_j) : i, j \in \{0, \dots, n\}, v_i, v_j \in V\}$ is the arc set. A matrix $L = (L_{i,j})$ can be defined on A, where the coefficient $L_{i,j}$ defines the distance between nodes v_i and v_j and is proportional to the cost of travelling the corresponding arc. There can be one or more vehicles starting off from the depot v_0 with a given capacity, visiting all or a subset of the vertices, and returning to the depot after having satisfied the demands at the vertices. The Stochastic Vehicle Routing Problem (**SVRP**) arises when elements of the vehicle routing problem are stochastic — the set of customers visited, the demands at the vertices, or the travel times. The solutions for the **VRP** problems no longer hold, considering the fundamental structural differences between the two. **SVRP** problems have been studied under different formulations: Travelling Salesman Problem with Stochastic Customers (**TSPSC**), Travelling Salesman Problem with Stochastic Travel Times (**TSPST**), Vehicle Routing Problems with Stochastic Demands (**VRPSD**), Vehicle routing problem with Stochastic Customers (**VRPSC**), Vehicle Routing Problem with Stochastic Customers and Demands (**VRPSCD**). For a survey of these methods and the techniques proposed for them, we refer to Gendreau et al. (1996a).

The predominant approach for solving the **SVRP** class of problems is to use some "here-and-now" optimization technique, where the sequence of customers to be visited is decided in advance. On the given route, if the vehicle fails to meet the demands of a customer, there is a recourse action taken. The recourse action could be in the form of going back to the depot to replenish and fulfilling the customers demand, and continuing with the remaining sequence of the customers to be visited or any other meaningful formulation. The problem then reduces to a stochastic optimization problem where the sequence with the minimum expected distance travelled (or equivalently, minimum expected cost) has to be arrived at. The alternative is to use a *re-optimization* strategy where upon failure at a node, the optimum route for the remaining nodes is recalculated. The degree of re-optimization varies. At one extreme is the use of a dynamic approach where one can re-optimize at any point, using the newly obtained data about customer demands, or to re-optimize after failure. Neuro Dynamic Programming has been used to implement techniques based on re-optimization; see, e.g., Secomandi (2000, 2001).

A strategy using some "here-and-now" optimization approach becomes meaningful if it is not feasible to adopt a re-optimization strategy due to resource constraints. Exact algorithms of that type for a number of **SVRPs** have been proposed by Laporte et al. (1989, 1992, 1994). The first algorithm for this was proposed by Tillman (1969) based on the Clark and Wright algorithm (Clark and Wright 1964). The chance constrained version of the problem, where the probability of failure must not exceed a certain threshold but the expected cost of the recourse action is not taken into account, has been studied by Golden and Yee (1979), Stewart Jr. and Golden (1983), and Laporte et al. (1989). Dror and Trudeau (1986) present a savings heuristic algorithm for this case. Bertsimas (1992) proposes an asymptotically optimal heuristic for the **VRPSD** problem with estimates of the average performance. In Bertsimas et al. (1995), Dynamic Programming is used to supplement the recourse actions taken in the earlier paper. Properties and formulations of the **VRPSD** based on a priori optimization have been also been investigated by Laporte and Louveaux (1990), Bastian and Kan (1992), and Trudeau and Dror (1992). Gendreau et al. (1995) solve the **VRPSCD** with demands following discrete probability distributions within the framework of stochastic integer programming, using an integer L-shaped algorithm to produce exact solutions. The demands at the vertices are taken to be discrete and independent. A similar method has been proposed by Hjorring and Holt (1999). Gendreau et al. (1996b) have proposed a heuristic based on Tabu Search with results comparing favorably with exact optimal solutions derived from Gendreau et al. (1995).

1.1 Problem Description

In this paper we focus on the vehicle routing problem with stochastic demands (**VRPSD**). A certain type of product is distributed from a plant to N customers, using a single vehicle, having a fixed capacity Q. The vehicle strives to visit all the customers periodically to supply the product and replenish their inventories. On a given periodical trip through the network, on visiting a customer, an amount equal to the demand of that customer is downloaded from the vehicle, which then moves to the next site. The demands of a given customer during each period are modelled as independent and identically distributed random variables with known distribution. A reasonable assumption is that all the customers' demands belong to a certain distribution (say normal) with varying parameters for different customers.

The vehicle sets out with the fixed capacity Q, and does not have knowledge of the demands that it will encounter on a given route, save for their individual probability distributions. Hence, there is a positive probability that the vehicle runs out of the product along the route, in which case the remaining demands of that customer and the remaining customers further along the route are not satisfied (a failure route). Such failures are discouraged with penalties, which are functions of the recourse actions taken. Each customer in the set can have a unique penalty cost for not satisfying the demand. The cost function for a particular route travelled by the vehicle during a period is calculated as the sum of all the arcs visited and the penalties (if any) imposed. If the vehicle satisfies all the demands on that route, the cost of that route will simply be the sum of the arcs visited including the arc from the plant to the first customer visited, and the arc from the last customer visited back to the plant.

Alternatively, if the vehicle fails to meet the demands of a particular customer, the vehicle heads back to the plant at that point, terminating the remaining route. The cost function is then the sum of all the arcs visited (including the arc from the customer where the failure occurred back to the plant) and the penalty for that customer. In addition, the penalties for the remaining customers who were not visited will also be imposed. Thus, a given route can have a range of cost function values associated with it. The objective is to find the route for which the *expected value* of the cost function is minimum compared to all other routes.

The problem above formulated is common to some specific industries where a single transportation entity caters to the demands of a limited (usually not greater than 25) number of customers for a single product. These customers must be served periodically and hence expected values of the cost function become important. The time frame within which demand must be met is critical. Failure to meet demand at certain points can result in lost revenue and/or emergency deliveries to those customers. This motivates our use of *penalties* that are incorporated to the cost function. Common industries where such type of problems occur include serving a group of restaurants, delivering products to hospitals/ pharmacies and delivering gases to industries. In fact, what motivated us to work on this problem were the issues faced by a large distributor of liquid air.

1.2 Solution Method

Studied from the point of view of the "here-and-now" optimization approach, i.e. deciding the sequence of vertices to be visited in advance and independent of the demands encountered, the problem can be cast as a *discrete stochastic optimization* problem with respect to the discrete set of finite routes that can be taken. The class of discrete stochastic optimization problems we are considering here are those of the form

$$\min_{r\in R}\{G(r) := \mathbb{E}[H(r, D_r)]\}, \tag{1}$$

where D_r is a random variable that may or may not depend on the parameter r, H is a deterministic, real valued function and R is the discrete and finite (or countably infinite) feasible set of values that r can take. Thus, $H(r, D_r)$ is a random variable whose expected value, $G(r)$ is usually estimated by Monte Carlo simulation or experimentation, since an analytical expression is elusive. The global optimum solution set can then be denoted by

$$R^* = \{r^* \in R : G(r^*) \le G(r)\ \forall\, r \in R\}.$$

To solve the above problem, we use a new combination of Monte-Carlo techniques and the *Cross Entropy (CE) method*, (referred henceforth as the CE method) developed by Rubinstein (1999, 2002). The basic idea of CE method is to connect the underlying optimization problem to a problem of estimating *rare-event probabilities*, and use tools derived for that class of problems. The method has been shown to work quite well in the context of deterministic optimization; see, e.g., the tutorial paper by de Boer et al. (2003) and references therein. Here we adapt the method to the context of discrete *stochastic* optimization. The result is a new algorithm on its own — indeed, we view that as one of the main contributions of the paper, see section 1.4.

1.3 Literature on Discrete Stochastic Optimization

Since the parameter space is discrete, methods for continuous stochastic optimization such as the stochastic approximation method which rely on derivatives cannot be applied to such class of problems. Ranking-and-selection and multiple comparison procedures have been proposed by Hochberg and Tamhane (1987), Bechhofer and Santner (1995), and Hsu (1996) for cases where the set of feasible solutions is small.

Global search methods via Markov Chains have been proposed by Yan and Mukai (1992), Andradottir (1995, 1996), Alrefaei and Andradottir (2001). An adaptation of simple random search to the stochastic optimization context is studied by Homem-de-Mello (2003), who also discuss convergence issues. There have been also a number of heuristic techniques based on simulated annealing to solve discrete stochastic optimization problems. Gelfland and Mitter (1989) show that when the noise is $N(0, \sigma^2)$ and under a suitable temperature scheme, the simulated annealing procedure applied to the stochastic version converges with the same probability as for the deterministic case to the set of global optimum solutions. Gutjahr and Pflug (1996) show that Simulated Annealing algorithm can be applied directly to solve the discrete optimization problems by using sampling to reduce the variance. They showed that if the noise is symmetric around 0 and sharply peaked, the convergence is similar to that of the algorithm applied to a deterministic problem. Their scheme requires that sampling be increased in successive iterations to reduce the variance of the noise. Homem-de-Mello (2001) obtains a similar result but without the constraint on the distribution of noise. Alrefaei and Andradottir (1999) apply the simulated annealing procedure with a constant temperature scheme to solve the discrete stochastic optimization problem, while avoiding excessive sampling to reduce the overall number of computations. For other discrete stochastic optimization methods based on simulated annealing, we refer to Fox and Heine (1995) and Gutjahr et al. (1999).

There has also been work on using heuristic procedures based on genetic algorithms to solve discrete stochastic optimization problems. For work on using genetic algorithms, we refer to Aizawa and Wah (1994). Allen and Ittiwattana (2002) have proposed a procedure for general simulation optimization that can be extended to the domain of discrete stochastic optimization problems. Zhai et al. (1996) study the convergence properties of schemes using genetic algorithms to solve such problems. A different approach is taken by Pichitlamken and Nelson (2002), who propose a method that combines global guidance, local search and selection procedures used in simulation.

1.4 Discussion of contributions

We discuss now the contributions of this work. They are two-fold, in the sense that part of the contributions relates to the vehicle routing problem and part relates to the Cross-Entropy technique.

From the viewpoint of vehicle routing problems, we propose a new heuristic method to solve a particular class — the **VRPSD** — as an alternative to existing techniques. In addition, the bounds and exact formulations derived in section 3 are, to the best of our knowledge, new.

We also contribute to the ongoing development of the CE method. Our contribution here has two aspects: first, we formalize the use of a trajectory generation procedure in routing problems as a mechanism of sampling from some distribution. Although trajectory generation methods have been proposed before (see, e.g., de Boer et al. 2003, Margolin 2002, Rubinstein 1999, 2002), the link between such procedures and sampling had not been fully established in our view. Our formulation provides such link, which also allows us to prove the correctness of the algorithm; we refer the reader to section 4 for details.

Another contribution of this work to the CE area is the development of a CE-based algorithm for *stochastic* optimization. Notice that here we use this term to refer to problems of the form (1) — i.e., problems where some of the data are random with known distributions. The reason we emphasize this point is that "stochastic optimization" is sometimes used to refer to *probabilistic algorithms*, i.e., algorithms for deterministic problems that incorporate some randomization. Examples of methods in the latter category are the original CE method as well as simulated annealing and pure random search. The distinction, of course, is not just semantical — stochastic optimization *problems* have a completely different nature than deterministic ones and are in general harder to solve.

As discussed in section 4.2, one way to deal with the uncertainty in stochastic optimization problems is to use Monte Carlo sampling to approximate expected values. Although it is intuitive that using large sample sizes should yield good approximating solutions — see, e.g., Kleywegt et al. (2001), Homem-de-Mello (2003) for comprehensive discussions — in practice the choice of the sample size is critical to the performance of the algorithm, since a very large sample may unnecessarily slow down the convergence to the optimal solution. *The method we propose balances the effort between Monte Carlo sampling and the trajectory generation inherent to the CE algorithm.* Such an approach is new, since previous papers such as Allon et al. (2003) that consider CE for stochastic optimization problems do not touch the issue of choice of sample sizes.

2 Problem Formulation

The problem can be defined on a graph $G = (V, A)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of $n + 1$ vertices and $A = \{(v_i, v_j) : i \neq j; v_i, v_j \in V\}$ is the arc set. As before, define $L_{i,j}$ as the distance between v_i and v_j.

Define the route r as the sequence of vertices visited in V always starting with v_0 and coming back to v_0. That is, $r := (r_0, r_1, \ldots, r_n, r_{n+1})$ with r_0, r_{n+1} always representing v_0, the starting node.

During a particular route r, denote the surplus quantity of the product in the vehicle after delivery at a node r_i as $Sur(r_i)$, where D_{r_i} denotes the demand at node r_i. Set $Sur(r_0)$ as Q, the initial quantity in the vehicle. Then, define $Sur(r_i)$ as:

$$Sur(r_i) := \begin{cases} Sur(r_{i-1}) - D_{r_i} & : \quad \text{if } Sur(r_{i-1}) > 0 \\ -\infty & : \quad \text{if } Sur(r_{i-1}) \leq 0 \end{cases} \tag{2}$$

In (2), $Sur(r_i) = -\infty$ serves to indicate that the vehicle has failed to meet the demand at one of the earlier nodes.

Define the penalty scheme as

$$f(r_i) := \begin{cases} 0 & : \quad \text{if } Sur(r_i) > 0 \\ L(r_i, r_0) & : \quad \text{if } Sur(r_i) = 0 \\ \phi_{r_i} + L(r_i, r_0) & : \quad \text{if } -\infty < Sur(r_i) < 0 \\ \phi_{r_i} & : \quad \text{if } Sur_{r_i} = -\infty, \end{cases}$$

where ϕ_{r_i} — a number, not a random variable — denotes the penalty for failing to meet the demand at node r_i. Notice that ϕ_{r_i} includes the cost of traversing back to the depot from the node where the vehicle ran out of items to deliver. Define the indicator function

$$I(r_i) := \begin{cases} 0 & : \quad \text{if } Sur(r_i) \leq 0 \\ 1 & : \quad \text{if } Sur(r_i) > 0 \end{cases}$$

That is, $I(r_i) = 1$ if the vehicle has items in stock after servicing node r_i, so that it must travel to the next node.

We are now in a position to define the cost function of the route r as:

$$H(r, D_r) \ := \ \sum_{i=0}^{n} L(r_i, r_{i+1}) \cdot I(r_i) + \sum_{i=1}^{n} f(r_i), \tag{3}$$

where $D_r := [D_{r_1}, \ldots, D_{r_n}]$, the demands encountered at the nodes. The recourse action if the demand is not satisfied at a particular node is to simply terminate the route and go back to the depot and impose the penalties as per the scheme defined. Note that D_{r_i} is a random variable. Since the cost function H is dependent on the demands, the quantity $H(r, D_r)$ is also a random variable.

Let $G(r) := \mathbb{E}[H(r, D_r)]$ denote the expected value of the cost function. The "here-and-now" optimization problem of finding the route with minimum expected cost value can be written as

$$\min_{r \in R} G(r), \tag{4}$$

where R is the feasibility set of all the possible routes that can be constructed on the graph G. This is a **VRPSD**.

3 Lower Bounds and Exact Solutions

The problem formulated in the previous section lends itself to different approaches depending on the nature of the penalties and demands encountered at the nodes. It is reasonable to assume that the demands are positive valued independent random variables. Then, the problem can be categorized into four levels, each lending itself to a different type of analysis:

1. the demands are independent and identically distributed (iid), the penalties are identical;
2. the demands are iid, the penalties are non-identical;
3. the demands are non-iid, the penalties are identical;
4. the demands are non-iid, the penalties are non-identical.

In what follows, we will discuss each of these categories separately.

3.1 Demands are iid

Since the distribution of demand D_{r_i} is independent of node r_i, the probability of failure at the ith node is independent of the sequence taken. Consequently, it is possible to compute the cost due to the path taken analytically. Let C denote the capacity of the vehicle. Define the quantities

$$\begin{aligned} p_0 &:= 1 && (5) \\ p_i &:= P\left(\sum_{k=1}^{i} D_k < C\right) \qquad i = 1, \dots, n-1. && (6) \end{aligned}$$

The value p_i corresponds to the probability that the vehicle continues the trip from r_i to r_{i+1}. For the first $n-1$ nodes, this happens when the sum of the demands on the path up to node r_i is strictly less than the vehicle capacity C. Notice that if the vehicle goes from r_{n-1} to r_n then the trip from r_n to r_{n+1} will happen regardless of whether the demand at node r_n is met or not. Therefore, we can count that cost in the "cost of trip back to depot" below. The expected cost of traversing the arcs due to a route $r := \{r_0, r_1, \dots, r_{n+1}\}$ is then

$$T(r) := \sum_{i=0}^{n-1} p_i \cdot L(r_i, r_{i+1}). \tag{7}$$

Notice that if nothing is left in the vehicle after visiting node r_i, then the vehicle interrupts its journey and makes an unplanned trip back to the depot. Let p_i° denote the probability of that event. We have

$$\begin{aligned} p_1^\circ &:= P\left(D_1 \geq C\right) && (8) \\ p_i^\circ &:= P\left(\sum_{k=1}^{i} D_k \geq C, \sum_{k=1}^{i-1} D_k < C\right) \qquad i = 2, \dots, n-1 && (9) \\ p_n^\circ &:= P\left(\sum_{k=1}^{n-1} D_k < C\right). && (10) \end{aligned}$$

Node r_n is treated separately, since as remarked above the cost of travelling from r_n to $r_{n+1} \equiv r_0$ will incur whenever the vehicle reaches r_n. The expected cost of trips back to the depot is then

$$T_B(r) \;:=\; \sum_{i=1}^{n} p_i^{\circ} \cdot L(r_i, r_0). \tag{11}$$

To determine the expected cost of penalties, define

$$q_i \;:=\; P\left(\sum_{k=1}^{i} D_k > C\right) \qquad i = 1, \ldots, n.$$

The quantity q_i denotes the probability that the demand at node r_i is not met and hence a penalty incurs. Next, recall that the penalty at node r_i is defined as ϕ_{r_i}. The expected cost of penalties can then be formulated as

$$P(r) \;:=\; \sum_{i=1}^{n} q_i \cdot \phi_{r_i}. \tag{12}$$

By putting the above equations together, we have that the total expected cost of route r is given by

$$G(r) \;=\; \mathbb{E}[H(r, D_r)] \;=\; T(r) + T_B(r) + P(r) \tag{13}$$

so that the optimization problem is the one defined in (4).

3.1.1 Solving as an Integer Linear Programming Problem

The optimization problem defined in (4) can be cast as a quadratic integer problem and solved exactly. In what follows, the index $k \in \{0, \ldots, n\}$ refers to the position in the route and $i, j \in \{0, \ldots, n\}$ refer to the nodes. x_{ki} is a binary decision variable to indicate if node v_i is in the k'th position in the route. Then, problem (4) can be recast as

$$\min_x \sum_{k=1}^{n} p_{k-1} \sum_{i=0}^{n} \sum_{j=0}^{n} x_{(k-1)i} \cdot x_{kj} \cdot L(v_i, v_j) + \sum_{k=1}^{n} \sum_{i=1}^{n} (x_{ki} \cdot \phi_{v_i} \cdot q_k + p_k^{\circ} \cdot x_{ki} \cdot L(v_i, v_0)) \tag{14}$$

subject to the constraints

$$\sum_{i=0}^{n} x_{ki} \;=\; 1, \quad k = 0, \ldots, n+1 \tag{15}$$

$$\sum_{k=0}^{n} x_{ki} \;=\; 1, \quad i = 0, \ldots, n \tag{16}$$

$$x_{00} \;=\; 1. \tag{17}$$

The quadratic integer program given by (14)-(17) can be reduced to a Linear Integer Program in the standard fashion (see, e.g., Nemhauser and Wolsey 1988) by defining $W_{kij} := x_{(k-1)i} \cdot x_{kj}$. We obtain

$$\min_{x,W} \sum_{k=1}^{n} p_{k-1} \sum_{i=0}^{n} \sum_{j=0}^{n} W_{kij} \cdot L(v_i, v_j) + \sum_{k=1}^{n} \sum_{i=1}^{n} [x_{ki} \cdot \phi_{v_i} \cdot q_k + p_k^{\circ} \cdot x_{ki} \cdot L(v_i, v_0)] \tag{18}$$

subject to the constraints

$$\begin{aligned} W_{kij} &\leq x_{(k-1)i} && (19) \\ W_{kij} &\leq x_{kj} && (20) \\ W_{kij} &\geq x_{(k-1)i} + x_{kj} - 1 && (21) \end{aligned}$$

(for each $k \in \{1, \cdots, n+1\}$ and each $i, j \in \{0, \cdots, n\}$) along with constraints given by (15)-(17). This method can be used to obtain an exact solution for the problem at hand and provide a means to test the validity of the solution using the CE method.

3.2 Demands are non-iid, penalties are uniform

To simplify the analysis, we will assume that the cost of going back to the depot — because of failure at a node, or after serving all the nodes — can be disregarded. Such is the case, for example, when the depot is approximately equidistant from all nodes, of when that cost is low in comparison to penalty costs. If this is not the case, we can easily bound that cost from below by the minimum distance from any node back to the depot.

The analysis is complicated by the fact that the probability coefficients depend on the route since demands are non-iid. Hence, it is not possible to get an exact solution for the optimization problem. Instead, we shall endeavor to find a tight lower bound for this formulation as a means to assess the CE method's performance. In this context, define for a given route r,

$$\begin{aligned} p_0(r) &:= 1 && (22) \\ p_i(r) &:= P\left(\sum_{k=1}^{i} D_{r_k} < C\right), \quad i = 1, \ldots, n && (23) \\ q_i(r) &:= P\left(\sum_{k=1}^{i} D_{r_k} > C\right), \quad i = 1, \ldots, n, && (24) \end{aligned}$$

where as before D_{r_i} is the demand at node r_i for a route r. We note that $p_i(r)$ and $q_i(r)$ are analogous to the quantities p_i and q_i defined in the previous section, except that they are also functions of the route. Denote the identical penalties imposed at nodes by ϕ. Then, the expected cost function of any route r ignoring the expected cost of going back to the home depot can be formulated as

$$F(r) \ := \ \sum_{i=0}^{n-1} p_i(r) \cdot L(r_i, r_{i+1}) + \phi \cdot \sum_{i=1}^{n} q_i(r), \tag{25}$$

and the optimization problem can be restated as

$$\min_{r \in R} F(r). \tag{26}$$

We now present a method to obtain a tight lower bound for the problem defined in (26). Let us assume the following properties for the demands at the nodes:

1. The demands at the nodes are drawn from the same family of distributions $\mathcal{F}$ whose parameter depends on the node i, i.e., $D_i \sim \mathcal{F}(\mu_i)$.
2. Also, $\mathcal{F}$ has the property that $D_i + D_j \sim \mathcal{F}(\mu_i + \mu_j)$.
3. The demands can only take on positive values.
4. The parameter of $\mathcal{F}$ yields a *stochastic ordering*, so that if $\mu_i \geq \mu_j$, then $P(D_i < C) \leq P(D_j < C)$.
5. The demand realizations at the nodes are independent of one another.
6. Demands have continuous distributions, so that $P(D_i > C) \equiv P(D_i \geq C)$.

These properties are satisfied, for example, if $D_i, \cdots, D_n$ are independent random variables with $D_i \sim \text{Gamma}(\eta_i, \beta)$. For the sake of clarifying our notation, a $\text{Gamma}(\eta, \beta)$ distribution has density

$$f(x) \ := \ \frac{\beta^{-\eta} \cdot x^{\eta - 1} \cdot e^{-x/\beta}}{\Gamma[\eta]}.$$

For any route $\widehat{r}$, consider the problem

$$\min_{r \in R} \sum_{i=0}^{n-1} p_i(\widehat{r}) \cdot L(r_i, r_{i+1}). \tag{27}$$

Let $r(\widehat{r})$ be the solution to the optimization problem in (27). The optimization problem can be easily solved by reducing (27) to a Linear Integer Problem as shown in the previous section. Define $h(\widehat{r})$ as the optimal value of (27), i.e.,

$$h(\widehat{r}) \ := \ \sum_{i=0}^{n-1} p_i(\widehat{r}) \cdot L(r(\widehat{r})_i, r(\widehat{r})_{i+1}). \tag{28}$$

The quantity $h(\widehat{r})$ has no physical interpretation since the probability coefficients for the path $\widehat{r}$ have been used over the route $r(\widehat{r})$. Analogously to the definition in (12), let $P(\widehat{r})$ denote the expected value of the penalty costs for the route $\widehat{r}$. We have

$$P(\widehat{r}) \ = \phi \cdot \sum_{i=0}^{n} q_i(r) \ = \ \phi \cdot \sum_{i=0}^{n} (1 - p_i(\widehat{r})).$$

Finally, define $\psi(\widehat{r})$ as

$$\psi(\widehat{r}) \ := \ h(\widehat{r}) + P(\widehat{r}).$$

Comparing the above equation with (25), it is easy to see that $\psi(\widehat{r}) \leq F(\widehat{r})$. Now, consider the route $r^\star$ such that

$$\mu_{r_1^\star} \ \leq \ \mu_{r_2^\star} \ \leq \ \ldots \ \leq \mu_{r_n^\star}. \tag{29}$$

Then since stochastic ordering is ensured, we have

$$1 \ \geq \ p_i(r^\star) \ \geq \ \ldots \ \geq p_n(r^\star).$$

Further, $p_i(r^\star) \geq p_i(r)$ for all $r \in R$, so

$$P(r^\star) \ \leq \ P(r) \quad \text{for all } r \in R.$$

The lemma below shows that ψ provides a lower bound on the optimal value.

Lemma 1 *Suppose that the penalty value ϕ is bigger than the largest possible arc length M. Then, $\psi(r^\star)$ is a tight lower bound to $\min_r F(r)$.*

Proof: Given an arbitrary route $\widehat{r}$, consider the difference $\psi(\widehat{r}) - \psi(r^\star)$. We have

$$\begin{aligned}
\psi(\widehat{r}) - \psi(r^\star) \ &= \ \sum_{i=0}^{n-1} L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \cdot p_i(\widehat{r}) + P(\widehat{r}) - \sum_{i=0}^{n-1} L[r(r^\star)_i, r(r^\star)_{i+1}] \cdot p_i(r^\star) - P(r^\star) \\
&= \ \sum_{i=0}^{n-1} L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \cdot (p_i(\widehat{r}) - p_i(r^\star) + p_i(r^\star)) + P(\widehat{r}) - \\
& \quad - \sum_{i=0}^{n-1} L[r(r^\star)_i, r(r^\star)_{i+1}] \cdot p_i(r^\star) - P(r^\star).
\end{aligned}$$

Rearranging, we have

$$\psi(\widehat{r}) - \psi(r^\star) \ = \ \sum_{i=0}^{n-1} L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \cdot p_i(r^\star) - L[r(r^\star)_i, r(r^\star)_{i+1}] \cdot p_i(r^\star) - \tag{30}$$

$$- \sum_{i=0}^{n-1} (L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] - \phi) \cdot (p_i(r^\star) - p_i(\widehat{r})). \tag{31}$$

Note that the term in (30) is always a non-negative quantity, since $r(r^\star)$ is the solution to the optimization problem defined in (27). Moreover, by the assumption in the lemma we have that $L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \leq \phi$ for all i. Since $p_i(r^\star) \geq p_i(r)$ for all $r \in R$, it follows that the term in (31) is non-negative and thus $\psi(\widehat{r}) - \psi(r^\star)$ is non-negative. As seen above, $\psi(\widehat{r}) \leq F(\widehat{r})$. Finally, since $\widehat{r}$ was chosen arbitrarily, we conclude that $\psi(r^\star) \leq \min_r F(r)$. The lower bound becomes tight if $r^\star \equiv r(r^\star)$ so that $p_i(r^\star) = p_i(r(r^\star))$. In this case, $r^\star$ is the solution to the discrete stochastic optimization problem $\min_r F(r)$. ■

Note that if the converse holds, i.e. if ϕ is smaller than the lowest possible arc length m, then $\psi(r^\bullet)$ is a lower bound to $\min_r F(r)$, where $r^\bullet$ is the path such that

$$\mu_{r_1^\bullet} \geq \mu_{r_2^\bullet} \cdots \geq \mu_{r_2^\bullet}. \tag{32}$$

If $m \leq \phi \leq M$, then it is not possible to get a tight lower bound and a loose lower bound for $\min_r F(r)$ can be given as $h(r^\bullet) + P(r^\star)$ where $r^\star, r^\bullet$ are defined as before.

3.3 Demands are non-iid, penalties are non-uniform

Let the assumptions of the previous section hold here also. In this case, the expected cost function of any route ignoring the expected value of going back to the home depot can be defined as

$$F(r) \;:=\; \sum_{i=0}^{n-1} L(r_i, r_{i+1}) \cdot p_i(r) + \sum_{i=1}^{n} (1 - p_i(r)) \cdot \phi_{r_i}. \tag{33}$$

Here, ϕ_{r_i} denotes the penalty at node r_i for any route r. As in the previous section, define $h(\widehat{r})$ for any route $\widehat{r}$ as

$$h(\widehat{r}) \;:=\; \min_{r \in R} \sum_{i=0}^{n-1} L(r_i, r_{i+1}) \cdot p_i(\widehat{r}). \tag{34}$$

with $r(\widehat{r})$ being the solution to the optimization problem. Similarly, define $g(\widehat{r})$ as

$$g(\widehat{r}) \;:=\; \min_{r \in R} \sum_{i=1}^{n} (1 - p_i(\widehat{r})) \cdot \phi_{r_i}. \tag{35}$$

Now consider the route $r^\bullet$ defined by (32). Then, by stochastic ordering we have

$$p_i(r^\bullet) \;\leq\; p_i(r)$$

for any $1 \leq i \leq n$ and any r. Now consider $h(r^\bullet) = \sum_{i=0}^{n-1} p_i(r^\bullet) \cdot L[r(r^\bullet)_i, r(r^\bullet)_{i+1}]$. Then we have, for any $\widehat{r}$,

$$\begin{aligned} h(r^\bullet) &\leq \sum_{i=0}^{n-1} p_i(r^\bullet) \cdot L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \\ &\leq \sum_{i=0}^{n-1} p_i(\widehat{r}) \cdot L[r(\widehat{r})_i, r(\widehat{r})_{i+1}] \\ &= h(\widehat{r}). \end{aligned}$$

Hence, $h(r^\bullet) \leq h(\widehat{r})$ for any $\widehat{r} \in R$. Similarly, $g(r^\star) \leq g(\widehat{r})$ for any $\widehat{r} \in R$, where $r^\star$ is the route such that

$$\mu_{r_1^\star} \leq \mu_{r_2^\star} \leq \cdots \leq \mu_{r_n^\star}.$$

It follows that $h(r^\bullet) + g(r^\star)$ is a non-tight lower bound to $\min_r F(r)$.

4 Application of the Cross Entropy Method to Solve the VRPSD

The problem formulated in section 2 can be solved as a discrete stochastic optimization problem. We propose a modified version of the CE method to solve it. Before describing our method, we first review some basic concepts of the CE technique.

The CE method was developed by Reuven Rubinstein in the context of rare event simulations, where it is used in combination with the *importance sampling* (IS) technique. In those problems, the goal is to estimate the probability of occurrence of a very rare event. The difficulty lies in the fact that a standard Monte Carlo method will yield zero as the estimate, unless an extremely large sample size is used. Roughly speaking, the IS technique aims to select a probability measure that makes the occurrence of extremely rare events more frequent, thereby reducing the variance of the estimator. It is known that an optimal zero-variance measure exists, but it is of impractical use since it depends on the quantities one wants to estimate.

In the CE method, the Kullback-Leibler cross entropy is used to measure the distance between the optimal zero-variance measure and the importance sampling distribution. Thus, one chooses the distribution that minimizes that distance. The appeal of the method is that such minimization problem can be solved even though the optimal zero-variance measure is not completely known. This idea can be combined with and *adaptive* scheme, yielding a probability measure that can be used as an approximation of the optimal IS distribution. In Homem-de-Mello and Rubinstein (2002) the method is described in detail; see also Homem-de-Mello (2004) for further discussion.

In Rubinstein (1999, 2002), the idea behind the CE technique is applied to combinatorial optimization. The key concept is to view the selection of an optimal solution at random from the domain of possible values as a *rare event.* More specifically, suppose we want to minimize a deterministic function $h(x)$ over a finite set $\mathcal{X}$, and assume that h has a unique minimizer x^*. Let $p(\cdot)$ be a uniform distribution on $\mathcal{X}$, and let Y be a random variable on $\mathcal{X}$ with distribution p. Then, $\{Y = x^*\}$ is a rare event under p. As it turns out, the corresponding optimal IS distribution for this rare event is the atomic measure pointed at x^*. Thus, the CE method can be used as a heuristics to obtain the optimal solution. The resulting method has been applied successfully to numerous problems; we refer the reader to the tutorial by de Boer et al. (2003) for a more detailed discussion and references.

To describe more specifically the application of the CE method to the problem under study, let us consider first a simpler problem, where demand is not random and all demands and penalties are identical. It is clear that in this case only the transportation costs matter; therefore, the problem reduces to the well-known travelling salesman problem (**TSP**). Notice that such simplification is done just to motivate the developments below — the actual problems we consider become far more complicated than the **TSP** due to the introduction of random demands. The use of the CE method for the **TSP** has been widely studied in CE literature, see for instance de Boer et al. (2003), Margolin (2002), Rubinstein (1999, 2002). Despite that fact, there is a mathematical gap in the formulation of the method for the **TSP** that has not been addressed by previous works. Such issue is explained in detail below.

To proceed, let us put the **TSP** in the context of rare events. Let $\mathcal{R}$ denote the set of all paths of length $n+1$ that form a hamiltonian circuit on V. That is,

$$\mathcal{R} \;:=\; \{r \in V^{n+2} \,:\, r_0 = r_{n+1} = v_0,\; r_j \notin \{r_0, \ldots, r_{j-1}\} \text{ for } j = 1, \ldots, n.\}$$

For the sake of terminology, we will say that the paths $r \in \mathcal{R}$ are *valid routes.*

Following the approach outlined above, we would like to define a distribution q on the feasibility set $\mathcal{R}$. A natural way to do that, as proposed by Rubinstein (1999), is to define probabilities of transitions between any pair of nodes. That is, consider a Markov chain $\{X_k\}$ with state space $S = V$ (recall that $V = \{v_0, \ldots, v_n\}$ is the set of vertices of the underlying graph) and transition probability matrix P with no self-loops, i.e. $P_{uu} = 0$ for all $u \in V$. We wish to associate the desired distribution on the feasibility set $\mathcal{R}$ with the probability that the Markov chain follows a certain valid route. In other words, we would like to set

$$q(r) \;=\; P(X_0 = r_0, \ldots , X_{n+1} = r_{n+1}). \tag{36}$$

The problem with such approach, of course, is that q is not a probability distribution on $\mathcal{R}$, since $\sum_{r\in\mathcal{R}} q(r) < 1$. To illustrate, consider for example the case where there are only three nodes v_0, v_1 and v_2, and suppose that P is such that $P_{ij} = 1/2$ for all $i \neq j$. Clearly, there are only two valid routes: $r^1 = (v_0, v_1, v_2, v_0)$ and $r^2 = (v_0, v_2, v_1, v_0)$. The probability that the Markov chain $\{X_k\}$ follows route r^j is $(1/2)^3 = 1/8$. Therefore, the total probability of valid routes is $1/8 + 1/8 = 1/4 < 1$.

One way to fix the problem is to define a measure $\hat{q}(r)$ as the probability that the Markov chain $\{X_k\}$ follows route r *conditionally on the fact that r is valid route*. In that case, $\hat{q}(r)$ for $r \in \mathcal{R}$ is defined as

$$\hat{q}(r) \;=\; P(X_0 = r_0, \ldots , X_{n+1} = r_{n+1} \,|\, (X_0, \ldots , X_{n+1}) \in \mathcal{R}) \;=\; \frac{P(X_0 = r_0, \ldots , X_{n+1} = r_{n+1})}{P((X_0, \ldots , X_{n+1}) \in \mathcal{R})}. \tag{37}$$

Applying this idea in the above example yields $\hat{q}(r^1) = \hat{q}(r^2) = (1/8)/(1/4) = 1/2$. The calculation is coherent — since there are only two valid routes and P is uniform, the probability of each valid route should be $1/2$.

Definition (37), although correct, does not provide a constructive way to determine $\hat{q}(r)$ since computing the denominator is difficult. Moreover, it is unclear how to draw samples from $\hat{q}$. In what follows, we describe an alternative definition of $\hat{q}(r)$ to address these issues.

Let $\{Y_k\}$ be a stochastic process associated with the matrix P defined above such that, for any $r \in V^{n+2}$,

$$P(Y_0 = r_0) \;=\; I\{r_0 = v_0\} \tag{38}$$

$$P(Y_{n+1} = r_{n+1}) \;=\; I\{r_{n+1} = v_0\} \tag{39}$$

$$P(Y_k = r_{k+1} \,|\, Y_0 = r_0, \ldots , Y_k = r_k) \;=\; I\{r_{k+1} \in V \setminus \{r_0, \ldots , r_k\}\} \frac{P_{r_k,r_{k+1}}}{\sum_{s\in V\setminus\{r_0,\ldots,r_k\}} P_{r_k,s}}, \tag{40}$$

where the k in the last equation spans $0, \ldots , n-1$. At this point, we shall impose the following assumption:

Assumption A: For any $k = 0, \ldots , n-1$, given $r_0, \ldots , r_k$ such that $P(Y_0 = r_0, \ldots , Y_k = r_k) > 0$, there exists $s \in V \setminus \{r_0, \ldots , r_k\}$ such that $P_{r_k,s} > 0$. Moreover, P has no self-loops, i.e., $P_{uu} = 0$ for all $u \in V$.

Assumption A ensures that the the expression in (40) is well defined, that is, the denominator in it is always positive. It automatically holds, for example, if $P_{uv} > 0$ for all $u, v \in V$ such that $u \neq v$, but this need not be the case. For example, if P is the transition matrix corresponding to the deterministic chain whose only possible trajectory is $v_0, v_1, \ldots, v_n, v_0$, then assumption A holds even though P has only a few nonzero entries.

It is clear from (38)-(40) that the sample paths of the process $Y = (Y_0, \ldots, Y_{n+1})$ are valid routes. Notice that, because of assumption A, there exists some $r \in \mathcal{R}$ such that $P(Y = r) > 0$ and so the process Y is well defined.

We can now define a measure $\tilde{q}$ as the probability measure on $\mathcal{R}$ induced by the process Y, i.e., $\tilde{q}(r) = P(Y = r)$. Notice that, since

$$\tilde{q}(r) \;=\; P(Y_0 = r_0)P(Y_1 = r_1 \,|\, Y_0 = r_0) \ldots P(Y_n = r_n \,|\, Y_0 = r_0, \ldots, Y_{n-1} = r_{n-1})P(Y_{n+1} = r_{n+1}),$$

from (38)-(40) we see that $\tilde{q}$ is completely determined by the underlying transition matrix P. Moreover, $\tilde{q}(r)$ is easily computable for any given r. It is also easy to generate *samples* from $\tilde{q}(\cdot)$, see the discussion in section 4.1.1.

To illustrate, consider for example the case where P is the uniform matrix — i.e., $P_{uv} = 1/n$ for $u \neq v$, $P_{uu} = 0$. Then, the fraction in (40) becomes $(1/n)/[(n-k+1)(1/n)] = 1/(n-k+1)$ and hence we have, for $r \in \mathcal{R}$,

$$\tilde{q}(r) \;=\; \prod_{k=1}^{n} \frac{1}{n-k+1} \;=\; \frac{1}{n!}.$$

Also, $\tilde{q}(r) = 0$ for $r \notin \mathcal{R}$. Thus, the above construction yields the correct probability measure in this case.

Given a valid route $r \in \mathcal{R}$, let $F(r)$ be the associated cost. For a given threshold value a, the probability (with respect to the measure $\tilde{q}$) that the cost is not more than a is then given by

$$f(a) \;:=\; \mathbb{E}_{\tilde{q}}[I\{F(r) \leq a\}] \;=\; \sum_{r \in \mathcal{R}} I\{F(r) \leq a\}\, \tilde{q}(r), \tag{41}$$

where $I\{\cdot\}$ is the indicator function. Let q_a be a measure in $\mathcal{R}$ such that if $q_a(r) = 0$ then either $\tilde{q}(r) = 0$ or $F(r) > a$. We say that such a distribution is a *valid importance sampling (IS)* distribution. We can then write

$$f(a) \;=\; E_{q_a}[I\{F(r) \leq a\} W(\tilde{q}, q_a)], \tag{42}$$

where

$$W(\tilde{q}, q_a) \;=\; \frac{\tilde{q}(r)}{q_a(r)}. \tag{43}$$

Let a^* be the optimal value of the **TSP**, and suppose that the corresponding optimal solution r^* is *unique*. Since $I\{F(r) \leq a^*\} = 0$ for any $r \neq r^*$, it is easy to see that the measure defined by

$$q_{a^*}(r) \;:=\; \begin{cases} 1 & \text{if } r = r^* \\ 0 & \text{otherwise} \end{cases}$$

is a valid IS distribution. Moreover, the random variable inside the expectation in (42) is equal to a constant — $\tilde{q}(r^*)$ — with q_{a^*}-probability one. In other words, it yields a *zero variance* estimator. It is easy to see that this implies that q_{a^*} is also the solution of the corresponding CE problem (see, e.g., Homem-de-Mello and Rubinstein (2002)). Finally, notice that the measure q_{a^*} corresponds to the transition probability matrix given by

$$P^*_{u,v} \;:=\; \begin{cases} 1 & \text{if } u = r^*_\ell \text{ and } v = r^*_{\ell+1} \text{ for some } \ell \\ 0 & \text{otherwise,} \end{cases}$$

and that P^* satisfies assumption A. In other words, *the CE-optimal measure assigns probability one to the arcs in the optimal route.* This result is well known, see for instance Rubinstein (1999). It implies that we can use the CE method to solve the **TSP** problem. It is important to notice however that, since the algorithm must run for finite time, it provides only an approximation to the optimal measure. Nevertheless, the method yields a *heuristic procedure* which appears to behave very well, according to results reported in the literature.

To summarize, in this section we have discussed how to put the **TSP** problem in the context of rare events. This requires the definition of a suitable probability distribution from which it is easy to draw samples. We emphasize that the above issue has not been discussed in previous papers in the literature, which in our view creates a certain mathematical "gap" in the application of CE method to the **TSP**. The purpose of this discussion was to build a bridge to cover such gap. In section 4.1.1 we discuss how do draw samples from the aforementioned distribution.

4.1 The algorithm for deterministic case

We describe now the algorithm in more detail. We initially state the algorithm for the **TSP**, in a similar form to the description in Rubinstein (1999, 2002), where a justification for the algorithm can be found. Later we will describe the changes needed for the more general **VRPSD**.

Algorithm 1: CE for TSP

1. Let P be an initial transition probability matrix; set $k := 1$, $\rho := 0.01$.

2. Generate N valid routes $r^1, \dots, r^N$ according to the transition probabilities in P, and compute the cost F of each route.

3. Let γ_k be the sample $(1-\rho)$-quantile of the sequence $F(r^1), \dots, F(r^N)$.

4. (*) If necessary, decrease ρ and/or increase N and go back to step 2.

5. Update the transition probabilities as follows:

$$P_{uv} \;:=\; \frac{\sum_{j=1}^{N} I\{(u,v) \in r^j\} \cdot I\{F(r^j) \le \gamma_k\}}{\sum_{j=1}^{N} I\{F(r^j) \le \gamma_k\}}. \tag{44}$$

6. Test stopping criterion; if satisfied, STOP; otherwise, let $k := k+1$ and go back to step 2.

Step 4 of the above algorithm is in a sense optional, since it constitutes an enhancement to the algorithm. In fact, many of the reported implementations omit that step. In the context of rare event simulation, this step is discussed in Homem-de-Mello (2004).

4.1.1 Trajectory Generation

We discuss now step 2 of the algorithm — the generation of valid routes. With the notation defined above, this amounts to generating values from the distribution $\tilde{q}(\cdot)$ defined as $\tilde{q}(r) = P(Y = r)$ corresponding to a given transition matrix P. From (38)-(40) we have

$$\tilde{q}(r) \;=\; I\left\{r_0 = v_0, r_{n+1} = v_0\right\} \prod_{k=0}^{n-1} I\left\{r_{k+1} \in V \setminus \{r_0, \dots, r_k\}\right\} \frac{P_{r_k, r_{k+1}}}{\sum_{s \in V \setminus \{r_0, \dots, r_k\}} P_{r_k, s}}. \tag{45}$$

The construction of $\bar{q}$ suggests the procedure described below:

Algorithm 2: Trajectory Generation

1. Let $r_0 := r_{n+1} := v_0$, $P^{(0)} := P$.

2. Repeat for $k = 0, \dots, n-1$:

   (a) Choose node r_{k+1} according to the probability distribution $P^{(k)}_{r_k, \cdot}$.

(b) Define a new matrix $P^{(k+1)}$, obtained from $P^{(k)}$ by eliminating the row and column corresponding to r_k, and then normalizing the remaining elements.

The above algorithm is very similar to algorithms previously proposed in the literature, see e.g. Margolin (2002), Rubinstein (1999). In our context, however, Algorithm 2 is designed as a sampling technique from the distribution $\tilde{q}(\cdot)$ defined in (45). This puts the **TSP** formulation into the standard rare events framework. In addition, it allows us to prove the *correctness* of the algorithm — again, an issue that has not been discussed in previous papers. The result is stated below.

Proposition 1 *Suppose assumption A holds. Then, the trajectories generated by Algorithm 2 have distribution $\tilde{q}(\cdot)$ defined in (45).*

Proof. Note initially that assumption A ensures that Algorithm 2 is well defined, i.e. the normalization in step 2(b) is always possible (without this assumption, it might happen that after eliminating the row and column of $P^{(k)}$ corresponding to r_k one is left with a row of zeros).

Let $(r_0, \dots, r_{n+1})$ be a trajectory generated by Algorithm 2. We shall prove first by induction that, for any $k \in \{0, 1, \dots, n\}$, $P^{(k)}$ has columns and rows corresponding to $V \setminus \{r_0, \dots, r_{k-1}\}$ and

$$P^{(k)}_{w,u} = \frac{P_{w,u}}{\sum_{s \in V \setminus \{r_0,\dots,r_{k-1}\}} P_{w,s}} \tag{46}$$

for any $w, u \in V \setminus \{r_0, \dots, r_{k-1}\}$. The statement is true for $k = 0$, since $\sum_s P_{w,s} = 1$ and $P^{(0)} = P$ by definition. Suppose now the statement is true for all iterations from 0 to an arbitrary k, $0 \le k \le n-1$. In step 2(b), $P^{(k+1)}$ is obtained from $P^{(k)}$ by eliminating the row and column corresponding to r_k and then normalizing the remaining elements. That is, $P^{(k+1)}$ has columns and rows corresponding to $V \setminus \{r_0, \dots, r_k\}$. For $w, u \in V \setminus \{r_0, \dots, r_k\}$ we have

$$P^{(k+1)}_{w,u} = \frac{P^{(k)}_{w,u}}{\sum_{s \in V \setminus \{r_0,\dots,r_k\}} P^{(k)}_{w,s}}.$$

By applying the induction hypothesis to the right-hand side of the above equation we obtain

$$\begin{aligned} P^{(k+1)}_{w,u} &= \frac{P_{w,u} / \sum_{t \in V \setminus \{r_0,\dots,r_{k-1}\}} P_{w,t}}{\sum_{s \in V \setminus \{r_0,\dots,r_k\}} P_{w,s} / \sum_{t \in V \setminus \{r_0,\dots,r_{k-1}\}} P_{w,t}} \\ &= \frac{P_{w,u}}{\sum_{s \in V \setminus \{r_0,\dots,r_k\}} P_{w,s}}, \end{aligned}$$

so the statement is true for $k+1$.

Next, notice that in step 2(a) r_{k+1} is generated — conditionally on $r_0, \dots, r_k$ — from the distribution $P^{(k)}_{r_k,\cdot}$. Because P has no self-loops, when $w = r_k$ in (46) the sum in the denominator spans $s \in V \setminus \{r_0, \dots, r_k\}$. Thus, $r_{k+1} \in V \setminus \{r_0, \dots, r_k\}$ and, by unconditioning, we conclude that the resulting route $r = (r_0, r_1, \dots, r_{n+1})$ is generated with probability $\tilde{q}(r)$ as in (45). ■

4.2 Application of the CE method to the VRPSD

We now apply the ideas behind the CE method to solve the underlying stochastic vehicle routing problem. It important to notice that the presence of random demands and penalties at each node makes this problem *much harder* than the **TSP**. Moreover, as seen earlier the objective function includes the *expected value* of the total cost, a feature that arises from the fact that the demands are stochastic. Nevertheless, in view of the structural similarities between the two problems — both aim at finding the optimal route — it is desirable to exploit the empirically observed efficiency of the CE method in order to solve the **VRPSD**.

A version of the CE method for *stochastic* optimization problems has been proposed in Rubinstein (1999). In that paper, the goal is to minimize a function of the form $\mathbb{E}[H(r, D(r))]$, where $D(r)$ is a random variable that depends on the route r. Essentially, the idea in Rubinstein (1999) is to incorporate the randomness into the generation of tours (step 2 of Algorithm 1), so that the cost corresponding to the ith generated route is $H(r^i, d(r^i))$ — here, $d(r^i)$ denotes one realization of the random variable $D(r^i)$. Aside from that, the algorithm remains the same as Algorithm 1. Although in theory the method enjoys similar properties to the version for deterministic problems, our studies show that the variability of the objective function due to randomness precludes the use of such algorithm. In our experiments, this algorithm took very long to converge, and the solution obtained was often not optimal.

The alternative we propose is to apply *Monte Carlo techniques* to the problem. An important observation is that the distribution of the random vector D_r in the objective function defined in (3) *does not* depend on the route — only the order of the components of vector does. Thus, we can approximate the objective function $G(r) = \mathbb{E}[H(r, D_r)]$ by drawing a sample $D^1, \dots, D^K$ from the distribution of the vector D and computing

$$G_K(r) \ := \ \frac{1}{K} \sum_{j=1}^{K} H(r, D^j). \tag{47}$$

We can then solve the problem $\min_r G_K(r)$ as an approximation to the original one. Notice the resulting problem is no longer stochastic and can be solved with a deterministic method, e.g., the deterministic version of the CE method. General convergence properties of such an approach — sometimes called *sample average approximation* — are discussed in Kleywegt et al. (2001). One variation of this basic idea is to incorporate sampling into a deterministic algorithm, so that different sample averages are computed at different iterations. This is the approach we take here. A general discussion of its convergence properties can be found in Homem-de-Mello (2003).

4.3 The Modified CE algorithm for the Stochastic TSP

The algorithm we implemented to solve the underlying **VRPSD** incorporates the Monte Carlo techniques described above into the CE method. The outline is as follows.

Algorithm 3: CE for VRPSD

1. Take the initial transition matrix P such that any route is equally likely to be generated. That is, P has zero in the diagonal and all remaining elements are identical. Set $\ell := 1$, $\rho := 0.01$.

2. Generate N_ℓ trajectories $r^1, \dots, r^{N_\ell}$ from the transition matrix P using Algorithm 2 and estimate the expected cost $G(r)$ of each route r^i as per (47), using a number K_ℓ of demand samples. The same samples are used for each route. We will call N_ℓ the *simulation size* and K_ℓ the *sampling size*. Notice that both N_ℓ and K_ℓ can follow fixed schemes, or changing schemes as explained in detail later.

3. Select the trajectories with $G_{K_\ell}(r)$ in the top $100 \cdot \rho\%$, and denote by x_ℓ the cut off value for this quantile. That is, x_ℓ is the $100 \cdot (1-\rho)\%$ quantile of $G_{K_\ell}(r^1), \dots, G_{K_\ell}(r^{N_\ell})$. Store the trajectory with the minimum value generated thus far as $(x_{\min}, r_{\min})$. Denote by x^*_ℓ the minimum objective function value obtained during the current iteration ℓ.

4. Update the transition matrix P based on the generated trajectories according to the equation below:
$$P_{uv} \;:=\; \frac{\sum_{j=1}^{N_\ell} I\{(u,v) \in r^j\} \cdot I\{G_{K_\ell}(r^j) \leq x_\ell\}}{\sum_{j=1}^{N_\ell} I\{G_{K_\ell}(r^j) \leq x_\ell\}}. \tag{48}$$

5. If for some fixed t, $x^*_\ell = x^*_{\ell-1} \dots x^*_{\ell-t}$, STOP. Otherwise, set $\ell := \ell + 1$ and go to step 2.

At termination, the solution will be $(x_{\min}, r_{\min})$.

Two issues must be clarified at this point. First, in our actual implementation we used a modified version of the update (48) that computes a weighted average between the right hand side of (48) and the old matrix P. That is, the matrix is updated as

$$P_{uv}^{\ell} \;:=\; \alpha^{\star} \cdot \frac{\sum_{j=1}^{N} I\{(u,v) \in r^j\} \cdot I\{G_K(r^j) \leq x_\ell\}}{\sum_{j=1}^{N} I\{G_K(r^j) \leq x_\ell\}} + (1-\alpha^{\star}) \cdot P_{uv}^{\ell-1}. \tag{49}$$

The purpose of this modification is to prevent elements of P from going to zero prematurely. Such a scheme has become standard in implementations of the CE method, see for instance de Boer et al. (2003), Margolin (2002). In our implementation we used a weight $\alpha^{\star} = 0.9$ but found that, due to the inherent variability in the problem, the behavior of the algorithm is fairly insensitive to the choice of $\alpha^{\star}$. In section 6 we present some numerical results to illustrate this phenomenon.

We must also emphasize that Algorithm 3 is significantly different from previously proposed variations of the CE method. The major difference is the presence of sampling (with a variable sample size) as well as the variation of the number of trajectories generated at every iteration. In our experience, such features are crucial for the implementation of the CE method for a problem with high variability such as the **VRPSD**.

5 Heuristic Implementation and Performance Characteristics

We discuss now some implementation aspects of the algorithm outlined above. There are three main aspects that contribute to the performance of the algorithm :

1. The number of paths generated during each iteration, which we call the *simulation scheme*;
2. The number of demand samples used to estimate the expected cost of each route, which we call the *sampling scheme*;
3. Structural features, such as whether the demands are iid, or whether the penalties are identical for all nodes.

We describe now some characteristics of our implementation regarding each of the above topics.

5.1 Sampling scheme

- The sampling size and the data generated are common for all the paths during a given iteration. This reduces the randomness to a certain extent by ensuring that the objective function value is the same for identical paths generated during the iteration. The sampling size and the data generated may be fixed throughout the course of the run. Alternately, the sample size may be fixed but a different set generated for every iteration. Notice that it is difficult to estimate the appropriate sampling size for a given iteration as it depends on the route, and the structural complexity. On one hand, an extremely large sampling size may result in unnecessary computations and a small sampling size can result in convergence to a wrong solution. We have attempted to tackle this problem by using a variable-sampling scheme as described in the next bullet.

- For demands following normal distributions, a variable-sampling scheme seems to work well. The optimal strategy seems to be to start at an appropriate size and then increase as $n \log(n)$ or $O(n)$, until stability is achieved and further increase of the sampling is not required. Homem-de-Mello (2003) presents theoretical results on variable sampling schemes used in the context of discrete stochastic optimization problems where the sampling size is adaptively decided with every iteration even as the search for the optimum expected value narrows. This scheme was implemented by starting with an initial sampling size of twice the problem size and increasing by steps of 10 linearly for a fixed number of iterations. This is because accuracy is not crucial at the early stages of algorithm. After this customary period, the sampling size is checked such that the half-width of a 95% confidence interval is not more than 1% of the sampling mean. If this is not the case, the sampling is increased; otherwise it is kept a constant. During the initial iterations, arriving at the exact value of the expected cost of a path is not so important. Thus, the sampling size can start at a smaller level and then be progressively increased. This increases the total number of iterations but the overall number of computations decreases.

- Alternatively, we implemented a pre-processing routine to find a “good” sampling size a priori. Here, a path is randomly picked and the expected value of the path function is estimated using Monte-Carlo sampling. The sample size is taken to be very large (around 10,000) and the sampling size is brought down in steps such that the interval in the subsequent iteration always contains the interval of the previous iteration and the half-width of a 95% confidence interval never exceeds 5% percent of the estimated cost function value. Having picked the optimal sampling size using the above mentioned pre-processing routine, we start the CE procedure. At every iteration, we generate a sample with the given sample size. If the minimum cost function value’s half-width exceeds it by 5%, we increase the sample size for the next iteration. Thus, the sampling size is still adaptive. This partially nullifies the possibility that the preprocessing scheme picked an unrepresentative path for sample size selection .

- The main advantage for the adaptive sampling size scheme is that we need not make a priori assumptions about the sampling size, which is a function of the structural complexity of the problem and hence, difficult to estimate.

5.2 Simulation scheme

For the **TSP**, a common choice for the simulation size is $5n^2$ (Rubinstein 1999). In our case, considering the additional complexity, a higher value for the simulations size ought to be considered. Our studies indicate that ideally the simulation scheme should start at a high level and then be decreased. The simulation size does not matter towards the end since the probability of generating a certain path is high and it is unlikely that any other path will be generated, irrespective of the simulation size. Our numerical studies indicate that having a larger simulation size during the earlier iterations seems to improve the chances of converging to the optimal solution. We have implemented the simulation scheme by starting with a simulation size of $Bn^2 (B = 105)$, and decreasing the values of B linearly with every subsequent iteration until B becomes 5, from which point it is not changed anymore.

5.3 Structural features

When demands are iid, we have shown that it is possible to get an exact solution. This provides a framework to assess the performance of the heuristics. The heuristics is very accurate in this case as results indicate. In the case of demands being non-iid, penalties being uniform and ignoring the expected cost of going back to the home depot, we have a tight lower bound under conditions discussed previously. The amount of variation between trials and the average percent deviation from the lower bound gives an assessment of the performance of the heuristics.

6 Numerical Results

We describe now the results of some numerical experiments with the algorithm described above. For simulation purposes, the demands at nodes were modelled as having Gamma distribution, i.e. $D_i \sim \text{Gamma}(\eta_i, \beta)$. The value of β was identical for all the nodes. The distance arcs $L(i,j)$ were generated as uniform independent random numbers between 0 and 100. The *filling coefficient* ζ is defined as the ratio of vehicle capacity C to the sum of mean demands across all the nodes, i.e.,

$$\zeta \ := \ \frac{C}{\beta \cdot \sum_{k=1}^{n} \eta_k}.$$

The coefficient ζ indicates, in some sense, the degree of difficulty of the problem. A larger ζ means that the vehicle is less likely to run out of items before finishing the route, i.e., penalties are less likely to occur. Similarly, very small values of ζ correspond to the case where the vehicle runs out of items at the first nodes and so penalties occur for sure. Thus, for medium values of ζ (say, between 0.5 and 1.0) there will be larger variability in the objective function, since penalties may or may not occur. Notice that a similar concept is used in Gendreau et al. (1995). To test the algorithm for different degrees of difficulty of the problem, we used the values 0.5, 0.75, 1.0 and 1.25 for ζ in our experiments.

We discuss next each of the structural cases.

6.1 Iid demands

For this case, the demands were modelled as $D_i \sim \text{Gamma}(\eta, \beta)$ with $\eta = 5$ and $\beta = 10$, whereas the penalty at each node i was generated randomly between 0 and 50.

Recall from section 3.1 that when demands are iid an exact solution can be computed. To do so, notice that, using the analytical formulation for the cumulative distribution function for the Gamma distribution, the probability coefficients p_i, $i = 1, \ldots, n$, and p_i°, $i = 2, \ldots, n-1$ defined in (6) and (9) are easily computed as:

$$p_i = 1 - e^{-C/\beta} \sum_{j=0}^{i\cdot(\eta-1)} \frac{(C/\beta)^j}{j!}$$

and

$$p_i^\circ \;=\; \int_0^C (1 - F(C-x)) \cdot f(x,i) \cdot dx. \tag{50}$$

In (50),

$$f(x,i) \;:=\; \frac{\beta^{-(i-1)\cdot\eta} \cdot x^{(i-1)*\eta-1} \cdot e^{-x/\beta}}{\Gamma[(i-1)*\eta]},$$

and

$$F(x) \;:=\; 1 - e^{-x/\beta} \sum_{j=0}^{\eta-1} \frac{(x/\beta)^j}{j!}.$$

Equation (50) refers to a convolution integral where $f(x,i)$ is the probability density function for the $\text{Gamma}((i-1)\cdot\eta, \beta)$ distribution and $F(x)$ is the cumulative distribution function for the $\text{Gamma}(\eta, \beta)$ distribution, with η being an integer. The coefficients p_1°, p_n° defined in (8),(10) are easily computable since $p_1^\circ = 1 - p_1$ and $p_n^\circ = p_{n-1}$. As defined in (5), p_0 is set to 1.

For each value of ζ, the vehicle capacity was accordingly fixed. For each problem size (in this case, $n = 11$ and $n = 16$) and value of ζ, we ran 10 trials of our method to solve the problem defined in (4) and then compared the results with the exact problem defined in (18) subject to the constraints given by (15)-(17) and (19)-(21). To solve this integer program, we implemented a branch-and-bound technique using the ILOG SOLVER 4.4.

For the sake of space, we do not present results for each individual configuration or each individual run. Instead, we present the summarized Table 1, where the numbers in parenthesis indicate the number of times out of 10 the objective value using our modified CE method was within 5% of the exact value for a given problem size and filling coefficient. It was not feasible to get exact solutions using the solver for problem instances with $n > 16$ due to the computational requirements.

$n \setminus \zeta$	0.5	0.75	1.00	1.25
11	(9)	(10)	(10)	(10)
16	(10)	(9)	(10)	(10)

Table 1: iid demands and non-uniform penalties

A few comments about the numbers in Table 1 are in order. Those numbers show that the algorithm sometimes failed to reach the optimal value. This is due to two reasons: first, it is known that the CE method may end up at a sub-optimal solution (even in the deterministic case) since convergence is only guaranteed asymptotically. Second, the large variability inherent to the underlying problem (due to its stochastic nature) precludes exact evaluations of the objective function and hence convergence to optimal solution may not occur. Nevertheless, the table shows that the algorithm often finds a value within 5% of optimum, and in fact only in one occasion was the obtained value more than 10% bigger than the exact one. Notice also that the algorithm tends to behave better for larger values of the filling coefficient. This is expected, since as we commented earlier the problem is easier (due to less variability) in those cases.

In Table 2 we present the results for one particular configuration to illustrate the variability of the results across runs. For the sake of space, we only list five runs. A complete set of tables can be found in Chepuri (2003).

t	Estimated optimal value	Iterations
1	286.18 ± 1.42	13
2	285.96 ± 1.16	19
3	286.18 ± 1.42	13
4	292.80 ± 1.29	15
5	297.02 ± 1.23	16

Table 2: Results for iid demands, non-uniform penalties; Size=11; $\zeta = 0.75$; Exact solution = 285.74

6.2 Non-iid demands, uniform penalties

For this case, we ignored the expected cost of going back to the home depot in order to fit the framework of section 3.2. The demand at each node i was modeled as a Gamma(η_i, β) random variable, with the parameter η_i being picked randomly as an integer between 0 and 10. The parameter β was set to 5. The penalties were fixed at 50 for each given problem instance.

Recall that in this case a lower bound for the optimal value of (4) can be obtained. The lower bound was computed using the ILOG SOLVER 4.4 system by solving the following problem:

$$\min_x \sum_{k=1}^{n} p_{k-1}(r) \sum_{i=0}^{n} \sum_{j=0}^{n} x_{(k-1)i} \cdot x_{kj} \cdot L(v_i, v_j) + \sum_{k=1}^{n} \sum_{i=1}^{n} \phi \cdot (1 - p_k(r)) \cdot x_{ki} \tag{51}$$

subject to the constraints (15)-(17). In (51), $r \equiv r^{\bullet}$ (defined by (29)) or $r \equiv r^{\star}$ (defined by (32)) depending on the value of ϕ as discussed in Section 3.2. The quantity $p_k(r)$ is defined in (22)-(23).

Again, we ran 10 trials of our method to solve the problem defined in (4). Since in these cases we do not know the exact solution, we cannot provide a table in the format of Table 1. Instead, we present in tables 3-6 the results for $n = 11$ and each of the filling coefficients (again, we show only five runs instead of ten to keep the table small). Observe that the obtained values are very consistent across runs, and that for larger filling coefficients the objective value is very close to the lower bound.

6.3 Non-iid demands, Non-uniform penalties

As before, here the demands were modeled as Gamma(η_i, β) random variables, with the distribution parameter η_i being picked randomly as an integer between 0 and 10. The parameter β was set to 1. A lower value for β (compared to the previous examples) was picked to reduce the variance of

t	Value	Iterations
1	369.17 ± 2.48	12
2	375.26 ± 2.73	15
3	375.40 ± 2.71	15
4	376.08 ± 3.17	13
5	370.27 ± 2.93	14

Table 3: Results for non-iid demands, uniform penalties; Size =11; Lower Bound = 328.4; $\zeta = 0.5$

t	Value	Iterations
1	254.82 ± 3.17	12
2	256.86 ± 3.65	15
3	256.38 ± 3.15	15
4	256.69 ± 4.23	16
5	257.37 ± 3.16	16

Table 4: Results for non-iid demands, uniform penalties; Size=11; Lower Bound = 236.00; $\zeta = 0.75$

t	Value	Iterations
1	160.05 ± 3.09	12
2	159.60 ± 2.88	15
3	160.44 ± 2.71	23
4	159.97 ± 3.14	16
5	160.53 ± 3.37	15

Table 5: Results for non-iid demands, uniform penalties; Size=11, Lower Bound = 150.13; $\zeta = 1.00$

t	Value	Iterations
1	114.67 ± 1.64	12
2	115.02 ± 0.99	12
3	114.13 ± 0.77	14
4	114.47 ± 1.26	12
5	114.53 ± 1.39	12

Table 6: Results for non-iid demands, uniform penalties; Size=11; Lower Bound = 113.00; $\zeta = 1.25$

the demands and make the problem more tractable, which allows for solving problems of slightly larger size, namely 16 and 21.

The reduction in noise also helps to discern trends by varying the weighting coefficient $\alpha^\star$ defined in (49) from 0.9 to 0.1. Problem instances were constructed by choosing arc lengths and penalties randomly between 0 and 100. Five simulations for each problem instance were run for the original problem defined by (4) using our modified CE method.

Detailed results are displayed in Tables 7-10 below. For the sake of space, we only present results for size 21. The numbers in parentheses indicate the number of iterations to termination. As before, the filling coefficient ζ proves to be a significant factor in the overall performance. As ζ is increased, the heuristic performance seems to improve. On the other hand, the weighting parameter $\alpha^\star$ seems to marginally affect performance, although this has to be tested within a rigorous statistical framework. Notice that the overall spread is greater for the case of non-iid demands, non-uniform penalties compared to the other cases, which reflects the fact the variability is much larger here.

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	228.66 ± 1.86 (19)	231.32 ± 1.45 (26)	238.63 ± 1.67 (40)
2	237.64 ± 1.82 (18)	231.79 ± 1.48 (20)	238.67 ± 1.28 (40)
3	242.38 ± 1.66 (17)	234.53 ± 1.32 (20)	241.73 ± 0.97 (40)
4	247.81 ± 1.81 (20)	237.13 ± 1.49 (25)	245.56 ± 1.83 (40)
5	254.49 ± 1.26 (16)	242.35 ± 1.71 (21)	251.26 ± 1.11 (40)

Table 7: Results for non-iid demands, non-uniform penalties; Size=21; $\zeta = 0.50$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	173.63 ± 1.84 (18)	176.58 ± 1.24 (23)	180.06 ± 2.09 (40)
2	175.19 ± 1.73 (17)	177.49 ± 1.72 (21)	184.88 ± 0.81 (40)
3	179.53 ± 1.05 (20)	177.77 ± 1.46 (24)	185.27 ± 1.51 (40)
4	186.37 ± 1.62 (16)	182.50 ± 1.24 (22)	190.38 ± 2.94 (40)
5	188.27 ± 2.03 (17)	189.41 ± 1.55 (19)	191.87 ± 1.69 (40)

Table 8: Results for non-iid demands, non-uniform penalties; Size=21; $\zeta = 0.75$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	125.92 ± 2.10 (17)	124.01 ± 1.28 (21)	127.85 ± 1.05 (40)
2	128.95 ± 1.16 (30)	129.61 ± 1.74 (25)	134.43 ± 2.03 (40)
3	133.08 ± 1.15 (17)	129.78 ± 1.34 (24)	135.12 ± 1.66 (40)
4	133.17 ± 1.94 (17)	131.88 ± 1.77 (22)	135.68 ± 1.26 (40)
5	138.32 ± 1.55 (18)	137.39 ± 2.08 (21)	136.08 ± 1.41 (40)

Table 9: Results for non-iid demands, non-uniform penalties; Size=21; $\zeta = 1.00$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	100.99 ± 0.44 (16)	96.93 ± 0.49 (21)	99.13 ± 0.46 (40)
2	103.46 ± 0.46 (17)	99.20 ± 0.42 (25)	106.13 ± 0.40 (40)
3	106.45 ± 0.29 (20)	102.88 ± 0.31 (24)	108.22 ± 0.48 (40)
4	107.45 ± 0.18 (19)	106.44 ± 0.22 (18)	109.55 ± 0.35 (40)
5	113.94 ± 0.44 (16)	107.16 ± 0.49 (22)	114.85 ± 0.34 (40)

Table 10: Results for non-iid demands, non-uniform penalties; Size=21; $\zeta = 1.25$

6.3.1 Elite sampling

As discussed earlier, the algorithm proposed in this paper introduces a sampling component into the standard CE method. In the past few years, however, much effort has been put into finding suitable modifications of the basic CE algorithm in order to speed up its performance. Thus, it is natural to consider adapting these new techniques into our modified method.

In this section we discuss numerical results that illustrate the effect of the sampling component over one of the CE variations, namely, the *elite sampling method* proposed in Margolin (2002). We implemented a version of the elite method for the case of non-iid demands, non-uniform penalties. The outline of new algorithm is as follows:

Algorithm 4: Iterative scheme

1. Take the initial transition matrix P such that any route is equally likely to be generated. That is, P has zero in the diagonal and all remaining elements are identical. Set $k := 1, \rho := 0.01$.

2. Generate N trajectories from the transition matrix P based on the TSP trajectory generation rule. If the current iteration number k is greater than 10, replace the first $N \cdot \rho$ routes randomly generated by the $N \cdot \rho$ routes with the least expected cost $G_K(r)$ computed as per (47) so far.

3. Select the trajectories with $G_K(r)$ in the top $100 \cdot \rho\%$, and denote by x_k the cut off value for this quantile. Store the trajectory with the minimum value generated thus far as (x_{min}, r_{min}). Denote by x_k^* the minimum objective function value obtained during the current iteration k.

4. Update the transition matrix P based on (49). Update the list of the $N \cdot \rho$ routes with the least expected costs obtained so far.

5. If for some fixed $t, x_k^* = x_{k-1}^* \cdots x_{k-t}^*$, STOP. Otherwise, set $k := k + 1$ and go to step 2.

6. At termination, the solution will be (x_{min}, r_{min}).

The modified version of the CE based on elite sampling was implemented based on a fixed simulation size, thus keeping the number of elite samples introduced during each iteration at a fixed size and the ratio of the elite routes to randomly generated routes also constant. Elite routes were introduced from the tenth iteration so as to not converge prematurely to a sub-optimal solution.

The main controls parameters in the elite sampling method are the iteration number k at which the elite routes are introduced, the number of elite routes introduced and ratio of the elite routes to randomly generated routes at each iteration.

The algorithm was tried out on the problem sizes 16 and 21 based on the same instances for which we obtained results discussed before for the non-iid, non uniform penalties case (cf. Tables 7-10) . This was done to facilitate a comparison with those results. The numbers suggest that the elite sampling method does improve the performance of the algorithm, which encourages further research on this topic.

The numerical results are shown in Tables 11-14 below.

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	229.97 ± 1.52 (20)	221.72 ± 1.31 (23)	229.38 ± 1.79 (40)
2	231.59 ± 1.68 (18)	228.65 ± 1.66 (25)	229.77 ± 1.61 (40)
3	232.59 ± 1.36 (19)	228.92 ± 1.99 (25)	233.68 ± 1.48 (40)
4	232.61 ± 1.23 (18)	231.72 ± 1.46 (21)	234.33 ± 2.21 (40)
5	232.65 ± 1.54 (19)	232.35 ± 1.35 (26)	237.40 ± 1.45 (32)

Table 11: Results for non-iid demands and non-uniform penalties based on elite sampling variant; Size =21; $\zeta = 0.50$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	169.42 ± 1.68 (24)	169.82 ± 1.12 (23)	167.35 ± 1.79 (40)
2	171.15 ± 1.95 (18)	170.42 ± 1.97 (24)	167.77 ± 1.61 (40)
3	170.69 ± 1.81 (19)	170.69 ± 1.81 (25)	168.68 ± 1.48 (40)
4	173.54 ± 2.13 (20)	172.58 ± 1.20 (23)	171.12 ± 2.21 (40)
5	174.12 ± 1.54 (19)	173.35 ± 1.35 (26)	174.40 ± 1.45 (32)

Table 12: Results for non-iid demands and non-uniform penalties based on elite sampling variant; Size =21; $\zeta = 0.75$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	123.96 ± 1.17 (21)	123.66 ± 1.09 (21)	121.82 ± 1.07 (40)
2	123.98 ± 1.27 (18)	123.70 ± 1.20 (23)	122.77 ± 1.61 (40)
3	125.67 ± 2.23 (16)	123.96 ± 1.00 (20)	123.62 ± 1.56 (40)
4	125.89 ± 2.04 (17)	124.05 ± 1.32 (23)	124.33 ± 1.21 (40)
5	125.12 ± 1.54 (19)	124.41 ± 1.31 (22)	124.40 ± 1.72 (32)

Table 13: Results for non-iid demands and non-uniform penalties based on elite sampling variant; Size =21; $\zeta = 1.00$

	$\alpha^\star = 0.9$	$\alpha^\star = 0.5$	$\alpha^\star = 0.1$
1	97.09 ± 0.70 (17)	96.32 ± 0.79 (21)	96.45 ± 0.79 (40)
2	97.12 ± 0.73 (18)	97.23 ± 0.60 (23)	96.77 ± 0.61 (38)
3	98.67 ± 0.83 (16)	98.96 ± 0.45 (20)	96.83 ± 0.67 (40)
4	99.12 ± 0.74 (17)	99.05 ± 0.71 (22)	97.33 ± 0.71 (40)
5	99.62 ± 0.54 (19)	99.41 ± 0.81 (22)	97.40 ± 0.65 (32)

Table 14: Results for non-iid demands and non-uniform penalties based on elite sampling variant; Size =21; $\zeta = 1.25$

7 Conclusion

We have discussed a stochastic vehicle routing problem (**VRPSD**) and proposed a new technique to solve it. Our method is a modified version of the CE method that incorporates Monte Carlo techniques. The algorithm has produced encouraging results. Another contribution of this paper is the development of theoretical results regarding exact solutions and lower bounds for the problem, which can also serve as a good framework to test other heuristics for the problem formulation we have considered.

We have also provided a formalization on the use of the CE method for **TSP**-like problems. As a result, we have devised an algorithm for generating trajectories and proved its correctness. This covers an existing gap in the literature.

The main advantage in using the CE algorithm for vehicle routing problems is that the method is fairly independent of particular problem formulations but at the same time it does exploit of the structure inherent in this class of problems. Thus, we envision that the CE method can also be applied to other problem formulations in the **SVRP** class of problems. In particular, the CE may be extended to multiple vehicle routing problems. Further studies are required to perfect the method as before it can be used to provide solutions to real world logistics problems.

Acknowledgement. We would like to thank Anton Kleywegt from Georgia Tech for introducing this problem to us. Also, we thank Marc Posner from Ohio State University for his help with integer programming formulations. We thank Reuven Rubinstein and two referees for their comments and suggestions.

References

A. N. Aizawa and B. W. Wah. Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2:97–122, 1994.

T. T. Allen and W. Ittiwattana. An elitist genetic algorithm incorporating sequential subset selection. 2002. Manuscript, The Ohio State University.

G. Allon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. Accepted for publication in *Annals of Operations Research*, 2003.

M. H. Alrefaei and S. Andradottir. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science*, 45:748–764, 1999.

M. H. Alrefaei and S. Andradottir. A modification of the stochastic ruler method for discrete stochastic optimization. *European Journal of Operational Research*, 133:160–182, 2001.

S. Andradottir. A method for discrete stochastic optimization. *Management Science*, 41:1946–1961, 1995.

S. Andradottir. A global search method for discrete stochastic optimization. *SIAM Journal on Optimization*, 6:513–530, 1996.

C. Bastian and R. Kan. The stochastic vehicle routing problem revisited. *European Journal of Operational Research*, 56:407–412, 1992.

R. E. Bechhofer and T. Santner. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. Wiley, 1995.

D. Bertsimas, P. Chervi, and M. Peterson. Computational approaches to stochastic vehicle routing problems. *Transportation Science*, 29:342–352, 1995.

D. J. Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40: 574–585, 1992.

K. Chepuri. Solving the vehicle routing problem with stochastic demands using the cross entropy method. Master's thesis, Ohio State University, 2003.

G. Clark and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.

P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals Of Operations Research*, 2003. Forthcoming.

M. Dror and P. Trudeau. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research*, 23:228–235, 1986.

B. L. Fox and G. W. Heine. Probabilistic search with overrides. *Annals of Applied Probability*, 5: 1087–1094, 1995.

S. B. Gelfland and S. K. Mitter. Simulated annealing with noisy or imprecise energy measurements. *JOTA*, 62:49–62, 1989.

M. Gendreau, G. Laporte, and R. Seguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29:143–155, 1995.

M. Gendreau, G. Laporte, and R. Seguin. Invited review: Stochastic vehicle routing. *European Journal of Operational Research*, 88:3–12, 1996a.

M. Gendreau, G. Laporte, and R. Seguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research*, 44:469–477, 1996b.

B. L. Golden and J. R. Yee. A framework for probabilistic vehicle routing. *AIIE Transactions*, 11: 109–112, 1979.

W. J. Gutjahr, A. Hellmayr, and G. C. Pflug. Optimal stochastic single-machine tardiness scheduling by stochastic branch and bound. *European Journal of Operational Research*, 117:396–413, 1999.

W. J. Gutjahr and G. C. Pflug. Simulated annealing for noisy cost functions. *Journal of Global Optimization*, 8:1–13, 1996.

C. Hjorring and J. Holt. New optimality cuts for a single-vehicle stochastic routing problem. *Annals of Operations Research*, 86:569–584, 1999.

Y. Hochberg and A. C. Tamhane. *Multiple Comparison Procedures*. Wiley, 1987.

T. Homem-de-Mello. On convergence of simulated annealing for discrete stochastic optimization. Manuscript, Ohio State University, 2001.

T. Homem-de-Mello. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation*, 13:108–133, 2003.

T. Homem-de-Mello. A study on the cross-entropy method for rare event probability estimation. Working Paper 04-002, Department of Industrial Engineeting and Management Sciences, Northwestern University, 2004.

T. Homem-de-Mello and R. Y. Rubinstein. Rare event probability estimation using cross-entropy. In E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 310–319, 2002.

J. C. Hsu. *Multiple Comparisons, Theory and Methods.* Chapman and Hall, 1996.

A. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2001.

G. Laporte and F. V. Louveaux. *Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies.* Wolsey, North Holland, Amsterdam, first edition, 1990.

G. Laporte, F. V. Louveaux, and H. Mercure. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research*, 39:71–78, 1989.

G. Laporte, F. V. Louveaux, and H. Mercure. The vehicle routing problem with stochastic travel times. *Transportation Science*, 26:161–170, 1992.

G. Laporte, F. V. Louveaux, and H. Mercure. A priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 42:543–549, 1994.

L. Margolin. Cross-entropy method for combinatorial optimization. Master's thesis, Technion — Israel Institute of Technology, 2002.

G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York, NY, 1988.

J. Pichitlamken and B. L. Nelson. A combined procedure for optimization via simulation. Manuscript, Northwestern University, 2002.

R. Y. Rubinstein. The cross-entropy method for combinatorial and continous optimization. *Methodology and Computing in Applied Probability*, 2:127–190, 1999.

R. Y. Rubinstein. Cross-entropy and rare events for maximal cut and bipartition problems. *ACM Transactions on Modeling and Computer Simulation*, 12:27–53, 2002.

N. Secomandi. Comparing neuro-dynamic programming algosithms for the vehicle routing problem with stochastic demands. *Computers and Operations Research*, 27:1171–1200, 2000.

N. Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49:796–802, 2001.

W. R. Stewart Jr. and B. L. Golden. Stochastic vehicle routing. *European Journal of Operational Research*, 14:371–385, 1983.

F. Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3:192–204, 1969.

P. Trudeau and M. Dror. Stochastic inventory routing: Route design with stockouts and route failure. *Transportation Science*, 26:171–184, 1992.

D. Yan and H. Mukai. Discrete stochastic optimization. *SIAM Journal on Control and Optimization*, 30:594–612, 1992.

W. Zhai, P. Kelly, and W. B. Gong. Genetic algorithms with noisy fitness. *Mathematical Computer Modelling*, 23:1310–142, 1996.