

# A Priori Policy Evaluation and Cyclic-Order-Based Simulated Annealing for the Multi-Compartment Vehicle Routing Problem with Stochastic Demands

Justin C. Goodson\*

May 30, 2014

## Abstract

We develop methods to estimate and exactly calculate the expected cost of a priori policies for the *multi-compartment vehicle routing problem with stochastic demands*, an extension of the classical vehicle routing problem where customer demands are uncertain and products must be transported in separate partitions. We incorporate our estimation procedure into a cyclic-order-based simulated annealing algorithm, significantly improving the best-known solution values for a set of benchmark problems. We also extend the updating procedure for a cyclic order's candidate route set to duration-constrained a priori policies.

**Keywords:** logistics, stochastic multi-compartment vehicle routing, cyclic-order search, simulated annealing

## 1 Introduction

Vehicle routing problems with stochastic demands (VRPSDs) underlie operational problems in logistics where routes must be planned without full knowledge of customer demand levels. In

---

\*Department of Operations & Information Technology Management; John Cook School of Business; Saint Louis University; goodson@slu.edu; 3674 Lindell Blvd. St. Louis, MO 63108; 314-977-2027; 314-977-1483 (fax)

such cases, actual demands are first observed upon arrival to customer locations. In some situations, products must be transported in independent vehicle compartments, giving rise to the *multi-compartment vehicle routing problem with stochastic demands* (MCVRPSD), a generalization of the single-compartment case (see the review of Gendreau et al. (1996) or the work of Laporte et al. (2002) for examples). Applications of the MCVRPSD include the collection and delivery of different types and qualities of milk (Caramia and Guerriero, 2010), distribution of various grades of fuel (Brown et al., 1987), transport of animal food (El Fallahi et al., 2008), pickup and delivery of livestock (Oppen and Løkketangen, 2008), selective waste collection, and transport of groceries requiring different levels of refrigeration.

A common approach to handle uncertainty in customer demands is to restrict attention to *a priori* policies (see Campbell and Thomas (2008) for a review). An *a priori* policy requires vehicles to visit customers in the order specified by a set of pre-defined routes, returning to the depot to replenish in the event vehicle capacity is inadequate to fully serve demand, i.e., a *route failure*. *A priori* routes are routinely used in industry (Erera et al., 2010) and create a regularity of service that can be beneficial for both the customers and the drivers – customers may be served at roughly the same time each day they require service and the drivers become familiar with their routes.

We make two main contributions. First, we develop methods to estimate and exactly calculate the expected cost of an *a priori* MCVRPSD policy. Straightforward procedures exist to calculate the expected cost of an *a priori* policy when all products are transported in a single compartment (Teodorović and Pavković, 1992). However, in the multi-compartment case, *a priori* policy evaluation is more difficult. Mendoza et al. (2008, 2010, 2011) derive an expression to calculate the expected cost of an *a priori* policy, but the expression requires probability calculations to be made over random variables representing partially unserved customer demands. The distributions of these random variables are unknown, thereby making it difficult to implement the expression. Recognizing this challenge, Mendoza et al. (2008) explore methods to approximate the expected cost of an *a priori* policy for the MCVRPSD. When the ratio of compartment capacities to customer demands is high, Mendoza et al. (2008) find a *take-all* approximation yields good estimates of the expected cost. However, the bias of the estimate increases as the ratio of compartment capacities

to customer demands decreases.

The method we propose to calculate the exact expected cost of an a priori MCVRPSD policy builds on Goodson et al. (2013) and applies to problem instances where customer demands follow discrete probability distributions with finite support. The method requires exponential time to execute and is not practical for use in optimization procedures. However, the exact method leads to a simulation scheme that provides unbiased and consistent estimates of a policy's expected cost for both discrete and continuous customer demand distributions. Because the simulation scheme can be efficiently incorporated into local search procedures, it provides an attractive alternative when the bias of the take-all approximation is unacceptable.

Our second contribution is the development of a cyclic-order-based simulated annealing procedure for the MCVRPSD. Goodson et al. (2012) propose cyclic-order neighborhoods as the basis of local search methods for a broad class of routing problems, of which the MCVRPSD is a member. We utilize our simulation scheme for a priori MCVRPSD policies to accelerate the initial iterations of the simulated annealing procedure. We also extend the cyclic-order updating procedure of Goodson et al. (2012) for the candidate route set to the duration-constrained a priori policies considered in this paper.

As of this writing, our method appears to be the most effective heuristic for the MCVRPSD. Without tailoring the cyclic-order search procedure to the MCVRPSD, we improve the best-known solution values for 159 of 180 benchmark problem instances and match the best-known solution values for the remaining 21 instances. These results, in conjunction with the results of Goodson et al. (2012), support cyclic-order-based local search as an effective solution procedure for a variety of routing problems.

The remainder of the paper is organized as follows. In §2, we formally state the MCVRPSD. In §3, we review related literature. In §4, we discuss methods to exactly calculate and estimate the expected cost of an a priori MCVRPSD policy. In §5, we present our cyclic-order-based simulated annealing procedure. In §6, we discuss the results of computational experiments. We make concluding remarks and suggestions for future research in §7. Appendix A provides a guide to the primary notation used in the paper.

## 2 Problem Statement

The MCVRPSD generalizes the classical vehicle routing problem, a NP-hard optimization problem (Toth and Vigo, 2001). The MCVRPSD is characterized by a complete graph  $G = (\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N} = \{0, 1, \dots, N\}$  is a set of  $N + 1$  nodes and  $\mathcal{E} = \{(n, n') : n, n' \in \mathcal{N}\}$  is the set of edges connecting the nodes. Nodes  $1, \dots, N$  represent customer locations and node 0 represents a depot from which a set  $\mathcal{M} = \{1, 2, \dots, M\}$  of identical vehicles operates. Each vehicle contains a separate compartment for each product in the product set  $\mathcal{P} = \{1, 2, \dots, P\}$ . The capacity of the compartment assigned to product  $p$  in  $\mathcal{P}$  is  $Q_p$ . Customer demands for products are assumed to be independent random variables. The random demand for product  $p$  in  $\mathcal{P}$  at customer  $n$  in  $\mathcal{N}$  is denoted  $D_{n,p}$ . We denote the distribution function by  $F_{D_{n,p}}(\cdot)$  with support  $\mathcal{S}(D_{n,p})$ , which we require to be a subset of the range  $[0, \bar{D}]$ , where  $\bar{D}$  is a finite number. Prior to arrival at customer locations, customer demands are known only in distribution. Upon arrival, customer demands for each product are observed and served to the maximum extent possible, subject to available vehicle capacity for each product. When capacity in one or more compartments is exhausted (i.e., a route failure occurs), a vehicle returns to the depot and replenishes the capacity of all compartments. Vehicle routes begin and end at the depot and travel times among customers are known. The time to travel from location  $n$  to location  $n'$  is denoted  $t(n, n')$ . The objective of the MCVRPSD is to obtain a policy that serves customer demand with minimal expected travel time subject to a route duration limit  $L$  (e.g., end of a working day), by which time all vehicles must return to the depot. To be consistent with the literature, we use travel time to represent cost, noting that  $t(\cdot, \cdot)$  may be replaced with a general cost function.

We focus on a priori routing policies for the MCVRPSD. A priori policies are characterized by a priori routes, or predetermined sequences of customers. We denote an a priori route for vehicle  $m$  in vehicle set  $\mathcal{M}$  by the sequence of customers  $v^m = (v_0^m = 0, v_1^m, \dots, v_{I^m}^m, v_{I^m+1}^m = 0)$ . We denote by  $(v^m)_{m \in \mathcal{M}}$  a set containing an a priori route for each vehicle  $m$  in  $\mathcal{M}$ . In a set of a priori routes, each customer appears exactly once on exactly one route. We adopt the classical *detour-to-depot* policy, the same policy employed by Mendoza et al. (2010, 2011). The policy requires

vehicles to serve customers in the order they appear in a set of a priori routes. Vehicles must travel directly to the next customer on the route, i.e., preemptive capacity replenishment is not allowed. In the event of a route failure, vehicles must make return trips to the depot until customer demands are fully served. An a priori route is feasible if its expected travel time is less than or equal to the route duration limit  $L$ . If we denote by  $\mathcal{V}$  the set of all a priori route sets, by  $A_{v_{I^m+1}}^m$  the random arrival time of vehicle  $m$  to final destination  $v_{I^m+1}^m$ , and by  $\mathbb{E}[A_{v_{I^m+1}}^m]$  the expected arrival time of vehicle  $m$  to  $v_{I^m+1}^m$ , then the problem we seek to solve is

$$\min \left\{ \sum_{m \in \mathcal{M}} \mathbb{E} \left[ A_{v_{I^m+1}}^m \right] : \right. \quad (1)$$

$$\mathbb{E} \left[ A_{v_{I^m+1}}^m \right] \leq L, m \in \mathcal{M}, \quad (2)$$

$$\left. (v^m)_{m \in \mathcal{M}} \in \mathcal{V} \right\}. \quad (3)$$

Throughout the remainder of the paper, because we refer primarily to a single a priori route, to ease the notation we drop the superscript  $m$  and refer to a route simply by  $v$  and to the  $i^{\text{th}}$  customer on a route by  $v_i$ .

### 3 Related Literature

Research related to multi-compartment vehicle routing appears to begin with van der Bruggen et al. (1995), who consider the task of designing delivery routes for gasoline distribution. van der Bruggen et al. (1995) and others model customer demands as deterministic. In this section, we focus on literature treating customer demands as stochastic.

Mendoza et al. (2008) study *take-none* and *take-all* approximations of the recourse cost component of a priori policies. The take-all approximation assumes that when a route failure occurs at a particular customer, all demands for all products at that customer are served in full before replenishing capacity at the depot; the take-none approximation assumes none of the demands are served. Both approximations circumvent the issue of making probability calculations over random variables with unknown distributions. Computational experiments conclude the take-all scheme is

superior, but that the quality of both approximations degrades as the ratio of compartment capacities to customer demands decreases.

The methods we propose to estimate and exactly calculate the expected cost of a priori policies serve as alternatives to the take-none and take-all approximations. Our methods differ in three ways. First, the take-none and take-all approximations can be used when demands follow discrete or continuous probability distributions, provided the convolution of random demands can be readily calculated (e.g., Poisson- or normally-distributed demands). In contrast, our exact evaluation procedure can only be used when customer demands follow discrete distributions, but there is no limit imposed by the need to calculate convolutions of random variables. The estimation procedure may be applied regardless of the nature of customer demand distributions. Second, our procedures do not depend on the assumption that demand realizations are less than or equal to compartment capacities, thereby accounting for the possibility of multiple route failures at one customer. Third, when fully serving a customer's demands exhausts vehicle capacity, rather than require the vehicle to return to the current customer after replenishing capacities at the depot, our methods route the vehicle directly to the next customer on the a priori route.

Using the take-all scheme to approximate expected route costs, Mendoza et al. (2010) build on the memetic algorithm of El Fallahi et al. (2008), improving solution values obtained by a constructive heuristic and by an alternative memetic procedure. Also using the take-all approximation, Mendoza et al. (2011) propose construction heuristics that yield good solutions with short computing times. Adapting our cyclic-order procedure to the take-all scheme, we improve many of the solution values obtained by Mendoza et al. (2010, 2011).

Seeking to balance the trade-off between expected routing costs and variability, Mendoza et al. (2009) place a constraint on the probability that route cost exceeds a given threshold. Within a memetic algorithm, the expected route cost and the variance of the route cost are estimated via simulation. We also employ simulation to estimate expected route costs, explicitly showing how the simulation can be incorporated into local search procedures.

For a given sequence of customers, Pandelis et al. (2012) develop a dynamic programming algorithm to determine an optimal restocking policy. Demonstrating a threshold-type structure

on the optimal policy for any integer number of vehicle compartments, Pandelis et al. (2012) generalize the work of Tatarakis and Minis (2009), which applies to only two compartments. The restocking policies of Pandelis et al. (2012) and Tatarakis and Minis (2009) are a relaxation of the a priori policies we consider, which do not permit preemptive capacity replenishment prior to route failures. In contrast to the work of Pandelis et al. (2012) and Tatarakis and Minis (2009), which assumes a fixed delivery sequence, we search the space of a priori delivery sequences seeking lower-cost a priori policies.

The procedures of §4 build on Appendix A of Goodson et al. (2013), which develops a priori policy evaluation methods for a single-compartment VRPSD with route duration constraints where the objective is to maximize expected demand served. We extend the methods of Goodson et al. (2013) to the multi-compartment case and to the objective of minimizing expected route costs. These extensions increase the computational complexity of a priori policy evaluation from pseudo-polynomial to exponential in the number of vehicle compartments.

The heuristic developed in §5 builds on Goodson et al. (2012), which develops cyclic-order-based local search methods for a broad class of routing problems. Based on the concept of optimal partitioning (Beasley, 1983), a cyclic order maps a permutation of the customer set to a collection of many possible routing solutions, the best of which can be identified via the polynomial-time procedure of Ryan et al. (1993) and Boctor and Renaud (2000). Our heuristic uses the neighborhood structures proposed by Goodson et al. (2012), but extends the solution updating procedure to accommodate duration-constrained routes.

## **4 Evaluation of A Priori Policies**

In §4.1, we develop a procedure to calculate the expected travel time incurred by an a priori route. When customer demands follow discrete probability distributions, the procedure in §4.1 yields an exact calculation of the expected travel time. In §4.2, we discuss how to use simulation to estimate the expected travel time of an a priori route. The simulation scheme yields unbiased and consistent estimates of the expected travel time for both continuous and discrete customer demand

distributions. Applying the methods of §4.1 or §4.2 to each a priori route in a set of routes allows for the calculation or estimation of the objective function (1) and the constraints (2).

## 4.1 Exact Evaluation

Below, we develop a method to calculate the expected time of initial arrival to a particular customer on an a priori route. Given a set of a priori routes  $v$ , the associated policy can be evaluated by applying the method to the last customer on each route,  $v_{I+1}$ , and summing the results.

The initial arrival time at a customer on an a priori route is a function of the initial arrival time at the previous customer, the demands at the previous customer, and available vehicle capacities upon initial arrival to the previous customer. We denote by  $A_{v_i}$  the random time of the initial arrival of the vehicle to customer  $v_i$  on a priori route  $v$ . We denote by  $C_{v_i,p}$  the random capacity of compartment  $p$  of the vehicle upon the initial arrival to customer  $v_i$  on a priori route  $v$ . We denote by  $a_{v_i}$  and  $c_{v_i}$  realizations of  $A_{v_i}$  and  $C_{v_i}$ , respectively. If demand for a product is greater than or equal to the available capacity for that product, then return trips to the depot are required to fully serve demand at that customer. In such cases, if we define

$$\psi(v_i) = \max_{p \in \mathcal{P}} \left\{ \frac{d_{v_i,p} - c_{v_i,p}}{Q_p} \right\}, \quad (4)$$

where  $\lceil \cdot \rceil$  is the ceiling function and  $d_{v_i,p}$  is a realization of  $D_{v_i,p}$ , then  $\lceil \psi(v_i) \rceil$  is the number of return trips to the depot required to fully serve demand for all products at customer  $v_i$ .

We separate the calculation of  $c_{v_i,p}$  and  $a_{v_i}$  into three cases:

$$c_{v_i,p} = \begin{cases} c_{v_{i-1},p} - d_{v_{i-1},p}, & d_{v_{i-1},p} < c_{v_{i-1},p} \quad \forall p \in \mathcal{P}, \\ Q_p, & \psi(v_{i-1}) = \lceil \psi(v_{i-1}) \rceil, \\ \min \{ \lceil \psi(v_{i-1}) \rceil Q_p - d_{v_{i-1},p} + c_{v_{i-1},p}, Q_p \}, & \psi(v_{i-1}) < \lceil \psi(v_{i-1}) \rceil, \end{cases} \quad (5)$$

and



$$a_{v_i} = \begin{cases} a_{v_{i-1}} + t(v_{i-1}, v_i), & d_{v_{i-1}, p} < c_{v_{i-1}, p} \quad \forall p \in \mathcal{P}, \\ a_{v_{i-1}} + (\psi(v_{i-1}) + 1) \cdot t(v_{i-1}, 0) \\ \quad + \psi(v_{i-1}) \cdot t(0, v_{i-1}) + t(0, v_i), & \psi(v_{i-1}) = \lceil \psi(v_{i-1}) \rceil, \\ a_{v_{i-1}} + t(v_{i-1}, v_i) \\ \quad + \lceil \psi(v_{i-1}) \rceil \cdot (t(v_{i-1}, 0) + t(0, v_{i-1})), & \psi(v_{i-1}) < \lceil \psi(v_{i-1}) \rceil. \end{cases} \quad (6)$$

The boundary conditions in equations (5) and (6) are  $c_{v_0, p} = Q_p$  for all products  $p$  in  $\mathcal{P}$  and for all vehicles and  $a_{v_0} = 0$  for all vehicles. In the first case, demand for each product at customer  $v_{i-1}$  is less than the available capacity for that product, thus vehicle capacity for each product is simply decremented by the amount of demand for that product and the vehicle travels directly from  $v_{i-1}$  to  $v_i$ . The second and third cases account for situations where the demand for at least one product at customer  $v_{i-1}$  is greater than or equal to vehicle capacity for that product upon arrival to  $v_{i-1}$ , thereby requiring  $\lceil \psi(v_{i-1}) \rceil$  return trips to the depot to replenish capacity. In the second case, satisfying demand for all products at  $v_{i-1}$  exactly depletes vehicle capacity for at least one product, thereby requiring the vehicle to replenish at the depot one additional time and then travel directly to customer  $v_i$  with full capacity in each compartment. In the third case, some capacity remains in all vehicle compartments after serving demands for all products at  $v_{i-1}$ . After making the necessary return trips to the depot, the vehicle travels directly from  $v_{i-1}$  to  $v_i$ .

We denote by  $C_{v_i} = (C_{v_i, p})_{p \in \mathcal{P}}$  the vector of remaining capacities for the vehicle upon initial arrival to customer  $v_i$  and by  $D_{v_i} = (D_{v_i, p})_{p \in \mathcal{P}}$  the vector of random demands at customer  $v_i$ . We denote by  $c_{v_i}$  and  $d_{v_i}$  realizations of  $C_{v_i}$  and  $D_{v_i}$ , respectively. For clarity, we sometimes write  $c_{v_i, p}$  and  $a_{v_i}$  as functions of the terms they depend on. Specifically, we denote by  $c_{v_i}(d, c) = (c_{v_i, p}(d, c))_{p \in \mathcal{P}}$  the capacities upon initial arrival to customer  $v_i$  when  $d_{v_{i-1}} = d$  and  $c_{v_{i-1}} = c$ . Similarly, we denote by  $a_{v_i}(d, c, a)$  the time of initial arrival to customer  $v_i$  when  $d_{v_{i-1}} = d$ ,  $c_{v_{i-1}} = c$ , and  $a_{v_{i-1}} = a$ . Further, we denote the  $p^{\text{th}}$  element of a demand vector  $d$  and capacity vector  $c$  by  $d_p$  and  $c_p$ , respectively.

Proposition 1 establishes a recursion to calculate the joint probability of initial arrival time at customer  $v_i$  and capacities upon initial arrival to customer  $v_i$ . Proposition 1 also provides an ex-

pression to calculate the expected initial arrival time at customer  $v_i$ . We assume customer demands follow a discrete probability distribution and we denote the probability mass function for demand at customer  $n$  for product  $p$  by  $f_{D_{n,p}}(\cdot)$ . A proof of Proposition 1 is provided in Appendix B.

**Proposition 1** (Properties of A Priori MCVRPSD Policies). *The joint probability of arriving to customer  $v_i$  with available capacities  $c$  and at time  $a$  is*

$$\mathbb{P}\{C_{v_i} = c, A_{v_i} = a\} = \sum_{d'} \sum_{c'} \sum_{a'} \mathbf{1}\{c = c_{v_i}(d', c'), a = a_{v_i}(d', c', a')\} \cdot \mathbb{P}\{C_{v_{i-1}} = c', A_{v_{i-1}} = a'\} \cdot \prod_{p=1}^P f_{D_{v_{i-1},p}}(d'_p), \quad (7)$$

where  $\mathbf{1}\{\cdot\}$  is an indicator function that returns 1 if the statement  $\{\cdot\}$  is true and 0 otherwise. The expected time of initial arrival at customer  $v_i$  is

$$\mathbb{E}[A_{v_i}] = \sum_d \sum_c \sum_a a_{v_i}(d, c, a) \cdot \mathbb{P}\{C_{v_{i-1}} = c, A_{v_{i-1}} = a\} \cdot \prod_{p=1}^P f_{D_{v_{i-1},p}}(d_p), \quad (8)$$

where  $\mathbb{P}\{C_{v_{i-1}} = c, A_{v_{i-1}} = a\}$  is calculated via equation (7).

Algorithm 1 employs Proposition 1 to calculate the expected time at which the vehicle first arrives to customer  $v_i$ . Applying Algorithm 1 when index  $i = I + 1$ , the index of the last location on route  $v$ , results in the expected time required to serve all customer demands on a priori route  $v$ . To ease the notation, we define  $g(v_j, c, a) = \mathbb{P}\{C_{v_j} = c, A_{v_j} = a\}$ . We denote by  $\mathcal{S}(D_n) = \times_{p \in \mathcal{P}} \mathcal{S}(D_{n,p})$  the joint support of demands at customer  $n$  for the product set  $\mathcal{P}$ , where  $\times$  is the Cartesian product.

Lines 1-4 of Algorithm 1 initialize the procedure. Line 1 assigns a probability of one to the initial state of the vehicle – the vehicle is initially located at the depot, with full capacity, and at time zero. Line 2 assigns a probability of zero to all other possible initial states. Line 3 initializes joint capacity and arrival probabilities for all other customers to zero and line 4 initializes to zero the expected time of initial arrival to customer  $v_i$ ; these quantities are updated as Algorithm 1 is executed. Lines 5-8 utilize equation (7) to build the joint probabilities of capacities and arrival

times at customers  $v_1, v_2, \dots, v_{i-1}$ . Finally, lines 9-11 apply equation (8) to calculate the expected time of initial arrival to customer  $v_i$ .

---

**Algorithm 1** Calculation of Expected Time of Initial Arrival to Customer  $v_i$

---

```

1:  $g(v_0, (Q_p)_{p \in \mathcal{P}}, 0) \leftarrow 1$ 
2:  $g(v_0, c, a) \leftarrow 0$  for all  $(c, a) \neq ((Q_p)_{p \in \mathcal{P}}, 0)$ 
3:  $g(v_j, c, a) \leftarrow 0$  for  $j = 1, 2, \dots, i - 1$  and for all  $(c, a)$ 
4:  $\mathbb{E}[A_{v_i}] \leftarrow 0$ 
5: for  $j = 1, 2, \dots, i - 1$  do
6:   for  $g(v_{j-1}, c, a) \neq 0$  do
7:     for  $d \in \mathcal{S}(D_{v_{j-1}})$  do
8:        $g(v_j, c_{v_j}(d, c), a_{v_j}(d, c, a)) \leftarrow$ 
            $g(v_j, c_{v_j}(d, c), a_{v_j}(d, c, a)) + g(v_{j-1}, c, a) \cdot \prod_{p=1}^P f_{D_{v_{j-1}, p}}(d_p)$ 
9:   for  $g(v_{i-1}, c, a) \neq 0$  do
10:    for  $d \in \mathcal{S}(D_{v_{i-1}})$  do
11:       $\mathbb{E}[A_{v_i}] \leftarrow \mathbb{E}[A_{v_i}] + a_{v_i}(d, c, a) \cdot g(v_{i-1}, c, a) \cdot \prod_{p=1}^P f_{D_{v_{i-1}, p}}(d_p)$ 

```

---

Under the assumption that customer demands, vehicle capacities, and travel times are all integer-valued, Proposition 2 establishes the worst-case complexity of Algorithm 1 is exponential in the number of products. A proof of Proposition 2 is provided in Appendix B. The driver of the exponential complexity is the size of the supports for demands at each customer. Even when other parameters take on small values, Algorithm 1 iterates over the supports, leading to a best-case complexity that is also exponential in the number of products.

**Proposition 2** (Worst-Case Complexity). *Let  $\bar{Q} = \max_{p \in \mathcal{P}} \{Q_p\}$ , let  $\underline{Q} = \min_{p \in \mathcal{P}} \{Q_p\}$ , let  $\bar{L} = \max_{j=1, \dots, i-1} \{t(v_j, 0) + t(0, v_j)\}$ , and let  $\bar{S} = \max\{|\mathcal{S}(D_{v_j, p})| : j = 1, \dots, i \text{ and } p = 1, \dots, P\}$ . Assume customer demands, vehicle capacities, and travel times are all integer-valued. Then, the worst-case computational complexity of Algorithm 1 is  $O(i \cdot \bar{L} \cdot \bar{Q} \cdot P \cdot \lceil \bar{D}/\underline{Q} \rceil \cdot \bar{S}^P)$ .*

Although Algorithm 1 can in principle be used to evaluate a priori MCVRPSD policies, the practical use of Algorithm 1 is limited. When route size, number of products, and support sizes

exceed modest values, execution time and demands on computer memory become excessive. We explore the behavior of Algorithm 1 in §6.2. The computational limitations of Algorithm 1 motivate the simulation-evaluation method of §4.2.

## 4.2 Evaluation via Simulation

Because Algorithm 1 requires exponential time to execute, statistical estimation of the value of an a priori MCVRPD policy may be an attractive alternative to exact evaluation. Below, we show how to estimate the expected time of initial arrival to a particular customer. We also discuss how the estimation procedure can be implemented within local search schemes.

Let  $\hat{d}_{n,p}$  be a randomly generated demand from distribution function  $F(\cdot)_{D_{n,p}}$  for product  $p$  at customer  $n$ . Denote by  $\hat{d}_n = (\hat{d}_{n,p})_{p \in \mathcal{P}}$  a vector of randomly generated demands at customer  $n$  and by  $\hat{d} = (\hat{d}_n)_{n \in \mathcal{N} \setminus \{0\}}$  a vector of randomly generated demands at all customers and for all products. Let  $\hat{d}^1, \hat{d}^2, \dots, \hat{d}^H$  be  $H$  randomly generated demand vectors. Then, the expected arrival time at customer  $v_i$  can be estimated by

$$\widehat{\mathbb{E}}[A_{v_i}] = \frac{1}{H} \sum_{h=1}^H a_{v_i}^{\hat{d}^h}, \quad (9)$$

where  $a_{v_i}^{\hat{d}^h}$  is the time of the vehicle's initial arrival to customer  $v_i$  when customer demands are  $\hat{d}^h$ . The quantity  $a_{v_i}^{\hat{d}^h}$  can be calculated recursively using equations (5) and (6). This is accomplished by initializing  $a_{v_0}$  to 0 and  $c_{v_0}$  to  $(Q_p)_{p \in \mathcal{P}}$ . Then, for  $j = 1, 2, \dots, i$ , calculate  $c_{v_j} = c_{v_j}(\hat{d}_{v_{j-1}}^h, c_{v_{j-1}})$  and  $a_{v_j} = a_{v_j}(\hat{d}_{v_{j-1}}^h, c_{v_{j-1}}, a_{v_{j-1}})$ , yielding  $a_{v_i}^{\hat{d}^h}$  equal to  $a_{v_i}$ . Because  $\widehat{\mathbb{E}}[A_{v_i}]$  is the sample mean of the arrival times across all  $H$  demand samples, it is an unbiased and consistent estimator of  $\mathbb{E}[A_{v_i}]$ . Calculating  $\widehat{\mathbb{E}}[A_{v_i}]$  when index  $i = I + 1$ , the index of the last location on route  $v$ , results in an unbiased and consistent estimate of  $\mathbb{E}[A_{v_i}]$ , the expected time required to serve all customer demands on a priori route  $v$ .

Within a local search scheme, when calculating  $\widehat{\mathbb{E}}[A_{v_{I+1}}]$ , it may not be necessary to estimate the time of initial arrival to each customer in a priori route  $v$ . Because the time of initial arrival at a customer depends only on the preceding sequence of customers' demands, capacities, and arrival

times, it is only necessary to estimate the expected time of initial arrival where the preceding sequence of customers has changed; initial arrival times at other customers remain the same.

More generally, suppose the current solution in a local search procedure contains a priori route  $v$ , that  $\widehat{\mathbb{E}}[A_{v_{I+1}}]$  has been calculated, and that for each demand sample  $\hat{d}^h$ ,  $c_{v_i}$  and  $a_{v_i}$  have been stored for  $i = 0, 1, \dots, I + 1$ . Further, suppose a neighbor solution  $\bar{v}$  is generated by changing the customer sequence of  $v$  beginning at customer  $v_k$ . For demand sample  $h$ , because  $c_{\bar{v}_{k-1}}$  and  $a_{\bar{v}_{k-1}}$  are known from the current solution,  $a_{\bar{v}_{I+1}}^{\hat{d}^h}$  can be obtained by calculating  $c_{\bar{v}_j} = c_{\bar{v}_j}(\hat{d}_{\bar{v}_{j-1}}^h, c_{\bar{v}_{j-1}})$  and  $a_{\bar{v}_j} = a_{\bar{v}_j}(\hat{d}_{\bar{v}_{j-1}}^h, c_{\bar{v}_{j-1}}, a_{\bar{v}_{j-1}})$  for  $j = k, k + 1, \dots, |\bar{v}| - 1$ . Repeating this procedure for each demand sample and averaging the results yields  $\widehat{\mathbb{E}}[A_{v_{I+1}}]$ . We employ this updating procedure when estimating the cost of a priori routes in the cyclic-order-based local search we describe in §5.

Because the computation required to calculate  $\widehat{\mathbb{E}}[A_{v_i}]$  grows linearly with the sample size  $H$ , equation (9) may be an attractive alternative to Algorithm 1. In accordance with Proposition 2, the computational experiments we present in §6.2 suggest the difference in computation time between estimation and exact evaluation grows exponentially with route size, number of products, and support sizes. Thus, for many problem instances, estimation-based local search for a priori MCVRPSPD policies may be the only computationally tractable alternative.

## 5 Cyclic-Order Simulated Annealing Procedure

In §5.1, we describe a cyclic-order simulated annealing procedure for a priori MCVRPSPD policies. Our procedure utilizes a cyclic-order solution encoding (based on Ryan et al. (1993)), which maps a permutation of the customer set to a collection of many possible MCVRPSPD solutions, the best of which is identified via a polynomial-time algorithm. We employ the neighborhood structures of Goodson et al. (2012), which operate directly on cyclic orders. In §5.2, we extend to duration-constrained a priori policies the updating procedure of Goodson et al. (2012) for a cyclic order's candidate route set. Although we summarize the cyclic-order approach and our modifications in this section, the reader is referred to Goodson et al. (2012) for more details.

To compare the output of our simulated annealing procedure with solution values reported

in the literature, we evaluate a priori policies using the take-all scheme of Mendoza et al. (2008, 2010, 2011). We do this because the literature does not report exact solution values for MCVRPSD problem instances, thus a comparison with such a benchmark is not possible.

The take-all approximation assumes that when a route failure occurs at a particular customer, all demands for all products at that customer are served in full before replenishing capacity at the depot. The take-all approximation is achieved via a small modification to equation (5): in the third case, set the compartment capacity equal to  $Q_p$ . Thus, the procedures outlined in §4 can be used to estimate and exactly calculate the take-all cost of an a priori MCVRPSD policy. Because exact evaluation via our procedures is limited to discrete customer demand distributions, we use the procedure of Mendoza et al. (2008, 2010, 2011) to evaluate the take-all cost when customer demands follow continuous distributions.

## 5.1 Two-Stage Simulated Annealing Algorithm

Simulated annealing (Kirkpatrick et al., 1983; Johnson et al., 1989, 1991) is a local search algorithm in which non-improving moves are probabilistically accepted in an attempt to avoid becoming trapped in a low-quality, locally-optimal solution. Our simulated annealing procedure is executed in two phases. In the first phase, we estimate the take-all value of a priori MCVRPSD policies using the method outlined in §4.2, modified to accommodate the take-all scheme, as described above. The second phase seeks to improve the solution by exactly calculating the take-all value of a priori MCVRPSD policies. Because customer demands follow continuous distributions in the benchmark problem instances of §6.1, we employ the procedure of Mendoza et al. (2008, 2010, 2011) to calculate the take-all cost.

The primary benefit of the two-phase procedure is reduction in computation time. Although the second phase can serve as a stand-alone solution procedure (and this yields good MCVRPSD solutions), initializing the second phase with the first-phase solution can accelerate discovery of high quality MCVRPSD solutions. Our experience suggests setting the number of randomly generated demand vectors to  $H = 3$  in phase one is sufficient to identify quality initial solutions for phase two. For the problem instances we consider, setting  $H$  larger than 10 tends to negate the compu-

tational benefit of the first phase, i.e., the exact take-all calculation of Mendoza et al. (2008, 2010, 2011) is faster than estimating the take-all value via simulation. In §6.2, we present additional discussion on the effect of parameter  $H$ .

---

**Algorithm 2** Simulated Annealing

---

```

1: repeat
2:   repeat
3:     Randomly select neighborhood  $U(\cdot) \in \{1\text{-shift}, 2\text{-shift}, 3\text{-shift}, \text{reverse}, \text{exchange}\}$ 
4:     Randomly select  $\pi_{\text{neigh}} \in U(\pi_{\text{curr}})$ 
5:      $\pi_{\text{curr}} \leftarrow \pi_{\text{neigh}}$  with probability  $\exp\{-((u(\pi_{\text{neigh}}) - u(\pi_{\text{curr}})))^+ / \tau\}$ 
6:     if  $u(\pi_{\text{curr}}) < u(\pi_{\text{best}})$  then
7:        $\pi_{\text{best}} \leftarrow \pi_{\text{curr}}$ 
8:   until  $\beta$  iterations
9:    $\tau \leftarrow \tau \times 0.99$ 
10: until At least  $\gamma$  iterations and not update  $\pi_{\text{best}}$  for  $\lambda$  successive iterations

```

---

Algorithm 2 outlines the steps required to execute a single phase of our simulated annealing procedure. Three cyclic orders are maintained –  $\pi_{\text{best}}$ ,  $\pi_{\text{curr}}$ , and  $\pi_{\text{neigh}}$  – corresponding to the best-found, the current, and a neighbor cyclic order, respectively. An iteration of the inner loop begins on line 3 by randomly selecting one of five neighborhood structures proposed by Goodson et al. (2012): 1-shift, 2-shift, 3-shift, reverse, or exchange. Denoting the resulting neighborhood structure by  $U(\cdot)$ , line 4 randomly selects a cyclic order  $\pi_{\text{neigh}}$  in  $U(\pi_{\text{curr}})$ , the neighborhood of the current cyclic order. Line 5 probabilistically updates  $\pi_{\text{curr}}$  with  $\pi_{\text{neigh}}$ , where  $(y)^+ = y$  if  $y > 0$  and 0 otherwise,  $\tau$  is the current temperature, and  $u(\cdot)$  is a function representing the evaluation of a cyclic order (see Goodson et al. (2012) for details) and returns the take-all value of the associated route set. Lines 6 and 7 update the best-found cyclic order. The inner loop terminates after  $\beta$  iterations. The outer loop implements a geometric cooling schedule and terminates after at least  $\gamma$  iterations and without having updated the best-found cyclic order for  $\lambda$  successive iterations.

The first phase initializes  $\pi_{\text{best}}$  and  $\pi_{\text{curr}}$  with a radial ordering of customers about the depot (i.e., customers are ordered by their polar angle with respect to a horizontal axis through the depot). The

second phase initializes  $\pi_{\text{best}}$  and  $\pi_{\text{curr}}$  with the best cyclic order returned in the first phase. In the first phase, we set the termination criteria via  $\gamma = 100$  and  $\lambda = 20$ . In the second phase, we set  $\gamma = 200$  and  $\lambda = 100$ . Phase one begins with an initial temperature of 10 and phase two begins with an initial temperature of 5. Based on computational tests that explore a wide set of parameters, we observe these parameters to perform well for the MCVRPSD instances considered in this paper. We vary the inner-loop iteration counter  $\beta$  as part of our computational experiments in §6.

## 5.2 Cyclic-Order Updating Procedure

As discussed by Goodson et al. (2012), evaluation of a cyclic order is a two-step process. First, a set of candidate routes is generated. Second, the optimal routing solution with respect to the set of candidate routes is identified. When generating a neighbor cyclic order from a current cyclic order in a local search procedure, Goodson et al. (2012) show the respective sets of candidate routes may contain many of the same routes. To avoid the unnecessary task of generating these routes again, Goodson et al. (2012) develop an updating procedure to obtain the set of candidate routes for a neighbor cyclic order by updating the set of candidate routes for the current cyclic order.

Goodson et al. (2012) employ their updating scheme to reduce the computational expense of a local search procedure for a single-compartment VRPSD. In their single-compartment problem, the primary operational constraint is a restriction on the expected demand assigned to an a priori vehicle route. The cyclic-order updating procedure of Goodson et al. (2012) exploits the fact that the expected load assigned to a route does not decrease when a customer is inserted into the route, regardless of the insertion point. In the MCVRPSD we consider in this paper, the main constraint is a limit on expected duration of an a priori route. Because the a priori MCVRPSD policy does not allow preemptive capacity replenishment, inserting a customer in an a priori route can lead to opportune route failures that decrease the expected route duration. Consequently, the cyclic-order updating procedure of Goodson et al. (2012) requires modification.

The example in Figure 1 illustrates how the expected duration of an a priori route can decrease when an additional customer is inserted. The example displays a depot (node 0) and four customers



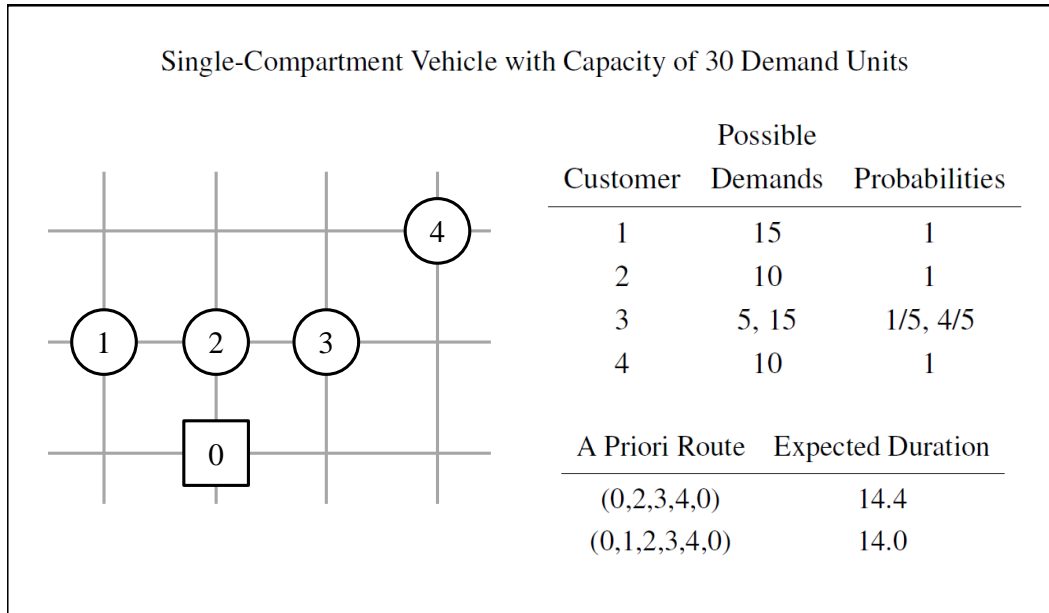


Figure 1: Effect of Customer Insertion on Expected Route Duration

(nodes 1, 2, 3, and 4) with Manhattan distances defined by a unit grid. The table in the top-right portion of Figure 1 indicates that demands for customers 1, 2, and 4 are deterministic and that demand for customer 3 is stochastic. The table in the bottom-right portion of Figure 1 displays the expected duration of two a priori routes for a single-compartment vehicle with a capacity of 30 demand units. When demand at customer 3 is 5 units, the duration of route (0, 2, 3, 4, 0) is 8; when demand at customer 3 is 15 units, a route failure occurs at customer 4, resulting in a duration of 16. Route (0, 1, 2, 3, 4, 0) experiences a route failure at customer 3 regardless of the demand realization, resulting in durations of 14. Thus, the expected duration decreases from 14.4 to 14.0 when customer 1 is added to the beginning of the visit sequence.

To account for the possibility of a decrease in route duration when additional customers are inserted into an a priori route, we modify Proposition 1 of Goodson et al. (2012) by removing the latter portion of the proposition. Proposition 3 displays the modification, which uses the cyclic-order notation employed by Goodson et al. (2012). We modify the updating procedures accordingly by removing the “else break” statements in Algorithms 2, 3, and 4 of Goodson et al. (2012). Specifically, in Algorithm 2 of Goodson et al. (2012), we remove lines 15, 16, 26, 27, 37, and 38; in Algorithm 3 of Goodson et al. (2012), we remove lines 22 and 23; and in Algorithm 4 of

Goodson et al. (2012), we remove lines 19, 20, 30, and 31. Removing the “else break” statements potentially requires additional iterations of the associated loops. However, the remaining lines in each loop require only a check of the cardinality of certain portions of the candidate route set, an  $O(1)$  operation. Thus, the modified updating procedure does not experience a significant slow down due to additional loop iterations.

**Proposition 3** (Modified Updating Procedure). *For all  $i, j \in \{1, \dots, n\}$ , if  $|\mathcal{R}(\pi(i))| > \text{dist}(i, j)$ , then  $|\Lambda(\pi(i), \pi(j))| = |\mathcal{R}(\pi(i))| - \text{dist}(i, j)$ .*

## 6 Computational Results

We perform two sets of computational experiments. Problem instances are described in §6.1. In §6.2, we explore the computational behavior of the exact MCVRPSD a priori policy evaluation procedure of §4.1 and examine the performance of the simulation-evaluation procedure proposed in §4.2. In §6.3, we study how our cyclic-order-based simulated annealing procedure performs relative to methods from the literature. As discussed in §5, because the literature does not report exact solution values for the benchmark problem instances, and to ensure a fair comparison, our simulated annealing procedure evaluates a priori policies via the take-all approximation. We implement our procedures in C++ and execute all computational experiments on 2.8GHz Intel Xeon processors with 12-48GB of RAM running the CentOS 5.3 operating system.

### 6.1 Problem Instances

We perform computational experiments on the set of 180 benchmark problems employed by Mendoza et al. (2010, 2011). The problems are composed of three sets of 60 with  $N = 50$ ,  $N = 100$ , and  $N = 200$  customer instances, respectively. Customers are spread randomly over a Euclidian space. Each customer has demand for  $P = 3$  products. Demand for product  $p$  at customer  $n$  is normally distributed with a mean value chosen randomly from the range  $[10, 100]$ . In each set of 60 problems, half of the instances set the standard deviation such that the coefficient of variation is 0.1 and half of the instances set the standard deviation such that the coefficient of variation is 0.3. The

duration limit for a given problem is set to  $L = \alpha \cdot \max_{n \in \mathcal{N} \setminus \{0\}} t(0, n)$ , where  $\alpha$  is chosen randomly from the range  $[3, 4]$ . Compartment capacity  $Q_p$  for product  $p$  is set to  $10 \cdot \max_{n \in \mathcal{N} \setminus \{0\}} \mathbb{E}[D_{n,p}]$ , rounded to the nearest integer (this corrects a minor oversight in Mendoza et al. (2010, 2011)). Setting capacities in this fashion guarantees the ratio of compartment capacity to each customer demand is at least 10, thereby ensuring the take-all scheme yields a very accurate approximation of the expected cost of an a priori policy (Mendoza et al., 2008).

## 6.2 Exact Evaluation versus Simulation

To study the computational requirements of the exact MCVRPSD a priori policy evaluation procedure of §4.1, we apply Algorithm 1 to a priori routes with various numbers of products, locations, and support sizes. We allow the number of products  $P$  to take on values in the set  $\{1, 2, 3, 4, 5, 6\}$ . We consider route sizes  $|v|$  in the set  $\{3, 4, 5, 6, 7\}$ . Because the depot is the first and last location, these routes serve 1, 2, 3, 4, and 5 customers, respectively. Each customer on the route has the same support for each demand and for each product. Support sizes  $|\mathcal{S}(D_{n,p})|$  take on values in the set  $\{1, 3, 5\}$ . These parameter combinations lead to 90 different a priori routes. For each route, we calculate the exact policy value via Algorithm 1 and estimate the policy value via equation (9) with the number of randomly generated demand vectors  $H$  taking on values in the set  $\{10, 100, 1000, 10000, 100000\}$ . In our implementation, expanding parameters beyond these values is often not possible due to Algorithm 1’s demands on computer memory.

We perform our experiments on problem instance number 1 of Mendoza et al. (2010, 2011); Appendix C identifies this problem instance. We use only the customer locations specified in the problem. An a priori route with  $x$  customers is the sequence  $v = (v_0 = 0, v_1 = 1, v_2 = 2, \dots, v_x = x, v_{x+1} = 0)$ . We set all compartment capacities to 20 demand units. Customer demands for each product follow a discrete uniform distribution with mean 10 and support in the range  $[5, 15]$ . Specifically, for sizes of 1, 3, and 5, we set customer demand supports for each product to  $\{10\}$ ,  $\{5, 10, 15\}$ , and  $\{5, 7.5, 10, 12.5, 15\}$ , respectively.

Tables 1, 2, and 3 display the computing time required for various numbers of products, support sizes, and route sizes, respectively. The first column of each table lists the respective parameters.

Table 1: Change in Computing Time with Number of Products

$P$	CPU Seconds	
	Exact	Estimate ( $H=100,000$ )
1	0.00	0.72
2	0.00	0.71
3	0.01	0.72
4	0.25	0.72
5	7.24	0.75
6	218.83	0.73

The second and third columns of each table show the number of CPU seconds, averaged across all other parameters, required to exactly calculate and estimate the value of MCVRPSD a priori policies. Tables 1 and 2 support Proposition 2: the computation required to execute Algorithm 1 is exponential in the number of products and the size of the supports. Table 3 demonstrates computation time also increases with route size, although not as dramatically.

Table 2: Change in Computing Time with Support Size

$ \mathcal{S}(D_{n,p}) $	CPU Seconds	
	Exact	Estimate ( $H=100,000$ )
1	0.00	0.72
3	0.14	0.74
5	113.02	0.72

Because optimization routines often evaluate large numbers of MCVRPSD a priori policies, utilizing the exact evaluation procedure of Algorithm 1 is only practical for small problem instances where the result can be obtained in a reasonable amount of time. For problems of practical interest, policy value estimation provides a computationally viable alternative. In contrast to the exact evaluation procedure, Tables 1, 2, and 3 suggest computing time required to estimate policy values is nearly constant across product number, route size, and support size. Although in principle

Table 3: Change in Computing Time with Route Size

$ v $	CPU Seconds	
	Exact	Estimate ( $H=100,000$ )
3	0.00	0.73
4	3.96	0.71
5	11.38	0.72
6	43.81	0.73
7	129.46	0.74

Table 4: Change in Computing Time with Number of Randomly Generated Demand Vectors

$H$	CPU Seconds
10	0.00
100	0.00
1,000	0.01
10,000	0.07
100,000	0.72

we should observe an increase in CPU time as various parameter values increase, this growth is insignificant compared to the computational overhead required to manage the data structures facilitating the experiment. As demonstrated by Table 4, the number of CPU seconds required to estimate policy values grows linearly with  $H$ , the number of randomly generated demand vectors. Thus, evaluation via simulation is a computationally attractive alternative to exact evaluation.

For a route with size  $|v| = 7$ ,  $P = 6$  products, and customer supports of size  $|\mathcal{S}(D_{n,p})| = 3$ , Figure 2 graphically depicts simulation estimation error relative to the exact policy value and to the take-all approximation. For each value of  $H$ , the error bars represent the 95 percent confidence interval around the estimate. As expected, the estimate becomes more accurate and confidence interval widths decrease with an increase in sample size. Because we use Algorithm 1 to obtain the take-all approximation (modified as described in §5), the computing time is comparable to the time required to calculate the exact value. When convolutions are straightforward to compute, the method of Mendoza et al. (2008, 2009, 2011) quickly obtains the take-all value. For a more in-depth comparison of exact versus approximate values, we refer to Mendoza et al. (2008).

### 6.3 Benchmark Performance

We explore the performance of our cyclic-order simulated annealing procedure with different values of  $\beta$ , the number of neighbor cyclic orders generated at each temperature in Algorithm 2. For each of the 180 benchmark problems, and for each value of  $\beta$  in the set  $\{1000, 3000, 5000, 7000, 10000, 15000\}$ , we execute our simulated annealing algorithm 10 times.

Figure 3 summarizes the results of our experiments. The horizontal axis displays the values of  $\beta$  we consider, bars are associated with the left vertical axis, and lines are associated with the right vertical axis. We aggregate results by 50-, 100-, and 200-customer problem instances. For a given problem instance and value of  $\beta$ , we calculate the percent decrease in cost of our average solution value across 10 runs over the value of the best-found solution of Mendoza et al. (2010, 2011). We also calculate the percent decrease in cost of the best solution value across 10 runs over the value of the best-found solutions of Mendoza et al. (2010, 2011). The percent decreases in cost over best-known solution values are displayed in the bar series. Two numbers are displayed over

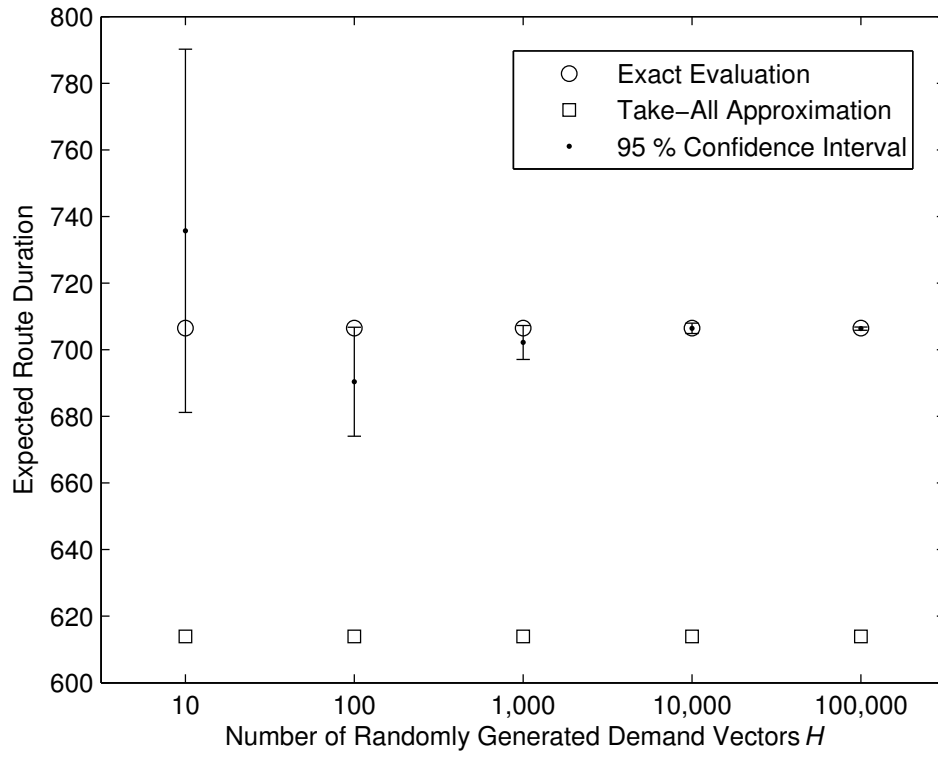


Figure 2: Evaluation via Simulation with  $|v| = 7$ ,  $P = 6$ , and  $|\mathcal{S}(D_{n,p})| = 3$

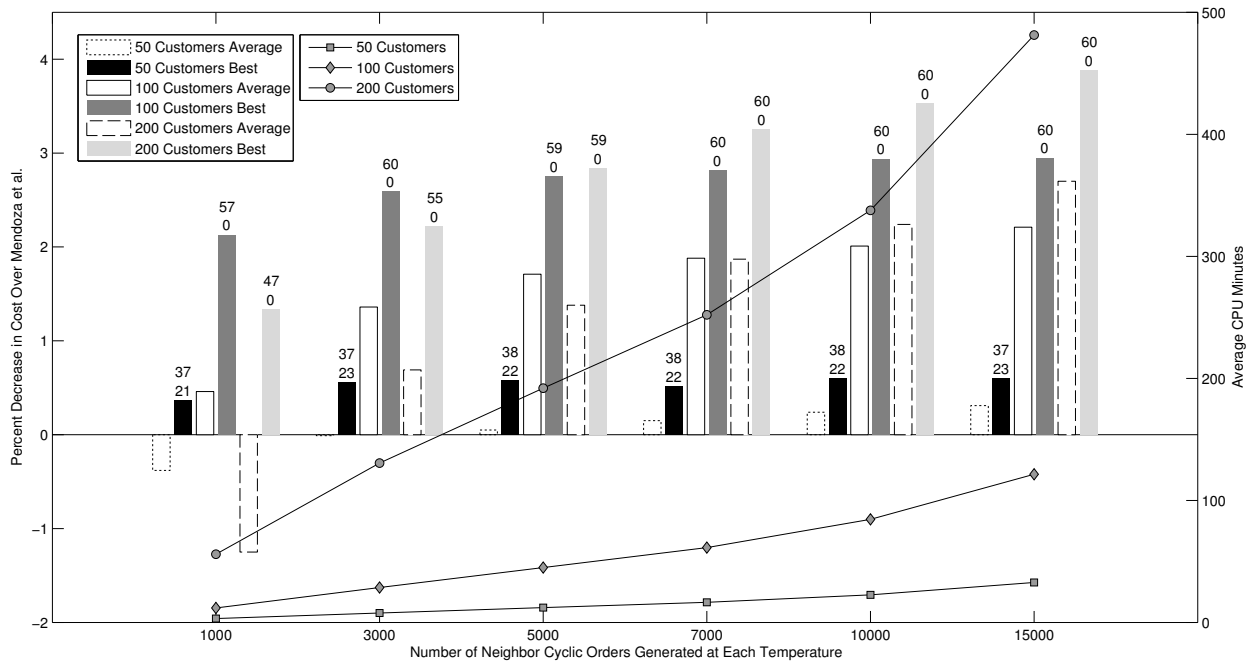


Figure 3: Aggregate Results of Computational Experiments

each member of the “best” bar series. The top number indicates the number of problem instances, out of 60, for which our best-found solution value improves upon the best-found solution value of Mendoza et al. (2010, 2011). The bottom number indicates the number of problem instances, out of 60, for which our best-found solution value matches the best-found solution value of Mendoza et al. (2010, 2011). For example, when  $\beta = 1000$  and  $N = 50$  customers, the average percent decrease in cost of our best-found solution values over the best-found solution values of Mendoza et al. (2010, 2011) is 0.37 percent; our best-found solution values improve upon the best-found solution values of Mendoza et al. (2010, 2011) in 37 of 60 problem instances; our best-found solution values match the best-found solution values of Mendoza et al. (2010, 2011) in 21 of 60 problem instances; and our best-found solution values do not improve upon or match the best-found solution values of Mendoza et al. (2010, 2011) in 2 of 60 problem instances. For a given problem instance and value of  $\beta$ , we calculate the average CPU minutes across 10 runs of our simulated annealing procedure. A point on a line series aggregates this average across 60 problem instances. For example, when  $\beta = 1000$  and  $N = 50$  customers, the average number of CPU minutes required to execute a single run of the simulated annealing algorithm is 3.16. Disaggregated results of our computational experiments are provided in Appendix C.

The bar series in Figure 3 indicate that best and average percent decrease in cost over best-known solutions values increases as  $\beta$  increases. This trend is expected, although improvements over best-known solution values are more pronounced for problem instances with 100 and 200 customers. Moreover, for these larger problems, our cyclic-order simulated annealing procedure improves the best-known solution values on all 120 instances when  $\beta$  is 7000 and larger. Collectively, we improve the best-known solution values on 159 of 180 problem instances and match the best-known solution values on the remaining 21 problem instances. Because our cyclic-order solution procedure is not tailored to the MCVRPSPD, these results, in conjunction with the results of Goodson et al. (2012), support cyclic-order-based local search as an effective solution procedure for a variety of routing problems.

Figure 3 shows computation times tend to increase linearly with  $\beta$ , but at a steeper rate for larger problem instances. Figure 3 also indicates a positive correlation between computation time



and percent decrease over best-known solution values, particularly for larger problem instances. Thus, the choice of parameter  $\beta$  may be used to manage the tradeoff between computation time and solution quality. Because most practitioners use a priori routes over extended periods of time (Savelsbergh and Goetschalchx, 1995), a large upfront investment in computing time may be acceptable. However, the computational requirements of our procedure may be prohibitive if the method is to be used as part of a heuristic scheme to dynamically manage vehicle operations (e.g., Goodson et al. (2013)). In such cases, the construction heuristics of Mendoza et al. (2011) may yield adequate-quality solutions in acceptable CPU times.

## 7 Conclusion

We propose methods to estimate and exactly calculate the value of a priori MCVRPSD policies. Our exact procedure requires exponential time to execute, but leads to a simulation scheme that provides unbiased and consistent estimates of the value of an a priori MCVRPSD policy. We incorporate our estimation procedure into a cyclic-order-based simulated annealing algorithm. Without tailoring our solution procedure to the MCVRPSD, we significantly improve upon the best-known solution values reported in the literature. We also extend the updating procedure for a cyclic order's candidate route set to duration-constrained a priori policies.

A promising direction for future research is to expand upon the classical detour-to-depot a priori policy considered in this paper. Allowing preemptive capacity replenishment is one way to accomplish this. For a given sequence of customers, Pandelis et al. (2012) provide a procedure to obtain the optimal restocking policy. However, Pandelis et al. (2012) do not explore methods to improve the customer sequence to potentially obtain a restocking policy with lower expected cost. Incorporating the method of Pandelis et al. (2012) into a local search procedure may yield policies with lower costs than the policies we obtain in this paper. Vehicle coordination may also result in lower-cost policies. One possibility is to extend the paired-vehicle recourse strategy of Ak and Erera (2007) to the multi-compartment case. Alternatively, allowing multiple vehicles to satisfy demands for different products at a single customer may yield savings. This direction is explored

in the literature for multi-compartment vehicle routing problems with deterministic demands, but has yet to be addressed when demands are stochastic.

It may also be possible to augment our models to capture variation in travel times. Although the addition of stochastic travel time is likely to further complicate the optimization, accounting for such variability may provide a more realistic model. A deterministic alternative to modeling travel times as random variables is to set different travel durations as a function of time of day. The work of Figliozzi (2012) may serve as a baseline for incorporation of time-dependent travel times.

## Acknowledgements

The author appreciates the comments of anonymous reviewers, many of which led to improvements in the manuscript. The author wishes to express appreciation to Jorge Mendoza for helpful conversations and for providing the data sets used in the computational experiments. The author would also like to thank Barrett Thomas for feedback on an early draft of the paper. The author is grateful for the use of a computing cluster provided by Saint Louis University.

## References

- Ak, A. and A. Erera (2007). A paired-vehicle routing recourse strategy for the vehicle routing problem with stochastic demands. *Transportation Science* 41(2), 222–237.
- Beasley, J. (1983). Route first–cluster second methods for vehicle routing. *OMEGA* 11(4), 403–408.
- Doctor, F. and J. Renaud (2000). The column-circular, subsets-selection problem: complexity and solutions. *Computers and Operations Research* 27(4), 383–398.
- Brown, G., C. Ellis, G. Graves, and D. Ronen (1987). Real-time, wide area dispatch of mobil tank trucks. *Interfaces* 17(1), 107–120.

- Campbell, A. and B. Thomas (2008). Challenges and advances in a priori routing. In B. Golden, S. Raghavan, and E. Wasil (Eds.), *The vehicle routing problem: latest advances and new challenges*. New York, NY: Springer.
- Caramia, M. and F. Guerriero (2010). A milk collection problem with incompatibility constraints. *Interfaces* 40(2), 130–143.
- El Fallahi, A., C. Prins, and R. Calvo (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 35(5), 1725–1741.
- Erera, A., J. Morales, and M. Savelsbergh (2010). The vehicle routing problem with stochastic demand and duration constraints. *Transportation Science* 44(4), 474–492.
- Figliozzi, M. (2012). The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E: Logistics and Transportation Review* 48(3), 616–636.
- Gendreau, M., G. Laporte, and R. Séguin (1996). Stochastic vehicle routing. *European Journal of Operational Research* 88, 3–12.
- Goodson, J., J. Ohlmann, and B. Thomas (2012). Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. *European Journal of Operational Research* 217(2), 312–323.
- Goodson, J., J. W. Ohlmann, and B. Thomas (2013). Rollout policies for dynamic solutions to the multi-vehicle routing problem with stochastic demand and duration limits. *Operations Research* 61(1), 138–154.
- Johnson, D., C. Aragon, L. McGeoch, and C. Schevon (1989). Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. *Operations Research* 37(6), 865–892.

- Johnson, D., C. Aragon, L. McGeoch, and C. Schevon (1991). Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning. *Operations Research* 39(3), 378–406.
- Kirkpatrick, S., J. Gelatt, C., and M. Vecchi (1983). Optimization by simulated annealing. *Science* 220(4598), 671–680.
- Laporte, G., F. Louveaux, and L. Van Hamme (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* 50(3), 415–423.
- Mendoza, J., C. B., C. Guéret, A. Medaglia, and N. Velasco (2008). Approximating the expected cost of recourse on a multi-compartment vehicle routing problem with stochastic demands. Technical report, Institut de Recherche en Communications et Cybernétique de Nantes (IRC-CyN).
- Mendoza, J., C. B., C. Guéret, A. Medaglia, and N. Velasco (2009). A simulation-based MOEA for the multi-compartment vehicle routing problem with stochastic demands. In *Proceedings of the 8th Metaheuristics International Conference*.
- Mendoza, J., C. B., C. Guéret, A. Medaglia, and N. Velasco (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research* 37(11), 1886–1898.
- Mendoza, J., C. B., C. Guéret, A. Medaglia, and N. Velasco (2011). Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science* 45(3), 346–363.
- Oppen, J. and A. Løkketangen (2008). A tabu search approach for the livestock collection problem. *Computers & Operations Research* 35(10), 3213–3229.
- Pandelis, D., C. Kyriakidis, and T. Dimitrakos (2012). Single vehicle routing problems with a pre-

- defined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research* 217, 324–332.
- Ryan, D., C. Hjorring, and F. Glover (1993). Extensions of the petal method for vehicle routing. *Journal of the Operational Research Society* 44(3), 289–296.
- Savelsbergh, M. and M. Goetschalchx (1995). A comparison of the efficiency of fixed versus variable vehicle routes. *Journal of Business Logistics* 16(1), 163–187.
- Tatarakis, A. and I. Minis (2009). Stochastic vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research* 197(2), 557–571.
- Teodorović, D. and G. Pavković (1992). A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand. *Transportation Planning and Technology* 16(4), 261–273.
- Toth, P. and D. Vigo (2001). *The vehicle routing problem*. United States: SIAM.
- van der Bruggen, L., R. Gruson, and M. Salomon (1995). Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research* 81(3), 460–473.

## A Notation Guide

Table 5 displays commonly used variables and their definitions.

Table 5: Guide to Primary Notation

Variable	Description
$\mathcal{N} = \{0, 1, \dots, N\}$	Node set
$\mathcal{M} = \{1, 2, \dots, M\}$	Vehicle set
$\mathcal{P} = \{1, 2, \dots, P\}$	Product set
$Q_p$	Compartment for product $p$
$D_{n,p}$	Random demand for product $p$ at customer $n$
$d_{n,p}$	Realization of $D_{n,p}$
$F_{D_{n,p}}(\cdot)$	Distribution function for $D_{n,p}$
$\mathcal{S}(D_{n,p})$	Support for $D_{n,p}$
$t(n, n')$	Time to travel from location $n$ to location $n'$
$L$	Route duration limit
$v$	A set of a priori routes or a single a priori route
$A_{v_i}$	Random time of initial arrival to customer $v_i$
$a_{v_i}$	Realization of $A_{v_i}$
$C_{v_i,p}$	Random capacity of compartment $p$ upon initial arrival to customer $v_i$
$c_{v_i,p}$	Realization of $C_{v_i,p}$
$H$	Number of randomly generated demand vectors
$\beta$	Number of inner loop simulated annealing iterations

## B Proofs

### Proof of Proposition 1

We first prove equation (7).

$$\begin{aligned} \mathbb{P}\{C_{v_i} = c, A_{v_i} = a\} \\ = \sum_{d'} \sum_{c'} \sum_{a'} \mathbb{P}\{C_{v_i} = c, A_{v_i} = a \mid C_{v_{i-1}} = c', A_{v_{i-1}} = a', D_{v_{i-1}} = d'\} \end{aligned} \quad (10)$$

$$\cdot \mathbb{P}\{C_{v_{i-1}} = c', A_{v_{i-1}} = a', D_{v_{i-1}} = d'\} \quad (11)$$

$$= \sum_{d'} \sum_{c'} \sum_{a'} \mathbf{1}\{c = c_{v_i}(d', c'), a = a_{v_i}(d', c', a')\} \quad (12)$$

$$\cdot \mathbb{P}\{C_{v_{i-1}} = c', A_{v_{i-1}} = a'\} \cdot \prod_{p=1}^P f_{D_{v_{i-1},p}}(d'_p). \quad (13)$$

Equations (10) and (11) hold by the law of total probability. The conditional probability term in (10) is one if  $c = c_{v_i}(d', c')$  and  $a = a_{v_i}(d', c', a')$  and is zero otherwise. Thus, the conditional probability term is equivalent to the indicator function in (12). Equation (13) follows from the independence of demands at customer  $v_{i-1}$ , therefore the joint probability of customer demands can be expressed a product of the marginal probabilities.

We next prove equation (8).

$$\mathbb{E}[A_{v_i}] = \sum_d \sum_c \sum_a \mathbb{E}\{A_{v_i} \mid C_{v_{i-1}} = c, A_{v_{i-1}} = a, D_{v_{i-1}} = d\} \quad (14)$$

$$\cdot \mathbb{P}\{C_{v_{i-1}} = c, A_{v_{i-1}} = a, D_{v_{i-1}} = d\} \quad (15)$$

$$= \sum_d \sum_c \sum_a a_{v_i}(d, c, a) \cdot \mathbb{P}\{C_{v_{i-1}} = c, A_{v_{i-1}} = a\} \cdot \prod_{p=1}^P f_{D_{v_{i-1},p}}(d_p). \quad (16)$$

Equations (14) and (15) hold by the law of iterated expectations. Because the expectation in (14) is conditioned on  $c$ ,  $a$ , and  $d$ , it is given directly by (6), which we denote by  $a_{v_i}(d, c, a)$ . The remainder of (16) follows from the independence of demands at customer  $v_{i-1}$ , therefore the joint probability of customer demands can be expressed a product of the marginal probabilities.

## Proof of Proposition 2

Because customer demands are no larger than  $\bar{D}$ , the maximum number of route failures experienced at any customer is  $\lceil \bar{D}/\underline{Q} \rceil$ . The maximum travel time for one return trip to the depot originating from any customer on route  $v$  is  $\bar{L}$ . Because capacities are assumed to be integer-valued,

the maximum number of capacity vectors is  $\bar{Q} \cdot P$ . Because customer demands, vehicle capacities, and travel times are assumed to be integer-valued, the maximum number of capacity-arrival time combinations at any one customer is  $\bar{L} \cdot \bar{Q} \cdot P \cdot \lceil \bar{D}/\underline{Q} \rceil$ . Thus, the loops in lines 5-7 result in line 8 being executed at most  $(i - 1) \cdot \bar{L} \cdot \bar{Q} \cdot P \cdot \lceil \bar{D}/\underline{Q} \rceil \cdot \bar{S}^P$  times, where  $\bar{S}^P$  is the largest possible cardinality of the joint support  $\mathcal{S}(\cdot)$ . Similarly, the loops in lines 9 and 10 result in line 11 being executed at most  $\bar{L} \cdot \bar{Q} \cdot P \cdot \lceil \bar{D}/\underline{Q} \rceil \cdot \bar{S}^P$  times. Thus, the computational complexity of Algorithm 1 is  $O(i \cdot \bar{L} \cdot \bar{Q} \cdot P \cdot \lceil \bar{D}/\underline{Q} \rceil \cdot \bar{S}^P)$ .

## C Disaggregated Results of Computational Experiments

Table 6 displays the disaggregated solution values for the computational experiments discussed in §6. The first column displays the instance number (#), the number of customers ( $N$ ), the number of products ( $P$ ), and the coefficient of variation for customer demands (CV). The second column displays the best-found solution values of Mendoza et al. (2010, 2011). The remaining 12 columns display, for six different values of  $\beta$ , the best and average solution values across 10 runs of the cyclic-order simulated annealing procedure.



Table 6: Disaggregated Results of Computational Experiments

#/N/P/CV	Mendoza et al.		$\beta = 1000$		$\beta = 3000$		$\beta = 5000$		$\beta = 7000$		$\beta = 10000$		$\beta = 15000$	
	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
1/50/3/0.1	794.71	780.23	785.15	782.72	780.23	785.00	780.23	783.75	780.23	783.75	780.23	780.23	780.23	780.23
2/50/3/0.1	683.44	683.44	696.32	694.08	683.44	687.47	683.44	683.17	680.73	683.17	681.99	686.71	683.44	683.78
3/50/3/0.1	710.09	710.09	718.99	717.87	711.08	711.75	710.09	711.91	710.99	711.91	710.99	712.33	710.99	711.49
4/50/3/0.1	651.49	651.49	662.18	653.36	651.49	657.89	651.49	654.32	651.49	654.32	651.49	653.56	651.49	652.99
5/50/3/0.1	670.42	670.42	672.32	670.65	670.42	670.65	670.42	670.65	670.42	670.65	670.42	670.63	670.42	670.44
6/50/3/0.1	701.00	701.00	707.50	704.27	701.00	708.18	701.00	705.72	701.00	705.72	701.00	703.75	701.00	702.35
7/50/3/0.1	718.33	712.86	717.04	713.76	712.49	713.61	712.49	713.74	712.49	713.74	706.93	713.43	706.93	711.45
8/50/3/0.1	758.02	742.50	754.84	751.69	742.50	750.79	742.50	749.25	742.50	749.25	742.50	745.55	742.50	747.41
9/50/3/0.1	664.26	655.93	667.46	660.69	655.93	661.65	655.93	658.00	655.93	658.00	655.93	656.49	655.93	658.00
10/50/3/0.1	722.63	716.37	724.14	722.00	716.37	720.84	716.37	721.38	716.37	721.38	716.37	722.00	716.37	722.00
11/50/3/0.1	676.87	664.84	672.43	671.84	665.91	671.86	665.91	671.10	665.91	671.10	664.84	667.37	664.84	669.34
12/50/3/0.1	685.96	682.49	684.64	681.42	655.84	681.42	654.28	678.38	682.49	685.15	654.28	672.63	654.28	678.24
13/50/3/0.1	770.84	759.03	782.05	778.59	759.52	778.59	757.18	781.31	758.49	779.06	756.92	769.91	756.92	766.29
14/50/3/0.1	656.50	656.50	657.70	656.50	656.50	656.50	656.50	656.50	656.50	656.50	656.50	656.50	656.50	656.50
15/50/3/0.1	635.48	635.48	638.44	635.62	635.48	635.62	635.48	635.48	635.48	635.48	635.48	635.76	635.48	635.60
16/50/3/0.1	750.91	740.37	749.29	739.04	739.04	744.72	739.04	744.02	737.96	741.79	737.96	743.22	738.19	739.40
17/50/3/0.1	641.46	676.61	677.69	669.37	641.46	673.27	641.46	652.01	641.46	652.01	641.46	666.14	641.46	653.75
18/50/3/0.1	729.39	728.82	732.42	727.96	727.96	731.76	727.96	729.56	725.36	729.81	725.36	728.53	725.36	727.99
19/50/3/0.1	713.42	713.42	717.29	717.09	713.42	713.65	713.42	713.61	713.42	713.61	713.42	713.73	713.42	713.43
20/50/3/0.1	656.28	656.28	657.58	657.44	656.28	656.28	656.28	656.28	656.28	656.28	656.28	656.28	656.28	656.28
21/50/3/0.1	711.02	706.45	709.20	708.31	706.45	708.31	703.10	706.48	703.10	707.07	703.82	705.91	703.10	705.36
22/50/3/0.1	667.11	667.11	674.50	674.36	667.11	667.11	667.11	667.11	667.11	671.84	667.11	667.11	667.11	667.11
23/50/3/0.1	695.15	694.73	700.15	695.42	694.73	695.05	694.73	695.13	694.73	695.13	694.73	694.96	694.73	694.75
24/50/3/0.1	714.81	705.92	711.17	709.37	705.92	708.59	705.92	706.71	705.92	706.71	705.92	709.62	705.92	709.15
25/50/3/0.1	664.13	664.13	664.13	667.58	664.13	664.13	664.13	664.13	664.13	664.13	664.13	664.13	664.13	664.13
26/50/3/0.1	710.04	710.04	722.72	722.27	710.04	716.35	710.04	719.00	710.04	719.00	710.04	712.86	710.04	712.29
27/50/3/0.1	667.76	667.76	668.21	667.76	667.76	667.76	667.76	667.76	667.76	667.76	667.76	667.76	667.76	667.76
28/50/3/0.1	730.72	731.42	732.35	730.72	730.72	731.76	730.72	732.96	730.72	731.47	730.72	731.06	730.72	730.79
29/50/3/0.1	758.87	749.38	754.47	749.38	749.38	750.22	749.38	752.58	749.38	749.90	749.38	750.41	749.38	749.66
30/50/3/0.1	699.17	697.61	703.47	698.55	697.61	698.71	697.61	698.71	697.61	698.88	697.61	698.86	697.61	698.19
31/100/3/0.1	1107.47	1102.12	1119.99	1100.04	1100.04	1111.97	1108.37	1113.38	1100.04	1108.72	1097.54	1111.81	1102.09	1109.89
32/100/3/0.1	1257.81	1243.51	1263.56	1237.00	1237.00	1248.15	1235.73	1245.91	1230.62	1239.68	1232.27	1241.71	1223.29	1235.19
33/100/3/0.1	1126.79	1096.82	1125.49	1095.57	1095.57	1114.81	1092.74	1098.76	1092.74	1104.64	1092.74	1098.66	1092.74	1098.13
34/100/3/0.1	1110.99	1102.12	1123.88	1102.13	1102.13	1110.25	1103.30	1108.42	1101.49	1109.79	1103.12	1111.84	1100.29	1104.13
35/100/3/0.1	1285.88	1281.13	1289.71	1268.57	1268.57	1278.45	1265.99	1278.52	1267.87	1275.45	1266.72	1272.61	1266.08	1273.23
36/100/3/0.1	1302.66	1281.57	1293.11	1266.21	1266.21	1284.16	1272.09	1282.26	1258.14	1274.88	1254.46	1275.84	1269.88	1276.47
37/100/3/0.1	1045.87	1038.62	1049.29	1028.16	1028.16	1046.58	1027.95	1033.38	1030.61	1039.63	1027.95	1033.12	1027.95	1033.37

Continued on next page

Table 6 – Continued from previous page

#N/FP/CV	Mendoza et al.	$\beta = 1000$			$\beta = 3000$			$\beta = 5000$			$\beta = 7000$			$\beta = 10000$			$\beta = 15000$		
		Best	Avg	Best	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	
38/1000/3/0.1	1116.77	1086.96	1097.18	1072.68	1084.70	1071.68	1081.51	1071.68	1081.23	1071.68	1081.23	1071.68	1081.23	1071.62	1076.63	1071.74	1079.12		
39/1000/3/0.1	1103.03	1077.90	1103.64	1074.02	1086.89	1075.28	1089.87	1075.28	1089.87	1075.28	1089.87	1075.28	1089.87	1065.31	1078.13	1071.96	1076.09		
40/1000/3/0.1	1167.53	1154.72	1166.77	1139.21	1151.91	1140.36	1149.70	1140.36	1149.70	1140.36	1149.70	1140.36	1149.70	1140.13	1148.18	1133.98	1145.79		
41/1000/3/0.1	1233.70	1214.07	1235.16	1207.49	1226.19	1209.77	1219.26	1209.77	1219.26	1209.77	1219.26	1209.77	1219.26	1205.59	1216.13	1204.13	1217.10		
42/1000/3/0.1	1215.20	1226.82	1256.68	1214.14	1241.41	1209.82	1232.93	1209.82	1232.93	1209.82	1232.93	1209.82	1232.93	1210.08	1230.80	1212.41	1223.75		
43/1000/3/0.1	1255.63	1197.64	1221.77	1189.49	1211.17	1189.51	1202.21	1189.51	1202.21	1189.51	1202.21	1189.51	1202.21	1183.72	1196.26	1187.31	1194.77		
44/1000/3/0.1	1267.04	1206.95	1228.47	1199.08	1208.85	1200.28	1206.48	1200.28	1206.48	1197.91	1206.95	1197.91	1206.95	1191.33	1203.24	1192.88	1199.51		
45/1000/3/0.1	1026.13	1017.24	1033.32	1014.46	1021.89	1001.71	1019.21	1001.71	1019.21	1000.97	1013.40	1000.97	1013.40	1004.93	1016.77	1002.13	1012.30		
46/1000/3/0.1	1229.92	1203.43	1244.04	1203.52	1229.89	1198.20	1225.95	1198.20	1225.95	1193.06	1214.60	1193.06	1214.60	1192.51	1208.53	1197.67	1207.59		
47/1000/3/0.1	1326.54	1318.27	1330.40	1310.65	1315.97	1311.22	1314.68	1311.22	1314.68	1288.25	1311.89	1288.25	1311.89	1291.71	1311.43	1288.18	1307.79		
48/1000/3/0.1	1316.28	1287.68	1304.76	1279.66	1293.16	1279.77	1291.37	1279.77	1291.37	1280.37	1289.24	1280.37	1289.24	1282.15	1287.13	1276.04	1284.07		
49/1000/3/0.1	1183.00	1142.17	1160.27	1145.79	1157.64	1142.61	1153.66	1142.61	1153.66	1140.65	1146.43	1140.65	1146.43	1140.65	1148.36	1140.65	1147.43		
50/1000/3/0.1	1256.53	1269.37	1293.65	1254.03	1269.74	1252.65	1270.19	1252.65	1270.19	1252.65	1270.84	1252.65	1270.84	1253.52	1264.00	1252.20	1259.32		
51/1000/3/0.1	1140.61	1127.51	1146.56	1116.59	1138.75	1107.61	1127.35	1107.61	1127.35	1110.40	1123.46	1110.40	1123.46	1108.36	1121.29	1107.60	1118.35		
52/1000/3/0.1	1172.26	1139.93	1153.91	1127.49	1145.78	1137.95	1142.42	1137.95	1142.42	1130.74	1138.47	1130.74	1138.47	1125.98	1139.18	1126.87	1136.14		
53/1000/3/0.1	1241.20	1223.93	1238.70	1213.41	1226.20	1205.49	1224.56	1205.49	1224.56	1213.29	1219.15	1213.29	1219.15	1205.11	1221.51	1204.34	1214.75		
54/1000/3/0.1	1185.92	1145.82	1175.88	1150.23	1158.31	1144.72	1155.01	1144.72	1155.01	1139.52	1148.45	1139.52	1148.45	1139.52	1148.82	1139.52	1142.81		
55/1000/3/0.1	1021.34	991.23	1001.04	988.12	995.88	990.50	995.81	990.50	995.81	985.71	989.53	985.71	989.53	988.66	993.86	988.12	992.60		
56/1000/3/0.1	1363.41	1286.57	1314.50	1288.50	1309.88	1278.32	1296.76	1278.32	1296.76	1277.02	1294.85	1277.02	1294.85	1278.88	1296.10	1275.55	1287.25		
57/1000/3/0.1	1198.46	1143.62	1185.17	1149.40	1167.96	1141.87	1157.88	1141.87	1157.88	1146.35	1166.41	1146.35	1166.41	1146.33	1155.92	1142.10	1151.07		
58/1000/3/0.1	1197.72	1160.92	1185.76	1163.03	1183.43	1160.01	1177.41	1160.01	1177.41	1159.83	1168.53	1159.83	1168.53	1161.43	1172.90	1160.13	1171.62		
59/1000/3/0.1	1299.32	1243.46	1267.36	1254.10	1259.58	1237.12	1255.65	1237.12	1255.65	1243.36	1255.97	1243.36	1255.97	1225.91	1253.26	1232.27	1247.89		
60/1000/3/0.1	1200.49	1148.47	1170.00	1130.98	1144.26	1130.98	1139.13	1130.98	1139.13	1131.00	1140.75	1131.00	1140.75	1130.98	1140.12	1131.69	1138.11		
61/2000/3/0.1	1802.16	1757.65	1801.89	1726.84	1753.42	1703.16	1745.66	1703.16	1745.66	1701.22	1724.26	1701.22	1724.26	1696.78	1726.79	1691.13	1708.05		
62/2000/3/0.1	1830.41	1861.02	1895.01	1838.33	1862.20	1824.83	1840.60	1824.83	1840.60	1825.76	1839.33	1825.76	1839.33	1819.58	1840.89	1810.13	1823.98		
63/2000/3/0.1	2033.28	1966.01	2017.82	1966.52	2008.58	1976.65	2010.58	1976.65	2010.58	1945.93	1980.06	1945.93	1980.06	1951.26	1975.67	1921.83	1956.46		
64/2000/3/0.1	2078.73	1980.43	2055.54	1952.90	1990.63	1957.82	1978.37	1957.82	1978.37	1957.91	1978.36	1957.91	1978.36	1934.57	1953.28	1936.40	1952.49		
65/2000/3/0.1	1973.13	1955.65	2026.20	1924.65	1941.10	1918.90	1943.22	1918.90	1943.22	1897.58	1931.06	1897.58	1931.06	1910.91	1927.34	1888.65	1906.87		
66/2000/3/0.1	2354.84	2325.57	2394.08	2280.74	2339.28	2302.29	2325.50	2302.29	2325.50	2263.75	2311.69	2263.75	2311.69	2285.10	2311.66	2256.30	2289.53		
67/2000/3/0.1	1956.55	1909.53	1981.01	1889.73	1933.19	1894.40	1918.19	1894.40	1918.19	1893.18	1907.82	1893.18	1907.82	1883.91	1902.38	1861.99	1885.04		
68/2000/3/0.1	2170.60	2173.58	2255.98	2179.45	2213.54	2133.64	2180.81	2133.64	2180.81	2135.64	2182.30	2135.64	2182.30	2147.04	2164.90	2139.38	2154.14		
69/2000/3/0.1	2173.99	2119.17	2191.91	2115.12	2156.06	2075.26	2119.72	2075.26	2119.72	2090.57	2117.80	2090.57	2117.80	2042.84	2101.59	2079.25	2099.53		
70/2000/3/0.1	1711.10	1614.13	1664.45	1624.32	1637.01	1609.68	1622.47	1609.68	1622.47	1592.44	1608.38	1592.44	1608.38	1586.84	1617.54	1578.95	1609.55		
71/2000/3/0.1	1854.96	1806.04	1866.02	1765.38	1822.39	1763.24	1820.57	1763.24	1820.57	1775.19	1803.71	1775.19	1803.71	1755.96	1801.56	1738.52	1784.66		
72/2000/3/0.1	1927.83	1891.76	1922.54	1860.91	1902.72	1853.12	1879.92	1853.12	1879.92	1849.76	1866.37	1849.76	1866.37	1846.39	1870.76	1838.73	1857.18		
73/2000/3/0.1	2092.02	1987.24	2075.05	1991.05	2037.64	1995.56	2020.67	1995.56	2020.67	1983.33	2010.26	1983.33	2010.26	1963.95	1988.57	1958.87	1989.63		
74/2000/3/0.1	2128.84	2130.41	2212.32	2106.08	2148.52	2089.20	2113.35	2089.20	2113.35	2096.87	2125.38	2096.87	2125.38	2050.50	2101.27	2067.96	2094.21		
75/2000/3/0.1	1893.72	1865.80	1919.54	1816.82	1849.09	1802.26	1837.44	1802.26	1837.44	1809.23	1831.49	1809.23	1831.49	1809.03	1821.35	1784.81	1807.33		

Continued on next page

Table 6 – Continued from previous page

#/N/FP/CV	Mendoza et al.			$\beta = 1000$			$\beta = 3000$			$\beta = 5000$			$\beta = 7000$			$\beta = 10000$			$\beta = 15000$		
	Best	Avg	Best	Best	Avg	Best	Best	Avg	Best	Avg	Best	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
76/2000/3/0.1	2297.20	2210.38	2261.41	2205.51	2219.64	2165.53	2211.60	2153.99	2212.98	2165.49	2195.65	2159.94	2175.70								
77/2000/3/0.1	1616.05	1592.62	1653.78	1587.99	1622.33	1592.62	1619.97	1572.98	1596.21	1567.59	1588.62	1567.13	1589.20								
78/2000/3/0.1	2033.50	1998.62	2048.58	1988.45	2011.10	1957.49	1997.07	1962.43	1986.23	1963.73	1982.48	1946.26	1965.25								
79/2000/3/0.1	2285.51	2338.18	2373.62	2278.10	2396.38	2265.31	2290.32	2220.23	2265.26	2219.60	2245.40	2223.48	2247.59								
80/2000/3/0.1	1951.87	1894.45	1976.76	1873.24	1907.39	1877.63	1908.17	1855.88	1885.71	1842.29	1869.80	1859.53	1875.33								
81/2000/3/0.1	2109.31	2043.14	2117.19	2033.30	2074.64	2016.93	2049.61	2018.90	2051.93	2027.15	2055.66	2022.83	2040.52								
82/2000/3/0.1	1884.15	1866.42	1913.51	1848.95	1877.08	1828.83	1861.84	1810.31	1844.98	1826.06	1837.34	1808.42	1837.12								
83/2000/3/0.1	1809.76	1797.14	1859.70	1777.11	1801.54	1755.17	1794.11	1748.21	1776.93	1743.45	1767.69	1730.77	1756.42								
84/2000/3/0.1	2175.61	2186.10	2241.24	2123.84	2160.55	2134.53	2148.92	2122.92	2141.90	2103.11	2125.84	2085.70	2115.29								
85/2000/3/0.1	2212.21	2133.85	2196.05	2127.19	2150.89	2089.78	2133.37	2066.42	2104.46	2059.75	2092.82	2069.80	2087.90								
86/2000/3/0.1	2047.55	2079.00	2144.35	2078.07	2107.06	2052.61	2086.88	2042.23	2077.66	2036.95	2077.28	2010.23	2050.16								
87/2000/3/0.1	1729.46	1707.94	1743.41	1648.62	1715.63	1671.45	1705.33	1645.95	1690.94	1630.70	1681.50	1640.29	1668.96								
88/2000/3/0.1	2109.23	2101.97	2132.35	2072.77	2106.57	2016.54	2087.62	2035.55	2068.02	2035.26	2065.46	2026.23	2059.50								
89/2000/3/0.1	2114.11	2101.82	2181.91	2082.88	2117.55	2059.73	2091.90	2040.70	2067.23	2036.11	2064.55	2049.04	2073.77								
90/2000/3/0.1	1873.80	1832.83	1895.62	1817.37	1845.75	1808.07	1830.11	1802.29	1827.25	1812.15	1825.52	1791.30	1813.07								
91/500/3/0.3	705.98	695.01	698.25	695.01	695.52	695.01	695.67	695.01	695.40	695.01	695.56	695.01	695.22								
92/500/3/0.3	687.65	687.65	687.65	687.65	688.66	687.65	688.66	687.65	688.02	687.65	687.65	687.65	687.65								
93/500/3/0.3	770.52	765.50	774.66	765.50	770.59	765.50	770.76	765.50	767.53	765.50	768.29	765.50	767.17								
94/500/3/0.3	708.78	705.08	707.88	702.99	705.37	702.99	705.89	702.99	706.37	702.99	707.21	702.99	704.96								
95/500/3/0.3	788.73	781.86	789.69	781.86	788.29	781.86	788.48	781.86	787.04	780.95	787.80	780.95	785.27								
96/500/3/0.3	759.87	759.87	760.33	759.87	762.18	759.87	760.87	759.87	761.26	759.87	760.12	759.87	759.87								
97/500/3/0.3	740.97	737.19	741.10	737.19	740.39	737.19	739.33	737.19	740.09	737.19	738.92	737.19	738.46								
98/500/3/0.3	730.45	730.45	731.46	730.45	730.45	730.45	731.09	730.45	730.45	730.45	730.45	730.45	730.60								
99/500/3/0.3	721.98	721.98	722.04	721.98	721.99	721.98	722.06	721.98	721.98	721.98	722.43	721.98	721.98								
100/500/3/0.3	659.33	659.33	662.36	659.33	659.33	659.33	659.84	659.33	659.33	659.33	659.33	659.33	659.33								
101/500/3/0.3	699.88	692.30	696.69	692.30	696.14	692.30	696.11	692.30	696.91	692.30	697.61	692.30	695.69								
102/500/3/0.3	764.96	761.77	764.56	761.77	764.43	761.77	762.49	761.77	761.79	761.77	762.10	761.77	761.79								
103/500/3/0.3	632.80	632.80	646.99	632.80	634.37	632.80	635.35	632.80	632.80	632.80	632.80	632.80	632.80								
104/500/3/0.3	766.19	761.41	763.40	761.41	761.65	761.41	761.73	761.41	761.50	761.41	761.41	761.41	761.41								
105/500/3/0.3	746.85	739.24	744.01	738.84	739.81	738.84	740.59	739.24	741.23	739.24	739.70	738.84	740.09								
106/500/3/0.3	690.18	690.18	691.87	690.18	690.18	690.18	690.18	690.18	690.18	690.18	690.18	690.18	690.18								
107/500/3/0.3	729.90	729.90	736.37	729.90	730.51	729.90	730.10	729.90	730.87	729.90	730.56	729.90	730.30								
108/500/3/0.3	747.53	743.11	757.94	743.11	748.59	743.11	751.00	743.11	746.40	743.11	743.55	743.11	747.20								
109/500/3/0.3	672.98	672.98	678.33	672.98	673.43	670.53	673.02	672.98	675.04	672.98	672.98	672.98	672.98								
110/500/3/0.3	748.43	748.02	750.43	738.60	745.98	738.60	746.07	738.60	747.27	738.60	744.76	738.60	746.15								
111/500/3/0.3	711.95	708.60	711.37	707.94	709.49	707.94	709.29	707.94	708.74	707.94	708.74	707.94	708.69								
112/500/3/0.3	735.88	726.93	744.59	726.93	737.83	726.93	735.05	726.93	736.20	726.93	738.10	726.93	735.23								
113/500/3/0.3	700.21	700.21	700.21	700.21	700.26	700.21	700.21	700.21	700.21	700.21	700.21	700.21	700.21								

Continued on next page

Table 6 – Continued from previous page

Mendoza et al.	#/N/FP/CV	$\beta = 1000$			$\beta = 3000$			$\beta = 5000$			$\beta = 7000$			$\beta = 10000$			$\beta = 15000$		
		Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg	Best	Avg
	114/500/3/0.3	736.49	734.41	734.36	734.38	734.36	734.36	735.50	734.36	734.36	734.36	734.36	734.36	734.36	734.36	734.36	734.36	734.36	734.36
	115/500/3/0.3	783.61	786.88	783.07	783.98	783.07	783.26	783.57	783.07	783.39	783.07	783.20	783.07	783.20	783.07	783.11	783.07	783.11	783.07
	116/500/3/0.3	773.96	781.84	769.06	779.51	769.06	775.57	775.57	769.06	775.68	769.06	775.68	769.06	775.68	769.06	773.57	769.06	773.57	769.06
	117/500/3/0.3	664.78	665.29	664.07	665.58	664.05	665.21	665.21	664.05	664.93	664.05	664.70	664.05	664.70	664.05	664.70	664.05	664.70	664.05
	118/500/3/0.3	817.56	824.10	817.56	819.32	817.56	818.69	818.69	817.56	819.84	817.56	818.76	817.56	818.76	817.56	817.95	817.56	817.95	817.56
	119/500/3/0.3	651.96	653.44	649.30	650.02	649.30	650.65	650.65	649.30	650.37	649.30	650.09	649.30	650.09	649.30	649.83	649.30	649.83	649.30
	120/500/3/0.3	772.44	769.82	766.97	769.12	766.97	768.19	768.19	766.97	767.99	766.97	767.84	766.97	767.84	766.97	768.79	766.97	768.79	766.97
	121/1000/3/0.3	1255.14	1238.86	1208.32	1226.29	1209.79	1221.44	1221.44	1209.79	1223.88	1216.46	1223.88	1216.46	1223.88	1216.46	1208.91	1216.46	1208.91	1216.46
	122/1000/3/0.3	1114.09	1116.37	1092.53	1103.14	1092.53	1098.69	1098.69	1092.53	1092.70	1092.53	1093.28	1092.53	1093.28	1092.53	1089.30	1092.53	1089.30	1092.53
	123/1000/3/0.3	1206.87	1193.80	1176.42	1182.85	1173.93	1181.46	1181.46	1173.93	1181.49	1176.42	1181.96	1176.42	1181.96	1173.93	1177.67	1176.42	1177.67	1176.42
	124/1000/3/0.3	1262.49	1244.34	1229.27	1249.64	1231.37	1243.17	1243.17	1231.37	1244.98	1229.27	1244.98	1229.27	1244.98	1229.27	1234.98	1229.27	1234.98	1229.27
	125/1000/3/0.3	1237.69	1217.39	1211.22	1223.25	1213.31	1225.41	1225.41	1213.31	1221.62	1211.22	1215.01	1211.22	1215.01	1211.22	1212.87	1211.22	1212.87	1211.22
	126/1000/3/0.3	1303.00	1285.33	1259.75	1282.80	1261.24	1280.19	1280.19	1261.24	1269.14	1259.75	1267.66	1259.75	1267.66	1259.75	1275.60	1259.75	1275.60	1259.75
	127/1000/3/0.3	1111.24	1085.55	1084.53	1101.50	1084.66	1089.10	1089.10	1084.66	1092.86	1084.53	1087.15	1084.53	1087.15	1084.07	1084.33	1084.07	1084.33	1084.07
	128/1000/3/0.3	1387.74	1346.49	1333.53	1350.35	1333.47	1346.64	1346.64	1333.47	1343.97	1332.22	1343.25	1332.22	1343.25	1332.22	1339.70	1332.22	1339.70	1332.22
	129/1000/3/0.3	1195.46	1186.07	1168.66	1183.91	1162.53	1179.59	1179.59	1162.53	1177.64	1162.53	1177.53	1162.53	1177.53	1162.53	1172.47	1162.53	1172.47	1162.53
	130/1000/3/0.3	1348.81	1319.32	1317.47	1334.46	1320.85	1328.82	1328.82	1320.85	1328.04	1322.63	1328.17	1322.63	1328.17	1311.37	1319.95	1311.37	1319.95	1311.37
	131/1000/3/0.3	1144.99	1124.10	1109.09	1131.09	1109.27	1133.98	1133.98	1109.27	1133.32	1113.04	1130.28	1113.04	1130.28	1108.83	1132.86	1108.83	1132.86	1108.83
	132/1000/3/0.3	1065.59	1043.08	1038.46	1046.65	1040.63	1047.49	1047.49	1040.63	1043.91	1037.63	1041.44	1037.63	1041.44	1038.43	1042.94	1038.43	1042.94	1038.43
	133/1000/3/0.3	1073.22	1060.35	1058.89	1069.31	1055.35	1067.82	1067.82	1055.35	1064.98	1046.08	1069.87	1046.08	1069.87	1055.35	1057.51	1046.08	1057.51	1046.08
	134/1000/3/0.3	1242.15	1243.94	1235.67	1249.54	1233.74	1247.05	1247.05	1233.74	1242.15	1237.89	1240.69	1237.89	1240.69	1233.48	1237.87	1233.48	1237.87	1233.48
	135/1000/3/0.3	1136.82	1123.56	1118.48	1125.53	1109.19	1123.05	1123.05	1109.19	1117.33	1101.30	1121.77	1101.30	1121.77	1102.59	1114.10	1102.59	1114.10	1102.59
	136/1000/3/0.3	1293.94	1285.69	1281.74	1297.53	1272.55	1287.83	1287.83	1272.55	1283.83	1277.80	1286.50	1277.80	1286.50	1269.97	1284.23	1269.97	1284.23	1269.97
	137/1000/3/0.3	1109.23	1080.92	1078.22	1095.26	1073.93	1083.27	1083.27	1073.93	1089.33	1078.01	1081.61	1078.01	1081.61	1072.20	1080.80	1072.20	1080.80	1072.20
	138/1000/3/0.3	1075.20	1029.56	1034.31	1040.78	1029.40	1036.47	1036.47	1029.40	1039.14	1029.40	1037.56	1029.40	1037.56	1029.74	1041.77	1029.74	1041.77	1029.74
	139/1000/3/0.3	1186.45	1156.56	1154.31	1170.02	1153.74	1163.99	1163.99	1153.74	1158.91	1154.69	1160.76	1154.69	1160.76	1152.57	1160.30	1152.57	1160.30	1152.57
	140/1000/3/0.3	1265.67	1195.87	1193.88	1201.07	1178.53	1195.00	1195.00	1178.53	1200.51	1197.26	1196.35	1197.26	1196.35	1178.19	1192.75	1178.19	1192.75	1178.19
	141/1000/3/0.3	1207.86	1167.83	1168.94	1193.39	1167.83	1186.11	1186.11	1167.83	1179.45	1168.78	1184.71	1168.78	1184.71	1167.83	1184.30	1167.83	1184.30	1167.83
	142/1000/3/0.3	1233.56	1228.17	1227.11	1239.24	1223.25	1234.61	1234.61	1223.25	1234.27	1222.36	1231.21	1222.36	1231.21	1223.18	1228.67	1223.18	1228.67	1223.18
	143/1000/3/0.3	995.97	990.31	989.74	995.44	989.74	996.95	996.95	989.74	997.11	989.74	994.95	989.74	994.95	989.74	992.54	989.74	992.54	989.74
	144/1000/3/0.3	1178.13	1172.38	1168.74	1174.03	1162.12	1178.06	1178.06	1162.12	1174.07	1163.34	1171.04	1163.34	1171.04	1171.41	1172.78	1171.41	1172.78	1171.41
	145/1000/3/0.3	1253.00	1214.08	1211.21	1235.00	1214.56	1227.94	1227.94	1214.56	1227.10	1209.75	1222.45	1209.75	1222.45	1207.20	1217.42	1207.20	1217.42	1207.20
	146/1000/3/0.3	1024.36	1010.85	1011.33	1025.37	1009.98	1022.78	1022.78	1009.98	1016.77	1008.03	1016.64	1008.03	1016.64	1006.74	1018.92	1006.74	1018.92	1006.74
	147/1000/3/0.3	1398.16	1351.76	1375.89	1360.58	1347.43	1355.14	1355.14	1347.43	1355.65	1347.43	1350.68	1347.43	1350.68	1347.43	1350.02	1347.43	1350.02	1347.43
	148/1000/3/0.3	1319.92	1285.49	1274.69	1297.88	1278.60	1298.12	1298.12	1278.60	1296.75	1276.39	1290.47	1276.39	1290.47	1274.55	1293.53	1274.55	1293.53	1274.55
	149/1000/3/0.3	1195.02	1143.95	1145.75	1160.46	1145.54	1158.46	1158.46	1145.54	1162.73	1148.50	1155.15	1148.50	1155.15	1143.27	1151.71	1143.27	1151.71	1143.27
	150/1000/3/0.3	1086.36	1069.87	1072.69	1093.58	1068.72	1082.54	1082.54	1068.72	1077.31	1067.79	1077.22	1067.79	1077.22	1068.06	1071.90	1068.06	1071.90	1068.06
	151/2000/3/0.3	2229.39	2177.66	2160.14	2193.31	2136.12	2177.37	2177.37	2136.12	2183.49	2147.36	2164.80	2147.36	2164.80	2145.93	2162.34	2145.93	2162.34	2145.93

Continued on next page

Table 6 – Continued from previous page

#/N/IP/CV	$\beta = 1000$			$\beta = 3000$			$\beta = 5000$			$\beta = 7000$			$\beta = 10000$			$\beta = 15000$		
	Best	Avg	Best	Best	Avg	Best	Best	Avg	Best	Best	Avg	Best	Avg	Best	Best	Avg	Best	Avg
152/200/3/0.3	2324.48	2261.87	2301.95	2258.45	2285.78	2247.99	2273.42	2219.38	2253.97	2228.97	2252.53	2226.15	2245.89					
153/200/3/0.3	1672.86	1658.65	1739.76	1662.13	1680.83	1653.35	1674.89	1654.44	1669.88	1643.71	1660.01	1642.56	1655.98					
154/200/3/0.3	1877.88	1848.19	1890.63	1827.90	1844.10	1823.69	1838.18	1818.06	1834.81	1800.95	1823.17	1804.52	1822.77					
155/200/3/0.3	1714.89	1725.06	1756.69	1706.40	1724.68	1707.04	1721.59	1702.33	1715.36	1681.49	1709.64	1692.16	1708.31					
156/200/3/0.3	2063.66	2036.42	2085.74	2034.07	2065.79	2011.38	2040.25	1988.58	2028.21	2013.00	2030.65	2004.78	2029.18					
157/200/3/0.3	2064.84	2071.53	2115.49	2043.92	2076.64	2046.82	2064.52	2024.03	2047.84	1986.75	2042.19	1986.24	2031.83					
158/200/3/0.3	2304.87	2290.72	2327.16	2255.84	2293.00	2255.30	2286.51	2243.82	2272.87	2245.25	2262.19	2237.58	2259.84					
159/200/3/0.3	1858.81	1816.72	1845.96	1805.37	1828.06	1791.41	1809.46	1768.31	1810.19	1782.00	1802.47	1765.06	1791.68					
160/200/3/0.3	2302.03	2243.01	2299.44	2238.49	2285.71	2231.67	2268.32	2211.20	2253.90	2227.90	2252.88	2202.56	2230.82					
161/200/3/0.3	2377.83	2382.60	2420.71	2359.16	2394.18	2326.69	2359.44	2317.73	2354.90	2304.89	2335.27	2299.15	2331.57					
162/200/3/0.3	2039.44	2032.12	2071.58	2022.28	2063.32	1992.08	2026.34	1985.44	2014.64	1992.24	2012.69	1960.56	2000.94					
163/200/3/0.3	2530.78	2517.13	2559.86	2493.31	2525.67	2464.00	2517.45	2461.61	2486.77	2435.34	2480.53	2448.65	2458.89					
164/200/3/0.3	1819.30	1781.65	1842.70	1758.56	1782.55	1757.95	1772.99	1752.51	1766.11	1749.58	1759.54	1747.19	1755.17					
165/200/3/0.3	2211.05	2225.85	2273.15	2216.95	2254.71	2187.08	2228.32	2199.83	2215.79	2198.70	2222.28	2181.21	2196.12					
166/200/3/0.3	2166.74	2123.20	2165.59	2133.87	2146.00	2111.56	2127.27	2114.19	2137.10	2085.49	2118.98	2088.75	2103.94					
167/200/3/0.3	1745.57	1751.33	1790.86	1739.98	1760.25	1729.57	1756.43	1725.40	1756.27	1723.33	1744.97	1718.49	1739.03					
168/200/3/0.3	1733.99	1703.10	1759.65	1703.20	1718.50	1700.28	1716.85	1687.23	1712.44	1690.11	1706.44	1687.89	1696.57					
169/200/3/0.3	2093.52	2066.99	2109.00	2059.78	2085.04	2022.44	2062.18	2033.07	2053.13	2010.02	2052.89	2014.64	2035.27					
170/200/3/0.3	2114.49	2061.35	2125.15	2059.34	2094.49	2047.40	2071.69	2051.96	2072.21	2006.29	2057.99	2011.50	2049.86					
171/200/3/0.3	2206.53	2229.72	2269.53	2207.32	2237.53	2191.45	2213.90	2162.32	2209.06	2160.31	2189.03	2155.28	2193.72					
172/200/3/0.3	2198.16	2167.87	2204.94	2143.99	2175.38	2149.14	2177.72	2134.97	2161.51	2134.39	2147.72	2118.82	2142.00					
173/200/3/0.3	2393.52	2391.48	2433.00	2374.52	2402.82	2369.44	2395.52	2352.89	2376.90	2322.50	2366.30	2332.57	2352.34					
174/200/3/0.3	1779.67	1763.40	1778.26	1735.57	1755.25	1722.47	1739.90	1723.68	1742.72	1714.94	1731.96	1713.39	1737.73					
175/200/3/0.3	1837.94	1813.77	1868.02	1795.04	1824.71	1789.19	1817.23	1798.67	1811.75	1791.48	1802.45	1772.72	1802.34					
176/200/3/0.3	1816.40	1824.67	1851.49	1785.01	1820.09	1769.85	1809.30	1777.02	1799.77	1787.37	1794.31	1752.49	1782.32					
177/200/3/0.3	2654.61	2632.39	2673.73	2619.07	2643.98	2604.60	2630.26	2580.64	2612.78	2599.52	2619.25	2587.03	2597.29					
178/200/3/0.3	2261.55	2245.18	2286.53	2250.09	2277.87	2212.50	2245.25	2201.57	2241.30	2203.83	2233.55	2196.47	2219.77					
179/200/3/0.3	2166.89	2129.65	2175.17	2120.49	2146.28	2121.21	2135.67	2112.33	2128.80	2108.91	2117.44	2081.86	2102.57					
180/200/3/0.3	1874.54	1844.15	1913.55	1824.41	1865.35	1833.24	1853.62	1816.36	1840.45	1833.52	1848.73	1807.48	1842.01					